Technische
Universität
Braunschweig

Establishing Trust in Heterogeneous Networks

Von der
Carl-Friedrich-Gauß-Fakultät
der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung des Grades eines

Doktoringenieurs (Dr.-Ing.)

genehmigte Dissertation
(kumulative Arbeit)

von **Dominik Schürmann**
geboren am **1987-03-06**
in **Hildesheim**

Eingereicht am:	**2018-02-28**
Disputation am:	**2018-04-25**
1. Referent:	**Prof. Dr. Lars Wolf**
2. Referent:	**Prof. Dr. Stephan Sigg**

(2018)

Kurzfassung

In den vergangenen Jahren wurden eingebettete Computer in immer neuen Gebieten eingesetzt und vernetzt, wodurch eine Vielzahl heterogener Netze entstand. Viele Netzwerke nutzen Kontextinformationen, um Aufgaben zu automatisieren oder die tägliche Arbeit zu vereinfachen. Leider führt dies zu einer erhöhten Übermittlung personenbezogener Informationen. Traditionell werden zentrale Autoritäten zur Verfügung gestellt, die das Vertrauen zwischen Netzwerkentitäten herstellen und Man-in-the-Middle-Angriffe verhindern. In der gegenwärtigen Netzwerklandschaft ist dies aber nicht mehr ausreichend. Zentrale Autoritäten wurden kompromittiert und sind nicht in der Lage, die Vertrauenswürdigkeit der ausgetauschten Informationen zu bewerten. In der vorliegenden Arbeit werden alternative Wege der Vertrauensbildung vorgestellt. Als direkte ad-hoc Möglichkeit sind Trust Anchors vorgesehen. Da diese auf Besonderheiten des Einsatzgebietes beruhen, wurden netzwerkabhängige Verfahren entwickelt und analysisiert. Die vorliegende Arbeit gliedert sich in drei Kapitel: Physical Trust Anchors, Biometric Trust Anchors und Usability.

Physical Trust Anchors werden in Fahrzeugnetzwerken eingesetzt, um langfristige Reputationsbewertungen zu ermöglichen. Als konkreter Anwendungsfall wird die Parkplatzsuche mit Hilfe von Delay-Tolerant Networks gelöst. Die Informationen über freie Parkplätze werden durch Fahrzeuge verteilt und anhand von Reputationsbewertungen auf ihre Vertrauenswürdigkeit hin bewertet. Weiterhin wird ein Handover zwischen Fahrzeugen vorgestellt um schlecht ausgebaute Ladeinfrastruktur sinnvoll auszulasten. Hierzu werden Rechnungen ad-hoc zwischen Fahrzeugen aufgeteilt, wobei die Kosten für das Umparken von Fahrzeugen mit einbezogen sind.

Biometric Trust Anchors können einen natürlichen Weg darstellen, um Vertrauen zwischen Geräten herzustellen. Das ZRTP-Protokoll ermöglicht dies, indem Teilnehmer die Stimmen der anderen erkennen und eine kurze Zeichenkette vergleichen. Angriffe auf ZRTP wurden analysiert und Implementierungsprobleme aufgedeckt. Dazu gehört auch ein neuartiger Angriff namens ‚Shared Man-in-the-Middle'. Für Body Area Networks wird das BANDANA-Protokoll für Device-to-Device Pairings vorgestellt. Hier wird der Gang des Menschen als Biometric Trust Anchor verwendet. Die Entropie der generierten Schlüssel wird mit der Schlüsselentropie in anderen Protokollen aus der Literatur verglichen und bewertet.

Während einige Trust Anchors automatisiert funktionieren, erfordern viele eine bewusste Benutzerinteraktion. Daher ist Usability ein Hauptziel dieser Forschung. Zur Unterstützung von Security Tokens, wie z.B. Smartcards, auf Smartphones wurde eine Full-Stack-Architektur entwickelt und implementiert. Eine high-level API bietet kryptographische Funktionalitäten über Near Field Communication und abstrahiert von der zugrunde liegenden Komplexität. Die Nutzbarkeit der vorgeschlagenen Architektur wurde in einer Laborstudie evaluiert. Darüber hinaus wurde die klassische Art der Vertrauensbildung durch den Vergleich von Key-Fingerprints untersucht. Dazu wurden existierende Berechnungsmethoden von Key-Fingerprints systematisiert und hinsichtlich ihrer Sicherheitseigenschaften und Benutzerfreundlichkeit verglichen.

Abstract

In recent years, a vast amount of heterogeneous networks emerged from the deployment of devices in novel scenarios. Most networks utilize context information to automate tasks or help in day-to-day activities. Unfortunately, this led to an increased dissemination of personally identifiable information threatening the user's privacy. Traditionally, security architectures provide central authorities to initiate trust between network entities and prevent Man-in-the-Middle attacks. In today's network landscape, this is no longer sufficient. Authorities have been compromised and are not capable of proofing the trustworthiness of exchanged information. In this thesis, alternative ways of establishing trust are introduced. As a direct ad-hoc way of trust establishment, trust anchors are envisioned. As these rely on specifics of the context a network is deployed in, heterogeneous networks are analyzed and adapted to establish trust in a decentralized way. The thesis at hand is divided into three chapters: Physical trust anchors, biometric trust anchors, and usability.

Physical trust anchors are implemented in vehicular networks to establish long-term reputation ratings. As a specific use case, the search for parking space is solved using Delay-Tolerant Networks. Vehicles' information about free parking spaces is disseminated and assessed in regards to their trustworthiness using reputation ratings. To solve the problem of scarce charging infrastructure for electric vehicles, the handover between vehicles is incentivized by splitting charging bills. This results in a cooperative ad-hoc approach that incorporates the cost for re-parking.

Biometric trust anchors can provide a natural way to establish trust between devices. The ZRTP protocol allows to establish trust by recognizing the participants' voices and comparing a short number of characters. Attacks against ZRTP have been analyzed and implementation issues uncovered. This includes a novel attack called 'Shared Man-in-the-Middle'. For Body Area Networks, the BANDANA protocol for device-to-device pairings is introduced. Here, a human's gait, the unique way how someone walks, is used as a biometric trust anchor. The entropy of generated keys is compared with similar schemes from the literature.

While some trust anchors work in an automated fashion, many require conscious user interaction. Thus, usability is a main goal of this research. To support security tokens, such as smart cards, on smartphones, a full-stack architecture has been designed and implemented. A high-level API provides cryptographic functionality over Near Field Communication abstracting away the underlying complexity. The usability of the proposed architecture has been evaluated in a lab study. Furthermore, the traditional way of establishing trust via comparison of key-fingerprints is evaluated. For this, key-fingerprint calculations have been systematized and compared in regards to their security properties and usability.

List of Publications

2018

- [1] Arne Brüsch, Ngu Nguyen, Dominik Schürmann, Stephan Sigg, and Lars Wolf. “On the secrecy of publicly observable biometric features: security properties of gait for mobile device pairing”. Submitted to: *IEEE Transactions on Mobile Computing (TMC)*. 2018
- [2] Dominik Schürmann, Arne Brüsch, Ngu Nguyen, Stephan Sigg, and Lars Wolf. “Moves like Jagger: Exploiting variations in instantaneous gait for spontaneous device pairing”. In: *Pervasive and Mobile Computing* 47 (2018), pp. 1 –12
- [3] Dominik Schürmann, Georg von Zengen, Marvin Priedigkeit, and Lars Wolf. “uDTNSec: A security layer with lightweight certificates for Disruption-Tolerant Networks on microcontrollers”. Submitted to: *Annals of Telecommunications*. 2018
- [4] Ngu Nguyen, Caglar Yuce Kaya, Dominik Schürmann, Arne Brüsch, Stephan Sigg, and Lars Wolf. “Demo of BANDANA - Body Area Network Device-to-device Authentication Using Natural gAit”. Accepted for publication at: *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. 2018

2017

- [5] Signe Rüsch, Dominik Schürmann, Rüdiger Kapitza, and Lars Wolf. “Forward Secure Delay-Tolerant Networking”. In: *Proceedings of the 12th Workshop on Challenged Networks*. CHANTS ’17. Snowbird, Utah, USA: ACM, Oct. 2017, pp. 7–12
- [6] Dominik Schürmann, Sergej Dechand, and Lars Wolf. “OpenKeychain: An Architecture for Cryptography with Smart Cards and NFC Rings on Android”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 1.3 (Sept. 2017), 99:1–99:24
- [7] Dominik Schürmann, Fabian Kabus, Gregor Hildermeier, and Lars Wolf. “Wiretapping End-to-End Encrypted VoIP Calls: Real-World Attacks on ZRTP”. in: *Proceedings on Privacy Enhancing Technologies* 2017.3 (July 2017), pp. 4–20
- [8] Dominik Schürmann, Georg von Zengen, Marvin Priedigkeit, and Lars Wolf. “uDTNSec: A Security Layer for Disruption-Tolerant Networks on Microcontrollers”. In: *Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. June 2017, pp. 1–7
- [9] Dominik Schürmann. “Ph.D. Forum: Establishing Trust in Heterogeneous Networks”. In: *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. Mar. 2017, pp. 107–108
- [10] Dominik Schürmann, Arne Brüsch, Stephan Sigg, and Lars Wolf. “BANDANA – Body Area Network Device-to-device Authentication using Natural gAit”. In: *IEEE International Conference on Pervasive Computing and Communications (PerCom)*. Mar. 2017, pp. 190–196

- [11] Dominik Schürmann, Sebastian Willenborg, Felix Büsching, and Lars Wolf. "RAIM: Redundant Array of Independent Motes". In: *International Conference on Networked Systems (NetSys)*. Mar. 2017, pp. 1–8
- [12] Dominik Schürmann, Julian Timpner, and Lars Wolf. "Cooperative Charging in Residential Areas". In: *IEEE Transactions on Intelligent Transportation Systems* 18.4 (Apr. 2017), pp. 834–846

2016

- [13] Sergej Dechand, Dominik Schürmann, Karoline Busse, Yasemin Acar, Sascha Fahl, and Matthew Smith. "An Empirical Study of Textual Key-Fingerprint Representations". In: *Proceedings of the 25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 193–208
- [14] Dominik Schürmann and Lars Wolf. "Surreptitious Sharing on Android". In: *Sicherheit 2016*. Vol. P-256. Lecture Notes in Informatics. Bonn, Germany: Gesellschaft für Informatik, Apr. 2016, pp. 137–148
- [15] Julian Timpner, Dominik Schürmann, and Lars Wolf. "Trustworthy Parking Communities: Helping Your Neighbor to Find a Space". In: *IEEE Transactions on Dependable and Secure Computing* 13.1 (Jan. 2016), pp. 120–132

2014

- [16] Sergej Dechand, Dominik Schürmann, Jürgen Koslowski, and Matthew Smith. "Poster: CryptoCall: Simple End-to-End Cryptography for Voice Calls on Android". In: *Network and Distributed System Security Symposium (NDSS)*. Feb. 2014

2013

- [17] Julian Timpner, Dominik Schürmann, and Lars Wolf. "Secure Smartphone-based Registration and Key Deployment for Vehicle-to-cloud Communications". In: *Proceedings of the 2013 ACM Workshop on Security, Privacy & Dependability for Cyber Vehicles (CyCAR '13)*. Berlin, Germany: ACM, Nov. 2013, pp. 31–36
- [18] Dominik Schürmann and Stephan Sigg. "Poster: Handsfree ZRTP - A Novel Key Agreement for RTP, Protected by Voice Commitments". In: *Symposium On Usable Privacy and Security (SOUPS)*. July 2013
- [19] Dominik Schürmann, Jörg Ott, and Lars Wolf. "Authenticated Resource Management in Delay-Tolerant Networks using Proxy Signatures". In: *10th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. Banff, Alberta, Canada, Mar. 2013, pp. 44–51
- [20] Dominik Schürmann and Stephan Sigg. "Secure Communication Based on Ambient Audio". In: *IEEE Transactions on Mobile Computing (TMC)* 12.2 (Feb. 2013), pp. 358–370
- [21] Felix Büsching, Andreas Figur, Dominik Schürmann, and Lars Wolf. "Poster: Utilizing Hardware AES Encryption for WSNs". In: *Proceedings of the 10th European Conference on Wireless Sensor Networks (EWSN 2013)*. Feb. 2013, pp. 33–36

2011

- [22] Stephan Sigg, Dominik Schürmann, and Yusheng Ji. “PINtext: A Framework for Secure Communication Based on Context”. In: *Mobile and Ubiquitous Systems: Computing, Networking, and Services*. Ed. by Alessandro Puiatti and Tao Gu. Vol. 104. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg, Dec. 2011, pp. 314–325

Contents

1 Introduction **1**
- 1.1 Publications . . . 2
- 1.2 Contributions . . . 3
- 1.3 Outline . . . 5

2 Background and Terminology **7**
- 2.1 Decentralized Trust Establishment . . . 7
  - 2.1.1 Trust Anchors . . . 7
  - 2.1.2 Reputation and Cooperation . . . 8
- 2.2 Security Analysis . . . 8
  - 2.2.1 Threat Model . . . 8
  - 2.2.2 Attacker Strength . . . 9

3 Physical Trust Anchors **11**
- 3.1 Decentralized Trust Establishment for Vehicles . . . 11
  - 3.1.1 Threat Model and Implementation . . . 12
  - 3.1.2 Results . . . 13
- 3.2 Cooperative Charging of Electric Vehicles . . . 14
  - 3.2.1 Protocol . . . 14
  - 3.2.2 Threat Model . . . 15

4 Biometric Trust Anchors **17**
- 4.1 Analysis of Real-World Attacks on ZRTP . . . 17
  - 4.1.1 Systemization of Attacks . . . 18
  - 4.1.2 Evaluation of ZRTP Clients . . . 18
- 4.2 Device-to-Device Pairing using Gait . . . 19
  - 4.2.1 Quantization Scheme . . . 19
  - 4.2.2 Security Model . . . 21
  - 4.2.3 Evaluation . . . 21
- 4.3 Entropy and Security Analysis of Gait Pairings . . . 22
  - 4.3.1 Quantization Schemes . . . 22
  - 4.3.2 Entropy Evaluation and Algorithm Improvements . . . 24
  - 4.3.3 One-Shot Success Probability . . . 24

5 Usability **27**
- 5.1 Architecture for Security Tokens over NFC . . . 27
  - 5.1.1 Threat Model . . . 28
  - 5.1.2 API Design . . . 29
  - 5.1.3 Evaluation . . . 29

5.1.4 User Study . . . 29
5.2 Empirical Study of Textual Key-Fingerprints . . . 30
5.2.1 Key-Fingerprint Representations and Algorithms . . . 30
5.2.2 Attacker Strength Model . . . 31
5.2.3 Results . . . 32

6 Conclusions 33
6.1 Future Work . . . 33

Bibliography 35

Acronyms 42

Publications 43
Publication 1: Trustworthy Parking Communities: Helping Your Neighbor to Find a Space (appeared in IEEE Transactions on Dependable and Secure Computing 13.1 (2016)) 45
Publication 2: Cooperative Charging in Residential Areas (appeared in IEEE Transactions on Intelligent Transportation Systems 18.4 (Apr. 2017)) . . . 59
Publication 3: Wiretapping End-to-End Encrypted VoIP Calls: Real-World Attacks on ZRTP (appeared in Proceedings on Privacy Enhancing Technologies 2017.3 (July 2017)) 73
Publication 4: BANDANA – Body Area Network Device-to-device Authentication using Natural gAit (appeared in IEEE International Conference on Pervasive Computing and Communications (PerCom), Mar. 2017) . . . 91
Publication 5: Moves like Jagger: Exploiting variations in instantaneous gait for spontaneous device pairing (appeared in Pervasive and Mobile Computing 47 (2018)) . . . 99
Publication 6: On the secrecy of publicly observable biometric features: security properties of gait for mobile device pairing (submitted to IEEE Transactions on Mobile Computing (TMC), 2018) . . . 115
Publication 7: OpenKeychain: An Architecture for Cryptography with Smart Cards and NFC Rings on Android (appeared in Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT) 1.3 (Sept. 2017)) . . . 129
Publication 8: An Empirical Study of Textual Key-Fingerprint Representations (appeared in Proceedings of the 25th USENIX Security Symposium (USENIX Security 16)) . . 153

1 Introduction

Nowadays, computing and communication capabilities are integrated in our everyday surroundings. In 2025, it is expected that up to 50 billion devices will be deployed, collecting and exchanging large amounts of data [23]. They will be introduced in an increasing number of novel scenarios, often without permanent Internet connection. Thus, new types of ad-hoc networks with specialized properties evolved [24].

Network protocols are designed for their designated field of application, but even protocols at first sight coming from the same field can pose differing requirements. For example, the required transmission delay in vehicular networks varies greatly. On one end of the spectrum, Vehicular Vehicular Delay-Tolerant Networks (VDTNs) enable forwarding of messages over long physical distances without central infrastructure, but for the expense of high delays [25, 26]. On the other end, Vehicle-to-Vehicle (V2V) networks require direct real-time communication to exchange safety information [27]. Ad-hoc networks for the Internet of Things and Body Area Networks again pose different requirements. Due to their constraints in power consumption, sensor readings may only be synchronized periodically over a local gateway, such as a smartphone. All these new networks have in common that Personally Identifiable Information (PII), such as location and behavior related data, is exchanged. However, in many scenarios only a tiny amount of this information is actually processed and analyzed [23]. It is crucial that these PII are protected from unauthorized access by malicious attackers and state actors. While, in principle, this goal is shared between all networks, its implementation depends on the specifics of the scenario. Without relying on central authorities trust between humans, vehicles or pervasive devices must be established in an ad-hoc manner.

Therefore, the goal of this thesis is to identify how trust can be established in heterogeneous networks with varying requirements. In this work, it is argued that trust should be established directly between devices while still being as unobtrusive as possible. To accomplish high usability, human interaction is kept to a minimum or removed completely. This research deals with trust anchors, i. e., something that allows the assignment of real-world entities to digital identities. Trust anchors allow for ad-hoc authentication between devices previously unknown to each other without requiring a third party introducer. In this thesis, trust anchors are categorized into *physical trust anchors*, where digital identities are assigned to real-world objects, and *biometric trust anchors*, where human biometrics are utilized as a shared feature for agreeing on a cryptographic key.

Physical trust anchors prevent the duplication of digital identities by binding them to a physical object. In vehicular networks, vehicles are typically introduced to each other using digital certificates, signed by car manufacturers and government agencies. Still, a valid attack scenario is to obtain a number of valid certificates, e. g., by extracting them from other vehicles, and using them to fake valid safety messages. Consequently, the trustworthiness of safety messages must be evaluated independently of central authorities. Vehicle sensors and a history of previous encounters can be used to establish other vehicles as physical trust anchors that correspond to observed certificates, effectively preventing these attacks. While, in this example, vehicles serve as trust anchors, there exist many other network types where the duplication of identities can be prevented by assigning them to unclonable real-world objects. A second scenario considered in this thesis is the usage of security tokens, such as smart cards or wearables, as physical trust anchors. To prevent duplication of these low-cost objects, public key cryptography is implemented. By storing secret keys solely on these security tokens and preventing direct memory access using special hardware constructions, it is guaranteed that a public key always corresponds to one unique security token.

Biometric trust anchors are based on human biometrics to generate unique keys in a deterministic way. Fingerprint scanner, iris recognition, and face unlock are well known examples of applied authentication techniques. While human body parts cannot be duplicated par for par, biometric trust anchors do not provide the same strong uniqueness as public key cryptography, e. g., as implemented in vehicles and security tokens. This is due to the fact that it is difficult to detect the liveliness of a biometric trust anchor. Several attacks have successfully extracted fingerprints from high-resolution images to unlock other persons' smartphones or used 3D modeled faces to outsmart face unlock implementations. Consequently, in this thesis, biometric features are not utilized to unlock smartphones, but to establish trust between devices. In stark contrast to the selection of reproducible biometric features for matching them against a template database, fresh secrets are generated to secure an ad-hoc key agreement. Thus, as opposed to user authentication, there is no way to attack these protocols before or after the key agreement takes place. In Voice over IP (VoIP) applications, trust is established by recognizing the participants' voices and comparing Short Authentication Strings (SASs). For Body Area Networks, the usage of time-variant human gait is proposed to create a shared key between devices. The methods are still deterministically producing the same key for a specific time span, but a different key at a later point in time.

The main research presented in this thesis is how trust anchors are verified and used to protect against adversaries. While some verification methods, such as vehicle detection and quantization of human gait, are fully automated, others require different forms of human interaction. Besides an objective threat model, the perceived security and mental model of users is a crucial factor for adoption. In this thesis, an architecture for using security tokens over Near-Field Communication (NFC) has been proposed. To evaluate its usability, its API design has been compared with existing ones and a laboratory user study has been conducted. Since the beginning of public key cryptography, key-fingerprints have been used to bind keys to unique textual identifiers. These serve as trust anchors by verbal comparison. In this thesis, different textual key-fingerprint schemes have been analyzed to define an appropriate attacker strength model.

1.1 Publications

The article thesis at hand makes several contributions in the area of trust establishment in heterogeneous networks. The individual contributions are contained in the following eight scientific publications both in conference proceedings and journals. These eight publications are contained in this document and are listed (with their respective page numbers) in the following:

1. Julian Timpner, Dominik Schürmann, and Lars Wolf. "Trustworthy Parking Communities: Helping Your Neighbor to Find a Space". In: *IEEE Transactions on Dependable and Secure Computing* 13.1 (Jan. 2016), pp. 120–132 (on page 45)

2. Dominik Schürmann, Julian Timpner, and Lars Wolf. "Cooperative Charging in Residential Areas". In: *IEEE Transactions on Intelligent Transportation Systems* 18.4 (Apr. 2017), pp. 834–846 (on page 59)

3. Dominik Schürmann, Fabian Kabus, Gregor Hildermeier, and Lars Wolf. “Wiretapping End-to-End Encrypted VoIP Calls: Real-World Attacks on ZRTP”. in: *Proceedings on Privacy Enhancing Technologies* 2017.3 (July 2017), pp. 4–20 (on page 73)
4. Dominik Schürmann, Arne Brüsch, Stephan Sigg, and Lars Wolf. “BANDANA – Body Area Network Device-to-device Authentication using Natural gAit”. In: *IEEE International Conference on Pervasive Computing and Communications (PerCom)*. Mar. 2017, pp. 190–196 (on page 91)
5. Dominik Schürmann, Arne Brüsch, Ngu Nguyen, Stephan Sigg, and Lars Wolf. “Moves like Jagger: Exploiting variations in instantaneous gait for spontaneous device pairing”. In: *Pervasive and Mobile Computing* 47 (2018), pp. 1 –12 (on page 99)
6. Arne Brüsch, Ngu Nguyen, Dominik Schürmann, Stephan Sigg, and Lars Wolf. “On the secrecy of publicly observable biometric features: security properties of gait for mobile device pairing”. Submitted to: *IEEE Transactions on Mobile Computing (TMC)*. 2018 (on page 115)
7. Dominik Schürmann, Sergej Dechand, and Lars Wolf. “OpenKeychain: An Architecture for Cryptography with Smart Cards and NFC Rings on Android”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 1.3 (Sept. 2017), 99:1–99:24 (on page 129)
8. Sergej Dechand, Dominik Schürmann, Karoline Busse, Yasemin Acar, Sascha Fahl, and Matthew Smith. “An Empirical Study of Textual Key-Fingerprint Representations”. In: *Proceedings of the 25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 193–208 (on page 153)

1.2 Contributions

The contribution of this article thesis in general and of the research papers stated in the previous section in particular can be summarized as follows:

Phyiscal Trust Anchors

- **Decentralized Trust Establishment for Vehicles [15]:** The first contribution is a decentralized mechanism for establishing trust in vehicular networks. It works without Trusted Third Parties (TTPs) and has been designed for the particular scenario of trustfully finding parking spaces via query response communication. The protocol is end-to-end encrypted and establishes a network of trusted peers called ‘Parking Communities’. A threat model and a qualitative comparison with other schemes is provided.

- **Cooperative Charging of Electric Vehicles [12]:** The second contribution is a protocol for cooperative handovers extending ISO 15118, the standard for automated charging of Electric Vehicles (EVs). To prevent competition for scarce Charging Stations (CSs), it provides incentives to hand over an CS to the next EV without introducing new components or trust establishment. It works by exchanging digitally signed receipts and delegating parts of the payment to a potential successor. The protocol has been designed and integrated with ISO 15118. Furthermore, a detailed threat model is provided.

Biometric Trust Anchors

- **Analysis of Real-World Attacks on ZRTP [7]:** The third contribution provides an evaluation of real-world attacks on ZRTP and in particular on applications implementing this standard. ZRTP is a protocol for establishing trust between two endpoints in real-time VoIP communication. By verbally comparing a few on-screen characters or words, called SASs, the participants can be sure that no one is wiretapping the call. The evaluation of seven applications includes seven protocol and four non-protocol tests. A novel attack, called 'Shared Man-in-the-Middle', is introduced and several weaknesses in current implementations have been uncovered.

- **Device-to-Device Pairing using Gait [10, 2]:** The fourth contribution is a novel device-to-device pairing scheme. Instead of requiring manual comparison or input of PINs, the scheme allows for pairing of devices in a Body Area Network (BAN) without any user interaction. As a trust anchor, the unique time-sensitive variations of the user's gait biometrics are used. Fingerprints generated independently on participating devices from the same gait sequence are used as secrets for a Password-Authenticated Key Agreement (PAKE). To account for remaining differences between body parts, error correcting codes are utilized. A comprehensive evaluation of the proposed device-to-device pairing scheme is provided. The final security model and pairing protocol requires only 12 seconds of human gait to establish a secure pairing. The fingerprint similarity of this approach has been evaluated on a large-scale dataset with over 480 participants for pairing between three locations on a body's waist. To show the similarity and success probability between different sensor locations and movements, a second dataset with 15 participants and seven on-body locations has been used. All possible intra-body similarities are evaluated against the case of an attacker (inter-body) for walking, running, as well as climbing up and down.

- **Entropy and Security Analysis of Gait Pairings [1]:** The fifth contribution is a systematic comparison of existing device-to-device pairing schemes based on gait. For the first time, all algorithms are compared in regards to their achieved intra- and inter-body key similarity using the same dataset. In addition, this work shows that classical entropy tests, such as DieHarder and ENT tests, are not suited to verify the degree of randomness exhibited from quantized keys of short length. Thus, a set of entropy tests has been designed especially for visually analyzing the entropy of keys quantized from gait features. A detailed evaluation of attack scenarios for gait-based pairing is provided and evaluated. Several weaknesses have been identified and thwarted by proposing modifications.

Usability

- **Architecture for Security Tokens over NFC [6]:** The sixth contribution is an architecture to establish end-to-end encrypted asynchronous networks using NFC smart cards and rings. Instead of trusting the smartphone to store secret keys, they are bound to external NFC hardware and cannot be extracted by attackers. A high-level API has been designed for Android and deployed in OpenKeychain, an application with over 100 000 users. The contribution includes a threat model and NFC performance measurements. As an alternative form factor to smart cards, the prototype of an NFC signet ring has been created. A lab study with 40 participants, which includes measurements and a subsequent interview, shows that NFC-based solutions are more user friendly in comparison to traditional password-protected keys.

- **Empirical Study of Textual Key-Fingerprints [13]:** The seventh contribution is a evaluation on the cryptographic design of textual key-fingerprint representations used by systems, such as OpenPGP or OpenSSH, for establishing trust to a public key. These include representations for numeric, alphanumeric, word, and sentence encoding similar to their real-world usage in standardized protocols and mobile applications. For a 1047 participant large usability study, an attacker strength model has been designed. It consists of an estimated attacker strength for partial preimage attacks to properly configure a Key Derivation Function (KDF) for fingerprint calculation.

1.3 Outline

This extended overview of the article thesis is meant to be a summary of the publications listed in Section 6.1. A full list of publications by the author is given on page vii of this document. Those publications that are considered as an essential part of this work are contained in the thesis and start on page 43. The thesis makes explicit references to those papers to highlight their context and the relation between the publications. The rest of this thesis is structured as follows:

Section 2 gives an overview over the terminology used in this thesis and provides a discussion about the definition of trust and the required background to understand the difference between centralized and decentralized trust establishment. In Section 3, physical trust anchors are introduced. Their purpose and properties are discussed and how they can be utilized to protect against different attack scenarios. Furthermore, how these trust anchors can be cryptographically bound to identities is explored. In Section 4, interactive and non-interactive biometric trust anchors are introduced. The general protocol design for authenticating a key agreement is presented for both cases. Potential pitfalls and design issues are discussed and a novel way how information entropy can be visually evaluated is presented. A description how the usability of proposed verification mechanisms can be tested is discussed in Section 5. Finally, before the research papers are presented, a summary of this extended overview and a short outlook on future work is given in Section 6.

2 Background and Terminology

Novel networks, such as industrial, environmental, wearable and vehicular sensing systems, cannot be directly connected using cellular technologies [28]. In addition, the trustworthiness of collected sensor data might be doubtful. Traditionally, trust is established by installing a Certificate Authority (CA) in the infrastructure issuing cryptographic certificates for all participating devices. This may (a) not be possible in networks without permanent connection to central infrastructures and (b) does not necessarily proof the trustworthiness of exchanged data. In addition, CAs are prone to targeted attacks [29]. Alternative approaches, such as the Web-of-Trust and Namecoin struggle with adoption due to their complexity [30, 31]. While end-to-end encryption has finally been implemented by voice and messaging platforms, such as WhatsApp [32] and Facebook Messenger [33], key discovery is still handled using central key servers. These could potentially be subverted by law enforcements or criminals. While certificate transparency architectures [34, 35] can help to identify miss-issued certificates and keys, the trustworthiness of exchanged data might still be doubtful. In addition, future networks are expected to experience frequent fluctuations in device count and identity as devices join or leave [36]. Consequently, novel scenarios require novel ways for establishing trust between devices in a direct ad-hoc way. Furthermore, mechanisms for managing reputation and deriving actionable decisions must be developed.

In the literature, definitions of trust vary greatly [37, 38]. Therefore, the terminology as used in thesis is defined in the following. The fundamentals of establishing trust in a decentralized way using trust anchors is introduced. For managing trust, reputation-based models and ideas for incentivized cooperation are discussed and how decisions can be derived. Furthermore, the basics for defining a threat model and evaluating the attacker strength are introduced. This section makes no claim to completeness but is instead intended to give an overview limited to the scope of this thesis.

2.1 Decentralized Trust Establishment

In the following, components required for establishing trust between devices in a direct way are presented and discussed. An initial ad-hoc trust establishment requires trust anchors, while reputation or incentivized cooperative scenarios are used to manage relationships over a longer timespan, finally leading to actionable decisions.

2.1.1 Trust Anchors

Authentication between participants works by verifying if someone or something is what it claims to be. A proof can be constructed to verify the claim. In this proof, a trust anchor serves as the root entity that verifiers directly accept as reliable [39]. The proof could involve several entities building a chain where trust is derived from one to another.

In this thesis, the most approachable example of trust anchors are security tokens, i. e., external cryptographic hardware such as smart cards. A public/secret key pair is generated uniquely for each security token. By cryptographically signing something, the security token serves as a trust anchor.

It is guaranteed that a one-to-one assignment between the physical security token and the digital public key exists. By using tamper-proof hardware, the security token is made unclonable and the secret key can not be copied. Conclusively, the security token is accepted as a reliable trust anchor.

Trust anchors are categorized into physical and biometric ones. While physical trust anchors bound to real-world objects can easily be made unique by implementing cryptography, biometric features do not provide this by themselves. Instead, a quantization scheme must be designed to derive a unique digital identity from the analog biometric features. While the probability of finding a colliding public/secret key pair is proven to be mathematically negligible, this is not easily shown for biometric trust anchors. To show their uniqueness, an evaluation using a human dataset must be conducted. This allows for an estimation how much or how long the biometric features need to be recorded to reach an entropy level acceptable for cryptographic applications.

An additional categorization can be made between trust anchors that are fully automated (zero-interaction) and trust anchors that require conscious human interaction. A network between several wearables on the same body can autonomously use the body as a shared trust anchor without user interaction. However, an end-to-end encrypted messaging platform that connects people over long distances requires user interaction. Here, biometric trust anchors must be verified over an inbound channel and participants are requested to verify them by using their human intuition, e. g., recognizing the voice.

2.1.2 Reputation and Cooperation

It is important to note that trust anchors are only used to establish the initial trust, e. g., during first encounters or device-to-device pairings. Protocols to manage relationships over longer time spans can be build on-top. Reputation-based schemes assign ratings to each digital identity to assess the trustworthiness of data received from these devices. Reputation ratings should be updated continuously based on the observed behavior of other devices. An alternative to managing reputation ratings is to incentivize a cooperation between devices: A protocol can be designed to prevent selfish behavior by providing an advantage for devices to act cooperatively. These protocols are often sub-classified into barter-based and credit-based schemes [40]. In addition, protocols exist that unify credit- and reputation-based ideas in one scheme [41]. Modern schemes, often based on cryptocurrencies, provide self organization without a central virtual bank [42]. Finally, based on reputations or incentives, an actionable decision can be derived. This decision is often binary, corresponding to a yes/no question. For example, a device might decide to send its sensor data only to devices where trust has been established using long-term ratings.

2.2 Security Analysis

The practical security of a system results from protocol design, implementation, and the interaction between protocols and their respective implementations. A specified system must provide a carefully defined set of security properties, while also explicitly defining attacks that are out of scope. This is summarized using a threat model. In addition, theoretical properties can be evaluated. This includes an analysis of key entropy and upper bound for the assumed attacker strength.

2.2.1 Threat Model

In the following, typical attacks in the context of this thesis are summarized and classified.

Eavesdropping: An attacker could try to record messages exchanged between devices. Eavesdropping is classified as a passive attack, i. e., the attacker is not intercepting, modifying and re-transmitting the messages. This attack is typically prevented by implementing end-to-end encryption.

Man-in-the-Middle Attacks: Even with end-to-end encryption, an attacker could place herself on the route between devices in the network, intercepting the initial key exchange, and replacing public keys with her own to be able to decrypt messages later. Man-in-the-Middle attacks are classified as active attacks, i. e., instead of simply recording messages, they need to be actively modified. Naively, this attack is prevented by providing central key servers distributing the keys and proofing their ownership. This defers the trust decision to a central entity, who in turn could be subverted and act as a Man-in-the-Middle. Trust anchors provide a viable alternative by directly establishing trust between devices.

Impersonation: Without cryptographic protocols, a device can pretend to be a different device by observing network communication and copying its address. In networks with digital certificates, the effort is higher as the corresponding secret key must be stolen from the victim. By binding the secret key to specialized hardware, e. g., security tokens, this is again made more difficult. In the field of biometric trust anchors, the security assessment is more complex. In today's environment, people are surrounded by smartphone and surveillance cameras. These present the biggest threat to most biometric features, such as fingerprints, iris/facial features, or the person's gait. Image recognition techniques can be used to extract the required features and re-construct the biometrics allowing impersonation.

Sybil Attacks: This attack is named after a case in psychology, where a medical doctor was searching for interesting patients to study multiple personality disorder. Due to his high interest, he told patients about his research. Then, one of his patients—later given the pseudonym Sybil—exhibited her multiple personalities. Today, it is assumed that Sybil developed these personalities just to impress the doctor [43]. Transferred to network protocols, a Sybil attack is an attack where one entity fakes multiple identities, i. e., it is an impersonation attack on a larger scale. Sybil attacks can be either prevented by binding physical trust anchors to digital identities and carefully allowing only verified devices or by managing reputations over a longer time.

Privacy Threats: As with all technology, certain protocols with trust anchors or reputation schemes could expose parts of a person's privacy. In vehicular networks, location tracking is a nearly unsolvable problem made worse when long-term reputation ratings are stored. Protocols using biometric trust anchors, on the other hand, must take care of protecting the biometric features they are using. By using fuzzy cryptography, features can be stored in a secure form without revealing them to an attacker. In practice, this is rarely implemented.

2.2.2 Attacker Strength

The strength of an attacker trying to brute force generated keys can be approximated. Depending on the properties of the evaluated cryptographic protocol, it must be differentiated between an offline and online attacker.

For example, to impersonate a victim in a traditional Public Key Infrastructure (PKI), an attacker needs to generate two certificates producing the same hash. One certificate is issued for a properly owned domain, while the other is issued for the victim's domain. By proof of ownership, a valid signature can be obtained from a CA for the first one. Unfortunately, this signature is then also valid for the second certificate. Thus, the system requires *collision resistance*. A hash collision can be found offline by brute-forcing possible combinations.

A similar offline attack exists in systems, where trust is established directly instead of using a chain of trust. Instead of finding a hash collision, a hash preimage needs to be found, i. e., for an already existing hash, another input needs to be found generating the same hash. Thus, the system must be *preimage resistant*. This attack is much harder than a collision, approximately by a factor of two in the exponent (depending on the hash function) [44, A.1]. For example, SHA-256 provides a security strength of 128 bit against collision attacks (worst case: 2^{128} brute force steps) and 256 bit against preimage attacks (worst case: 2^{256} steps). As with the certificate collision, the only constraint in this attack is given by the available computing power to brute force hashes. Computing power in turn depends on the available money an attacker has at her disposal. To get a feel for this, Percival calculated the monetary costs of brute forcing a hash generated by the Key Derivation Function (KDF) *scrypt* of length 2^{38} (8 letters) with $610k, while 2^{53} (8 characters) costs $16B [45]. Depending on the chosen KDF, the brute force attack can be faster or slower. Thus, if a protocol is designed to withstand an offline attacker, a modern secure memory-hard KDF must be chosen.

For interactive pairing protocols, such as Password-Authenticated Key Agreements (PAKEs), the attacker can be reduced to an one-shot online attacker with only one try per protocol execution. PAKEs are protected by a shared key only known to the participating devices. Thus, if a key length of 2^{16} is chosen, the attacker has a success probability of 2^{-16} once per protocol execution. Here, the maximum number of allowed parallel protocol runs must be considered, which could increase the attacker's success. Thus, interactive protocols must be constrained, for example to 2^{10} maximum parallel executions [46]. In addition, the time between consecutive protocol runs must be considered. These estimations assume that perfect entropy has been used for key generation, i. e., the likelihood of occurrence of each possible key is uniformly distributed. For keys quantized from biometric features, this may not be true. Thus, it is crucial to analyze the quality of biometric trust anchors.

3 Physical Trust Anchors

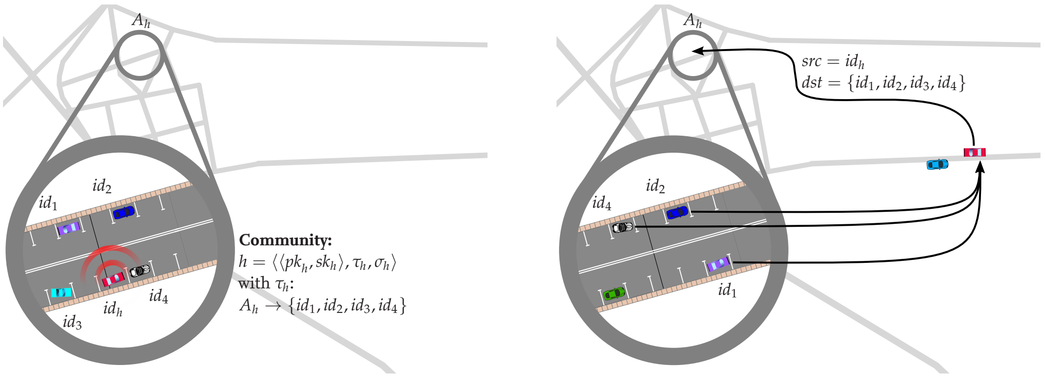
As introduced in the previous section, the establishment of physical trust anchors can serve as an alternative to centralized trust management. Typically, a public/secret key pair is generated per physical trust anchor to provide a unique digital identifier. In this work, physical trust anchors were utilized to provide decentralized services without requiring a permanent connection to central infrastructure. How the assignment of physical objects to public keys work depends on the conditions and components of the scenario.

In vehicular networks, public keys, more precisely certificates issued by car manufacturers, can be assigned to real-world vehicles by verifying that received messages correspond to the real-world vehicles in the surrounding area [47]. This can be implemented using close-to-market sensors that measure the distance to closest objects, in this case the distance to other vehicles [48]. During the V-Charge project, which was funded by the EU 7th Framework Programme for Research and Technological Development (FP7), different scenarios for automated valet parking and charging arose [48]. To validate exchanged information about free parking spaces and sparsely available Charging Stations (CSs), novel decentralized services were envisioned. In this work, protocols have been designed based on managing long-term reputation ratings or alternatively providing incentives for cooperation between vehicles. The discussion in this section is primarily based on the following publications: [15, 12]. These publications are part of this article thesis and hence are included in this document. An overview can be found in Section 1, whereas the publications can be found starting on page 43.

3.1 Decentralized Trust Establishment for Vehicles

The work discussed in the present section has been published in “IEEE Transactions on Dependable and Secure Computing” [15] (see page 45). Here, the idea of using vehicle sensors to establish trust in other vehicles has been applied in the real use case of decentralized search for parking space. In this scenario, it is assumed that vehicles send long-distance geocast [26] queries to their intended driving destination to ask for information about free parking space. Other vehicles in this destination area can decide to answer with availability information deduced from their sensors. Because standards, e. g., defined by the Car2Car Consortium [27], require a centralized security architecture, it is assumed that a PKI or Identity-Based Cryptography (IBC) systems are already in-place. However, as discussed in the introduction, other vehicles’ digital certificates could be stolen by attackers or maliciously used on the vehicle itself to cryptographically sign faked responses to these parking queries. An attacker could be interested in keeping free parking spaces for herself to use. Thus, it is important to note that a digital certificate including a Vehicle Identification Number (VIN) only proofs that a vehicle with this VIN exists somewhere. While a certificate could be perfectly valid, the validity of signed data cannot be deduced from this.

The idea in this work is to establish reputation ratings of other vehicles and manage so called ‘Parking Communities’, i. e., a set of vehicles normally parking in the same home area day-to-day helping each other with the exchange of valid parking information. Formally, each vehicle is


A_h
id_1
id_2
id_3
id_4
id_h
Community:
$h = \langle\langle pk_h, sk_h\rangle, \tau_h, \sigma_h\rangle$
with τ_h:
$A_h \rightarrow \{id_1, id_2, id_3, id_4\}$
A_h
$src = id_h$
$dst = \{id_1, id_2, id_3, id_4\}$
id_4
id_2
id_1

(a) Collecting IDs via neighbor discovery and establishing a physical trust anchor

(b) Encrypted and signed query/response for a free spot via geocast

Figure 3.1: Creating and querying a Parking Community

assumed to retrieve a public/secret key pair $\langle pk, sk\rangle$ from the central security infrastructure for each community. For example, as depicted in Figure 3.1, the community in the vehicle's home area is defined as $h = \langle\langle pk_h, sk_h\rangle, \tau_h, \sigma_h\rangle$ with $\tau_h : A_h \rightarrow \{id_1, \ldots, id_n\}$. Here, τ_h is a trust anchor defined by a set of areas mapped to a set of vehicles that are part of this community. To each vehicle id, two counting variables are defined by $\sigma : id \rightarrow \{r, s\}$ and used for managing a beta reputation function [49] that changes over time. In practice, vehicles parking in an area can collect other vehicles public keys via neigbor discovery. The verification that real vehicles correspond to these keys is done with high probability by an neighbor position algorithm by Fiore et al. [47]. Over time, vehicles can collect a number of public keys corresponding to others often parking in its home areas. To hide future parking space queries from surveillance, the geocast queries can be encrypted using the collected public keys. A response for parking space is defined by 1 if parking space is available or -1 if no space is available. To establish reputation ratings, each originator keeps track how many estimates turned out to be correct and incorrect. These results are fed into the beta reputation function to estimate future availability. It establishes reputation ratings that change over days, resulting in an improved approximation on future queries, while still protecting against malicious or defective vehicles. The paper [15] presenting parking communities includes a detailed description of the approach and protocol. The contributions in this thesis are the definition of a threat model, the design and implementation of the protocol for Delay-Tolerant Networks and a comparison with existing schemes from the literature. The usage of the beta reputation function to calculate ratings over time were mainly contributions of a co-author and are not further elaborated here.

3.1.1 Threat Model and Implementation

A security architecture has been designed based on a carefully defined threat model, summarized in the following.

Impersonation: A malicious vehicle could try sending messages in another vehicle's name to decrease reputation ratings. This is easily prevented by encoding a vehicle's public key directly as their network address. This imposes no problems as public keys are already unique due to their cryptographic generation and vehicles do not require pronounceable names. In case of an Elliptic Curve Cryptography (ECC)-based protocol with 256 bit keys, the success probability of generating the same key is 2^{-256} and thus considered infeasible.

Sybil Attack: An attacker could try to win the consensus for a parking query and convince the originator that no space is available. For this one would need to generate enough public keys and convince parking vehicles that these are real-world vehicles. This is not possible since neigbor vehicles are verified using the neighbor position algorithm [47].

Denial of Service: An attacker could try to exhaust computing power by querying several times in a row. To counteract this, a rate limit approach is proposed. On high consumption, a vehicle can constrain its responses to reputable members of their own Parking Communities.

Location Tracking: As an inherent property of decentralized query-response protocols, by querying other vehicles, the originator's own location and context is exposed [50]. Pseudonym certificates have been proposed in Vehicle-to-Vehicle (V2V) networks [51], but would interfere with the approach of managing a long-term reputation rating. The usage of a KDF is proposed to built a deterministic way for members of the same community to change their IDs. For this, a secret is shared during the neighbor discovery phase that is later used to derive new IDs.

Based on the requirements resulting from this threat model, IBR-DTN [52] has been extended with a security architecture. IBR-DTN is an implementation of the bundle protocol allowing the required multi-hop communication over long delays. Instead of using Diffie-Hellman-based key exchanges, the cryptographic algorithms Elliptic Curve Digital Signature Algorithm (ECDSA) and Elliptic Curve Integrated Encryption Scheme (ECIES) have been integrated to provide asynchronous ECC. Where possible, the implementation follows the recommendations of the *Bundle Security Protocol Specification* (RFC 6257) [53]. As a notable property, a new Endpoint Identifier (EID) scheme has been designed to derive EIDs from public keys (cf. impersonation attack) that is defined by

$$eid_c := \text{‘sec://’} \parallel base64url(pk)$$

Here, $base64url()$ is the URL-safe Base64 encoding [54]. The Scheme-Specific Part (SSP) consists at minimum of the length of the ECC public key, i. e., 32 byte. An ECC public key is 32 byte long. Thus, due to Base64 encoding, the length is $len_{ssp}(n) = \lceil \frac{n}{3} \rceil \cdot 4$.

3.1.2 Results

The scheme 'Parking Communities' has been compared with other existing key and trust/reputation management approaches. Key management schemes can be sub-categorized into ones with Trusted Third Parties (TTPs) using PKI, IBC or Hierarchical Identity-Based Cryptography (HIBC). Trust and reputation establishment schemes have been categorized into credit schemes using a central virtual bank or self organizing ones. In comparison, 'Parking Communities' is the only one that uses physical encounters as trust anchors to protect against Sybil attacks, while keeping a low network overhead, e. g., no credit transactions need to be processed.

Simulating the scheme using The ONE [55] and a Working Day Movement over 8 full days, showed that is is functioning well for its intended purpose. After 5 days, 50 % of all communities have 2 to 4 members, while the sizes stabilize around day 6 and 7. Starting with a reputation ranking of 0.5, the average of the honest vehicles' reputations raises to 0.75 on day 7, while the malicious vehicles' reputations drops to 0.3. The evaluation using the probabilities of lying of $\psi = 50\,\%$ and

$\psi = 85\,\%$ shows that even under these circumstances, the rate of correct parking decisions raises over the simulated days, reaching a maximum of 80 % ($\psi = 50\,\%$) and 75 % ($\psi = 85\,\%$).

While the proposed trust management scheme has been applied to the problem of searching for parking space, it could also be applied to other scenarios, such as exchanging traffic information on a planned route.

3.2 Cooperative Charging of Electric Vehicles

A model for cooperative charging of Electric Vehicles (EVs) has been published in "IEEE Transactions on Intelligent Transportation Systems" [12] (on page 59). Instead of establishing cooperation by long-term reputation ratings as proposed by "Parking Communities", short-term cooperation based on incentives has been envisioned. In this case, the specific problem of unfair and inefficient utilization of charging stations is solved. The charging infrastructure required by EVs is not fully developed in European residential areas. Based on the predictions by the German National Electric Mobility Platform (NPE), 1 000 000 EVs will require about 70 000 public on-street charging spots [56]. Because it is crucial to use this infrastructure as efficiently as possible, a protocol between EVs for fairly utilizing charging stations has been designed. The contributions in this thesis are the definition of a threat model as well as the protocol design an integration into existing standards.

3.2.1 Protocol

An EV currently charging at one station (Provider) is incentivized to free up the space for the next vehicle (Requester) by coordinating a handover. The protocol prevents competition with other requesters and facilitates a coordination via direct V2V connections. It integrates with the existing ISO 15118 [57, 58] standard that defines the communication protocols between charging infrastructure and EVs.

The protocol is designed as follows: The requester can query for charging station availability by sending a request into the designated area, receiving the estimated time providers still require and a cryptographic proof (valid metering receipt per ISO 15118). The communication is secured using traditional V2V PKI. The cooperative handover takes place when a provider selects a requester by random (for fairness) and announces its cost consisting of reparking and possibly forgoing a full battery charge by leaving early. Instead of introducing a centralized payment step or virtual currencies, the protocol extension allows for directly handing over the current charging session secured by Transport Layer Security (TLS) and splitting the provider's charging bill between provider and requester. More detailed, a charging station sends cryptographically signed meter receipts consisting of a timestamp and the charged kWh. The provider maintains copies and delegates them to the requester that cryptographically signs the negotiated share of it as a payment for the handover. During the payment delegation, the requester's TLS session, which is only unilaterally authenticated, is tunneled through the mutually authenticated TLS session between both vehicles.

A detailed cost model and its effects on a market have been discussed and simulated in the paper. As these are mainly contributions of a co-author, they are only summarized here to show the effectiveness of the protocol. The scenarios with and without the proposed cooperative charging extension have been simulated. The amount of EVs and charging stations have been approximated based on the report by 'Nationale Plattform Elektromobilität' [56]. The simulations show that the utilization of charging stations can be increased from 21 % to 46 %. The number of daily charging sessions is increased by a factor of 7.

3.2.2 Threat Model

Due to the interaction between V2V PKI, TLS and specifics of ISO 15118, a lot more attack scenarios can be executed in comparison to 'Parking Communities'. The most interesting attacks are reported in the following:

Impersonation and Sybil Attacks: By replaying an eavesdropped message, an EV could impersonate another EV occupying a charging spot. Creating a large amount of fake EVs for a Sybil attack could increase the probability of being selected as the next requester. These attacks are prevented using the V2V PKI and TLS' replay protection.

Requesting a Spot without Having a Contract: A contract with a provider of charging stations is required to be able to pay. An attacker could be selected as a requester while not having a contract but still freeing up the spot as a parking place. As a solution, before freeing up a spot the contract is validated over the TLS connection. The provider cancels the process and re-selects a different requester.

Requester Sending Invalid/No Metering Receipts: The requester could send metering receipts that are not cryptographically valid or no receipts at all. To protect against this, the process is either canceled automatically when verifying the receipts or by timeout.

Replaying Metering Receipts: An attacker could replay metering receipts from a different previous session. Here, metering receipts are made unique per session and are thus not accepted.

Honeypot Provider A provider could falsely attract many requesters by broadcasting an acceptance message to all. She could then split the metering receipts between all and let them pay the whole charging. As metering receipts are cryptographically signed, they can be used as a proof to resolve this fraud at a clearing house.

Profit by Blocking Charging Spot: A malicious EV could keep a charging spot blocked with the idea of making profit by letting other EVs pay. It is difficult to make a long-term business out of this, because valid metering receipts are required for delegation, which in turn require a preceding charging process.

4 Biometric Trust Anchors

Computing devices can be programmed to generate cryptographic keys that are assigned to real-world objects providing physical trust anchors. In contrast, human bodies cannot directly generate keys. Instead, biometric features can be used to protect a protocol for agreeing on a cryptographic key. In case a key agreement is initiated between humans, it likely requires conscious human interaction. This is because the involved devices have no prior knowledge about the biometric features of the other participants. Still, biometrics can be verified by relying on humans' intuition in identifying the participants, e. g., by means of recognizing the participants' voices. For agreeing on a key between devices worn on the same body, zero-interaction protocols are possible because shared biometrics may be available. For this, analog biometric features must be turned into digital fingerprints. This process is called quantization. The generated keys must be evaluated in regards to their degree of randomness (information entropy) to rate their security level.

In this work, the security and usability of crypto phone apps implementing the ZRTP standard [59] have been evaluated. It uses the participants' voices as biometric trust anchors to secure a Voice over IP (VoIP) call. The focus lies on real-world ZRTP clients and pitfalls resulting from implementing the specification. Moreover, for wearables, the BANDANA protocol has been designed and compared with other proposals from the literature [60, 61, 62]. It uses a person's gait, the way how someone walks, as a unique biometric trust anchor for protecting the initial pairing process between devices. A detailed threat model has been defined and the entropy of keys generated by BANDANA's quantization scheme have been visually evaluated and compared with keys generated by other schemes. The discussion in this section is primarily based on the following publications: [7, 10, 2, 1]. These publications are part of this article thesis and hence are included in this document. An overview can be found in Section 1, whereas the publications can be found starting on page 43.

4.1 Analysis of Real-World Attacks on ZRTP

The work in this section has been published in "Proceedings on Privacy Enhancing Technologies" [7] (on page 73). The ZRTP key agreement protocol has been designed to authenticate end-to-end encrypted phone calls. Instead of relying on a central key infrastructure, it uses the voice of the participants as trust anchors. It is assumed that participants know each other's voice in advance and can identify if they are speaking to the correct person. To bind the biometric trust anchor to the key agreement, Short Authentication Strings (SASs) are displayed on-screen for verbal comparison.

Since its standardization in 2011 [59], ZRTP has been typically implemented in conjunction with the Session Initiation Protocol (SIP). It has been formally verified and several protocol attacks were discussed [63, 64]. Still, no systemization of attacks is provided and attacks have not been applied to modern implementations of ZRTP. Thus, the contributions in this thesis are a systematic overview of all possible attack scenarios and an evaluation of modern ZRTP clients. The co-authors of the corresponding paper helped testing the ZRTP clients and implemented attack attack prototypes.

4.1.1 Systemization of Attacks

In the following, an excerpt of the most interesting attacks is presented for evaluating if the protocol has been properly implemented.

Version Downgrade: Older versions of the ZRTP protocol were vulnerable. While these design issues have been fixed, an implementation may also support older versions. In this attack, an attacker modifies the version header in transit to force the clients to use a vulnerable older version. As shown by previous research, ZRTP's version negotiation is not protected against downgrade attacks [64].

Weak Diffie-Hellman Parameters: During the initial Diffie-Hellman key exchange an attacker could force the clients to use an insecure public key of 1. Since in finite field Diffie-Hellman, the result is calculated as $DHResult = pvr^{svi}$ on the Initiators side and $DHResult = pvi^{svi}$ on the Responders side, a received public value of 1 always leads to $DHResult = 1$.

Invalid Shared Secret: In ZRTP, successive calls may use cached shared secrets derived from the initial Diffie-Hellman key exchange. This means that participants only need to verify SASs once. The case discussed here can happen when either the cache is corrupted or an actual attacker tries to impersonate one participant. The designers of ZRTP consider this a highly critical event that must result in an error dialog shown to the user.

Invalid Commit: ZRTP's underlying cryptographic building blocks include a hash commitment scheme, which allows to reduce the length of the verbally compared SAS to only 16-20 bit. This works by constraining the attacker to a single try during the handshake. For this to work, the implementation needs to verify that a committed cryptographic hash matches a calculated one. If this is not properly done, a Man-in-the-Middle attack can work by sending an invalid commit.

Shared Man-in-the-Middle: For this attack, a scenario is assumed where calls between Alice and Eve and between Bob and Eve have been conducted in the past. This means, that shared secrets have been cached for Alice-Eve on Alice' and Eve's device and for Bob-Eve on Bob's and Eve's device. As described in the 'invalid shared secret' attack, these are used to derive shared secrets for future calls. Now, Alice and Bob have a ZRTP-secured call, but Eve acts as a Man-in-the-Middle attacker. Because for every hop on the route Alice-Eve-Bob cached shared secrets exist, the connection works in principle. This attack can only be detected if caching of secrets is simply not implemented or cache entries were labeled to show Alice that Eve's secret is used instead of Bob's (called ZID labels).

4.1.2 Evaluation of ZRTP Clients

For the evaluation, six ZRTP clients have been selected: Acrobits Softphone, CSipSimple, Jitsi, Linphone Android, Signal on iOS and Android. The criteria for selection is that they must not operate in a closed-network (required for testing) and are actively used (> 100 000 installations). The most noteworthy results are summarized in Table 4.1.

A critical issue has been found in Linphone (CVE-2016-6271) not verifying the commit (cf. Invalid Commit). In addition, under certain conditions, a normal call in Jitsi was misinterpreted as an

Table 4.1: Noteworthy results from the evaluation of ZRTP clients. Full results can be found in the corresponding publication [7] starting on page 73.

Application	Operating System	Version	Library	Version Downgrade	Weak Diffie-Hellman	Invalid Shared Secret	Invalid Commit	Shared MitM
Acrobits Softphone	iOS	5.8.1	-	●	●	●	●	●
CSipSimple	Android	1.02.03	ZRTP4PJ	●	●	○	●	○
Jitsi	Win, Lin, MacOS	2.9.0	ZRTP4J	●	●	●	●	○
Linphone Android	Android	3.1.1	bzrtp	●	●	○	○[a]	○
Signal	Android	3.15.2	-	–	●	–[b]	●	–[b]
Signal	iOS	2.6.4	-	–	●	–[b]	●	–[b]

● = pass, ◐ = partially, ○ = fail, – = not supported

[a] CVE-2016-6271

[b] Signal is a *cacheless implementation.* It does not support preshared mode.

attack resulting in a false security warning (cf. Invalid Shared Secret). Finally, the Shared Man-in-the-Middle attack works for all clients except Signal, which does not implement a shared secret cache, and Acrobits Softphone, which implements ZID labels.

4.2 Device-to-Device Pairing using Gait

For zero-interaction pairings between devices worn on the same body, the BANDANA protocol has been designed. It uses human gait as a trust anchor to agree on a shared secret. The Body Area Network Device-to-device Authentication using Natural gAit (BANDANA) has been published first in “IEEE International Conference on Pervasive Computing and Communications” [10] (on page 91). A revised and extended version has been published in “Pervasive and Mobile Computing” [2] (on page 99). The contributions in this thesis are the design of the BANDANA protocol and the evaluation of its functionality. The co-author implemented data pre-processing steps and another co-author provided the attack based on video recordings.

The BANDANA protocol works by recording acceleration patterns on devices at the same time and then using a quantization scheme to derive bit fingerprints. These are used as a password for a PAKE. The protocol allows continuous re-pairing of devices and ad-hoc implicit security bound to a specific human’s body. No TTP is required for the key agreement.

4.2.1 Quantization Scheme

The quantization scheme defines the way a digital binary fingerprint is derived from the analog acceleration signal. The scheme must produce fingerprints with high similarity between different sensor locations on the same body, while it must not produce high similarity between different bodies. In this section the quantization scheme is summarized following the protocol flow in Figure 4.1. A more formal definition using mathematical notation can be found in the corresponding publication.

First, the acceleration sequences are pre-processed to account for different body alignments, e. g., when an accelerometer is attached to a swinging arm, it produces different results in comparison to one attached to the waist. For this, sensor fusion utilizes gyroscopes for correcting the orientation error. Madgwick’s algorithm [65] is used to rotate all measurements such that the z-axis always

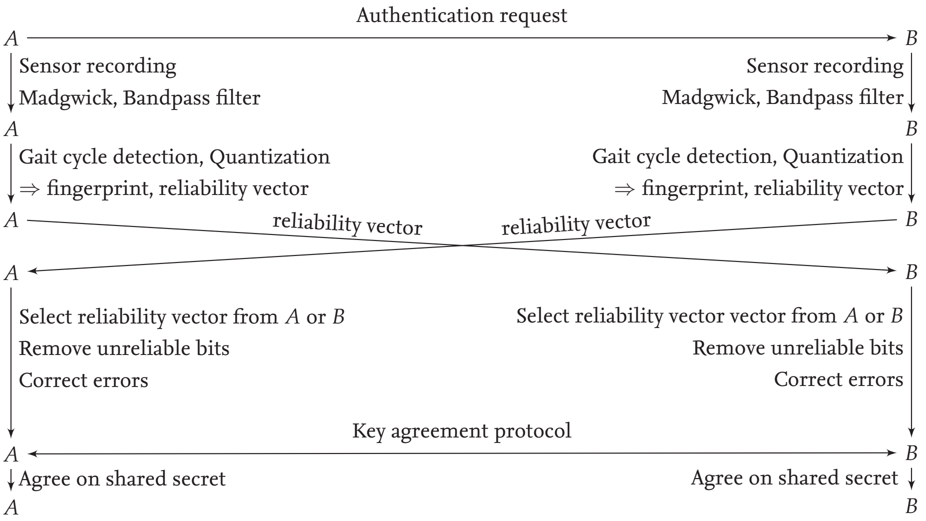

Authentication request
A
B
Sensor recording
Madgwick, Bandpass filter
Sensor recording
Madgwick, Bandpass filter
A
B
Gait cycle detection, Quantization
⇒ fingerprint, reliability vector
Gait cycle detection, Quantization
⇒ fingerprint, reliability vector
A
reliability vector
reliability vector
B
A
B
Select reliability vector from A or B
Remove unreliable bits
Correct errors
Select reliability vector vector from A or B
Remove unreliable bits
Correct errors
Key agreement protocol
A
B
Agree on shared secret
Agree on shared secret
A
B

Figure 4.1: Simplified BANDANA protocol sequence between two devices A and B worn on the same body. A detailed version can be found in the corresponding publication [2] starting on page 99.

faces in the opposite direction of the gravity. For BANDANA, the user's facing direction is not required and thus not corrected. By evaluating the gait's spectral coherence it has been found that movements between 0.5 Hz and 12 Hz are significantly correlated. Thus, a Type-II-Chebyshev bandpass filter has been applied to remove these.

The resulting pre-processed gait sequence is separated into single gait cycles. One cycle is defined by the time interval between two successive steps. The separation is done by finding repetitive parts using autocorrelation. The resulting autocorrelation is used to extract non-ambiguous local maxima with a mean distance between them. Cycles are selected by using this mean distance to select clear-cut minima. Finally, cycles are normalized to 40 samples/gait (original frequency is at maximum 50 Hz).

The final quantization step is inspired by [66], but instead of exploiting the difference to a mean gait of all participants, the differences to the mean of a specific gait sequence are calculated. A mean gait cycle is defined by summing up each sample individually over all gait cycles and dividing each sample by the number of gait cycles in the gait sequence. Finally, 4 bits are quantized per gait cycle by dividing it into 4 parts. The difference between the sample in each part in comparison to the mean gait cycle translates to 1 bit. If the difference is positive, a 1 is used, otherwise a 0. In BANDANA, 12 gait cycles are required to generate 48 bit fingerprints. To increase similarity between different sensor locations, the most unreliable bits are disregarded. This is done by removing the bits generated with a calculated difference close to 0.

The generated fingerprints are not perfectly matching and cannot be used to directly authenticate a key agreement. To account for the remaining differences between sensor locations, a (K, N)-error correcting code is used. As evaluated, the threshold must be set around 75 % error correction, i. e., a $(32, 16)$-error correcting code is chosen to correct up to $\frac{32-16}{2} = 8$ bits. The resulting 16 bit keys can be used as a password for the PAKE.

4.2.2 Security Model

BANDANA can be integrated with various PAKEs. To use the proposed number of only 16 bit it must provide a two-party adversarial model, where the attacker is reduced to a one-shot attacker. This is typically done by extending the Diffie-Hellman key exchange with a hash commitment as in ZRTP (cf. Section 4.1). The main goal is that the chance of a successful attack should not depend on an attackers offline computing power, but solely on the interaction during the protocol execution. PAKEs can roughly be categorized by (a) their way of storing the password, (b) encrypting transmitted public-keys, and (c) their number of participants [67]. In BANDANA, a “balanced” PAKEs should be used to derive a shared secret on both sides because either party can initiate an exchange (a). Whether public-keys are transmitted encrypted or not can independently be chosen as it is not influenced by BANDANA’s threat model (b). The focus lies on a two-party adversarial model (c). There exists a range of other security properties. Here it is important to note that BANDANA does not require passkey secrecy of a previous authentication attempt. For real-world deployments, an integration with Bluetooth is crucial. Bluetooth 4.2 with *Secure Connection* and *Secure Simple Pairing* fits well into BANDANA’s threat model. BANDANA can be integrated as an additional Out of Band (OoB) mode besides Near-Field Communication (NFC), providing the error-corrected key as the Bluetooth passkey. This is considered secure under the PE(i) model in [68].

To estimate the security of BANDANA’s key size, it is evaluated in comparison to established PAKE models. In the original security model by Vaudenay [46], 2^{10} parallel protocol runs are allowed. In BANDANA, parallel protocol runs are forbidden. In addition, threat models, such as the one by Farb et al. [69], choose a relatively high key length of 24 bit to even have a negligible attacker’s success probability when only 16 out of 24 bits are compared correct. Similar margins have been chosen in Bluetooth for PIN comparison with ~ 20 bits and ZRTP for word comparison with 20 bits. These key lengths include additional margins to be resilient to differences potentially staying unnoticed during the manual comparison by users. In contrast, BANDANA can keep a smaller margin as the key is generated automatically and not manually compared by users. Thus, a target bit size of 16 bits with a one-shot success probability for the attacker of 2^{-16} is proposed.

In the following, the one-shot success probability is calculated. An attacker may want to exhaust the key-space $\mathcal{C} = \mathbb{F}_{2^{16}}$. In BANDANA, after each single try, a completely new process is started generating a new independent key, making it impossible to exhaust $\mathcal{C}$. For 48 bit long sequences, BANDANA’s full process takes about ~ 12 s. Thus, an optimal imposter is constrained to not more than ~ 7200 tries per day. From each 48 bit sequence, 16 bit are disregarded for reliability amplification. From the remaining 32 bit fingerprints, up to 8 bit are error-corrected, resulting in 16 bit long keys. The success probability of a single randomly drawn fingerprint is therefore

$$\sum_{k=0}^{8} \binom{32}{k} / 2^{32} = \frac{\sum_{k=0}^{8} \left(\frac{32!}{(32-k)! \cdot k!} \right)}{2^{32}} \approx 0.0035.$$

4.2.3 Evaluation

As discussed before, BANDANA must produce fingerprints with high similarity between different body parts on the same body, while it must not produce high similarity between different bodies. To evaluate this, BANDANA was implemented to generate keys from two gait datasets and for various gait types (walking, running, descending and ascending stairs). The *Mannheim* dataset [70], previously used for position aware activity recognition, and the *Osaka* OU-ISIR Gait Database [71] were utilized.

The first features 15 subjects with 7 sensors on different body parts performing different activities, while the second features 482 subjects with triaxial accelerometers and gyroscopes worn on different parts of the waist (left, right, center).

As expected, the similarity between different bodies is centered at 50 %, which is the same as a similarity between two randomly generated bit sequences. For similarity between body parts, but the same body, torso and head are correlated with 81 % on average during walking, while other parts are correlated with 75-78 % on average. During running, these values are more homogeneous because acceleration is stronger propagated over the whole body. Descending and ascending stairs produce even better unique fingerprints with 87 % on average on the upper body. These results already include the previously discussed step of removing unreliable bits, which accounts for an increase of 3 %-points. The fingerprints during walking, generated from the *Mannheim* dataset, were confirmed using the large scale *Osaka* dataset, which produces a similarity of 75 % on average.

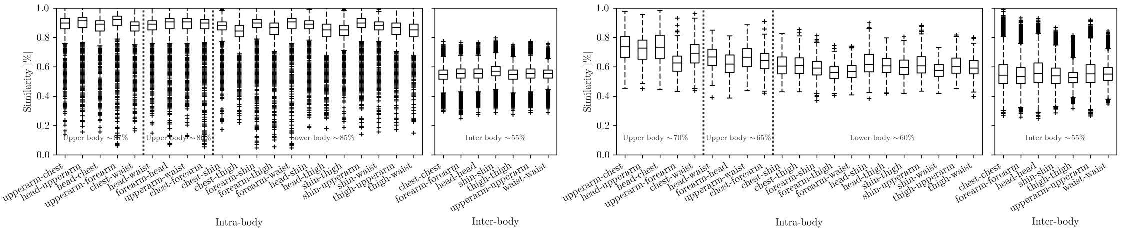
4.3 Entropy and Security Analysis of Gait Pairings

Besides BANDANA, other device-to-device pairing protocols based on gait have been proposed. In this work, relevant quantization schemes—including BANDANA—have been evaluated and compared. This work has been submitted to "IEEE Transactions on Mobile Computing (TMC)" [1] (on page 115). The contributions in this thesis are the evaluation of the functionality of discussed quantization schemes, the systemization of possible attack scenarios, and specific evaluations of BANDANA leading to improvements. The implementation of the quantization schemes and the entropy evaluation have been done in cooperation with the main co-author. Another co-author provided the experimental results for the video recording attack. All authors contributed equally to this work and are listed in alphabetical order.

The evaluation concentrates on device-to-device pairing protocols. Authentication schemes using gait as a biometric pattern to unlock devices are not considered. In contrast to pairing protocols, using gait for authentication requires the extraction of reproducible features to make the unlock process work each time. Thus, similar to how easily fingerprints leaked, traces of a person's gait is easily left on videos. Instead of choosing characteristics of a human's gait that are independent of the time, for pairing protocols, the opposite is chosen: Characteristics that are only unique for the time span a specific pairing is executed. First, a comparison of appropriate time-dependent quantization schemes is given. A classification of attacks is presented and possible countermeasures are discussed. The main contribution is the randomness evaluation, i. e., entropy, of generated keys from each quantization scheme.

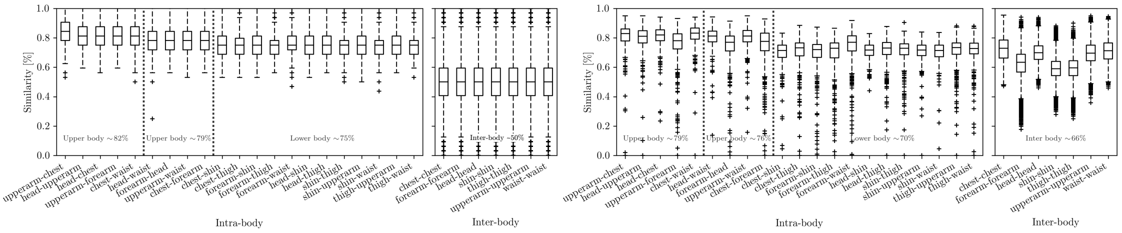
4.3.1 Quantization Schemes

After BANDANA has been described in detail in Section 4.2, in this section, the quantization algorithms Simple Accelerometer based wireless Pairing with HEuristic trees (SAPHE) [60], Walkie-Talkie [61] and Inter-Pulse-Interval (IPI) [62] are summarized. Here, only the basic ideas are summarized. A detailed more formal description using mathematical notation can be found in the corresponding publication. The schemes have been implemented and tested against the *Mannheim* dataset [70]. Generally, all schemes start by recording acceleration sequences independently on the participating devices. In addition, all schemes require a method to pre-process the sequences to account for differently oriented body parts, though, the details how re-orientation is implemented differ between the schemes.



(a) SAPHE

(b) Walkie-Talkie



(c) BANDANA

(d) IPI

Figure 4.2: Comparison of intra-body against inter-body similarity for the evaluated quantization schemes. Each value in the *intra-body* boxplot is defined by the similarity of two *different* sensor locations on the same subject (all possible combinations within each subject). For *inter-body*, each boxplot defines a different sensor location. Only *different* subjects are tested against each other with the *same* sensor location. Results are part of the corresponding publication [1] starting on page 115.

SAPHE The scheme works by randomly generating points with a fix period in the range of the recorded acceleration signal on both devices. A hash commitment (as in ZRTP) is executed between the devices to disallow modification of the values after this step. Recorded acceleration signal points are compared to the generated ones and each device generates a challenge. Each result of this challenge is defined by whether a signal point exceeds a random point or not.

As depicted in Figure 4.2a, SAPHE generated keys with high similarity of over 85 % for different sensor locations, while having 55 % similarity over different bodies (5 % higher than random guess). Conclusively, despite its simplicity, it works extremely well.

Walkie-Talkie The scheme uses a fairly complex quantization scheme with a high bit generation rate of 1 key bit per signal sample. Before gait recordings, devices agree on a heel-strike count. Independent Component Analysis (ICA) is utilized to account for differences between sensor locations. Because Walkie-Talkie also includes the user facing direction for quantization, the coordinate system must be fully re-oriented on all axis using a gyroscope. The recorded signal is separated in three regions using a guard band around 0. Signal values inside the guard band are disregarded. Values above (below) the guard band are quantized as 1 (0) per signal sample. A *reconciliation* step ensures that devices agree on samples in the sequence that shall constitute the key. Because the algorithm produces long consecutive sequences of 1s or 0s, the authors use a step called *privacy amplification* mixing consecutive 30 bit long windows via XOR.

Walkie-Talkie generated keys with a probability of 60-70 % for upper body and 55-65 % for lower body locations on the same body, while having 55 % similarity over different bodies (cf.

Figure 4.2b). The results show that further error correction is required to reliably produce keys using Walkie-Talkie.

IPI While BANDANA normalizes the step length to 40 samples/gait, IPI exploits this variance to generate keys. The individual bits per gait cycle are derived using a graycode. Pre-processing is done as in BANDANA.

IPI exhibited a similarity of 70-79 % for different sensor locations on the same body (cf. Figure 4.2d). Unfortunately, the similarity over different bodies is similar for some combinations. For example, the similarity between keys generated on the chest of different bodies is with 70 % on average better than some combinations on the same (lower) body. There is no clear security margin, selecting one at 70 % would produce false positives with a high probability.

4.3.2 Entropy Evaluation and Algorithm Improvements

Traditional statistical tests for evaluating the entropy are the DieHarder and ENT Pseudorandom Number Sequence Tests. These tests uncovered some obvious problems with IPI using the *birthdays* test, the *Overlapping Quadruples Sparce Occupancy (oqso)* test, the *3D sphere* test as well as the *rgb permutation* and *rgb Kolmogorov Smirnov* test. The results of the ENT test heavily depend on the chosen encoding, e. g., the number of bits per gait cycle in BANDANA. However, no significant problems have been uncovered for other quantization schemes. It is concluded that with the low number of keys, no meaningful results can be produced using traditional tests.

Thus, in this paper novel visual evaluation methods are provided. A major factor defining the entropy is the distribution of bits over a number of generated keys. In this work, visual heatmaps based on a Galton board are proposed. The distribution is shown from top to bottom by selecting left on 0 and right on 1, i. e., the heatmap's height is defined by the key length. The color intensity in the heatmap defines how many bit sequences follow the same pattern. From these heatmaps, a markov property can be derived and plotted separately. It depicts the probability of assigning 1 for the *n*th position in these keys. Finally, a cumulative sums distributions is plotted using the heatmaps' last rows.

The heatmaps revealed that SAPHE carries some characteristics of the original acceleration signal over to the generated keys. Walkie-Talkie shows no interesting deviations, BANDANA's heatmap is too narrow, i. e., of low variance, and IPI shows a significant bias towards 1s. The markov property exhibits patterns in IPI and Walkie-Talkie. In IPI, 4-bit-chunks are repeated with a probability of 60 % paving the way for a practical attack. Walkie-Talkie's periodicity exhibited here is attributed to the *privacy amplification* step.

BANDANA's issues exhibited here have been evaluated further by plotting the distribution of individual gait cycles instead of full keys. This showed a bias towards the patterns 1010 and 0101. By normalizing both the mean and instantaneous gait prior to comparing them and disregarding pattern according to inverse occurrences probabilities, BANDANA was substantially improved. It now exhibits a good distribution with only a slight bias towards 1s.

4.3.3 One-Shot Success Probability

A detailed security analysis is done by considering relevant points of attack in a conceptual overview generalized over all gait based pairing protocols. This summary focuses on the success probability of a theoretical attacker without additional knowledge of the victim's gait. She may either execute a Man-in-the-Middle or impersonation attack by brute forcing the key. For comparison between protocols, the same key length of 16 bit is assumed. In all protocols, the key space can not be exhausted by naive brute force as a completely new pairing process is started after each failed attempt. Thus, the attacker is reduced to a one-shot adversary.

The probability for a single randomly drawn key in SAPHE and Walkie-Talkie is

$$\frac{1}{2^{16}} \approx 1.52588 \cdot 10^{-5}.$$

Attacking BANDANA has a success probability of 0.0035. The corresponding equation is presented in Section 4.2.2. Since IPI follows BANDANA by employing a error-correcting code, it has the same success probability.

5 Usability

The usability of verifying trust anchors plays an essential role for the adoption of decentralized applications and services. Therefore, usable security has always been a goal during this research and let to the design of high-level APIs abstracting away the difficulties of cryptographic implementations. To evaluate the end-user's performance and understanding when engaging with interactive trust anchors, user studies have been conducted.

In this section, two research projects are presented. The first evaluates the usage of security tokens over NFC. They provide a tighter coupling between public keys and real-world objects by using tamper-proof Integrated Circuits (ICs). Read-out of corresponding secret keys from these external tokens is prevented [72]. Due to the complexity of implementation and usage, an architecture has been designed and evaluated for the usage of security tokens over NFC. In a laboratory study, the form factors smart card and NFC ring were compared with traditional password-based protection. The second work evaluates textual key-fingerprints, i. e., human-readable representations of public keys [73]. For this, the currently practiced key-fingerprint algorithms have been compared and an attacker model has been defined by calculating its brute force success probability. The discussion in this section is primarily based on the following publications: [6, 13]. These publications are part of this article thesis and hence are included in this document. An overview can be found in Section 1, whereas the publications can be found starting on page 43.

5.1 Architecture for Security Tokens over NFC

An architecture for using security tokens, such as smart cards, as physical trust anchors over NFC has been published in "Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)" [6] (on page 129). In recent years, public key encryption has been integrated into an increasing number of smartphone apps, e. g., to provide end-to-end encryption of cloud storage and messaging services. Unfortunately, Original Equipment Manufacturers (OEMs) still do not accomplish timely updates for smartphones leaving them in a state of insecurity. Thus, secret keys generated and stored on the devices themselves are potentially exposed by network and operating system attacks. Even when systems are up-to-date, widespread vulnerabilities in messaging apps have been found that expose secret keys, e. g., by the author of this thesis [14]. The contributions in this thesis are the design and implementation of the full-stack architecture, its API, threat model, and comparative evaluation. While the study has also been conducted by the author of this thesis, the design and analysis has been done in collaboration with the paper's co-author.

To solve this issue, security tokens are proposed to serve as external trust anchors implementing only the required operations of public key cryptography. By binding secret keys to physical objects and preventing memory access, they can be guaranteed to be unique. An additional PIN consisting of 6 numbers provides access control. While there exist USB On-the-Go cables for connecting peripherals, users are not willing to carry around additional cables for their smartphones. Thus, NFC can be leveraged to power external security tokens and communicate with them via induction. In this work, requirements and a threat model have been defined resulting in a full stack architecture

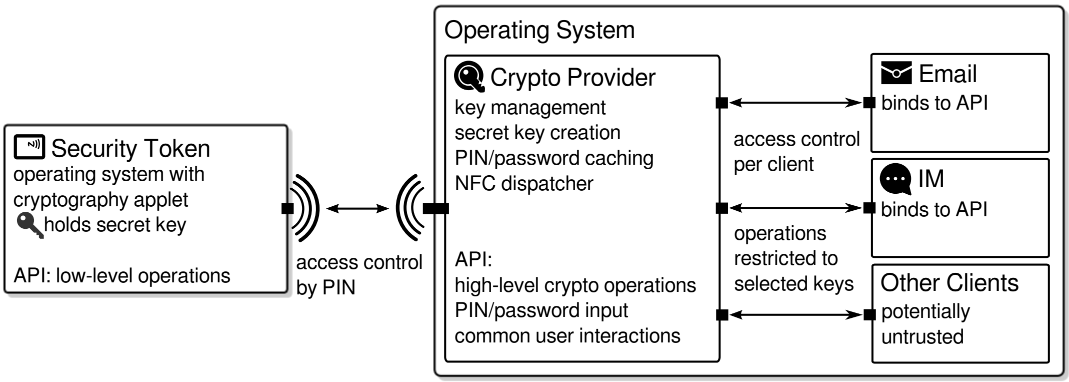


Figure 5.1: Architecture Overview (Taken from the corresponding publication [6] starting on page 129)

as shown in Figure 5.1. It provides an Android API with key management, PIN/password caching usable by multiple applications while never exposing the secret key itself over NFC. Previous research has shown that developers often make errors when using cryptographic APIs [74]. To prevent misuse, this API provides all necessary user interfaces to increase developer usability in contrast to traditional APIs that solely provide low-level operations. To evaluate the performance and usability, a user studies with 40 participants has been conducted. In addition to testing the typical smart card form factor, an NFC ring has been created, which can be worn on the finger.

5.1.1 Threat Model

In comparison to trust anchors in vehicular networks, impersonation and Sybil attacks are already inherently solved by binding secret keys to security tokens. There is no additional information, such as traffic or parking information, which require an assessment by reputation ratings. The threat model discussed in this work can be categorized into attacks against the communication method NFC per se, the security token, and the smartphone. In the following, representative attacks are summarized:

Relay Attacks: A NFC connection can be established by holding a NFC reader against the victim's trouser pocket and relaying the connection over the Internet to decrypt data on the other end. This attack is typically executed for carrying out payment fraud. In this architecture it is prevented by requiring a PIN.

Eavesdropping: It has been shown that NFC connections can be eavesdropped in a range of up to 29 cm [75]. A perfect countermeasure can be provided by implementing the Secure Remote Password (SRP-6a) protocol [76]. Even without this, the implications are small, because plaintext is not communicated over NFC. Instead, only a decrypted session key is transmitted back to the smartphone. Conclusively, an attacker needs to compromise both the NFC connection and the targeted online service.

Man-in-the-Middle: For a Man-in-the-Middle (MitM) attack, NFC needs to be blocked, which is difficult to achieve. With an active-passive or passive-active connection, an attacker has to both block the originator's channel and to create an own RF field with perfect timing [77]. Conclusively, these attacks are considered extremely difficult.

Side-Channel Attacks: Side-channel attacks exploit information available when measuring power or time consumption of the cryptographic operations executed on the security token. In the worst case, non-deterministic cryptography can lead to measurable time differences, which in turn allow to derivate the secret key after observing their execution several times.

UI Spoofing/Task Hijacking: A malicious app installed on the smartphone can mimic the PIN input dialog and thus persuading the user to enter her PIN. More sophisticated attacks building upon this scenario exploiting Android specific mechanisms have been proposed and discussed [78, 79]. This can be prevented by letting the user choose a personal image shown in all trustworthy UI components. While the design of the trusted app is known to an attacker, the personal image is not and works like an additional trust anchor between user and app.

5.1.2 API Design

One of the primary goals was to design a high-level API for NFC-based cryptography. This has been achieved by designing a callback-based API. Using the two methods `SIGN_AND_ENCRYPT` and `DECRYPT_VERIFY` the basic functionality is exposed to the developers. A novelty here is that missing or invalid parameters do no lead to a crash on unhandled errors. Instead a special Inter-Process Communication (IPC) object (PendingIntent) is returned, which can be used to show user interfaces handling the situation directly with the end-user. For example, this is provided for missing public keys, password/PIN input and caching. Thus, even developers without knowledge of public key cryptography can effectively use the API successfully. Especially in the case of NFC communication, this highly simplifies the situation for developers, because the complex interaction of holding security tokens against the smartphone's NFC reader is implemented in a ready-to-use user screen. This interaction is automatically executed, when a secret key is not available in the device' storage, but on an external security token.

5.1.3 Evaluation

The API has been compared with existing cryptographic APIs. Its novelty lies in the included user-facing interactions and support of high-level operations independent of the storage-location of the secret keys (on-device or over NFC). The proposed API has been rolled out to over 100 000 users of the app *OpenKeychain* on Google Play and works in conjunction with an email client, password manager, and messaging client.

A performance evaluation shows that the day-to-day operations of generating signatures and decrypting session keys take less than 1 s over NFC. Because key generation on security tokens has been shown to be slow and unreliable, tokens are generated on the smartphone and moved to the security token's storage afterwards. A detailed user study to evaluate the usability of the user-facing API interactions as well as the form factors smart card and NFC rings has been conducted. The study with 40 participants is summarized in the Section 5.

5.1.4 User Study

A user study of the architecture for NFC-based cryptography has been done to evaluate the usability of user-facing interactions and favored form factors.

40 subjects participated in this study, which consisted of a lab experiment for measuring performance and a follow-up interview to analyze user perception. During the lab experiment, the study started with one of the authentication types smart card, NFC ring, or—for baseline comparison—

password-based protection. The NFC ring has been created especially for this study using an IC and a 3D printer. To prevent learning effects during the within-group design, the order was shuffled using Latin square. Three tasks were performed with each type: Key creation using the in-app wizard, read an encrypted email, and finally, reply with a secure email.

Objective measurements of user performance shows that the password-based protection mechanism performs significantly worse during the setup wizard with a median of 114.5 s in comparison to both NFC tokens ($p < 0.0001$). Smart card performed with 83.5 s and NFC ring with 68.5 s, but no significance between these two methods can be observed ($p = 0.083$). It is interesting to note that only 14 subjects were able to type in a valid password, while 22 subjects were able to position the ring correctly and choose a valid PIN on the first try. Similar relative results have been measured during the email decryption task.

The user perception has been evaluated by completing questions in relation to the tasks. The majority of the participants ranked the NFC-based methods higher in comparison to the password-based protection ($p < 0.0001$). However, no significant difference between smart card and NFC ring could be observed ($p = 0.073$). During the interview, several interesting insights were made. Paradoxically, some participants considered rings to be more easily misplaced than smart cards, while others viewed this the other way around. Also, notably 9 out of 33 men preferred smart cards instead of rings because they usually do not wear rings at all.

5.2 Empirical Study of Textual Key-Fingerprints

The work presented in this section has been published in "Proceedings of the 25th USENIX Security Symposium" [13] (on page 153). To authenticate an end-to-end encrypted communication channel without relying on central key management, key-fingerprints have long been used as trust anchors, which can be verified over phone calls or real-life encounters. Textual key-fingerprints are widely implemented, e. g., in OpenPGP or SSH, representing public keys as hexadecimal strings. Lately, key-fingerprints have been introduced to messaging systems, such as Signal and WhatsApp, using a numeric format. In this work, an online study with 1047 participants has been conducted evaluating the security, performance, and usability of 6 different representation schemes. Furthermore, an attacker strength model has been defined to establish recommended boundaries for key-fingerprint length. The contributions in this thesis are the evaluation on existing fingerprint schemes and the prototype implementation. The author of this thesis and the main author contributed equally to the design of the attacker strength model, i. e., the cryptographic details of the fingerprint method and the estimated attacker strength for partial preimage attacks. The user study, including the study platform and implementation of the algorithms, has been designed and conducted by the main author. Other authors provided valuable reviews and feedback.

5.2.1 Key-Fingerprint Representations and Algorithms

A great number of different key-fingerprint representations have been proposed over the years. In addition, there is no common standard which attributes a key-fingerprint calculation should include. This led to a number of different schemes with differing requirements and threat models. The most common representations are summarized here. In the following, $\mathit{SHA\text{-}1}(x)^n$ defines the execution of n rounds of nested *SHA-1* on x, a truncation to the leftmost n bits is defined by $x[0, \ldots, n]$, and pk is used as an abbreviation for a public key.

The *numeric* representation used by Signal, WhatsApp, and SafeSlinger uses only numeric digits to represent a public key. In case of WhatsApp specifically, the fingerprint is calculated by $SHA\text{-}256(pk)^{5200}[0,\dots,240]$. This fingerprint is split up into six chunks, where each chunk is represented by a five digits long number modulo 100 000 [32].

Alphanumeric representations consist of mixed numbers and letters to represent public keys. Typical examples are *Hexadecimal*, *Base32*, *Base58*, and *Base64*. In OpenPGP, fingerprints are calculated by $Hex(SHA\text{-}1(0x99 \parallel len \parallel 4 \parallel creation\_time \parallel algo \parallel pk))$ where len is the length, $creation\_time$ is the key creation timestamp and $algo$ is the algorithm identifier [80]. All implementations encode them in hexadecimal form with uppercase characters in 16 bit blocks separated by whitespaces with an additional whitespace after 5 blocks. In SSH, fingerprints are calculated by $Hex(MD5(Base64(algo \parallel pk)))$ [81, 82]. Fingerprints are encoded as “hexadecimal with lowercase letters and separated by colons” [81]. *Base32* uses the upper-case Latin alphabet and numbers without the letters O and I (due to the confusion with numbers 1 and 0). There is no difference between lower-case letters and upper-case letters. In ZRTP, the leftmost 20 bits of the 32 bit SAS value are directly encoded as Base32 (cf. ZRTP, Section 4.1). In addition to all characters in Base32, *Base64* also includes lower-case characters as well as the characters “+”, “/”, “=” and is implemented as an alternative to the hexadecimal representation since OpenSSH 6.8. In addition, SHA-256 is used instead of MD5.

A number of representation based on natural language have been proposed, but are used less often in practice. These include representations based on *word lists*, such as the PGP Word List, and algorithms deterministically generating *sentences* based on bit sequences.

5.2.2 Attacker Strength Model

Many of the presented algorithms in the previous section produce long key-fingerprints without properly defining a threat model. The longer the key-fingerprints are, the less usable they are as time required for visual or verbal comparison increases. To find a reasonable fingerprint length, a proper attacker strength model is defined and algorithms for key-fingerprint generation are discussed.

This model is explicitly constrained to the case of human comparison of key-fingerprints by textual representations. It is important to note that full collision resistance of utilized hash functions is not required in this model, while threat models of X.509 certificates must consider the case of full hash collisions. In case of direct human comparison of key-fingerprints, only preimage attacks need to be protected against, because no central trusted authority is involved. Properly parameterizing a memory-hard KDF allows for bit stretching, i. e., the security of a 112 bit fingerprint is increased to provide 2^{128} bit security [45, 83].

In this model, an attacker would need to find a 112 bit preimage for an existing key-fingerprint. In the following, a *partial preimage* attack is discussed that considers the case where a user leaves out some characters during key-fingerprint comparison. An attacker might try to find a preimage for the most obvious characters hoping that the rest are not compared correctly. Here, the first 24 and last 24 bits are considered to be controlled by the attacker using a partial preimage as these are the most obvious ones. In addition, of the 64 bit in the midle, it is assumed that 32 bits are controlled, i. e., using an attacker strength to find a 112 bit fingerprint under these constrained 80 bits. The probability of finding such a partial preimage for a fingerprint when executing 2^{49} brute force steps is calculated approximately by

$$1 - \left(\frac{2^{112} - \sum_{k=1}^{32} \binom{64}{k}}{2^{112}} \right)^{2^{49}} \approx 0.66.$$

The inner parentheses of this equation define the probability that no partial preimage exists for one specific bit permutation. Instead of using $\binom{64}{32}$, a sum over 32 variations has been inserted to include permutations with more than the uncontrolled 32 bit that are also valid partial preimages. Finally, the probability to find a partial preimage is defined by the inverse of the exponentiation.

5.2.3 Results

Under the specified attacker strength model, six fingerprint representations have been evaluated in an online study. Here, the results are only summarized as the study has been conducted primarily by the main author. The study showed that the attack detection rate and user perception for hexadecimal representations is significantly lower than those of most alternative ones. Representations based on natural language showed an improved attack detection rate but were perceived less secure. Multi-language applications might still consider the use of numeric or alphanumeric schemes. Here, numeric performed better than all alphanumeric schemes and achieved higher usability ratings.

6 Conclusions

Everyday devices and machines are becoming smart and interconnected. Applications previously only available on smartphones are increasingly deployed as unobtrusive solutions, e. g., interconnected vehicles exchanging parking information or wearable devices monitoring the patient's health. For these varied fields of application, heterogeneous network protocols have been designed and deployed in our everyday life. As these networks exchange sensitive information, a crucial requirement is their protection against adversaries. The security of the initial communication channel depends on how devices are introduced to each other. While it is easy to protect against passive eavesdropping by implementing a key agreement based on Diffie-Hellman, active Man-in-the-Middle attacks can only be detected by introducing an externally verifiable trust anchor. Recent years have shown that central authorities could be subverted by individuals and government agencies. Thus, trust should be established in an ad-hoc decentralized manner. This thesis spans over multiple protocols and scenarios, where *physical and biometric trust anchors* have been introduced to enable an ad-hoc way of establishing trust.

In vehicular networks, physical trust anchors are established for finding parking spaces via decentralized request response communication. The protocol is *end-to-end encrypted* and establishes a network of peers called 'Parking Communities' based on beta reputation ratings. A *threat model* and *attacker strength* calculation show the system's security scope. For *cooperative charging* of electric vehicles, ISO 15118 has been extended by a novel mechanism providing incentives to hand over charging stations to the next vehicle. It works by exchanging *digitally signed receipts and delegating parts of the payment to a potential successor.* An even stronger physical trust anchor is achieved using *security tokens.* An architecture is implemented and evaluated for providing *cryptography over Near-Field Communication (NFC).* For Voice over IP communication, *real-world attacks on the ZRTP protocol* have been evaluated. A novel attack, called 'Shared Man-in-the-Middle', has been introduced and several weaknesses in current implementations have been uncovered. *Algorithms for key-fingerprints,* a traditional trust anchor for public keys, have been compared and evaluated in regards to *partial preimage attacks.* Finally, the use of gait as a biometric trust anchor for Body Area Networks (BANs) is envisioned. Here, the unique *time-sensitive variations of the user's gait biometrics* are used as secrets for a key agreement. The final security model and pairing scheme requires only 12 seconds of human gait for a secure device pairing. The protocol has been compared to three other gait pairing protocols in the literature in regards to their *functionality, entropy, and threat model.* Based on the findings, the protocol has been improved and hardened.

6.1 Future Work

A lot of additional trust anchors can be envisioned for novel scenarios. While this thesis deals with Vehicular Delay-Tolerant Networks (VDTNs), BANs, VoIPs systems, NFC, and end-to-end encrypted messaging protocols, a lot more networks exist.

Physical trust anchors in vehicular networks should receive more attention by researchers and practitioners. Unfortunately, due to the difficulty of monetizing decentralized services, Mobility Operators are moving to central cloud-based approaches. More research and practical applications are necessary for routing traffic information over long-distances and evaluating their trustworthiness.

In the area of biometric trust anchors, the evaluation methods for information entropy should be standardized and consolidated in a test suite. The results from the analysis, comparing the quantization schemes, can be used to create a new protocol by selecting the individual ‘good parts’ of each scheme. Furthermore, alternative biometric features should be considered, such as heart rate and bioelectrical impedance measurements. By combining multiple biometric trust anchors, the security level could be maintained while reducing the time a pairing process takes.

Bibliography

[1] Arne Brüsch, Ngu Nguyen, Dominik Schürmann, Stephan Sigg, and Lars Wolf. “On the secrecy of publicly observable biometric features: security properties of gait for mobile device pairing”. Submitted to: *IEEE Transactions on Mobile Computing (TMC)*. 2018.

[2] Dominik Schürmann, Arne Brüsch, Ngu Nguyen, Stephan Sigg, and Lars Wolf. “Moves like Jagger: Exploiting variations in instantaneous gait for spontaneous device pairing”. In: *Pervasive and Mobile Computing* 47 (2018), pp. 1 –12.

[3] Dominik Schürmann, Georg von Zengen, Marvin Priedigkeit, and Lars Wolf. “uDTNSec: A security layer with lightweight certificates for Disruption-Tolerant Networks on microcontrollers”. Submitted to: *Annals of Telecommunications*. 2018.

[4] Ngu Nguyen, Caglar Yuce Kaya, Dominik Schürmann, Arne Brüsch, Stephan Sigg, and Lars Wolf. “Demo of BANDANA - Body Area Network Device-to-device Authentication Using Natural gAit”. Accepted for publication at: *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. 2018.

[5] Signe Rüsch, Dominik Schürmann, Rüdiger Kapitza, and Lars Wolf. “Forward Secure Delay-Tolerant Networking”. In: *Proceedings of the 12th Workshop on Challenged Networks*. CHANTS ’17. Snowbird, Utah, USA: ACM, Oct. 2017, pp. 7–12.

[6] Dominik Schürmann, Sergej Dechand, and Lars Wolf. “OpenKeychain: An Architecture for Cryptography with Smart Cards and NFC Rings on Android”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 1.3 (Sept. 2017), 99:1–99:24.

[7] Dominik Schürmann, Fabian Kabus, Gregor Hildermeier, and Lars Wolf. “Wiretapping End-to-End Encrypted VoIP Calls: Real-World Attacks on ZRTP”. In: *Proceedings on Privacy Enhancing Technologies* 2017.3 (July 2017), pp. 4–20.

[8] Dominik Schürmann, Georg von Zengen, Marvin Priedigkeit, and Lars Wolf. “uDTNSec: A Security Layer for Disruption-Tolerant Networks on Microcontrollers”. In: *Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. June 2017, pp. 1–7.

[9] Dominik Schürmann. “Ph.D. Forum: Establishing Trust in Heterogeneous Networks”. In: *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. Mar. 2017, pp. 107–108.

[10] Dominik Schürmann, Arne Brüsch, Stephan Sigg, and Lars Wolf. “BANDANA – Body Area Network Device-to-device Authentication using Natural gAit”. In: *IEEE International Conference on Pervasive Computing and Communications (PerCom)*. Mar. 2017, pp. 190–196.

[11] Dominik Schürmann, Sebastian Willenborg, Felix Büsching, and Lars Wolf. “RAIM: Redundant Array of Independent Motes”. In: *International Conference on Networked Systems (NetSys)*. Mar. 2017, pp. 1–8.

[12] Dominik Schürmann, Julian Timpner, and Lars Wolf. “Cooperative Charging in Residential Areas”. In: *IEEE Transactions on Intelligent Transportation Systems* 18.4 (Apr. 2017), pp. 834–846.

[13] Sergej Dechand, Dominik Schürmann, Karoline Busse, Yasemin Acar, Sascha Fahl, and Matthew Smith. “An Empirical Study of Textual Key-Fingerprint Representations”. In: *Proceedings of the 25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 193–208.

[14] Dominik Schürmann and Lars Wolf. “Surreptitious Sharing on Android”. In: *Sicherheit 2016*. Vol. P-256. Lecture Notes in Informatics. Bonn, Germany: Gesellschaft für Informatik, Apr. 2016, pp. 137–148.

[15] Julian Timpner, Dominik Schürmann, and Lars Wolf. “Trustworthy Parking Communities: Helping Your Neighbor to Find a Space”. In: *IEEE Transactions on Dependable and Secure Computing* 13.1 (Jan. 2016), pp. 120–132.

[16] Sergej Dechand, Dominik Schürmann, Jürgen Koslowski, and Matthew Smith. “Poster: CryptoCall: Simple End-to-End Cryptography for Voice Calls on Android”. In: *Network and Distributed System Security Symposium (NDSS)*. Feb. 2014.

[17] Julian Timpner, Dominik Schürmann, and Lars Wolf. “Secure Smartphone-based Registration and Key Deployment for Vehicle-to-cloud Communications”. In: *Proceedings of the 2013 ACM Workshop on Security, Privacy & Dependability for Cyber Vehicles (CyCAR ’13)*. Berlin, Germany: ACM, Nov. 2013, pp. 31–36.

[18] Dominik Schürmann and Stephan Sigg. “Poster: Handsfree ZRTP - A Novel Key Agreement for RTP, Protected by Voice Commitments”. In: *Symposium On Usable Privacy and Security (SOUPS)*. July 2013.

[19] Dominik Schürmann, Jörg Ott, and Lars Wolf. “Authenticated Resource Management in Delay-Tolerant Networks using Proxy Signatures”. In: *10th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. Banff, Alberta, Canada, Mar. 2013, pp. 44–51.

[20] Dominik Schürmann and Stephan Sigg. “Secure Communication Based on Ambient Audio”. In: *IEEE Transactions on Mobile Computing (TMC)* 12.2 (Feb. 2013), pp. 358–370.

[21] Felix Büsching, Andreas Figur, Dominik Schürmann, and Lars Wolf. “Poster: Utilizing Hardware AES Encryption for WSNs”. In: *Proceedings of the 10th European Conference on Wireless Sensor Networks (EWSN 2013)*. Feb. 2013, pp. 33–36.

[22] Stephan Sigg, Dominik Schürmann, and Yusheng Ji. “PINtext: A Framework for Secure Communication Based on Context”. In: *Mobile and Ubiquitous Systems: Computing, Networking, and Services*. Ed. by Alessandro Puiatti and Tao Gu. Vol. 104. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg, Dec. 2011, pp. 314–325.

[23] McKinsey Global Institute. *The Internet of Things: Mapping the Value Beyond the Hype*. June 2015. URL: `https://www.mckinsey.de/files/unlocking_the_potential_of_the_internet_of_things_full_report.pdf`.

[24] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications”. In: *IEEE Communications Surveys Tutorials* 17.4 (2015), pp. 2347–2376.

[25] Omprakash Kaiwartya and Sushil Kumar. “Geocast Routing: Recent Advances and Future Challenges in Vehicular Adhoc Networks”. In: *Proceedings of the International Conference on Signal Processing and Integrated Networks (SPIN)*. Noida, India: IEEE, Feb. 2014, pp. 291–296.

[26] Julian Timpner and Lars Wolf. “Query-response geocast for vehicular crowd sensing”. In: *Ad Hoc Networks* 36 (2016). Vehicular Networking for Mobile Crowd Sensing, pp. 435 –449.

[27] Car 2 Car Communication Consortium. *Manifesto: Overview of the C2C-CC system, V1. 1.* Aug. 2007.

[28] Zaher Dawy, Walid Saad, Arunabha Ghosh, Jeffrey G Andrews, and Elias Yaacoub. “Toward massive machine type cellular communications”. In: *IEEE Wireless Communications* 24.1 (2017), pp. 120–128.

[29] Vasco.com. *DigiNotar reports security incident.* Sept. 2011. URL: `http://www.vasco.com/company/about_vasco/press_room/news_archive/2011/news_diginotar_reports_security_incident.aspx`.

[30] Carl Ellison. “Establishing identity without certification authorities”. In: *USENIX Security Symposium.* USENIX Association, 1996, pp. 67–76.

[31] Namecoin Project. *Namecoin.* Nov. 2014. URL: `https://namecoin.org`.

[32] WhatsApp. *WhatsApp Encryption Overview.* Apr. 2016. URL: `https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf`.

[33] Facebook. *Messenger Secret Conversations.* July 2016. URL: `https://fbnewsroomus.files.wordpress.com/2016/07/secret_conversations_whitepaper-1.pdf`.

[34] Mark D Ryan. “Enhanced certificate transparency and end-to-end encrypted mail”. In: *NDSS Symposium 2014.* Internet Society. 2014.

[35] Marcela S. Melara, Aaron Blankstein, Joseph Bonneau, Edward W. Felten, and Michael J. Freedman. “CONIKS: Bringing Key Transparency to End Users”. In: *24th USENIX Security Symposium (USENIX Security 15).* Washington, D.C.: USENIX Association, 2015, pp. 383–398.

[36] Ming Ki Chong, Rene Mayrhofer, and Hans Gellersen. “A survey of user interaction for spontaneous device association”. In: *ACM Computing Surveys (CSUR)* 47.1 (2014), p. 8.

[37] Sini Ruohomaa and Lea Kutvonen. “Trust management survey”. In: *Trust Management.* Springer, 2005, pp. 77–92.

[38] Dusko Pavlovic. “Towards a science of trust”. In: *Proceedings of the 2015 Symposium and Bootcamp on the Science of Security.* ACM. 2015, p. 3.

[39] John Linn. “Trust models and management in public-key infrastructures”. In: *RSA laboratories* 12 (2000). URL: `ftp://ftp.rsasecurity.com/pub/pdfs/PKIPaper1.pdf`.

[40] Jingwei Miao, Omar Hasan, Sonia Ben Mokhtar, Lionel Brunie, and Kangbin Yim. “An Analysis of Strategies for Preventing Selfish Behavior in Mobile Delay Tolerant Networks”. In: *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS).* 2012, pp. 208–215.

[41] Rongxing Lu, Xiaodong Lin, Haojin Zhu, Xuemin Shen, and B. Preiss. “Pi: A practical incentive protocol for delay tolerant networks”. In: *IEEE Transactions on Wireless Communications* 9.4 (2010), pp. 1483–1493.

[42] Lifei Wei, Haojin Zhu, Zhenfu Cao, and Xuemin(Sherman) Shen. “MobiID: A User-Centric and Social-Aware Reputation Based Incentive Scheme for Delay/Disruption Tolerant Networks”. In: *Ad-hoc, Mobile, and Wireless Networks.* Ed. by Hannes Frey, Xu Li, and Stefan Ruehrup. Vol. 6811. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 177–190.

[43] Debbie Nathan. *Sybil exposed: The extraordinary story behind the famous multiple personality case.* Free Press, June 2011.

[44] National Institute of Standards and Technology. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions (FIPS PUB 202).* Tech. rep. Aug. 2015.

[45] Colin Percival. “Stronger key derivation via sequential memory-hard functions”. In: *The Technical BSD Conf. (BSDCan 2009).* 2009.

[46] Serge Vaudenay. “Secure Communications over Insecure Channels Based on Short Authenticated Strings”. In: *Advances in Cryptology – CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18,* ed. by Victor Shoup. Berlin, Heidelberg: Springer Berlin Heidelberg, Aug. 2005, pp. 309–326.

[47] Marco Fiore, Claudio Ettore Casetti, Carla-Fabiana Chiasserini, and Panagiotis Papadimitratos. “Discovery and Verification of Neighbor Positions in Mobile Ad Hoc Networks”. In: *IEEE Transactions on Mobile Computing* 12.2 (2013), pp. 289–303.

[48] Paul Furgale, Ulrich Schwesinger, Martin Rufli, Wojciech Derendarz, Hugo Grimmett, Peter Mühlfellner, Stefan Wonneberger, Julian Timpner, Stephan Rottmann, Bo Li, Bastian Schmidt, Thien Nghia Nguyen, Elena Cardarelli, Stefano Cattani, Stefan Brüning, Sven Horstmann, Martin Stellmacher, Holger Mielenz, Kevin Köser, Markus Beermann, Christian Häne, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, René Iser, Rudolph Triebel, Ingmar Posner, Paul Newman, Lars Wolf, Marc Pollefeys, Stefan Brosig, Jan Effertz, Cédric Pradalier, and Roland Siegwart. “Toward Automated Driving in Cities using Close-to-Market Sensors: An Overview of the V-Charge Project”. In: *Intelligent Vehicle Symposium (IV ’13).* Gold Coast, AU: IEEE, June 2013, pp. 809–816.

[49] Audun Jsang and Roslan Ismail. “The Beta Reputation System”. In: *In Proceedings of the 15th Bled Electronic Commerce Conference.* 2002.

[50] Julien Freudiger, Murtuza Jadliwala, Jean-Pierre Hubaux, Valtteri Niemi, and Philip Ginzboorg. “Privacy of Community Pseudonyms in Wireless Peer-to-Peer Networks”. In: *Mobile Networks and Applications* 18.3 (2013), pp. 413–428.

[51] Krishna Sampigethaya, Leping Huang, Mingyan Li, Radha Poovendran, Kanta Matsuura, and Kaoru Sezaki. “CARAVAN: Providing location privacy for VANET”. In: *Defense Technical Information Center* (2005).

[52] Wolf-Bastian Pöttner, Johannes Morgenroth, Sebastian Schildt, and Lars Wolf. “Performance Comparison of DTN Bundle Protocol Implementations”. In: *Proceedings of the 6th ACM Workshop on Challenged Networks.* CHANTS ’11. Las Vegas, Nevada, USA: ACM, 2011, pp. 61–64.

[53] Susan Symington, Stephen Farrell, Howard Weiss, and Peter Lovell. *Bundle Security Protocol Specification.* RFC 6257. IETF, May 2011.

[54] Simon Josefsson. *The Base16, Base32, and Base64 Data Encodings.* RFC 4648. IETF, Oct. 2006.

[55] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. “The ONE simulator for DTN protocol evaluation”. In: *Simulation Tools and Techniques (SIMUTools ’09).* New York, NY: ICST, 2009.

[56] Nationale Plattform Elektromobilität. *Fortschrittsbericht 2014 – Bilanz der Marktvorbereitung.* Berlin, Germany, Dec. 2014. URL: `https://www.bmbf.de/files/NPE_Fortschrittsbericht_2014_barrierefrei.pdf`.

[57] *ISO/IEC DIS 15118-1: Road vehicles – Vehicle to grid communication interface – Part 1: General information and use-case definition*. Int. Organization for Standardization, 2013.

[58] *ISO/IEC DIS 15118-2: Road vehicles – Vehicle to grid communication interface – Part 2: Network and application protocol requirements*. Int. Organization for Standardization, 2014.

[59] Phil Zimmermann, Alan Johnston, and Jon Callas. *ZRTP: Media Path Key Agreement for Unicast Secure RTP*. RFC 6189 (Informational). IETF, Apr. 2011.

[60] Bogdan Groza and Rene Mayrhofer. "SAPHE: simple accelerometer based wireless pairing with heuristic trees". In: *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*. ACM. 2012, pp. 161–168.

[61] Weitao Xu, Girish Revadigar, Chengwen Luo, Neil Bergmann, and Wen Hu. "Walkie-Talkie: Motion-Assisted Automatic Key Generation for Secure On-Body Device Communication". In: *15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 2016, pp. 1–12.

[62] Yingnan Sun, Charence Wong, Guang-Zhong Yang, and Benny Lo. "Secure key generation using gait features for Body Sensor Networks". In: *IEEE 14th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. 2017, pp. 206–210.

[63] Riccardo Bresciani and Andrew Butterfield. "ProVerif Analysis of the ZRTP Protocol". In: *International Journal for Infonomics (IJI)* 3.3 (2010).

[64] Karthikeyan Bhargavan, Christina Brzuska, Cédric Fournet, Matthew Green, Markulf Kohlweiss, and Santiago Zanella-Béguelin. "Downgrade Resilience in Key-Exchange Protocols". In: *IEEE Symposium on Security and Privacy (SP)*. May 2016, pp. 506–525.

[65] Sebastian OH Madgwick, Andrew JL Harrison, and Ravi Vaidyanathan. "Estimation of IMU and MARG orientation using a gradient descent algorithm". In: *2011 IEEE International Conference on Rehabilitation Robotics*. IEEE. 2011, pp. 1–7.

[66] Thang Hoang, Deokjai Choi, and Thuc Nguyen. "Gait authentication on mobile phone using biometric cryptosystem and fuzzy commitment scheme". In: *International Journal of Information Security* 14.6 (2015), pp. 549–560.

[67] Joern-Marc Schmidt. *Requirements for Password-Authenticated Key Agreement (PAKE) Schemes*. RFC 8125. IETF, Apr. 2017.

[68] Raphael C.-W. Phan and Patrick Mingard. "Analyzing the Secure Simple Pairing in Bluetooth v4.0". In: *Wireless Personal Communications* 64.4 (2012), pp. 719–737.

[69] Michael Farb, Yue-Hsun Lin, Tiffany Hyun-Jin Kim, Jonathan McCune, and Adrian Perrig. "SafeSlinger: easy-to-use and secure public-key exchange". In: *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*. ACM. 2013, pp. 417–428.

[70] Timo Sztyler and Heiner Stuckenschmidt. "On-body Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition". In: *International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2016, pp. 1–9.

[71] Thanh Trung Ngo, Yasushi Makihara, Hajime Nagahara, Yasuhiro Mukaigawa, and Yasushi Yagi. "The largest inertial sensor-based gait database and performance evaluation of gait-based personal authentication". In: *Pattern Recognition* 47.1 (2014), pp. 228–237.

[72] Michael Tunstall. *Attacks on Smart Cards.* 2006. URL: `http://www.cs.bris.ac.uk/home/tunstall/presentation/AttacksonSmartCards.pdf`.

[73] Peter Gutmann. “Do users verify SSH keys?” In: *USENIX;login:* 36.4 (2011).

[74] Manuel Egele, David Brumley, Yanick Fratantonio, and Christopher Kruegel. “An Empirical Study of Cryptographic Misuse in Android Applications”. In: *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security.* CCS ’13. ACM, 2013, pp. 73–84.

[75] Henning Siitonen Kortvedt and Stig F. Mjolsnes. “Eavesdropping near field communication”. In: *The Norwegian Information Security Conference (NISK).* Vol. 27. 2009.

[76] Michael Hölzl, Endalkachew Asnake, René Mayrhofer, and Michael Roland. “Mobile Application to Java Card Applet Communication using a Password-authenticated Secure Channel”. In: *12th International Conference on Advances in Mobile Computing and Multimedia (MoMM).* New York, NY, USA: ACM Press, 2014, pp. 147–156.

[77] Ernst Haselsteiner and Klemens Breitfuß. “Security in Near Field Communication (NFC)”. In: *Printed Handout of Workshop on RFID Security (RFIDSec).* Philips Semiconductors, July 2006.

[78] Brett Cooley, Haining Wang, and Angelos Stavrou. “Activity Spoofing and Its Defense in Android Smartphones”. In: *Applied Cryptography and Network Security: 12th International Conference (ACNS).* Ed. by Ioana Boureanu, Philippe Owesarski, and Serge Vaudenay. Cham: Springer International Publishing, June 2014, pp. 494–512.

[79] Chuangang Ren, Yulong Zhang, Hui Xue, Tao Wei, and Peng Liu. “Towards Discovering and Understanding Task Hijacking in Android”. In: *24th USENIX Security Symposium (USENIX Security 15).* Washington, D.C.: USENIX Association, 2015, pp. 945–959.

[80] Jon Callas, Lutz Donnerhacke, Hal Finney, David Shaw, and Rodney Thayer. *OpenPGP Message Format.* RFC 4880 (Proposed Standard). IETF, Nov. 2007.

[81] Joseph Galbraith and Rodney Thayer. *The Secure Shell (SSH) Public Key File Format.* RFC 4716 (Informational). IETF, Nov. 2006.

[82] Tatu Ylonen and Chris Lonvick. *The Secure Shell (SSH) Transport Layer Protocol.* RFC 4253. Updated by RFC 6668. IETF, Jan. 2006.

[83] Alex Biryukov, Daniel Dinu, and Dmitry Khovratovich. *Argon2: the memory-hard function for password hashing and other applications.* Dec. 2015. URL: `https://password-hashing.net/argon2-specs.pdf`.

Acronyms

BAN Body Area Network

BANDANA Body Area Network Device-to-device Authentication using Natural gAit

CA Certificate Authority

CS Charging Station

ECC Elliptic Curve Cryptography

ECDSA Elliptic Curve Digital Signature Algorithm

ECIES Elliptic Curve Integrated Encryption Scheme

EID Endpoint Identifier

EV Electric Vehicle

HIBC Hierarchical Identity-Based Cryptography

IBC Identity-Based Cryptography

IC Integrated Circuit

ICA Independent Component Analysis

IPC Inter-Process Communication

IPI Inter-Pulse-Interval

KDF Key Derivation Function

MitM Man-in-the-Middle

NFC Near-Field Communication

NPE German National Electric Mobility Platform

OEM Original Equipment Manufacturer

OoB Out of Band

PAKE Password-Authenticated Key Agreement

PII Personally Identifiable Information

PKI Public Key Infrastructure

SAPHE Simple Accelerometer based wireless Pairing with HEuristic trees

SAS Short Authentication String

SIP Session Initiation Protocol

SSP Scheme-Specific Part

TLS Transport Layer Security

TTP Trusted Third Party

V2V Vehicle-to-Vehicle

VDTN Vehicular Delay-Tolerant Network

VIN Vehicle Identification Number

VoIP Voice over IP

Publications

Trustworthy Parking Communities: Helping Your Neighbor to Find a Space

Julian Timpner, *Student Member, IEEE,* Dominik Schürmann, *Student Member, IEEE,*
and Lars Wolf, *Member, IEEE*

Abstract—Cooperation between vehicles facilitates traffic management, road safety and infotainment applications. Cooperation, however, requires trust in the validity of the received information. In this paper, we tackle the challenge of securely exchanging parking spot availability information. Trust is crucial in order to support the decision of whether the querying vehicle should rely on the received information about free parking spots close to its destination and thus ignore other potentially free spots on the way. Therefore, we propose Parking Communities, which provide a distributed and dynamic means to establish trusted groups of vehicles helping each other to securely find parking in their respective community area. Our approach is based on high-performance state-of-the-art encryption and signature algorithms as well as a well-understood mathematical trust rating model. This approach allows end-to-end encrypted request-response communications in combination with geocast and can be used as an overlay to existing vehicular networking technologies. We provide a comprehensive comparison with other security architectures and simulation results showing the feasibility of our approach.

Keywords—VANET, Vehicular Networks, Parking Search, Trust Management, Reputation, Security, Identity Management

✦

1 INTRODUCTION

MODERN vehicles are equipped with an array of sensor systems and assistance functions, which can greatly enhance driving comfort and safety. However, in order to maximize their effect, these disparate systems need to cooperate with each other. Hence, vehicles do not have to rely on on-board sensors only, but can acquire further information from other systems, both mobile and fixed, in their environment. As an example, consider a scenario where a driver on his way home from work is interested in a free parking spot on his downtown home street. The vehicle thus uses a geocast (a specialized form of multicast, in which destination nodes are addressed by their geographic location instead of by their IDs) to send a corresponding query into the destination area. Here, vehicles use their sensor systems to gather information about their surroundings, such as distance to the closest objects (e.g., cars), and respond to the query originator. Thus, the vehicle can advise the driver where to find parking, preferably close to his home location.

In the example, trust is crucial in order to support the decision of whether the query originator should rely on the received information about free parking spots close to his destination and thus ignore other potentially free spots on the way. This bears the risk of learning that there is no available spot at all in the destination area, and the previously ignored spots might be taken by then. Conversely, trust alleviates prioritizing incoming queries and can provide an incentive to help other vehicles, such that they will also be provided with inquired information, in a tit-for-tat manner. Moreover, attackers are likely to try to gain an advantage, e.g., by providing false data to keep parking spots to themselves or by intercepting parking spot availability information in order to reach free spots earlier than competing drivers. Unfortunately, there is no easy way to decide which vehicles to trust, or more specifically, to what extent. Even if a Trusted Third Party (TTP) exists, for instance in form of a Certificate Authority (CA) providing pseudonym certificates [1], it cannot necessarily verify the trustworthiness of vehicle responses. In order to do so, it would require trusted sensors at each parking spot throughout the city, which is expensive [2] and requires infrastructure networking support.

We thus propose, design, implement, and evaluate the concept of Parking Communities, which, in the style of good neighborly help, provide a distributed and dynamic means to establish trusted groups of vehicles helping each other to find parking in their respective community area. Our approach is based on high-performance state-of-the-art encryption and signature algorithms, in particular Elliptic Curve Cryptography (ECC), as well as a well-understood mathematical trust rating model.

- *Manuscript received 29 June 2014; revised 16 Apr. 2015; accepted 21 Apr. 2015. Date of publication 28 Apr. 2015; date of current version 20 Jan. 2016. The authors are with the Institute of Operating Systems and Computer Networks, Technische Universität Braunschweig, 38106 Braunschweig, Germany, (E-mail: [timpner|schuermann|wolf]@ibr.cs.tu-bs.de).*

©2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works. This is the author's post-print version.
The definitive version was published in IEEE Transactions on Dependable and Secure Computing. DOI: 10.1109/TDSC.2015.2427838

1.1 Contributions

In this paper, we present the design, implementation and evaluation of Parking Communities, a novel trust management for vehicular parking applications without reliance on a central TTP or Road-side Units (RSUs). Its novel features include a distributed trust model for parking applications as well as encrypted and signed request-response communication in combination with geocast. It thereby achieves protection against impersonation, Sybil attacks, interception and tampering despite its distributed design. Further, it can be used as an overlay to existing vehicular networking technologies [1, 3], thus benefiting from established security mechanisms, e.g., pseudonym certificates for anonymity and location privacy. We give a detailed analysis of attack scenarios and describe our implementation of the proposed security architecture in IBR-DTN [4], an open source RFC 5050 [5] implementation. We further provide a comprehensive evaluation in terms of a comparative analysis with other key and trust management protocols and simulation results.

1.2 Outline

The remainder of this paper is structured as follows. Section 2 discusses related work in the field of key and trust management in vehicular networks. The proposed Parking Community concept is introduced in Section 3. Attack scenarios on Parking Communities and their mitigations are presented in Section 4. Section 5 describes a prototypical implementation in an overlay network based on IBR-DTN. We analyze the protocol in comparison to existing solutions in Section 6, which can also serve for balancing the implementation tradeoffs of Parking Communities. We provide simulation results in Section 7. The paper concludes in Section 8.

2 RELATED WORK

This section provides a short introduction to cryptographic fundamentals, such as ECC. Related work on vehicular key and trust management is discussed. A detailed comparison of how our key and trust management relates to existing ones can be found in Section 6.

2.1 ECC Fundamentals

ECC is a recognized cipher for vehicular networks and is already employed by the IEEE 1609.2 [3] and ETSI (TS 103 097) standards. From a theoretical perspective, ECC is based on the difficulty to solve the Elliptic Curve Discrete Logarithm Problem (ECDLP) [6]. Modern representatives of ECC signature algorithms are the Elliptic Curve Digital Signature Algorithm (ECDSA) [7] and Edwards-Curve Digital Signature Algorithm (EdDSA) [8]. In most cases, ECC is not directly used to encrypt messages; rather, the peers agree on a session key using key agreement protocols, such as Diffie-Hellman (DH) [9].

2.2 Key Agreement Fundamentals

In addition to the DH key agreement based on the Discrete Logarithm Problem, there also exist ECC variants, which require a smaller key size resulting in less energy, memory, and bandwidth consumption. DH-based key agreement protocols are designed for synchronous communications as opposed to the asynchronous Elliptic Curve Integrated Encryption Scheme (ECIES). Since end-to-end connectivity cannot be guaranteed in vehicular networks and the number of roundtrips should thus be minimized, the asynchronous ECIES is more feasible in this context.

2.3 Trust in Vehicular Networks

There is an urgent need to assess the quality of information received in vehicular networks, lest a node reports false or inaccurate information to gain an advantage, e.g., allegedly congested roads in the hope that other vehicles avoid them and thus clear the path. Hence, the notion of trust among nodes is an important issue. Trust allows vehicles to detect dishonest and malicious data and to give incentives for honest and altruistic behavior.

There is a rich literature on trust models, which is why we do not aim to provide a comprehensive summary here, but instead refer the interested reader to the excellent surveys on trust management in vehicular networks [10, 11]. In this paper, we focus on self-organizing trust models which do not rely on an online connection to a security infrastructure in order to retrieve trust ratings (though a *key* management infrastructure can be used to achieve accountability, as described in Section 2.4). Instead, nodes form trust relationships directly with each other. These models can be classified into entity-oriented, data-oriented, and hybrid trust models. Entity-oriented trust models [12] focus on modeling the trustworthiness of nodes, but typically do not evaluate the trustworthiness of the data itself. This issue is addressed by data-oriented models. Raya et al. [13], for instance, use several decision logics, such as Bayesian inference and Dempster-Shafer theory to determine the level of trust that can be put in the received data. Vinel et al. [14] evaluated the effects on the decision delay when deploying a majority consensus algorithm to decide upon safety messages. They were able to show that a majority consensus works in practice, while decision delays should not exceed 6 seconds. A drawback of these approaches and, typically, of data-oriented models in general, is that only ephemeral trust in data is established, but no long-term trust relationships between nodes are formed. Hybrid trust models combine both aforementioned approaches and model the trustworthiness of nodes and use the result to evaluate the reliability of received data. Patwardhan et al. [15], for instance, determine a node's reputation by validating its data, which is similar to the approach in Parking Communities. Yet, the authors assume that certain nodes are pre-authenticated and thus provide inherently trustworthy data. Parking Communities differ in that they do not assume any inherently trusted nodes. Instead, trust is only established by actually and physically validating received data. Similar to our approach, Park et al. [16] propose to make use of vehicles' daily commute routine to build up long-term reputation. The proposed system, however, relies heavily on support from roadside infrastructure, which we consider impractical.

To the best of our knowledge, we are the first to investigate a hybrid trust model with physical verification and no additional infrastructure support in the context of parking detection applications to build trusted communities.

2.4 Key Management

To allow for long-term reputation, accountability in form of non-repudiable key-identity bindings is vital. Common key management standards for vehicular communication are based on traditional Public Key Infrastructures (PKIs), subdivided into CA regions and extended with pseudonym certificates [1, 3, 17, 18]. RSUs are introduced as additional infrastructure for communication between vehicles and central services, such as pseudonym CAs. Key pairs are usually generated on the nodes themselves, and the binding of a key pair to a node's identity is verified by a CA. Certificates serve as a proof of this binding and can be verified by any node in the network. IEEE 1609.2 [3], for instance, defines the format of security messages and uses anonymous public keys to sign and verify messages and short-lived anonymous certificates to automatically revoke keys. Studer et al. [17] improves upon the IEEE standard and provides temporary anonymous certified keys and automatic key change when entering a new region.

An alternative to PKIs are key management techniques based on Identity-Based Cryptography (IBC), as proposed by several authors [19–24]. In IBC, public keys are derived from IDs, while all key pairs are generated and stored by a central trusted authority. Using a secret only known to this authority, key pairs are generated using a cryptographic pairing scheme, such as Weil Pairing [25], resulting in node IDs. Using the pairing scheme and public parameters, nodes in the network are able to directly derive public keys from the ID. It provides certificateless cryptography and requires no retrieval of public keys as PKI schemes do.

There is a typical tradeoff between PKIs and IBC—pseudonym certificates achieve a limited form of anonymity, while IBC has the advantage of binding keys to identities without certificates. In Parking Communities, we operate on a more abstract level and can thus use either system, allowing us to make the most appropriate choice per use case. Each Parking Community member regularly collects its fellow members' public keys (as described in Section 3), independent from whether these derive from pseudonym certificates or IBC IDs.

3 PARKING COMMUNITIES

The motivation for Parking Communities is the interest to learn about free parking spots before reaching a destination area. We consider a typical working day with people parking their vehicle on their home street by night, at a primary work place by day, and visit different areas mostly in the evening [26]. A driver on his way home from work, for instance, sends a corresponding query via geocast into the destination area. Vehicles driving through or parking in this area can use their sensor systems to gather information about their surroundings [27], such as distance to the closest objects (e.g., other parked cars), and respond to the query originator. In this scenario, each vehicle requires an estimate of the trustworthiness of its communication partners in order to prioritize incoming queries or to determine a response's validity. To this end, drivers (to be more precise, their vehicles) regularly visiting the same area, such as neighbors or co-workers, dynamically create trusted Parking Communities to cooperate in exchanging parking spot information. By establishing trust anchors, signed and encrypted communication with previously encountered vehicles is facilitated. Thus, message interception and tampering is mitigated. Through a sophisticated mathematical rating model, vehicles dynamically establish an estimate of other vehicles' trustworthiness, without the need of a central TTP or RSUs.

In this section, we present the conceptual design of the Parking Community protocol.

3.1 Creating a Community

A vehicle uses a new public/private key pair $\langle pk, sk \rangle$ (obtained via IBC or PKI) exclusively for each community c. Further, c includes a trust anchor τ, consisting of a set of areas $\mathcal{A}$ mapped to a set $ID_c \subset \mathcal{ID}$ of IDs encountered in these areas, i.e., vehicles that are part of the community c. Moreover, c comprises a mapping σ of each vehicle v's ID $id_v \in ID_c$ to two counting variables r_v and s_v. Formally, c is defined by the tuple

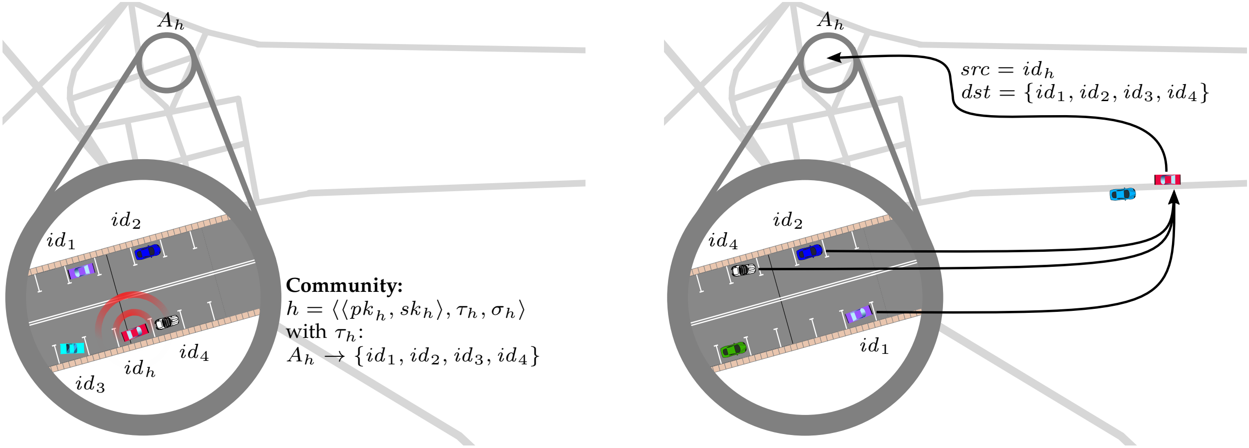
$$c = \langle \langle pk, sk \rangle, \tau, \sigma \rangle, \text{ with} \tag{1}$$

$$\tau : \mathcal{A} \to \mathcal{ID}, \tag{2}$$

$$\sigma : ID_c \to \{r, s\}. \tag{3}$$

In vehicular networks, there is no need to use human-readable IDs because networks are created ad hocly without human interaction, which allows us to generate them randomly. Because of this, we propose encoding pk_c directly as a vehicle's community ID, id_c. Thus, knowledge of id_c enables encrypted message exchange without prior key retrieval from TTPs.

Referring to the running example, suppose a driver returning home at night and parking on his home street. After the engine is turned off, a new home community h with $id_h = pk_h$ is generated for the home parking area, if it does not exist yet. Else, the existing home


A_h
id_1
id_2
Community:
$h = \langle\langle pk_h, sk_h\rangle, \tau_h, \sigma_h\rangle$
with τ_h:
$A_h \to \{id_1, id_2, id_3, id_4\}$
id_4
id_h
id_3
A_h
$src = id_h$
$dst = \{id_1, id_2, id_3, id_4\}$
id_4
id_2
id_1

(a) Collecting IDs via neighbor discovery with physical verification and establishing a trust anchor

(b) Encrypted and signed query/response for a free spot via geocast

Figure 1: Creating and querying a Parking Community

community is selected based on location information. For communications with the community, id_h is actively used as source address src. For privacy reasons, a more sophisticated scheme is required in practice, which we describe in Section 4.4.

As depicted in Figure 1a, IDs (i.e., public keys) of vehicles in the home area A_h are collected in the set ID_h via neighbor discovery while parking. To prevent Sybil attacks, position announcements of vehicles can be verified with a high probability as shown by previous work [28]. $A_h \to ID_h$ is added as a mapping to the trust anchor τ_h. Vehicle with id_h adding vehicle id_1 to its Parking Community does not require that the vehicle with id_1 adds id_h (cf. Figure 1a). Thus, Parking Communities are not reciprocative and typical secure group management primitives such as join and leave are not required. Vehicles are only responsible for their own sets of communities. This reduces the communication overhead as no messages for group management are required. The mapping σ is initialized with $r = s = 0$ for each $id \in ID_h$.

As the engine is started again, e.g., when the driver leaves for work, the ID collection for this community is stopped. While at work, a corresponding Parking Community is created or updated with vehicle IDs via neighbor discovery. Of course, additional Parking Communities are created based on driver habits, e.g., for locations visited regularly such as shopping malls and friends' houses.

3.2 Querying

When driving back home, the set ID_h of previously collected IDs for A_h is looked up from τ_h. A query for available parking spots is cryptographically signed with h's private key sk_h. An ephemeral symmetric key is generated randomly and asymmetrically encrypted with the respective public key decoded from each $id \in ID_h$ as depicted in Figure 1b. Conclusively, the query is sent via geocast into the home location A_h. The message contains (a) the symmetrically encrypted payload, and (b) the symmetric key encrypted for each vehicle in the corresponding community h, which comes with reasonable overhead compared to the overall message size which is dominated by the payload.

3.3 Responding

Each vehicle v with $id_v \in ID_h$ that is located in A_h (in Figure 1b this includes the vehicles with IDs id_1, id_2, id_4, while id_3 has not arrived yet) can decrypt the query and verify its source because v also collected the ID of the querying vehicle in Step 3.1, when the community was created or updated. By means of this authentication, incoming queries can also be prioritized, as is further described in Section 3.5. Receiving vehicles encrypt their responses using the source ID src of the message, which corresponds to the public key. The response consists of an estimate e:

$$e = \begin{cases} 1 & \text{if a space is available} \\ -1 & \text{if no space is available} \end{cases} \tag{4}$$

For the sake of simplicity, we do not further elaborate on how exactly vehicles come up with this estimate, but assume that each vehicle is able to use on-board sensor systems (e.g., ultra sonic, cameras) to determine which parking spots are available while driving through the home area A_h and while parking there, as was demonstrated in previous work [27]. Based on these data, as well as the time passed since the data was recorded, and other parameters, each vehicle estimates the likelihood of available parking spots in A_h that is finally mapped to a binary estimate e as shown in Equation 4. If no clear estimate is possible, we assume that the corresponding vehicle does not respond to the query at all in order to not provide potentially false data and to not risk deteriorating its rating (as described in Section 3.4).

3.4 Rating

The query originator finally receives the responses from an arbitrary number of community vehicles, depending on how many of them are located in the destination area and have chosen to respond with an estimate.

For each community vehicle v, the originator keeps a count of how many estimates e_v (see Section 3.3) turned out to be correct and incorrect, which we refer to as r_v and s_v, respectively. These values are used to calculate a reputation rating $Rep_v(r_v, s_v)$, based on the beta probability density function which can be used to represent probability distributions of binary events such as the estimation process $e_v \in \{-1; +1\}$ described in Section 3.3. The mathematical background of the beta function is analyzed in many text books on probability theory [29]. We therefore only present results based on the beta reputation system [30], which provides us with a mathematically sound and well-understood indication of how a particular vehicle is expected to behave in the future, that is in our case, to correctly or incorrectly announce a free parking spot. To this end, the probability expectation value $E(p)$ of the beta reputation function $\varphi(p|r, s)$ is a very suitable representation for this indicator, as argued by Jøsang et al. [30]. This gives us a reputation rating in the range $[0, 1]$ where the value 0.5 represents a neutral rating. Formally, the reputation rating $Rep_v(r_v, s_v)$ for vehicle v is thus defined as

$$\begin{aligned} Rep_v(r_v, s_v) &= E(\varphi(p|r_v, s_v)) \\ &= \tfrac{r_v+1}{r_v+s_v+2}, \end{aligned} \tag{5}$$

with φ being the beta reputation function [30]

$$\varphi(p|r, s) = \frac{\Gamma(r+s+2)}{\Gamma(r+1)\Gamma(s+1)} p^r (1-p)^s, \tag{6}$$

where $0 \le p \le 1, 0 \le r, 0 \le s$ and Γ being the gamma function.

After a timeout, the querying vehicle weighs all n received responses e_i with the corresponding vehicle i's reputation rating Rep_i to determine a consensus ω about the likelihood of a free parking spot in the destination area.

$$\omega = \frac{\Sigma_i^n \left(Rep_i(r_i, s_i) \cdot e_i\right)}{n} \tag{7}$$

If the outcome ω is below the threshold $\omega_{thresh} = 0$, the driver is advised to not rely on finding parking in his home area, but instead take the first free spot that he considers close enough, for example.

If the driver decides to drive to the home area (most likely if $\omega \ge \omega_{thresh}$), the vehicle scans the street for available spots itself and thus compares the actual situation with the received estimates, updating each r_v and s_v accordingly and providing feedback for the next calculation of the reputation rating.

3.5 Prioritization

Prioritization of incoming queries is done by responding vehicles solely based on their community information. Two different levels are possible: (a) member and (b) non-member prioritization.

(a) Receiving vehicles can prioritize incoming queries based on the reputation rating of the originator, who signed the query. The reputation rating thereby directly correlates to a priority level—reputable vehicles are thus more likely to receive a response than those with a lower reputation. Consequently, it is in the vehicle's own interest to obtain a high reputation rating, such that it will also be provided with inquired information. This incentivizes frequent and honest responses and discourages dishonest and uncertain estimates in a tit-for-tat manner.

(b) Vehicles receiving a query will typically favor community members over non-member requests and thus save resources, e.g., computing power. No reputation rating is available for non-members and thus the lowest priority level is assigned. Different advanced priority and resource management schemes can be considered to save energy or other resources, in particular while vehicles are parking. One option is a modification of the leaky bucket algorithm [31], for instance, with two buckets of, say, energy supply, one for members of a particular Parking Community and another for unknown requesters. Since this is not the focus of this paper, though, we do not elaborate on resource management.

3.6 Robustness

If vehicle density is sparse, there might not be sufficient vehicles in a destination area to get a response to a parking query. This is particularly true if the query is encrypted for the community and can thus only be responded to by community members, which excludes potential non-member communication partners. In a sparse network, this restriction could be relaxed such that queries are only signed by the originator, but not encrypted. Consequently, members as well as non-members are able to respond to the query, thus increasing the robustness of the protocol because a higher number of communication partners is available. Signing but not encrypting queries also allows vehicles to query for parking spots in irregularly or newly visited locations where they are not part of a community (and cannot predict which vehicles are currently located in that area).

Since an originator does not have a reputation rating Rep_i for non-members, though, their responses are only taken into consideration in our protocol if the originator does not receive any responses from members, lest Sybil attacks become possible. Existing communities are not influenced and thus not put at risk by non-member responses.

From a receiving vehicle point of view, members and non-members will prioritize queries differently as explained in Section 3.5, but in either case the responses can be encrypted using the public key of the originator (which can be obtained as explained in Section 2.4), thus providing confidentiality of the parking availability data.

4 ATTACK SCENARIOS

In this section we first introduce the main security challenges for creating Parking Communities based on trust establishment and then analyze common attack scenarios.

Our scheme should work as an overlay on existing vehicular network protocols and without reliance on a central TTP. When a consensus for free parking spots is established, the scheme needs to account for impersonation and Sybil attacks to prevent impersonated answers and forged identities to reach a majority. Already generated key pairs used in the underlying network protocol can directly be utilized as unique identifiers. This prevents impersonation attacks, as it is not feasible to generate a private key, e.g., for signing messages, to a given public key, i.e., a given ID. In the case of ECC, public keys are short and can easily be encoded as identifiers (cf. Section 5.3). Sybil attacks, however, are harder to account for when establishing a consensus without a TTPs. We therefore propose a Trust On First Use (TOFU) model to verify the existence of an actual vehicle for each identity used for answering parking spot queries through physical encounters [28].

Our attack model is as follows: As little information as possible should be transmitted in the open, protecting the driver's anonymity against passive adversaries. Collecting physically encountered vehicle IDs makes it difficult to perform global Sybil attacks. Considering active attackers, capable of executing Man-in-the-Middle (MitM) and constrained targeted Sybil attacks, access to resources must be regulated. It should be prevented that information about vacant parking spaces is intercepted by a third party along the communication path. Conversely, vehicles (especially while parking) must be able to prioritize incoming queries in order to prevent Denial of Service (DoS) attacks, where malicious vehicles deplete resources by generating queries with multiple fake IDs (Sybil attack). Attacks and their mitigations are further discussed in the following subsections.

4.1 Impersonation and Sybil Attacks

In all scenarios, our key management prevents impersonation attacks, where a vehicle impersonates another vehicle by adopting its ID during an ongoing communication. Because we require all messages to be signed, a message's signature always corresponds to the public key pk_s encoded in the message's src. An attacker would need to generate sk_s corresponding to an existing pk_s. This requires to randomly generate key pairs until a collision with the existing public key is found. In case of an ECC based protocol, the success probability is 2^{256} and the attack is thus considered infeasible. This is true if the difficulty of ECDLP holds and ECDSA as well as its implementation has no critical flaws (e.g., insufficient entropy). When a Parking Community is created, context information such as the origin of a communication signal [28] allows a collecting vehicle to differentiate between physical vehicles. Thus, an attacker needs to be physically present when the victim is parking and is constrained in how many vehicles can be forged for a Sybil attack due to the difficulty of forging communication signals originating from different locations.

4.2 Interception of Parking Spot Availability

In Parking Communities, vehicles cooperate in order to gain an informational advantage. The information of available resources, namely 'parking spots', is to be protected against passive adversaries as it could be used for reaching available spaces earlier than the original requester, without being part of the community. By encrypting query responses (confidentiality), intercepted information is of no value for eavesdropping adversaries.

4.3 Denial of Service

An attacker could try to exhaust available resource of a parking vehicle by querying many times for available parking spots. While the main purpose of the proposed Parking Communities is to provide a way to reach a consensus regarding specific parking locations, we introduced the idea of limiting computing resources for incoming queries. As described in Section 3.5 b), vehicles can decide to only answer queries originating from reputable members of their own Parking Community. This works as a self-protecting feature in case of a Denial of Service attack.

4.4 Location Tracking

Existing privacy threats have been thoroughly investigated before [32], as have challenge-response protocols been proposed to prevent the exposure of context information. Global passive adversaries, on the one hand, can always track vehicles using RSUs, independent of whether IDs are changed regularly or not. Simply because of wireless emissions originating from vehicles, transmitted messages can be tracked from source to destination. It has been shown that such an attacker can correlate beacon messages to specific vehicles with a probability of nearly $100\,\%$ [33]. On the other hand, local adversaries that physically follow a tracked vehicle cannot be protected against via any digital privacy mechanism either.

Yet, there is a wide spectrum in between these two extreme cases of attackers. Therefore, pseudonym certificates, e.g. [1], are deployed to cover the identity of vehicles. In addition to changing pseudonyms regularly, Sampigethaya et al. [34] have shown that a silent period between pseudonym changes is necessary. However, the concept of distributed communities requires vehicles to be uniquely identifiable by their peers.

We therefore propose using a Key Derivation Function (KDF) allowing vehicles to change pseudonyms regularly but in a deterministic and reproducible way for members of the Parking Community (and only for them). During neighbor discovery (see Section 3.1), a common secret is shared besides the ID. This secret as well as the last valid pseudonym ID are input parameters to the KDF, which computes a new ID. This is done by both the vehicle changing its pseudonym and by all community members that have collected its ID and secret. Generally, each vehicle starts with a dedicated pseudonym per area, which is also only used for communication with the community. For other purposes, such as safety messages (e.g., CAM/DENM [1]), other pseudonyms according to the underlying security architecture are used and changed frequently [32]. The dedicated pseudonym per community area is typically only used once per day (e.g., when driving home), and can thusly be changed in intervals of 1 day using the KDF as described above. Consequently, Parking Communities also provide a means for anonymity and location privacy.

4.5 Accountability

Independent from using PKI or IBC as the underlying key management, we assume that a central trusted authority provides a means to unambiguously verify a vehicle's public key.

5 IMPLEMENTATION

As described above, Parking Communities can be implemented on top of existing networking stacks, thus benefiting from standardization and security efforts already in place. To show the feasibility of our approach, we have implemented a prototype for the underlying security architecture by extending IBR-DTN[1], a high-performance [35] Bundle Protocol [5] implementation in C++, to provide integration of ECDSA and ECIES, key management for ECC keys, encoding public keys as IDs, and our trust rating model. Since Delay-Tolerant Networking (DTN) is an overlay network, we can transparently exchange the underlying networking stack, such as TCP/IP, IEEE 802.15.4, or IEEE 802.11p and its higher layer standard IEEE 1609. In DTN terminology, an ID is called Endpoint Identifier (EID), and messages are called bundles. This section describes the implementation details and cryptographic algorithms used for the Parking Community prototype.

5.1 Crypto Libraries

IBR-DTN uses OpenSSL[2], which provides support for ECDSA, but no ECC encryption schemes, e.g., ECIES, out of the box. Furthermore, OpenSSL's ECDSA implementation has been attacked via a side-channel [36]. Matured cryptographic libraries are Botan[3] and Crypto++[4]. Crypto++ has a long development history and is thus available on almost all Unix-like systems and Windows. While Botan only provides ECDSA, Crypto++ provides a wide range of functionality, among others the ECC-based algorithms ECDSA, ECNR, ECIES, ECDH, and ECMQV. For using recently proposed curves like Curve25519 [37], its authors provide a library called NaCl[5]. However, as described in Section 5.2.1, an integration of ECC into the Bundle Security Protocol requires an asynchronous ECC encryption scheme and access to underlying cryptographic primitives. NaCl only provides synchronous DH key agreement and high-level access. Conclusively, we chose Crypto++ for our implementation.

The DTN daemon has been configured to reject bundles not cryptographically signed and has been extended to support and manage communities via an API.

5.2 Encryption and Signature Algorithm

This section introduces our extensions to the Bundle Security Protocol and discusses the security background of the used algorithms.

5.2.1 Extending the Bundle Security Protocol

The *Bundle Security Protocol Specification* (RFC 6257) [18] defines RSA-based cipher suites in conjunction with the AES block-cipher using Galois/Counter Mode (GCM) for fast symmetric encryption of payload. Since modern ECC implementations are much faster than RSA implementations [7] and allow for shorter but equally secure key lengths[6], we use ECC. We chose the widely used signature scheme ECDSA and the encryption scheme ECIES for Payload Integrity Blocks (PIBs) and Payload Confidentiality Blocks (PCBs), respectively. In traditional public key cryptosystems, the cryptographic principle of key separation is applied, i.e., generating different key pairs for signing and encrypting [38]. This was mainly motivated by the properties of the RSA trapdoor function. Degabriele et al. [39], however, have proven that ECDSA and ECIES can be securely combined using the same key pair. Breaking the key separation principle

---

[1] http://www.ibr.cs.tu-bs.de/projects/ibr-dtn

[2] http://www.openssl.org

[3] http://botan.randombit.net

[4] http://www.cryptopp.com

[5] http://nacl.cr.yp.to

[6] http://www.keylength.com

allows us to generate one key pair only. Thus, only one public key needs to be encoded as an EID, resulting in short EIDs.

5.2.2 Elliptic Curve Cryptography

We chose the curve 'secp256k1' [40], since it has a sufficiently long security history and is provided by nearly all cryptographic libraries available. It is also used in conjunction with ECDSA to sign Bitcoin transactions [41]. Bitcoin has undergone a comprehensive five-year analysis since its beginning and has shown no major weaknesses. In contrast to curves like NIST's P-256, 'secp256k1' is not based on hashing unexplained seeds and is thus considered "somewhat rigid" [42].

In recent years, there have been advances in cryptanalysis of curves based on non-prime fields, e.g., $\mathbb{F}_{2^n}$, while the "overall security picture [has been] unchanged for prime-field ECC" [37, 43]. 'secp256k1' is a generalization of the Koblitz curve but associated to a prime field $\mathbb{F}_p$ with $p = 2^{256} - 2^{32} - 977$. It has two known primary weaknesses: Due to its structure, it has an efficiently computable endomorphism, which also leads to speed ups in Pollard's rho algorithm [44]. The other weakness is its twist security [45]. Conversely, carefully implemented, problems due to twist security can be avoided. Besides those weaknesses, 'secp256k1' is mathematically sound and it has shown no major drawbacks in the past [42].

5.3 Key Management

In DTNs, nodes are identified by an EID, which is formed by a Uniform Resource Identifier (URI) [46], whereas the precise structure leaves room for adapting it for specific network structures. URIs offer a variable length and a standardized syntax, which can also be used to define groups of related nodes.

In Parking Communities, each vehicle v has a set of IDs, or EIDs, i.e., $EID_v \subset \mathcal{EID}$, with $\mathcal{EID}$ being the set of all valid endpoint identifiers. Each community's $eid_c \in EID_v$ is derived from its public key pk according to the following form:

$$eid_c := \text{'sec://'} \parallel base64url(pk) \tag{8}$$

Here, $base64url()$ corresponds to URL-safe Base64 encoding [47]. We introduced a new URI scheme 'sec' to indicate that the following Scheme-Specific Part (SSP) consists of the encoded public key instead of the typical node part and optional client/application specific parts. In our scheme, the SSP consists at minimum of the bytes consumed by the encoded public key. An ECC public key is $32\,\text{B}$ long. Base64 uses 4 characters to represent $3\,\text{B}$, always resulting in a multiple of 4; thus the length of n bytes encoded in Base64 is defined by

$$len_{ssp}(n) = \left\lceil \frac{n}{3} \right\rceil \cdot 4 \tag{9}$$

Conclusively, the SSP consumes $44\,\text{B}$ without the application/client specific part. This is well below the maximum length of $1023\,\text{B}$ as defined by RFC 4648 [47].

6 DISCUSSION

In this section, we provide a comparison of key and trust management schemes from the literature. Parking Communities can be implemented on top of different key management approaches, thus the following description can be used as a guideline for choosing the most appropriate architecture per use case. Moreover, existing trust management approaches are compared to Parking Communities. In particular, traditional certificate-based PKI, IBC, and incentive-based schemes are elaborated on. In the following subsections we compare selected aspects of these architectures and summarize the results in Table 1.

PKI-based and IBC architectures have been introduced in Section 2.4. IBC schemes are subdivided into flat and hierarchical ones. Hierarchical Identity-Based Cryptography (HIBC) schemes are organized by tree-based hierarchy structures to distribute trust among intermediate authorities, e.g., affiliated to geographical regions for example [24], instead of having one central point of failure.

Incentive schemes, designed to protect against selfish behavior, are classified into barter-based, credit-based and reputation-based schemes [48]. As credit- and reputation-based schemes often engage with each other (e.g., [49]), they are treated as one category. However, a subdivision between schemes requiring a TTP acting as a virtual bank and self-organizing ones has been investigated. These schemes introduce credits, similar to virtual currencies, traded between nodes to pay for forwarding/routing of bundles. Reputation-based schemes are similar, while also providing protection against adversaries with high computational power.

6.1 Trusted Third Parties

Most schemes' authentication is based on one or more centralized TTPs. They are required for the initial authentication of new nodes and bootstrapping of trust. Traditional PKIs are organized hierarchically but without any restrictions with regard to which identities they are allowed to issue certificates. Thus, one compromised intermediate authority can compromise the whole network. Additionally, message exchange requires retrieval of public keys from TTPs before encryption/verification is possible. In IBCs, derivation of public keys from IDs allows encryption/verification without retrieving keys from TTPs in advance [20]. PKI certificates are issued using Certificate Signing Requests (CSRs), whereas key pairs were generated solely by the node itself; IBC schemes issue IDs by generating and storing key pairs. Thus, compromising an IBC infrastructure has much broader consequences to a network. Credit-based schemes require TTPs for reputation dissemination or a credit clearance process. Wei et al. distribute this task to a self-organizing network, leaving only the initial bootstrapping of nodes to an offline TTP [52].

Table 1: Comparison of Key and Trust Management Approaches

		Key Management			Credit/Reputation	
Property	Parking Com.	PKI[a]	IBC[b]	HIBC[c]	Bank[d]	SO[e]
No TTP Required	✓	✗	✗	✗	✗	✓(setup)
Revocation/Expiry	✓	✓	✓(expiry)	✓(expiry)	–	–
Anonymity	–[g]	✓/✗[f]	✗	✓(limited)	✗	✗
Confidentiality	✓/✗	✓	✓	✓	–	–
Integrity and Authenticity	✓	✓	✓	✓	✓	✓
Forward Secrecy	–[g]	✓	✓(limited)	✓(limited)	–	–
No Physical Encounters Required	✗	✓	✓	✓	✓	✓
Required Network Connectivity	sparse	high	medium	medium	medium	sparse
Protocol Complexity	medium	low	low	low	medium	high
No Single Point of Failure	✓	✗	✗	✓	✗	✓
Protects against Impersonation	✓	✓	✓	✓	–	–
Protects against Sybil Attacks	✓/✗	✓	✓	✓	–	–
Protects against Selfish Behavior	✓	✗	✗	✗	✓	✓

[a] PKI schemes with traditional (X.509) or pseudonym certificates [1]
[b] IBC schemes: [20]
[c] HIBC schemes: [19, 21, 22, 24]
[d] Credit schemes, virtual bank: [50, 49, 51]
[e] Credit schemes, self organizing: [52]
[f] ✓(limited): pseudonym certificates [1]; ✗: X.509 certificates
[g] Depending on underlying key management

✓/✗	Only true for specific scenarios/proposed protocols
–	Not part of this scheme's objectives

6.2 Revocation

Revocation of certificates is typically achieved by distributing revocation lists, which can cause a significant overhead and poses a problem in sparse and intermittent networks. IBC schemes propose to encode an expiry date into the IDs themselves. While no direct revocation is possible, using short expiry dates, nodes are required to renew their ID regularly by contacting the IBC TTP over a secure channel. In Parking Communities, revocation of a public key is achieved by its owner digitally signing a revocation message and distributing it in the community, ensuring that nobody but the possessor of the private key can inject such a message.

6.3 Anonymity

Anonymity as a property is difficult to measure in real-world applications. To complicate data aggregation by attackers with limited capabilities, such as malicious vehicles recording metadata of forwarded bundles, pseudonyms are required. In vehicular protocols, such as proposed by the Car 2 Car Communication Consortium (C2C-CC), vehicles are issued a limited amount of pseudonym certificates by a central TTP. Vehicles iterate over this set until it has been depleted allowing a certain degree of pseudonymity [1]. As we have shown in Section 5, Parking Communities can be implemented on top of different networking stacks, including recent C2C-CC standards. Therefore, its underlying certificate infrastructure can be used to allow for a certain level of pseudonymity. As defined in our attack model in Section 4.4, Parking Communities require vehicles to recognize their peers for which we have provided a secure KDF-based solution. Consequently, the same level of anonymity (and location privacy) as in the underlying technology is achieved.

6.4 Trust Management

To establish trust, Parking Communities introduce trust anchors based on physical encounters to distinguish surrounding vehicles [28], preventing certain attacks as described in Section 4. In typical PKI or IBC schemes, central entities decide which nodes can be trusted, in case of PKI by providing lookup services. While IBC already provides an advantage over the traditional PKI system, as no public key lookup needs to be performed before transmissions, it still requires connectivity in regular intervals to extend the validity of IDs, though. A significant advantage of Parking Communities is that they require only sparse network connectivity because no lookup or renewal using central services is required.

6.5 Summary

Table 1 summarizes the aspects of the examined key and trust management schemes most relevant to vehicular networks. Some of these aspects have been discussed in the previous sections.

Similar to self-organizing credit-based schemes, our scheme does not require a security infrastructure to retrieve trust ratings. However, existing key management solutions, such as PKI or IBC, can be used to establish accountability. HIBCs improve over IBCs by hierarchical organization, but still leave a single root TTP. This is suitable for military scenarios, but has been proven ineffective against global, active adversaries.

While public key protocols with TTPs provide perfect protection against impersonation and Sybil attacks, our scheme additionally offers protection against impersonation attacks despite its distributed design. By means of the proposed trust anchor concept, it is also able to mitigate Sybil attacks, as discussed in Section 4.

We argue that the advantages of IBC in comparison to traditional PKI are minimal because both infrastructures need to somehow authenticate nodes on deployment. This is a major challenge, as a secure key-identity binding is crucial for any authenticated scenario. Establishing key-identity bindings with IBC leaves the key-escrow problem unsolved. Incentive schemes introduce high protocol complexity and more infrastructure [50] to allow distributed agreements in disruptive networks. Similar to Parking Communities, they allow prioritization based on incentives like virtual currencies or reputation and thus protect against selfish behavior.

Conclusively, this comparison illustrates the difficulty of balancing the trade-offs between centralized and decentralized key and trust management schemes. Parking Communities are a lightweight approach that integrates aspects from the wide range of existing architectures creating a novel approach for highly decentralized scenarios.

7 SIMULATION

We use The ONE [53] to simulate Parking Communities in a working day scenario [26] in the city of Helsinki, Finland. The model presents the everyday life of people going to work in the morning, spending their day at work, and commute back home at night. Our goal is to evaluate the development of reputation ratings over time and to show the general feasibility of our approach, i.e., if a car encounters sufficient other cars in order to create a sufficiently large community to get replies to its queries.

7.1 Setup

In the Working Day Movement model, over 1000 nodes move on a map of the Helsinki area with the size of roughly 7000 x $8500\,\mathrm{m}^2$. The nodes and their home zones are assigned to 4 main and 3 overlapping artificial districts, as depicted in Figure 2 and further described by Ekman et al. [26]. Each node has its own home zone, which typically overlaps with other zones depending on the node density per district. $25\,\%$ of all nodes are either malicious nodes or benign nodes with potentially false sensor information, i.e., they may report false positives. For the sake of readability, we subsume both groups under the term malicious nodes because it is irrelevant why false information is reported.

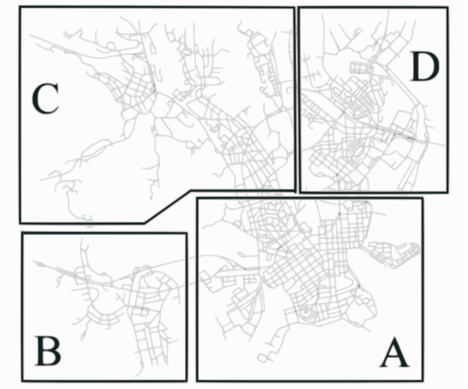


Figure 2: Map of Helsinki with artificial districts [26]

In contrast to the original movement model, we assume that all nodes are regular vehicles, instead of also including busses and taxis. We used a warmup period of a full day (as opposed to half a day), due to the periodic nature of the proposed protocol as well as of the mobility model. We set the transmit range of all nodes to $100\,\mathrm{m}$ and the home zone radii to $300\,\mathrm{m}$. Hence, vehicles always park within a radius of $300\,\mathrm{m}$ to their home zone center, with a random offset, and create a community by collecting vehicle IDs in their communication range. Every morning, each vehicle leaves for work at a specified time, and stays there for $8\,\mathrm{h}$, before it either commutes back home or follows an evening activity first. Halfway home, though, each vehicle geocasts a query into the home zone according to Section 3. It then waits for responses from its community members. In our simulations, the probability of a free spot in the home zone (the *ground truth*) is 0.5. Honest nodes receiving the query always respond with the ground truth, while malicious nodes lie with a probability of $\psi = 0.5$, i.e., respond with the opposite of the ground truth. The querying vehicles then receive the responses and calculate a weighted consensus ω. In the home zone, they compare the responses with the ground truth and update the reputation ratings accordingly.

The simulation runs for $700\,000\,\mathrm{s}$, which corresponds to 8 full days. We repeat the simulation 10 times.

7.2 Results

Figure 3 shows the number of members per Parking Community per simulation day, averaged over all 10 simulations runs. It is observable that after 5 days $50\,\%$

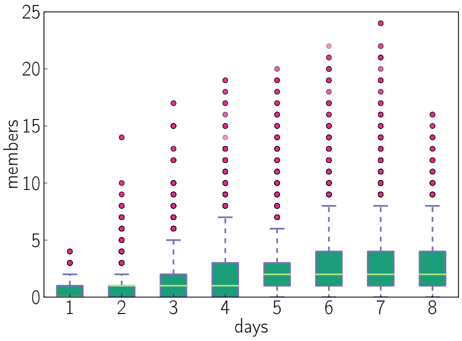


Figure 3: Parking Community sizes

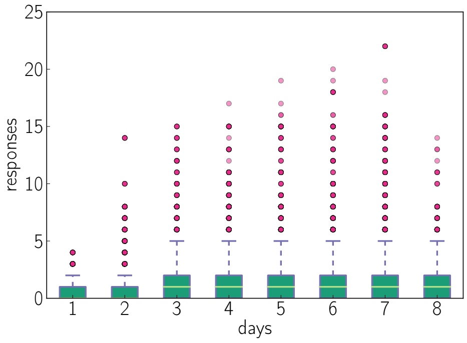


Figure 4: Number of responses received per day

of all communities have at least 2 to 4 members, with another 25 % having between 4 and 20 members. These values further increase during the following days, as vehicles park at random locations in their home zones, thus meeting new vehicles. For the simulated scenario, the community sizes basically stabilize around days 6 and 7. In sum, at least 75 % of all vehicles have between 3 and more than 20 vehicle IDs collected after a few days. Due to the specific geography of Helsinki, with some remote and isolated areas (e.g,. on islands only connected by a bridge to the mainland), some vehicles can only create very small communities, while vehicles in densely populated areas, such as District A in Figure 2, have quite large communities after a short period of time.

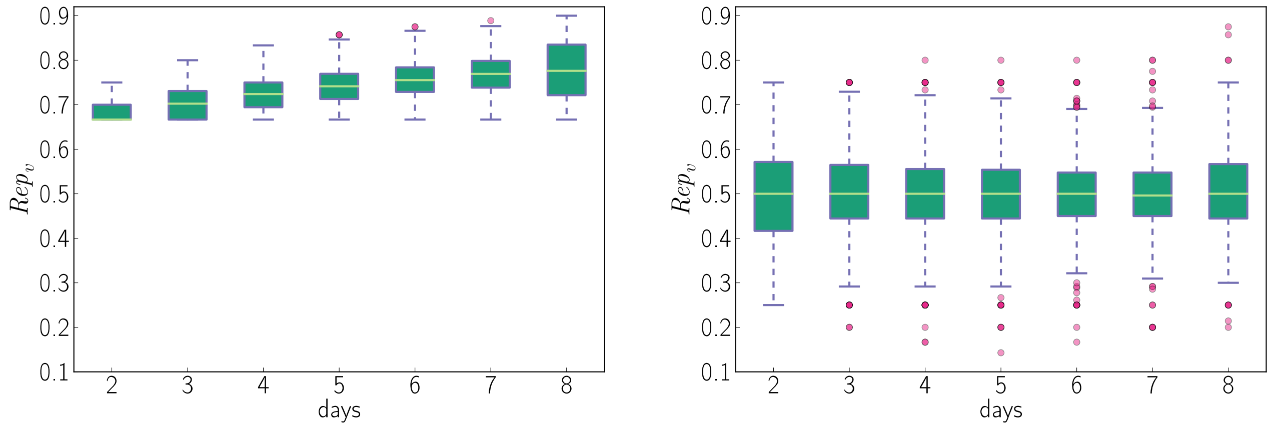
Figure 4 now correlates the community sizes with the number of successful query/response exchanges. It can be observed that from day 3 on, vehicles receive 2 responses on average. Remarkably, 25 % of vehicles received significantly more responses, up to 15. The maximum number of responses further increases to up to 23 which is almost the maximum Parking Community size. In this particular case, this indicates that the querying vehicle was (a) part of a large community, and (b) was returning home as one of the latest out of his peers, such that almost every other node was already located in the home zone and thus able to respond to the query. As described above, vehicles in densely populated areas (and thus with a large community size) have a significant advantage over remote areas. In downtown areas these vehicles receive sufficiently many responses to make a meaningful contribution to the parking search.

We further evaluate how reputation ratings develop over time, in particular by comparing honest and malicious nodes in Figure 5. As we have a decentralized model, in which no single entity is in charge of keeping track of a vehicle's reputation rating, but each community member establishes its own rating per peer, we average the reputation rating for each vehicle over all other nodes that have it in their respective communities.

All nodes start with a reputation value of 0.5, which represents a neutral rating. As the reputation $Rep(r, s)$ depends on the physical verification of received responses, Figure 5 omits the simulated day 1, since only after the vehicles parks in the home zone, the respective values r, s can be updated, while the reputation is already updated halfway home when a consensus ω is calculated. As can be seen in Figure 5a, honest nodes' reputation continually increases over the simulation time, but has already reached an average of 0.7 on day 2. A peculiar observation is that on day 8, the box (i.e., the interquartile range) is larger than on the previous days, indicating a larger variance. This is because some vehicles have not yet reached their home area before the simulation ends, which does not affect the general validity of the observations. In comparison, Figure 5b shows the reputation ratings for malicious nodes. At first sight, it may seem curious that malicious nodes' reputation remains at 0.5 on average, with some outliers being at par with honest nodes' reputation. However, this is clearly expected as we have modeled the behavior of malicious nodes to arbitrarily lie or tell the truth. Hence, vehicles cannot identify and downrate malicious nodes, but have to remain neutral, which is reflected in the simulation results. Yet, as we have shown above, honest vehicles are uprated quite quickly in comparison, such that a weighted consensus ω is nevertheless a meaningful criterion. To provide further evidence, though, Figure 6 shows the reputation ratings for malicious nodes with $\psi = 0.85$, instead of $\psi = 0.5$ (while keeping constant all other parameters). It can be clearly observed that malicious vehicles can clearly be identified and are downrated significantly (and continually) from day 2 on. On day 7, for instance, the average rating is 0.3, with 75 % of all (malicious) nodes having a lower rating than 0.35.

Finally, we evaluate how often vehicles make the right decision about relying on available parking spots in their home area, as described in Section 3.4. A decision is correct, if (a) a spot is free and $\omega \geq \omega_{thresh} = 0$ or (b) no spot is available and $\omega < \omega_{thresh} = 0$.

Figure 7 shows the relative frequency of correct decisions per simulated day for different probabilities ψ of lying. As expected, the rate of correct decisions increases



(a) Reputation ratings for honest nodes per day

(b) Reputation ratings for malicious nodes per day

Figure 5: Development of reputation ratings averaged over nodes and 10 simulations runs

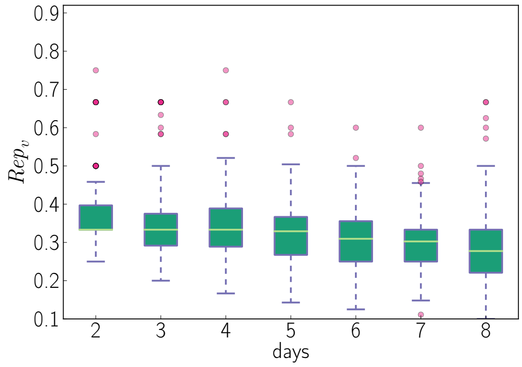


Figure 6: Reputation for malicious nodes, $\psi = 0.85$

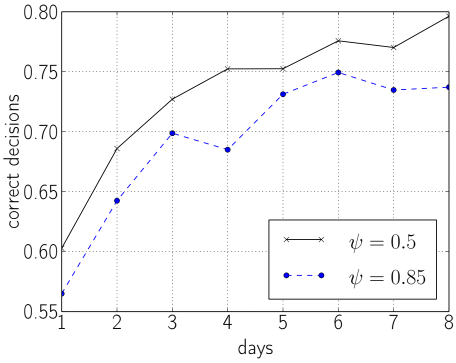


Figure 7: Rate of correct decisions over time

over time because the reliability of reputation ratings increases as well. For $\psi = 0.5$, the correct decision rate is already higher than 0.75 after day 4 and keeps rising. It takes longer to reach the same values for $\psi = 0.85$ as the system has to cope with liars that are more chronic. In sum, though, good values are achieved after only a few days (remember that, in our simulations, the system is used once per day when driving home), showing the feasibility of the approach.

8 CONCLUSION

In this paper, Parking Communities have been presented. They provide a novel trust management for vehicular parking applications without reliance on a central TTP for retrieving trust ratings. For this purpose, vehicles create communities, trusted groups helping their members to find parking in their respective community area. Trust anchors enable signed and encrypted request-response communication in disrupted environments. As our approach can be used as an overlay to existing vehicular networking technologies, it can directly benefit from established security mechanisms, e.g., pseudonym certificates. Our approach is based on high-performance state-of-the-art encryption and signature algorithms, in particular ECC, as well as a well-understood mathematical trust rating model. Attack scenarios and their mitigations are discussed. Without requiring a TTP, our scheme provides protection against impersonation and Sybil attacks utilizing trust anchors and physical verification. The underlying security architecture of Parking Communities has been implemented in the open-source IBR-DTN, which is publicly available. We provide a comprehensive comparison with existing key and trust management schemes for vehicular networks, as well as simulations showing the concept's feasibility.

8.1 Future Work

We plan to design fine-grained access control mechanisms to improve resource management and prioritization of incoming queries, e.g., based on energy/response budgets or additional properties verifiable by trusted third parties, such as certificates of disability. In order to further increase the frequency of correct decisions, vehicles with high mutual trust could exchange and merge their sets of communities. The expected results are an increase in the size and number of communities as well as more robust reputation ratings.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement Number [269916, V-Charge].

REFERENCES

[1] Car 2 Car Commun. Consortium. *Manifesto: Overview of the C2C-CC system, V1. 1*. Tech. rep. Aug. 2007.

[2] Federal Highway Administration. *Advanced Parking Management Systems: A Cross-cutting Study: Taking the Stress Out of Parking*. Intelligent Transportation Systems, U.S. Department of Transportation, 2007.

[3] *Wireless Access in Vehicular Environments (WAVE)–Security Services for App. and Manag. Messages (IEEE Std 1609.2-2006)*. IEEE. July 2006, pp. 1–105.

[4] S. Schildt, J. Morgenroth, W.-B. Pöttner, and L. Wolf. "IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation". In: *Electronic Communications of the EASST* 37 (2011).

[5] K. Scott and S. Burleigh. *Bundle Protocol Specification*. RFC 5050. IETF, Nov. 2007.

[6] N. Koblitz. "Elliptic Curve Cryptosystems". In: *Mathematics of computation* 48.177 (1987), pp. 203–209.

[7] C. Paar and J. Pelzl. *Understanding cryptography: a textbook for students and practitioners*. Springer, 2010.

[8] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang. "High-speed high-security signatures". In: *Journal of Cryptographic Engineering* 2.2 (2012), pp. 77–89. URL: http://cr.yp.to/papers.html#ed25519.

[9] W. Diffie and M. E. Hellman. "New Directions in Cryptography". In: *IEEE Trans. on Information Theory* 22.6 (1976), pp. 644–654.

[10] J. Zhang. "A Survey on Trust Management for VANETs". In: *Adv. Information Networking and App.* Tiruchengode, India: IEEE, Mar. 2011, pp. 105–112.

[11] P. Wex, J. Breuer, A. Held, T. Leinmuller, and L. Delgrossi. "Trust Issues for Vehicular Ad Hoc Networks". In: *VTC Spring*. Singapore: IEEE, May 2008, pp. 2800–2804.

[12] M. Gerlach. "Trust for Vehicular Applications". In: *Symp. Autonomous Decentralized Systems (ISADS'07)*. Sedona, AZ: IEEE, Mar. 2007, pp. 295–304.

[13] M. Raya, P. Papadimitratos, V. D. Gligor, and J.-P. Hubaux. "On Data-Centric Trust Establishment in Ephemeral Ad Hoc Networks". In: *INFOCOM*. Phoenix, AZ: IEEE, Apr. 2008, pp. 1238–1246.

[14] A. Vinel, C. Campolo, J. Petit, and Y. Koucheryavy. "Trustworthy Broadcasting in IEEE 802.11p/WAVE Vehicular Networks: Delay Analysis". In: *IEEE Communications Letters* 15.9 (2011), pp. 1010–1012.

[15] A. Patwardhan, A. Joshi, T. Finin, and Y. Yesha. "A Data Intensive Reputation Management Scheme for Vehicular Ad Hoc Networks". In: *Mobile and Ubiq. Sys. Workshops*. San Jose, CA: IEEE, July 2006, pp. 1–8.

[16] S. Park, B. Aslam, and C. C. Zou. "Long-term reputation system for vehicular networking based on vehicle's daily commute routine". In: *Consumer Commun. and Netw. Conf. (CCNC)*. IEEE. 2011, pp. 436–441.

[17] A. Studer, E. Shi, F. Bai, and A. Perrig. "TACKing Together Efficient Authentication, Revocation, and Privacy in VANETs". In: *Sensor, Mesh, Ad Hoc Commun.Netw.* Rome, Italy: IEEE, June 2009, pp. 1–9.

[18] S. Symington, S. Farrell, H. Weiss, and P. Lovell. *Bundle Security Protocol Specification*. RFC 6257. IETF, May 2011.

[19] A. Seth and S. Keshav. "Practical security for disconnected nodes". In: *Secure Network Protocols (NPSEC'05)*. Boston: IEEE, 2005, pp. 31–36.

[20] N. Asokan, K. Kostiainen, P. Ginzboorg, J. Ott, and C. Luo. "Applicability of identity-based cryptography for disruption-tolerant networking". In: *Workshop Mobile Opport. Netw. (MobiOpp '07)*. San Juan, Puerto Rico: ACM, 2007, pp. 52–56.

[21] A. Kate, G. Zaverucha, and U. Hengartner. "Anonymity and security in delay tolerant networks". In: *Security and Privacy in Comm. Netw. (SecureComm 2007)*. 2007, pp. 504–513.

[22] R. Patra, S. Surana, and S. Nedevschi. "Hierarchical identity based cryptography for end-to-end security in DTNs". In: *Intell. Computer Comm. and Processing (ICCP 2008)*. 2008, pp. 223–230.

[23] W. L. Van Besien. "Dynamic, non-interactive key management for the bundle protocol". In: *Workshop on Challenged Networks (CHANTS '10)*. Chicago, IL: ACM, 2010, pp. 75–78.

[24] M.-R. Fida, M. Ali, A. Adnan, and A. Arsalaan. "Region-Based Security Architecture for DTN". In: *Infor. Techn.: New Gen. (ITNG)*. 2011, pp. 387–392.

[25] D. Boneh and M. Franklin. "Identity-based encryption from the Weil pairing". In: *Advances in Cryptology*. Springer. 2001, pp. 213–229.

[26] F. Ekman, A. Keränen, J. Karvo, and J. Ott. "Working Day Movement Model". In: *MobilityModels*. Hong Kong, China: ACM, 2008, pp. 33–40.

[27] P. Furgale, U. Schwesinger, M. Rufli, W. Derendarz, H. Grimmett, P. Mühlfellner, S. Wonneberger, J. Timpner, et al. "Toward Automated Driving in Cities using Close-to-Market Sensors: An Overview of the V-Charge Project". In: *Intelligent Vehicle Symposium (IV '13)*. Gold Coast, AU: IEEE, June 2013, pp. 809–816.

[28] M. Fiore, C. Ettore Casetti, C. Chiasserini, and P. Papadimitratos. "Discovery and Verification of Neighbor Positions in Mobile Ad Hoc Networks". In: *IEEE Trans. on Mobile Computing* 12.2 (2013), pp. 289–303.

[29] G. Casella and R. L. Berger. *Statistical Inference*. Duxbury advanced series. Duxbury Press, 1990.

[30] A. Jøsang and R. Ismail. "The Beta Reputation System". In: *15th Bled Electronic Commerce Conf.* 2002.

[31] J. Turner. “New directions in communications(or which way to the information age?)” In: *IEEE communications Magazine* 24.10 (1986), pp. 8–15.

[32] J. Freudiger, M. Jadliwala, J.-P. Hubaux, V. Niemi, and P. Ginzboorg. “Privacy of Community Pseudonyms in Wireless Peer-to-Peer Networks”. English. In: *Mobile Networks and Applications* 18.3 (2013), pp. 413–428.

[33] B. Wiedersheim, Z. Ma, F. Kargl, and P. Papadimitratos. “Privacy in inter-vehicular networks: Why simple pseudonym change is not enough”. In: *Wireless On-demand Network Systems and Services (WONS)*. 2010, pp. 176–183.

[34] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, and K. Sezaki. *CARAVAN: Providing location privacy for VANET*. Tech. rep. DTIC, 2005.

[35] W.-B. Pöttner, J. Morgenroth, S. Schildt, and L. C. Wolf. “An Empirical Performance Comparison of DTN Bundle Protocol Implementations”. In: *Workshop on Challenged Networks (CHANTS’11)*. Las Vegas, NV, Sept. 2011.

[36] D. Brumley and D. Boneh. “Remote timing attacks are practical”. In: *Computer Netw.* 48.5 (2005), pp. 701–716.

[37] D. J. Bernstein. “Curve25519: new Diffie-Hellman speed records”. In: *Public Key Cryptography (PKC 2006)*. Springer, 2006, pp. 207–228.

[38] S. Haber and B. Pinkas. “Securely Combining Public-key Cryptosystems”. In: *Computer & Communications Security*. Philadelphia, PA: ACM, 2001, pp. 215–224.

[39] J. P. Degabriele, A. Lehmann, K. G. Paterson, N. P. Smart, and M. Strefler. *On the Joint Security of Encryption and Signature in EMV*. Cryptology ePrint Archive, Report 2011/615. 2011.

[40] S. SEC. “2: Recommended elliptic curve domain parameters”. In: (2000). URL: http://www.secg.org.

[41] S. Nakamoto. “Bitcoin: A peer-to-peer electronic cash system”. In: *Consulted* 1 (2008), p. 2012.

[42] D. J. Bernstein and T. Lange. *SafeCurves: choosing safe curves for elliptic-curve cryptography*. [Online; accessed 2013-11-14]. URL: http://safecurves.cr.yp.to.

[43] D. J. Bernstein and T. Lange. *Security dangers of the NIST curves*. [Pres]. May 2013. URL: http://www.hyperelliptic.org/tanja/vortraege/20130531.pdf.

[44] J. M. Pollard. “A Monte Carlo method for factorization”. In: *BIT Numerical Mathematics* 15.3 (1975), pp. 331–334.

[45] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow. *Elliptic Curve Cryptography in Practice*. Cryptology ePrint Archive, Report 2013/734. 2013.

[46] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. *Delay-Tolerant Networking Architecture*. RFC 4838. IETF, Apr. 2007.

[47] S. Josefsson. *The Base16, Base32, and Base64 Data Encodings*. RFC 4648. IETF, Oct. 2006.

[48] J. Miao, O. Hasan, S. Mokhtar, L. Brunie, and K. Yim. “An Analysis of Strategies for Preventing Selfish Behavior in Mobile Delay Tolerant Networks”. In: *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*. 2012, pp. 208–215.

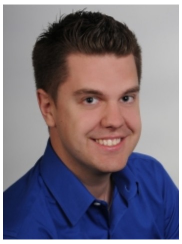
[49] R. Lu, X. Lin, H. Zhu, X. Shen, and B. Preiss. “Pi: A practical incentive protocol for delay tolerant networks”. In: *IEEE Trans. on Wireless Communications* 9.4 (2010), pp. 1483–1493.

[50] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen. “SMART: A Secure Multilayer Credit-Based Incentive Scheme for Delay-Tolerant Networks”. In: *IEEE Trans. on Vehicular Technology* 58.8 (2009), pp. 4628–4639.

[51] C. Gong, W. Bo, and Z. Faru. “SIS: Secure Incentive Scheme for Delay Tolerant Networks”. In: *Symp. on Distributed Computing and Applications to Business, Engineering Science (DCABES)*. 2012, pp. 310–313.

[52] L. Wei, H. Zhu, Z. Cao, and X. Shen. “MobiID: A User-Centric and Social-Aware Reputation Based Incentive Scheme for Delay/Disruption Tolerant Networks”. In: *Ad-hoc, Mobile, and Wireless Networks*. Ed. by H. Frey, X. Li, and S. Ruehrup. Vol. 6811. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 177–190.

[53] A. Keränen, J. Ott, and T. Kärkkäinen. “The ONE simulator for DTN protocol evaluation”. In: *Simulation Tools and Techniques (SIMUTools ’09)*. Rome, Italy: ICST, 2009.



Julian Timpner received the BS and MS degrees in computer science in 2009 and 2012, respectively, from Technische Universität Braunschweig. In 2010, he was a visiting student at the University of California, San Diego. Since 2012, he works as a research fellow at the Institute of Operating Systems and Computer Networks at Technische Universität Braunschweig, where he is also pursuing a PhD degree. His research interests include vehicular networks and e-mobility.

Dominik Schürmann received the BS and MS degrees in 2010 and 2014 respectively, from Technische Universität Braunschweig. Since 2014, he works as a research fellow at the Institute of Operating Systems and Computer Networks at Technische Universität Braunschweig, where he is also pursuing a PhD degree. His research interests include unobtrusive security in distributed systems and cryptographic algorithms in general.

Lars Wolf received the diploma degree in 1991 and the doctoral degree in 1995, both in computer science. From 1991 to 1996 he worked at IBM’s European Networking Center, until he joined the Technische Universität Darmstadt as assistant professor. Dr. Wolf joined Universität Karlsruhe (TH), in 1999 where he was associated professor in the computer science department and alternate director of the computer center. Since spring 2002 Lars Wolf is full professor for computer science at the Technische Universität Braunschweig where he is head of the Institute of Operating Systems and Computer Networks. His current research interests include wireless and mobile networking in general, sensor networks, vehicular networks, delay-tolerant networks, and network & system support for mobile systems.

Cooperative Charging in Residential Areas

Dominik Schürmann, Julian Timpner, and Lars Wolf, *Senior Member, IEEE*

***Abstract*—Electric Vehicles (EVs) require a well-developed charging infrastructure. Especially when used for the daily commute, most EV drivers will rely on a nightly charge in their garage, for instance. In typical European urban residential areas, however, private parking and charging resources are severely limited. Therefore, public on-street charging often is the only option. Yet, it faces several limitations that lead to an inefficient and unfair utilization of charging stations, or Electric Vehicle Supply Equipment (EVSE). For instance, EVSEs are often blocked by fully-charged vehicles. We thus propose and evaluate a cooperative protocol for EVs that facilitates coordinated handovers of EVSEs. We integrate this protocol with the ISO 15118 standard and provide a detailed security analysis. In the evaluation, we show that coordinated handovers significantly improve both EVSE utilization (helping to amortize the expensive operating costs) and provide benefits for EV owners by providing sufficient charging resources. This reduces range anxiety and saves them from cruising for charging.**

***Keywords*—E-mobility, charging, EV, EVSE, ISO 15118.**

I. INTRODUCTION

E-MOBILITY has the potential to harness renewable energy sources for ensuring efficacy and affordability of modern transportation systems. The typical challenges of e-mobility, which have to be solved to make EVs feasible and attractive to customers, include limited range and lack of charging infrastructure. This situation is exacerbated by the high installation and maintenance cost of EVSEs of up to 27150 € and 3075 € p.a. [1], respectively. While the problem is less grave in suburban areas (where each house or garage has a power supply) and in commercial parking lots (where centralized optimization for the available charging infrastructure can be applied [2]), it is highly doubtful whether a sufficient coverage with charging infrastructure is realizable for public on-street parking in urban residential areas, especially in light of the expected influx of electric cars in the near future. As an example, the German National Electric Mobility Platform (NPE) predicts about 1 000 000 EVs in Germany by 2020 [1], with a demand of about 70 000 public on-street charging spots alone. Thus, it is crucial to use this sparsely available infrastructure as efficiently as possible. In this paper, we focus on residential areas without private charging infrastructure, where on-street parking and charging is predominant. In these scenarios, several limitations are encountered, which result in an inefficient and unfair utilization of EVSEs.

First, different companies will operate the EVSEs. This will prevent the development of a unified reservation backend. Thus, occupancy information—and, more importantly, predictions—may not be available.

Second, on-street EVSEs are an unmanaged resource that is used in a First Come First Served (FCFS) order according to the working hours of the residents.

Third, fully-charged or even non-charging vehicles often block EVSEs, drastically reducing the total utilization and the chance of finding charging spots for other drivers. Government agencies [3] and standardization institutions [4] envision legal regulations as a means to solve this problem. However, regulations can hardly enforce an efficient usage pattern, as we will elaborate on in Section III.

A. Approach

The combination of e-mobility with autonomous driving capabilities can alleviate some of the aforementioned problems. Automatic driving applications, e.g., for parking assistance (BMW Remote Parking) or highway driving (Tesla Autopilot), are already on the market and fully automated driving has been demonstrated by several car manufacturers. It is therefore expected that limited autonomous driving capabilities, e.g., for parking scenarios, will become market-available in the next decade [5]. A fully charged (autonomous) vehicle could move to a regular parking spot in order to make the EVSE available for the next car. Yet, this does neither solve the occupancy information deficit nor does it provide a coordinated (and thus more efficient) strategy. Consequently, a more complete solution will also facilitate Inter-Vehicle Communication (IVC). IVC enables cars to be notified when an EVSE becomes available. However, if multiple vehicles learn about the availability of an EVSE, they will compete for this scarce resource and only one of them will succeed, while the others will waste time and energy in their failed attempt. To prevent this, we propose a coordinated strategy that incentivizes cooperation and mitigates malicious behavior.

B. Contribution

In this paper, we design, implement and evaluate a cooperative protocol for EVs that facilitates coordinated handovers of EVSEs. This protocol solves the abovementioned problems

---

Manuscript received December 2, 2015; revised March 15, 2016; accepted July 09, 2016. This project has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement Number [269916, V-Charge].

The authors are with the Institute of Operating Systems and Computer Networks, Technische Universität Braunschweig, 38106 Braunschweig, Germany, (E-mail: [schuermann|timpner|wolf]@ibr.cs.tu-bs.de). *(Dominik Schürmann and Julian Timpner contributed equally to this work.) (Corresponding authors: Dominik Schürmann; Julian Timpner)*

©2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works. This is the author's post-print version. The definitive version was published in IEEE Transactions on Intelligent Transportation Systems. DOI: 10.1109/TITS.2016.2590960

of scarce on-street charging as follows. EVs charging at an EVSE are incentivized to make it available to the next vehicle as soon as possible, while avoiding competition between the possible successors. It further provides occupancy availability and projection to interested EVs. A security analysis shows the protocol's practical feasibility. We integrate this protocol with the ISO 15118 [6], [7] standard, which specifies Vehicle-to-Grid (V2G) communications, certificate infrastructure and payment models for e-mobility, to demonstrate a practical implementation. To the best of our knowledge, we are the first to address the problem of efficiently and securely managing scarce on-street charging in a cooperative manner.

C. Outline

The remainder of this paper is structured as follows. Section II discusses related work on parking/charging management as well as e-mobility architecture and standards. We define our scenario and describe assumptions in Section III. Section IV presents design decisions. The proposed protocol itself and its integration with ISO 15118 is introduced in Section V. Possible attacks on the system and their mitigations are presented in Section VI. Section VII describes our simulation setup. We provide simulation results in Section VII. The paper concludes in Section IX.

II. RELATED WORK

In this section we provide a short summary of related solutions for an efficient charging/parking management. We also present the technical background of V2G communications, especially ISO 15118, as well as related work on e-mobility.

A. Centralized Reservations

A seemingly obvious solution to the three problems described in the introduction is a *central reservation* system—an approach that is widely used in parking management systems. In commercial parking lots or garages, for instance, this allows to take customer requirements (such as arrival/departure time, State of Charge (SoC), etc.) into account for a centralized scheduling and thus for a local optimization of charging resources [2]. For on-street charging, however, a central reservation system is unfeasible for the following reasons.

On the one hand, current developments suggest that the EVSE market will be highly partitioned between utility companies, gas station operators, and even vehicle manufacturers[1]. Thus, a single (i.e., unified) backend, as assumed by Bedogni *et al.* [8], providing EVSE reservations for all these different operators is implausible, just as "gathering [all parking lots] under the same authority is hard if not unfeasible" [9].

On the other hand, if there was a single backend, how would reservations be enforced? In an environment with a severe shortage of parking spots, it is not uncommon that drivers deliberately park in no-park zones (see Section III). Legal regulations alone cannot solve this problem.This is also a weakness in current standardization effort ETSI 101 556-3 [4], which assumes that "there is no point for the EV to stay longer than reserved".

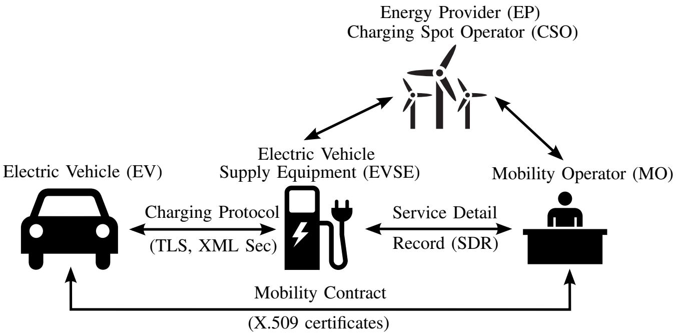


Figure 1: Relationship and security of ISO 15118 actors.

B. Decentralized Reservations

An alternative solution that overcomes some of the mentioned problems is a *decentralized* reservation mechanism. Delot *et al.* [10] present a reservation protocol for parking spaces that avoids the competition between drivers for an available spot. This approach is based on a coordinator vehicle which is responsible for assigning its spot to (one-hop) neighbor vehicles interested in an available parking space. This approach, however, assumes that vehicles in the coordinator's communication range are interested in a free parking space at this point in time. Reservations cannot be made in advance, e.g., via a query-based mechanism. This makes the protocol unsuitable for managing reservations for scarce EVSEs. What is more, we seek to make EVSEs available again as soon as possible (potentially before an EV is fully charged).

Delot *et al.* [10] also provide an in-depth study of parking management approaches. For instance, a game-theoretic approach by Ayala *et al.* [11] takes competition between drivers for available spots into account. Szczurek *et al.* [12] propose machine learning methods for finding the probability that a given parking location will be available at the time of arrival. Similarly, Caliskan *et al.* [13] estimate the future parking lot occupancy from the available information received through a VANET. Verroios *et al.* [14] investigate how to determine the best way to visit parking spots reported to be free. To the best of our knowledge, though, none of the above provides a decentralized and cooperative mechanism for on-street charging scenarios, which differ significantly from traditional parking search scenarios as described above. In addition, our protocol includes a financial incentive system and corresponding security considerations.

C. Charging Architecture: ISO 15118

Ideally, charging should be as simple as parking for the driver—yet, a sophisticated backend architecture and protocols for facilitating information exchange between EV and EVSE are required in the background. In the following, we provide a brief overview of the most important standardization efforts. Due to the extent of the matter, we have to refer the interested reader to the full standards for all details.

As depicted in Figure 1, ISO 15118-1 [6] defines the actors in the backend system and protocols for load management,

---

[1] http://www.teslamotors.com/supercharger

billing and clearing, as well as digital certificates. While it defines several payment options, we focus on the Plug and Charge (PnC) system, as it offers the highest usability. PnC requires the user to have a contract with a Mobility Operator (MO). The MO can be the utility provider that collects charging fees together with the monthly energy bill. The MO also issues X.509 certificates to the driver and EV. After a charging session, the MO receives a Service Detail Record (SDR) with all information required to pay the Energy Provider (EP) (which we assume to be identical to the Charging Spot Operator (CSO), for the sake of readability). After a charging session, the EVSE informs the EP by sending EV-signed meter receipts. The EP can use these in case of disputes. When a billing period ends, the MO provides a bill to the contracted user for received SDRs.

ISO 15118-2 [7] defines detailed communication protocols and application layer messages including security mechanisms based on Transport Layer Security (TLS) and XML Security. The message exchange between EV and EVSE happens in five phases after the EV is physically connected. In the *Communication Setup*, a session is initiated. The EV will then choose the desired charging service and agree with the EVSE on payment options in the *Identification, Authentication and Authorization* phase. We assume a contract-based payment here, as it requires no user interaction and allows PnC. Other payment options include credit, debit and prepaid cards. Before the actual charging process, parameters such as desired departure time, requested amount of energy, available power, etc. are exchanged in *Target Setting and Charge Scheduling*. Periodic ChargingStatus and MeteringReceipt messages provide updates about the charging progress during the *Charge Control and Re-scheduling* phase. In the *End of Charging Process*, the charging session is stopped.

D. E-Mobility and Smart Grid

A hot research topic is the integration of EVs into smart grid applications. This integration allows adopting dynamic pricing tariff schemes, limiting power peaks and lowering electricity bills by shifting consumption [15] towards low-demand times, typically at night. Further, EVs can support the electric grid by supplying energy from their batteries during peak hours [16]. To be fully effective, however, these approaches require many EVs to be plugged into the grid throughout the day. This is reasonable in a suburban environment with private charging infrastructure and several cars per household. In this paper's scenario, however, we focus on downtown areas with on-street charging only. As a consequence, the availability of EVSEs is highly limited (which some authors try to address through planning frameworks for EVSE locations [17]).

Reservation systems that plan routes along where EVSEs are available, either at highway exits [18] or parking lots [19], make sense for long-distance travel, but not for typical commute distances and residential parking as considered in this paper. What is more, we aim to maximize the overall EVSE utilization and EV throughput, which is contradictory to EVs being constantly plugged into the grid. Typical charging station scheduling rather focuses on maximizing revenue for load aggregators [20] or on compensating the time-varying reactive power of the grid [21].

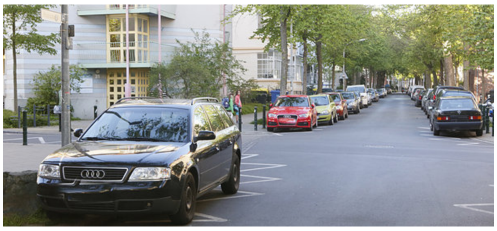


Figure 2: Multiple parking violations due to lack of regular parking spaces [22]. Hatched areas are no-parking zones.

III. SCENARIO

We focus on typical European downtown residential areas with large apartment buildings without private parking infrastructure (see Figure 2). In this environment, public on-street parking is predominant and there is a chronic shortage of parking spots, especially in the evening when most residents return from work. As a result, parking violations are quite common. In particular, drivers deliberately park in no-park zones as depicted in Figure 2, despite the risk of getting fined. This shows that legal regulations do not necessarily lead to correct behavior, if the infrastructure cannot cope with the demand.

A similar overdemand can be expected for public EVSEs, which can lead to fully-charged or non-charging vehicles blocking EVSEs, too. The German government anticipates an influx of up to 1 000 000 EVs by the year 2020, with a demand for about 70 000 public on-street AC EVSE alone (corresponding to 5 % of the total demand). With installation and maintenance cost of about 10500 € and 1750 € p.a. per unit, respectively [1], the financial feasibility of so many on-street EVSEs is highly questionable though. Fast charging DC EVSEs (e.g., Combined Charging System (CCS) [23]) are even more expensive at 27150 € plus 3075 € p.a.

A promising low-cost solution is to turn street lamp-posts into charging stations. In Berlin, Germany, several dozen are already installed, at unit costs of less than 500 €. Being connected to the low-voltage grid (which saves significant installation costs), their charging power is limited to about 3.7 kW which is also the power output of typical home chargers. Yet, most market-available EVs do not support high-power charging per default anyway. Moreover, CCS EVSEs are very expensive and thus scarce. Furthermore, fast charging generally reduces battery capacity and longevity as shown by Li *et al.* [24].

In our scenario, we consider both types of charging. In residential areas, slow (and cheap) charging at 3.7 kW prevails, resulting in several hours of charging time. For fast and emergency charging, CCS EVSEs with a power output of up to 50 kW are available along arterial roads, providing charging times of less than 30 min (resembling classical gas stations).

We assume that vehicles are equipped with IEEE 802.11p radios for Inter-Vehicle Communication (IVC) and have on-

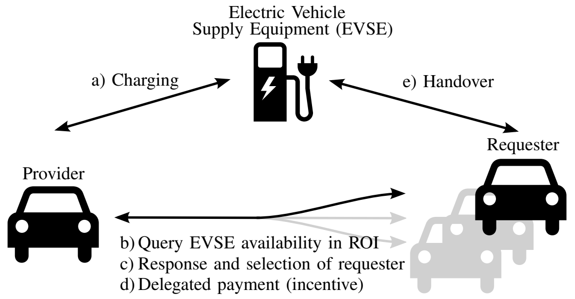

Electric Vehicle
Supply Equipment (EVSE)
a) Charging
e) Handover
Requester
Provider
b) Query EVSE availability in ROI
c) Response and selection of requester
d) Delegated payment (incentive)

Figure 3: Protocol Design

board digital maps including the position of EVSEs that are generally compatible (in terms of plug standard, voltage, etc.). Further, we assume low-speed autonomous driving/parking capabilities, which have been successfully demonstrated already [25]. This first deployment phase of autonomous vehicles is expected in the next five years [5], which corresponds to the forecasting horizon of when a large number of EVs is to be expected on the streets. Autonomous EVs can either be charged inductively [26] or be connected automatically via robotic arms [27]. These systems can be integrated with lantern-post systems (and thus use the existing low-voltage grid).

IV. PROTOCOL DESIGN

In this section, we look at the scenario's main research issues as described in Section I and propose solutions. Figure 3 depicts the resulting protocol design from a high-level perspective, while Section V describes each protocol phase in detail. Initially, a vehicle occupies an EVSE in a specific Region of Interest (ROI) and starts charging (Figure 3, Step a). This EV is called *provider* in the following.

A. Distribution of Occupancy Information

Drivers should be able to learn in advance whether specific EVSEs, for instance close to their home, are currently occupied or when they will be available again. We therefore facilitate IVC to send a query for EVSE availability information into the geographical ROI (Figure 3, Step b, and Figure 4). EVs querying for information are therefore referred to as *requesters*.

The provider(s) in the ROI respond with the estimated time to complete the charging process (Figure 3, Step c). The requester EV thus learns about whether or not it is practical to drive to this particular EVSE.

B. Cooperation instead of Competition

To increase the overall utilization and the efficient use of available EVSEs, we seek a coordinated strategy for handing over EVSEs to the next EV. The main goals are to (i) increase the overall chances to use an EVSE by preventing exclusive First Come First Served (FCFS) usage, and (ii) to avoid inefficient competition which would result from naïvely broadcasting EVSE availability, for instance.

This is realized as follows. Similar to Delot *et al.* [10], the provider becomes the coordinator for its resource and chooses a successor from the set of interested requesters (Figure 3, Step c). While different selection processes are conceivable, we deliberately use a uniform distribution to randomly select a successor from the set of requesters. We do so to ensure a fair selection process. As the selection process is random, requesters do not need to provide any additional personally identifiable information. Future models might select the highest bidder or the EV with the least charging time remaining in order to maximize EVSE usage. However, this would introduce security risks, as such properties cannot easily be verified and business models for malicious nodes could arise.

C. Incentivized Cooperation

In order to minimize blocking of EVSEs by fully-charged or non-charging vehicles, we propose a financial incentive system for making EVSEs available to other vehicles as soon as possible.

The basic idea is to reimburse a provider, who vacates an EVSE for a requester, for the expenses of (i) reparking and (ii) possibly forgoing a full battery charge by leaving early. We measure these expenses in kWh, meaning that providers do not actually get paid for vacating an EVSE. Instead, the provider can split his bill with a requester, who then pays the EP for a certain amount of energy of the provider's current charging session.

1) Payment Model: As described in Section II-C, the EVSE informs the EP about a charging session by sending EV-signed meter receipts consisting of a timestamp and the charged kWh. The provider EV keeps copies and has the requester EV sign a share of the receipts. The signed receipts are forwarded to the EVSE which acknowledges the delegated payment (Figure 3, Step d). The provider vacates the EVSE and the requester takes his place, starting the charging process (Figure 3, Step e).

The exact amount that is delegated to the requester depends on the provider's current State of Charge (SoC).

We assume that every driver estimates a minimum energy level required for the next driving task, SoC_{min}. This estimation can, for example, be based on recorded energy consumptions from previous commuting times, which vary depending on the traffic situation and other uncertainties. In our scenario, drivers are not willing to vacate an EVSE if $SoC < SoC_{\text{min}}$.

If, however, $SoC \geq SoC_{\text{min}}$, a driver would vacate the EVSE and thus forego a full battery charge in exchange for a small incentive. If the provider's SoC is close to 100 %, he is more likely to do so and the incentive can be smaller. The lower the actual SoC is, the larger the incentive needs to be. To the best of our knowledge, there is no study on the percentage of users willing to do this.

2) Cost: The total cost c that the provider bills the requester for vacating an EVSE includes the cost of forgoing a full battery charge and the cost c_{park} of reparking. Providers will only participate in the protocol if the available meter receipts cover the cost c, as follows. The cost c_{park} of reparking depends on the scenario, i.e., how many parking spots are available in the vicinity in comparison to the total demand, the time of day, the day of the week, etc. This can be estimated by each vehicle from empirical values, coordinated between a group of vehicles as proposed in Parking Communities [28], or be provided by a backend service [29]. The cost of forgoing a full battery charge can be expressed as a percental surcharge on c_{park}:

$$\begin{aligned} SoC \geq SoC_{\min} : c &= c_{\text{park}} \cdot (1 + (1 - SoC)) \\ &= c_{\text{park}} \cdot (2 - SoC). \end{aligned}$$

Consequently, if $SoC = 100\,\%$, only the cost of reparking needs to be reimbursed: $c = c_{\text{park}}$.

3) Discussion: The model has several benefits. First, no actual money is transferred and no new virtual currencies are required, as it integrates with the existing ISO 15118 standard. Second, it handles different pricing models: some EPs charge per hour, others per kWh. Hybrid models exist as well. Meter receipts include both the charged kWh as well as the charging time and can thus be used independently from the actual pricing model. In particular, if a bill is split between provider and requester, both of them can pay their share according to their respective tariff model. The total revenue of the EVSEs operator is not affected, as charging sessions are always paid completely (but may be split between provider and requester).

In the future, energy prices may vary greatly, possibly in minutes. ISO 15118 prepares for this: EVSEs provide EVs with a sales tariff table to calculate a charging schedule. The proposed protocol can take advantage of this as follows: EV A is not willing to pay the current high price and thus waits for when charging is cheaper according to the tariff table. EV B, on the other hand, is willing to accept a higher price to charge as soon as possible. Consequently, B hands over the EVSE according to the proposed protocol, for as long as the higher price is valid. When B has been charged to $SoC_{\min}$, it hands over the EVSE to A again.

V. PROTOCOL INTEGRATION WITH ISO 15118

In this section, we build upon the design decisions in Section IV to derive a cooperative protocol for EVs that facilitates coordinated EVSE handovers. The terms *resource*, *requester*, and *provider* are used as introduced the previous section. We integrate this protocol with ISO 15118-2 [7], which specifies Vehicle-to-Grid (V2G) communications as described in Section II-C. Consequently, Figure 5 and Figure 6 include clearly marked references to the standard (as described in Section II-C) where it interfaces with our original contributions.

A. Phase I: Query

In Phase I, requesters send a SpotReq message via geocast[2] [30] into the EVSE areas that come into question (because of their vicinity to the driver's home location, for instance). Figure 4 depicts this situation. In addition, vehicles might query backend services for occupancy information [29].

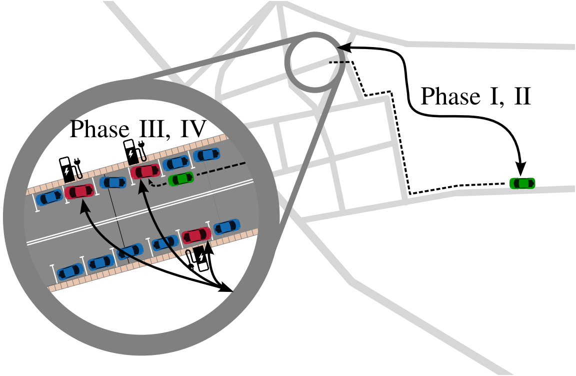


Figure 4: Phase I: A requester geocasts a SpotReq into a ROI while driving home and receives SpotRes messages. Phase II: Provider selects requester. Phase III: Requester waits close to the provider while exchanging receipts. Phase IV: Payment is delegated and the EVSE is handed over.

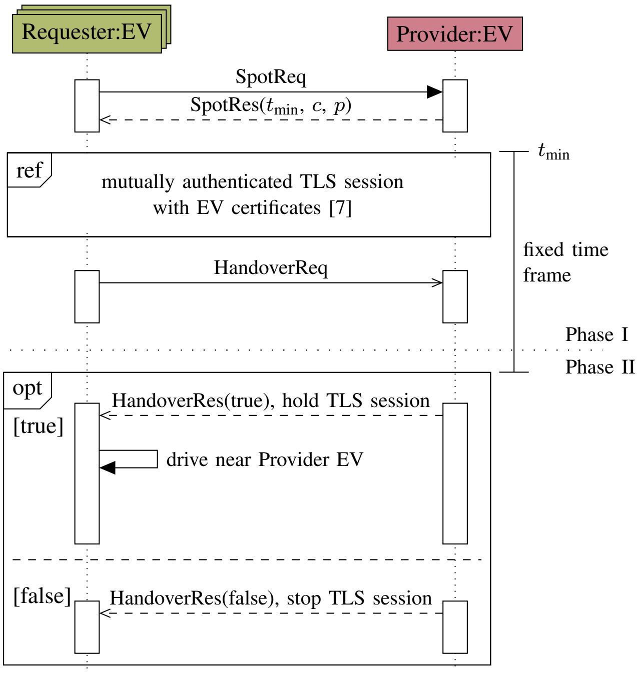


Figure 5: Phase I and II: Query and Competition.

If there is an EVSE available, the requester drives there. This is the case, either if the backend knows about the EVSE status or if no SpotRes message is received. However, if the EVSE is occupied, the occupying EV (provider) can be considered a full-fledged network node as connected EVs do not suffer from a limited energy supply. The provider will thus respond with a SpotRes message as depicted in Figure 5.

1) Provider: The provider can further estimate (based on its current SoC and the EVSE's power output) when it will have reached $SoC_{\min}$. SpotRes thus contains an estimated time

---

[2] A geocast is a special form of multicast in which destination nodes are identified by geographical positions.

t_{min} of when SoC_{min} has been reached, that is the earliest point in time when the charging resource can be released, such that another EV can charge. Moreover, SpotRes includes the provider's estimated cost c to vacate the EVSE early (see Section IV-C) as well as a proof p that the provider is actually charging (by means of a valid metering receipt per ISO 15118).

2) Requester: It is noteworthy that the SpotReq/SpotRes message pair is authenticated via regular Vehicle-to-Vehicle (V2V) Public Key Infrastructure (PKI) certificates [31]. Due to geocasting, the provider is not known at this point. Thus, the provider's certificate is not available and the query cannot be encrypted. Requesters check the plausibility of c after receiving a SpotRes message and then decide to (a) drive to a different EVSE or to (b) park in a regular parking spot and wait until t_{min}. EVs still interested in the occupied EVSE at t_{min} announce their interest and their taking part in the selection process via a HandoverReq. This polling mechanism avoids having the provider to keep track of vehicles that have sent queries at some point in the past, thus reducing storage overhead. From now on, mutually authenticated TLS sessions will be established between requester(s) and provider using EV certificates signed by the V2V PKI as shown in Figure 5. The provider notices the demand for "his" blocked EVSE with the first HandoverReq. It starts a timer, waiting for other HandoverReq to arrive.

B. Phase II: Competition

After a timeout, Phase II starts with the selection process. The provider determines a successor as described in Section IV-B. The selected requester is notified via a HandoverRes and instructed to hold the TLS session. It can now either notify the driver or, if applicable, automatedly drive towards the EVSE and wait close to the provider as shown in Figure 4 (green car). All other petitioners are also notified and the corresponding TLS sessions are stopped (see Figure 5). The rejected requesters wait for HandoverRes messages from other providers they may have queried or, if necessary, periodically send additional SpotReq messages to (increasingly larger) ROIs. Note that the time frame between t_{min} and the beginning of Phase II is fixed, so that the overhead of keeping TLS sessions alive is limited.

C. Phase III: Receipt Exchange

While the selected requester is approaching the EVSE, the provider is typically still charging and following the ISO 15118 protocol for V2G communications between EVs and EVSEs. In particular, in the *Charge Control and Re-scheduling* loop (see Figure 6), it receives MeterInfo data in the ChargingStatusRes and sends signed MeteringReceiptReq messages to the EVSE. The signed meter receipts can be used for billing purposes as they provide proof that the charging process has taken place, as explained in Section II-C. The provider keeps records of these receipts.

Remember that all communication between requester and provider is now secured via the previoulsy created TLS session (see Phase II). Using this secure connection, the provider sends a share of the receipts to the requester via the V2VConnectReq/-Res message pair. This share equals the cost c in kWh as defined in Section IV-C. The requester checks the validity of the received receipts and compares them against the cost c of the initial SpotRes message.

D. Phase IV: Payment Delegation

The last phase is concerned with the actual payment as shown in Figure 6 (lower half). To this end, the requester first sets up a communication session with the EVSE *resource*. As the EVSE has no wireless networking support itself, this connection setup is done multi-hop via the provider, who is still physically connected to the EVSE. In other words, the requester's ISO 15118 [7] TLS session, which is only unilaterally authenticated, is tunneled through the mutually authenticated TLS session between both vehicles.

The requester agrees to pay the split bill by signing the meter receipts using his own contract certificate. It sends the signed meter receipts to the resource with the DelegatedPayReq. The resource verifies the requester's signature and acknowledges the pay delegation to the provider including the E-Mobility Account Identifier (eMAID), which uniquely identifies the requester. The provider can now safely assume that the requester is assuming liability and further forwards the DelegatedPayRes as an acknowledgement. It can now close the connection with the resource and drive away, while the requester takes its place. The requester is now able to continue the session using a direct connection proceeding to its own *Charge Control and Re-scheduling* loop.

VI. ATTACK MODEL

To evaluate the security of our protocol, possible attack scenarios are discussed in detail. Our security design is based on well known primitives, which we accept as assumptions. This includes the Public Key Infrastructure (PKI) of the V2V communication [31] as well as Transport Layer Security (TLS). Following the Dolev–Yao model, we also assume that the resource EVSE, i.e., the endpoint, is not compromised. Nevertheless, before going into protocol details, classical attacks are discussed in the context of our protocol design.

A. Classical Attacks

Man-in-the-Middle (MitM) An attacker who acts as a MitM in the TLS connection between requester and provider could decipher, inject, and alter messages.

Solution: These attacks are prevented by executing the mutually authenticated TLS handshake using EV certificates, which are in turn signed by an Original Equipment Manufacturer (OEM) organized via the V2V PKI.

Impersonation and Sybil attacks Impersonating another EV would allow to acquire a charging spot that was already paid for by the legitimate EV. This could be done by replaying an eavesdropped message. Sybil attacks would allow to increase the probability to be selected in the Phase II, i.e., by using many fake vehicle identities while querying.

Solution: Similar to the MitM scenario, these attacks are mitigated by the V2V PKI and TLS' replay protection mechanisms [32].

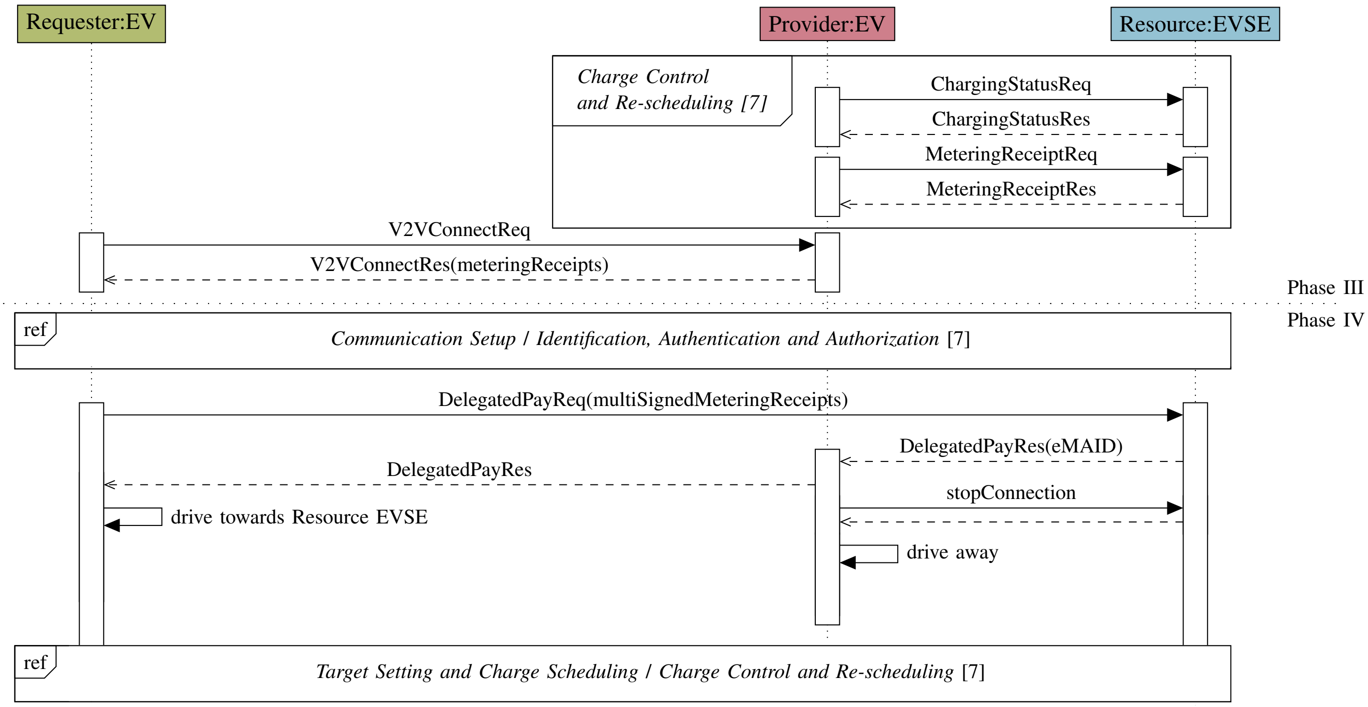

Requester:EV
Provider:EV
Resource:EVSE
Charge Control and Re-scheduling [7]
ChargingStatusReq
ChargingStatusRes
MeteringReceiptReq
MeteringReceiptRes
V2VConnectReq
V2VConnectRes(meteringReceipts)
Phase III
Phase IV
ref
Communication Setup / Identification, Authentication and Authorization [7]
DelegatedPayReq(multiSignedMeteringReceipts)
DelegatedPayRes(eMAID)
DelegatedPayRes
stopConnection
drive towards Resource EVSE
drive away
ref
Target Setting and Charge Scheduling / Charge Control and Re-scheduling [7]

Figure 6: Phase III and IV: Receipt Exchange and Payment Delegation.

Compromising private keys If private keys are compromised, the corresponding certificates have to be revoked to prevent their usage by adversaries.

Solution: Mechanisms to distribute revocations are revocation lists or queries via Online Certificate Status Protocol (OCSP). ISO 15118-2 [7] requires OCSP for Sub-CA certificates inside the chain to the EVSE certificate. EVSE certificates themselves are short-term, thus no revocation mechanism is deployed here. How EV certificates are revoked is defined by the V2V PKI.

B. Denial of Service (DoS)

Naïve DoS A simple DoS can be executed by sending many SpotReq messages.

Solution: Thanks to EV certificates and digital signatures in V2V communication, providers can block excessive requesters by their identity.

Requesting charging spot without MO contract An EV requests a parking spot via SpotReq but does not have a valid contract with a MO.

Solution: In case this EV is selected, the contract is verified by the resource in Phase III of the protocol. If this verification fails, the resource stops the process and informs the provider. The provider cancels the protocol and waits for new SpotReq messages. The attacking EV should then be blocked by its identity.

Requester sending invalid multiSignedMeteringReceipts If the requester sends multiSignedMeteringReceipts with invalid signatures, the resource cancels the process.

Solution: Similar to the previous scenario the provider falls back to receive new SpotReq messages.

Requester not sending multiSignedMeteringReceipts The requester can cancel the protocol without informing the provider or resource, e.g., by not sending multiSignedMeteringReceipts.

Solution: The protocol defines a timeout to handle this situation. After its expiry, the provider falls back to receive new SpotReq messages.

C. Protocol Attacks

Location privacy The SpotReq/-Res messages can be read by neighboring EVs as they are not encrypted. Thus, other EVs know when and where a charging spot is vacated.

Solution: SpotReq messages cannot be encrypted as they are sent via geocast routing to a target location, not to a specific previously known EV. This is a conceptual limitation of such routing algorithms.

Replaying multiSignedMeteringReceipts The requester replays multiSignedMeteringReceipts from a previous payment delegation.

Solution: These meteringReceipts cannot be used again because only meteringReceipts valid for this particular session are accepted, i.e., meteringReceipts that were created before and signed by the EVSE.

Honeypot The provider sends a HandoverRes(true) to multiple requesters, thus decoying them to drive to his resource. He then splits his metering receipts between the waiting requesters, delegating a larger percentage of his bill than usually possible. Only one requester can obtain the EVSE after the provider leaves, though.

Solution: This fraud is easily detectable: A requester, who has paid for EVSE access, has the metering receipts as a proof which can be used by a clearing house to resolve this. In fact, the physical attacks *Third-Party occupying EVSE* and *Provider not driving away* are very similar.

D. Physical Attacks

Naïve blocking of charging spot An EV could block the charging spot to make profit.

Solution: As explained in the *Replaying multiSignedMeteringReceipts* scenario, no valid meteringReceipts are available that could be delegated. Thus a provider cannot easily make a business out of this.

Third-party occupying EVSE A third-party vehicle drives into the spot vacated by the provider although the requester has paid for it.

Solution: This improbable scenario is averted by requiring the requester to drive as near as possible to the current provider before proceeding with Phase III.

Provider not driving away After a delegated payment by the requester to the resource, the provider does not drive away and still occupies the charging spot.

Solution: In this case, the requester should park besides the provider and inform local authorities. Because the provider still occupies the spot, it can easily be held responsible for not following the protocol and be towed away. A naïve solution to this attack is to postpone the payment until the provider vacated the spot. However, this bears the risk for the provider to not get paid, which is more difficult to resolve than the original attack.

VII. Simulator Extension and Configuration

Veins [33], a framework for vehicular network simulations based on SUMO and OMNeT++, has been extended to support our scenarios. We extend SUMO vehicles with a battery, working day movement, and a behavior model.

A. Battery Model

The EV's battery model has been adopted from Bedogni *et al.* [8]. Because a realistic simulation of battery physics would be too resource consuming, their battery model is an efficient approximation for large scale vehicle simulations. In comparison to real-world EV battery discharging, it is still highly accurate [8]. Incorporating force, current speed, and vehicle properties, the average power consumption P_{mean} is calculated for each step t_{step}. The total battery capacity B_{capacity} and the efficiency η of transforming electric energy to mechanical energy are configured as constants. Battery discharging, i.e., the consumed battery capacity SoC_t for each simulation step t_{step} is calculated as:

$$SoC_t = SoC_{t-1} - \frac{P_{\text{mean}} \cdot t_{\text{step}}}{B_{\text{capacity}}} \cdot \frac{1}{\eta}$$

Table I: Summary of simulation parameters.

EVs	Number of EVs	325 [a]
	Vehicle type	e-Golf
	Weight	1585 kg
	Energy usage	12.7 kWh/100km
	Battery capacity	24 kWh
	Max charging power	AC 3.6 kW, DC 50 kW [b]
	Charging time 3.6 kW	~8 h [c]
EVSEs	Number of AC EVSEs	22 [a] / 44
	Charging power	AC 3.7 kW, DC 40 kW
Movement	Start of day	between 07:00 - 09:00 [d]
	Working duration	8 h (+ up to 60 min [d])
	Pr for leisure activities	30 %
	Avg. speed outside [e]	70 km/h
	Commuting distances	cf. Figure 8
	Pr to charge at work	25 % [a]
	SoC_{eager}	High ($\geq$85 %) [f] / Low (50 %)

[a] estimation based on Nationale Plattform Elektromobilität [1]
[b] with special CCS charging equipment
[c] approx. calculation based on Bedogni *et al.* [8]
[d] leisure activities, uniform distribution
[e] for workplaces outside of the simulated area
[f] threshold based on commuting distance, forces recharging after every trip

Battery recharging uses $pow(t)$, the power delivered by an EVSE for each simulation step t_{step}, and $w(SoC_{t-1})$, a battery-depended coefficient representing charging characteristics. Recharging is calculated as:

$$SoC_t = SoC_{t-1} + \frac{(pow(t) \cdot t_{\text{step}}) \cdot w(SoC_{t-1})}{B_{\text{capacity}}}$$

By varying $pow(t)$, AC and fast DC charging can be simulated. As represented in Table I, we assume a Volkswagen *e-Golf* as the vehicle type, lamp-post AC EVSEs (cf. Section III), and fast charging DC EVSEs.

B. Map

As described in Section III, we focus on a typical residential area, namely a city district of Braunschweig, Germany. The Östliches Ringgebiet[3] is densely populated (6400 people/km^2), with an area of 4 km^2 and a population of 26 616 people [34]. This area has been exported from OpenStreetMap and converted to a SUMO road network. As depicted in Figure 7, 22 AC EVSEs (based on NPE estimation [1]) have been sensibly placed inside this area. In addition, a second scenario with 44 EVSEs has been generated to simulate a highly optimistic estimation. DC EVSEs' locations are not mapped, instead these are dynamically used while commuting between home area and workplace.

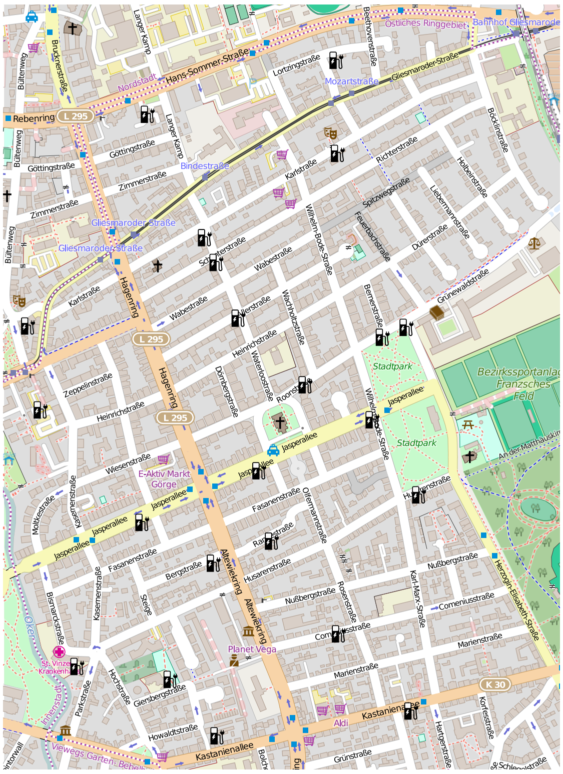


Figure 7: Map of Östliches Ringgebiet with 22 EVSEs.

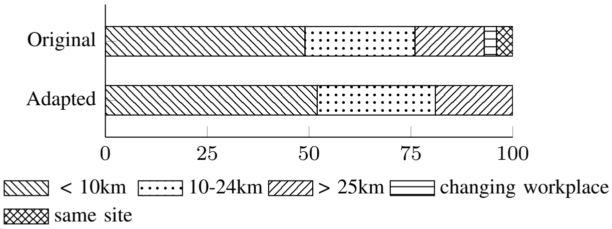


Figure 8: Distribution of commuting distances (from Statistisches Bundesamt [35]). For our simulation, *same site* values are neglected and *changing workplace* values are proportionately distributed among the remaining parts.

C. Working Day Movement

We have implemented a simplified version of the Working Day Movement by Ekman *et al.* [36]. A home area and a workplace location are assigned to each individual vehicle where the distance between these locations is based on the vehicle's commuting distance. The latter are distributed according to Figure 8. Home areas are randomly mapped into the simulated area. Depending on the commuting distance, workplaces are likely to lie outside of the simulated map. If vehicles leave the map (and thus town), we assume an average speed of 70 km/h.

Table II: Lookup table to determine the search radius and n, the number of EVSEs an EV is trying to recharge at.

Distance [km]	SoC [%]	n	Radius [m]
$1 \leq x < 10$	SoC < 15	1 [a]	150
	$15 \leq$ SoC < 25	2 [a]	200
	$25 \leq$ SoC < 50	4	400
	$50 \leq$ SoC < 75	2	200
	$75 \leq$ SoC < 90	2	200
	$90 \leq$ SoC < 100	1	150
$10 \leq x < 25$	SoC < 25	5	500
	$25 \leq$ SoC < 50	4	500
	$50 \leq$ SoC < 75	3	400
	$75 \leq$ SoC < 90	2	300
	$90 \leq$ SoC < 100	1	150
$25 \leq x \leq 60$	SoC < 50	5	500
	$50 \leq$ SoC < 75	4	400
	$75 \leq$ SoC < 90	3	300
	$90 \leq$ SoC < 100	2	150

[a] SoC is so low that only a small number of EVSEs can be tried lest the battery fully discharges.

A SoC_{min} value is assigned to each individual vehicle. In our simulation, it is a fixed value based on the vehicle's average energy consumption and the vehicle's commuting distance. In real-world implementations, however, SoC_{min} can be calculated dynamically if historic data such as previous energy consumptions are available (cf. Section IV-C). As presented in Table I, a working day starts between 07:00–09:00 when the EVs drive to their assigned workplaces. Besides parking at work for 8 h (plus a 30 % chance of up to 60 min to simulate shopping or leisure activities), EVs also have a 25 % chance to charge at work. On their way back home, the drivers' eagerness to charge comes into play. Current research [37] shows that drivers feel an urge to charge earlier than actually necessary. If an EV's SoC_t falls below the eagerness threshold SoC_{eager}, it tries to find charging in its home area. The higher SoC_{eager}, the earlier a driver wants to charge. If an EV is looking for charging, its search radius and the maximum number n of EVSEs it will try to charge at are looked up in Table II, which defines a rough approximation of realistic human behavior. An EV with a low commuting distance, for example, that has a SoC_t of 50 % or lower tries up to 4 EVSEs in a radius of 400 m, i.e., without the proposed protocol it drives to each one and checks if it is occupied or not until a free one is found. A larger commuting distance results in a higher urge to recharge in general and a lower SoC_t means that more EVSEs are tried in a larger radius. If an EV's SoC_t reaches the lowest defined threshold in the lookup table while driving, a DC emergency charging at 40 kW (cf. Table I) is dynamically scheduled. Thus, fully discharged vehicles are prevented.

---

[3] https://en.wikipedia.org/wiki/%C3%96stliches_Ringgebiet

D. Scenarios

Two general scenarios have been implemented. The uncooperative scenario (abbreviated as U) without our protocol, where EVs block EVSEs until the next morning, and the cooperative scenario (C), simulating the protocol behavior defined in Section V. To evaluate effects resulting from an increased number of available EVSEs or different SoC_{eager} levels, we define the following 8 fine grained configurations:

U-22-H Configuration U-22-H represents an uncooperative scenario with 22 EVSEs and a high $SoC_{\text{eager}} \geq 85\,\%$. The number of EVs and EVSEs correspond to the NPE predictions [1]. Furthermore, a high SoC_{eager} value represents careful/selfish drivers recharging their EVs after every trip even though their SoC is sufficiently high for the next trip.

U-22-L In contrast, in U-22-L a rather low SoC_{eager} of 50 % represents less anxious and more friendly drivers. It has been shown that the mean and median SoCs [37] at which recharging is performed are 55.5 % and 56 %, respectively. The number of EVSEs and EVs is not modified.

U-44-H, U-44-L In configurations U-44-H and U-44-L, the number of EVSEs is doubled. All remaining parameters are chosen analogue to U-22-H and U-22-L, respectively.

C-22-H, C-22-L, C-44-H, C-44-L To simulate our cooperative protocol, the same configurations are used. Additionally, each scenario is extended by $SoC_{\text{min}} = 90\,\%$ (minimum SoC required for the next driving task, see Section IV-C).

VIII. EVALUATION

In this section we evaluate the efficiency of the proposed protocol in terms of its impact on the number of satisfied charging requests, the number of required emergency DC charging sessions and EVSE utilization.

A. Satisfied Charging Requests

Figure 10 depicts the total number of charging sessions per day for each configuration. In the uncooperative scenarios, the available EVSEs clearly cannot satisfy the demand, as the number of charging sessions equals the number of available charging resources. The primary reason of course is that EVs misuse the EVSEs as parking spots after they have been fully charged. Noticeably more EVs are recharging in the cooperative scenario. From Day 3 on, the number of charging sessions increases as a lot of vehicles consumed most of their energy on the first days and are now required to recharge. Especially in C-22-H and C-44-H (in comparison to U-22-H and U-44-H, respectively), a substantial increase from 22 (or 44) to 160 charging sessions due to EVSE handovers is visible. C-22-L and C-44-L show a more moderate increase beginning on Day 3, caused by a lower number of requesters because of a more friendly SoC_{eager} of 50 %.

For a more detailed analysis of how many charging requests can actually be satisfied and how many futile attempts are required to do so, Figure 9 exemplarily plots the following metrics for U-22-H and C-22-H:

1) **Maximum visits** m: Sum of each individual maximum number of EVSEs an EV considers to visit in order to recharge: $0 \leq m \leq n$ (cf. Table II)

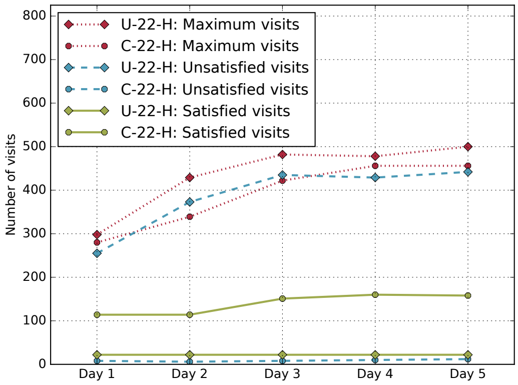


Figure 9: Comparison of the number of satisfied and unsatisfied visits between U-22-H and C-22-H.

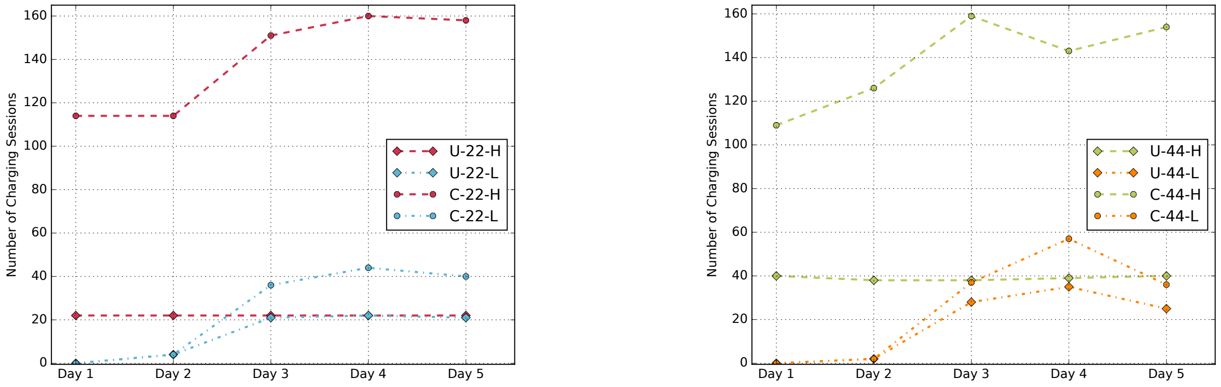
2) **Unsatisfied visits** u: Total number of visited EVSEs that were occupied: $0 \leq u \leq m$
3) **Satisfied visits** s: Total number of visited EVSEs that were available: $s \in \{0, 1\}$

Note that $m \neq u + s$ because if an EV would try up to $m = 3$ EVSEs and the first EVSE is available, there are no unsatisfied visits $u = 0$, but $s = 1$.

In the uncooperative scenario, Figure 9 shows a very high number of futile visits to occupied EVSEs, indicating a constant high need to recharge which cannot be satisfied in most cases. Up to about 440 unsatisfied visits per day are required to achieve 22 satisfied visits only. Again, these 22 (or 44 in U-44-H/L) visits equal the total number of available EVSEs which clearly indicates an overload situation. Configurations with a low eagerness to charge (U-22-L and U-44-L) just delay this overload until Day 3 (not shown).

In the cooperative scenario as shown by Figure 9, almost no unsatisfied visits occur for two reasons. First, EVs do not have to drive to occupied EVSEs since they learn about their availability beforehand via SpotReq/SpotRes message pairs. If all EVSEs in its ROI are occupied, an EV drives directly to its home zone. Second, EVSE handovers avoid competition so that only the chosen requester will drive to it. Hence, unsatisfied visits are generally avoided. However, sporadic unsatisfied visits can occur if at least two EVs are heading towards the same available EVSE at the same time because they both received a positive SpotRes message. The total number of such occurences is negligible, though, as Figure 9 shows. Further, the number of satisfied visits is considerably higher than in the corresponding uncooperative scenario, namely up to 160 (cf. Figure 10a).

In sum, the proposed protocol enables up to 160 vehicles to charge per day, while only 22 EVs were able to do so in the uncooperative scenario with the same number of EVSEs. At the same time, the number of unsatisfied visits per day is reduced from about 440 to less than 12, thus saving time and energy as well as reducing traffic. Even in a more optimistic scenario with 44 EVSEs the increase of charging sessions (159 compared to 44) and the saved futile visits (up to 329) are



(a) 22 EVSEs

(b) 44 EVSEs

Figure 10: Comparison of the number of charging sessions per day between uncooperative and cooperative scenarios.

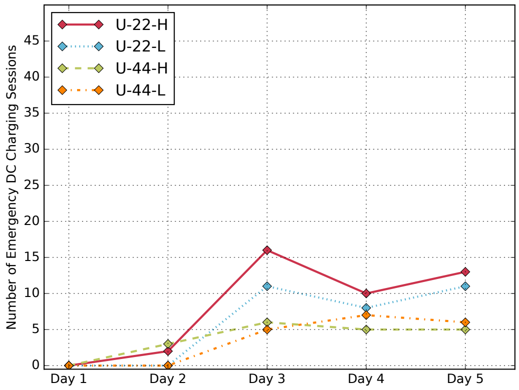


Figure 11: Number of emergency DC charging sessions in the uncooperative scenario.

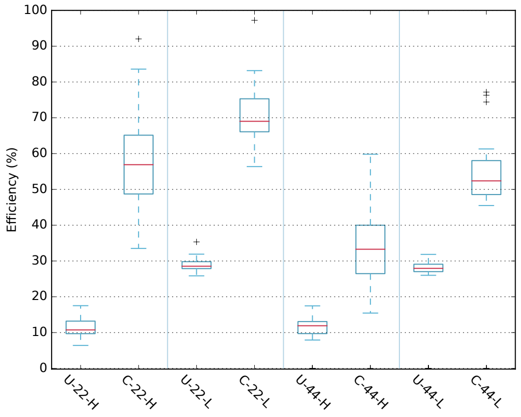


Figure 12: Efficiency of EVSE utilization in uncooperative and cooperative scenarios.

substantial.

B. Emergency DC Charging

As the previous section has shown, 22 EVSEs cannot satisfy the number of charging requests in the uncooperative scenario. This also applies to low eagerness configurations, in which drivers are very friendly and only search for an EVSE if it is really necessary. Even doubling the available EVSEs to 44 does not significantly reduce the problem.

Another proof for this overload situation is depicted in Figure 11, which shows the number of required emergency DC charging sessions per day. DC charging sessions are performed if all EVSEs that are considered by a specific EV are occupied, but the EV's current SoC requires charging to cover the upcoming commuting distance. As shown, DC charging is necessary to prevent fully discharged EVs' batteries starting from Day 2 for U-22/44-H and Day 3 for U-22/44-L, respectively.

In the cooperative scenario, however, there is no need for DC charging at all throughout the simulation period (not depicted). Thus, each EV recharges and maintains a SoC that is high enough to cover upcoming commuting distances via regular AC EVSEs alone. This not only shows that the existing number of 22 EVSEs can completely satisfy the total demand if the proposed protocol is used, but that the more expensive and harmful (in terms of battery life [24]) DC charging can be avoided.

C. EVSE Utilization

Figure 12 shows the efficiency of EVSE utilization. In particular, the corresponding uncooperative and cooperative configurations are directly compared. The efficiency is measured as the ratio of how long an EVSE was actually providing energy and how long it was blocked by an EV. A high efficiency thus indicates that the EVSE was not misused as a parking spot by fully-charged or non-charging vehicles.

Figure 12 illustrates the highly inefficient use of EVSEs in the uncooperative scenarios—recharging in U-22-H is finished after about 11 % of the occupancy time. For the remaining time, the EVSE is misused as a parking spot. EVs with a lower SoC_{eager} in U-22-L need more time to fully recharge, leading to a higher but still insufficient utilization. EVSE handovers significantly reduce the amount of time EVSEs are blocked by non-charging EVs, as the significant increase of charging sessions per day in Section VIII-A shows. Further, DC charging can be completely avoided (see Section VIII-B). Consequently, the EVSE efficiency improves substantially as shown in Figure 12. For 22 EVSEs, an increase of up to 46 % is possible. Moreover, it is observable that the variance in the cooperative scenario is higher because of a larger range of EVs' SoCs recharging at different EVSEs. In an optimistic (but more unrealistic) scenario with 44 EVSEs, there is still an improvement of about 21 %. The reason for the smaller improvement is of course due to fewer requesters per EVSE in the cooperative scenario as the demand is distributed across more EVSEs.

IX. CONCLUSION

In this paper, we have presented a cooperative protocol for EVs that facilitates coordinated handovers of EVSEs to solve the challenges of scarce on-street charging. EVs charging at an EVSE are incentivized to make it available to the next vehicle as soon as possible, while avoiding competition between the possible successors. It further provides occupancy availability and projection to interested EVs. In a detailed security analysis we have shown the protocol's practical feasibility. We have further integrated this protocol with the ISO 15118 [6], [7] standard, which specifies V2G communications, certificate infrastructure and payment models for e-mobility, to demonstrate a practical implementation.

Simulation results show that in an uncooperative (i.e., without our protocol) scenario, the anticipated number of publicly available AC EVSEs in an urban residential area cannot satisfy the expected charging demand. This overload results in a large number of emergency DC charging sessions (which strain battery life) as well as additional mileage and energy consumption for vehicles that are cruising to find charging resources. Even doubling the number of available EVSEs does not considerably improve the situation. We have shown that the proposed protocol is able to substantially improve the efficient utilization of existing EVSEs by 21 % to 46 %. The number of daily charging sessions can be increased by a factor of 7. As a result, emergency DC charging as well as cruising for charging resources can be completely avoided. In sum, coordinated handovers have been shown to significantly improve both EVSE utilization (helping to amortize the expensive operating costs) and to provide benefits for EV owners by providing sufficient charging resources. This reduces range anxiety and saves them from cruising for charging.

A. Future Work

In this paper we assume low-speed autonomous driving capabilities that allow for automatically executing handovers. Future versions might be based on a smartphone application that notifies drivers of a handover request which they can either confirm or decline. Nightly handovers would then be affected by the driver's mood, behavior or if he/she is sleeping. Moreover, we are planning to evaluate the impact of alternative incentive models which have been out of scope for this paper.

ACKNOWLEDGMENT

We would like to thank Florian Franke for his support.

REFERENCES

[1] Nationale Plattform Elektromobilität, "Fortschrittsbericht 2014 – Bilanz der Marktvorbereitung", Gem. Geschäftsst. Elektromobilität der Bundesreg., Dec. 2014, p. 76.

[2] J. Timpner and L. Wolf, "Design and Evaluation of Charging Station Scheduling Strategies for Electric Vehicles", *IEEE Trans. Intell. Transp. Sys.*, vol. 15, no. 2, pp. 579–588, 2014.

[3] Nationale Plattform Elektromobilität, "Vision und Roadmap der Nationalen Plattform Elektromobilität", Gem. Geschäftsst. Elektromobilität der Bundesreg., 2013, p. 27.

[4] *TS 101 556-3 - V1.1.1 - Intelligent Transport Systems (ITS); Infrastructure to Vehicle Communications; Part 3: Communications system for the planning and reservation of EV energy supply using wireless networks*, ETSI, 2014.

[5] Inventivio GmbH. (2015). Autonomous car forecasts, [Online]. Available: http://www.driverless-future.com/?page_id=384 (visited on 08/19/2015).

[6] *ISO/IEC DIS 15118-1: Road vehicles – Vehicle to grid communication interface – Part 1: General information and use-case definition*, Int. Organization for Standardization, 2013.

[7] *ISO/IEC DIS 15118-2: Road vehicles – Vehicle to grid communication interface – Part 2: Network and application protocol requirements*, Int. Organization for Standardization, 2014.

[8] L. Bedogni, L. Bononi, M. DiFelice, *et al.*, "An Integrated Simulation Framework to Model Electric Vehicles Operations and Services", *IEEE Trans. Vehicular Technology*, vol. PP, no. 99, pp. 1–17, 2015.

[9] N. Mejri, M. Ayari, R. Langar, *et al.*, "Cooperation Versus Competition Towards An Efficient Parking Assignment Solution", in *IEEE Communications*, Jun. 2014, pp. 2921–2926.

[10] T. Delot, S. Ilarri, S. Lecomte, *et al.*, "Sharing with caution: Managing parking spaces in vehicular networks", *Mobile Information Systems*, vol. 9, no. 1, pp. 69–98, 2013.

[11] D. Ayala, O. Wolfson, B. Xu, *et al.*, "Parking in Competitive Settings: A Gravitational Approach", in *13th IEEE MDM*, Jul. 2012, pp. 27–32.

[12] P. Szczurek, B. Xu, O. Wolfson, *et al.*, "Learning the Relevance of Parking Information in VANETs", in *Seventh ACM VANET*, Sep. 2010, pp. 81–82.

[13] M. Caliskan, A. Barthels, B. Scheuermann, *et al.*, "Predicting Parking Lot Occupancy in Vehicular Ad Hoc Networks", in *65th IEEE VTC (Spring)*, Apr. 2007, pp. 277–281.

[14] V. Verroios, V. Efstathiou, and A. Delis, "Reaching Available Public Parking Spaces in Urban Environments Using Ad Hoc Networking", in *12th IEEE MDM*, Jun. 2011, pp. 141–151.

[15] N. G. Paterakis, O. Erdinc, I. N. Pappi, *et al.*, "Coordinated Operation of a Neighborhood of Smart Households Comprising Electric Vehicles, Energy Storage and Distributed Generation", *IEEE Trans. Smart Grid*, vol. PP, no. 99, pp. 1–12, 2016.

[16] M. J. E. Alam, K. M. Muttaqi, and D. Sutanto, “Effective Utilization of Available PEV Battery Capacity for Mitigation of Solar PV Impact and Grid Support With Integrated V2G Functionality”, *IEEE Trans. Smart Grid*, vol. PP, no. 99, pp. 1–10, 2015.

[17] H. Zhang, Z. Hu, Z. Xu, *et al.*, “An Integrated Planning Framework for Different Types of PEV Charging Facilities in Urban Area”, *IEEE Trans. Smart Grid*, vol. PP, no. 99, pp. 1–12, 2015.

[18] H. Qin and W. Zhang, “Charging Scheduling with Minimal Waiting in A Network of Electric Vehicles and Charging Stations”, in *8th ACM VANET*, Sep. 2011, pp. 51–60.

[19] S. Hashimoto, R. Kanamori, and T. Ito, “Auction-Based Parking Reservation System with Electricity Trading”, in *IEEE 15th Business Informatics*, Jul. 2013, pp. 33–40.

[20] C. Jin, J. Tang, and P Ghosh, “Optimizing Electric Vehicle Charging With Energy Storage in the Electricity Market”, *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 311–320, 2013.

[21] B. Jiang and Y. Fei, “Decentralized scheduling of PEV on-street parking and charging for smart grid reactive power compensation”, in *IEEE ISGT*, Feb. 2013.

[22] M. Korth. (May 15, 2015). ÖSTLICHES RINGGEBIET: MORGENS UM 7 IST DIE WELT IN UNORDNUNG, [Online]. Available: http://www.unser38.de/braunschweig-innenstadt/menschen/oestliches-ringgebiet-morgens-um-7-ist-die-welt-in-unordnung-d12320.html (visited on 10/30/2015).

[23] *IEC 62196-3:2014 - Plugs, socket-outlets, vehicle connectors and vehicle inlets - Conductive charging of electric vehicles - Part 3: Dimensional compatibility and interchangeability requirements for d.c. and a.c./d.c. pin and contact-tube vehicle couplers*, Electrotechnical Commission (IEC), 2014.

[24] Y. Li, F. El Gabaly, T. R. Ferguson, *et al.*, “Current-induced transition from particle-by-particle to concurrent intercalation in phase-separating battery electrodes”, *Nature Materials*, vol. 13, no. 12, pp. 1149–1156, Dec. 2014.

[25] P. Furgale, U. Schwesinger, M. Rufli, *et al.*, “Toward Automated Driving in Cities using Close-to-Market Sensors: An Overview of the V-Charge Project”, in *IEEE IV*, Jun. 2013, pp. 809–816.

[26] J. Meins, F. Soyck, B. Engel, *et al.*, “Application of high-power inductive charging of electric buses in scheduled line service”, in *Hybrid and Electric Vehicles Symposium*, ITS Niedersachsen, Feb. 2014.

[27] “Produkte: e-smartConnect”, *ATZelektronik*, no. 4, pp. 30–33, Apr. 2015.

[28] J. Timpner, D. Schürmann, and L. Wolf, “Trustworthy Parking Communities: Helping Your Neighbor to Find a Space”, *IEEE Trans. Dependable and Secure Comp.*, vol. 13, no. 1, 2016.

[29] B. C. Communications. (Jun. 3, 2015). The next step in connected navigation – on-street parking prediction, [Online]. Available: https://www.press.bmwgroup.com/global/pressDetail.html?id=T0220542EN (visited on 01/02/2016).

[30] J. Timpner and L. Wolf, “Query-response geocast for vehicular crowd sensing”, *Ad Hoc Networks*, vol. 36, Part 2, no. Veh. Netw. for Mobile Crowd Sensing, pp. 435–449, Jun. 2016.

[31] P. Papadimitratos, L. Buttyan, T. Holczer, *et al.*, “Secure vehicular communication systems: design and architecture”, *IEEE Communications Mag.*, vol. 46, no. 11, pp. 100–109, Nov. 2008.

[32] T. Dierks and E. Rescorla, *The Transport Layer Security Protocol Version 1.2*, RFC 5246, Updated by RFCs 5746, 5878, 6176, IETF, Aug. 2008.

[33] C. Sommer, R. German, and F. Dressler, “Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis”, *IEEE Trans. Mobile Comp.*, vol. 10, no. 1, pp. 3–15, Jan. 2011.

[34] Stadt Braunschweig. (2010). Braunschweig in der Statistik, [Online]. Available: http://www.braunschweig.de/politik_verwaltung/statistik/BS_Statistik_Jahrbuch_2010_neu.pdf (visited on 10/03/2015).

[35] (2014). Berufspendler: Infrastruktur wichtiger als Benzinpreis, [Online]. Available: https://www.destatis.de/DE/Publikationen/STATmagazin/Arbeitsmarkt/2014_05/2014_05Pendler.html (visited on 10/03/2015).

[36] F. Ekman, A. Keränen, J. Karvo, *et al.*, “Working day movement model”, in *ACM MobilityModels*, 2008, pp. 33–40.

[37] G. M. Fetene, C. G. Prato, S. Kaplan, *et al.*, “Harnessing Big-Data for Estimating the Energy Consumption and Driving Range of Electric Vehicles”, in *95th Annual Meeting of the Transp. Research Board*, 2016.

Dominik Schürmann received the BS and MS degrees in 2010 and 2014 respectively, from Technische Universität Braunschweig. Since 2014, he works as a research fellow at the Institute of Operating Systems and Computer Networks at Technische Universität Braunschweig, where he is also pursuing a PhD degree. His research interests include unobtrusive security in distributed systems and cryptographic algorithms in general.

Julian Timpner received the BS and MS degrees in computer science in 2009 and 2012, respectively, from Technische Universität Braunschweig. In 2010, he was a visiting student at the University of California, San Diego. Since 2012, he works as a research fellow at the Institute of Operating Systems and Computer Networks at Technische Universität Braunschweig, where he is also pursuing a PhD degree. His research interests include vehicular networks and e-mobility.

Lars Wolf received the diploma degree in 1991 and the doctoral degree in 1995, both in computer science. From 1991 to 1996 he worked at IBM’s European Networking Center, until he joined the Technische Universität Darmstadt as assistant professor. Dr. Wolf joined Universität Karlsruhe (TH), in 1999 where he was associated professor in the computer science department and alternate director of the computer center. Since spring 2002 Lars Wolf is full professor for computer science at the Technische Universität Braunschweig where he is head of the Institute of Operating Systems and Computer Networks. His current research interests include wireless and mobile networking in general, sensor networks, vehicular networks, delay-tolerant networks, and network & system support for mobile systems.

Dominik Schürmann*, Fabian Kabus, Gregor Hildermeier, and Lars Wolf

Wiretapping End-to-End Encrypted VoIP Calls: Real-World Attacks on ZRTP

Abstract: Voice calls are still one of the most common use cases for smartphones. Often, sensitive personal information but also confidential business information is shared. End-to-end security is required to protect against wiretapping of voice calls. For such real-time communication, the ZRTP key-agreement protocol has been proposed. By verbally comparing a small number of on-screen characters or words, called Short Authentication Strings, the participants can be sure that no one is wiretapping the call. Since 2011, ZRTP is an IETF standard implemented in several VoIP clients.

In this paper, we analyzed attacks on real-world VoIP systems, in particular those implementing the ZRTP standard. We evaluate the protocol compliance, error handling, and user interfaces of the most common ZRTP-capable VoIP clients. Our extensive analysis uncovered a critical vulnerability that allows wiretapping even though Short Authentication Strings are compared correctly. We discuss shortcomings in the clients' error handling and design of security indicators potentially leading to insecure connections.

Keywords: ZRTP, VoIP, SIP, key exchange

DOI 10.1515/popets-2017-0025

Received 2016-11-30; revised 2017-03-15; accepted 2017-03-16.

1 Introduction

Following Snowden's global surveillance revelations from 2013, public awareness of privacy and information security has increased and driven the demand for products aiming to protect their users. Thus, the design and implementation of end-to-end secure messaging protocols received a lot of attention [9, 34]. In 2016, these protocols have been adopted by mainstream messaging apps, such as WhatsApp and Facebook Messenger [10, 35]. As a result, mobile messaging, the most popular smartphone feature, finally includes end-to-end encryption for average users. Comparing their security features with that of voice calls shows a major imbalance. While making voice calls is the second most popular smartphone feature with 93% popularity [25], its security is often neglected. It is difficult to retrofit the traditional Public Switched Telephone Network with end-to-end security, but it is feasible to protect users of modern Voice over IP (VoIP) apps.

To protect real-time communication channels, the ZRTP key agreement protocol has been proposed. Based on the Diffie-Hellmann (DH) key exchange, it has been standardized in 2011 as RFC 6189 [37]. It can be implemented independently of the actual signaling protocol, however it is often used in conjunction with the Session Initiation Protocol (SIP) [16]. Instead of relying on a central Public Key Infrastructure (PKI), participants have to compare a few digits or words, called Short Authentication Strings (SASs). If done correctly, no one should be able to actively wiretap the call, i.e., perform an unnoticed Man-in-the-Middle (MitM) attack. The exchanged secrets are utilized to encrypt the stream end-to-end, usually using the Secure Real-Time Transport Protocol (SRTP).

Before its standardization, ZRTP has been formally verified by Bresciani et al. [6]. The authors analyzed the protocol with the "correctly verified SAS" assumption. In 2007/2008, Gupta and Smatikov [14] as well as Petraschek et al. [24] discuss practical attacks, in particular a flaw in the handling of ZRTP IDs (ZIDs). A recent study by Shirvanian and Saxena with 128 online-participants found that for a two-word SAS, an attacker can stay undetected with about 30% probability [32]. In 2016, two theoretical attacks against ZRTP have been published by Bhargavan et al. [3]. They discuss a version downgrade attack as well as a downgrade from DH to Preshared mode. To this day, no systematization of attacks has been done. Also, protocol attacks have only been discussed theoretically or applied to the abandoned Zfone desktop software.

In this paper, we analyze attacks against modern real-world ZRTP systems. It is known that an evil SIP operator can conduct MitM attacks, which can only be

---

***Corresponding Author: Dominik Schürmann:** TU Braunschweig, E-mail: schuermann@ibr.cs.tu-bs.de
Fabian Kabus: TU Braunschweig, E-mail: kabus@ibr.cs.tu-bs.de
Gregor Hildermeier: TU Braunschweig, E-mail: hilderme@ibr.cs.tu-bs.de
Lars Wolf: TU Braunschweig, E-mail: wolf@ibr.cs.tu-bs.de

detected by SAS comparison. We demonstrate the simplicity how to design a minimally invasive MitM attack that re-routes calls and records conversations in real-time. In the main part of our paper, we analyze attacks against specific ZRTP clients. Here, we assume that SASs are *correctly compared* by end users. We define a set of protocol test cases for verification of standard compliance as well as UI conformance tests. The most common ZRTP clients on major platforms, such as Android, iOS, Windows, and Linux, have been evaluated. Our findings include a critical vulnerability in Linphone (CVE-2016-6271) allowing wiretapping even though SASs have been compared correctly. We report about an issue in Jitsi, were a normal call is misinterpreted as an attack, resulting in a security warning that should have not been displayed. Furthermore, several weaknesses in the clients' user interfaces have been uncovered. By adapting our test cases and best practices we provide guidance on how to properly implement ZRTP.

After introducing ZRTP in Section 2, we explore the possibility of wiretapping encrypted VoIP calls in Section 3. Assuming the use of ZRTP and correct comparison of SASs, we provide protocol and non-protocol specific test cases in Section 4. Using these tests, we evaluate protocol compliance and usability of ZRTP clients in Section 5. In Section 6, we propose best practices for client developers on how to properly design a SAS verification UI. Previous studies and other related work are discussed in Section 7. In Section 8, we discuss the implications of our findings, before concluding the paper in Section 9.

2 ZRTP Fundamentals

The ZRTP key agreement protocol has been standardized in RFC 6189 [37] and uses SASs to detect MitM attacks. This agreement is transported over a Real-Time Transport Protocol (RTP) communication channel that has previously been established by a signaling protocol, such as SIP. The SASs are derived from the DH shared secrets and displayed on end users' displays. They need to be compared verbally by reading them out loud and verifying that the peer's SAS matches with the displayed one. In case of a MitM attack, the participants end up with different shared secrets and thus different SAS. The SASs are very short, e.g., 'bz4f' (B32 encoding) or 'spearhead Yucatan' (B256 encoding with PGP Wordlist [18, 19]), while still providing enough security due to the usage of a hash commitment [36]. This restricts a MitM to only one attempt to guess the correct key for generating the same SAS.

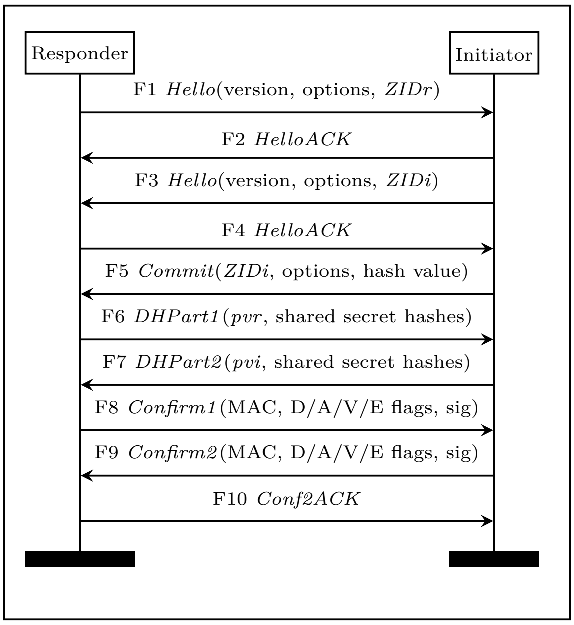

Responder
Initiator
F1 Hello(version, options, ZIDr)
F2 HelloACK
F3 Hello(version, options, ZIDi)
F4 HelloACK
F5 Commit(ZIDi, options, hash value)
F6 DHPart1(pvr, shared secret hashes)
F7 DHPart2(pvi, shared secret hashes)
F8 Confirm1(MAC, D/A/V/E flags, sig)
F9 Confirm2(MAC, D/A/V/E flags, sig)
F10 Conf2ACK

Fig. 1. ZRTP handshake in DH Mode between an Initiator (right) and a Responder (left). The protocol consist of three phases: *Discovery and Version/Algorithm Negotiation (F1-F4)*, *Key Agreement (F5-F7)*, and *Key Confirmation and Derivation (F8-F10)*.

In this section, we provide an overview of ZRTP following the notation of RFC 6189 [37]. We focus on parts of the protocol relevant to our analysis in this paper. A representative call flow can be seen in Figure 1. During the exchange, one party ends up as the Initiator and the other as the Responder. The underlying transport layer protocol is most certainly UDP. Because errors of 16 bit UDP checksums cannot be distinguished from active MitM attacks, all ZRTP packets contain an additional Cyclic Redundancy Check (CRC) to detect errors. Additionally, two exponential backoff retransmission timers are utilized: one for *Hello* messages, the other for all messages sent after *HelloAck*. After the ZRTP handshake is complete, the SASs and keys for a SRTP session are derived and the SRTP session is established. The SAS then has to be compared verbally to ensure that no MitM was between the endpoints. If something goes wrong during the exchange an *Error* message is sent with a specific error code encoding what caused the handshake to fail. Following Figure 1, we will look into the three phases of the protocol, namely *Discovery and Version/Algorithm Negotiation (F1-F4)*, *Key Agreement (F5-F7)*, and finally *Key Confirmation and Derivation (F8-F10)*.

2.1 Discovery and Version/Algorithm Negotiation (F1-F4)

Both endpoints begin the exchange by sending a *Hello* message ensuring the peer also supports ZRTP. The endpoint is identified by a unique randomly generated 96 bit ZID. *Hello* includes the supported ZRTP version, which is used for the version negotiation: The highest version supported by both parties is used. At the time of this writing the ZRTP version is 1.10. The *Hello* message also includes supported hash and cipher algorithms, as well as authentication tag, key agreement and SAS types. The chosen key agreement type then is the fastest both parties have in common. For the remaining parameters, the Initiator may choose one mutually supported type. Received *Hello* messages are acknowledged by subsequent *HelloACK* messages.

2.2 Key Agreement (F5-F7)

After both *Hello* messages have been received, a *Commit* message begins the key agreement. The *Commit* message is sent by the Initiator containing her ZID as *ZIDi*. It is now possible to either proceed in Diffie-Hellmann mode (DH mode) or with existing cached shared secrets in Preshared mode.

2.2.1 DH Mode

In DH mode, the endpoints carry out a straightforward DH key exchange to derive a shared DH secret using the *DHPart1* and *DHPart1* messages (cf. Figure 1). To extend the DH in a way to protect against brute force attacks, the Initiator first commits to a public value *pvi* by including $hvi = hash(pvi)$, where SHA-256 is used for $hash()$ by default, in the *Commit* message. The Responder answers directly with her public key *pvr*. Only after receiving *pvr*, the Initiator sends her public key *pvi* she has committed to in the *Commit* message. This hash commitment provides a protection against pre-computation of hash collisions during the DH exchange [36]. Thus, an attacker can only guess with a chance of one out of 65536 when using a 16 bit SAS [37]. Finally, the DH secret is derived using the Initiator's or Responder's own secret key *svi* / *svr*:

$$\mathit{DHResult} = pvr^{svi} \mod p = pvi^{svr} \mod p$$

If both sides send a *Commit* at the same time, there are rules to break the tie: If one *Commit* is for DH mode and the other for Preshared mode, the DH mode wins. Otherwise, the one with the larger *hvi* value wins. The party with the winning *Commit* becomes the Initiator, the peer becomes the Responder.

2.2.2 Preshared Mode

In Preshared mode, the endpoints can skip the DH if they have shared secrets *rs1* / *rs2* from a previous session. Subsequent shared secrets are derived from the previous one that gives this mode similar properties like in DH mode, such as forward secrecy: If an attacker gains access to this secret, previous calls still can not be decrypted as old key material is immediately destroyed after use.

Shared secrets *rs1* / *rs2* are held in a long-term cache and associated with a ZID. An additional boolean flag is stored to indicate if the SAS has already been compared and verified. It is important to note that entries are not associated to the SIP address, only to the ZID. Furthermore, there is no mapping between ZIDs and SIP addresses in the ZRTP protocol. One can use many devices each with their own ZID but configured for the same SIP address. There can even be many SIP addresses configured using the same ZID. The RFC proposes to allow labeling of ZIDs to indicate the devices they are associated to, such as “Alice on her office phone”.

Identifiers for the cached shared secrets are transmitted in the *DHPart* messages. By the Responder they are calculated as:

$$\mathit{rs1IDr} = MAC(rs1, \text{‘}\texttt{Responder}\text{’})$$
$$\mathit{rs2IDr} = MAC(rs2, \text{‘}\texttt{Responder}\text{’})$$

rs1IDi, *rs2IDi* on the Initiator's side are calculated analogously using ‘`Initiator`’ as the second argument for the MAC. If a shared secret is not available, a random value is used instead. This hides from an eavesdropper that shared secrets are actually available.

2.3 Key Confirmation and Derivation (F8-F10)

First, a hash is calculated over the previous messages:

$$\begin{aligned}\mathit{total\_hash} = hash(&\mathit{Hello}_{\mathit{Responder}} \parallel \mathit{Commit} \\ &\parallel \mathit{DHPart1} \parallel \mathit{DHPart2})\end{aligned}$$

Both parties then continue to calculate $s0$ that dependents on the *Key Agreement Type*. In the following a simplified version of the protocol is presented[1].

2.3.1 DH Mode

For DH mode:

$$s0 = \mathit{hash}(\mathit{counter} \parallel \mathit{DHResult} \parallel \text{'ZRTP-HMAC-KDF'} \parallel \mathit{ZIDi} \parallel \mathit{ZIDr} \parallel \mathit{total\_hash})$$

where $\mathit{counter} = 1$.

2.3.2 Preshared Mode

For Preshared mode, prior to sending the *Commit* message the Initiator calculated:

$$\mathit{preshared\_key} = \mathit{hash}(\mathit{len}(\mathit{rs1}) \parallel \mathit{rs1})$$

where $\mathit{len}()$ denotes the length in octets. Finally, both participants proceed to calculate $s0$:

$$\mathit{KDF\_Context} = \mathit{ZIDi} \parallel \mathit{ZIDr} \parallel \mathit{total\_hash}$$

$$s0 = \mathit{KDF}(\mathit{preshared\_key}, \text{'}\texttt{ZRTP PSK}\text{'}, \mathit{KDF\_Context}, \text{negotiated hash length})$$

where the Key Derivation Function is defined as:

$$\mathit{KDF}(\mathit{KI}, \mathit{Label}, \mathit{Context}, L) = \mathit{HMAC}(\mathit{KI}, i \parallel \mathit{Label} \parallel 0x00 \parallel \mathit{Context} \parallel L)$$

2.3.3 Updating Shared Secret Cache

The cache is updated with a new derived $rs1$:

$$\mathit{rs1} = \mathit{KDF}(\mathit{s0}, \text{'}\texttt{retained secret}\text{'}, \mathit{KDF\_Context}, 256)$$

2.3.4 SAS and SRTP Key Derivation

$s0$ has been derived regardless of the utilized mode. The following calculations are identical for both modes:

$$\mathit{ZRTPSess} = \mathit{KDF}(s0, \text{'}\texttt{ZRTP Session Key}\text{'}, \mathit{KDF\_Context}, \mathit{hash\_len})$$

where $\mathit{hash\_len}$ is the negotiated hash algorithm length. The SAS is calculated by:

$$\mathit{sashash} = \mathit{KDF}(s0, \text{'}\texttt{SAS}\text{'}, \mathit{KDF\_Context}, 256)$$

where the leftmost 16 bit or 20 bit of the *sashash* are used for B32 or B256 encoding of the SAS respectively.

Finally, *srtpkey*, *srtpsalt*, *mackey*, and *zrtpkey* are calculated. The *srtpkey* and *srtpsalt* are used to encrypt the SRTP traffic, *mackey* is used by ZRTP as the key for subsequent HMAC calculations and *Confirm* messages are encrypted with the *zrtpkey*:

$$\begin{aligned}
\mathit{srtpkeyi} &= \mathit{KDF}(s0, \text{'}\texttt{Initiator SRTP master key}\text{'}, \mathit{KDF\_Context}, \mathit{aes\_length}) \\
\mathit{srtpsalti} &= \mathit{KDF}(s0, \text{'}\texttt{Initiator SRTP master salt}\text{'}, \mathit{KDF\_Context}, 112) \\
\mathit{mackeyi} &= \mathit{KDF}(s0, \text{'}\texttt{Initiator HMAC key}\text{'}, \mathit{KDF\_Context}, \mathit{hash\_len}) \\
\mathit{zrtpkeyi} &= \mathit{KDF}(s0, \text{'}\texttt{Initiator ZRTP key}\text{'}, \mathit{KDF\_Context}, \mathit{aes\_length})
\end{aligned}$$

where *aes_length* is the negotiated AES key length. The Responder keys are calculated analogously. Successful generation of keys is then confirmed via the *Confirm* messages, which signal that the SRTP session may now start.

2.3.5 PBX Enrollment

To support PBX (essentially a telephone system) in large companies, the ZRTP standard includes a PBX enrollment procedure. This feature is a critical component because it officially supports MitMs. The enrollment can be initiated by setting the PBX Enrollment flag (E) in the *Confirm* message. The client should provide a UI to let the user decide if she wants to allow this PBX for future communications. For this, a secret is calculated:

$$\mathit{pbxsecret} = \mathit{KDF}(\mathit{ZRTPSess}, \text{'}\texttt{Trusted MiTM key}\text{'}, \mathit{ZIDi} \parallel \mathit{ZIDr}, 256)$$

If a PBX has been accepted, *pbxsecret* is cached and the MitM flag (M) can be set in future *Hello* messages allowing the PBX forwarding of SAS via a special *SASRelay* message.

---

[1] In contrast to RFC 6189 [37], we exclude *pbxsecret* and *auxsecret* in our calculations and omit the values $s1, s2, s3$, which are usually part of the $s0$ calculation.

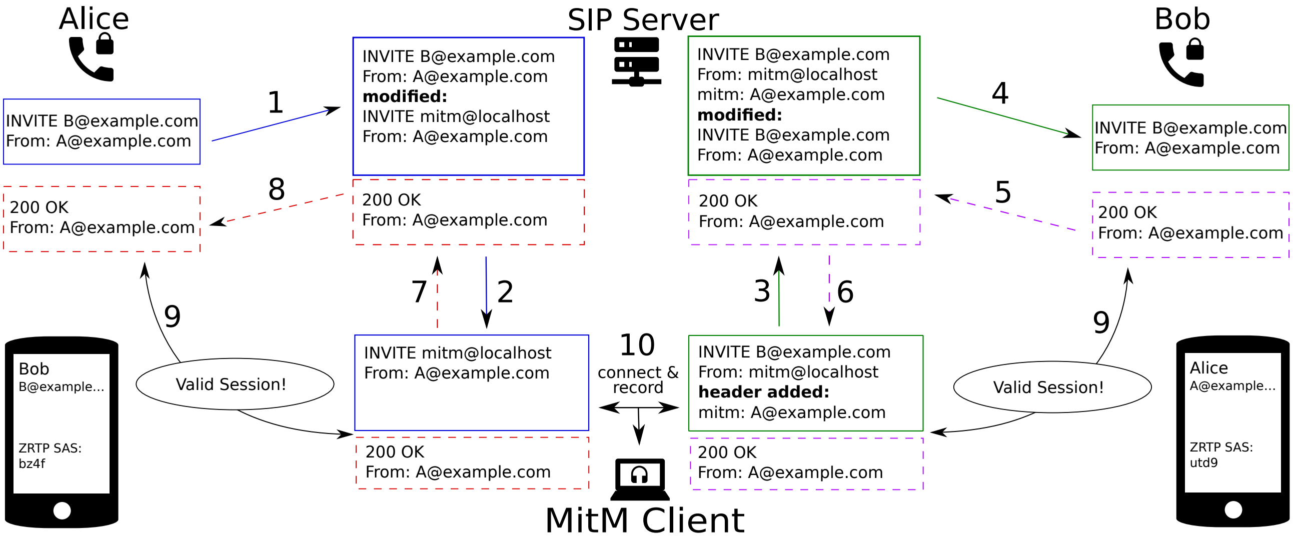


Fig. 2. Flow of minimally invasive wiretapping: The SIP server re-routes INVITE messages to a MitM client connecting the multimedia streams and records the conversation of Alice and Bob. Between the MitM client and the SIP server a 'mitm' header is introduced to pass-through the original 'From' header. As expected, the displayed SASs are different.

3 Wiretapping VoIP Calls

We motivate the importance of end-to-end encryption and authentication support in VoIP clients by showing the simplicity of wiretapping calls by an *evil operator*, i.e., someone having access to the central components of the VoIP network. Our implementation is designed to be as non-invasive as possible regarding to the original SIP flow between the caller Alice and callee Bob. In this section, we do not attempt to break or circumvent ZRTP. Instead we demonstrate the feasibility of wiretapping VoIP calls if SASs are *not* compared.

3.1 Design

To keep the interference and modifications to a minimum, we decided to implement the MitM by hooking into the SIP flow. An attack possibility is given by modifying incoming messages to forward calls to a recording MitM client. This leads to a tunnel through the MitM instead of a direct connection between Alice and Bob. The modification takes care that the header of the messages always contains the originally called SIP address(es) to cover up the attack. A special MitM SIP client accepts any incoming call automatically, starts a second call to the original callee Bob, and connects the incoming data stream from Alice with the new outgoing stream to Bob. Now, everything works according to the protocol but with a MitM recording the multimedia stream, i.e., the conversation. Following Figure 2, wiretapping works along these steps:

1. Alice initiates a call to Bob.
2. The server manipulates the INVITE message such that it is forwarded to the MitM client at 'mitm@localhost'. The information that the message should have been forwarded to Bob is not lost during the modification, because the 'To' header has not been changed.
3. The MitM client initiates, before accepting the incoming call, a second call to Bob. A new 'mitm' header is added to the outgoing call containing the original 'From' address from the incoming one.
4. The server manipulates the INVITE message from the MitM client such that it looks like it came from Alice.
5. Bob acknowledges the INVITE message.
6. This acknowledgment is forwarded as normal to the MitM client, because the MitM is the original caller.
7. The MitM client automatically accepts Alice's call.
8. This acceptance is forwarded as normal to Alice, because she is the caller.
9. Now, two valid connections have been established. Optionally securing these with ZRTP would now lead to divergent SASs.
10. The MitM client connects both multimedia channels and records everything.

We extended Kamailio [20] with the described MitM capabilities. Our patch forwards calls to a MitM client using PJSIP [26] and ZRTP4PJ [8]. Messages are modified based on their type, e.g., INVITE, BYE, or OK. Multimedia streams are connected together using PJSIP's conference bridge framework. The whole call is recorded using PJSIP's recorder class. Our implementation has been published as open source software[2].

3.2 Summary

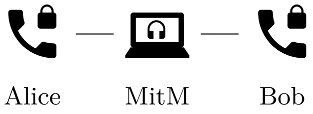
Our MitM implementation for the SIP allows wiretapping of VoIP calls. We were able to verify the correctness of our implementation by initiating calls between our accounts 'A@example.com' and 'B@example.com'. While the clients of Alice and Bob still show the expected SIP addresses, the calls were recorded by our MitM. These attacks can only be protected by end-to-end authentication methods. Enabling ZRTP allows to detect the attacker because the SAS displayed on Alice's client is unequal to Bob's SAS, e.g., "bz4f" vs. "utd9".

4 Attacking ZRTP Clients

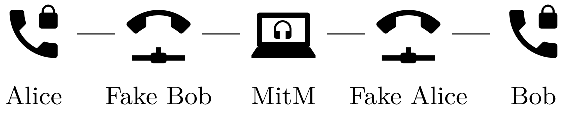
After motivating the importance of ZRTP by showing how MitM attacks stay undetected without it, we now focus on the security of ZRTP-capable VoIP clients. Assuming more cautious participants, who actually compare SASs by voice, an attacker may choose to employ specific attacks tailored towards the usability and correctness of specific software. It is important to note that we will not analyze the possibility of forging spoken SASs, which has already been done in several user studies [23, 31, 32]. Instead, we analyze specific issues of common ZRTP-capable VoIP clients.

4.1 Attack Methodology

We differentiate between two types of MitM attacks depending of the power of an *evil operator*: In the non-impersonating MitM attack (cf. Figure 3a), the attacker is a node on the route between Alice and Bob. While multiple SIP servers can be involved, only one server on the route needs to be behave maliciously for the whole call to be wiretapped. The attacker can read and modify forwarded ZRTP messages, as well as inject her own. The encrypted parts are opaque to the attacker, as she does not have the encryption key.



(a) Non-impersonating MitM: The attacker is a node in between and can forward as well as inject packets.



(b) Active MitM: The attacker maintains two separate calls to Alice and Bob and connects the streams.

Fig. 3. Types of MitM attacks

In contrast, if the attacker is an active MitM (cf. Figure 3b) she maintains two separate calls to Alice and Bob, impersonating the peers. In Section 3, this powerful MitM attacker has been implemented to be able to freely modify multimedia streams. ZRTP aims to detect these types of attacks by having the participants compare SASs by voice.

4.2 Protocol Test Cases

In the following, we provide functional test cases. These either follow the protocol specification to test basic implementation requirements or explicitly violate parts of RFC 6189 [37]. An app passes a test if it behaves according to the expected results defined per test. For a brief overview, our test cases are also summarized in Table 1.

[zrtpCall] Basic Call Functionality: Two calls are made: One should be secured using DH mode, the other using Preshared mode (if supported). To initiate DH mode, the first call uses a new random ZID. This simulates the scenario that either the participants never spoke before over ZRTP-secured VoIP or that one participant uses a new device or client (no shared secrets in cache). In malicious scenarios, an active MitM impersonating one SIP addresses can force a new ZID to bypass Preshared mode and fallback to DH mode.
Test: Conduct two consecutive ZRTP-protected calls.
Expected Results: The calls should succeed as expected.

[verDown] Version Downgrade (Non-Impersonating MitM): As analyzed by Bhargavan et al. the version negotiation in ZRTP is not protected against downgrade attacks [3]. In this test, an old version number is announced instead of the current version

---

[2] https://gitlab.ibr.cs.tu-bs.de/groups/zrtp

Table 1. Overview of protocol and non-protocol test cases

Protocol Tests

	Test	Expected Result
[zrtpCall]	Basic calls in DH and Preshared mode	Successful calls
[verDown]	Version downgrade to '`1.0`'	Abort key agreement
[weakDH]	DH public key set to '`1`'	Abort key agreement
[invSS]	Invalid shared secret in ZID cache	Inform user and re-execute SAS comparison
[invCom]	Invalid commit *hvi*	Abort key agreement
[sharedMitM]	Third person who shares secrets with victims acts as a MitM	ZID labeling / Association between SIP and ZID
[pbxEnroll]	Enrollment for PBX with *Confirm* message	Proper UI or abort key agreement if unsupported

Non-Protocol Tests

	Expected Result
[statusInd]	Security indicators distinguishing the provided security levels with icon and text
[confSAS]	Confirmation dialog with button to confirm SAS
[termError]	On protocol error, terminate the connection automatically
[secDef]	Provide secure defaults for VoIP providers

of 1.10. Previous versions are susceptible to an attack described by Gupta and Shmatikov [14], thus we expect that current implementations only provide 1.10.

Test: The version number inside the *Hello* message is set to '`1.0`'.

Expected Results: The ZRTP key agreement must be aborted. The standard specifies that a received *Hello* message must be ignored if an unsupported version number is received.

[weakDH] Weak DH (Non-Impersonating MitM): This simulates a non-impersonating MitM attack, where the endpoints operate in DH mode. In this attack, a weak DH is enforced by using a *DHPart* (cf. Section 2) with a public value of 1. Since in finite field DH the result is calculated as $DHResult = pvr^{svi}$ on the Initiators side and $DHResult = pvi^{svi}$ on the Responders side, a received public value of 1 always leads to $DHResult = 1$. This effectively breaks encryption, as the encryption key is now known by the attacker. For ECDH, a public value of 1 also must not be accepted as defined in the standard.

Test: The public key of all *DHPart* messages (sent and received) is set to '`1`'.

Expected Results: The ZRTP key agreement must be aborted upon receipt of a *DHPart* with a public value of 1. This means that at least an indicator should display the insecure connection, even better, the connection should be terminated (cf. [termError]).

[invSS] Invalid Shared Secret (Active MitM): Invalid shared secrets *rs1* / *rs2* are used in the following scenarios: 1) The client cache got corrupted and the shared secret is now invalid or deleted. 2) A shared secret has been established previously and the endpoints operate in Preshared mode. Then, a MitM impersonates one endpoint and tries to establish a connection with a wrong shared secret. The ZRTP authors consider this to be a more critical event in comparison to a call made with a new ZID (cf. [zrtpCall]). Thus, according to RFC 6189 [37], a cache mismatch must result in a security dialog explicitly stating that the SAS needs to be compared again indicating a higher risk of a MitM attack.

Test: The shared secret in the ZID cache is replaced with repeated '`0xDEADBEEF`'.

Expected Results: The user must be notified via a security dialog and the SAS comparison needs to be executed again.

[invCom] Invalid Commit (Active MitM): This simulates an active MitM attack, where the endpoints operate in DH mode. In case of a commit clash, that is both parties tried to commit, the higher *hvi* wins. *hvi* most certainly does not match the corresponding public key. This means the participant sending the commit is impersonated by a MitM who does not want to commit to a public key. This would allow him to receive the other public key and repeatedly generate a public key so that it results in a SAS collision. This is feasible since the SAS consists of 16 bit for a B256 SAS and 20 bit for a B32 SAS. Then Alice and Bob would end up with the same SAS, even though a MitM attack has been executed. It is essential to verify that the revealed public key actually corresponds to the *Commit*.

Test: In the *Commit* the *hvi* is set to repeated '`0xFFFFFFFF`'.

Expected Results: The ZRTP key agreement must be aborted when receiving such a commit.

[sharedMitM] Shared MitM (Active MitM): A third person Eve conducts a normal ZRTP-protected call with Alice and one with Bob at different times. All participants verify the SASs. Now, Alice as well as Bob have shared secrets associated to Eve's ZID in their cache. Eve can now act as an active MitM by announcing her own ZID instead of forwarding that of Alice and Bob. Because ZIDs are not associated to a participant's SIP address, this attack stays undetected. The RFC proposes to implement labeling of ZIDs (cf. Section 2). This would allow the detection of such attacks if the participants compare the SIP address with the displayed ZID label. It is important to note that SAS verification does not imply that someone trusts a conversation partner, e.g., Trump does not necessarily trust Clinton, but they will likely want to verify their SASs. Thus, it is valid to assume that users will establish shared secrets with persons outside their social community.
Test: An attacker conducts normal calls with Alice and Bob. Then she acts as a MitM and modifies the key agreement by sending her own ZID instead of forwarding Bob's and Alice's ZID.
Expected Results: Either a) ZID labeling is implemented, b) Preshared mode is not supported, or c) non-standard association between SIP address and ZIDs is available.

[pbxEnroll] PBX Enrollment: This test verifies that the client either provides an appropriate user interface for accepting PBX when *Confirm* is received with the E flag or that the PBX enrollment is not supported. In case this feature is unsupported, the client must not handle *Hello* with the M flag and *SASRelay* messages in any way.
Test: 1) *Confirm* with the E flag is sent to the client. 2) *Hello* with the M flag is sent. 3) *SASRelay* is sent.
Expected Results: Abort ZRTP key agreement if not supported or show proper UI for PBX enrollment.

4.3 Non-Protocol Test Cases

Besides verifying that ZRTP clients are compliant with RFC 6189 [37], we evaluated an additional set of requirements not specified in the RFC. These tests focus on usability, error handling, and defaults. A general proposal how to implement security indicators, SAS verification, and error handling is given in our 'Best Practices' in Section 6.

[statusInd] Clear Status Indicators: If a client supports different levels of security, these should be communicated via indicators to the user. According to "Rethinking Connection Security Indicators" [12], a clear status indicator should be a combination of an icon and corresponding text. As presented in their study, not all participants understood the presented icons in their intended way. Furthermore, displaying only icons means that visually impaired users, such as red-green or total color-blind individuals, must rely on the icon's shape. Thus, the used icons and texts must be clearly distinguishable from each other to separate the security levels.
Expected Results: Security indicators distinguishing the provided security levels with icon and text.

[confSAS] Explicit Confirmation of SAS: To encourage the user to perform the SAS comparison, we expect a dialog that explicitly asks for confirmation of the shared SAS. It should explain the process with at least a way to confirm the SAS with a button.
Expected Results: Confirmation dialog with button to confirm SAS.

[termError] Terminate on Protocol Error: In case of a ZRTP protocol failure, such as [weakDH] or [invCom], the call should not fallback to an insecure connection. It can be assumed that either the server is trying to intercept the connection or that the participant's client is severely broken. It has been shown that users prefer functionality over security and will use an insecure fallback mode if provided. For example, studies analyzing the effectiveness of SSL warnings showed that non-experts clicked through warnings with probability over 30% to be able to still access the website [1, 33]. In our scenario, we suspect that users will not terminate a call manually and instead continue a conversation. Also, when talking on the phone, users usually do not pay attention to the on-screen state and will miss the insecure fallback indicator. Thus, the connection should be terminated automatically.
Expected Results: On protocol error, terminate the connection automatically.

[secDef] Secure Defaults: For better usability of the client's security features, secure defaults should be provided. If the client is targeted at a specific service, this can easily be done by enabling all security features by default. If a client supports different SIP providers, a setup procedure should allow the selection of the service and then enable security features accordingly.
Expected Results: Provide secure defaults for VoIP providers.

Table 2. Evaluation results for the most common ZRTP-capable VoIP clients using our protocol and non-protocol tests (ascending alphabetical order by name).

Application	OS	Version	Library	Protocol Tests							Non-Protocol Tests			
				[zrtpCall]	[verDown]	[weakDH]	[invSS]	[invCom]	[sharedMitM]	[pbxEnroll]	[statusInd]	[confSAS]	[termError]	[secDef]
Acrobits Softphone	iOS	5.8.1	-	●	●	●	●	●	●	–	○	●	○	○
CSipSimple	Android	1.02.03	ZRTP4PJ	●	●	●	○	●	○	–	◐	●	○	●
Jitsi	Win, Lin, MacOS	2.9.0	ZRTP4J	◐	●	●	●	●	○	–	●	●	○	●
Linphone Android	Android	3.1.1	bzrtp	●	●	●	○	○[a]	○	–	○	●	○	○
Signal	Android	3.15.2	-	●	–	●	–[b]	●	–[b]	–	●	○	●	●
Signal	iOS	2.6.4	-	●	–	●	–[b]	●	–[b]	–	●	○	◐	●

● = pass, ◐ = partially, ○ = fail, – = not supported

[a] CVE-2016-6271

[b] Signal is a *cacheless implementation*. It does not support Preshared mode.

5 Evaluation

For our evaluation, we selected common ZRTP-capable VoIP clients. Our selection criteria was as follows:

1. We must be able to execute our test cases:
   - If the client supports federated SIP and does **not operate in a closed network**, we can test it by conducting a call to a special Jitsi client that has been modified to execute our test cases against the calling client.
   - If the client is **operating in a closed network**, we need to implement our test cases directly into this client. Thus, the client's **source code must be available**.
2. The client should be relevant:
   - The client should be **actively used**, i.e., with a user base of at least 100,000 installations.
   - The implementation should be under **active development**, i.e., new versions have been released in 2016.

ZRTP + SIP: While ZRTP can be deployed independently of the signaling protocol, RFC 6189 [37] mainly focuses on its usage with SIP. The developer collective *Guardian Project* provides the Open Secure Telephony Network (OSTN) specification, a de-facto standard to deploy secure federated VoIP services based on SIP [13]. Using their testbed, we analyzed the following SIP clients:

Acrobits Softphone Closed source but standard-conform SIP client with ZRTP extension for iOS. Recommended on the OSTN website.

CSipSimple Free Software for Android with an active user base of over 1M users. Several proprietary forks have been released on Google Play.

Jitsi Open Source software for desktop operating systems that is actively developed with at least one new git tag per month.

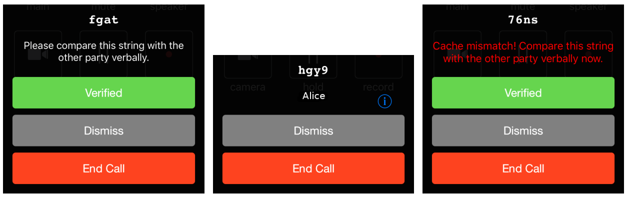
Linphone Free Software for Android with an active user base of over 100,000 users.

We excluded clients that are no longer available, such as Zfone (abandoned since 2011-01-29), Qutecom (abandoned since 2015-03), and SFLPhone (successor app named *Ring* no longer uses ZRTP). We excluded PrivateWave because we were not able to execute our test cases due to its operation in a closed network and because no source code is available. Silent Phone has been excluded because it operates in a closed network and we were not able to compile its source code[3], which is available on GitHub.

ZRTP + XMPP: Besides SIP, ZRTP can also be integrated with the Extensible Messaging and Presence Protocol (XMPP). XMPP has been extended by Jingle in XEP-166/XEP-167 [29, 30] to support media sessions between peers, primarily used for voice communication. For end-to-end security, ZRTP has been standardized as an extension in XEP-0262 [27]. To the best of the authors' knowledge, only Jitsi supports ZRTP over Jingle. Other XMPP clients, such as Empathy and Pidgin, do not support XEP-0262.

---

[3] https://github.com/SilentCircle/silent-phone-android/issues/11


fgat

Please compare this string with the other party verbally.

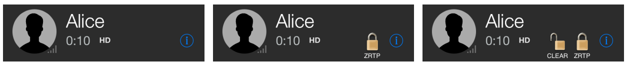
hgy9

76ns

Cache mismatch! Compare this string with the other party verbally now.

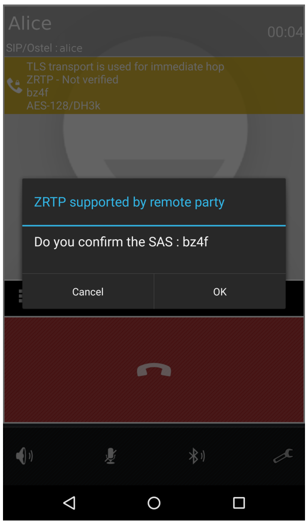
(a) [zrtpCall] in DH mode **(b)** [zrtpCall] in Preshared mode with ZID label **(c)** Warning message for [invSS]

Fig. 4. Acrobits Softphone: Dialog containing instructions, SAS verification, and ZID label



(a) no security **(b)** full ZRTP **(c)** One ZRTP-protected and one unprotected stream

Fig. 5. Acrobits Softphone: States of security indicators

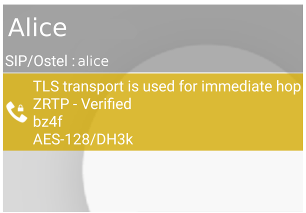


TLS transport is used for immediate hop
ZRTP - Not verified
bz4f
AES-128/DH3k

ZRTP supported by remote party

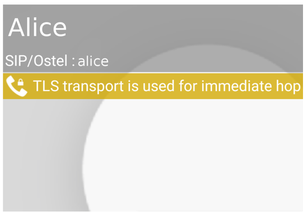
Do you confirm the SAS : bz4f

(a) SAS verification UI shown for [zrtpCall] in DH mode, no warning for [invSS]



TLS transport is used for immediate hop
ZRTP - Verified
bz4f
AES-128/DH3k

(b) Security indicator for [zrtpCall] in Preshared mode



TLS transport is used for immediate hop

(c) Security indicator for non-ZRTP connections

Fig. 6. CSipSimple: UI of interesting test cases

ZRTP + HTTP: Due to design objectives, such as high asynchronicity in mobile scenarios, Open Whisper Systems decided to design their own minimal signaling protocol with a RESTful HTTP API [21]. This has been deployed in conjunction with ZRTP in their messaging and VoIP app *Signal*. While Signal does not support federation, we were able to implement our test cases based on their source code for Android and iOS and thus selected it for our evaluation:

Signal Free Software for Android and iOS with an active user base of over 1M users.

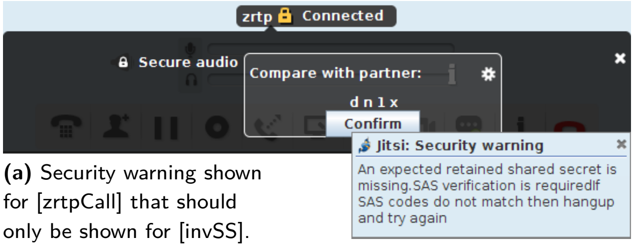
Our results are summarized in Table 2. In the following we discuss them in detail for each app individually.

5.1 Acrobits Softphone

As shown in Figure 4, Acrobits Softphone behaved perfectly in all protocol tests. It is the only implementation that implemented the labeling of ZIDs and thus provides protection against [sharedMitM]. For [verDown], [weakDH], [invSS], and [invCom] the key agreement was aborted falling back to non-ZRTP. Unfortunately, the connection was not terminated ([termError]). It lacks a proper wizard to setup an account with the OSTN, thus users are required to configure a secure server connection by themselves ([secDef]). What negatively stood out in our analysis is the use of security indicators. As shown in Figure 5, there is no icon for insecure connections making it difficult to assess the current security for an end user. Furthermore, we were able to establish connections where two indicators were displayed: Unfortunately, it was not clear for us what these meant just by looking at them. Acrobit's support explained that the icon labeled with 'CLEAR' indicates an insecure video stream besides the ZRTP-protected voice channel.

5.2 CSipSimple

No serious protocol issues have been encountered when testing for [verDown], [weakDH], and [invCom]. As expected for [confSAS], a SAS confirmation dialog is shown (cf. Figure 6a). No warning message is shown for [invSS], thus this test does not pass. Also, there is no way to label ZIDs, thus [sharedMitM] cannot be detected. Regarding [statusInd]: Detailed security information is displayed (cf. Figure 6b): The underlying network layer (TLS), ZRTP verification status, the SAS, the block cipher algorithm (AES 128), and the key agreement type (finite field DH with 3072 bit modular exponentiation group). Users with a security background will be well informed, but other users will probably mistake the small lock icon in the insecure fallback for guaranteed confidentiality (cf. Figure 6c). As seen in Figure 6c, CSipSimple does not terminate the call on errors but falls back to non-ZRTP ([termError]). This behavior can easily go unnoticed by users, especially because the different statuses are not differentiated by icon or color.



An expected retained shared secret is missing.SAS verification is requiredIf SAS codes do not match then hangup and try again

(a) Security warning shown for [zrtpCall] that should only be shown for [invSS].

zrtp Connected
Connected

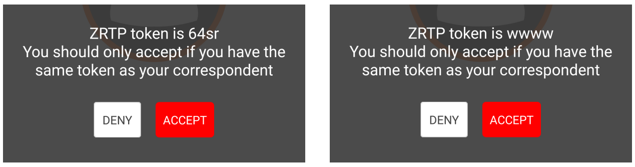
(b) Security indicator for [zrtpCall] in Preshared mode after SAS confirmation

(c) Security indicator for non-ZRTP connections

Fig. 7. Jitsi: UI of interesting test cases

5.3 Jitsi

No protocol issues have been encountered when testing for [verDown], [weakDH], [invSS], and [invCom]. ZID labeling is not implemented, thus [sharedMitM] cannot be detected. Two strong visual cues are used to convey the security status (cf. Figure 7b/7c): an closed or opened lock ([statusInd]). Additionally, the states between verified and unverified SAS are differentiated by green and yellow. The corresponding text says “zrtp Connected” for ZRTP connections or “Connected” for other connections. While this can still be improved as described in Section 6, Jitsi provides the best representation for end users compared to other analyzed SIP-based clients. The warning message for [invSS] is a little bit misleading, but the client still responds correctly (cf. Figure 7a). However this warning is also shown for [zrtpCall] after two other calls have been made. We analyzed this problem in detail and fixed the issue: Instead of generating a completely new ZID entry in-memory, the last read entry from the ZID cache was used for a call with a new participant. This happened because a variable was not resetted properly. As seen in Figure 7c, Jitsi does not terminate the call on errors but falls back to non-ZRTP ([termError]).

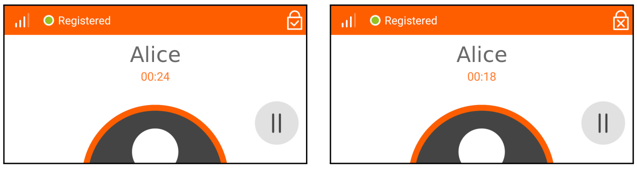


ZRTP token is 64sr
You should only accept if you have the same token as your correspondent

ZRTP token is wwww
You should only accept if you have the same token as your correspondent

(a) SAS verification UI shown for [zrtpCall] in DH mode

(b) Succeeded SAS attack



(c) Security indicator for [zrtpCall]

(d) Security indicator for non-ZRTP connections

Fig. 8. Linphone: UI of interesting test cases

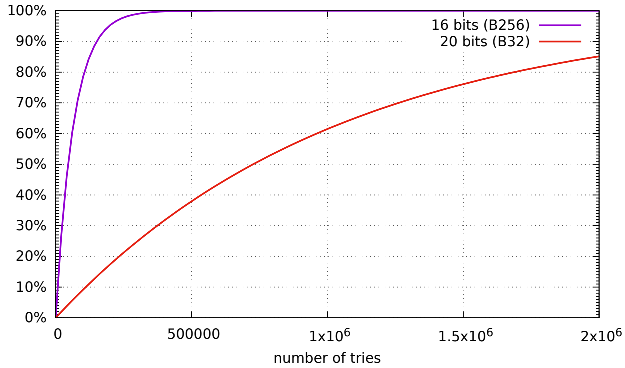


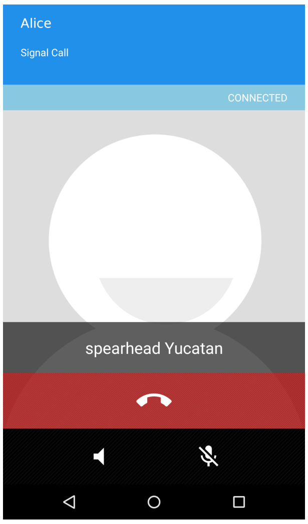
Fig. 9. Linphone: Probability of hitting a targeted SAS when exploiting CVE-2016-6271

5.4 Linphone Android

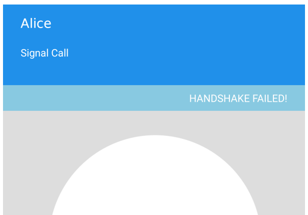
[verDown] and [weakDH] succeeded as expected. [invSS] did not pass as no warning message is shown here. ZID labeling is not implemented, thus [sharedMitM] cannot be detected. The test case [invCom] ended *fatal*: We uncovered a critical security vulnerability that gives an attacker full control over the displayed SAS. We implemented a fully working exploit using a patched Jitsi client that simulates a MitM. There are two variants to this attack, when taking the perspective of an active MitM:

a) Only one client is vulnerable: The SAS of the other client is random and the attacker forces the newly generated SAS to collide with the already established SAS. If b is the number of bits in the SAS, finding a collision after k trials is a Bernoulli experiment and the probability is $1 - (\frac{2^b-1}{2^b})^k$, where b is the number of bits (cf. Figure 9).

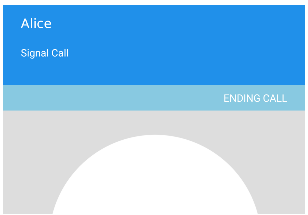
b) Both clients are vulnerable: The search for an SAS collision becomes a lot easier. The attacker is not required to collide with one certain SAS, but any SAS that can be forced on the other side suffices. Note however, that this is not a *birthday attack*. In theory, the attacker could hit a SAS twice on one side without reaching a collision between both sides. There is no limit to when a collision is inevitable, as there is when performing a birthday attack.



(a) Only DH mode is supported for [zrtpCall].

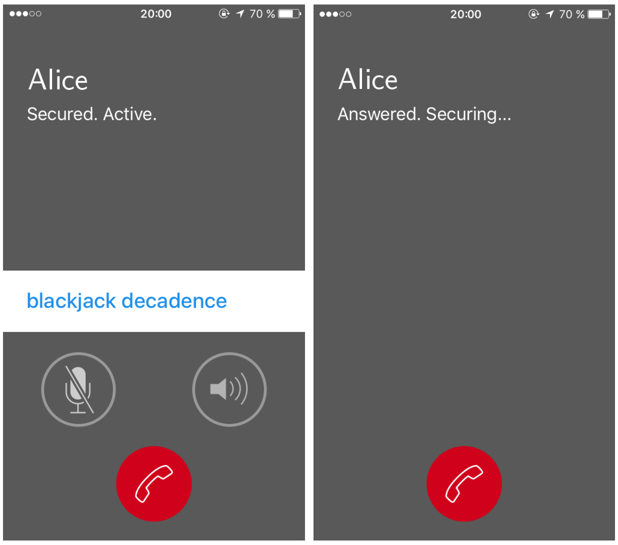


(b) Call termination on [invCom]



(c) Call termination on [weakDH]

Fig. 10. Signal on Android: UI of interesting test cases



(a) [zrtpCall]

(b) Unsuccessfull call termination for [weakDH], [invCom]

Fig. 11. Signal on iOS: UI of interesting test cases

As we simulate a MitM without actually having two targets, our attack tackles variant a), the more challenging case. In Figure 8b, Linphone was exploited to display a SAS of 4 matching digits. Imagine the SAS being a 4-letter word (like 'fake' or 'okay'), then the SAS blends in with the message and can alter its meaning. We responsibly disclosed this vulnerability on 07/05/2016 to Belledonne Communications and got CVE-2016-6271 assigned.

The UI for SAS verification is implemented properly as seen in Figure 8a. A small indicator in the top right displays the security status (cf. Figure 8c). Unfortunately, the insecure fallback mode is very difficult to detect, as seen in Figure 8d. Instead of a check mark, a 'X' is displayed inside the lock placed in the top right corner. We consider this as insufficient for [statusInd] to pass. More clearer icons combined with texts and strong colors displayed in a focus area are required.

5.5 Signal Android

No protocol issues have been encountered when testing for [weakDH] and [invCom]. Figure 10a shows Signal's call screen without errors. Notice that in the [weakDH] case in Figure 10b the call is ended directly while in the [invCom] case in Figure 10c the error message "Handshake failed!" is displayed on screen before ending the call. Signal ignores the version field, because it uses a closed network, where Signal clients can only communicate among each other. Thus, even when set to 'NOPE', nothing happens ([verDown]). Because Signal is a *cacheless implementation* [37], the [invSS] and [sharedMitM] tests are ignored.

In Signal, no security indicators are displayed ([statusInd]). We interpret this positively, because only ZRTP calls are supported in its closed network and thus no indicators to differentiate between secure and non-secure calls are required. Due to the fact that the verification status of communication partners cannot be stored for future calls, Signal does not pass [confSAS].

5.6 Signal iOS

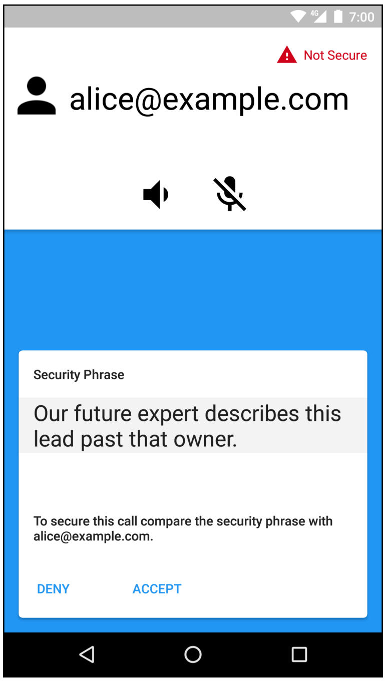
Signal on iOS behaves similar to the Android implementation. A simple [zrtpCall] is shown in Figure 11a. Its behaviour only differs for [termError]: While the Android client successfully terminates the connection, the iOS client hangs at the screen indicating that the key agreement is still in progress, as shown in Figure 11b.

6 Best Practices

As shown in our analysis in Section 5, most apps comply with the protocol, but greatly differ regarding their SAS verification UI and use of security indicators. Other publications focusing on voice forgability of SAS found that users often did not detect forged voices or dismiss security warnings [23, 31, 32]. To improve the user ex-

Table 3. Our proposal for security indicators and actions corresponding to specific ZRTP states. For federated SIP clients we propose *Base Configuration*. A configuration for *High-Sensitive Communication* is proposed for scenarios with higher security requirements. Our extension adds an additional error state in *SIP-ZID Binding*.

ZRTP State	Indicator	Action
	Base Configuration	
SAS Verified	🔒 Secure	-
SAS Unverified	ⓘ Not Secure	SAS Verification
No ZRTP	ⓘ Not Secure	-
Errors, e.g., [invCom]	-	Terminate
Error: [invSS]	⚠ Not Secure	SAS Verification+Warn
	High-Sensitive Communication	
SAS Unverified	⚠ Not Secure	-
No ZRTP	⚠ Not Secure	-
	SIP-ZID Binding	
Error: Mismatch	⚠ Not Secure	SAS Verification+Warn



Security Phrase

Our future expert describes this lead past that owner.

To secure this call compare the security phrase with alice@example.com.

Fig. 12. Our proposed SAS verification UI for high-sensitive communication. The security indicator is chosen in accordance with Table 3. Instead of four characters (B32) or two words (B256), we propose the usage of sentences generated from the SAS hash. The buttons are design in a neutral way to prevent priming the end user for a specific choice.

perience, better convey the current security status, and assist end users' decision making during the SAS verification, we propose a set of improvements for ZRTP clients. These improvements are based on results from other publications and justified individually.

The following design criteria are primarily written for ZRTP client developers. The SIP-ZID binding and SAS SENTENCE encoding could be published as an IETF Internet Draft if accepted by the community.

Sentences for SAS Verification: We propose a SENTENCE encoding [2] that uses the leftmost 50-90 bit from *sashash* to deterministically generate sentences as depicted in Figure 12. It has been shown that deviations in sentences are more easily detected [7]. This also protects against forged SAS as synthesized sentences can be better distinguished from human-spoken sentences [22]. While single words are spoken separately, the tone of words as part of the sentence depend on each other. Single words can easily be synthesized into voice samples for a specific victim and then stored in a lookup table for the actual attack (size: $256 + 256$). In comparison, the cost to synthesize all possible sentences is too high (2^{50} to 2^{90} depending on the algorithm).

SAS Verification UI: The comparison of SAS should be provided in a way that provides clear guidance for users and does not habituate users to always accept SAS. Thus, our card-like design in Figure 12 provides guidelines and buttons that are tinted with a neutral color to prevent users from automatically clicking 'accept'. In our future work, this UI definitely needs to be evaluated in larger user study.

Security Indicators: For apps providing ZRTP alongside insecure communication, non-ambiguous security indicators should be implemented. As depicted in Table 3, we propose the usage of 3 different indicators based on the recommendations by Porter Felt et al. [12] and adapted for ZRTP. For federated SIP clients we propose the usage of grey 'Not Secure' indicators for non-ZRTP calls that happen quite often. For clients configured for *High-Sensitive Communication*, these should be replaced with red indicators in addition to a warning icon (cf. Table 3).

Terminate on Error: ZRTP errors should lead to call termination (in contrast to insecure fallback as in Acrobits Softphone, CSipSimple, Jitsi, and Linphone). Error messages should be displayed in full screen for a short time to be recognized, not inside the call screen (like in Signal).

Warning Message for [invSS]: Together with the red security indicator in Table 3, we propose the following warning message for [invSS] in red: "The security phrase of alice@example.com changed. To verify that no one is wiretapping the conversation, compare the new security phrase with alice@example.com."

Shared Secret Cache: We propose to implement a cache for shared secrets and not just use DH mode for every connection. End users should be annoyed very rarely by SAS verification to not get habituated to clicking 'accept'.

6.1 Extension: SIP-ZID Binding

As discussed in Section 2, RFC 6189 [37] proposes to let the user write a label for each encountered ZID to describe its usage, such as "Alice on her office phone". This has been implemented in Acrobits Softphone and protects against [sharedMitM] attacks. Without redesigning the whole protocol, we propose to use SIP addresses as ZID labels without requiring user input. In this way the [sharedMitM] attack can be detected automatically.

Because the same ZID can be used for many SIP accounts, a mismatch can happen. Then, the following warning message should be shown: "Your participant uses a new address. To secure this call compare the security phrase again." After the SAS is verified again, the new SIP address should be added to this ZID. Thus, our extension requires that the client-side cache allows to a save a set of labels (SIP addresses) associated to each ZID. Conclusively, the extension provides a way to pin SIP addresses to specific ZIDs.

7 Related Work

ZRTP has been verified formally by Bresciani et al. [4–6]. It was shown that it is a secure key agreement protocol under the Dolev-Yao model. For their verification, the authors assume the SAS comparison to be able to detect MitM attacks. They have not analyzed voice forgery and similiar attacks. Gupta and Smatikov discovered a flaw in one of the previous versions of ZRTP before its standardization [14]: ZIDs were not authenticated early enough in the protocol exchange. Because they are used to look up shared secrets from the cache, an attacker could spoof a known ZID to conduct a MitM attack. Petraschek et al. analyze theoretical and practical attacks against ZRTP [24]. On the one hand, they focus on bypassing the SAS comparison by tampering with the audio signal once a MitM has been established. On the other hand, practical attacks are discussed, similar to ours. They show how to get into the media path for a MitM attack and analyze the behaviour of the—now abandoned—ZRTP client Zfone. They recognize that if an attacker uses a new ZID and spoofs the SIP address of the target, it is easily dismissed by inattentive users that this new session now longer uses shared secrets from the cache and instead should be verified again to detect the attacker. Recently, two theoretical attacks have been published by Bhargavan et al. [3]. They discuss a version downgrade attack as well as a downgrade from DH to preshared mode. To this day, uncovered attacks have not been systematized or applied for testing modern ZRTP clients.

Petraschek's attacks on SAS could be combined with recent work showing the feasibility of crowdsourcing voice imitation [23]. Similarly, imitating the voice of a participant to forge the SAS has been researched by Shirvanian et al. [31]. Two approaches were investigated: the *short voice reordering attack* takes prerecorded SAS strings of the target and uses them to forge the SAS, the *short voice morphing attack* generates arbitrary strings in the victim's voice given just a few minutes of eavesdropped sentences. The effectiveness is demonstrated by testing against manual detection as well as automatic detection. In the user study with 30 participants, about 50% of morphing attacks and 80% of reordering attacks were undetected. In a subsequent study with 128 online-participants, they found that for a two-word SAS, an attacker succeeds with about 30% probability [32]. This is due to human errors, such as failed speaker identification or wrong checksum comparison. While we do not provide an overview over the large amount of research in the area of voice synthesis, newer results, such as DeepMind's WaveNet [22], can drastically decrease the detectability of these attacks. All discussed results show the feasibility of replacing spoken voice with imitated recordings. In their SoK paper, Unger et al. compare the verification via SAS with others trust establishment approaches [34]. In particular, they classify SAS as not being *inattentive user resistant* because users are often required to manually end the call on failed verification.

Mechanisms to mitigate MitM attacks other than SAS that do not require the parties' active participation have also been proposed by Hlavacs et al. [15]. The first one assumes that Alice and Bob do not know each other (and therefore the attacker is not likely to know Bob either and can not predict that Alice is going to call Bob in the future), a situation in which the confidentiality is most threatened by attacks on SAS. Bob is obliged to send Alice the solution of computational puzzle within a small timeframe (10 seconds for example) that involves: A period of validity, Bob's URI, a temporary public key that is used to create a VPN. An improvement by associating ZIDs to SIP addresses is proposed by Petraschek et al. to protect against previously discussed voice forgery attacks [17]. A different idea has been investigated by Schürmann et al. by utilizing audio fingerprinting to replace the manual comparison of SAS [28]. This enables the use of devices without displays and hands-free equipment. However, to the best of the authors knowledge, no research specifically investigated implementation aspects in which security might go wrong in regard to ZRTP. This includes errors and UI weaknesses in real-world clients that secure their communications with ZRTP.

A lot of non-ZRTP-specific research exists related to the verification of public keys via fingerprints and hash commitments. In a 1000-participant large usability study, Dechand et al. evaluate fingerprint representations [7]. They recommend a sentence-based encoding, which achieves the highest attack detection rate and best usability perception. A hash commitment protocol with up to 10 peers is proposed by Farb et al. in SafeSlinger [11]. Their implementation includes interesting new UI concepts for verifying SAS by displaying radio buttons with three possible SAS choices where only one has been generated from the shared secret.

8 Ethics and Follow-Up

We hope that our findings contribute to the security of the VoIP ecosystem by having an impact on protocol designers, developers, and subsequently the end users.

We provide a MitM implementation to show wiretapping of unprotected VoIP calls. Our intention is not to harm end users, but to demonstrate the simplicity of interception software.

The security vulnerability CVE-2016-6271 in Linphone has been responsibly disclosed on 07/05/2016 to Belledonne Communications and fixed in Linphone 3.2.0[4]. We directly fixed the issue that a MitM warning is shown in Jitsi for normal calls due to erroneously reading the last entry from the ZID cache[5]. CSipSimple and Linphone did not implement a warning dialog for [invSS]. While RFC 6189 [37] requires this, it is not a fatal protocol error and its usefulness is limited. No tested client except Acrobits Softphone is protected against [sharedMitM]. RFC 6189 proposes ZID labeling to provide users a way to detect this attack. We suspect that the adoption of ZID labeling is hindered by its UI complexity. Here, we want to encourage a broader discussion how to prevent this attack automatically, e.g., by SIP-ZID binding.

We encountered different status indicators, which were not optimal and easily dismissed. Developers should follow our best practices and use indicators from Table 3. To prevent accidental insecure usage, clients should terminate on errors and provide secure defaults for SIP accounts. We hope that our proposed best practices encourages a discussion about usability and UI elements in the ZRTP developer community.

9 Conclusion

In this paper, we analyzed how VoIP calls can be wiretapped despite end-to-end protection by the ZRTP key exchange protocol. We motivated the importance of ZRTP by showing that active MitM attacks are easy to deploy for operators of VoIP infrastructure. As the main part of our research, we evaluated six of the most used open-source ZRTP clients available. We found one critical vulnerability (CVE-2016-6271) where Linphone on Android does not follow the standard and implements no verification of the hash commitments. This vulnerability allows successful wiretapping, even when comparing SASs. In Jitsi, a normal call was misinterpreted as an attack resulting in a false security warning. We also found that most implementations fall back to insecure non-ZRTP connections on ZRTP errors, which is hard to see for end users, who do not observe their screen when calling. This is made worse by bad UI practices, where security indicators are difficult to differentiate or not placed in central UI areas. Finally, we proposed best practices on how to overcome the deficiencies related to the way ZRTP has been integrated in VoIP user interfaces.

---

[4] https://github.com/BelledonneCommunications/bzrtp/commit/bbb1e6e2f467ee4bd7b9a8c800e4f07343d7d99b

[5] https://github.com/wernerd/ZRTP4J/pull/6
https://github.com/jitsi/jitsi/issues/303

References

[1] Devdatta Akhawe and Adrienne Porter Felt. Alice in warningland: A large-scale field study of browser security warning effectiveness. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*, pages 257–272, Washington, D.C., 2013. USENIX.

[2] akwizgran. basic-english. https://github.com/akwizgran/basic-english. (Accessed: 10/2016).

[3] K. Bhargavan, C. Brzuska, C. Fournet, M. Green, M. Kohlweiss, and S. Zanella-Béguelin. Downgrade resilience in key-exchange protocols. In *IEEE Symposium on Security and Privacy (SP)*, pages 506–525, May 2016.

[4] R. Bresciani and A. Butterfield. A formal security proof for the ZRTP protocol. In *International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 1–6, Nov 2009.

[5] Riccardo Bresciani. The ZRTP protocol analysis on the diffie-hellman mode. *Computer Science Department Technical Report TCD-CS-2009-13, Trinity College Dublin*, 2009.

[6] Riccardo Bresciani and Andrew Butterfield. ProVerif analysis of the ZRTP protocol. *International Journal for Infonomics (IJI)*, 3(3), 2010.

[7] Sergej Dechand, Dominik Schürmann, Karoline Busse, Yasemin Acar, Sascha Fahl, and Matthew Smith. An Empirical Study of Textual Key-Fingerprint Representations. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 193–208, Austin, TX, August 2016. USENIX.

[8] Werner Dittmann. ZRTP4PJ README. https://github.com/wernerd/ZRTP4PJ/tree/develop, 2015. (Accessed: 10/2016).

[9] Electronic Frontier Foundation. Secure Messaging Scorecard. https://www.eff.org/secure-messaging-scorecard, November 2014.

[10] Facebook. Messenger Secret Conversations. https://fbnewsroomus.files.wordpress.com/2016/07/secret_conversations_whitepaper-1.pdf, July 2016.

[11] Michael Farb, Yue-Hsun Lin, Tiffany Hyun-Jin Kim, Jonathan McCune, and Adrian Perrig. SafeSlinger: easy-to-use and secure public-key exchange. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 417–428. ACM, 2013.

[12] Adrienne Porter Felt, Robert W. Reeder, Alex Ainslie, Helen Harris, Max Walker, Christopher Thompson, Mustafa Embre Acer, Elisabeth Morant, and Sunny Consolvo. Rethinking Connection Security Indicators. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pages 1–14, Denver, CO, June 2016. USENIX.

[13] Guardianproject. Open {Secure,Source,Standards} Telephony Network (OSTN). https://dev.guardianproject.info/projects/ostn/wiki/Wiki, April 2016.

[14] Prateek Gupta and Vitaly Shmatikov. Security Analysis of Voice-over-IP Protocols. In *20th IEEE Computer Security Foundations Symposium (CSF 2007)*, pages 49–63, Venice, Italy, July 2007.

[15] Helmut Hlavacs, Wilfried Gansterer, Hannes Schabauer, Joachim, Martin Petraschek, Thomas Hoeher, and Oliver Jung. Enhancing ZRTP by using Computational Puzzles. *Journal of Universal Computer Science*, 14(5), 2008.

[16] IETF. SIP Working Group. https://datatracker.ietf.org/wg/sip/, July 2009.

[17] O. Jung, M. Petraschek, T. Hoeher, and I. Gojmerac. Using sip identity to prevent man-in-the-middle attacks on zrtp. In *2008 1st IFIP Wireless Days*, pages 1–5, November 2008.

[18] Patrick Juola. Isolated-Word Confusion Metrics and the PGPfone Alphabet. In *International Conference on New Methods in Natural Language Processing*, 1996.

[19] Patrick Juola. Whole-word phonetic distances and the PGPfone alphabet. In *Fourth International Conference on Spoken Language (ICSLP 96)*, volume 1, pages 98–101 vol.1, October 1996.

[20] Kamailio. Kamailio SIP-Server. https://www.kamailio.org. (Accessed: 10/2016).

[21] Moxie Marlinspike. Creating a low-latency calling network. https://whispersystems.org/blog/low-latency-switching/, January 2013.

[22] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[23] Saurabh Panjwani and Achintya Prakash. Crowdsourcing Attacks on Biometric Systems. In *Symposium On Usable Privacy and Security (SOUPS 2014)*, pages 257–269, Menlo Park, CA, July 2014. USENIX.

[24] Martin Petraschek, Thomas Hoeher, Oliver Jung, Helmut Hlavacs, and Wilfried Gansterer. Security and Usability Aspects of Man-in-the-Middle Attacks on ZRTP. *Journal of Universal Computer Science*, 14(5):673–692, 2008.

[25] Pew Research Center. The Smartphone Difference. http://www.pewinternet.org/2015/04/01/us-smartphone-use-in-2015/, April 2015.

[26] PjProject. PjProject. http://pjsip.org/, 2015. (Accessed: 10/2016).

[27] Peter Saint-Andre. Use of ZRTP in Jingle RTP Sessions. XEP-0262, June 2011.

[28] Dominik Schürmann and Stephan Sigg. Poster: Handsfree ZRTP - A Novel Key Agreement for RTP, Protected by Voice Commitments. In *Symposium On Usable Privacy and Security (SOUPS)*, July 2013.

[29] Joe Beda Peter Saint-Andre Robert McQueen Sean Egan Scott Ludwig and Joe Hildebrand. Jingle. XEP-0166, May 2016.

[30] Peter Saint-Andre Sean Egan Robert McQueen Scott Ludwig and Diana Cionoiu. Jingle RTP Sessions. XEP-0167, July 2016.

[31] Maliheh Shirvanian and Nitesh Saxena. Wiretapping via Mimicry: Short Voice Imitation Man-in-the-Middle Attacks on Crypto Phones. In *Proc. of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS 14)*, pages 868–879, New York, NY, USA, 2014. ACM.

[32] Maliheh Shirvanian and Nitesh Saxena. On the security and usability of crypto phones. In *Proceedings of the 31st Annual Computer Security Applications Conference*, ACSAC 2015, pages 21–30, New York, NY, USA, 2015. ACM.

[33] Joshua Sunshine, Serge Egelman, Hazim Almuhimedi, Neha Atri, and Lorrie Faith Cranor. Crying wolf: An empirical study of ssl warning effectiveness. In *Proceedings of the 18th Conference on USENIX Security Sympo-*

sium, SSYM'09, pages 399–416, Berkeley, CA, USA, 2009. USENIX.

[34] N. Unger, S. Dechand, J. Bonneau, S. Fahl, H. Perl, I. Goldberg, and M. Smith. SoK: Secure Messaging. In *IEEE Symposium on Security and Privacy*, pages 232–249, May 2015.

[35] WhatsApp. WhatsApp Encryption Overview. https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf, April 2016.

[36] Wikipedia. Commitment scheme. http://en.wikipedia.org/wiki/Commitment_scheme. (Accessed: 10/2016).

[37] P. Zimmermann, A. Johnston, and J. Callas. ZRTP: Media Path Key Agreement for Unicast Secure RTP. RFC 6189 (Informational), April 2011.

BANDANA – Body Area Network Device-to-device Authentication using Natural gAit

Dominik Schürmann*, Arne Brüsch*†, Stephan Sigg† and Lars Wolf*
*Institute of Operating Systems and Computer Networks, TU Braunschweig
†Ambient Intelligence, Comnet, Aalto University

***Abstract*—Secure spontaneous authentication between devices worn at arbitrary locations on the same body is a challenging, yet unsolved problem. We propose BANDANA, the first-ever implicit secure device-to-device authentication scheme for devices worn on the same body. Our approach leverages instantaneous variations in acceleration patterns from the user's gait to extract always-fresh secure secrets. It enables secure spontaneous pairing of devices worn on the same body or interacted with. The method is robust against noise in sensor readings and active attackers.**

I. INTRODUCTION

Device pairing mostly comprises one-time manual pairing of a limited number of devices. However, the personal device-network in the Internet of Things (IoT) is expected to experience frequent fluctuation in device count and identity as devices are added and discarded in the context of use [1]. While seamless device pairing without user interaction promises new personalized services, the user's privacy must be protected. This requires novel secure pairing schemes that scale.

We propose BANDANA, enabling convenient interaction-free secure pairing of devices conditioned to the context of use. As depicted in Figure 1, potential devices are any wearables, for instance, glasses, watches, smartphones, tablet computers or notebooks, smart textile, shoes or devices worn in bags or backpacks. In professional environments, further devices include helmets, Virtual Reality headsets and any co-used tools and wearables shared among workers. In addition, external devices such as a treadmill in a gym can be temporarily and spontaneously paired and BANDANA might be extended to pair with shopping carts, bicycles or cars.

BANDANA exploits common movement patterns to generate robust secure keys for pairs of devices worn at arbitrary locations on the same body. In contrast to previous work, proximity of devices on the body is not necessary as gait can be extracted at arbitrary body locations. The protocol is flexible in the strength of the generated key and can, for instance, replace Bluetooth PIN authentication with 24 seconds of gait while highly secure device pairing with 128 bit keys requires about 96 seconds of gait. We exploit instantaneous variations in gait sequences for implicit shared secrets among all devices on the same body. The contributions of our work are (A) a secure ad-hoc pairing scheme for devices worn on the same body, and (B) the experimental verification of the protocol on a large-scale gait dataset.

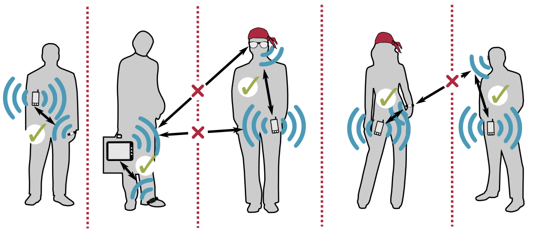


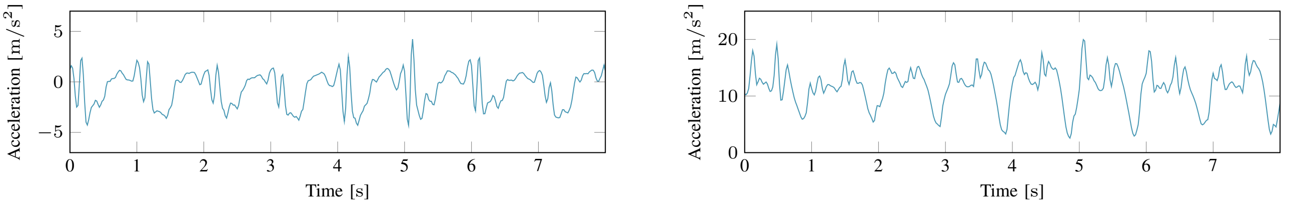
Fig. 1: BANDANA creates implicit security barriers towards devices in proximity, while establishing ad-hoc spontaneously secure connections between devices worn on the same body.

In a nutshell, a device (1) records acceleration sequences, (2) corrects their rotation error, (3) computes the mean gait from the previous gait cycles, and (4) generates a binary feature vector as the difference between this mean gait and the individual gait cycles. The feature vector reflects the pattern in which the mean gait exceeds or falls below the individual gait. Although individual and mean gait differ for various body locations, BANDANA exploits the correlation in the deviation from the mean. Utilizing fuzzy cryptography, device pairs are then able to (5) generate identical secret keys from similar binary fingerprints without disclosing any information about the fingerprints or keys on the wireless channel.

II. RELATED WORK

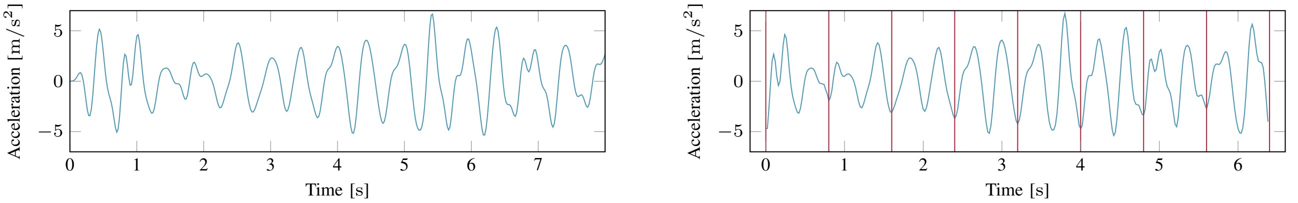
For authentication based on arbitrary co-aligned sensor data, Mayrhofer [2] proposes the candidate key protocol. It interactively exchanges hashes from feature sequences as short secrets and concatenates the key from the secrets with matching hashes. Based on this protocol, unlocking of a mobile device can be achieved by shaking it simultaneously with a smartwatch [3], [4]. Their approach, however, requires that acceleration sequences are exchanged and compared via an established secure channel and also that both devices are spatially close in order for acceleration sequences to be sufficiently similar. Sensor modalities suited for unattended co-presence-based device pairing extend to magnetometer [5], RF-signals [6], [7] luminosity [8] or audio [9]. In contrast to our study, however, these allow pairings not to the same body but only to devices in proximity.

Cornelius et al. [10] identified devices co-located on the same body via correlated acceleration readings. Even though after abstracting to the magnitude, the resulting signal still differed greatly due to inherently differing movement of underlying body parts (e.g. arm vs. head vs. legs) [11], the



(a) Unmodified accelerometer reading (z-axis) at 50 Hz.

(b) After Madgwick's algorithm. Gravity $g = \sim 9.81\,\mathrm{m/s^2}$ can now be recognized, indicating a correct orientation relative to the ground.



(c) Application of Type-II Chebyshev bandpass filter.

(d) Resampling to $\rho = 40$ and gait detection with $q = 8$ cycles.

Fig. 2: Pre-processing and gait cycle detection. Z-axis of an accelerometer attached to the forearm is depicted.

authors showed good correlation among all body locations from mean, standard deviation, variance, mean absolute deviation and interquartile range as well as signal's energy. This is a strong indication that secure keys conditioned on co-location on the same body exist. However, as correlation can be alternating positively and negatively, it remains unsolved how this can be exploited for the generation of keys, when the sequences shall not be disclosed to an adversary listening to any communication between nodes.

An activity well recognized over the whole body is walking [12]. For instance, identical step patterns from acceleration were utilized for co-location detection [13]. Hoang et al. [14] generated a key from the difference of a mean world gait (spanning the complete population) to the individual's mean gait. In this way, the authors assured that the resulting sequence is well balanced and uniformly distributed.

Recent studies on gait-based authentication, however, (1) do not address the impact of different on-body locations and sensor orientation and (2) use gait as a unique biometric feature that does not change for an individual over time. In contrast, we generate always-fresh keys from instantaneous accelerations for arbitrary locations on the human body.

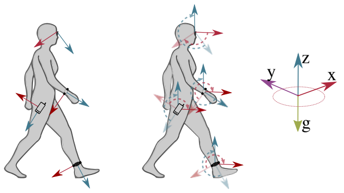
III. FUNDAMENTALS

In this section, our gait cycle detection algorithm is presented which builds on ideas by Hoang et al. [14], [15]. In addition, we also utilize gyroscope readings to normalize the sensor's orientation and keep only the z-Axis that points in the opposite direction of gravity. A gait cycle is defined as the "time interval between two successive steps" [16]. The algorithms input is a vector of amplitude values $\boldsymbol{z} = (z_1, \dots, z_n)$ of the accelerometer z-axis (cf. Figure 2a). Its output is a gait sequence of consecutive gait cycles with normalized length.

To find repetitive parts in the signal, we extract the local minima with similar distance to each other to define clearly separated cycles. Our filtering method is based on autocorrelation and distance calculation. The discrete autocorrelation at time lag k and with variance σ^2 is estimated as $Acorr(k) = \frac{1}{(n-k)\sigma^2}\sum_{t\in\mathbb{Z}} z_{t+k} \cdot \overline{z}_t$ where $\overline{z}_t$ represents the conjugate of z_t. The resulting autocorrelation $\boldsymbol{a} = (a_1, \dots, a_n)$ leads to m non-ambiguous local maxima in $\boldsymbol{a}$, stored as $\boldsymbol{\zeta} = \{\zeta_1, \dots, \zeta_i, \dots \zeta_m\}$. The distances between these indices and a mean distance $\delta_{mean} = \left\lceil \frac{\sum_{i=1}^{m-1} \zeta_{i+1} - \zeta_i}{m-1} \right\rceil$ are calculated. δ_{mean} defines the length of *half* a cycle, i.e., the time between the initial contact of the starting foot followed by the initial contact of the subsequent foot. Thus, for q describing the number of gait cycles, $m = q \cdot 2$. For the gait-cycle extraction, we assume healthy subjects, where the movement of the right foot is sufficiently similar to the left foot and thus have nearly the same distance. δ_{mean} can now be used to select indices of minima from $\boldsymbol{z}$ that represent clear cycles with the same length: $\boldsymbol{\mu} = \{\mu_1, \dots, \mu_i, \dots, \mu_{m-1}\}$; $\mu_i = \arg\min(z_{\zeta_i - \tau}, z_{\zeta_i - \tau + 1}, \dots, z_{\zeta_i + \delta_{mean} + \tau})$. Every μ_j represents the index of a minimum in $\boldsymbol{z}$ limited to the range of δ_{mean} where τ defines an additional user defined factor to account for small deviations in the gait duration. The indices in $\boldsymbol{\mu}$ can now be used to split the raw data $\boldsymbol{z}$ into full gait cycles $\boldsymbol{Z} = \{Z_1, \dots, Z_i, \dots, Z_q\}$; $Z_i = (z_{\mu_{\frac{i}{2}}}, \dots, z_{\mu_i}, \dots, z_{\mu_{\frac{i+1}{2}} - 1})$. Finally, the length of gait cycles are normalized by resampling every Z_i using a Fourier method to a fixed number of samples ρ per gait cycle so that $|Z_i| = \rho$ (cf. Figure 2d). For ease of presentation, we will, in the following, describe such normalized gait cycle with $Z_i = \{Z_{i1}, \dots, Z_{i\rho}\}$. The choice of ρ depends on factors such as sample rate and requirements of the quantization algorithm discussed in Section IV.

A. Dataset

We used the real-world dataset by Sztyler et al. [17] for position-aware activity recognition. 15 subjects performed different actions for approximately 10 - 12 minutes each. They



(a) Before (b) After Madgwick

Fig. 3: Effect of applying Madgwick's algorithm.

were equipped with 7 sensors on different body locations. These locations were chosen in order to gather data from every part of the body that behaves different during human motion.

B. Data Pre-Processing

In real-world settings, sensors locations differ, which introduces changing orientations due to body part movements (cf. Figure 3a). For best results, it is crucial to rotate every data point such that at all time one of the axes is facing in the direction opposite of gravity (cf. Figure 3b).

Nowadays, most mobile devices contain gyroscopes in addition to accelerometers [18]. We therefore posses information about the initial device orientation (since the force of gravity is included in every measurement recorded by the accelerometer) as well as the angular velocity of the sensor platform itself. Thus, it is possible to correct the ongoing orientation error. We employ the algorithm proposed by Madgwick et al. [19] to rotate all measurements z_i accordingly, resulting in a signal as shown in Figure 2b. Note that the output is only guaranteed to be aligned along the z-axis. When comparing two readings, both other axes may point in different directions as no other fixed direction as, e.g., the direction of North is obtainable.

For noise removal, we apply a Type II Chebyshev bandpass filter with passband chosen between $0.5\,\text{Hz}$ and $12\,\text{Hz}$ (cf. Section V-B). The resulting signal is shown in Figure 2c.

IV. BANDANA

After correcting orientations from accelerometer and gyroscope data together with applying a band-pass filter, the gait cycle detection algorithm produces a periodic signal. Shared secrets need to be generated based on these signals on different devices independently without disclosing them on the channel.

A. Quantization

To generate binary fingerprints from the continuous gait sequence, we propose a quantization algorithm inspired by Hoang et al. [14]. Recall the definition of Z_i with the normalized gait cycle $|Z_i| = \rho$ and $Z_i = \{Z_{i1}, \ldots, Z_{i\rho}\}$. We define the average gait cycle as $\boldsymbol{A} = (A_1, \ldots, A_j, \ldots A_\rho)$; $A_j = \frac{\sum_{i=1}^{q} Z_{ij}}{q}$. Fingerprint bits are extracted by calculating the energy difference between each gait cycle Z_i and $\boldsymbol{A}$ as depicted in Figure 4. To extract b bit per Z_i, each Z_i is split into b parts of the same length ρ/b. Thus, a binary fingerprint is defined by $\tilde{\boldsymbol{f}} = (\tilde{f}_{11}, \ldots, \tilde{f}_{1\frac{\rho}{b}}, \ldots, \tilde{f}_{b1}, \ldots, \tilde{f}_{b\frac{\rho}{b}})$; .

$$\tilde{f}_{ij} = \begin{cases} 1, & \delta_{ij} > 0 \\ 0, & \text{otherwise.} \end{cases}$$

as exemplary shown in Figure 4a. In the following, the fingerprint vector is written as $\tilde{\boldsymbol{f}} = (\tilde{f}_1, \ldots, \tilde{f}_M)$.

B. Reliability

To calculate the reliability of the extracted bits, the differences of the quantization algorithm are stored as $\boldsymbol{\delta} = (\delta_{11}, \ldots, \delta_{1b}, \ldots, \delta_{q1}, \ldots, \delta_{qb})$. The indices of $\boldsymbol{\delta}$ are sorted in descending order by their absolute value $|\delta_{ij}|$ to retrieve the reliability ordering $\boldsymbol{r} = (r_1, \ldots, r_M)$ with $r_i \geq r_{i+1}$. We refer to $\boldsymbol{r}$ as the *reliability vector* containing indices which experienced the highest difference between the mean gait $\boldsymbol{A}$ and an instantaneous normalized gait Z_j. These bits are most reliable since they have high probability to be identical at arbitrary body locations. In Figure 4b colors to indicate the associated reliability. The elements of $\tilde{\boldsymbol{f}}$ are then sorted according to their values of $\boldsymbol{r}$ and the most reliable first N are the fingerprint $\boldsymbol{f} = (f_{r_1}, \ldots, f_{r_N})$ (cf. Figure 4c).

C. Fuzzy Cryptography

To derive unique shared secrets on two devices without disclosing the fingerprint, error correcting codes are used, which encode messages from the messagespace $m \in \mathcal{M}$ into codewords of the (larger) codespace $c \in \mathcal{C}$ introducing redundancies. Then, errors from transmission of c over lossy channels are corrected before decoding back to m.

In a sense, our fingerprints $\boldsymbol{f}$ are lossy as they are not entirely equal on the devices trying to mutually authenticate. Here, the codespace $\mathcal{C}$ is chosen in a way that we can directly pick a fingerprint $\boldsymbol{f}$ from this codespace and apply the *Decode*-method to derive a binary key $\boldsymbol{k}$ that is error corrected. Due to the usage of binary fingerprints we propose the usage of BCH codes over the Galois field $\mathbb{F}_2$. A BCH code can be parameterized to correct up to t errors, which in our case must be chosen carefully to allow for errors within different locations on the same body but not for correction of errors between different bodies. As with the other parameters, t is chosen based on our evaluation in Section V.

D. Protocol

Figure 5 specifies the BANDANA protocol. For two coaligned devices A and B, fingerprints $\boldsymbol{f_A}$, $\boldsymbol{f_B}$ and reliability vectors $\boldsymbol{r_A}$, $\boldsymbol{r_B}$ are derived on both devices independently. The vector with the higher hash is used for reliability ordering on both sides. To account for errors, we apply the BCH decoding-method to reduce both $\boldsymbol{r_A}$ and $\boldsymbol{r_B}$ to a unique $\boldsymbol{k}$, which is then used as the password for a Password-Authenticated Key Agreement (PAKE). Both devices now share the same secret $\boldsymbol{s}$ protected by a key agreement authenticated by their gait fingerprints. We propose the usage of a modern non-patented PAKE that feature additional countermeasures for low entropy passwords, such as J-PAKE [20] or SRP [21].

For devices with high clock drift, the protocol can be extended to allow for multiple tries with shifted fingerprints.

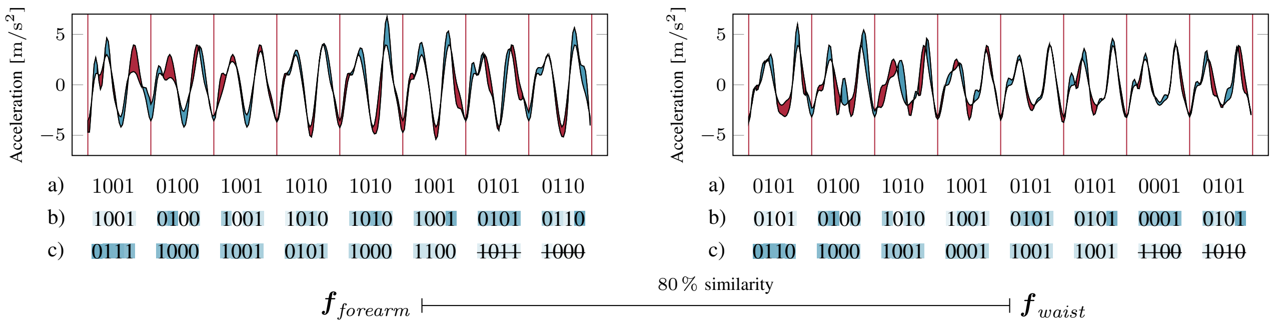


Fig. 4: Independent fingerprint generation on forearm and waist (forearm pre-processing is shown in Figure 2): Energy levels above the average gait cycle $\boldsymbol{A}$ are blue and below red. After quantization in a), reliabilities are calculated and assigned to each bit in b). Darker color, indicates higher reliability. In c) the fingerprint is sorted by reliability vector of the forearm.

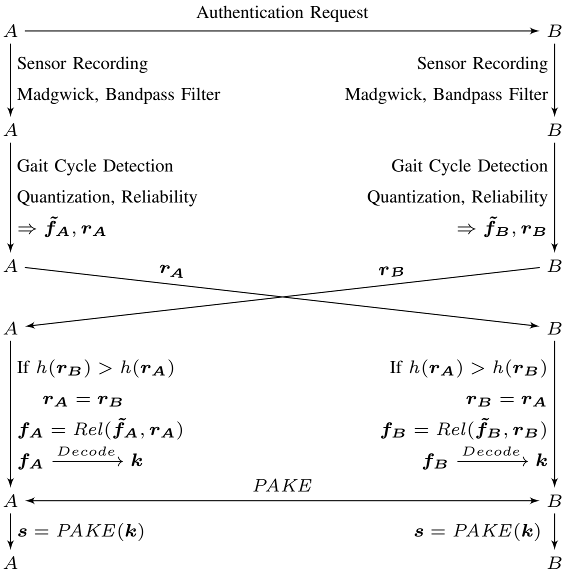

Authentication Request
A
B
Sensor Recording
Madgwick, Bandpass Filter
Sensor Recording
Madgwick, Bandpass Filter
Gait Cycle Detection
Quantization, Reliability
$\Rightarrow \tilde{f}_A, r_A$
Gait Cycle Detection
Quantization, Reliability
$\Rightarrow \tilde{f}_B, r_B$
r_A
r_B
If $h(r_B) > h(r_A)$
$r_A = r_B$
$f_A = Rel(\tilde{f}_A, r_A)$
$f_A \xrightarrow{Decode} k$
If $h(r_A) > h(r_B)$
$r_B = r_A$
$f_B = Rel(\tilde{f}_B, r_B)$
$f_B \xrightarrow{Decode} k$
PAKE
$s = PAKE(k)$
$s = PAKE(k)$

Fig. 5: BANDANA protocol sequence between two devices A and B worn on the same body.

V. EVALUATION

A. Signal Coherence

After applying Madgwick's algorithm (cf. Section III-B), we end up with sensor readings where the z-axis points to the ground. This allows to examine their relation. For this, we calculate the spectral coherence for different sensor combinations to test whether any causality between readings taken simultaneously by sensors located at different locations on the same body exists – apart from just the correlation for the motion in general. Figure 6 shows that there is high correlation between records taken simultaneously. Between arbitrary records, there is only correlation between 0 Hz up to 0.5 Hz. This leaves us with two major results: (a) There is a measurable causality between sensor readings taken simultaneously on the same body; (b) Some correlation at lower frequencies still exists.

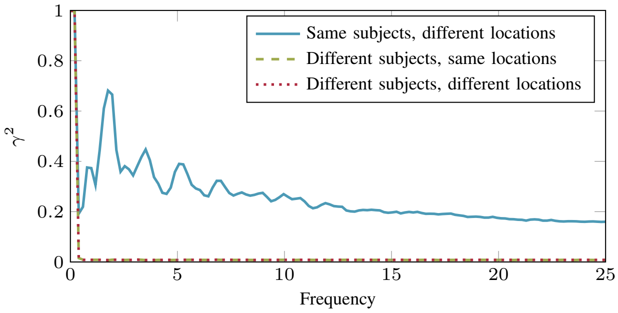


Fig. 6: Average spectral coherence over full sensor readings of the Mannheim dataset for same and different subject.

B. Bandpass Filter

As visualized in Figure 6, there still exists some unexpected correlation between arbitrary readings on low frequencies. As these frequencies - up to approximately 0.5 Hz - only add noise, we filter them out while keeping all the frequencies above. We thus employ a Type-II Chebyshev filter, which is known to have a very steep drop at the cutoff frequency. Furthermore, in contrast to Type-I, Type-II Chebyshev filters do not have any ripple in the passband. Researchers in the domain of Activity Recognition report that human motion does not affect frequencies significantly above 10 Hz [22]. Based on this observation and the coherence depicted in Figure 6, we decided to choose an upper cutoff frequency of 12 Hz.

C. Reliability

Our quantization scheme defines that iff $\delta_{ij} > 0$ for fixed i, j is true for A, the same has to apply for B for at least 80 %. Some Z_{ij} are less prone to leading to different bits between sensors at different body locations than others, namely those with a higher difference δ_{ij} to the mean gait $\boldsymbol{A}$. Both A and B keep a reliability value for each bit of the fingerprint. According to the protocol sequence (cf. Figure 5), one of these reliability vectors is chosen and the fingerprint is sorted by each party following the vector's order of indices (cf. Figure 4). In a last step, the fingerprint's most unreliable bits

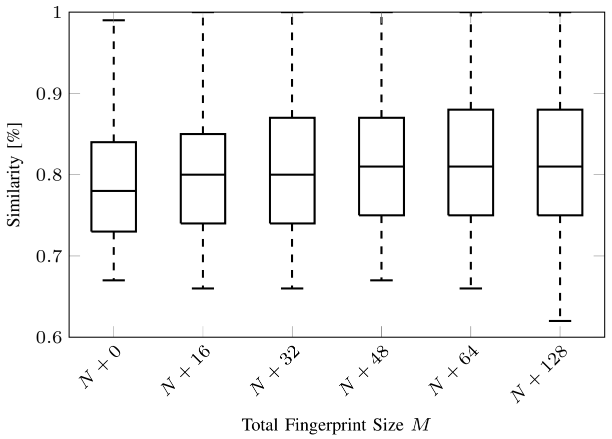


Fig. 7: Fingerprint similarity of different sizes M with cutoff at $N = 128$ to evaluate the influence of *Rel*(). Each boxplot value is defined by the similarity between two fingerprints at *different* sensor locations within the same subject (intra-body). All possible similarities over all combinations of sensor locations within each subject are evaluated. Fingerprints are generated by a sliding window over the sensor data with half-overlapping windows. Only fingerprints from the same window are matched against each other.

are discarded. To show the method's viability, we calculated the fingerprints' similarity over all 15 subjects and all 7 sensor locations. As shown in Figure 7, we chose different fingerprint sizes M with cutoff at $N = 128$ to test how many additional bits should be discarded to gain the best similarity. The mean-similarity improves with greater values of M and settles around $N+64$ with an average improvement of approximately $4\,\%$. Thus, we chose $N + 64$ for our configuration.

D. Discriminability of Intra- and Inter-body Fingerprints

Figure 8 illustrates the discriminability between intra-body and inter-body fingerprints. While the intra-body case tests only similarities between different sensor locations on the same body (315 similarities), the inter-body case is much larger (8880300 similarities). The mean similarity between different subjects is $50\,\%$, which is indistinguishable from a similarity between random bit sequences. In comparison, the inter-body similarity exhibits a clear security margin with $82\,\%$. It is important to note that this test evaluates the worst case of brute forcing all possible combinations between subjects. In reality, an attacker is constrained to ~ 900 tries per day since BANDANA's process takes up to $\sim 96\,\mathrm{s}$ with $M = 192$ bit long fingerprints. In the inter-body case, it can be seen that a small number of fingerprints match with unexpected high similarity values (outliers). We assume that these collisions happen in case of gait sequences with very low entropy still exhibiting specific pattern due to the design of the quantization scheme. While this should be investigated further, only $0.0642\,\%$ of these collisions show similarity values above $80\,\%$.

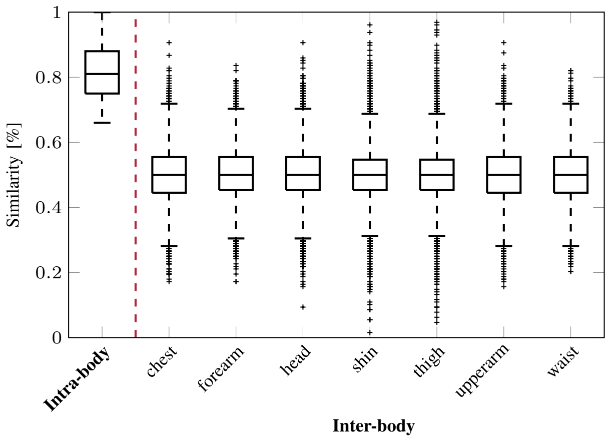


Fig. 8: Intra-body and inter-body fingerprint similarity. For *intra-body*, each boxplot value is defined by the similarity between two *different* sensor locations (all possible similarities over all combinations of sensor locations within each subject). For *inter-body*, each boxplot defines a different sensor location. Only *different* subjects are tested against each other with the *same* sensor locations. Fingerprints are generated by a sliding window over the sensor data with half-overlapping windows for $M = 192$ with cutoff at $N = 128$.

TABLE I: Fingerprint similarity between locations on the same body (intra-body). Shown is the mean over all 15 subjects.

	chest	forearm	head	shin	thigh	upperarm	waist
chest	1.0	0.82	0.74	0.78	0.78	0.88	0.81
forearm	0.82	1.0	0.8	0.81	0.88	0.89	0.89
head	0.74	0.8	1.0	0.8	0.76	0.77	0.78
shin	0.78	0.81	0.8	1.0	0.77	0.78	0.8
thigh	0.78	0.88	0.76	0.77	1.0	0.85	0.84
upperarm	0.88	0.89	0.77	0.78	0.85	1.0	0.88
waist	0.81	0.89	0.78	0.8	0.84	0.88	1.0

E. Similarities between Sensor Location-Combinations

Table I illustrates how well different sensor locations authenticate against each other. We found out that chest against other locations and head against other locations perform worse while forearm and waist perform best.

F. Statistical Bias

For the robustness against a potent adversary, it is important that the keys generated from gait sequences are random. For instance, Figure 10 exemplarily depicts 64 keys we extracted using BANDANA with fingerprint length $N = 256$ bits for an intuitive illustration of the randomness of the generated fingerprints. We tested the keys generated by BANDANA against statistical bias and employed the dieHarder battery of statistical tests for this end [23]. While these tests can not replace cryptanalysis, they are designed to uncover bias and

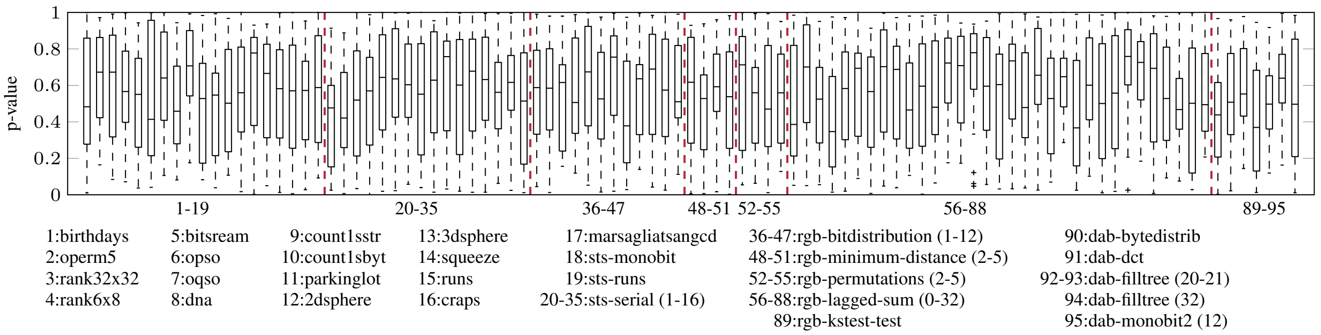


Fig. 9: Distribution of p-values achieved for 128 bit keys (fingerprint length $M = 192$, 64 unreliable bits removed) in 21 runs of the various statistical tests of the dieHarder set of statistical tests.

Fig. 10: Illustration of 64 binary keys. Each row contains one 256 bit fingerprint with $1 =$ black and $0 =$ white.

dependency in the pseudo random sequence. Every test has an expected distribution of outcomes. A p-value, between 0 and 1, describes the probability that a real Random Number Generator (RNG) would produce this outcome. A good RNG will have a range of p-values that follows a uniform distribution. A p-value below a significance level $\alpha = 0.001$ indicates a failure of the RNG with probability $1 - \alpha$. For instance, a p-value ≤ 0.05 is expected $5\,\%$ of the time. Our results in Figure 9 depict well distributed p-values clustered in the center which indicates a good random distribution.

VI. Conclusion

We have presented BANDANA, a secure device-to-device authentication scheme for devices worn on the same body. By generating unique fingerprints from the user's gait, we were able to establish shared secrets implicitly without user interaction. The protocol accounts for errors without comparing the fingerprints directly, instead it utilizes fuzzy cryptography based on error correcting codes. A novel quantization method for independently generating similar fingerprints at different sensor locations has been proposed and evaluated. By selecting only reliable fingerprint bits, we were able to boost the similarity by $4\,\%$. We evaluated the security by generating all possible fingerprints in our dataset for sensors worn on the same body (intra-body) in comparison to sensors worn on different bodies (inter-body). While intra-body similarity is indistinguishable from similarity between random bit sequences ($50\,\%$), inter-body similarity exhibits a clear security margin with $82\,\%$. Based on our evaluation, the final specification of BANDANA is depicted in Figure 11.

Parameters: We used a resampling rate of $\rho = 40$ to extract $b = 4$ bits per gait cycle R_i resulting in $\tau = \rho/b = 10$. For $N = 128$ bit keys we used $M = 192$ bit fingerprints ($q = 48$ gait cycles), disregarding 64 least reliable bits. Fuzzy pairing corrected at most $20\,\%$ (cf. Figure 8) dissimilar bits ($t = \lfloor 128 \cdot 0.2 \rfloor = 25$). Consequently, at least $80\,\%$ similarity between the fingerprints is required. This results in a 103-bit security level for the PAKE password $\boldsymbol{k}$.

Time to generate a secure key: The key-strength depends on the number of gait cycles. Our parameters $b = 4, \rho = 40, M = 192$ result in the worst-case duration of $\boldsymbol{r} = 96\,\mathrm{s}$ assuming that gait cycles do not exceed 2 seconds. Clearly, by extracting more bits from each cycle or requiring shorter key sequences, generation time can be reduced linearly.

Time after which secure key generation fails: After removal from the body, the gait-history is bit by bit replaced so that similarity in fingerprints gradually deteriorates from about $80\,\%$ to $50\,\%$ (cf. Figure 8). A fuzzy cryptography scheme requiring at least $75\,\%$ similarity (which is weaker than $80\,\%$ in our results), then fails after 9.6 gait cycles or 19.2 seconds ($0.8 \cdot 80\,\% + 0.2 \cdot 50\,\% \approx 74\,\%$).

Adaptive security levels: The key-length determines its strength. E.g. manual Bluetooth pairing (4-digit PIN) is equivalent to a 32 bit key, generated in 24 seconds ($b = 4, \rho = 40, M = 48$). Key generation after removing the device from the body would then fail after 5 seconds. An adaptive security protocol can alter security levels (and granted rights) conditioned on the co-presence duration.

Pairing in the absence of gait: For some activities, gait is not available. We did not consider this case in our study. The primary challenge in such a case is then to identify a feature recognizable from arbitrary locations on the same body, since else, device pairing is constrained to proximate body parts (e.g. in [24]).

Technical requirements: Devices should feature accelerometer and gyroscope. While instrumentations without gyroscope might also be feasible in some scenarios, continuous correction of accelerometer orientation works most reliable with gyroscope information. Given its low price and since most contemporary wearables with acceleration sensors also include a gyroscope, this is not a limitation.

Fig. 11: Technical specification and limitations of BANDANA.

References

[1] M. K. Chong, R. Mayrhofer, and H. Gellersen, “A survey of user interaction for spontaneous device association,” *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, p. 8, 2014.

[2] R. Mayrhofer, “The candidate key protocol for generating secret shared keys from similar sensor data streams,” in *European Workshop on Security in Ad-hoc and Sensor Networks*, Springer, 2007, pp. 1–15.

[3] R. D. Findling, M. Muaaz, D. Hintze, and R. Mayrhofer, “Shakeunlock: Securely unlock mobile devices by shaking them together,” in *Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia*, ACM, 2014, pp. 165–174.

[4] R. D. Findling, M. Muaaz, D. Hintze, and R. Mayrhofer, “ShakeUnlock: Securely Transfer Authentication States Between Mobile Devices,” *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2016.

[5] R. Jin, L. Shi, K. Zeng, A. Pande, and P. Mohapatra, “MagPairing: Pairing Smartphones in Close Proximity Using Magnetometers,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1306–1320, 2016.

[6] A. Varshavsky, A. Scannell, A. LaMarca, and E. De Lara, “Amigo: Proximity-based authentication of mobile devices,” in *UbiComp 2007: Ubiquitous Computing: 9th International Conference*. Berlin, Heidelberg: Springer, 2007, pp. 253–270.

[7] D. A. Knox and T. Kunz, “Wireless fingerprints inside a wireless sensor network,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 11, no. 2, p. 37, 2015.

[8] M. Miettinen, N Asokan, T. D. Nguyen, A.-R. Sadeghi, and M. Sobhani, “Context-based zero-interaction pairing and key evolution for advanced personal devices,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2014, pp. 880–891.

[9] D. Schürmann and S. Sigg, “Secure communication based on ambient audio,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 2, pp. 358–370, 2013.

[10] C. Cornelius and D. Kotz, “Recognizing whether sensors are on the same body,” in *Proceedings of the 9th International Conference on Pervasive Computing (Pervasive’11)*, San Francisco, USA: Springer-Verlag, 2011, pp. 332–349.

[11] E. A. Heinz, K. S. Kunze, S. Sulistyo, H. Junker, P. Lukowicz, and G. Tröster, “Experimental evaluation of variations in primary features used for accelerometric context recognition,” in *European Symposium on Ambient Intelligence*, Springer, 2003, pp. 252–263.

[12] M. Muaaz and R. Mayrhofer, “An analysis of different approaches to gait recognition using cell phone based accelerometers,” in *Proceedings of International Conference on Advances in Mobile Computing & Multimedia*, ACM, 2013, p. 293.

[13] A. Srivastava, J. Gummeson, M. Baker, and K.-H. Kim, “Step-by-step detection of personally collocated mobile devices,” in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, ACM, 2015, pp. 93–98.

[14] T. Hoang, D. Choi, and T. Nguyen, “Gait authentication on mobile phone using biometric cryptosystem and fuzzy commitment scheme,” *International Journal of Information Security*, vol. 14, no. 6, pp. 549–560, 2015.

[15] T. Hoang, D. Choi, V. Vo, A. Nguyen, and T. Nguyen, “A Lightweight Gait Authentication on Mobile Phone Regardless of Installation Error,” in *Security and Privacy Protection in Information Processing Systems: 28th IFIP TC 11 International Conference, SEC 2013, Auckland, New Zealand, July 8-10, 2013. Proceedings*, L. J. Janczewski, H. B. Wolfe, and S. Shenoi, Eds. Berlin, Heidelberg: Springer, 2013, pp. 83–101.

[16] M. W. Whittle, “Chapter 2 - Normal gait,” in *Gait Analysis (Fourth Edition)*, M. W. Whittle, Ed., Fourth Edition, Edinburgh: Butterworth-Heinemann, 2007, pp. 47–100.

[17] T. Sztyler and H. Stuckenschmidt, “On-body Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition,” in *IEEE International Conference on Pervasive Computing and Communications (PerCom’16)*, (Sydney, Australia, Mar. 14–18, 2016), IEEE Computer Society, 2016, pp. 1–9.

[18] GSMArena.com, *Phone finder results for accelerometer and gyrometer*, http://www.gsmarena.com/results.php3?chkAccelerometer=selected&chkGyro=selected (accessed on 09/2016), 2016.

[19] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, “Estimation of IMU and MARG orientation using a gradient descent algorithm,” in *2011 IEEE International Conference on Rehabilitation Robotics*, IEEE, 2011, pp. 1–7.

[20] F. Hao and P. Ryan, “J-PAKE: Authenticated Key Exchange without PKI,” in *Transactions on Computational Science XI: Special Issue on Security in Computing, Part II*, M. L. Gavrilova, C. J. K. Tan, and E. D. Moreno, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 192–206.

[21] T. D. Wu *et al.*, “The Secure Remote Password Protocol,” in *Network and Distributed System Security Symposium (NDSS’98)*, 1998, pp. 97–111.

[22] J. Lester, B. Hannaford, and G. Borriello, ““Are You with Me?”–Using Accelerometers to Determine If Two Devices Are Carried by the Same Person,” in *Pervasive Computing: Second International Conference (Pervasive’04)*. Berlin, Heidelberg: Springer, 2004, pp. 33–50.

[23] R. G. Brown, *Dieharder: A Random Number Test Suite*, http://www.phy.duke.edu/~rgb/General/dieharder.php, 2011.

[24] R. Mayrhofer and H. Gellersen, “Shake well before use: Authentication based on accelerometer data,” in *Pervasive computing*, Springer, 2007, pp. 144–161.

Moves like Jagger: Exploiting variations in instantaneous gait for spontaneous device pairing[☆]

Dominik Schürmann[a], Arne Brüsch[a], Ngu Nguyen[b], Stephan Sigg[b], Lars Wolf[a]

[a] *Connected and Mobile Systems, Institute of Operating Systems and Computer Networks,*
TU Braunschweig, Mühlenpfordtstr. 23, Braunschweig, Germany
[b] *Ambient Intelligence, Department of Communications and Networking,*
Aalto University, Maarintie 8, Espoo, Finland

Abstract

Seamless device pairing conditioned on the context of use fosters novel application domains and ease of use. Examples are automatic device pairings with objects interacted with, such as instrumented shopping baskets, electronic tourist guides (e.g. tablets), fitness trackers or other fitness equipment. We propose a cryptographically secure spontaneous authentication scheme, BANDANA, that exploits correlation in acceleration sequences from devices worn or carried together by the same person to extract always-fresh secure secrets. On two real world datasets with 15 and 482 subjects, BANDANA generated fingerprints achieved intra- (50%) and inter-body ($>75\%$) similarity sufficient for secure key generation via fuzzy cryptography. Using BCH codes, best results are achieved with 48 bit fingerprints from 12 gait cycles generating 16 bit long keys. Statistical bias of the generated fingerprints has been evaluated as well as vulnerabilities towards relevant attack scenarios.

Keywords: gait, authentication, fuzzy cryptography, ad-hoc secure pairing

---

1. Introduction

With increasing importance of short-term spontaneous interaction, ad-hoc device pairing promises seamless secure interaction in smart environments.

We envision short-term spontaneous pairing such that co-presence, i.e. devices worn or carried by the same person, suffices for autonomous, spontaneous secure connection establishing (not assuming any prior shared secret, not involving any trusted third party and without leaking information on the key via any communication channel). Pervasive Computing applications for such protocol are numerous and include, for example, the pairing between a personal device worn on the body, and other pervasive, computing, and sensing capable devices. For instance, a shopping basket carried by the same person, or even instrumented items carried inside the basket. Such pairing could enable synchronization of a shopping list on the personal device with items in the basket, or the display of advertisements on the personal device, tailored to match items in the basket.

Furthermore, in a Pervasive Computing setting, computing and sensing capable fitness equipment in a gym could spontaneously pair with a fitness app on a personal device during the context of use to provide accurate information on the intensity and performance of a specific workout.

Also, tablet-based electronic tourist guides could pair spontaneously with a personal on-body device in order to inquire information on language preferences, interest and background to tailor the provided experience on the respective user.

[☆] This paper is an extended version of "D. Schürmann, A. Brüsch, S. Sigg, L. Wolf, BANDANA – Body Area Network Device-to-device Authentication using Natural gAit".

Email addresses: `schuermann@ibr.cs.tu-bs.de` (Dominik Schürmann), `bruesch@ibr.cs.tu-bs.de` (Arne Brüsch), `le.ngu.nguyen@aalto.fi` (Ngu Nguyen), `stephan.sigg@aalto.fi` (Stephan Sigg), `wolf@ibr.cs.tu-bs.de` (Lars Wolf)

Preprint submitted to Pervasive and Mobile Computing on May 2017, revised October 2017 and February 2018

There exist many further examples and in all cases the spontaneous pairing shall break in the very moment that the device (e.g. basket, fitness equipment or tourist guide) is discarded or handed to another person, so that no privacy-related information is disclosed unwittingly. We present BANDANA, a spontaneous secure pairing scheme based on gait, which allows frequent re-pairing (restricted to the time-of-use), and ad-hoc implicit (no manual interaction required) secure authentication bound to an individual. Our solution does not require a trusted third party. In particular, we utilize instantaneous variations in gait sequences for implicit generation of a shared secret among all devices on the same body. Our contributions are (1) a secure ad-hoc pairing scheme for devices worn on the same body, (2) experimental verification of the protocol on two large gait datasets, and (3) security analysis on the pairing approach covering statistical bias, and attack scenarios.

Compared to [1], we integrate BANDANA with Password-Authenticated Key Agreements (PAKEs), such as in Bluetooth's Secure simple Pairing (SSP) to reduce extracted the gait fingerprint to $M = 48$ bits, while retaining security guarantees (cf. Section 4–6.) A new dataset and a consideration of new activities (running, ascending and descending stairs) was added to the evaluation (cf. Section 3.3, and Figures 10, 9), correlation distances for various body parts (cf. Figures 7–9), and a detailed threat model including video attacks have been added (cf. Section 7).

2. Related Work

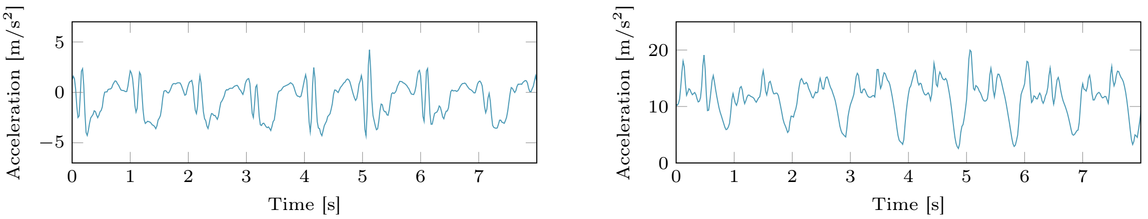
A popular sensor to detect co-presence is the accelerometer. For instance, [2, 3] present a process to generate shared keys via a threshold-based protocol conditioned on the magnitude of co-aligned acceleration processes. [4, 5, 6] further improve this protocol with respect to success probability, different sample rates and starting points as well as differing rotation. Implementations of this protocol have been presented in [7, 8].

For authentication based on arbitrary co-aligned sensor data, the candidate key protocol is proposed in [9]. It interactively exchanges hashes from feature sequences as short secrets and concatenates the key from the secrets with matching hashes. All above implementations require that pairing patterns are explicitly generated (e.g. devices are shaken together). In contrast, we propose to exploit derivation from mean acceleration (where the mean serves as a sort of normalization among devices) to pair devices implicitly conditioned on co-presence on the same body.

Other sensor modalities that can be used for unattended co-presence-based device pairing [10] are magnetometer [11], RF-signals [12, 13] luminosity [14] or audio [15]. For these, in contrast to our approach, pairing is not constrained by co-presence on the same body but, more generally, by proximity.

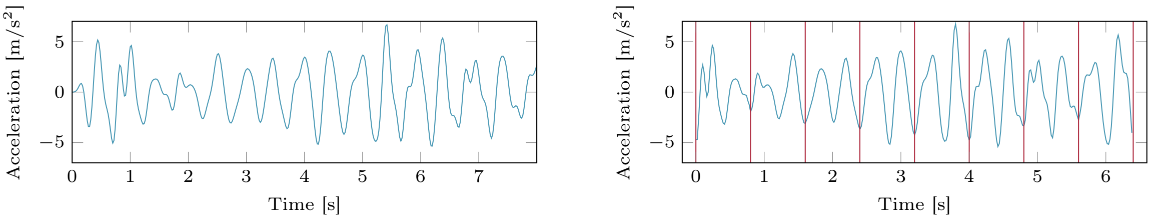
Acceleration sequences from devices worn or carried by the same person differ in orientation and placement [16]. To receive placement independent features one can (A) calculate norm or magnitude $m_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$ (discarding information on acceleration along individual axes [17]), (B) to first detect the location and then to try to deal with changes that occur due to placement [16], or (C) to tackle disorientation and misplacement errors by calculating the rotation matrix from magnetometer readings [18]. Even though after (A), the resulting signal still differed greatly due to inherently differing movement of underlying body parts (e.g. arm vs. head vs. legs) [19], Cornelius et al. [20] succeeded to show good correlation among all body locations. We implement (C) to remove additional uncertainty and noise due to the merged acceleration angles.

For many daily activities, upper body and lower body movements are only weakly or not correlated. We therefore propose to use gait, which can be well recognized over the whole body [21]. For instance, identical step patterns have been utilized for co-location detection [22]. The authors in [23, 18] employ gait cycles to authenticate a user on his smart-phone by matching the current walking pattern against a previously saved walking template exploiting a fuzzy commitment scheme [24]. In [25], it was shown that gait as a biometric feature is robust against an attacker mimicking the victims gait. In their study, professional actors with matching physical properties have been chosen. [23] recently presented an approach to generate a key fingerprint from the difference of a mean world gait (spanning the complete population) to the mean gait of an individual. By computing the mean gait over the whole population, the authors assured that the resulting sequence is well balanced and uniformly distributed.



(a) Accelerometer (z-axis) at 50 Hz

(b) Application of Madgwick's algorithm



(c) Type-II Chebyshev bandpass filter

(d) Resample and Gait Cycle Detection.

Figure 1: Pre-processing and gait cycle detection (z-axis, accelerometer at the forearm).

We also exploit this idea of using the difference to a mean for normalization, but, in contrast, we are not interested in a mean gait over a world population but instead, we derive a mean gait over few preceding gait cycles for all devices on the same body. This is important since the protocol shall generate always fresh instantaneous keys for ad-hoc pairing based on the recent gait history.

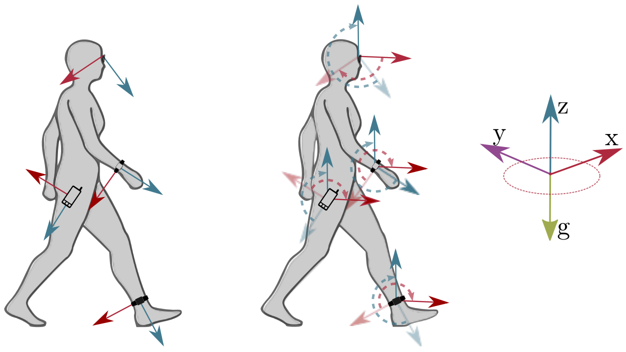
Summarizing, the related work on device pairing from on-body features does, in contrast to our work, (1) not address the impact of different on-body locations and sensor orientation, (2) require devices in close proximity and with strong, purpose-generated acceleration sequences, and (3) use biometric features for distinguishing distinct individuals, rather than instantaneous characteristic movement patterns that change over time. In contrast, we generate always-fresh authentication keys from instantaneous acceleration sequences for arbitrary location on the body. Muaaz et al. [26] confirmed the significant challenge of (1) but demonstrated gait-based authentication for selected related locations on the human body (from one to the other side of the hip), accepting a high error rate.

A work closely related to our study has been presented in [27]. The authors exploit independent component analysis to obtain meaningful gait sequences and extract binary patterns for device pairing by applying a threshold to the data. In contrast, our quantization exploits difference of an instantaneous gait to the mean gait of a respective body location. In addition, we demonstrate that our method is feasible on two freely available benchmark gait databases. In particular, the body locations considered by us cover, in contrast to [27], also lower body-parts, which are more challenging to pair as detailed in Section 6.3.

3. Fundamentals

3.1. Data Pre-Processing

Body-worn sensors feature dynamically changing orientations due to body part movement (cf. Figure 2a). To derive a correlated acceleration independently of the on-body location every data point must be rotated such that one of the axes is facing in the opposite direction of gravity as depicted in Figure 2b. We employ the algorithm proposed by Madgwick et al. [28] to rotate all measurements z_i accordingly, resulting in a signal for the z-axis as indicated in Figure 1b (Orientation and gravity are derived from gyroscope and accelerometer [29]). Compared to sensor fusion based on Kalman filters, Madgwick's algorithm is less computationally expensive due to its linearity and is thus suitable for mobile devices [28]. Afterwards, correlation between records taken simultaneously from devices worn or carried by the same person exists



(a) Before (b) After Madgwick

Figure 2: Body-worn sensors' coordinate systems before and after application of Madgwick's algorithm. Note the remaining degree of freedom along the xy-plane.

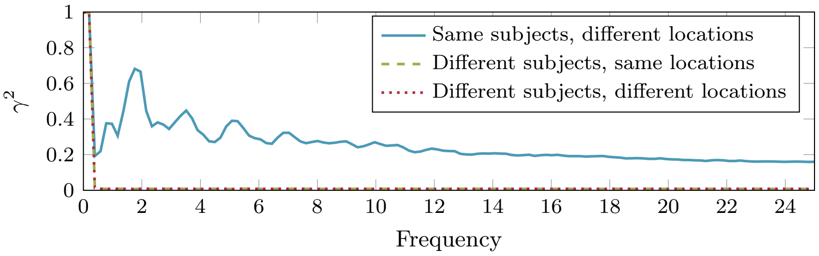


Figure 3: Average spectral coherence for the Mannheim dataset (same and different subject).

(cf. Figure 3). To remove additional noise for correlations in high and low frequencies, we apply a Type II Chebyshev bandpass filter with passband between 0.5 Hz and 12 Hz. The choice for these cutoff frequencies was taken since human motion does not affect frequencies significantly above 10 Hz [30] (cf. Figure 1c).

3.2. Gait Cycle Detection

A gait cycle is the time interval between two successive steps [31]. As discussed in the related work, our algorithm is based on ideas by Hoang et al. [18] providing a reliable method for segmentation. The algorithm's input is a vector of amplitude values $\boldsymbol{z} = (z_1, \ldots, z_n)$ of the accelerometer z-axis (cf. Figure 1a). Its output is a gait sequence of consecutive gait cycles with normalized length.

To find repetitive parts in the signal, clearly separated cycles are extracted based on autocorrelation and distance calculation. The discrete autocorrelation at time lag k and with variance σ^2 is estimated as

$$A_{corr}(k) = \frac{1}{(n-k)\sigma^2} \sum_{t \in \mathbb{Z}} z_{t+k} \cdot \overline{z}_t$$

where $\overline{z}_t$ is the conjugate of z_t. The resulting autocorrelation $\boldsymbol{a} = (a_1, \ldots, a_n)$ leads to m non-ambiguous local maxima in $\boldsymbol{a}$, stored as $\boldsymbol{\zeta} = \{\zeta_1, \ldots, \zeta_i, \ldots \zeta_m\}$. The distances between these indices and a mean distance

$$\delta_{mean} = \left\lceil \frac{\sum_{i=1}^{m-1} \zeta_{i+1} - \zeta_i}{m-1} \right\rceil$$

are then calculated. δ_{mean} can now be used to select indices of minima from $\boldsymbol{z}$ that represent clear cycles with the same length:

$$\begin{aligned} \boldsymbol{\mu} &= \{\mu_1, \ldots, \mu_i, \ldots, \mu_{m-1}\}; \\ \mu_i &= \arg\min(z_{\zeta_i - \tau}, z_{\zeta_i - \tau + 1}, \ldots, z_{\zeta_i + \delta_{mean} + \tau}). \end{aligned}$$

Every μ_j represents the index of a minimum in $\boldsymbol{z}$ limited to the range of δ_{mean} where τ defines an additional correction factor to account for small deviations in gait duration. The indices in $\boldsymbol{\mu}$ are used to split raw data $\boldsymbol{z}$ into gait cycles

$$\begin{aligned} \boldsymbol{Z} &= \{Z_1, \ldots, Z_i, \ldots, Z_q\}; \\ Z_i &= (z_{\mu_{\frac{i}{2}}}, \ldots, z_{\mu_i}, \ldots, z_{\mu_{\frac{i+1}{2}}-1}); \\ & \quad \text{with } i = \{2, 4, ..., q\}. \end{aligned}$$

Finally, the length of gait cycles are normalized by resampling every Z_i using a Fourier method to ρ samples per gait cycle so that $|Z_i| = \rho$ ($Z_i = \{Z_{i1}, \ldots, Z_{i\rho}\}$; cf. Figure 1d). The choice of ρ depends on factors such as sample rate and requirements of the quantization algorithm discussed in Section 4.

3.3. Datasets

In order to verify that our approach is able to establish gait-based short-term spontaneous pairing for devices worn or carried jointly by the same person we employ two real-world datasets that feature specific characteristics well aligned with this aim. In particular, we utilize the *Mannheim* dataset presented in detail in [32] for the use in position aware activity recognition. It features 15 subjects (8 male, age 31.9 ± 12.4, height 173.1 ± 6.9, weight 74.1 ± 13.8), which are equipped with 7 sensors on different body parts (head, upper arm, chest, waist, forearm, thigh, shin), and which performed different activities (walking, running, ascending, descending stairs, ...) for a time period of 10 - 12 minutes each. It is well suited because it features several relevant sensor positions for on-body device pairing, multiple activities and complete ground truth is available from video recordings.

A single limitation of the *Mannheim* dataset is the limited number of participants. We therefore, in addition, verified our approach on the *Osaka* OU-ISIR Gait Database [33]. This dataset features acceleration recordings from a total of 496 subjects from which 482 have been used in this paper after removing samples with missing sensor locations or short duration. Samples are taken from three triaxial accelerometers and gyroscopes worn on different parts of the waist (left, right, center). Subjects traversed a course comprising a straight path, upstairs and down a slope. A conceptual issue in our case lies in the fact that all sensor units were located on rather close locations on the body and mounted to the same harness, potentially introducing an error.

4. BANDANA

For BANDANA's device-to-device authentication, shared secrets are generated based on acceleration sequences independently on participating devices and, in particular, without disclosing information on the gait sequence on the communication channel. For this, we generate binary fingerprints from the gait and utilize fuzzy cryptography to derive unique key sequences. Following Figure 4, we summarize the novel parts of our protocol.

Gait cycle detection is applied on accelerometer data recorded on A and B and corrected by Madgwick's algorithm and Type-II Chebyshev bandpass filter.

We propose a quantization algorithm inspired by [23], but instead of exploiting the difference to a mean world gait, we calculate differences to the mean of a specific gait sequence. The mean gait is thus defined as

$$\boldsymbol{A} = (A_1, \ldots, A_j, \ldots A_\rho); \quad A_j = \frac{\sum_{i=1}^{q} Z_{ij}}{q}$$

and compared to each gait cycle Z_i (Figure 5).

The mean normalizes differences in acceleration patterns at distinct body locations. To extract b bit per gait cycle Z_i, each Z_i is split into b parts of identical length ρ/b. A binary fingerprint $\tilde{\boldsymbol{f}} = (\tilde{f}_1, \ldots, \tilde{f}_M)$ is

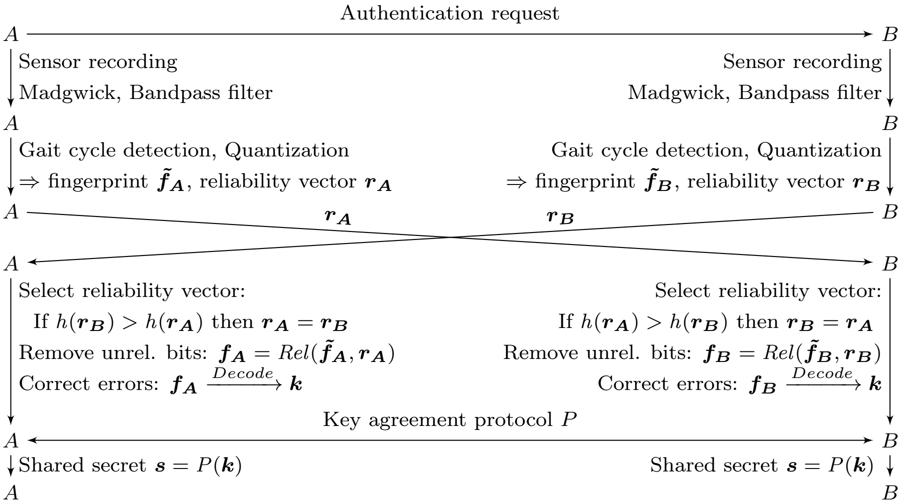

Authentication request
A
B
Sensor recording
Madgwick, Bandpass filter
Sensor recording
Madgwick, Bandpass filter
Gait cycle detection, Quantization
⇒ fingerprint $\tilde{\boldsymbol{f}}_{\boldsymbol{A}}$, reliability vector $\boldsymbol{r}_{\boldsymbol{A}}$
Gait cycle detection, Quantization
⇒ fingerprint $\tilde{\boldsymbol{f}}_{\boldsymbol{B}}$, reliability vector $\boldsymbol{r}_{\boldsymbol{B}}$
$\boldsymbol{r}_{\boldsymbol{A}}$
$\boldsymbol{r}_{\boldsymbol{B}}$
Select reliability vector:
If $h(\boldsymbol{r}_{\boldsymbol{B}}) > h(\boldsymbol{r}_{\boldsymbol{A}})$ then $\boldsymbol{r}_{\boldsymbol{A}} = \boldsymbol{r}_{\boldsymbol{B}}$
Remove unrel. bits: $\boldsymbol{f}_{\boldsymbol{A}} = Rel(\tilde{\boldsymbol{f}}_{\boldsymbol{A}}, \boldsymbol{r}_{\boldsymbol{A}})$
Correct errors: $\boldsymbol{f}_{\boldsymbol{A}} \xrightarrow{Decode} \boldsymbol{k}$
Select reliability vector:
If $h(\boldsymbol{r}_{\boldsymbol{A}}) > h(\boldsymbol{r}_{\boldsymbol{B}})$ then $\boldsymbol{r}_{\boldsymbol{B}} = \boldsymbol{r}_{\boldsymbol{A}}$
Remove unrel. bits: $\boldsymbol{f}_{\boldsymbol{B}} = Rel(\tilde{\boldsymbol{f}}_{\boldsymbol{B}}, \boldsymbol{r}_{\boldsymbol{B}})$
Correct errors: $\boldsymbol{f}_{\boldsymbol{B}} \xrightarrow{Decode} \boldsymbol{k}$
Key agreement protocol P
Shared secret $\boldsymbol{s} = P(\boldsymbol{k})$
Shared secret $\boldsymbol{s} = P(\boldsymbol{k})$

Figure 4: Protocol sequence between two devices A and B worn on the same body.

given by

$$\begin{aligned}
\tilde{\boldsymbol{f}} &= (\tilde{f}_{11}, \dots, \tilde{f}_{1\frac{\rho}{b}}, \dots, \tilde{f}_{b1}, \dots, \tilde{f}_{b\frac{\rho}{b}}) \\
\tilde{f}_{ij} &= \begin{cases} 1, & \delta_{ij} > 0 \\ 0, & \text{otherwise.} \end{cases} \\
\delta_{ij} &= \sum_{k=0}^{\rho/b} A_{i+k} - Z_{i+k,j}.
\end{aligned}$$

The differences of the quantization are

$$\boldsymbol{\delta} = (\delta_{11}, \dots, \delta_{1b}, \dots, \delta_{q1}, \dots, \delta_{qb}).$$

Larger δ_{ij} indicate higher probability to be identical for arbitrary body locations.

The indices of $\boldsymbol{\delta}$ are sorted in descending order by the absolute value $|\delta_{ij}|$ to retrieve the *reliability vector* $\boldsymbol{r} = (r_1, \dots, r_M)$ with $r_i \geq r_{i+1}$. The independently generated vectors $\boldsymbol{r}_{\boldsymbol{A}}$ and $\boldsymbol{r}_{\boldsymbol{B}}$ are exchanged. While their ordering is similar, it is decided that the one with a higher hash value generated by $h()$, e.g., SHA-256, is selected on both sides. Using $Rel(\tilde{\boldsymbol{f}}, \boldsymbol{r})$, the least reliable bits are disregarded for the fingerprint, so that the first N constitute the fingerprint $\boldsymbol{f} = (f_{r_1}, \dots, f_{r_N})$ (cf. Figure 5 (c)).

After reliability ordering, the remaining differences in the derived secrets are corrected with fuzzy cryptography. We choose the codespace $\mathcal{C}$ of an error correcting code (We propose to use BCH codes over the Galois field $\mathbb{F}_2$; A BCH code can be parameterized to correct up to u errors) such that we can directly map a fingerprint $\boldsymbol{f}$ into this codespace. By decoding it with $\boldsymbol{f} \xrightarrow{Decode} \boldsymbol{k}$ into the message-space of the error correcting code, a binary key $\boldsymbol{k}$ is derived that is identical for a pair of on-body devices. Using $\boldsymbol{f} \xrightarrow{Decode} \boldsymbol{k}$, a (K, N)-error correcting code can correct up to $\lfloor \frac{N-K}{2} \rfloor$ errors. Based on the targeted bit size K of the key $\boldsymbol{k}$ and the threshold u for a successful pairing, the required fingerprint size is therefore $N = \frac{K}{2u-1}$.

Finally, a shared secret $\boldsymbol{s}$ can be derived by executing a key agreement protocol $\boldsymbol{s} = P(\boldsymbol{k})$.

5. Key Agreement

BANDANA can be applied in conjunction with various key agreement protocols. To provide a large security margin, we propose to use protocols with a two-party adversarial model, where the attacker is

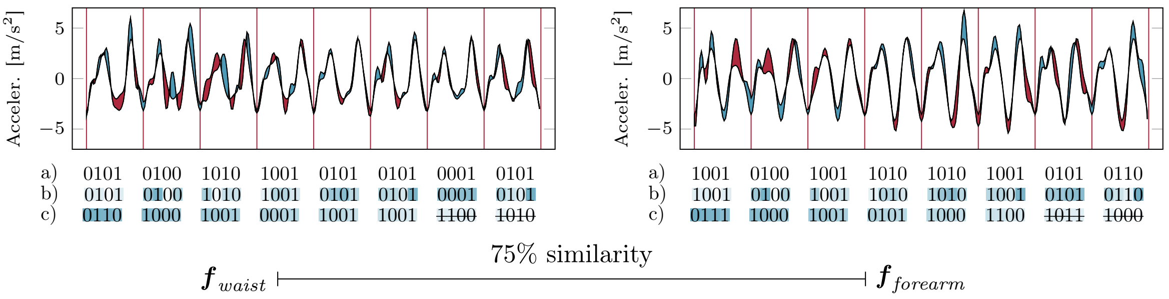


Figure 5: Fingerprint generation on waist and forearm: Energy above average gait cycle in blue and below red. a) after quantization; b) reliability for each bit (darker colors → more reliable); c) sorted by decreasing reliability of forearm and removing least reliable $M - N$ bits

reduced to a one-shot Man-in-the-Middle attacker. A typical design constrains the attacker to only one try by extending a Diffie-Hellmann key exchange. One possible implementation is to use a hash commitment before revealing public values over the channel (cf. Vaudenay [34]). Other protocols, called Password Authenticated Key Exchanges (PAKE), have been proposed with similar goals: The chance of a successful attack should not depend on an attackers offline computing power, but solely on the interaction during the protocol execution. Important standards implementing these primitives include Bluetooth Secure Simple Pairing (SSP), IPSec, and ZRTP [35, 36, 37].

PAKEs can roughly be categorized by (a) their way of storing the password, (b) encrypting transmitted public-keys, and (c) their number of participants [38]. In BANDANA, a “balanced” PAKEs should be used to derive a shared secret on both sides because either party can initiate an exchange (a). Whether public-keys are transmitted encrypted or not can independently be chosen as it is not influenced by BANDANA’s threat model (b). We focus on a two-party adversarial model (c). Besides this categorization, a modern PAKE should provide resilience to dictionary attacks, replay attacks, Unknown Key-Share attack, and Denning-Sacco attack [39]. As security attributes it should provide mutual authentication, key control, known-key security and forward secrecy. However, we note that BANDANA does not require passkey secrecy of a previous authentication attempt, as discussed in Section 7.2. While any modern PAKE within this category could be chosen, we focus on the integration of BANDANA into real-world applications and thus on the Bluetooth standard. Bluetooth 4.2 with *Secure Connection* and *Secure Simple Pairing* fits well into BANDANA’s threat model. BANDANA can be integrated as an additional Out of Band (OoB) mode besides NFC providing $\boldsymbol{k}$ as the Bluetooth passkey. This is considered secure under the PE(i) model in [40]. In Section 6.1 we discuss our choice of an appropriately short key size with a negligible success probability for an attacker (also cf. Section 7.3).

6. Length of Fingerprints and Keys in BANDANA

As sketched in Figure 4, BANDANA utilizes fuzzy cryptography and reliability amplification, both of which shorten the extracted bit sequence so that the final key length is smaller. In the following, we argue on a reasonable size of the key (Section 6.1) as well as on a suggestive number of bits to disregard for reliability amplification (Section 6.2). Finally, we discuss the discriminability of fingerprints (Section 6.3), which defines the parameters of the error correcting code. Final parameters are proposed in Section 6.4.

6.1. Key Size

PAKEs, as discussed in Section 5, prevent offline attacks and can thus provide a sufficiently large security margin even with short key sizes K. Most PAKEs allow for multiple parallel protocol runs per node, such as 2^{10} [34]. In BANDANA we suggest to forbid parallel protocol runs, as this would allow an attacker to boost her success probability by pretending to be multiple devices.

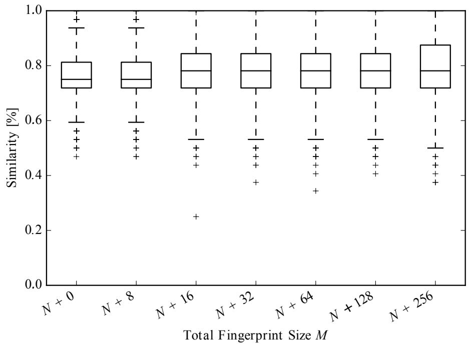


Figure 6: All possible similarities of simultaneous recordings from two *differing* sensor locations on the same subject (intra-body) in the Mannheim dataset (cutoff $N = 32$). Fingerprints are generated by a sliding window with 50% overlap.

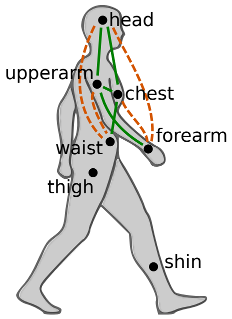


Figure 7: Correlation between on-body sensor locations (Mannheim dataset). green: $\sim 81\%$, dashed orange: $\sim 78\%$, other: $\sim 75\%$

In addition, threat models, such as [41], choose a relatively high $K = 24$ to even have a negligible attacker's success probability if only 16 out of 24 bits are compared correct. Similar margins have been chosen in Bluetooth for PIN comparison with $K =\sim 20$ and ZRTP for word comparison with $K = 20$. In contrast, we can keep a tighter margin as $\boldsymbol{k}$ is generated automatically. Thus, we propose to target a bit size of $K = 16$ with a one-shot success probability for the attacker of 2^{-16}.

6.2. Reliability

We evaluated the number of unreliable bits that could be removed by testing different sequence lengths M with cutoff at $N = 32$ bit using the Mannheim dataset (cf. Figure 6). For $M = 2^i$ the mean-similarity improves by approximately 3% for $i \to i+1$ and settles around $M = N+16$. Thus, we chose $M = N+16$ for our configuration. When repeating this test for $N = 64, 128$, we were able to observe a similar improvement always settling around $M = N + \frac{1}{2}N$.

6.3. Discriminability of Intra- and Inter-body Fingerprints

We observe that upper body locations share a greater similarity than lower body locations. In particular, we identify three similarity groups shown in Figure 7: torso and head ($\sim 81\%$ similarity), upper body with respect to more distant pairs ($\sim 78\%$ similarity), and lower-body locations ($\sim 75\%$ similarity).

Figure 8 illustrates the discriminability between intra-body and inter-body fingerprints. While the intra-body case tests only similarities between differing sensor location on the same body (8037 similarities), each inter-body location case contains 11968975 similarities[1]. As expected, the similarity between different subjects is centered at 50%.

Intra-body similarities for other actions are shown in Figure 9. Due to the strong acceleration during running, which effect the whole body, we observed more homogeneous mean values for this action (cf. Figure 9a). Unfortunately, these are less unique. In contrast, climbing stairs up and down has been shown to generate very unique fingerprints for the upper body (cf. Figure 9b,9c).

In addition to the Mannheim dataset, we evaluated BANDANA using the Osaka dataset, which contains just three sensor locations around the waist, but provides recordings of 482 subjects. Figure 10 illustrates the discriminability between intra-body and inter-body fingerprints. For Osaka, the intra-body test case contains 1446 similarities, while the inter-body case comprises 8694075 similarities.

In the inter-body case, a number of fingerprints (4.64% (Mannheim); 2.47% (Osaka)) match with above 75% similarity. We attribute these collisions to gait sequences with low entropy due to the design of the quantization scheme. To guard against this, we suggest to disregard gait sequences with low entropy.

---

[1] Note that an attacker is constrained to only ~ 3600 tries per day (cf. Section 7).

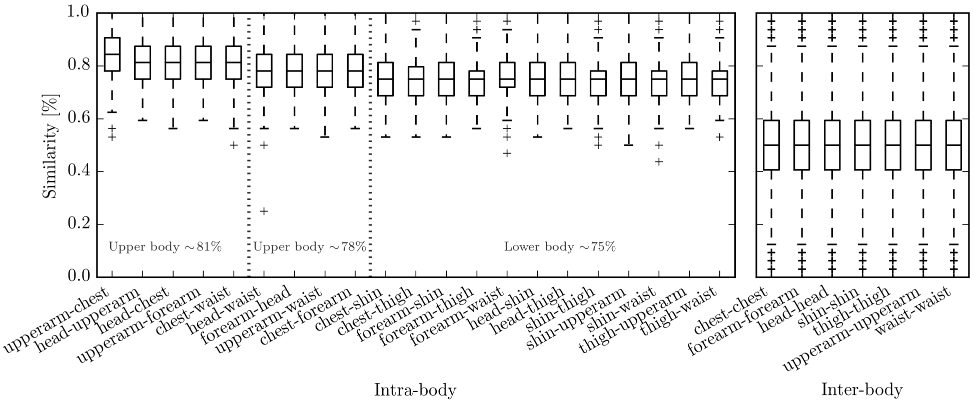


Figure 8: Mannheim (walking): Comparison of intra-body against inter-body similarity. Each value in the *intra-body* boxplot is defined by the similarity of two *different* sensor locations on the same subject (all possible combinations within each subject). For *inter-body*, each boxplot defines a different sensor location. Only *different* subjects are tested against each other with the *same* sensor location. Fingerprint length: $M = 48$ with cutoff at $N = 32$.

6.4. Choice of Parameters

We propose the following configuration for a deployment of BANDANA. As detailed in our security discussion in Section 7, the length K of the resulting key $\boldsymbol{k}$ should be $K = 16$. Following the results depicted in Figure 8, we chose to parameterize the BCH codes to allow correction of at maximum 25% of the bits in the fingerprint. Thus, calculating the error correction rate shows that $N = 32$ bit fingerprints are required: $N = \frac{K}{2u-1} = \frac{16}{2 \cdot 0.75 - 1} = 32$. When using an accelerometer resolution of 50 Hz, we propose a resampling rate of $\rho = 40$ for bit extraction of $b = 4$ bits per gait cycle R_i. Conditioned on ρ and b, we define the correction factor $\tau = \rho/b = 10$. As shown in Figure 6, removing $\frac{1}{3}$ of unreliable bits (i.e. $M = 48$ bit sequences from $q = 12$ gait cycles) provides the best trade-off. We estimate an upper bound for the required length of the recording $\boldsymbol{r}$ as $12 \cdot {\sim}1\ s \approx 12\ s$[2].

7. Security Discussion

In the following, we analyze BANDANA's security model by discussing possible attack scenarios and properties of the fingerprints. In particular, we focus on the risk that an adversary obtains a gait acceleration sequence that is sufficiently similar to pair with a device located on the subjects body following the BANDANA protocol (cf. figure 4). Since BANDANA corrects 8 bits from the 32 bit fingerprints derived, an adversary would be able to successfully pair with an on-body device provided that she is able to establish a 32 bit fingerprint in which at least 24 bits are identical to the fingerprint generated for the on-body device.

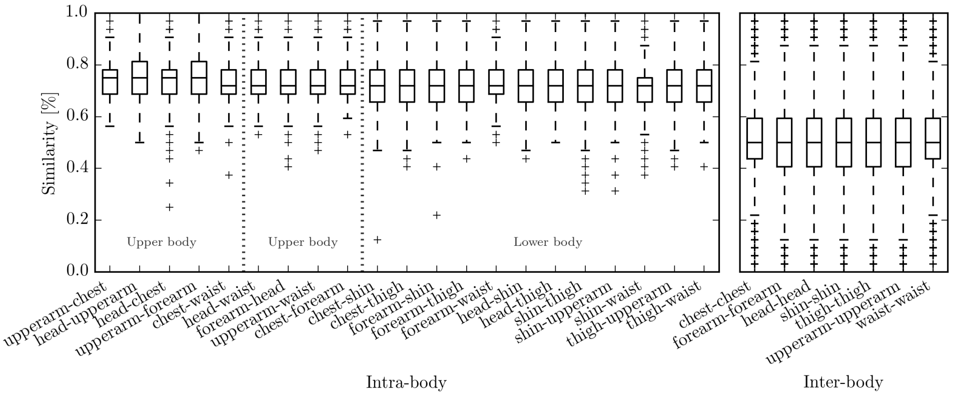
For instance, after successful pairing, an adversary might be able to access private information that shall be restricted to body-worn personal devices only. Considering the example applications specified in the introduction, this might be information related to a subject's shopping list (e.g. for user profiling or also dietary or health related), access to health related data from on-body bio sensors or workout performance, as well as demographics and personal interests.

7.1. Statistical Bias

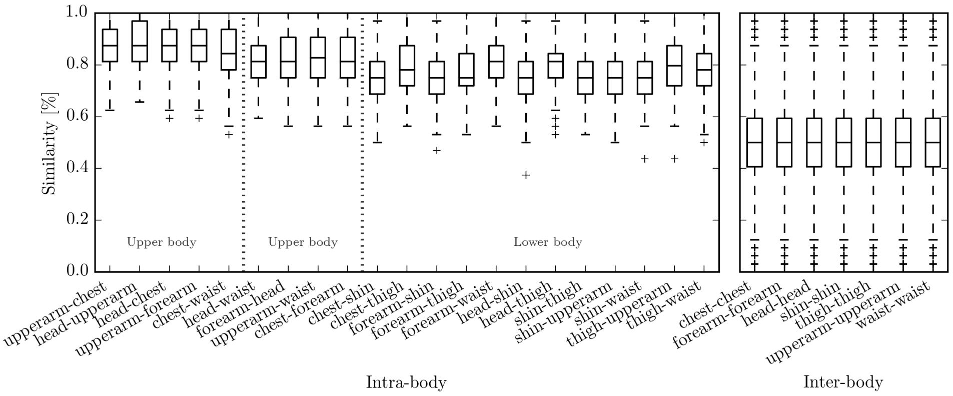
BANDANA is basically a pseudo random number generator (PRNG) conditioned on instantaneous gait acceleration sequences. As for any PRNGs, it is essential that the generated binary sequences are unbiased since the adversary could else exploit knowledge on the bias of the PRNG to boost her chances to guess the

---

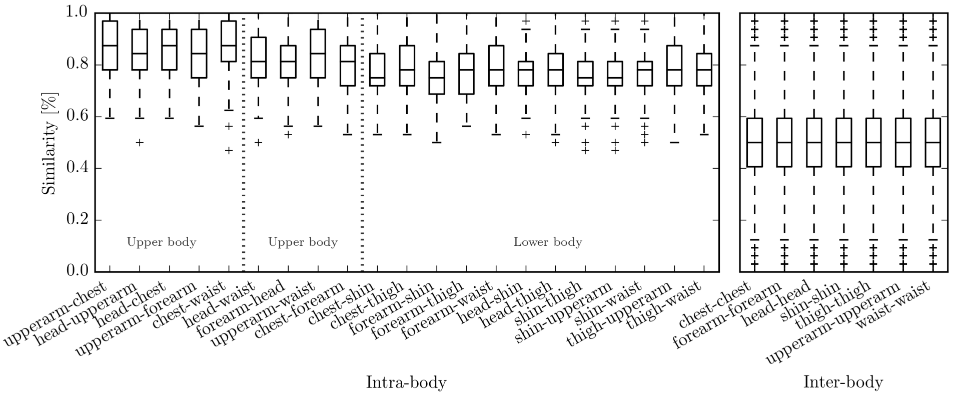
[2] Dall et al. [42] found a mean cadence of 109 steps/minute = 0.91 cycles/second.



(a) Running: Mean values between 75% (upper body) and 71% (lower body)



(b) Climbing down: Mean values between 87% (upper body) and 75% (lower body)



(c) Climbing up: Mean values between 87% (upper body) and 75% (lower body). Subject 2 has been excluded due to missing locations

Figure 9: Intra- vs. inter-body similarity for other actions of the Mannheim dataset. Fingerprint length: $M = 48$ with $N = 32$.

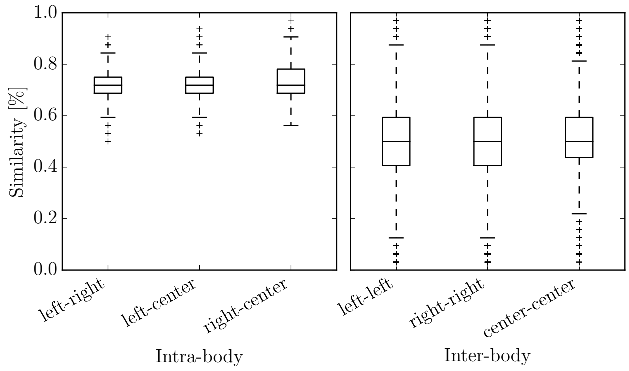


Figure 10: Osaka: Intra- vs. inter-body similarity. Fingerprint length: $M = 48$ with $N = 32$.

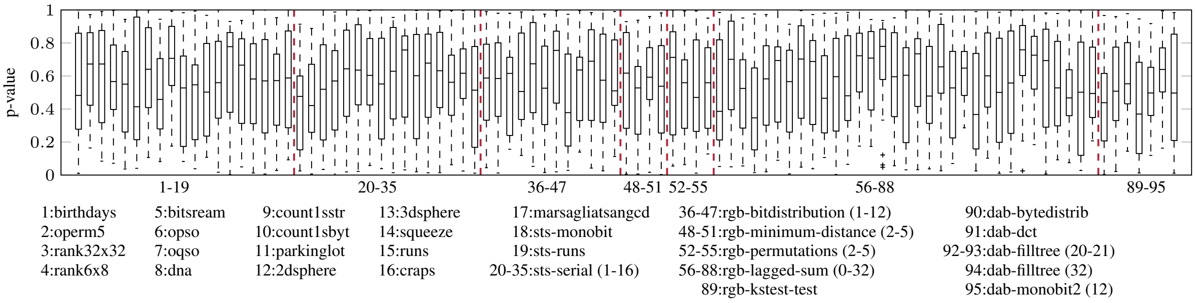


Figure 11: Distribution of p-values achieved for BANDANA fingerprints in 21 runs of the various statistical tests of the dieHarder set of statistical tests.

same binary fingerprint. We rigorously tested the keys generated against statistical bias via the dieHarder battery of statistical tests [43], to uncover bias and dependency in the pseudo random sequence. Test runs produce a value that is compared to the theoretical outcome. A p-value, describing the probability that a real Random Number Generator (RNG) would produce this outcome, between 0 and 1 is computed. A good RNG features uniformly distributed p-values. A p-value below a fixed significance level $\alpha = 0.001$ indicates a failure of the PRNG with probability $1-\alpha$. For instance, a p-value ≤ 0.05 is expected 5% of the time. Our results are depicted in Figure 11. Observe that the p-values are well distributed over the complete range and clustered in the center which indicates a good random distribution.

7.2. No Passkey Secrecy Required

In general, for a pairing scheme, an adversary might consider to exhaust the key-space via multiple repeated attacks. This is not possible for BANDANA though, since $\boldsymbol{k}$ changes with each attempt so that previously learned parts cannot be reused. The adversary is confined to challenge one-shot success probability in each new attempt. This is similar, for instance, to Bluetooth 4.2, which implements *Secure Connection* and *Secure Simple Pairing* (SSP). SSP realizes bit commitment, in which the individual bits of the key are iteratively validated in an interactive protocol. Because each Bluetooth pairing uses a new ephemeral passkey, by design SSP does not provide passkey secrecy [35, 40].

7.3. One-Shot Success Probability

Without requiring additional knowledge about the victim's gait, an attacker may want to exhaust the key-space $\mathcal{C} = \mathbb{F}_{2^{16}}$. However, in BANDANA, after each single try, a completely new authentication process (new $\boldsymbol{k}$ independent from the previous one) is started, thus making it impossible to exhaust $\mathcal{C}$. For $M = 48$

bit long sequences, BANDANA's full process takes about $\sim 12\,\mathrm{s}$. Thus, an optimal imposter is constrained to not more than ~ 7200 tries per day. From each 48 bit sequence, 16 bit are disregarded for reliability amplification. From the remaining 32 bit fingerprints, up to 8 bit are corrected by BCH codes, resulting in $|\boldsymbol{k}| = 16$ bit long keys (cf. Section 6.4). The success probability of a single randomly drawn fingerprint is therefore

$$\sum_{k=0}^{8} \binom{32}{k} / 2^{32} = \frac{\sum_{k=0}^{8} \left(\frac{32!}{(32-k)! \cdot k!} \right)}{2^{32}} \approx 0.0035 \tag{1}$$

7.4. Mimic Gait

A frequently envisioned attack on gait-based authentication and pairing schemes is that an adversary would walk next to the victim, thereby mimicing the victims gait so that a device on the body of the adversary would be able to establish a successful pairing to a device on the body of the victim.

Multiple studies have demonstrated that the success probability of an imposter trying to mimic a subjects gait are low [44] even when trained professionals with similar physical characteristics are employed [25].

For instance, Mjaaland et al [45] trained seven individuals to imitate one specific victim. Even after intensive training over two weeks (5 hours every day), and for one subject even for six weeks, it was not possible for the subjects to accurately imitate the walking pattern of the victim. Also, the provision of continuous visual feedback did not suffice to assist imitators in [46]. Furthermore, the authors of [44] investigated the success probability of an attacker towards a particular subject on a database of 100 subjects and concluded that it is unlikely for an adversary to mimic the subjects gait with sufficient accuracy. This result has been confirmed by [25] who employed professional actors to mimic the gait of 15 subjects with close physical properties. Indeed, the attempt to mimic gait incorporates the risk of asymmetric gait cycles and thus even lowers the chance of success. However, as indicated in [44], the probability of random matches significantly exceeds the expected probability in the birthday paradoxon. An attacker with knowledge to her closest person poses a serious threat to gait-based authentication, and does not even have to impersonate his or her nearest target. This is confirmed in [21, 47] who report an equal error rate (EER) (Equal rates for false acceptance and false rejection) of 20% for gait authentication. In addition, given the gait features of the victim and exploiting a treadmill to control speed, length of steps, thigh lift, hip movement and width of steps, the authors in [48] could reach a false acceptance rate (FAR) of 46.66%.

7.5. Video Recording

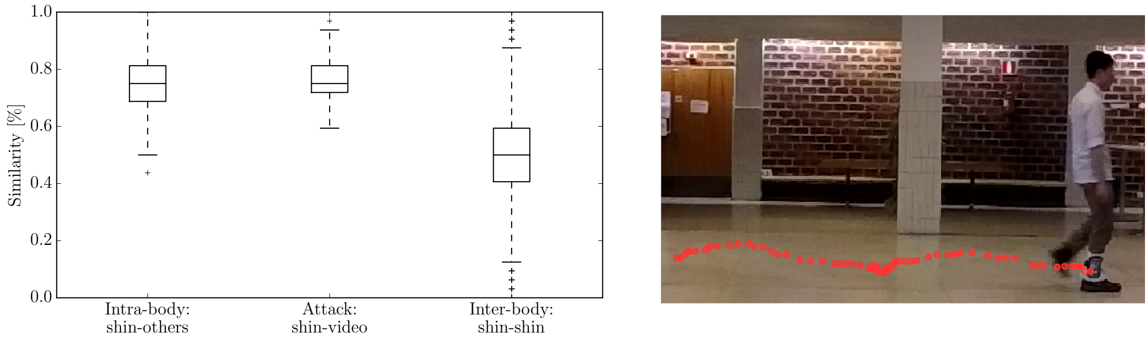
An attacker with access to surveillance cameras could create a video recording of the victim's gait for the timespan during which the device-to-device authentication happens in order to pair with an on-body device. To investigate this attack, we captured user's movement by a wearable inertial measurement unit (smartphone) attached to the subjects shin, and simultaneously with a high-speed camera at 90 fps. We chose the position shin as this location has clearly distinguishable movement from video. With automated video-tracking software, we have not been able to extract the gait from the video with sufficient accuracy. We therefore utilized Tracker[3] to manually track the smartphone on a frame-to-frame basis (cf. Figure 12b). Then, we estimated acceleration data of the smartphone from the tracking result. Figure 12a illustrates the results. The figure shows that a powerful attacker might achieve successful pairing. We considered optimal conditions (stationary high-speed camera at optimal height & subject passing in straight line). We did not succeed to adhere BANDANAs real-time constraints, but a powerful attacker might achieve this.

7.6. Attach Malicious Device

In order to establish a pairing with an on-body device, an attacker could attach a malicious device to the body of the victim, e.g. by slipping a small sensor node into the victim's jacket or by selling a compromised device to the victim. This device could create a second communication channel to forward traffic from inside the BAN to an outsider. Due to the fact that BANDANA works without explicit user interaction,

---

[3] http://physlets.org/tracker/



(a) Mannheim: Intra-body (shin wrt to all same-body pairs); video vs. acceleration at shin; inter-body shin-shin (all pairs)

(b) Tracking the z-axis acceleration by manually annotated video recordings (90 fps).

Figure 12: Approximating the acceleration reading from video.

this attack could succeed if executed properly and unnoticed. We would like to remark, though, that this physical attack also contains significant risk for the attacker to be revealed when such malicious device is detected.

8. Conclusion

We have discussed and analyzed implicit secure device-to-device authentication via the BANDANA protocol for devices worn on the same body. Shared secrets are implicitly extracted for fingerprints generated from the user's gait. The protocol accounts for errors without comparing the fingerprints directly, but utilizes fuzzy cryptography based on error correcting codes. A quantization method for independently generating similar fingerprints at differing sensor locations has been proposed and evaluated. By selecting only reliable bits, we were able to boost the similarity by 3%. Our fingerprints between devices worn on the same body have a minimum similarity of $\geq 75\%$ in contrast to devices worn on different bodies (50%). The protocol was verified on two large gait datasets and for various gait types (walking, running, descending and ascending stairs). The security properties of the protocol have been discussed. BANDANA enables novel pervasive applications such as the pairing between a personal device and a shopping basket in order to synchronize a shopping list on a personal device with items already placed in the basket, as well as for means to advertise offers tailored to a persons shopping items from the basket on a personal device.

Furthermore, fitness equipment in a gym could spontaneously pair with a fitness app on a personal device during the context of use in order to provide accurate information on the intensity and performance of a specific workout.

Also, tablet-based electronic tourist guides could pair spontaneously with a personal on-body device in order to inquire information on language preferences, interest and background to tailor the provided experience on the respective user.

The list of further examples is countless and in all cases the spontaneous pairing would break in the very moment that the device is discarded or handed to another person, so that no privacy-related information is disclosed unwittingly.

Acknowledgments

We appreciate partial funding from an EIT Digital HII Active project, Academy of Finland and the German Academic Exchange Service (DAAD).

References

[1] D. Schürmann, A. Brüsch, S. Sigg, L. Wolf, BANDANA – Body Area Network Device-to-device Authentication using Natural gAit, in: IEEE International Conference on Pervasive Computing and Communications (PerCom), 2017, pp. 190–196.

[2] D. Bichler, G. Stromberg, M. Huemer, M. Löw, Key generation based on acceleration data of shaking processes, in: International Conference on Ubiquitous Computing, Springer, 2007, pp. 304–317.

[3] R. Mayrhofer, H. Gellersen, Shake well before use: Authentication based on accelerometer data, in: Pervasive computing, Springer, 2007, pp. 144–161.

[4] B. Groza, R. Mayrhofer, SAPHE: simple accelerometer based wireless pairing with heuristic trees, in: Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia, ACM, 2012.

[5] Y. Liu, J. Niu, Overlapped-shaking: A local authentication method for mobile applications, in: 2014 IEEE Computing, Communications and IT Applications Conference (ComComAp), IEEE, 2014, pp. 93–97.

[6] R. Mayrhofer, H. Hlavacs, R. D. Findling, Optimal derotation of shared acceleration time series by determining relative spatial alignment, International Journal of Pervasive Computing and Communications 11 (4).

[7] R. D. Findling, M. Muaaz, D. Hintze, R. Mayrhofer, Shakeunlock: Securely unlock mobile devices by shaking them together, in: Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia, ACM, 2014, pp. 165–174.

[8] T. Van Goethem, W. Scheepers, D. Preuveneers, W. Joosen, Accelerometer-Based Device Fingerprinting for Multi-factor Mobile Authentication, in: International Symposium on Engineering Secure Software and Systems, Springer, 2016, pp. 106–121.

[9] R. Mayrhofer, The candidate key protocol for generating secret shared keys from similar sensor data streams, in: European Workshop on Security in Ad-hoc and Sensor Networks, Springer, 2007, pp. 1–15.

[10] S. Sigg, D. Schürmann, Y. Ji, PINtext: A Framework for Secure Communication Based on Context, 2012.

[11] R. Jin, L. Shi, K. Zeng, A. Pande, P. Mohapatra, MagPairing: Pairing Smartphones in Close Proximity Using Magnetometers, IEEE Transactions on Information Forensics and Security 11 (6) (2016) 1306–1320.

[12] A. Varshavsky, A. Scannell, A. LaMarca, E. De Lara, Amigo: Proximity-Based Authentication of Mobile Devices, Springer, Berlin, Heidelberg, 2007, pp. 253–270.

[13] D. A. Knox, T. Kunz, Wireless fingerprints inside a wireless sensor network, ACM Transactions on Sensor Networks (TOSN) 11 (2) (2015) 37.

[14] M. Miettinen, N. Asokan, T. D. Nguyen, A.-R. Sadeghi, M. Sobhani, Context-based zero-interaction pairing and key evolution for advanced personal devices, in: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2014, pp. 880–891.

[15] D. Schürmann, S. Sigg, Secure communication based on ambient audio, IEEE Transactions on Mobile Computing 12 (2) (2013) 358–370.

[16] K. Kunze, Compensating for on-body placement effects in activity recognition, Ph.D. thesis, Citeseer (2011).

[17] M. Muaaz, R. Mayrhofer, Orientation independent cell phone based gait authentication, in: Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia, ACM, 2014, pp. 161–164.

[18] T. Hoang, D. Choi, V. Vo, A. Nguyen, T. Nguyen, A lightweight gait authentication on mobile phone regardless of installation error, in: IFIP International Information Security Conference, Springer, 2013, pp. 83–101.

[19] E. A. Heinz, K. S. Kunze, S. Sulistyo, H. Junker, P. Lukowicz, G. Tröster, Experimental evaluation of variations in primary features used for accelerometric context recognition, in: European Symposium on AI, 2003.

[20] C. Cornelius, D. Kotz, Recognizing whether sensors are on the same body, in: Proceedings of the 9th International Conference on Pervasive Computing (Pervasive'11), Springer-Verlag, Berlin, Heidelberg, 2011, pp. 332–349.

[21] M. Muaaz, R. Mayrhofer, An analysis of different approaches to gait recognition using cell phone based accelerometers, in: International Conference on Advances in Mobile Computing & Multimedia, ACM, 2013.

[22] A. Srivastava, J. Gummeson, M. Baker, K.-H. Kim, Step-by-step detection of personally collocated mobile devices, in: 16th International Workshop on Mobile Computing Systems and Applications, 2015.

[23] T. Hoang, D. Choi, T. Nguyen, Gait authentication on mobile phone using biometric cryptosystem and fuzzy commitment scheme, International Journal of Information Security 14 (6) (2015) 549–560.

[24] A. Juels, M. Wattenberg, A fuzzy commitment scheme, in: 6th ACM conference on Computer and communications security, 1999, pp. 28–36.

[25] M. Muaaz, R. Mayrhofer, Smartphone-based gait recognition: From authentication to imitation, IEEE Transactions on Mobile Computing PP (99) (2017) 1–1.

[26] M. Muaaz, R. Mayrhofer, Cross Pocket Gait Authentication Using Mobile Phone Based Accelerometer Sensor, in: International Conference on Computer Aided Systems Theory, Springer, 2015, pp. 731–738.

[27] W. Xu, G. Revadigar, C. Luo, N. Bergmann, W. Hu, Walkie-talkie: Motion-assisted automatic key generation for secure on-body device communication, in: 2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), 2016.

[28] S. O. Madgwick, A. J. Harrison, R. Vaidyanathan, Estimation of IMU and MARG orientation using a gradient descent algorithm, in: 2011 IEEE International Conference on Rehabilitation Robotics, IEEE, 2011, pp. 1–7.

[29] GSMArena.com, Phone finder results for accelerometer and gyrometer, http://www.gsmarena.com (2016).

[30] J. Lester, B. Hannaford, G. Borriello, "Are You with Me?"–Using Accelerometers to Determine If Two Devices Are Carried by the Same Person, in: Pervasive, 2004.

[31] M. W. Whittle, Chapter 2 - Normal gait, in: M. W. Whittle (Ed.), Gait Analysis (Fourth Edition), fourth edition Edition, Butterworth-Heinemann, Edinburgh, 2007, pp. 47–100.
[32] T. Sztyler, H. Stuckenschmidt, Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition, in: PerCom, 2016.
[33] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, Y. Yagi, The largest inertial sensor-based gait database and performance evaluation of gait-based personal authentication, Pattern Recognition 47 (1) (2014) 228–237.
[34] S. Vaudenay, Secure Communications over Insecure Channels Based on Short Authenticated Strings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 309–326.
[35] J. Suomalainen, J. Valkonen, N. Asokan, Security Associations in Personal Networks: A Comparative Analysis, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 43–57.
[36] C. Kaufman, P. E. Hoffman, Y. Nir, P. Eronen, T. Kivinen, Internet Key Exchange Protocol Version 2 (IKEv2), RFC 7296 (Oct. 2014).
[37] P. Zimmermann, A. Johnston, J. Callas, ZRTP: Media Path Key Agreement for Unicast Secure RTP, RFC 6189 (Informational) (Apr. 2011).
[38] J.-M. Schmidt, Requirements for Password-Authenticated Key Agreement (PAKE) Schemes, RFC 8125 (Apr. 2017).
[39] M. Toorani, Security analysis of j-pake, in: IEEE Symposium on Computers and Communications (ISCC), 2014, pp. 1–6. doi:10.1109/ISCC.2014.6912576.
[40] R. C.-W. Phan, P. Mingard, Analyzing the secure simple pairing in bluetooth v4.0, Wireless Personal Communications 64 (4) (2012) 719–737.
[41] M. Farb, Y.-H. Lin, T. H.-J. Kim, J. McCune, A. Perrig, Safeslinger: Easy-to-use and secure public-key exchange, in: MobiCom'13, ACM, New York, NY, USA, 2013, pp. 417–428.
[42] D. P. Margaret, M. P. R. Walker, G. M. Howard, S. B. William, Step Accumulation per Minute Epoch Is Not the Same as Cadence for Free-Living Adults, Medicine & Science in Sports & Exercise 45 (10).
[43] R. G. Brown, Dieharder: A random number test suite, http://www.phy.duke.edu/∼rgb/General/dieharder.php (2011).
[44] D. Gafurov, E. Snekkenes, P. Bours, Spoof attacks on gait authentication system, IEEE Trans. on Information Forensics and Security 2 (3).
[45] B. B. Mjaaland, P. Bours, D. Gligoroski, Walk the walk: attacking gait biometrics by imitation, in: International Conference on Information Security, Springer, 2010, pp. 361–380.
[46] Ø. Stang, Gait analysis: Is it easy to learn to walk like someone else?, Master's thesis (2007).
[47] M. O. Derawi, C. Nickel, P. Bours, C. Busch, Unobtrusive user-authentication on mobile phones using biometric gait recognition, in: Intelligent Information Hiding and Multimedia Signal Processing, Sixth International Conference on, IEEE, 2010, pp. 306–311.
[48] R. Kumar, V. V. Phoha, A. Jain, Treadmill attack on gait-based authentication systems, in: 2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS), 2015.

On the Secrecy of Publicly Observable Biometric Features: Security Properties of Gait for Mobile Device Pairing

Arne Brüsch, Ngu Nguyen, *Member, IEEE,* Dominik Schürmann, *Member, IEEE,* Stephan Sigg, *Member, IEEE,* and Lars Wolf, *Member, IEEE*

Abstract—Gait has been proposed as a feature for mobile device pairing across arbitrary positions on the human body. Results indicate that the correlation in gait-based features across different body locations is sufficient to establish secure device pairing. However, the population size of the studies is limited and powerful attackers with e.g. capability of video recording are not considered. We present a concise discussion of security properties of gait-based pairing schemes including a discussion of popular quantization schemes, classification and analysis of attack surfaces, analysis of statistical properties of generated sequences, an entropy analysis, as well as possible threats and security weaknesses of gait-based pairing systems. For one of the schemes considered, we present modifications to fix an identified security flaw. As a general limitation of gait-based authentication or pairing systems, we further demonstrate that an adversary with video support can create key sequences that are sufficiently close to on-body generated acceleration sequences to breach gait-based security mechanisms.

◆

1 INTRODUCTION

With the proliferation of mobile devices and the upcoming Internet of Things, interaction between these devices will drastically increase [1]. In particular, on-body devices covering smart appliances, smart textile and digital assistants are to generate a dense body area network of connnected devices [2]. This is extended by spontaneous pairings with devices interacted with during the context of use [3]. In such environment where the number of device pairings raises by n with each $n + 1$st new device, and where device count and type changes on a sub-day schedule, manual pairing is impractical. Implicit pairing schemes have therefore been proposed e.g. based on acceleration [4], audio [5], magnetometer [6] and RF features [7]. From these, especially gait [8] is well suited in wearable settings as it is confined to a single person's body and can be easily read out at arbitrary location on the body [9].

Supported by user studies, it has been argued that gait can even be exploited as a biometric feature for user authentication e.g. unlocking smartphones [10], [11], [12], [13]. However, criticism on this assertion has been raised since the usual population considered is small and also powerful informed attackers have seldom been assumed [14]. Such attacker constitutes e.g. trained professional impostors, video supported attackers or sophisticated exploitation of the underlying protocols. Given that gait is continuously exposed in everyday interaction, it is further valid to ask whether such attackers might be able to steal gait [15] and use it for authentication at some different point in time.

In contrast to such gait-based authentication approaches, this paper investigates gait-based pairing of devices co-located on the same body. In particular, device-pairing exploits correlation in instantaneous gait features and does not assume or require that gait can be exploited as a biometric feature. It is therefore not feasible to use historical gait information in order to break a gait-based pairing scheme.

A number of gait-based pairing schemes have been proposed recently [16], [17], [18], [19]. However, no concise study of the security properties of quantization approaches for gait-based pairing has been presented to-date.

With this article, we close this gap by providing a comprehensive classification of attack surfaces for gait-based device pairing and authentication schemes. Furthermore, we present an in-depth analysis of four popular quantization schemes presented recently for gait-based on-body device pairing. This analysis covers protocol-specific attack surfaces and potential security weaknesses of these schemes, as well as distribution, statistical and entropy analysis of the key sequences generated. In particular, for one of the schemes, we identify and present an improvement to the quantization scheme that can mitigate the security weakness found. Finally, we show that a sophisticated adversary using video recordings is able to break gait-based pairing schemes if executed in real-time. This also constitutes a first empirical example of how gait can be stolen in gait-based authentication systems. Summarizing, our contributions are

- a concise investigation and comparison of popular quantization schemes for gait-based device-pairing,
- a comprehensive discussion of attack surfaces for gait-based pairing approaches,
- an entropy, pattern and statistical analysis for popular quantization schemes for gait-based pairing,
- an improved quantization approach for one of the

- *A. Brüsch, D. Schürmann and L. Wolf are with TU Braunschweig, Germany.*
- *N. Nguyen and S. Sigg are with Aalto University, Finnland.*

All authors contributed equally to this work and are listed in alphabetic order. *Manuscript received October 20, 2017.*

investigated approaches to mitigate the identified security weakness, as well as
- the first ever empirical demonstration that video is capable to estimate gait sufficiently accurate to break gait-based security schemes.

2 RELATED WORK

We discuss recent related studies on gait-based security mechanisms. In particular, we first discuss gait recognition approaches, before we summarize recent progress in gait-based authentication and gait-based pairing. In the remainder of the discussion, we are then focusing on using acceleration sequences from natural gait for device pairing. In particular, we do not consider the use of gait for user authentication. As discussed in Section 2.2, gait as a biometric measure has several limitations and its strength as the seed for the authentication is, even after many years of research, questioned by some. In contrast, we consider attempts to exploit gait as a stimuli acceleration-based ad-hoc pairing, which utilizes the correlation in sensor readings for devices in the same context. In contrast to authentication, ad-hoc pairing generates instantaneous one-time keys. It is therefore not affected by stolen historical gait sequences but relies on the entropy in short-time fluctuation of simultaneous readings from correlated sources. Specifically, machine learning is not feasible for ad-hoc pairing as the challenges created are always fresh and shall not follow common predictable or learnable patterns. The features exploited for key generation can further not have their origin in activity recognition, as such coarse classes would result in small, and therefore weak key spaces.

2.1 Gait Recognition

Traditionally, Gait recognition has been applied exploiting machine vision [20], [21], [22], [23]. Systems then comprise one or multiple cameras to capture natural gait and contain image recognition steps including background subtraction, feature extraction and recognition [24]. First work goes back to perception experiments on light point displays conducted in [10]. This work was further developed in [25] with computer vision approaches to recognize people from gait. In preceding years drastic improvements have been made in gait recognition algorithms [26], [27]. Gait recognition approaches can be grouped into (1) temporal alignment-based, (2) static parameter-based and (3) silhouette shape-based approaches [22]. From these, [28] found that shape is more significant for person identification than kinematics.

Temporal alignment-based approaches emphasize both shape and dynamics and first extract silhouette features before aligning sequences of these e.g. with temporal correlation, dynamic time warping or hidden Markov models.

Static parameter-based approaches exploit gait dynamics such as stride length, cadence and stride speed [29]. However, they are least successful for gait-based identification due to their need for 3D calibration information.

Finally, silhouette shape-based approaches use silhouette shape similarity and disregard temporal information, often considering averaged silhouettes or treating silhouette shapes as collection without specific order [22]. For all above methods, gait recognition can be improved by combining statistical gait features from real and synthetic templates [21]

Due to the increasing availability of wearable sensors such as gyroscopes (rotation), accelerometers (acceleration) or force sensors (force during walking), gait recognition via such wearable sensors is increasingly investigated [11], [30], [31], [32], [33], [34], [35]. In these approaches, acceleration sequences are recorded from devices located at various body locations, most prominently at the waist. The acceleration signal is then denoised e.g. by applying wavelet transformation [34] and changes in walking speed are mitigated utilizing dynamic time warping [36] or similar approaches. Individual steps are identified from the resulting signal by searching for minima and by applying pattern or template matching [34]. Similarity can be estimated by the computation of cross-correlation [33]. Alternatively, machine learning classifiers are trained and applied [32].

Finally, a recent technique employed to acquire human gait is to monitor phase changes of an electromagnetic signal reflected from a subject walking towards a transceiver [37], [38]. The authors exploit changes in channel state information (CSI) from WiFi devices for the detection of gait. After generating spectograms from CSI measurements, similar to Doppler radars, and applying autocorrelation on the torso reflection to remove imperfection in these spectrograms, fine-grained gait patterns are extracted.

Note that frequently, sensors installed in the floor such as pressure sensing mats are also mentioned as modalities for gait recognition [39], [40]. However, in these cases, not gait itself is extracted but other features such as footprints [40], ground reaction force [41] or heel-to-toe ratio [42].

2.2 Gait as a Biometric Pattern for Authentication

Authentication systems comprise sensors converting analog stimuli to digital input that can then be quantized and compared to a database of previously stored features. Thus, these systems allow authenticaton based on biometric features. Gait as a discriminating feature was first studied in [10], [43]. It has been realized that characteristic features in gait enable identification of subjects also in larger gait databases [44], [45], [46], [47]. In addition, multiple studies have demonstrated that the success probability of an imposter trying to mimic a subjects gait are low [14] even when trained professionals with similar physical characteristics are employed [8]. For instance, Hoang et al. [48] generated a key fingerprint from the difference of a mean gait spanning the complete population to the individual's mean gait. In this way, the authors assured that the resulting sequence is well balanced and uniformly distributed. A good overview on gait-based user authentication is provided in [49], [50].

However, despite studies asserting that gait can be used as biometric feature [11], [12], [13], we remark that there is a lack of studies investigating the security features and entropy of gait as an authentication mechanism.

Several attacks though, most significantly impersonation attacks, have been considered (cf. Table 1). For instance, Mjaaland et al [53] trained seven individuals to imitate one specific victim. Even after intensive training over two weeks (5 hours every day), and for one subject even for six weeks, it was not possible for the subjects to accurately imitate the

TABLE 1: Attacks on gait-based wearable authentication systems

Paper	Applications	Attacking
Muaaz et al. [8]	Gait recognition	Active imposter (imitation), 20% EER
Xu et al. [19]	Device pairing	Active imposter (imitation), passive imposter, MitM
Kumar et al. [51]	Gait recognition	Treadmill attack
Trippel et al. [52]	Injection of false acceleration	Poisoning acoustic injection attack
Derawi et al. [49]		Active imposter, 20% EER, significant random success probability
Mjaland et al. [53]	Gait biometrics	Active long-term trained impostors
Stang [54]	Gait biometrics	Training impostors with continuous visual feedback

walking pattern of the victim. Also, the provision of continuous visual feedback did not suffice to assist imitators in [54]. Furthermore, the authors of [14] investigated the success probability of an attacker towards a particular subject on a database of 100 subjects and concluded that it is unlikely for an adversary to mimic the subjects gait with sufficient accuracy. This result has been confirmed by [8] who employed professional actors to mimic the gait of 15 subjects with close physical properties. Indeed, the attempt to mimic gait incorporates the risk of asymmetric gait cycles and thus even lowers the chance of success. However, as indicated in [14], the probability of random matches significantly exceeds the expected probability in the birthday paradoxon. An attacker with knowledge to her closest person poses a serious threat to gait-based authentication, and does not even have to impersonate his or her nearest target. This is confirmed in [49], [55] who report an equal error rate (EER)[1] of 20% for gait authentication. In addition, given the gait features of the victim and exploiting a treadmill to control speed, length of steps, thigh lift, hip movement and width of steps, the authors in [51] could reach a false acceptance rate (FAR) of 46.66%.

In addition, the increasing accuracy of video-based gait recognition systems also empowers an adversary to generate a database of gait information on multiple subjects unnoticed. Video-based attacks on gait-authentication systems are insufficiently investigated in the literature. In Section 5.5, we demonstrate that a sophisticated adversary with video support can estimate gait sufficiently accurate in order to break gait-based authentication and pairing schemes.

We conclude that gait-based authentication faces serious security threats and gait appears not feasible as sole basis for authentication, especially in systems where the adversary is targeting not a specific but any subject in the system. Furthermore, gait changes over time [24] and is affected by clothing, footwear, walking surface [20], walking speed [24] and emotion [56]. These effects are insufficiently studied and render gait-based authentication a challenging undertaking.

2.3 Acceleration-Based Pairing of Devices

Device pairing protocols execute quantization on one or more devices at the same time to generate similar bit sequences. In contrast to user authentications, these sequences are not matched against a template database. Instead they are used to authenticate a key agreement between all participanting parties. Recently, several authors have considered acceleration or gait for the pairing of devices co-present on the same body [55], [57], [58]. In particular, these approaches exploit correlation in acceleration signals when devices are worn on the same body [59], [60] or shaken together [4], [61]. Note that for these approaches, in contrast to exploiting gait for authentication, the existence of a unique and reproducible biometric gait sequence is not required. Instead, the protocols exploit instantaneous, correlated acceleration sequences that can not be re-used at different time as the system can be restricted to single attempts [16]. The above weaknesses for gait as biometric pattern therefore does not apply. The strength of the pairing approach is, instead, conditioned on the quantization used, what entropy that approach can guarantee and whether or not it leaks information to a powerful (realistic) attacker.

In [4], [61] the ShakeUnlock protocol is presented to unlock a mobile device when it is shaken simultaneously with a smartwatch. The individual steps of this protocol are briefly described in Figure 1. This approach, however, requires the direct comparison of acceleration sequences in order to compute correlation and therefore needs an established secure channel to exchange this information.

However, other approaches that do not require already established secure connection between devices have been proposed recently. For authentication based on arbitrary co-aligned sensor data, Mayrhofer [62] proposes the candidate key protocol. A variant of it is also implemented in SAPHE [18]. It interactively exchanges hashes from feature sequences as short secrets and concatenates the key from the secrets with matching hashes (cf. Figure 1).

Walkie-Talkie, an alternative approach conditioned on correlated acceleration sequences from a person's gait, is presented in [19]. The authors achieve a high bitrate by using individual samples for the key if they deviate by at least α standard deviations from the mean (cf. Figure 1).

Furthermore, the BANDANA protocol [16] exploits acceleration along the z-axis only and conditions the gait fingerprint on the difference between instantaneous gait and mean gait at that body location. It thereby achieves normalization among acceleration sequences across body locations. Remaining dissimilarities in fingerprints are corrected with fuzzy cryptography exploiting BCH codes (cf. Figure 1).

Recently, the Inter-Pulse-Interval (IPI) between consecutive steps has been exploited for secure key generation from gait [17]. The protocol exploits the acceleration along the z-axis and concatenates the key sequence as gray-coded, scaled and rounded IPIs. As reported in [17] (cf. Table 2 in Section 5.1) the security and inter-class similarity depends on the speed of consecutive steps and steplength. The protocol was verified on gait captured from devices on the torso of subjects (lower back, upper right arm and right ear).

The quantization methods in these approaches diverge and result in different properties of the generated binary fingerprints. In brief, in SAPHE, challenge-threshold points are randomly drawn around the acceleration sequence. Conditioned on whether a challenge point falls above or below the acceleration sequence, it is interpreted as 1 or 0 for the fingerprint. In contrast, the quantization in Walkie-Talkie interprets samples that exceed (deceed) the mean by

---

[1] Equal rates for false acceptance and false rejection

ShakeUnlock protocol

1) Record acceleration sequences
2) Remove gravity per axis, calculate magnitude and normalize to $[-1, 1]$
3) Share magnitude via secure channel
4) Slice magnitude segments; transform to frequency domain
5) Compute pairwise coherence via cross spectral- & power spectral density
6) Calculate the mean over all coherence values
7) Unlock IFF mean coherence exceeds threshold

Candidate Key protocol (SAPHE)

1) Extract features on devices
2) Hash feature values
3) Exchange hashes to identify matching values
4) When sufficient entropy collected (matching values), concatenate matching values to give secure key.

Walkie-Talkie protocol

1) Agree on heel-strike count. Then, record acceleration.
2) Use ICA for source separation; apply FFT on independent components
3) Low-pass filter (3Hz) in gravity direction (reduce noise and detect local maxima (heel-strikes))
4) Rotate acceleration data using gyroscope to same body coordinate system
5) Low pass filter (10Hz); normalize 3D acceleration to zero mean, unit length
6) Samples $\gtreqless \mu + \alpha\sigma$ are interpreted as 1/0
7) Matching samples chosen define key. IFF $\leq 0.5 + \varepsilon$ overlap, abort (counter impersonation)
8) XOR bit sequences between consecutive windows to obtain keys

BANDANA protocol

1) Collect acceleration readings from the z-axis
2) Correct rotation wrt gravity (gyroscope)
3) Bandpass between 0.5Hz and 12Hz
4) Resampling (40 samples/gait) and gait detection
5) Compute mean gait
6) Difference between mean and instantaneous gait translates to binary sequence
7) Calculate reliability of bits, disregard least reliable
8) Share reliability ordering & create fingerprint
9) Fuzzy cryptography: Get key from fingerprint

Inter-Pulse-Interval (IPI) protocol

(1-4) Analog to the BANDANA Protocol
5) Detect left/right-foot-flat peaks from acceleration
6) $\overline{IPI}_{gray} = \text{Graycode}\left(\left\lfloor \frac{IPI}{m \cdot 1000/f_s} \mod 2^q \right\rfloor\right)$
7) Obtain key as first k bits in $\overline{IPI}_{gray}$

Fig. 1: Description of acceleration-based device-pairing protocols

α standard deviations as 1 (0). In BANDANA, quantization is achieved by comparing instantaneous gait with the mean of preceding gait cycles. Areas in a quarter of a gait cycle where the instantaneous gait exceeds (deceeds) the mean are mapped to 1 (0). Finally, the IPI-protocol generates keys as concatenation of gray-coded, scaled and rounded IPIs.

An attack on acceleration-based pairing is described in [52]. An active adversary emitting modulated acoustic interference at the resonant frequency of materials in MEMS sensors can control or modify measured acceleration, and thus inject changes to acceleration sequences.

3 COMPARISON OF QUANTIZATION SCHEMES

A crucial part in gait-based pairing is the quantization used. It has to preserve a *high similarity* between generated keys on different body parts, and generates *sufficiently unpredictable* bit sequences for the use as cryptographic keys that withstand a computationally unconstrained adversary.

In this section, we analyze the quantization of SAPHE, Walkie-Talkie, BANDANA and IPI and describe their working principles along Figure 2. Additionally, we evaluate how they fulfill the first requirement, i.e., to generate keys with *high similarity* between different locations on the same body (intra-body) and no similarity between different bodies (inter-body). Their second requirement of withstanding adversaries will be discussed in Section 4. In the following, we applied Madgwick's algorithm before each scheme to initialize them on the same accelerometer orientation.

3.1 SAPHE

In the SAPHE [18] protocol, after generating and exchanging the hash $H(r_A)$ ($H(r_B)$) of the random seed r_A (r_B) to compute threshold values $\bar{t}_A$ ($\bar{t}_B$), as points in an Acceleration-time coordinate system $\mathbb{K}$, devices derive acceleration sequences $\overline{v}_A$ ($\overline{v}_B$) in $\mathbb{K}$. Challenges c_A (c_B) that describe whether $\overline{t_A}$ ($\bar{t}_B$) exceed $\overline{v}_A$ ($\overline{v}_B$) are exchanged together with r_A (r_B). The protocol does not disclose information on the acceleration during this communication.

We remark though, that the authors propose a second version which leaks information on the acceleration since, in addition, a distance ordering $\overline{o}_A$ ($\overline{o}_B$) between $\bar{t}_A$ ($\bar{t}_B$) and $\overline{v}_A$ ($\overline{v}_B$) is exchanged. The purpose of this distance ordering is to guard against a specific attack on the hash function (described in [18]). However, an adversary could exploit that the threshold points $\bar{t}_A$ ($\bar{t}_B$) with small distance to $\overline{v}_A$ ($\overline{v}_B$) are good estimates of actual acceleration samples from $\overline{v}_A$ ($\overline{v}_B$). In addition, those threshold points $\bar{t}_A$ ($\bar{t}_B$) with large distance to $\overline{v}_A$ ($\overline{v}_B$) leak information on the probability of the resulting bit (0 or 1 for larger or smaller threshold).

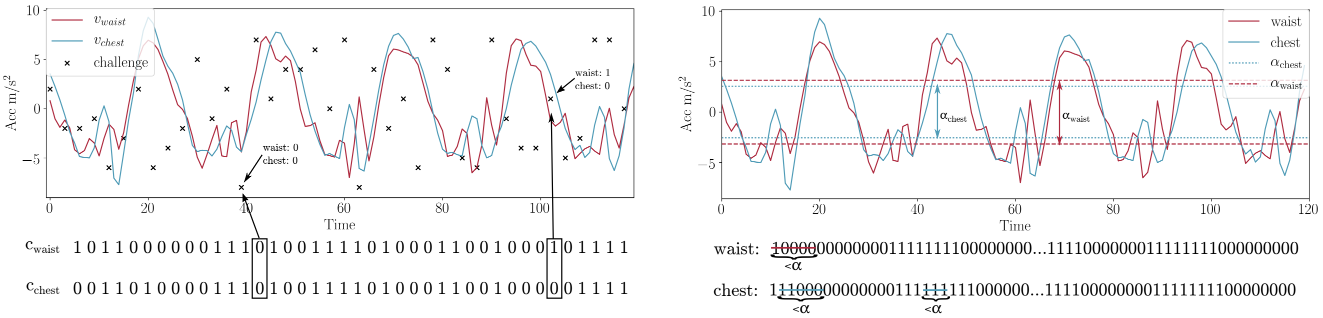
We investigated the pairing performance of SAPHE on walking data recorded in [63][2]. We applied Madgwick's algorithm first and executed SAPHE on the axis perpendicular to earth gravity to correct rotation at arbitrary body locations, . We then removed the gravity by subtracting 9.81 from each value and restricted the range where random seeds are chosen from to $\pm 1g$ (cf. Section 4). In particular, we studied the similarity of keys generated for pairs of devices on different body locations. As depicted in Figure 3a, although affected by outliers, SAPHE's generated key pairs match with high probability of 85% (lower body) to 86,87% (upper body) on average on devices worn on the same body (intra-body). The inter-body case matches on average with 55% i.e. is 5% higher than a random guess. Conclusively, SAPHE is able to generate keys that fulfill the requirement of a clear boundary between intra- and inter-body similarity.

3.2 Walkie-Talkie

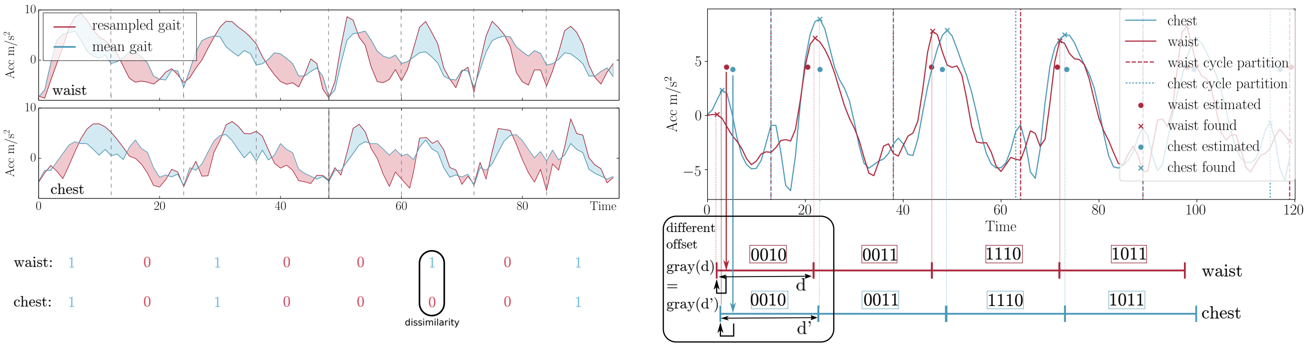
The Walkie-Talkie protocol is (in principle) able to extract up to 1 key bit per acceleration sample and variants that

---

[2] 15 subjects, 10 minutes walking each, acceleration sensors at 7 different body locations



(a) SAPHE (b) Walkie-Talkie



(c) BANDANA (d) IPI

Fig. 2: Descriptive examples for the evaluated quantization schemes

exceed this rate have been proposed in [64]. Acceleration samples are interpreted as 0 or 1 conditioned on whether their acceleration deceeds or exceeds a threshold region, while samples that fall into this region are ignored (cf. Figure 2b). To mitigate hardware originated differences in acceleration strength, devices exchange and agree on samples in the acceleration sequence that shall constitute the key (*reconciliation*). The authors remarked that the resulting sequence is biased towards alternating sequences of groups of 1-bits and 0-bits. This issue was addressed by applying an XOR between consecutive 30 bit long windows of the bit sequence (*privacy amplification*).

We investigated the similarity of keys generated by Walkie-Talkie on walking data from [63]. The protocol achieves bit-similarities of 60-70% for upper body locations and 55-65% for the lower body (cf. Figure 3b). Although the similarity therefore exceeds the random results of 55% achieved for the inter-body case, this performance suggests that further processing or error correction on the generated keys is required to provide reliable pairing among devices at different body location. However, due to the complexity of Walkie-Talkie, we cannot rule out issues on our side in understanding and implementing the algorithm.

3.3 BANDANA

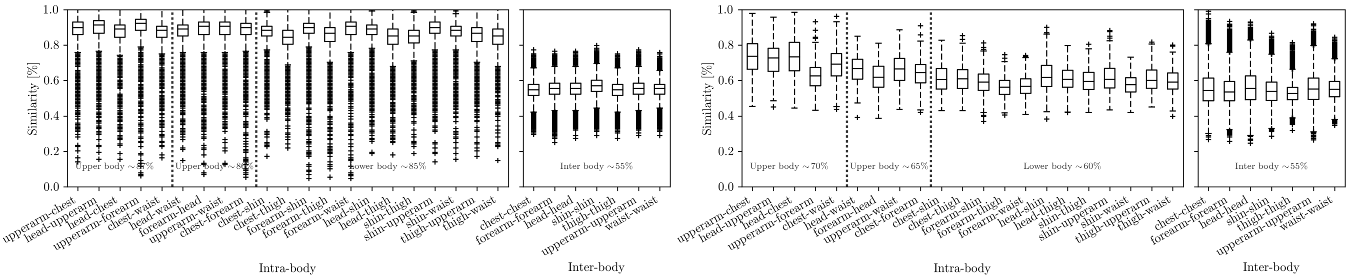
In BANDANA, key sequences are generated as a function of the difference between mean and instantaneous acceleration [16]. The approach of comparing to the mean at a particular body location serves as a normalisation procedure: The offset to the mean has a better correlation across various body locations than comparing absolute acceleration values. Furthermore, [65] argues that this approach might positively impact the distribution of bits in the key sequences towards uniformity as gait patterns are compared to their mean. To further amplify similarity of sequences of bits generated at different body locations, bits with low difference between mean and instantaneous gait are disregarded.

The similarity between keys generated from devices located at different positions on the body (evaluated on the walking data from [63]) is depicted in Figure 3c for the BANDANA protocol. The protocol achieves similarity results above 75% for all location-pairs and is still able to render the chances of the adversary (inter-body) to random guess. The protocol employs fuzzy cryptography in order to mitigate the remaining 25% of difference in the key sequences. We observe, however, a high variance for the inter-body case, which is due to a non-uniform distribution of the key sequences in the key space (cf. Section 4 and Section 5). In Section 6, we discuss how this problem can be addressed with a revised quantization approach.

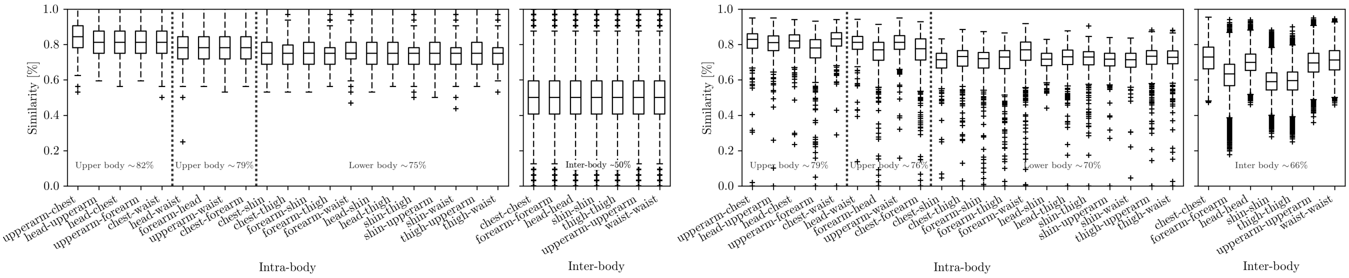
3.4 IPI

The Inter-Pulse-Interval (IPI) protocol [17] exploits the random offset by which individual steps deviate from the mean gait cycle in time domain (cf. Figure 2d). The number of secret bits that can be extracted from the gait signal then depends on the sampling frequency as gait cycle estimation is more accurate with higher sampling rate. The authors report a standard deviation of 40.8 milliseconds for the IPI.

Figure 3d shows the similarity achieved for IPI between keys generated from devices located at different positions on the body (for the walking data in [63]). The similarity in the intra-body case is good and close to the performance of BANDANA. IPI also employs fuzzy cryptography to correct



(a) SAPHE (b) Walkie-Talkie



(c) BANDANA (d) IPI

Fig. 3: Comparison of intra-body against inter-body similarity for the evaluated quantization schemes. Each value in the *intra-body* boxplot is defined by the similarity of two *different* sensor locations on the same subject (all possible combinations within each subject). For *inter-body*, each boxplot defines a different sensor location. Only *different* subjects are tested against each other with the *same* sensor location.

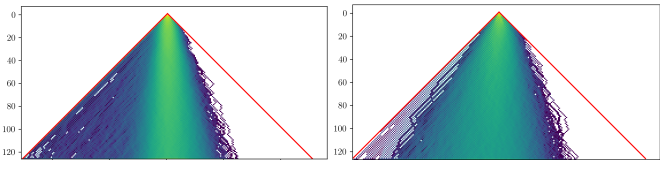
remaining bit-errors in the keys generated for devices across the same body. However, the figure also shows that the protocol does not prevent a remote adversary from paring with on-body devices, since inter-body similarities are as high as in the intra-body case. This is due to limited variation in the generated bit sequences. Inter-pulse intervals resemble a normal distribution centered around its mean. This variation around the mean is similar across subjects and the resolution employed is 4 bits only so that naturally similarity across generated bit sequences is high (cf. Section 5).

4 RANDOMNESS OF KEYS

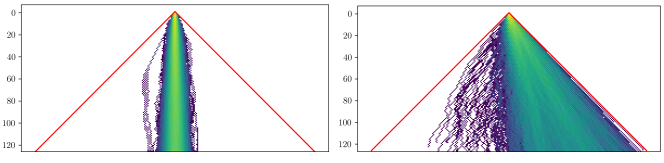
We have so far evaluated quantization schemes regarding their property of generating similar keys for locations on the same body and different keys for different bodies. In this section we investigate whether these keys are *sufficiently unpredictable* to withstand a computationally unconstrained adversary. For this, we analyze the randomness of keys by observing graphs generated from random walks and interpreting the results from the DieHarder and ENT Pseudorandom Number Sequence Tests.

4.1 Bit Distribution

To describe the randomness of keys, we compare their structure with random walks on a Galton board. Plotting a sufficient amount of these sequences will eventually show a binomial distribution [66]. Figure 4 shows heatmaps of random walks corresponding to the sequences generated by different quantization approaches. In addition, Figure 5 depicts each individual random walk such that specific patterns are observable. Based on the last row of each heatmap, Figure 7 depicts the cumulative sums distribution.



(a) SAPHE (b) Walkie-Talkie



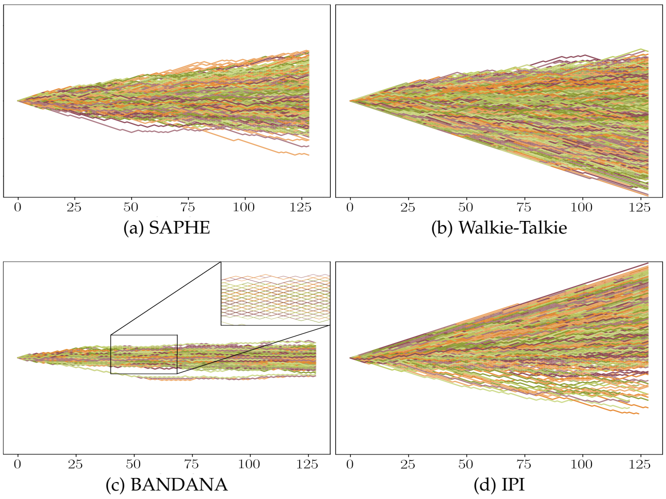
(c) BANDANA (d) IPI

Fig. 4: Heatmaps of random walks for 128 bit keys generated by the evaluated quantization schemes (0 → left; 1 → right). The red lines depict the boundaries for any possible random walk.

Assuming each bit position to be a state in a Markov chain, Figure 6 shows the resulting transition probabilities. We note that we do not analyze Markov properties of higher order.

The input data consisted of triaxial accelerometers, gyroscopes and magnetometers sampled at 50Hz for walking subjects in [63]. Our focus rather lies on same key lengths which means that the time to generate a key may vary heavily between the different approaches. We generated 128 bit fingerprints for each quantization approach.

SAPHE shows a close-to symmetric distribution centered around the mean with a tail reaching almost to the minimum value (cf. Figure 4a). Most likely, this is due to the special characteristic of acceleration readings which do not necessarily have to have zero-mean. Thus, while SAPHE shows good behaviour regarding similarity and usage of space in the Galton board, it carries some characteristics of the input into the output data. Still, this does not pave the way



(a) SAPHE (b) Walkie-Talkie

(c) BANDANA (d) IPI

Fig. 5: Cumulative plot of random walks for 128 bit keys generated by the evaluated quantization schemes (0 → bottom; 1 → top)

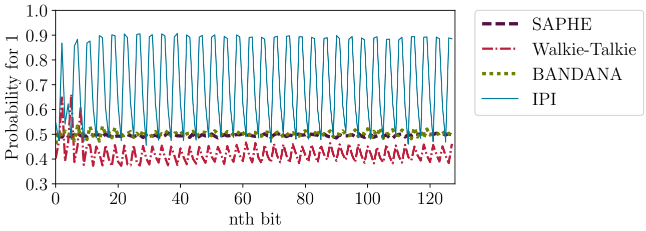
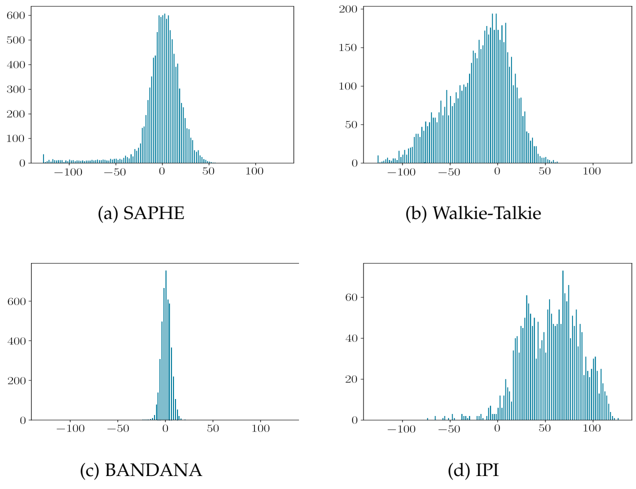


Fig. 6: Markov property: Probability of assigning 1 for the nth position in 128 bit keys

for a successful attack. The cumulative sums distribution is properly centered but shows deviations to include more '0's for a specific set of keys (cf. Figure7a). Note that with regard to the binomial distribution, SAPHE is skewed as it does not adhere to the sigma rule with 68% of the keys in one standard deviation of $\sigma = \sqrt{npq} = \sqrt{11288 \cdot 0.5 \cdot 0.5} = 53.12$. SAPHE shows a very good Markov property (cf. Figure 6).

The heatmap and distribution of Walkie-Talkie are depicted in Figure 4b and Figure 7b. The individual sequences do not show a bias (cf. Figure 5b). Walkie-Talkie, however, shows periodicity in the Markov property (cf. Figure 6). The BANDANA approach features symmetric behaviour but with low variance (cf. Figure 4c, 7c). We can observe from Figure 5c, that this weakness occurs since bit sequences consist of repetitive 'zig-zag' patterns. We discuss this problem in Section 5 and propose an improved quantization to mitigate it in Section 6. BANDANA shows a similar Markov property as SAPHE (cf. Figure 6). Finally, IPI shows good variance but a bias towards including more ones than zeros due to low variation in the quadruples generated as discussed above. IPI clearly deviates from a binomial distribution (cf. Figure 7d). We observed that consecutive 4-bit chunks repeat with a probability of 60%. This clearly shows in IPI's Markov property in Figure 6. Summarizing, while SAPHE and Walkie-Talkie exhibit reasonable randomness, BANDANA and IPI show biases in the generated keys.



(a) SAPHE (b) Walkie-Talkie

(c) BANDANA (d) IPI

Fig. 7: Cumulative sums distribution for 128 bit keys. This corresponds to the distribution in the heatmaps' last rows in Figure 4.

4.2 Statistical Tests

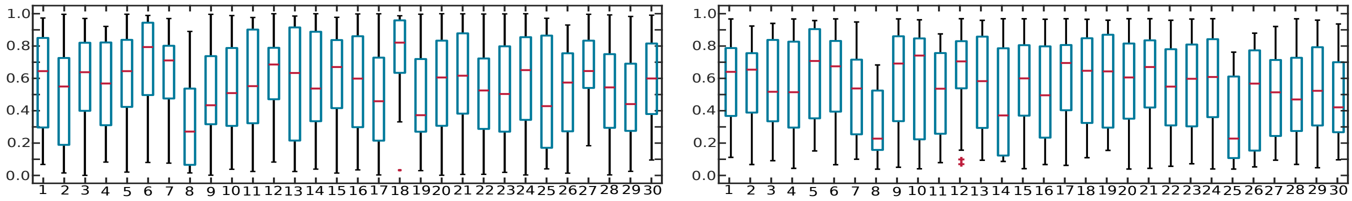
To test if the evaluated quantization schemes against bias in the produced random sequences, we ran the DieHarder statistical tests for each scheme. Figure 8 depicts the p-values computed from 20 runs of the DieHarder tests.

In SAPHE, the *dna* and *sts monobit* tests appear to be outliers. The *dna* test considers biases in the occurence of 10 letter words from an alphabet of 4 letters: C,G,A,T, determined by two designated bits in the sequence of random integers being tested. The *sts monobit* test counts the 1 bits in a long string of random entries and compares this to the expected number. Similar to SAPHE, Walkie-Talkie also shows a weakness in the *dna* test. In addition, the *rgb Kolmogorov-Smirnov* test falls out slightly and the *2D sphere* test features some outliers. The *kolmogorov-Smirnov* test applies a *Kuiper KS* test [67] and the *2D circle* test finds the minimum distance between pairs of randomly selected points to evaluate their randomness. BANDANA shows the most stable distribution of p-values. A slight bias might be associated with the *squeeze* test, which employs a *chi-square* test for cell frequencies on the number of multiplication with random integers that are required to reduce 2^{31} to 1. IPI shows potential weaknesses towards the *birthdays* test, the *Overlapping Quadruples Sparce Occupancy (oqso)* test, the *3D sphere* test as well as the *rgb permutation* and *rgb Kolmogorov Smirnov* test. The *rgb permutation* test counts the order of permutations of random numbers. *Birthdays* test determines the number of matching intervals from 512 'birthdays' drawn from a 24-bit 'year' while the *oqso* test, similar to the *dna* test, considers 4-letter words from an alphabet of 32 letters.

Additionally, we ran the *Ent Pseudorandom Number Sequence* Test[3]. The information density of bit sequences is computed together with reduction through optimal compression, chi square distribution, arithmetic mean of data bytes as well as serial correlation coefficient (cf. Table 2). We caution that these results are only showing the interdependence of single bits. Evaluating chunk instead of single bit interdependence, such as 4-bit chunks for BANDANA due to its 4 bit per gait cycle or 30-bit chunks for Walkie-Talkie's privacy amplification, heavily influences the test results.

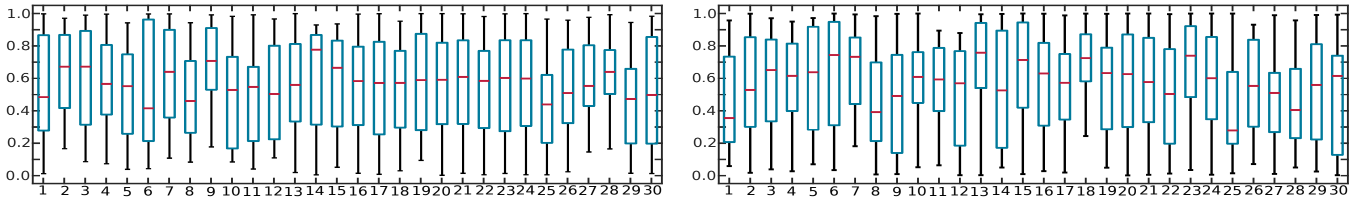
---

[3] http://www.fourmilab.ch/random/



(a) SAPHE

(b) Walkie-Talkie



(c) BANDANA

(d) IPI

Fig. 8: Distribution of p-values achieved for keys after 20 runs of the DieHarder set of statistical tests. Tests are: (1) birthdays (2) operm5 (3) rank32x32 (4) rank6x8 (5) bitstream (6) opso (7) oqso (8) dna (9) count-1s-str (10) count-1s-byt (11) parking (12) 2D circle (13) 3D sphere (14) squeeze (15) runs (16) craps (17) marsaglia (18) sts monobit (19) sts runs (20) sts serial [1-16] (21) rgb bitdistr. [1-12] (22) rgb min dist. [2-5] (23) rgb perm. [2-5] (24) rgb lagged sum [0-32] (25) rgb kstest (26) dab bytedistr. (27) dab dct (28) dab filltree (29) dab filltree 2 (30) dab monobit 2

TABLE 2: Results for keys generated by the evaluated protocols after running the ENT Pseudorandom Number Sequence Test Program.

	SAPHE	Walkie-Talkie	BANDANA	IPI
Sequence size (bit)	1444864	7758808	113792	456104
Entropy (bits per bit)	0.9999	0.9958	0.9999	0.8929
Optimum compression rate	0%	0 %	0%	10%
Chi square distribution	6.91	44993.29	0.3586	65969.75
Arithmetic mean (random=.5)	0.501094	0.461924	0.5	0.690156
Monte Carlo value for Pi (error)	3.122155	3.084985	3.642194	2.056830
Serial correlation coefficient (uncorrelated=0.0)	0.008204	0.080251	-0.644796	-0.002701

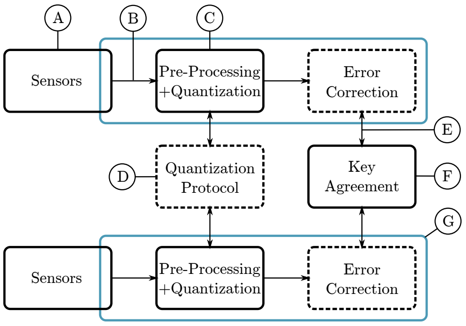


Fig. 9: Conceptual view of gait-based pairing protocols with attack vectors (blue line depicts device boundary, dashed parts are optional)

5 SECURITY ANALYSIS

As in most security applications, although much effort against it may have been taken, unwanted vulnerabilities might anyway sustain within an approach's conceptual design. In [55], [68], a general concept to analyse security applications and to identify possible pitfalls has been introduced. Furthermore, adversaries of different strength necessitate the weighting of the impact of certain weaknesses. Figure 9 shows such a conceptual view for sensor-based device-to-device pairing including possible attack vectors A-G. A pair of devices measure data from sensors, quantize the data to bit strings after some preprocessing, apply potentially error correction, and finally agree on a key. An obvious attack surface is exposed by the sensors themself (A). A device or its owner could be forced to behave in a certain way, e.g., by an adversary controlling stride speed with a treadmill. Depending on the pairing protocol, it could be also possible to bypass the sensor data acquisition (B) and reuse data from the past. With a biased quantization, a naïve brute force attack would become feasible (C). Some protocols employ a special communication phase before the actual key agreement (SAPHE: random seed and distance ordering, Walkie-Talkie: reconciliation phase, BANDANA: exchange of reliability indices). The exchanged indices could potentially be exploited as a side channel (D). After error correction (e.g. in BANDANA and IPI), the key agreement is executed between both participants. Here, the risk of a Man-in-the-Middle (MitM) attack (E) or impersonation attack on one participant (G) must be considered. Finally, the key agreement itself could be weak or based on false security assumptions, especially if it has been designed for this protocol only and is not based on established standards (F). In the following, we will discuss attacks we found during our analysis. No discussion on attack vector B is included as it assumes that the device is already compromised by malware, which falls outside the focus of this work.

5.1 One-Shot Success Probability (E, G)

Without requiring additional knowledge about the victim's gait, an attacker may want to exhaust the keyspace $\mathcal{C}$ of all keys $\boldsymbol{k}$ to execute a MitM (E) or impersonation attack (G). However, in all discussed protocols, after each single try, a completely new authentication process (new $\boldsymbol{k}$ independent from the previous one) is started. Thus, it is impossible to exhaust $\mathcal{C}$ making this a one-shot attack. For comparison between protocols, we target the same length of 16 bit for $\boldsymbol{k}$. The length of sequences sampled for a target key $\boldsymbol{k}$ of 16 bit may vary depending on the quantization scheme.

5.1.1 *Candidate Key Protocol Variants*

The candidate key protocol is, for instance, realized in SAPHE [18], which resolves its original vunlnerability against MitM attacks. In particular, first, random challenges are chosen, as depicted in Figure 2a and committed by sharing their hashes. Afterwards, the acceleration sequence is challenged with respect to these random thresholds where an acceleration point with value lower (higher) than a threshold is interpreted as 0 (1). The success probability for a single randomly drawn key $\boldsymbol{k}$ in SAPHE is

$$\frac{1}{2^{16}} \approx 1.52588 \cdot 10^{-5} \tag{1}$$

5.1.2 *Walkie-Talkie Protocol*

The bits generated in the Walkie-Talkie protocol feature a high bit rate of 15–55 bits per second as reported in [19] (Figure 12(e)). However, high agreement rates are reached only for $\alpha > 0.8$ (Figure 12(d) and 12(f) in [19]), which corresponds to 15–25 bits per second. A 16 bit binary key can therefore be generated in approximately 1 second and the success probability of an adversary for a single randomly drawn $\boldsymbol{k}$ is then

$$\frac{1}{2^{16}} \approx 1.52588 \cdot 10^{-5} \tag{2}$$

5.1.3 *BANDANA Protocol*

In the BANDANA protocol, $M = 48$ bit long sequences are generated in about 12 s. From each 48 bit sequence, 16 bit are disregarded for reliability amplification. From the remaining 32 bit fingerprints, up to 8 bit are corrected by BCH codes, resulting in $|\boldsymbol{k}| = 16$ bit long keys. The success probability of a single randomly drawn fingerprint is therefore (cf. Section 5.3)

$$\sum_{k=0}^{8} \begin{pmatrix} 32 \\ k \end{pmatrix} / 2^{32} = \frac{\sum_{k=0}^{8} \left(\frac{32!}{(32-k)! \cdot k!} \right)}{2^{32}} \approx 0.0035 \tag{3}$$

5.1.4 *IPI Protocol*

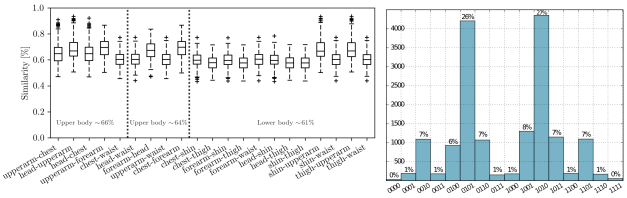
In the IPI protocol, dependent on the sampling frequency, between 2 and 20 secure bits are extracted from each gait cycle (cf. Table I in [17]). Depending on the sample rate of the accelerometer, the generation of 32 bits in the IPI protocol might therefore require from 2 to 16 seconds. Since the protocol also employs fuzzy cryptography protocol for error correction, the same success probability as in the BANDANA protocol of 0.0035 applies for a single randomly drawn fingerprint.

5.2 Quantization-Specific Attacks (C, D)

An attacker with insight to the quantization schemes might be able to exploit this knowledge in order to boost her one-shot success probability. We discuss our observations in the Walkie-Talkie, BANDANA and IPI protocols. For SAPHE, we did not identify any quantization-specific weakness.

5.2.1 *Walkie-Talkie Protocol*

As discussed in 3.2, Walkie-Talkie is biased towards generating alternating sequences of 1-bits and 0-bits. We note that an adversary overhearing the disregarded bit positions during reconciliation (D) is able to formulate an educated guess on the sequence after reconciliation, as 0-1 (1-0) changes in the generated sequences are marked by missing bit positions[4]. Figure 10a shows key similarities achieved by this attack when guessed sequences are compared to actual acceleration-based sequences.



(a) Walkie-Talkie: Overheared bit positions during reconciliation are used to generate bit sequences with similarities higher than 50%

(b) BANDANA: Non-uniform distribution of 4-bit chunks per gait cycles

Fig. 10: Increasing one-shot success probability due to bias in sequences

The application of the XOR in the protocol does not protect against this attack since the adversary can apply the same operation on her estimated sequence.

5.2.2 *BANDANA Protocol*

As indicated in Section 3.3, we found that the random success probability for the BANDANA protocol exceeds random guess. Indeed, as observed in Section 4 (Figure 4c), the variance in generated sequences is low and, in particular, sequences follow specific patterns (cf. Figure 5c). As depicted in Figure 10b, we found as the reason for this weakness that in the 4-bit chunks, which are generated per gait cycle (and before throwing away bits for reliability amplification), sequences of alternating binary value are significantly more frequent than others. In particular, sequences 1111 or 0000, where the instantaneous acceleration constantly exceeds or deceeds the mean acceleration, are seldom. Consequently, the distribution of key sequences in the key space is not uniform, and an adversary could utlilize this knowledge to launch an attack (C). We propose an approach to mitigate this problem in Section 6.

5.2.3 *IPI Protocol*

As discussed in Section 3.4, the IPI protocol suffers from measurement noise in accurately capturing the inter pulse interval due to the limited sampling rate of accelerometers. Especially for lower sampling rates, this significantly restricts the size of the key space. For instance, with 50 Hz (500Hz) sampling rate, one sample is taken every 20 milliseconds (every 2 ms). Since devices are not synchronized, this translates to an unavoidable inaccuracy of up to 10ms (1ms) for the sampled gait on devices (cf. Figure 2d). This measurement noise, compared with only 40.8ms standard deviation for the IPI results in a small keyspace and, since gray codes are employed (modulo 16; $q = 4$), not all bits in the generated quadruples change. In particular, we investigated the variation in 4 bit chunks generated by the IPI protocol on the walking data from [63]. In about 63% of the consecutive 4 bit chunks, all bits are identical. Furthermore, in 24% of all cases, just one bit changed, with 11% 2 bits changed and with only 0.02%, 3 bits were different. An adversary with approximate information on

[4] For a 0-1 change to occur in two consecutive samples, with a guard band size of 5m/s^2 an accelerometer sampling at 50Hz would be required to accelerate at $5\text{m/s}^3/0.02\text{s} = 250\text{m/s}^2$ or 25g per second.

the IPI can therefore boost her guessing success probability significantly beyond chance.

5.3 Benefits and Pitfalls in using Error Correction

In biometric authentication systems, noise of the biometric information is an intrinsic property (here: measurement noise in acceleration sensors). Prominently, Fuzzy cryptography has been proposed in order to employ error correcting codes to mitigate such noise. Error correcting codes encode messages from a messagespace $m \in \mathcal{M}$ into codewords of the (larger) codespace $c \in \mathcal{C}$ introducing redundancies. This process allows to correct errors introduced to c by decoding it back to m. In fuzzy cryptography, the biometric information or fingerprints contain noise or errors that can be corrected after mapping into $\mathcal{C}$. The redundancy introduced in the encoding process, however, dictates that an adversary also does not have to guess all bits in the fingerprint correctly, but can be sloppy. For instance, assume a key length of K and an error correcting code able to correct a fraction of u bits from the total fingerprint length N. Since we know that a (K, N)-error correcting code can correct up to $\lfloor \frac{N-K}{2} \rfloor$ errors, it follows that $N = \frac{K}{2u-1}$. This means that the success probability of a single randomly drawn fingerprint is not 2^N, but instead only

$$\sum_{k=0}^{u} \left(\begin{array}{c} N \\ k \end{array} \right) / 2^N = \frac{\sum_{k=0}^{u} \left(\frac{N!}{(N-k)! \cdot k!} \right)}{2^N} \qquad (4)$$

since up to K errors are allowed at arbitrary position in the fingerprint sequence. Careful choice of the parameters is therefore demanded to limit the advantage gained by an adversary through the use of fuzzy cryptography. From the protocols we investigated, BANDANA and the IPI-protocol employ fuzzy cryptography.

5.4 Gait Mimicry (A)

As recently discussed in [69], it is unlikely that an attacker would be able to mimic natural gait of a victim to a degree where gait sequences were sufficiently similar to break gait-based authentication or pairing schemes. In particular, the authors employed professional actors to mimic the gait of victims with similar physical properties (age, weight, height, shoe size, upper leg length) and showed that after guided training and instructions, all actors failed to mimic the observed gait of victims. In a second test, by walking next to a victim one out of five attackers was able though to achieve sufficient similarity in the gait acceleration sequence. In particular, the authors assumed that the victim instinctively adapted her walking speed to the common step pattern with the adversary. This was, however, not further investigated.

5.5 Impersonation via Video Recording (G)

Cameras are omnipresent in these days, for instance as CCTV systems, personal camcorders, or mobile phones. The quality of captured videos is sufficient to discriminate subtle movements. An adversary with camera-support might therefore be able to extract pairing keys from recorded video (G). In this section, we investigate the threat of video-based side-channel attacks. In particular, we consider how accurate acceleration sequences describing gait can be estimated by tracking movement of body parts from video.

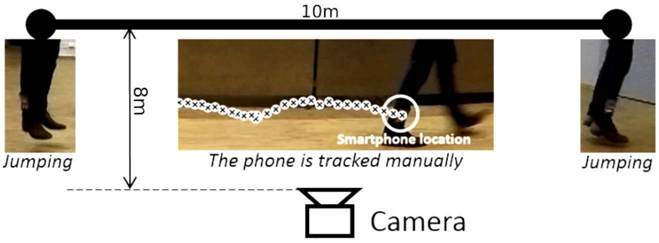


Fig. 11: Experimental setup for video-based attack on gait-based pairing

For our experiment, we captured movement of a subject both by a wearable inertial measurement unit (smartphone) and with a high-speed camera. The smartphone was attached to one leg. We chose this location since it is easy to track from video. If it is possible to pair with a device on that location, it is also possible to pair with devices on any part of the body, as demonstrated by the Walkie-Talkie, BANDANA and IPI protocols. Five subjects (4 male; height: 1.63-1.95m; $\mu = 1.76$m) walked in a straight line in approximately 8m distance to the camera (1080p resolution; 90fps) mounted on a tripod (cf. Figure 11). Acceleration data was sampled at 50Hz. For synchronization between video and inertial sensor, a single jump both at the beginning and at the end framed the walking segment. Each subject conducted the experiment twice.

We utilized Tracker[5] to manually track the location of the smartphone on the recorded video. Although human pose estimation [70] is able to estimate leg movements, we achieved higher accuracy by manually marking the location of the smartphone on the video frames. From the tracked trajectory we estimated the acceleration of the smartphone. This estimated acceleration sequence is then re-sampled to match the 50Hz sampling rate of the inertial sensor. Note that we estimated movement orthogonal to ground only while the inertial sensor might be rotated. However, such rotation is implicitly corrected by the pairing scheme. Figure 12a illustrates example sequences.

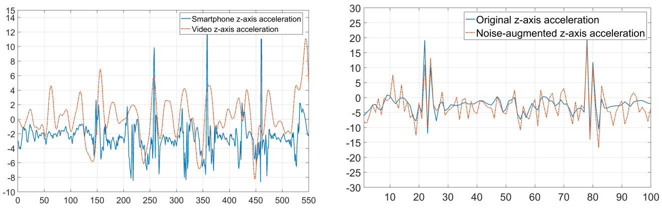
Since accurate manual frame-based tracking of 90fps videos is extremely labour-intensive, a large-scale study including high number of subjects and hours of gait-acceleration is not feasible. We instead estimated the mean $\mu_v = 2.0921$[6] and standard deviation $\sigma_v = 6.0210$ of disparity values between optimally synchronized[7] gait acceleration sequences (estimated and recorded) in our experiment. These values were then used as parameters for noise distributions, which we added to the walking data recorded by the dataset in [63]. We generated Gaussian ($p_n(n) = \frac{1}{\sqrt{\pi\sigma^2}} e^{-\frac{(n-\mu)^2}{\sigma^2}}$), Laplacian ($p_n(n) = \frac{1}{\sqrt{2\sigma}} e^{-\frac{\sqrt{2}|n-\mu|}{\sigma}}$), and

---

[5] http://physlets.org/tracker/

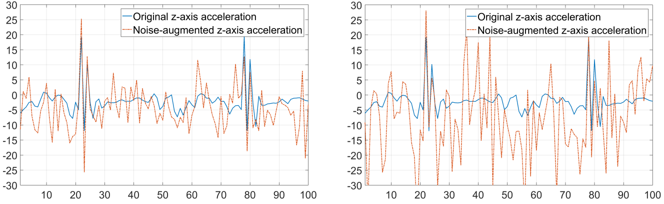
[6] This mean originates from the amplitude estimation error of the adversary due to inaccurate distance measurement between camera and walking subject. Since the adversary does not know the mean offset of the estimated sequence to the actual sequence, we keep this constant error also in our investigation.

[7] We refined the synchronization between the estimated and recorded acceleration sequences by shifting both sequences until a minimum root mean squared error is achieved



(a) Alignment of acceleration sequences from smartphone and video

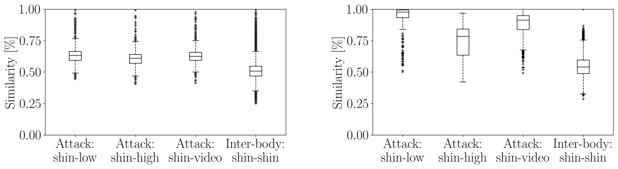
(b) Acceleration sequence augmented with low noise level ($\mathcal{N}(\frac{\mu_v}{2},\ \frac{\sigma_v^2}{4})$)



(c) Acceleration sequence augmented with video noise level ($\mathcal{N}(\mu_v,\ \sigma_v^2)$)

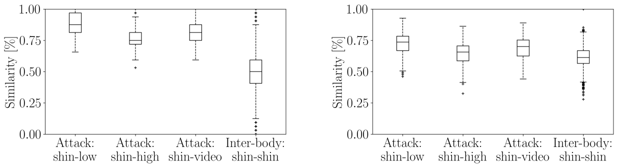
(d) Acceleration sequence augmented with high noise level ($\mathcal{N}(2\cdot\mu_v,\ 4\cdot\sigma_v^2)$)

Fig. 12: Acceleration signals featuring different noise levels



(a) SAPHE

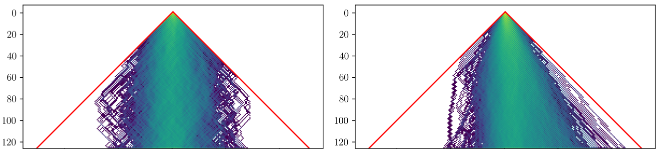
(b) Walkie-Talkie



(c) BANDANA

(d) IPI

Fig. 13: Attacks using Video-based impersonation: Similarity of gait-fingerprints with different noise levels over four pairing schemes



(a) Mapping approach

(b) Normalization approach

Fig. 14: BANDANA improvements: Heatmaps of random walks for 128 bit keys generated by improved versions

uniformly distributed ($p_n(n) = \frac{1}{2\sqrt{3}\sigma}$) noise.

We then generated noisy acceleration signals with $\mathcal{N}(\mu_v,\ \sigma_v^2)$ (noise observed from video-based acceleration estimation), $\mathcal{N}(\frac{\mu_v}{2},\ \frac{\sigma_v^2}{4})$ (low noise) and $\mathcal{N}(2\cdot\mu_v,\ 4\cdot\sigma_v^2)$ (high noise) as illustrated in Figure 12 for Gaussian additive noise. Note that higher noise causes more fluctuation to the original data. Other noise models are treated similarity, i.e. values following certain distributions are added to original acceleration signals.

Finally, we extracted fingerprints of the augmented data to evaluate their similarity. Figure 13 details the similarity achieved for intra-body, inter-body, and video-based acceleration sequences with three noise levels. We assessed the effectiveness of video-based attacks on four quantization schemes. The estimation based on the video noise level is able to generate fingerprints which are sufficiently close to the actually recorded acceleration sequence, so that this attack can break the gait-based pairing protocol for all three noise distributions considered. Walkie-Talkie [19] is the most sensitive protocol under video-based attacks. That means in this scheme attackers have the highest chance to obtain pairing keys if an accurate and real-time object tracking system is employed. On the other hand, SAPHE [18] is the most secure protocol against video-based attackes.

6 POTENTIAL IMPROVEMENTS

As shown, SAPHE is a promising approach as it introduces randomness instead of just observing it. As a potential improvement to our current implementation with a range of $1g$, we propose to implement a dynamic range. This would prevent threshold values that are outliers and produce the same bit independent of the acceleration. Due to SAPHE's quantization, an attack, where a simple sinusoidal acceleration signal is artificially generated in alignment with the heel-strike, might lead to a good estimate of the key. We propose to choose the threshold values as close to the acceleration reading as possible while still not revealing the actual unique gait features. This could be achieved by filtering out the dominant gait frequencies. Finally, instead of using hashed heuristic trees [18], we propose the usage of extensively studied cryptographic building blocks, such as fuzzy cryptography and a Password Authenticated Key Exchange.

As discussed in Section 3, the quantization approach of BANDANA is biased towards specific patterns which are generated significantly more often than others. A straightforward solution to this problem is to disregard these 4-bit patterns with probability inverse to their occurrence frequency. However, due to the significant distortion of the histogram (cf. Figure 10b), this is not feasible. Since some patterns occur with a frequency of only 1% or less, close to all frequent patterns would have to be discarded to arrive at a balanced random distribution.

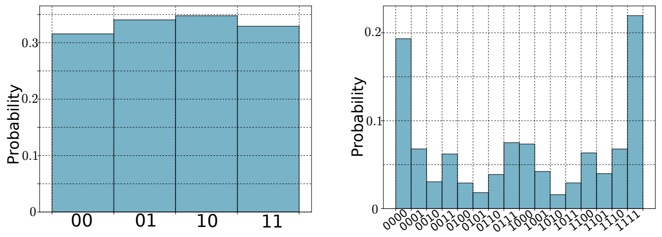
As one feasible solution, we investigate a mapping of each pair of consecutive bits in the generated key sequence to a single bit ($01, 11 \rightarrow 1,\ 10, 00 \rightarrow 0$)[8]. Figure 14a and 15a show the distribution of bit sequences in patterns after the mapping as well as the heatmap for fingerprints generated with the modified quantization protocol.

We see that the weakness described in Section 3 could be mitigated. However, note that, due to the strong unbalancedness, some bias still remains even after the mapping as depicted in the histogram in Figure 15b. A further mapping can reduce this bias, however, this process also increases the time required to generate a particular key sequences as well as the similarity for intra-body pairings (cf Figure 16a).

Another solution is to modify the comparison of gait sequences. The mean gait features an average amplitude with

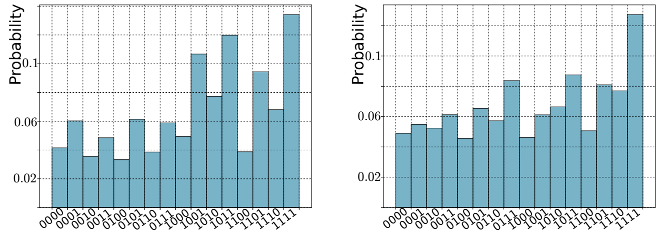
---

[8] Note that this does not help an adversary as the occurrence of 01 and 10 (11 and 00) sequences are equally probable due to the symmetry in the histogram in Figure 10b



(a) Mapping approach – 2 bits/bin

(b) Mapping approach – 4 bits/bin



(c) Normalization approach – normalizing acceleration amplitudes

(d) Normalization approach – additional disregarding of patterns according to inverse occurrence probabilities

Fig. 15: BANDANA improvements: Histograms generated from different improved versions of BANDANA

respect to the instantaneous gait sequences. Also, the acceleration peaks of the instantaneous gait fall with about equal probability to the left or right of the mean gait sequence. Consequently, the quantization, exploiting the difference between mean and instantaneous gait generates 0101 and 1010 patterns more often than other patterns. We suggest to normalize both mean and instantaneous gait prior to comparing them for gait generation. The heatmap and histogram for bit sequences generated with this modified versions are depicted in figures 14b and 15c. We observe that the distribution is significantly improved. Unfortunately, a bias towards including more '1'-s is introduced. However, since this bias is less severe than in the original BANDANA protocol, it can be addressed by disregarding patterns with probability inverse to their observed occurrence frequency (cf. Figure 15d). We observe in Figure 16b that the similarity for intra-body pairing is slightly reduced.

7 CONCLUSION

We have analyzed the quantization approaches of four popular recent acceleration-based pairing schemes. For this, we have compared their quantization schemes and discussed quantization-specific attacks on the protocols. Furthermore, their on-body pairing performance, statistical properties and entropy of generated key sequences have been investigated based on walking data from 15 subjects and with devices located at 7 on-body locations.

In particular, the SAPHE protocol was designed to pair devices that share acceleration sequences e.g. because they are shaken together. Although it could therefore not achieve similar sequences across different locations on the body, its security properties, distribution of generated keys and statistical properties exceed those of the other protocols.

The Walkie-Talkie protocol, which is able to generate the highest number of key bits from the gait acceleration achieves exact matching keys only across upper body locations and with low confidence. Together with the SAPHE protocol, it has the lowest one-shot success probability. This is, however, put into different perspective by a design flaw in the protocol. Even a naive adversary is able to boost her success probability to 0.125 by analysing the communication during the pairing process.

The BANDANA protocol is specifically designed for on-body pairing between devices and produces high similarity between sequences generated for different and also remote locations on the same body. However, the keys generated are insufficiently distributed and show a bias towards specific binary patterns. This problem originates from the quantization approach utilized and we proposed alternative quantization mechanisms that fix these issues.

Finally, the IPI protocol is also able to achieve high similarity across keys generated at different location on the same body. Our investigation revealed that the protocol suffers from a low variance in the generated binary patterns, so that similarity is also high for random gait sequences.

We further analyzed the threat of a video-based attack on gait authentication and gait-based pairing schemes and found that a sophisticated attacker with video support is able to estimate gait sequences sufficiently well to break the studied gait-based pairing approaches.

ACKNOWLEDGMENT

We appreciate partial funding in the frame of an EIT Digital HII Active project, as well as from the Academy of Finland and from the German Academic Exchange Service (DAAD).

REFERENCES

[1] B. Guo, D. Zhang, Z. Wang, Z. Yu, and X. Zhou, "Opportunistic iot: Exploring the harmonious interaction between human and the internet of things," *Journal of Network and Computer Applications*, vol. 36, no. 6, pp. 1531–1539, 2013.

[2] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, "Toward massive machine type cellular communications," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 120–128, 2017.

[3] S. Sigg, D. Schürmann, and Y. Ji, "Pintext: A framework for secure communication based on context," in *MobiQuitous 2011*, 2011.

[4] R. D. Findling, M. Muaaz, D. Hintze, and R. Mayrhofer, "ShakeUnlock: Securely Transfer Authentication States Between Mobile Devices," *IEEE Trans. on Mobile Computing*, vol. PP, no. 99, 2016.

[5] D. Schürmann and S. Sigg, "Secure communication based on ambient audio," *IEEE Trans. mobile computing*, vol. 12, no. 2, 2013.

[6] R. Jin, L. Shi, K. Zeng, A. Pande, and P. Mohapatra, "MagPairing: Pairing Smartphones in Close Proximity Using Magnetometers," *IEEE Trans. on Information Forensics and Security*, vol. 11, no. 6, 2016.

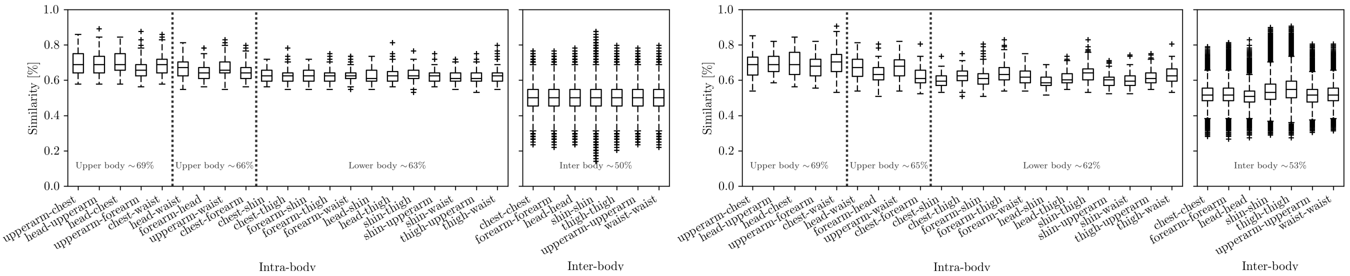
[7] S. Mathur, R. Miller, A. Varshavsky, W. Trappe, and N. Mandayam, "Proximate: proximity-based secure pairing using ambient wireless signals," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*. ACM, 2011, pp. 211–224.

[8] M. Muaaz and R. Mayrhofer, "Smartphone-based gait recognition: From authentication to imitation," *IEEE Trans. on Mobile Computing*, 2017.

[9] K. Kunze, "Compensating for on-body placement effects in activity recognition," Ph.D. dissertation, Citeseer, 2011.

[10] J. E. Cutting and L. T. Kozlowski, "Recognizing friends by their walk: Gait perception without familiarity cues," *Bulletin of the Psychonomic Society*, vol. 9, no. 5, pp. 353–356, 1977.

[11] L. Rong, D. Zhiguo, Z. Jianzhong, and L. Ming, "Identification of individual walking patterns using gait acceleration," in *Bioinformatics and Biomedical Engineering, 2007. ICBBE 2007. The 1st International Conference on*. IEEE, 2007, pp. 543–546.



(a) Mapping approach

(b) Normalization approach

Fig. 16: BANDANA improvements: Comparison of intra-body against inter-body similarity for our proposed improvements

[12] D. Gafurov, "A survey of biometric gait recognition: Approaches, security and challenges," in *Annual Norwegian computer science conference*, 2007, pp. 19–21.

[13] A. Jain, P. Flynn, and A. A. Ross, *Handbook of biometrics*. Springer Science & Business Media, 2007.

[14] D. Gafurov, E. Snekkenes, and P. Bours, "Spoof attacks on gait authentication system," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 491–502, 2007.

[15] M. L. Johnson, "Biometrics and the threat to civil liberties," *Computer*, vol. 37, no. 4, pp. 90–92, 2004.

[16] D. Schürmann, A. Brüsch, S. Sigg, and L. Wolf, "BANDANA – Body Area Network Device-to-device Authentication using Natural gAit," in *IEEE PerCom*, Mar. 2017, pp. 190–196.

[17] Y. Sun, C. Wong, G.-Z. Yang, and B. Lo, "Secure key generation using gait features for body sensor networks," in *IEEE BSN, 2017*, 2017, pp. 206–210.

[18] B. Groza and R. Mayrhofer, "SAPHE: simple accelerometer based wireless pairing with heuristic trees," in *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*. ACM, 2012, pp. 161–168.

[19] W. Xu, G. Revadigar, C. Luo, N. Bergmann, and W. Hu, "Walkie-talkie: Motion-assisted automatic key generation for secure on-body device communication," in *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2016.

[20] M. Nixon, J. Carter, D. Cunado, P. Huang, and S. Stevenage, "Automatic gait recognition," in *Biometrics*. Springer, 1996, pp. 231–249.

[21] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 2, pp. 316–322, 2006.

[22] Z. Liu and S. Sarkar, "Improved gait recognition by gait dynamics normalization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 6, pp. 863–876, 2006.

[23] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer, "The humanid gait challenge problem: data sets, performance, and analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 162–177, 2005.

[24] N. V. Boulgouris, D. Hatzinakos, and K. N. Plataniotis, "Gait recognition: a challenging signal processing technology for biometric identification," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 78–90, 2005.

[25] S. A. Niyogi and E. H. Adelson, "Analyzing gait with spatiotemporal surfaces," in *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, 1994, pp. 64–69.

[26] J. E. Boyd, "Synchronization of oscillations for machine perception of gaits," *Computer Vision and Image Understanding*, vol. 96, no. 1, pp. 35–59, 2004.

[27] M. S. Nixon and J. N. Carter, "Advances in automatic gait recognition," in *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*. IEEE, 2004, pp. 139–144.

[28] A. Veeraraghavan, A. K. Roy-Chowdhury, and R. Chellappa, "Matching shape sequences in video with applications in human movement analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1896–1909, 2005.

[29] A. Y. Johnson and A. F. Bobick, "A multi-view method for gait recognition using static body parameters," in *International Conference on Audio-and Video-Based Biometric Person Authentication*. Springer, 2001, pp. 301–311.

[30] D. Gafurov, "Performance and security analysis of gait-based user authentication," Ph.D. dissertation, University of Oslo, 2008.

[31] S. J. Morris, "A shoe-integrated sensor system for wireless gait analysis and real-time therapeutic feedback," Ph.D. dissertation, Massachusetts Institute of Technology, 2004.

[32] B. Huang, M. Chen, P. Huang, and Y. Xu, "Gait modeling for human identification," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 4833–4838.

[33] H. J. Ailisto, M. Lindholm, J. Mantyjarvi, E. Vildjiounaite, and S.-M. Makela, "Identifying people from gait pattern with accelerometers," in *Defense and Security*. International Society for Optics and Photonics, 2005, pp. 7–14.

[34] L. Rong, Z. Jianzhong, L. Ming, and H. Xiangfeng, "A wearable acceleration sensor system for gait recognition," in *IEEE Conference on Industrial Electronics and Applications*, 2007, pp. 2654–2659.

[35] E. Vildjiounaite, S.-M. Mäkelä, M. Lindholm, R. Riihimäki, V. Kyllönen, J. Mäntyjärvi, and H. Ailisto, "Unobtrusive multimodal biometrics for ensuring privacy and information security with personal devices," in *Int. Conference on Pervasive Computing*. Springer, 2006, pp. 187–201.

[36] A. Kale, N. Cuntoor, B. Yegnanarayana, A. Rajagopalan, and R. Chellappa, "Gait analysis for human identification," in *International Conference on Audio-and Video-Based Biometric Person Authentication*. Springer, 2003, pp. 706–714.

[37] W. Wang, A. X. Liu, and M. Shahzad, "Gait recognition using wifi signals," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '16. New York, NY, USA: ACM, 2016, pp. 363–373.

[38] Y. Zeng, P. H. Pathak, and P. Mohapatra, "Wiwho: Wifi-based person identification in smart spaces," in *International Conference on Information Processing in Sensor Networks*, April 2016.

[39] J. Jenkins and C. Ellis, *Using Ground Reaction Forces from Gait Analysis: Body Mass as a Weak Biometric*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 251–267.

[40] K. Nakajima, Y. Mizukami, K. Tanaka, and T. Tamura, "Footprint-based personal recognition," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 11, pp. 1534–1537, 2000.

[41] R. J. Orr and G. D. Abowd, "The smart floor: A mechanism for natural user identification and tracking," in *CHI'00 extended abstracts on Human factors in computing systems*. ACM, 2000, pp. 275–276.

[42] L. Middleton, A. A. Buss, A. Bazin, and M. S. Nixon, "A floor sensor system for gait recognition," in *Workshop on Automatic Identification Advanced Technologies*. IEEE, 2005, pp. 171–176.

[43] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception & Psychophysics*, vol. 14, no. 2, pp. 201–211, 1973.

[44] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer, "The humanid gait challenge problem: Data sets, performance, and analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 2, pp. 162–177, 2005.

[45] Y. Wang, S. Yu, Y. Wang, and T. Tan, "Gait recognition based on fusion of multi-view gait sequences," *Advances in Biometrics*, pp. 605–611, 2005.

[46] T. Lam and R. Lee, "A new representation for human gait recognition: Motion silhouettes image," *Advances in Biometrics*, pp. 612–618, 2005.

[47] M. S. Nixon, T. Tan, and R. Chellappa, *Human identification based on gait*. Springer Science & Business Media, 2010, vol. 4.

[48] T. Hoang, D. Choi, and T. Nguyen, "Gait authentication on mobile phone using biometric cryptosystem and fuzzy commitment

scheme," *International Journal of Information Security*, vol. 14, no. 6, pp. 549–560, 2015.

[49] M. O. Derawi, C. Nickel, P. Bours, and C. Busch, "Unobtrusive user-authentication on mobile phones using biometric gait recognition," in *Intelligent Information Hiding and Multimedia Signal Processing, Sixth International Conference on*, 2010, pp. 306–311.

[50] M. O. Derawi, "Smartphones and biometrics: gait and activity recognition," 2012.

[51] R. Kumar, V. V. Phoha, and A. Jain, "Treadmill attack on gait-based authentication systems," in *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2015.

[52] A. Kwong, C. Bolton, T. Trippel, W. Xu, and K. Fu, "Why do you trust sensors? analog cybersecurity attack demos."

[53] B. B. Mjaaland, P. Bours, and D. Gligoroski, "Walk the walk: attacking gait biometrics by imitation," in *International Conference on Information Security*. Springer, 2010, pp. 361–380.

[54] Ø. Stang, "Gait analysis: Is it easy to learn to walk like someone else?" Master's thesis, 2007.

[55] M. Muaaz and R. Mayrhofer, "An analysis of different approaches to gait recognition using cell phone based accelerometers," in *Proceedings of International Conference on Advances in Mobile Computing & Multimedia*. ACM, 2013, p. 293.

[56] L. Sloman, M. Berridge, S. Homatidis, D. Hunter, and T. Duck, "Gait patterns of depressed patients and normal subjects." *The American journal of psychiatry*, 1982.

[57] A. Srivastava, J. Gummeson, M. Baker, and K.-H. Kim, "Step-by-step detection of personally collocated mobile devices," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*. ACM, 2015, pp. 93–98.

[58] E. A. Heinz, K. S. Kunze, S. Sulistyo, H. Junker, P. Lukowicz, and G. Tröster, "Experimental evaluation of variations in primary features used for accelerometric context recognition," in *European Symposium on Ambient Intelligence*. Springer, 2003, pp. 252–263.

[59] J. Lester, B. Hannaford, and G. Borriello, *"Are You with Me?"–Using Accelerometers to Determine If Two Devices Are Carried by the Same Person*. Berlin, Heidelberg: Springer, 2004, pp. 33–50.

[60] C. Cornelius and D. Kotz, "Recognizing whether sensors are on the same body," in *Pervasive'11*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 332–349.

[61] R. D. Findling, M. Muaaz, D. Hintze, and R. Mayrhofer, "Shakeunlock: Securely unlock mobile devices by shaking them together," in *Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia*. ACM, 2014, pp. 165–174.

[62] R. Mayrhofer, "The candidate key protocol for generating secret shared keys from similar sensor data streams," in *European Workshop on Security in Ad-hoc and Sensor Networks*. Springer, 2007, pp. 1–15.

[63] T. Sztyler and H. Stuckenschmidt, "On-body Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition," in *IEEE International Conference on Pervasive Computing and Communications (PerCom'16)*. IEEE, 2016, pp. 1–9.

[64] W. Xu, C. Javali, G. Revadigar, C. Luo, N. Bergmann, and W. Hu, "Gait-key: A gait-based shared secret key generation protocol for wearable devices," *ACM Trans. Sen. Netw.*, vol. 13, no. 1, pp. 6:1–6:27, Jan. 2017.

[65] T. Hoang, D. Choi, and T. Nguyen, "Gait authentication on mobile phone using biometric cryptosystem and fuzzy commitment scheme," *International Journal of Information Security*, vol. 14, no. 6, pp. 549–560, 2015.

[66] Y. Wang, "On stochastic security of pseudorandom sequences."

[67] N. Kuiper, "Tests concerning random points on a circle," in *Proceedings of the Koinklijke Nederlandse Akademie van Wetenschappen*, vol. Series a 63, 1962, pp. 38–47.

[68] R. M. Bolle, J. Connell, S. Pankanti, N. K. Ratha, and A. W. Senior, *Guide to biometrics*, 2013.

[69] M. Muaaz and R. Mayrhofer, "Smartphone-based gait recognition: From authentication to imitation," *IEEE Trans. on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2017.

[70] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, "Vnect: Real-time 3d human pose estimation with a single rgb camera," vol. 36, no. 4, 2017.

Arne Brüsch received his B.Sc. degree in computer science in 2015. Currently, he is a master's student at TU Braunschweig. His research interests include quantization schemes for key generation in usable security as well as contextual security in general.

Ngu Nguyen is a doctoral student at Ambient Intelligence Group, Aalto University. He completed his bachelor's and master's degree at University of Science, Ho Chi Minh City, Vietnam. His research focuses on usable security and distributed machine learning.

Dominik Schürmann received the B.Sc. and M.Sc. degrees in 2010 and 2014 respectively, from TU Braunschweig. Since 2014, he works as a research fellow at the Institute of Operating Systems and Computer Networks at TU Braunschweig, where he is also pursuing a Ph.D. degree. His research interests include interaction-free security based on physical context and usable security in general.

Stephan Sigg received his M.Sc. degree in computer science from TU Dortmund, in 2004. and his Ph.D. degree from Kassel University, in 2008. Since 2015 he is an assistant professor at Aalto University, Finland. He has served as a TPC member of many conferences including IEEE PerCom, Ubicomp, etc. His research interests include Pervasive Computing, activity recognition, usable security and optimization of algorithms in mobile distributed systems.

Lars Wolf received his Ph.D. in 1995. In 1999 he was an associated professor at the computer science department of Universität Karlsruhe (TH). Since spring 2002 Lars Wolf is full professor for computer science at the TU Braunschweig where he is head of the Institute of Operating Systems and Computer Networks. His current research interests include wireless networking in general, sensor networks, vehicular networks, delay-tolerant networks, and mobile systems.

OpenKeychain: An Architecture for Cryptography with Smart Cards and NFC Rings on Android

DOMINIK SCHÜRMANN, TU Braunschweig, Germany
SERGEJ DECHAND, University of Bonn, Germany
LARS WOLF, TU Braunschweig, Germany

While many Android apps provide end-to-end encryption, the cryptographic keys are still stored on the device itself and can thus be stolen by exploiting vulnerabilities. External cryptographic hardware solves this issue, but is currently only used for two-factor authentication and not for communication encryption.

In this paper, we design, implement, and evaluate an architecture for NFC-based cryptography on Android. Our high-level API provides cryptographic operations without requiring knowledge of public-key cryptography. By developing OpenKeychain, we were able to roll out this architecture for more than 100,000 users. It provides encryption for emails, messaging, and a password manager. We provide a threat model, NFC performance measurements, and discuss their impact on our architecture design. As an alternative form factor to smart cards, we created the prototype of an NFC signet ring. To evaluate the UI components and form factors, a lab study with 40 participants at a large company has been conducted. We measured the time required by the participants to set up the system and reply to encrypted emails. These measurements and a subsequent interview indicate that our NFC-based solutions are more user friendly in comparison to traditional password-protected keys.

CCS Concepts: • **Security and privacy → Usability in security and privacy**; *Key management*; *Hardware-based security protocols*; • **Human-centered computing** → *Mobile devices*;

Additional Key Words and Phrases: NFC, near-field communication, smart card, ring

ACM Reference Format:
Dominik Schürmann, Sergej Dechand, and Lars Wolf. 2017. OpenKeychain: An Architecture for Cryptography with Smart Cards and NFC Rings on Android. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 99 (September 2017), 24 pages.
https://doi.org/10.1145/3130964

1 INTRODUCTION

Nowadays, it is widely acknowledged that smartphones hold the most sensitive information, such as emails, short messages, and photos. Besides private usage, the same devices are often used for accessing privileged company information due to Bring Your Own Device (BYOD) policies. Privacy-aware app developers take this into account and provide apps for secure messaging, encrypted cloud storage, and other use cases. Unfortunately, secret keys generated by these apps are unprotected and stored on the internal flash memory. Thus, an attacker can fully compromise end-to-end security by retrieving secret keys through privilege escalation exploits or direct physical access. While this is a well-known problem, the security of many apps and the mobile operating system is subpar [82]. Since the Stagefright-bug [85], Google started rolling out monthly Over-The-Air (OTA) updates [53] documented in Nexus Security Bulletins [4]. Unfortunately, not all fixes are backported for devices of other Original Equipment Manufacturers (OEMs). Even if the system is up-to-date, widespread vulnerabilities

© 2017 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery.
This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, https://doi.org/10.1145/3130964.

in secure email and messaging apps have been found, potentially exposing secret keys to attackers [78, 82]. Even worse, app developers often lack knowledge of cryptography, which leads to insecure implementations [25, 29].

Traditional desktop encryption software provides protection against key theft by encrypting keys using high-entropy passwords. One of the main reasons this is not done on mobile devices is due to the difficulty of entering passwords on mobile on-screen keyboards. One often recommended password alternative are biometric fingerprints. These can be used since Android 6 with the Keystore API. While they are suitable for locking devices, they should not be used alone to protect secret keys. They do not provide enough entropy to reach the same strength as passwords or hardware-backed solutions [49, 56].

For desktop systems, external hardware in form of smart cards exist which can be used with USB card readers. They replace password-protected key files with external cards and corresponding PINs. In contrast to traditional key files, which are stored on the same device where the password is entered, the secret key of a smart card is stored physically separated from the device where the PIN is entered. For smartphones, On-The-Go USB cables or other external peripherals, such as hardcovers with card slots, exist. Because these are unwieldy and, thus, unsuitable for day-to-day usage, Near-Field Communication (NFC) has been proposed for smart card communication. A small selection of NFC tokens, which are capable of encryption and signature generation and not only authentication is available. Still, no high-level cryptographic NFC API was available.

1.1 Contributions

In this paper, we introduce and evaluate an architecture for NFC-based cryptography on Android via external security tokens, such as smart cards. First, we discuss the problem setting by looking into currently available NFC hardware and software libraries to derive a clear set of requirements. In the threat model we consider for this work, three main attack areas have been identified: NFC, security token, and smartphone. All attacks are evaluated in comparison to traditional password-protected secret keys. Based on our requirements and threat model, we carefully design our architecture and, thus, make the following contributions in this paper:

(1) Our main contribution is the design of a high-level API for Android, which not only includes cryptographic primitives but also a variety of pre-defined UI components for common end-user interactions. Previously available cryptographic NFC APIs have been designed for authentication only.

(2) For developers, we provide a set of API methods, which can be used without knowledge of public-key cryptography. Keys are handled transparently independent of their storage location (password-protected key file or via NFC). In addition to the card's PIN authentication, we provide a security layer via an Android app that handles the PIN input and allows only certain apps access to cryptographic methods, which have been explicitly granted by the user. This way, under the assumptions of Android's security model, only one trusted app handles the PIN input, while many semi-trusted apps can access cryptographic methods.

(3) As one of the main developers of *OpenKeychain*, an encryption app for Android, we were able to roll out this architecture for more than 100,000 devices. Its API is used by several other apps, such as *K-9 Mail*, the XMPP client *Conversations*, and the password manager *Password Store*.

(4) As an alternative form factor, we consider a NFC ring. Because no NFC rings with a cryptographic processor are available, we created our own prototype using an NXP Integrated Circuit (IC) and a 3D printer.

(5) To evaluate the end-user usability of our architecture in combination with password-protected keys, NFC cards, and NFC rings, we conducted a usability lab study with 40 participants outside of the university environment in cooperation with the IT security department of a large company. As a use case scenario we implemented end-to-end encrypted email communication. Finally, an interview has been conducted to evaluate user perception and form factor acceptance.

2 RELATED WORK

The work presented in this paper touches the research areas of (a) usability of end-to-end encryption, (b) cryptographic API design, and (c) usability of security tokens, especially in conjunction with smartphones.

While our API supports a wide range of differing use cases, end-to-end email encryption is one prominent example, which we also included as an experiment in our study. The famous "Why Johnny Can't Encrypt" [81] publication discusses a case study on PGP 5.0 and concluded that end-to-end email encryption software was still not usable enough for most end-users. This led to several user studies over the few last years, e.g., by Garfinkel et al. [35, 36] and Fahl et al. [30]. They studied larger groups and proposed how email encryption or Facebook encryption can be made usable.

While there is a broad range of research on usable security for end-users, developer usability is often neglected. It has been shown that in a set of 12000 apps at least 88 % include at least one cryptographic error [25]. Similarly, developers often override safe defaults due to being unaware of the implications as shown in the evaluation of SSL apps by Fahl et al. [31]. To prevent these incidents from happening, high-level cryptographic APIs, such as NaCl [7], Sodium [20], and Keyczar [21], have been published in the last years. High-level 'crypto-box'-methods, based on a fixed set of algorithms, are provided that execute several steps at once, which are usually done individually when using low-level APIs such as Java Cryptography Extension (JCE) or Bouncy Castle [11]. Still, these libraries provide no automated way of handling password/PIN input and are not designed to fit into the developer ecosystem of mobile operating systems such as Android.

Research on the usability of security tokens for asymmetric cryptography in the context of encryption of emails or instant messages is limited. Most publications focus on authentication not encryption with security tokens, in particular smart cards. Sasse concludes that smart cards can offer usability benefits compared to password authentication [75]. Strouble et al. conducted a survey, answered by 300 participants, evaluating the usability of smart cards [79]. A notable result is that 67 % left their smart cards in the USB reader at least once, which increases the possibility of theft. Paul et al. conducted a field study with 24 participants over 10 weeks to evaluate the user behavior and perceptions in regards to smart cards [67]. They conclude that "The greatest perceived benefit was the use of an easy-to-remember PIN in replacement of complicated passwords. The greatest perceived drawback was the lack of smartcard-supported applications" [67]. Recently, Google compared their 'Security Keys' with other hardware tokens, passwords alone, and two factor authentications using smartphone apps [52] using the usability framework by Bonneau et al. [10]. They measured the raw performance and found out that users are twice as fast authenticating via 'Security Keys' in comparison to in-app One Time Passwords. Failure rates dropped from 3 % with OTP to 0 %. Taking into account the burden of physically carrying around authentication hardware, Mare et al. found that participants greatly differ in their preference regarding form factors [54]. To the best of the authors' knowledge, no studies are considering the usage of NFC security tokens, in our case NFC cards and rings, for end-to-end encryption instead of authentication on smartphones.

Available security tokens, which are qualified for our architecture, must include a cryptographic processor for asymmetric operations, such as RSA or ECC, and a NFC interface. For desktop systems with USB or card readers, tokens such as NitroKey [62], variants of YubiKey [83] (w/ OpenPGP support), and Java Card OpenPlatform (JCOP) smart cards with ICs by NXP are available. A number of smartphone accessories exist, e.g., back covers like the BlackBerry Smart Card Reader [72], modern variants like the Smart Card Reader by Precise Biometrics [69] (~100 EUR), and the Smart Fold Android Contact Smart Card Reader [46]. Due to the cost and bulkiness of external peripherals, NFC security tokens are considered. Only a limited number of the presented ones have an NFC interface, such as YubiKey NEO [83], the Fidesmo Card [32], and NXP developer cards with dual interface.

In addition to evaluating the traditional card form factor, we consider the usage of wearable NFC rings. A first example from 1998 is the Java Ring by Dallas Semiconductor [18] running an early version of the Java Card environment. During the following years, this form factor has been proposed for different communication scenarios [16, 60, 71, 73]. A famous commercially available NFC ring has been designed by McLear and marketed via Kickstarter [58]. To the best of the authors' knowledge, no consumer-ready NFC ring supporting asymmetric cryptography over NFC together with an implementation was available until the first author started the work outlined in this paper.

3 PROBLEM SETTING

As discussed in the introduction, important secret keys should not be kept on internal flash storage as they can be stolen using widely available exploits. Instead, it is good practice to store those on external security tokens. In this section, we discuss related security and usability problems to derive a set of requirements for our architecture.

Cryptographic operations should be executed via a connection between smartphone and token, while the secret key itself should never be exposed to the smartphone. This connection is traditionally established via card readers, which require On-The-Go USB cables or external peripherals to work with modern smartphones. Also, smart cards are often left inside the reader [79], which poses a security risk. Communication over NFC has been proposed as a mobile alternative but no API is available for encryption/digital signature generation. Even though a small selection of security tokens with NFC and cryptographic processor are available, these are designed as credit cards or USB sticks. While other form factors have been proposed for simple NFC tags with read/write capability, these are not available with cryptographic processors. Furthermore, while the usage of security tokens has been evaluated for two-factor authentication, no studies exist in the context of end-to-end encryption. Thus, it is not known how users perceive the usage of NFC in the context of email encryption for example. It is also not known if other form factors could improve the acceptance of tokens or their usability.

When using NFC, security tokens must be held against the device's NFC antenna for differing durations depending on the cryptographic operation. Users can easily get frustrated if these are too long. Access control to operations should be done by entering a numeric PIN on the device that is easy to remember. To restrict brute force attempts, the token should deactivate after a number of failed attempts and a special Admin PIN must allow the user to re-activate the token. In available implementations for desktop operating systems with card readers, the users are not properly guided through the selection of an appropriate PIN and Admin PIN. For example, in GnuPG [26] and Enigmail [14] users are not forced to change the default PINs after key generation.

Not all apps should have access to the PIN to execute arbitrary cryptographic operations while the security token is held against the NFC antenna. Also, in traditional desktop implementations, it is not possible to restrict the execution of cryptographic operations to a specific set of client applications. Thus, client applications often handle PIN input directly, even though they may not be trusted fully. Furthermore, usage of secret keys is not restricted in any way, as long as a contact smart card is inserted into the reader. Secret keys for special purposes or higher levels of classification are not protected differently than other keys. Currently, no high-level API exists for smartphones that supports cryptographic operations, but also securely handles PIN/password caching and provides user interaction for common functionality, such as public key import. All available low-level APIs integrated in mobile operating systems such as Java's Cipher API [2] on Android, whose internals are based on Bouncy Castle/OpenSSL, as well as iOS Cryptographic Services [5] require a substantial effort from the developer. This naturally leads to vulnerabilities implemented over and over in many apps as well as to re-implementations of the same functionality in different contexts, such as password input as well as caching layers. Furthermore, they do not provide re-usable UI components. Even high-level APIs, such as NaCl [7], Sodium [20], and Keyczar [21], require a substantial amount of app-specific code to handle password input and caching, migrations from older key algorithms, and user to key mappings.

In addition, these are designed without special hardware in mind, i.e., no external security tokens are supported out of the box. Thus, keys stored on security tokens must be handled completely different and require more complexity than keys stored on the device, e.g., when using Android APIs [27].

3.1 Requirements

Based on the outlined problem setting, a set of architectural requirements is derived:

- R_1 Support for multiple form factors without external smartphone accessories
- R_2 Short durations of cryptographic NFC operations
- R_3 PIN input and caching solely handled by a trusted entity on the smartphone
- R_4 High-level versionable cryptographic API including UI components for common user interactions, secure defaults, and standardized packet formats
- R_5 Access control on the token-side by numeric PIN/Admin PIN and on the app-side by restriction of secret keys to specific clients
- R_6 Transparent handling of secret keys independent of their storage location

3.2 Threat Model

The main advantage of security tokens is that its key storage is physically separated from the mobile device. The hardware and firmware of the security token does not provide an API to retrieve secret keys, only cryptographic operations are exposed executed on the processor of the token. In the following, the discussed threat model is subdivided by the attacked entity, i.e., NFC, security token, and smartphone. While we evaluate all scenarios relevant for the whole architecture, an emphasis lies on mitigations provided by our own contributions.

3.2.1 NFC. First, we discuss attacks against the NFC connection itself that is established between the smartphone as an active initiator and the security token as a passive target.

Denial of Service Since radio jamming in general is difficult to prevent, NFC lacks sophisticated countermeasures. However, simply preventing communication is of low value to an attacker. In particular, signing and decrypting emails is not a time-critical activity and can, thus, tolerate short-term disruptions. More importantly, this attack does not put the security of the secret key at risk.

Relay Attack This attack is also called a wormhole or Mafia attack. In the field of NFC payment and authentication systems, a connection is established between the victim's smart card and an attacker's NFC reader. This connection is relayed over the Internet to a second device of the attacker to actually authenticate or pay at a different NFC reader physically far away [34]. This attack can potentially be executed unnoticed by holding the attacker's NFC reader against the victim's pocket containing the smart card. In our architecture, the security token is protected by a PIN. Thus, NFC relay attacks can only be executed if the PIN has been compromised beforehand.

Eavesdropping Kortvedt and Mjolsnes were able to eavesdrop on NFC in a range of up to 29 cm [51]. Brown et al. experimentally showed that eavesdropping capabilities largely depend on the amount of background noise [12]. Thus, if the attacker is very close to the victim and has the required equipment, it might be possible to extract the following information: In case of signature generation, a hash is transmitted and a signed hash is received. In case of decryption, an encrypted session key packet is transmitted and the decrypted session key is received. It is important to note that the plaintext that should be signed or the ciphertext that should be decrypted is never transmitted. Furthermore, the secret key never leaves the security token. Therefore, to decrypt an email with an eavesdropped session key, the encrypted email must also be intercepted at the corresponding email provider. Still, this is a valid attack scenario against targeted individuals. While our current prototype assumes channel security, for a future version we consider deploying the NFC-SEC standard [23, 24] that provides an Elliptic Curve Diffie Hellman key exchange with AES encryption. An application-level alternative for securing NFC has been proposed by Hölzl et al. using the Secure Remote Password (SRP-6a) protocol authenticated by a user-provided password [45].

Man-in-the-Middle (MitM) While eavesdropping might be possible in a certain range, MitM are extremely difficult because the attacker needs to block existing NFC. In detail, these attacks differ depending on the type of NFC connection: With an active-passive or passive-active connection, an attacker has to both block the originator's channel and to create an own RF field with perfect timing [42]. This is hard to achieve in practice and can possibly be detected by the user. In case of active-active connections, the interceptor has to completely block the communication between both partners without them noticing. Usually, NFC should abort if two RF fields exists, but is has been shown for the EMV protocol that some implementations do not follow the standard and it is possible to win the timing race [59]. Yet, this has not been shown for other protocols besides EMV. Conclusively, while Haselsteiner and Breitfuß [42] consider MitM attacks practically impossible, we at least consider them extremely difficult.

3.2.2 Security Token. In general, access to the security token is protected by a PIN with a length of 6 digits to protect against attackers in physical proximity. Thus, every cryptographic operation requires an authentication step. For the remaining attacks, we assume that the security token is protected against manipulations until it is received by the end user. Hence, the manufacturing process, warehousing, and shipping are considered to be secure, such that the initial key generation by the user is uncompromised. From this point on, though, the security token is vulnerable to theft and loss. As stated in the Introduction, there is no pre-deployed secret key on the security token, but the keys are generated by the user. In the following, we will mainly focus on attacks against the authentication step and hardware.

Brute Force PIN For memorability, we let the user chose the PIN, but prevent certain commonly chosen combinations, such as 123456. An attacker gaining access to the device can brute force up to 3 possible combinations of the PIN. After this, the security token is locked to prevent further brute forcing, and can only be unlocked entering the Admin PIN. In our architecture, the Admin PIN is not chosen by the user, instead it is securely-generated from random.

Physical attacks Due to theft or while the owner leaves the token unattended, an attacker can gain access to the security token. Physical attacks [80] aim to read, to modify or to erase data on the security token. Examples are provoking a power outage, examination with a probe station, chip re-wiring, as well as addition and cutting of a track. Given the physical access to the security token, these are generally difficult to defeat completely. Yet, they are typically expensive, destructive, and time consuming, especially since attacks are very target dependent. Additional protections against physical attacks, such as additional metal layers, bus scrambling, or on-board sensors can also be implemented on the hardware side. For these countermeasures, we rely on the security of the utilized NXP IC.

Side-Channel Attacks While being in close proximity to the owner, who currently uses the token with her smartphone, information about the cryptographic operation can be leaked by the token and smartphone. Timing attacks, for instance, exploit that the computing time of an operation differs with the used parameters, which in turn, can then be derived. As with physical attacks, we rely on the countermeasures provided by the IC. As discussed for an attacker who eavesdrop on NFC communication, also for side-channel attacks against the hardware, it could provide an additional advantage to monitor the corresponding email communication to correlate the decryption process with a particular message.

3.2.3 Smartphone. Recent vulnerabilities, such as the Stagefright bug [85], show the limited security on mobile devices. While local and remote software/firmware vulnerabilities are considered, we assume state-of-the-art cryptographic algorithms to be secure.

Physical Access An attacker, who gains physical access to a smartphone, while the owner leaves it unattended, is assumed to be able to download all data. The secret key, however, is never stored on the phone and, thus, not at risk. If no sophisticated attacks are performed, such as flashing a whole new operating

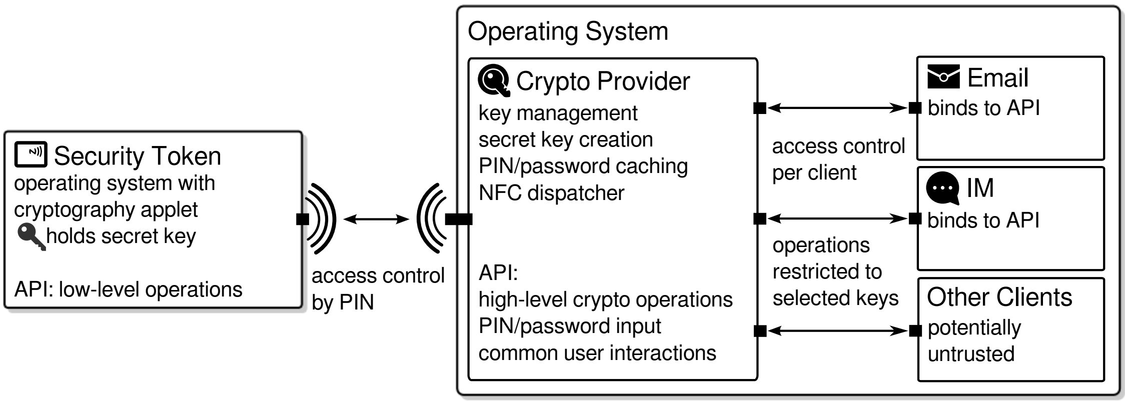


Fig. 1. Architecture Overview.

system, Android's system security prevents the installation of malware with root access. Still malicious apps with normal privileges could be installed. Due to the fact that PINs/passwords are only entered via a single trusted cryptography provider, malicious apps without root access cannot intercept these.

Vulnerabilities in Client Apps An attacker can try to exploit vulnerabilities in client apps that use our API. As described before, passwords and PINs cannot directly be retrieved as these are handled by the trusted cryptography provider only. Some vulnerabilities could potentially allow the attacker to trick the user into decrypting different content than originally opened by the user. In this case the password/PIN is properly entered by the user and the attack succeeds. To reduce the privacy impact in this scenario, a client app's API access is restricted to specific secret keys. Thus, only the keys selected for this client can be misused by an attacker.

Vulnerabilities in Android Critical vulnerabilities in Android can lead to exploits being used by an attacker to install malware with system access. In this case all installed apps are potentially insecure, even the PIN/password input and caching. Still, after detecting such a breach and removing the malware, future communications are secure, because the attacker was not able to retrieve the secret key.

UI Spoofing/Task Hijacking A malicious app could start a PIN/password dialog overlaying the original one and mimicking its design to intercept user secrets. This could be done by installing malware or a patched version of our cryptography provider. More sophisticated attacks building upon this scenario exploiting Android specific mechanisms are discussed by Cooley et al. [17] and Ren et al. [70]. As a future countermeasure we are plan to allow the user to set a personal image that will appear in all trustworthy UI components of our architecture. This has been implemented for example by Mailvelope [55].

GUI Side Channel Attacks As a special subcategory of attacks on Android, side channel attacks on the GUI of Android apps and hardware interrupts can potentially leak PINs/passwords to an attacker [15, 22]. As Diao et al. [22] remark, these side channels need to be closed on a system level by providing less runtime statistics to installed apps.

4 ARCHITECTURE

Considering the requirements and threat model, we propose a security architecture as depicted schematically in Figure 1. A security token, e.g., in credit card format, runs a smart card operating system together with an implementation of the cryptographic operations. The user's secret key is stored solely on this external token. It receives power via induction while holding it against the device and communicates with the operating system over NFC. The device's NFC interface is standardized and, thus, works with security tokens of multiple form factors (R_1). The token's API is protected via PIN authentication and provides all required cryptographic operations such as key generation, signature creation, and decryption. A single trusted cryptography provider is installed as an app on the smartphone's operating system. It implements all required low-level cryptographic operations as well as communication over NFC, optimized for short durations (R_2) and usability. PIN/password input and caching is done solely by this trusted cryptography provider (R_3). In addition to PIN/password input, several other common user interactions are supported to prevent re-implementations of the same interactions in different clients and decrease implementation complexity for client developers. An API is exposed to client developers providing high-level versioned cryptographic methods, e.g., a method for encryption combined with signatures, in the standardized OpenPGP message format (R_4). Access to this API is granted per app by user choice (R_5). The cryptography provider provides a key generation wizard that includes a secure selection of PIN/Admin PIN. Furthermore, it provides a unified way to transparently use secret keys without exposing their storage location or asymmetric algorithms (R_6).

4.1 Prerequisites

Our architecture is based on several existing technologies that are discussed briefly as prerequisites.

OpenPGP To integrate with existing protocols, the OpenPGP standard [13] has been chosen. It provides standardized email [28] and instant messaging [76, 77] encryption. Thus, it supports most common use cases with extensions for standardized communication protocols. OpenPGP support for smart cards has been standardized for ISO-compatible card operating systems [68] and primarily three open source implementations exist [33, 61, 84].

NFC Typically operating at 13.56 MHz, NFC is a wireless transmission technology for short ranges allowing active-active and active-passive modes. In our case, an active-passive connection is established, where the smartphone serves as the initiator and the security token is the passive target. The ISO 14443-4 [47] standard is used as the physical/link layer protocol between initiator and target and ISO 7816-4 [48] defines the basic structure of commands and Application Protocol Data Units (APDUs).

Operating System Support NFC support in Apple's current iOS 9 only supports NFC store loyalty cards as part of the Apple Pay API [6] and Windows Phone has only limited support for smart cards [1]. In contrast, Android's NFC API allows to exchange APDUs. It supports a foreground dispatch mode (≥ Android 2.3.3) and reader mode (≥ Android 4.4) that allow an app to manage the NFC connection without interfering with other installed NFC apps [3].

4.2 API Design

The presented architecture has been fully implemented for the Android operating system as part of OpenKeychain [65], an app implementing the OpenPGP standard. Currently, OpenKeychain has over 100,000 installations on Google Play and is also available via alternative stores, such as F-Droid. Besides the API proposed in this paper, OpenKeychain also provides encryption/decryption as well as signature generation/verification functionality of messages and files within the app. Since October 2015, a version has been released that was audited externally [43].

The following design provides a high-level API that complies with all requirements and can be used by other installed apps in a convenient way: In agreement with the OpenIntents project [64], the API definition lives in the namespace "org.openintents.openpgp". In contrast to similar architectures like JCE, these can also be chosen at runtime via app settings, i.e., the cryptographic backend for an email app can be provided by multiple implementations of the same high-level API.

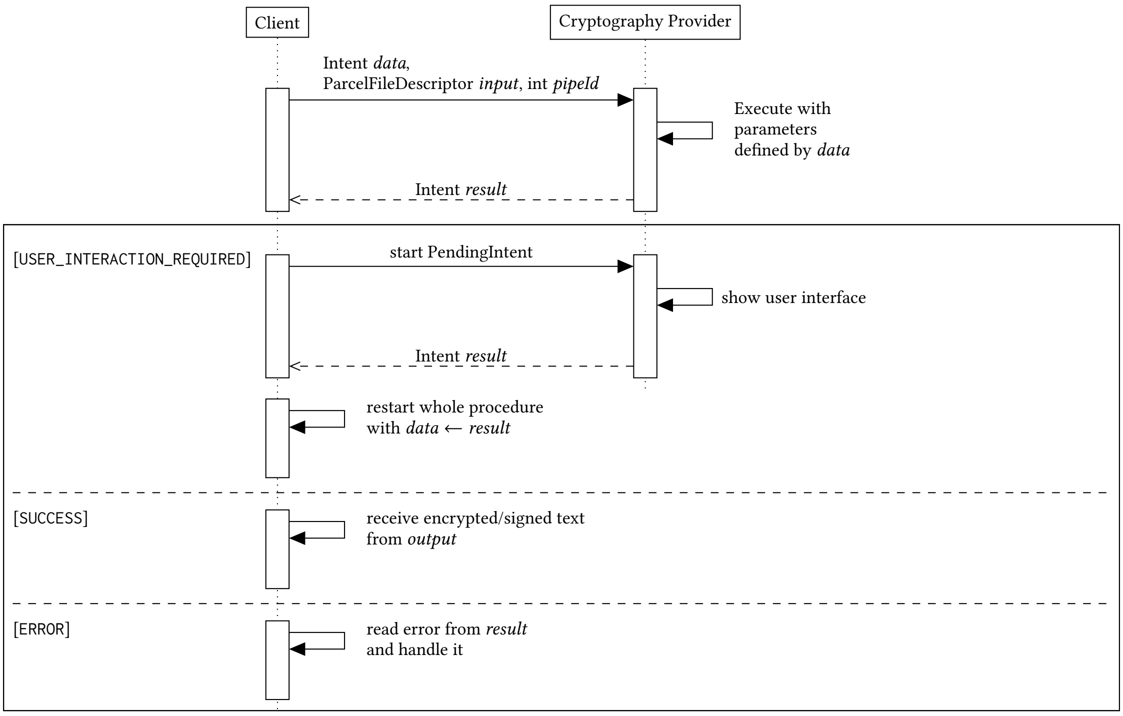
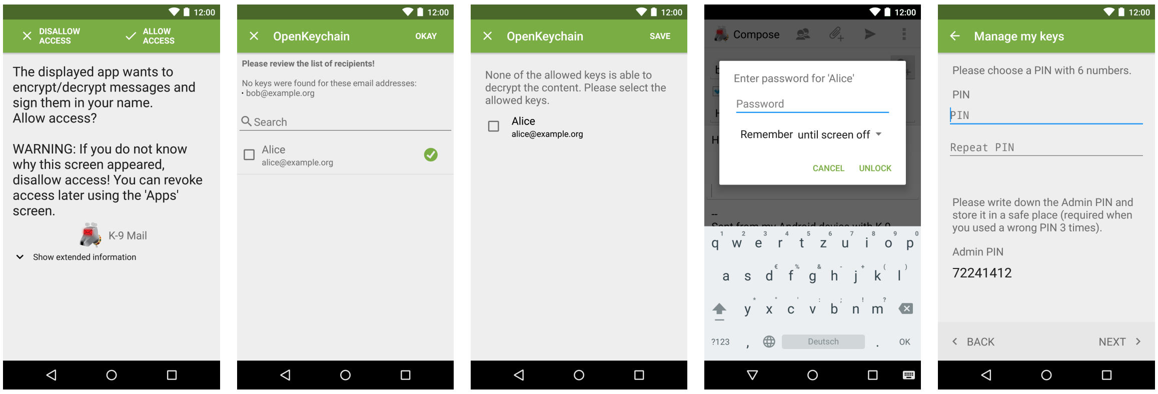

Client
Cryptography Provider
Intent *data*,
ParcelFileDescriptor *input*, int *pipeId*
Execute with parameters defined by *data*
Intent *result*
[USER_INTERACTION_REQUIRED]
start PendingIntent
show user interface
Intent *result*
restart whole procedure with *data* ← *result*
[SUCCESS]
receive encrypted/signed text from *output*
[ERROR]
read error from *result* and handle it

Fig. 2. Control flow of the proposed API.

Instead of providing an API via exported Android Activities, the API has been defined with the Android Interface Definition Language (AIDL). This allows for streaming of larger content via file descriptors. While this allows for performant access to shared memory between two apps, the immutable definition of the methods' type signatures makes it difficult to maintain backward compatibility with API definitions included in older client apps. Thus, instead of defining the method name and parameters directly as part of the type signature, these are defined using an Intent with a specific action (method name) and extras (method parameters), which are well-known to Android developers. Using Intents allows for easier backward compatibility and more flexible method definitions, which are not constrained to a specific parameter combination. To satisfy the requirement of backward compatibility, these are made versionable by including a version field together with a size calculated over the remaining fields at the first position when flattening the object for serialization.

Following Figure 2, after binding to the service, a client can execute a remote method. If a parameter is not specific enough or a required parameter is missing that can be provided by the user, the operation is canceled and the `USER_INTERACTION_REQUIRED` result code is returned together with a immutable PendingIntent. This PendingIntent can be started by the client at an appropriate time and is executed in the cryptography provider's process sandbox to handle interaction using appropriate UI components. One common use case is that a public key is missing for a given email address and must be downloaded. After user interactions, the operation is executed again with the parameters from the first execution combined with those retrieved from user input via the *result* Intent. The client app holds all parameters and decides by itself at which point in its control flow to



The displayed app wants to encrypt/decrypt messages and sign them in your name.
Allow access?

WARNING: If you do not know why this screen appeared, disallow access! You can revoke access later using the 'Apps' screen.

Please review the list of recipients!

No keys were found for these email addresses:
• bob@example.org

☐ Alice alice@example.org

None of the allowed keys is able to decrypt the content. Please select the allowed keys.

☐ Alice alice@example.org

Enter password for 'Alice'

Remember until screen off

Please choose a PIN with 6 numbers.

Please write down the Admin PIN and store it in a safe place (required when you used a wrong PIN 3 times).

Admin PIN
72241412

(a) Access control per app via user decision.

(b) Missing public key.

(c) Restriction of allowed keys per app.

(d) Password input for password-protected keys.

(e) PIN selection during key creation.

Fig. 3. A selection of UI components shown via PendingIntent for the `USER_INTERACTION_REQUIRED` state.

Table 1. API specification. No understanding of public-key cryptography is required.

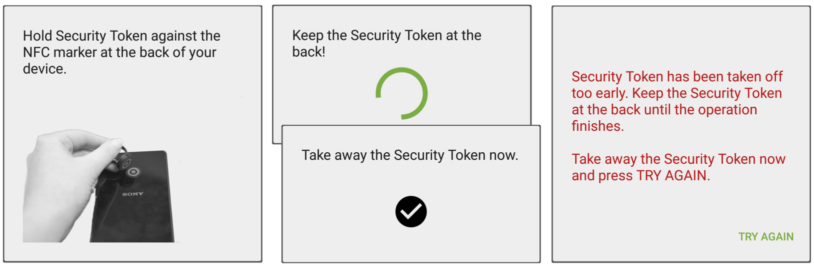
Action	Req. Extras	Description
`SIGN_AND_ENCRYPT`	`USER_IDS`	Encrypt to email addresses (`USER_IDS`) and generate signature
`DECRYPT_VERIFY`	-	Decrypt and verify signature

start the PendingIntent for user interaction. On `SUCCESS`, the encrypted and signed text has been streamed into the file descriptor given by *pipeId*.

For an API it is desirable to be stateless, i.e., the cryptography provider should be implementable without caching parameters or method calls for connected clients. Provider-side caching is unnecessary because the *result* Intent (previously passed through the PendingIntent), which is required for a second execution, is returned to the client after every user interaction (`USER_INTERACTION_REQUIRED` case). While this is possible for most parameters, PIN and passwords should never be exposed to the client and, thus, cannot be returned via the *result* Intent. Therefore, a PIN/password cache has been implemented using key IDs as unique identifiers. No session management is required inside the cryptography provider due to this architectural design. Due to their high abstraction, the exposed API methods work independently from the storage location of the secret key.

4.3 API

We provide a simple API specification in Table 1. A combined signature generation with encryption can be executed by creating an Intent with `SIGN_AND_ENCRYPT` with at least one extra holding the email addresses of the recipients named `USER_IDS`. The plaintext is streamed into the file descriptor and read from the file descriptor previously opened by the client. On first execution, the operation will result in the `USER_INTERACTION_REQUIRED` state three times before finishing with the ciphertext in `SUCCESS`. As shown in Figure 3a, the cryptography provider asks the end-user to allow the requesting client access to the API. Afterward, the user is asked to select her own key (secret key) by another UI component of the provider. If no key is available for the requested


Hold Security Token against the NFC marker at the back of your device.

Keep the Security Token at the back!

Take away the Security Token now.

Security Token has been taken off too early. Keep the Security Token at the back until the operation finishes.

Take away the Security Token now and press TRY AGAIN.

Fig. 4. Users are guided through the usage of security tokens over NFC.

email address a screen for key selection and retrieval is displayed, as shown in Figure 3b. Finally, based on the selection of the secret key, either a password (cf. Figure 3d) or a PIN is requested (similar to Figure 3d but with numeric keypad). This example shows that even without knowledge of public-key cryptography, a developer can effectively encrypt and sign data that can only be read by the recipient. Also, the secret key storage location is handled automatically and either a password input or NFC interaction with PIN input is returned. A second execution of the same Intent for different data will succeed earlier because access has now already been allowed, the public key is available and the PIN/password is already cached.

Regardless of whether the input is only encrypted, only signed, or a combined encryption with signature, an Intent with `DECRYPT_VERIFY` can be started to process the cryptographic input. No additional extras are required. Based on the PIN/password caching status of the specific secret key, the required screens are shown, e.g., Figure 3d. A special screen allows the user to restrict the secret keys that can be used by a particular client app (cf. Figure 3c). Besides the plaintext, two Parcelable objects are always returned indicating the decryption and signature result. These include the information if the given input was signed and/or encrypted.

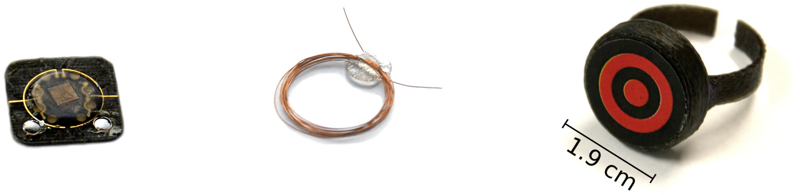
For security tokens, an appropriate PIN and Admin PIN must be chosen (cf. Figure 3e). Here, we prevent the user from chosing one of the top 20 common PIN combinations following Berry's PIN number analysis [9], e.g., 123456, 000000, and similar ones. An attacker can try up to 3 different PINs until the security token locks itself and can only be unlocked by the Admin PIN. We provided a trade-off between usability and security by letting the user select her favorite PIN, but securely generate an Admin PIN that should be written down. Our design decisions are similar to current practices of PIN/PUK selection for SIM cards.

Advanced API calls, for example to generate backups or detached signatures, can be found in OpenKeychain's API documentation [65].

4.4 NFC UI Component

We conducted a pre-study with 12 participants using a preliminary design of our NFC UI component. In this pre-study, we mainly focused on qualitative feedback, whereas the main goal of this pre-study was to find flaws in the UI design and user experience. We provided a Sony Xperia Z3 smartphone and a white NFC smart card that has been pre-configured for the scenario and asked the participants to send an encrypted email. We observed them during this task, especially their interaction with the NFC UI components. Finally, we interviewed them about their experience.

We found out that it is important to give clear instructions to guide the users through the steps of using a security token. In previous versions, users took away the security token too early or were confused when the dialog closed automatically after a successful operation. Thus, we improved the process by dividing it into three steps shown in Figure 4: 1) clearly depict how to hold the token against the device, 2) display a progress indicator together with the instruction to keep the token at the back, and finally 3) display the instruction that the token can



(a) IC extracted from NXP J3D081.

(b) Circular coil as new NFC antenna.

(c) 3D printed ring prototype.

Fig. 5. Components of our NFC signet ring prototype (photos have the same proportions).

be taken away. Our implementation keeps polling for the established NFC connection after successful operations to detect when the token is taken away. When the token indicates with an error that a wrong PIN has been used for authentication, the previous cached one for this key will be cleared for the next try.

4.5 NFC Smart Cards and Signet Rings

In addition to evaluating our architecture, in particular our proposed cryptographic API for developers, we want to study the impact of a different form factor for end-users. Widely available form factors are that of smart cards or USB sticks. Available NFC rings solely include read/write NFC tags (cf. Section 2) No challenge-response or asymmetric cryptographic protocols are supported. Thus, they do not satisfy the requirements in this paper to support full asymmetric cryptography.

Due to the unavailability of such rings, we created our own prototypes. Because smart card ICs are only sold to smart card manufacturers, we bought blank *NXP J3D081* developer cards with dual interface support for NFC. The IC, depicted in Figure 5a, has been extracted using acetone [74]). A new induction coil functioning as the NFC antenna has been created using magnet wire to fit the form factor (cf. Figure 5b). The number of turns for an NFC antenna operating at 13.56 MHz depends on the IC configuration. Thus, to estimate the correct number of turns, we measured the frequency of the original antenna with a signal generator and oscilloscope. The original antenna's frequency has been measured as 875 kHz, thus, the inductance can be calculated as $L = \frac{4.57}{875\,\mathrm{kHz}} \approx 5.223\,\mu\mathrm{H}$ [19]. According to NXP [63], the number of turns for circular coils can be calculated with

$$L = \frac{24.6 \cdot N^2 \cdot D}{1 + 2.75 \cdot \frac{s}{D}} .$$

The magnet wire has been wrapped around a metal cylinder with a diameter of $D = 1.53\,\mathrm{cm}$ resulting in a circular coil with $s = 0.2\,\mathrm{cm}$. Choosing the number of turns with $N = 14$ results in an inductance close to the original one:

$$5.426\,\mu\mathrm{H} \approx \frac{24.6 \cdot 14^2 \cdot 1.53\,\mathrm{cm}}{1 + 2.75 \cdot \frac{0.2\,\mathrm{cm}}{1.53\,\mathrm{cm}}}$$

The resulting coil has been soldered with the IC and inserted into a 3D printed ring prototype as depicted in Figure 5c. It should be noted that the cylinder height on top of the ring can be reduced drastically by a more sophisticated production process.

5 EVALUATION

In this section we evaluate different aspects of our architecture by API comparison, NFC performance measurements, and a user study of our UI components.

5.1 Methodology

The evaluation of our architecture consists of several parts. (1) To understand its designated use for developers, a comparison with existing APIs, implemented as libraries and applications, has been done. (2) The raw performance of the cryptographic operations over NFC have been measured in a controlled environment. This helps to understand the technical constraints we needed to design UI components for. (3) As mentioned in Section 4.4, we conducted walkthroughs with 12 users from our university. The results indicated that users did not know when to keep NFC tokens at the back or when to take them away, which led to an improved UI for NFC operations. (4) Finally, we recruited 40 participants from a large company to test the full architecture from an end-user perspective including the UI components of our API and different NFC security tokens in a real-world environment. Furthermore, the usability and user perception of our NFC signet ring prototype as an alternative form factor has been evaluated.

5.2 API Comparison

Many cryptographic APIs are available that have been designed with different features and goals in mind. Thus, not every API suites every purpose and when designing communication systems the selection of an appropriate one largely depends on the following: If the design should be interoperable with existing standards, the API must support *standardized formats*. If it operates in a closed ecosystem, modern *high-level APIs* can be chosen where less programming errors can be made [7]. For higher security standards, especially in cooperate environments, the API should *support security tokens*. Furthermore, a category of APIs exist supporting functionality that go beyond cryptographic methods and require storage and GUI, such as *PIN/password cache*, *key management*, and *Graphical User Interfaces (GUI)*. These features provide complex functionality via API calls and reduce the burden on client developers who otherwise need to implement these on their own.

We selected prominent representatives for the categories of traditional low-level APIs, modern high-level APIs, and fully integrated systems and compared them in regards to the discussed functionality in Table 2. Only APIs for supporting encryption and signature generation for end-to-end security are considered, no authentication or transport security APIs are included. While more APIs exist, they typically fall into one of these categories and are thus evaluated similar to the selected ones. While modern libraries such as libsodium or Keyczar provide 'crypto-box'-methods with a fixed set of algorithms, GnuPG's selected algorithms depend on local configuration files and preferred algorithms defined in public key files. In our implementation, similar high-level operations with fixed algorithms exist that do not even require the knowledge of public-key cryptography due to additional UI components. While libraries such as Bouncy Castle must be integrated with additional libraries, such as OpenSC, to support security tokens, in our architecture security token support is an integral part. Modern libraries often lack a standard format and a corresponding key/algorithm migration path.

The features that require either support by the operating system or depend on specific GUI toolkits are typically not found in libraries, but in apps/integrated systems. An exception is Keyczar that provides basic command line tools for key management. One of the main goals of our system is to provide common user interactions via UI components. GnuPG specifies a 'UI Server Protocol' [40] that has similar goals and is implemented for Kleopatra and GPA. The library GpgME makes accessing this API easier [40]. In comparison to our implementation, the specification is not stateless and the implementation in Kleopatra does not provide dialogs for security tokens. Its Inter-Process Communication (IPC) is based on libassuan for platform-independent sockets. GNOME's Seahorse,

Table 2. Feature comparison of cryptographic APIs for end-to-end security. Libraries that only offer authentication or transport security are not considered here.

			High-Level API w/ Secure Defaults	Supports Security Tokens	Standardized Formats	Cross-Platform	PIN/Password Cache	Key Management	GUI
Low-Level APIs	libcrypto	[66]	○	○	●	●	○	○	○
	Bouncy Castle	[11]	○	○	●	●	○	○	○
	OpenSC	[37]	○	●	●	◐	○	○	○
High-Level APIs	NaCl/libsodium	[8]	●	○	○	●	○	○	○
	Keyczar	[21]	●	○	○	●	○	◐	○
Fully Integrated Systems	GnuPG	[40]	○	●	●	◐	●	●	○
	GNU Privacy Assistant (GPA)[a]	[41]	○	●	●	◐	●	●	●
	Kleopatra[a]	[50]	○	●	●	◐	●	●	●
	GNOME Keyring[a]	[38]	○	●	●	○	●	●	●
	Our work		●	●	●	○	●	●	●

[a] uses GnuPG as its backend

which is the frontend to GNOME Keyring, provides similar capabilities using a dbus service. Still, it misses functionality such as searching for and importing keys when choosing recipients [39].

While our API is not used as widely as the listed competitors, several client applications have already been released with an active user base. Its usage spans easy use cases, such as password managers (*Password Store*), as well as sophisticated ones, such as instant messaging (*Conversations*) and email clients (*K-9 Mail*) [65]. Most client apps are developed by third-parties and available on Google Play.

5.3 NFC Performance

We measured the performance of executing cryptographic operations over NFC using a Sony Xperia Z3 and the NXP J3D081 smart card running Yubico's OpenPGP app version 1.0.10. The average durations can be found in Table 3. Besides generating secret keys, which is only done once for new users, our measurements show average durations below 1 s for day-to-day operations. Only the asymmetric operations are executed on the smart card: For signatures, the hash of the input is generated on the smartphone, only the RSA signature is calculated on the smart card. For decryption, AES is executed on the smartphone, only the session key is decrypted on the smart card. 2048 bit RSA keys have been transferred and generated. ECC has not been evaluated, because no OpenPGP applet with ECC support was available for JCOP operating system during this work.

Table 3. Mean durations (w/ standard deviation) of cryptographic operations over NFC (10 experiments per operation).

Operation	Duration	σ
Signature calculation	787.9 ms	3.18
Decrypt session key	830.9 ms	55.86

Operation	Duration	σ
Transfer existing secret key	711.9 ms	32.66
Generate secret key on-token[a]	9476.2 ms	2297.71

[a] Roughly, only every third key generation succeeded

Generating keys on the smart card turned out to be unreliable. Only roughly every third generation succeeded, while all other operations canceled by losing the connection. Even when having the card lying on a flat surface with the smartphone on top, we were never able to generate three keys in a row that makes this method unsuitable in practice. The same issues have been encountered with different smartphone-token combinations. By building a self-contained implementation, we ruled out issues in our architecture design. Instead, we suspect that the induction does not provide a perfectly stable energy supply, which is required by the key generation process. Because on-token key generation was too unreliable, in our current version keys are generated on-smartphone. We will investigate this issue further and will fix this in an upcoming version, possibly using ECC providing faster key generation methods.

5.4 User Study

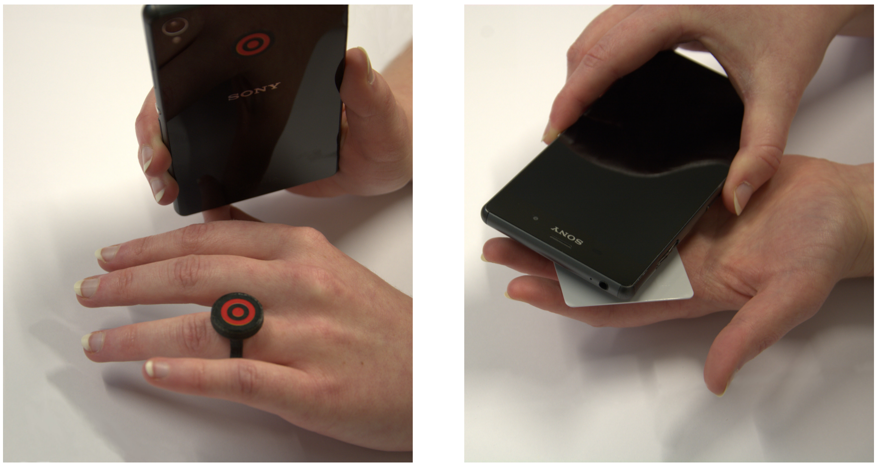
To evaluate the usability of NFC ring and card form factors in comparison to password-protected keys, we conducted an end-user study with 40 participants at a large company outside of the university environment. The main goal of our study is to test the usability of the NFC-based approaches in comparison with state-of-the-art password protection of secret key material. In this section, we present our study design and discuss the corresponding results.

5.4.1 Participant Recruitment. Our 40 participants were recruited in the IT department of a large company based in Germany. Taking part in the study was considered as working time, i.e., the participants were paid their normal hourly wages. Due to restrictions in their employment contract, we were not allowed to pay additional money on top. Our university does not have an Institutional Review Board (IRB), but the study conformed to the strict data protection law of Germany and informed consent was gathered from all participants.

5.4.2 Study Design. Our conducted study consists of two parts: (1) a lab experiment observing objective measurements such as *setup time*, *decryption time* and (2) a follow-up user survey for analyzing end-user perception.

The experiment consists of different tasks to be performed with different approaches.

5.4.3 Variables, Conditions and Participant Assignment. Our independent variable among all tasks was the chosen authentication type with following conditions *password*, *NFC card*, and *NFC ring*. The in the evaluation relevant dependent variables (objective measurements) were (1) duration to measure the efficiency of each task and (2) user perception based on a follow-up survey. The effectiveness was not considered as a separate dependent variable in our evaluation since all users were able to perform the tasks. For the condition assignment, we opted for a within-group design where all study participants had to perform tasks from all approaches: *password*, *NFC card*, and *NFC ring*. We did not test against other methods commonly used for authentication, such as biometric fingerprints or pattern-based techniques. Generally, these provide a much lower security level than passwords satisfying modern length requirements, let alone security tokens [56] and are thus not suitable for end-to-end encryption. Our design allows us to gather user perception at the end of the study where users give feedback and



(a) Ring with smartphone. (b) Card with smartphone.

Fig. 6. Handling of NFC rings and cards together with smartphones. A sticker has been used to indicate the best NFC spot on smartphone and ring.

ratings for all approaches. To mitigate learning and fatigue effects in our within-group study design, the order in which participants were asked to perform the approaches was randomized.

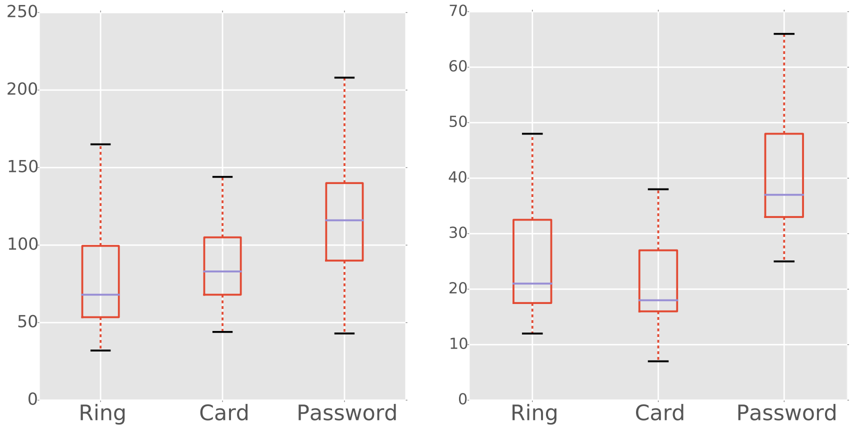
5.4.4 Tasks. The performed tasks are:

Task 1 When users start the application for the first time, they have to follow a wizard to create a key pair for usage with the app. They are guided through the process, which consists of entering a name and an email address, select a password/PIN and depending on the approach to hold an NFC security token against the smartphone. The actual key generation is indicated by a progress bar, while the user has to wait until it finishes.

Task 2 In the second task, the participants are asked to receive and read an encrypted email. Depending on the approach, users might be asked to enter a PIN or password, or hold an NFC device against their smartphone. To avoid bias due to variable password/PIN complexities, during this step we provide a pre-defined password/PIN.

Task 3 At last, the participants are asked to reply with a secure email by writing an appropriate response text and sending it.

To begin with the study, we provided a detailed explanation of the concept and the procedure to participants. We gave them a Sony Xperia Z3 smartphone and optionally depending on the approach, either the NFC ring or NFC card to let them get accustomed to the hardware themselves. As depicted in Figure 6, the usage patterns between the NFC ring and card differ due to their physical size. Before conducting the study, a sticker has been attached to the smartphone and ring indicating the best spot and the affiliation between these objects. Right after this, the participants continued with the key creation wizard of the first tested approach followed by the other remaining tasks. After completing the first tested approach, the other approaches follow. At the end of the study, we interviewed the participants for their ratings with regards to the single approaches and additional feedback. Finally, they were asked to participate in an anonymized questionnaire to collect demographic statistics.



(a) Setup time.

(b) Decryption time.

Fig. 7. Time measurements (in seconds, no outliers, lower is better).

5.4.5 Statistical Testing. For the statistical hypothesis testing, we opted for the common significance level of $\alpha = 0.05$. To account for multiple testing, all our study p-values are reported in the Holm-Bonferroni corrected version [44]. All time intervals and user-ratings are tested with the *nonparametric* (applicable to unknown statistical distributions) *Mann–Whitney U* test (two-tailed, Holm-Bonferroni corrected). We opted for this nonparametric statistical test due to the ordinal nature of our data and to avoid any statistical distribution assumptions. All our effect sizes are reported by mean comparisons and the usage of the *common language effect size* method [57], i.e., the meaning of the effect size is explained in plain English.

5.4.6 Objective Measurements. We measured following objective measurements in our experiment:

Setup time We measure the entire time of the setup process in Task 1. This includes input of name and email address, password/PIN selection, key generation on-smartphone, and optionally transfer to security token.

Decryption time Here, we measure only the time where the users have to perform an action related to the cryptographic operation of Task 2, i.e., password input and on-smartphone asymmetric decryption or PIN input and on-token operation (requiring holding the token against the smartphone's back side).

Sign/Encryption time Again, we measure the time where the users have to perform an action in Task 3. Due to PIN/password caching, no input is required for signing. Thus, we only measured the time required for executing the NFC operations by holding the security tokens against the smartphone's back side.

Figure 7a shows a box plot with a time distribution overview for the setup process. Our main hypothesis is that passwords are less efficient (especially on smartphones) in comparison to NFC-based approaches which is also a common belief in the usable security community. As can be clearly seen, the password-based approach tends to require extra time: a median of 114.5 seconds indicated by the blue line in box ($p < 0.0001$ in comparison to the NFC-based approaches supporting our main hypothesis). NFC-based approaches, on the other hand, have shown a better performance during the wizard process: a median of 83.5 and 68.5 seconds ($p = 0.083$ indicating that based on our sample size, we could not observe a significant difference between those). For instance, only 14 people were able to type in a valid password on the first try. By comparison, 22 people were able to position the ring correctly and choose a valid PIN on the first try.

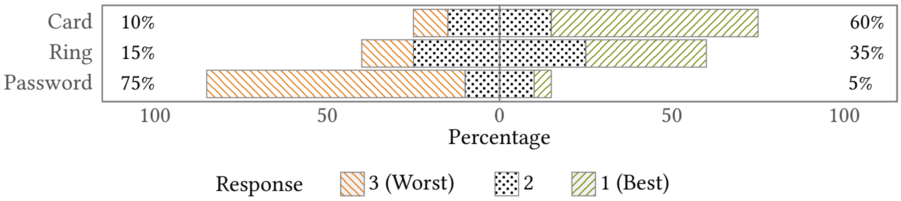


Fig. 8. Aggregated user perception showing the ranking choices in the interview.

Figure 7b shows a box plot with a time distribution overview for decrypting an email. This process consists of the following steps: when a secured email is opened, the cryptographic API immediately starts with the decryption process. To be able to decrypt, depending on the approach, either a password input or the positioning of the NFC device in combination with a PIN input is required. Note, that during this task we provided the participants a pre-defined password/PIN to avoid bias due to variable complexities. As can be clearly seen, the password-based protection not only performs worst in security and setup time, but also requires additional time for reading encrypted emails: median of 37.0 seconds (18.5 and 22.0 for NFC-based solutions) as indicated by the blue lines in the boxes ($p < 0.001$ supporting our main hypothesis). In the follow-up writing of a secured email where the password is cached, NFC-based approaches require additional time for positioning of the NFC devices: mean of 9.6 seconds (11 seconds for the ring and 8.3 for the card).

5.4.7 User Perception. To measure user perception, after completing all tasks in the experiment, we asked the users to answer questions with regards to their tested approaches in an interview. Here, we distinguish between *quantitative* results aggregated from users' ratings and *qualitative* open-end questions asking the users for their feedback and justifications. Our hypothesis is based on a common belief that passwords are troublesome on mobile devices. The full interview can be seen in the Appendix A. Furthermore, we asked them to fill out a short questionnaire form with additional details about their demographics, education, computer literacy, and previous usage of security apps. The full questionnaire can be seen in the Appendix B.

Quantitative. As our major quantitative question in the interview, we asked our participants to rank their tested approaches: 1 as best and 3 as worst. As depicted in Figure 8, the majority of our participants ranked NFC-based approaches to be superior to the password-based approach. The color coding shows the ranking level and the percentage numbers on the y-axes summarize the percentage of participants who ranked an approach as best (1) and as worst (3) respectively. A pairwise comparison between the approaches shows a high statistical significance between the token-based approaches and password-based protection ($p < 0.0001$ in both cases). However, we could not observe significant results between the NFC card and the ring ($p = 0.073$). The NFC card approach achieved a slightly higher mean of 1.5 in comparison the NFC ring that achieved a mean of 1.8. By comparison, password-based key storage achieved a mean rating of 2.7 where only 5 % of the users consider this approach to be best.

Qualitative. During the interview we asked the participants to describe the advantages and disadvantages of each approach (cf. Appendix A). While the participants accepted passwords as a well-known approach, most participants agreed that "imagining good passwords is incredibly difficult" and "good passwords are difficult to enter on smartphone keyboards". Whereas, cards and rings have the advantage of "requiring only a short PIN instead of a complicated password". In general, participants were in favor of cards, due to their common form factor, which allows to "store them easily in the wallet". Some mentioned that it is annoying to constantly take it out of the wallet, thus they prefer wearing the card attached to their belt. Many participants remarked that the card was more easily be placed below the smartphone and then worked perfectly with NFC and did not require precise positioning like the ring. Participants who favored the ring found the idea great and described it as a "cool gadget". Some noted that "rings are more secure than cards because they are more difficult to steal than wallets" and their "security purpose is not immediately obvious to an outsider". Interestingly, participants assessed it differently if cards or rings are more easily lost. Some argued that "rings can easily be forgotten on a bedside cabinet while not worn at night", while they argued that they "never forget the wallet in the morning before work". Others said that "cards are easily misplaced as they are not constantly worn on the body".

Demographics. A total of 40 users from the same company participated in our user study. 33 participants were male, 6 participants were female and one participant opted not to disclose the gender. The mean and median age of the participants was 34. In the quantitative analysis, we could not find any statistically significant differences between the genders, although women tend to prefer wearable NFC devices over cards. During the interview we also noted different reasons for liking/disliking particular approaches. For instance, 9 out of 33 men preferred cards instead of rings simply because they usually do not wear rings at all and are not accustomed to it. Some of them proposed the usage of watches or wristbands as an alternative form factor. A woman argued that, because dresses are often worn without belts, she "prefer[s] to wear cards attached to a necklace". Naturally, she and two other woman preferred the ring as it can be worn as a fashion accessory and has a smaller size.

14 participants did not have a university degree (3 of them were students planning to complete a degree). 8 participants completed a Bachelor's degree or similar, 17 a Master's degree or similar and 1 a doctorate's degree. We could not observe statistical significant differences between the degrees.

Our question set (cf. Appendix A) indicates a high level of technical background. On a scale from 1 (novice user) to 20 (experienced) the participants achieved a mean score of 17.1 wheres the participant with least knowledge achieved a score of 13.

Limitations. First-off, our study does not test whether end users will actually switch to NFC-based encryption or even start encrypting their emails during daily work. As with any lab study, more issues might arise in the field and thus an actual field study is an important future work. Our 40 participants were recruited in the IT department in a large company based in Germany, which is not the representative of the general population in Germany. As mentioned before, our questionnaire (cf. Appendix A) indicates a high level of technical background.

6 CONCLUSION

We proposed and implemented an architecture for NFC-based cryptography on Android devices. Our architecture includes a high-level cryptographic API especially designed for developers accustomed to Android's IPC mechanisms. It allows for cryptographic operations without knowledge of public-key cryptography, works transparently with password-protected key files as well as NFC security tokens, and provides carefully designed user interactions. In addition to traditional NFC smart cards, our NFC signet ring prototype represents an alternative form factor for end-users. Performance measurements show that cryptographic operations over NFC can be executed fast enough to be usable for day-to-day use. In our lab study with 40 participants, 95 % chose one of the NFC solutions as the best approach. Conclusively, we have shown the advantages of our architecture for NFC-based cryptography.

A INTERVIEW

The original questions were asked in German.

- Which operating system are you using on your own smartphone?
- For every approach {password, smart card, signet ring} (in the order the approaches have been tested by the participant):
  - What did you think was good?

  - What did you think was bad?
- Which approach was the best in your opinion? Please create an order by assigning the numbers 1, 2, 3.

Approach	Order
Password	[ ]
Smart Card	[ ]
Signet Ring	[ ]

- Would you consider replacing passwords by a signet ring?
- Would you consider replacing smart cards by a signet ring?

B QUESTIONNAIRE

The original questionnaire was written in German. Furthermore, it included in addition gender, age, and qualification questions (Question 1–3), which are not displayed in this appendix.

Question 4

Please rate how much you agree (or disagree) with the statements below.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I have a very good understanding of computers and the Internet.	☐	☐	☐	☐	☐
I often ask others for help when I have computer problems.	☐	☐	☐	☐	☐
Others often ask me for help when they have computer problems.	☐	☐	☐	☐	☐
I have a very good understanding of computer security.	☐	☐	☐	☐	☐

Question 5

Are you already using apps for encryption or secure communication on your *computer* in your *private* life?

☐ Yes, in particular: ....
☐ No

Question 6

Are you already using apps for encryption or secure communication on your *smartphone* in your *private* life?

☐ Yes, in particular: ....
☐ No

Question 7

Are you already using apps for encryption or secure communication on your *smartphone* in your *job*?

☐ Yes, in particular: ....
☐ No

Question 8

Have you already used NFC before this study?

☐ Yes, in particular for: ....
☐ No

ACKNOWLEDGMENTS

The authors would like to thank Vincent Breitmoser, Joey Castillo, and Signe Rüsch for contributing to the NFC-related code, Johannes van Balen and Yannic Schröder for their help creating the signet ring prototype, as well as Linda Fliss and Stephan Merker for their assistance and coordination during the lab study. Furthermore, we would like to thank all participants of the study.

REFERENCES

[1] 'Alex288'. 2014. NFC Smart Card Reader PC/SC Library: Project Description. (2014). Retrieved July 2017 from https://nfcsmartcardreader.codeplex.com

[2] Android Documentation. 2017. Cipher class. (2017). Retrieved July 2017 from http://developer.android.com/reference/javax/crypto/Cipher.html

[3] Android Documentation. 2017. Near Field Communication. (2017). Retrieved July 2017 from http://developer.android.com/guide/topics/connectivity/nfc/index.html

[4] Android Open Source Project. 2017. Nexus Security Bulletins. (2017). Retrieved July 2017 from https://source.android.com/security/bulletin

[5] Apple Inc. 2017. About Cryptographic Services. (2017). Retrieved July 2017 from https://developer.apple.com/library/ios/documentation/Security/Conceptual/cryptoservices/Introduction/Introduction.html

[6] Apple Inc. 2017. PassKit Package Format Reference. (2017). Retrieved July 2017 from https://developer.apple.com/library/ios/documentation/UserExperience/Reference/PassKit_Bundle/Chapters/TopLevel.html

[7] Daniel J. Bernstein, Tanja Lange, and Peter Schwabe. 2012. *Progress in Cryptology – LATINCRYPT 2012: 2nd International Conference on Cryptology and Information Security in Latin America, Santiago, Chile, October 7-10, 2012. Proceedings.* Springer Berlin Heidelberg, Berlin, Heidelberg, Chapter The Security Impact of a New Cryptographic Library, 159–176. https://doi.org/10.1007/978-3-642-33481-8_9

[8] Daniel J. Bernstein, Tanja Lange, and Peter Schwabe. 2012. *The Security Impact of a New Cryptographic Library.* Springer Berlin Heidelberg, Berlin, Heidelberg, 159–176. https://doi.org/10.1007/978-3-642-33481-8_9

[9] Nick Berry. 2012. PIN analysis. (Sept. 2012). Retrieved July 2017 from http://datagenetics.com/blog/september32012/index.html

[10] J. Bonneau, C. Herley, P. C. v. Oorschot, and F. Stajano. 2012. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. In *IEEE Symposium on Security and Privacy.* 553–567. https://doi.org/10.1109/SP.2012.44

[11] Bouncy Castle Inc. 2017. The Legion of the Bouncy Castle. (2017). Retrieved July 2017 from http://www.bouncycastle.org

[12] T. W. C. Brown, T. Diakos, and J. A. Briffa. 2013. Evaluating the eavesdropping range of varying magnetic field strengths in NFC standards. In *7th European Conference on Antennas and Propagation (EuCAP).* 3525–3528.

[13] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer. 2007. OpenPGP Message Format. RFC 4880 (Proposed Standard). (Nov. 2007).

[14] 'Cane', 'Topo', and 'Orso'. 2017. Privacy-Handbuch: GnuPG-SmartCard und NitroKey. (2017). Retrieved July 2017 from https://www.privacy-handbuch.de/handbuch_32r.htm

[15] Qi Alfred Chen, Zhiyun Qian, and Z. Morley Mao. 2014. Peeking into Your App without Actually Seeing It: UI State Inference and Novel Android Attacks. In *23rd USENIX Security Symposium (USENIX Security).* USENIX Association, San Diego, CA, 1037–1052.

[16] Yongsoon Choi, Jordan Tewell, Yukihiro Morisawa, Gilang A. Pradana, and Adrian David Cheok. 2014. Ring*U: A Wearable System for Intimate Communication Using Tactile Lighting Expressions. In *Proceedings of the 11th Conference on Advances in Computer Entertainment Technology (ACE '14).* ACM, Article 63, 4 pages. https://doi.org/10.1145/2663806.2663814

[17] Brett Cooley, Haining Wang, and Angelos Stavrou. 2014. *Activity Spoofing and Its Defense in Android Smartphones.* Springer International Publishing, Cham, 494–512. https://doi.org/10.1007/978-3-319-07536-5_29

[18] Stephen M. Curry. 1998. An introduction to the Java Ring. (April 1998). Retrieved July 2017 from http://www.javaworld.com/article/2076641/learn-java/an-introduction-to-the-java-ring.html

[19] Ronald Dekker. 2017. A Simple Method to Measure Unknown Inductors. (2017). Retrieved July 2017 from http://www.dos4ever.com/inductor/inductor.html

[20] Frank Denis. 2017. The Sodium crypto library (libsodium). (2017). Retrieved July 2017 from https://libsodium.org

[21] Arkajit Dey and Stephen Weis. 2008. Keyczar: A Cryptographic Toolkit. (Aug. 2008). Retrieved July 2017 from http://keyczar.googlecode.com/files/keyczar05b.pdf

[22] W. Diao, X. Liu, Z. Li, and K. Zhang. 2016. No Pardon for the Interruption: New Inference Attacks on Android Through Interrupt Timing Analysis. In *IEEE Symposium on Security and Privacy (SP)*. 414–432. https://doi.org/10.1109/SP.2016.32

[23] ECMA International. 2015. NFC-SEC-01: NFC-SEC Cryptography Standard using ECDH and AES, 4rd edition. ECMA-386. (June 2015). http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-386.pdf

[24] ECMA International. 2015. NFC-SEC: NFCIP-1 Security Services and Protocol, 4rd edition. ECMA-385. (June 2015). http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-385.pdf

[25] Manuel Egele, David Brumley, Yanick Fratantonio, and Christopher Kruegel. 2013. An Empirical Study of Cryptographic Misuse in Android Applications. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security (CCS '13)*. ACM, 73–84. https://doi.org/10.1145/2508859.2516693

[26] Rebecca Ehlers, Thorsten Ehlers, Werner Koch, and Matthias Kirschner. 2006. The GnuPG Smartcard HOWTO. (June 2006). Retrieved July 2017 from https://www.gnupg.org/howtos/card-howto/en/smartcard-howto.html

[27] Nikolay Elenkov. 2014. *Android Security Internals: An In-Depth Guide to Android's Security Architecture.* No Starch Press.

[28] M. Elkins, D. Del Torto, R. Levien, and T. Roessler. 2001. MIME Security with OpenPGP. RFC 3156 (Proposed Standard). (Aug. 2001).

[29] Sascha Fahl, Marian Harbach, Thomas Muders, Lars Baumgärtner, Bernd Freisleben, and Matthew Smith. 2012. Why Eve and Mallory love Android: An analysis of Android SSL (in) security. In *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 50–61.

[30] Sascha Fahl, Marian Harbach, Thomas Muders, Matthew Smith, and Uwe Sander. 2012. Helping Johnny 2.0 to Encrypt His Facebook Conversations. In *Proceedings of the Eighth Symposium on Usable Privacy and Security (SOUPS '12)*. ACM, Article 11, 17 pages. https://doi.org/10.1145/2335356.2335371

[31] Sascha Fahl, Marian Harbach, Henning Perl, Markus Koetter, and Matthew Smith. 2013. Rethinking SSL Development in an Appified World. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security (CCS '13)*. ACM, 49–60. https://doi.org/10.1145/2508859.2516655

[32] Fidesmo. 2017. Card App Store. (2017). Retrieved July 2017 from http://www.fidesmo.com

[33] 'Fluffy'. 2017. OpenPGP-Card. (2017). Retrieved July 2017 from https://github.com/FluffyKaon/OpenPGP-Card

[34] Lishoy Francis, Gerhard Hancke, Keith Mayes, and Konstantinos Markantonakis. 2010. *Practical NFC Peer-to-Peer Relay Attack Using Mobile Phones.* Springer Berlin Heidelberg, Berlin, Heidelberg, 35–49. https://doi.org/10.1007/978-3-642-16822-2_4

[35] Simson L. Garfinkel, David Margrave, Jeffrey I. Schiller, Erik Nordlander, and Robert C. Miller. 2005. How to Make Secure Email Easier to Use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM, 701–710. https://doi.org/10.1145/1054972.1055069

[36] Simson L. Garfinkel and Robert C. Miller. 2005. Johnny 2: A User Test of Key Continuity Management with S/MIME and Outlook Express. In *Proceedings of the 2005 Symposium on Usable Privacy and Security (SOUPS '05)*. ACM, 13–24. https://doi.org/10.1145/1073001.1073003

[37] GitHub. 2017. OpenSC/OpenSC. (2017). Retrieved July 2017 from https://github.com/OpenSC/OpenSC

[38] GNOME. 2014. Keyring. (2014). Retrieved July 2017 from https://wiki.gnome.org/action/show/Projects/GnomeKeyring

[39] GNOME. 2014. Seahorse Roadmap. (2014). Retrieved July 2017 from https://wiki.gnome.org/Apps/Seahorse/Roadmap

[40] GnuPG authors. 2017. Appendix A The GnuPG UI Server Protocol. (2017). Retrieved July 2017 from https://www.gnupg.org/documentation/manuals/gpgme/UI-Server-Protocol.html

[41] GnuPG authors. 2017. GPA - The Gnu Privacy Assistant. (2017). Retrieved July 2017 from https://www.gnupg.org/software/gpa/index.html

[42] Ernst Haselsteiner and Klemens Breitfuß. 2006. Security in Near Field Communication (NFC). In *Printed Handout of Workshop on RFID Security (RFIDSec)*. Philips Semiconductors.

[43] Mario Heiderich, Jann Horn, Abraham Aranguren, Jonas Magazinius, and Dario Weißer. 2015. Pentest-Report OpenKeychain. (Aug. 2015). https://cure53.de/pentest-report_openkeychain.pdf.

[44] Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics* (1979), 65–70.

[45] Michael Hölzl, Endalkachew Asnake, René Mayrhofer, and Michael Roland. 2014. Mobile Application to Java Card Applet Communication using a Password-authenticated Secure Channel. In *12th International Conference on Advances in Mobile Computing and Multimedia (MoMM)*. ACM Press, New York, NY, USA, 147–156. https://doi.org/10.1145/2684103.2684128

[46] Identiv. 2015. uTrust 2910 R Data Sheet. (Feb. 2015). http://www.identiv.com/pdf/technicaldata/technical-datasheets/uTrust_2910R_Reader_DS_2015_02.pdf

[47] ISO/IEC. 2008. *ISO/IEC 14443-4: Identification cards – Contactless integrated circuit cards – Proximity cards – Part 4: Transmission protocol.*

[48] ISO/IEC. 2013. *ISO/IEC 7816-4: Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange.*

[49] A. K. Jain, A. Ross, and S. Pankanti. 2006. Biometrics: a tool for information security. *IEEE Transactions on Information Forensics and Security* 1, 2 (June 2006), 125–143. https://doi.org/10.1109/TIFS.2006.873653

[50] KDE. 2017. Kleopatra - Certificate Manager and Unified Crypto GUI. (2017). Retrieved July 2017 from https://www.kde.org/applications/utilities/kleopatra/

[51] Henning Kortvedt and S Mjolsnes. 2009. Eavesdropping near field communication. In *The Norwegian Information Security Conference (NISK)*, Vol. 27.

[52] Juan Lang, Alexei Czeskis, Dirk Balfanz, Marius Schilder, and Sampath Srinivas. 2017. *Security Keys: Practical Cryptographic Second Factors for the Modern Web.* Springer Berlin Heidelberg, Berlin, Heidelberg, 422–440. https://doi.org/10.1007/978-3-662-54970-4_25

[53] Frederic Lardinois. 2015. Google And Samsung Will Now Release Monthly OTA Android Security Updates. (Aug. 2015). http://techcrunch.com/2015/08/05/google-and-samsung-will-now-release-monthly-ota-android-security-updates

[54] Shrirang Mare, Mary Baker, and Jeremy Gummeson. 2016. A Study of Authentication in Daily Life. In *Twelfth Symposium on Usable Privacy and Security (SOUPS)*. USENIX Association, Denver, CO, 189–206.

[55] Mario Heiderich and Krzysztof Kotowicz. 2013. Pentest-Report Mailvelope 12.2012 - 02.2013. (2013). Retrieved July 2017 from https://cure53.de/pentest-report_mailvelope.pdf

[56] Mindi McDowell, Jason Rafail, and Shawn Hernan. 2009. Cyber Security Tip ST04-002. US-CERT. (2009). Retrieved July 2017 from http://www.us-cert.gov/cas/tips/ST04-002.html

[57] Kenneth O McGraw and SP Wong. 1992. A common language effect size statistic. *Psychological bulletin* 111, 2 (1992), 361.

[58] John McLear. 2013. NFC Ring - One Smart Ring, Unlimited Possibilities. (July 2013). Retrieved July 2017 from https://www.kickstarter.com/projects/mclear/nfc-ring

[59] Maryam Mehrnezhad, Mohammed Aamir Ali, Feng Hao, and Aad van Moorsel. 2016. *NFC Payment Spy: A Privacy Attack on Contactless Payments.* Springer International Publishing, Cham, 92–111. https://doi.org/10.1007/978-3-319-49100-4_4

[60] MOTA. 2017. MOTA DOI SmartRing. (2017). Retrieved July 2017 from http://shop.mota.com/mota-doi-smartring.html

[61] Nitrokey. 2017. OpenPGP support. (2017). Retrieved July 2017 from https://github.com/Nitrokey

[62] Nitrokey. 2017. Secure your digital life. (2017). Retrieved July 2017 from https://www.nitrokey.com

[63] NXP Semiconductors. 2010. AN1445: Antenna design guide for MFRC52x, PN51x and PN53x. (Oct. 2010). http://data.nxp.com/doc/published_files/1270733179751

[64] OpenIntents. 2017. Where applications unite. (2017). Retrieved July 2017 from http://www.openintents.org

[65] OpenKeychain. 2017. Easy PGP. (2017). Retrieved July 2017 from https://www.openkeychain.org

[66] OpenSSL. 2016. Libcrypto API. (2016). Retrieved July 2017 from https://wiki.openssl.org/index.php/Libcrypto_API

[67] Celeste Lyn Paul, Emile Morse, Aiping Zhang, Yee-Yin Choong, and Mary Theofanos. 2011. A field study of user behavior and perceptions in smartcard authentication. In *Human-Computer Interaction–INTERACT 2011*. Springer, 1–17.

[68] A. Pietig. 2009. Functional Specification of the OpenPGP application on ISO Smart Card Operating Systems. (April 2009). http://www.g10code.com/docs/openpgp-card-3.0.pdf

[69] Precise Biometrics. 2017. Smart Card Readers for Convenient and Secure Access. (2017). Retrieved July 2017 from http://precisebiometrics.com/smart-card-reader

[70] Chuangang Ren, Yulong Zhang, Hui Xue, Tao Wei, and Peng Liu. 2015. Towards Discovering and Understanding Task Hijacking in Android. In *24th USENIX Security Symposium (USENIX Security 15)*. USENIX Association, Washington, D.C., 945–959.

[71] Arne Renkema-Padmos, Jerome Baum, Melanie Volkamer, and Karen Renaud. 2014. Shake Hands to Bedevil: Securing Email with Wearable Technology.. In *Proceedings of the Eighth International Symposium on. Human Aspects of Information Security & Assurance (HAISA 2014)*. 90–100.

[72] Research In Motion Limited. 2007. Smart Card Security Solved: The BlackBerry Smart Card Reader. (2007). Retrieved July 2017 from http://www.blackberry.com/newsletters/connection/it/i5-2007/smart-card-reader.shtml

[73] RINGLY. 2017. Smart Jewelry and Accessories. (2017). Retrieved July 2017 from https://ringly.com

[74] Michael Roland and Michael Hölzl. 2015. Evaluation of Contactless Smartcard Antennas. (July 2015). http://arxiv.org/abs/1507.06427

[75] Martina Angela Sasse. 2005. Usability and trust in information systems. In *Trust and Crime in Information Societies*, R Mansell and B Collins (Eds.). Edward Elgar, Cheltenham, UK, 319–348.

[76] Florian Schmaus, Dominik Schürmann, and Vincent Breitmoser. 2016. *XEP-0373: OpenPGP for XMPP.* Technical Report. XMPP Standards Foundation, http://xmpp.org/extensions/xep-0373.html.

[77] Florian Schmaus, Dominik Schürmann, and Vincent Breitmoser. 2016. *XEP-0374: OpenPGP for XMPP Instant Messaging.* Technical Report. XMPP Standards Foundation, http://xmpp.org/extensions/xep-0374.html.

[78] Dominik Schürmann and Lars Wolf. 2016. Surreptitious Sharing on Android. In *Sicherheit 2016 (Lecture Notes in Informatics)*, Vol. P-256. Gesellschaft für Informatik, Bonn, Germany, 137–148. http://www.ibr.cs.tu-bs.de/papers/schuermann-sicherheit2016.pdf

[79] Dennis D Strouble, GM Schechtman, and Alan S Alsop. 2009. Productivity and usability effects of using a two-factor security system. *Proceedings of SAIS* (2009), 196–201.

[80] Michael Tunstall. 2006. Attacks on Smart Cards. (2006). http://www.cs.bris.ac.uk/home/tunstall/presentation/AttacksonSmartCards.pdf

[81] Alma Whitten and J. Doug Tygar. 1999. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In *Proceedings of the 8th Conference on USENIX Security Symposium - Volume 8 (SSYM'99)*. USENIX Association.

[82] Meng Xu, Chengyu Song, Yang Ji, Ming-Wei Shih, Kangjie Lu, Cong Zheng, Ruian Duan, Yeongjin Jang, Byoungyoung Lee, Chenxiong Qian, Sangho Lee, and Taesoo Kim. 2016. Toward Engineering a Secure Android Ecosystem: A Survey of Existing Techniques. *ACM*

Comput. Surv. 49, 2, Article 38 (Aug. 2016), 47 pages. https://doi.org/10.1145/2963145

[83] Yubico. 2017. Trust the Net with YubiKey Strong Two-Factor Authentication. (2017). Retrieved July 2017 from https://www.yubico.com

[84] Yubico. 2017. YubiKey NEO's OpenPGP app. (maintained fork of "Java Card OpenPGP Card"). (2017). Retrieved July 2017 from https://github.com/Yubico/ykneo-openpgp

[85] Zimperium. 2015. Experts Found a Unicorn in the Heart of Android. (July 2015). Retrieved July 2017 from https://blog.zimperium.com/experts-found-a-unicorn-in-the-heart-of-android

Received May 2017; accepted July 2017.

An Empirical Study of Textual Key-Fingerprint Representations

Sergej Dechand
USECAP, University of Bonn

Dominik Schürmann
IBR, TU Braunschweig

Karoline Busse
USECAP, University of Bonn

Yasemin Acar
CISPA, Saarland University

Sascha Fahl
CISPA, Saarland University

Matthew Smith
USECAP, University of Bonn

Abstract

Many security protocols still rely on manual fingerprint comparisons for authentication. The most well-known and widely used key-fingerprint representation are hexadecimal strings as used in various security tools. With the introduction of end-to-end security in WhatsApp and other messengers, the discussion on how to best represent key-fingerprints for users is receiving a lot of interest.

We conduct a 1047 participant study evaluating six different textual key-fingerprint representations with regards to their performance and usability. We focus on textual fingerprints as the most robust and deployable representation.

Our findings show that the currently used hexadecimal representation is more prone to partial preimage attacks in comparison to others. Based on our findings, we make the recommendation that two alternative representations should be adopted. The highest attack detection rate and best usability perception is achieved with a sentence-based encoding. If language-based representations are not acceptable, a simple numeric approach still outperforms the hexadecimal representation.

1 Introduction

Public key cryptography is a common method for authentication in secure end-to-end communication and has been a part of the Internet throughout the last two decades [7, 11]. While security breaches have shown that systems based on centralized trusted third parties such as Certificate Authorities and Identity Based Private Key Generators are prone to targeted attacks [42], decentralized approaches such as Web of Trust and Namecoin struggle with beeing adopted in practice due to usability issues [7, 13, 30]. Certificate transparency systems, such as CONIKS and others [24, 39, 27], aim to solve a subset of these issues by providing an auditable directory of all user keys. Still, manual key verification, i. e., the link between public keys and the entities, such as hostnames or people, remains a challenging subject, especially in decentralized systems without pre-defined authorities, such as SSH, OpenPGP, and secure messaging [12, 41].

Many traditional authentication systems still rely on manual key-fingerprint comparisons [17]. Here, key-fingerprints are generated by encoding the (hashed) public key material into a human readable format, usually encoded in hexadecimal representation. A variety of alternatives such as QR Codes, visual fingerprints, Near Field Communication (NFC), and Short Authentication Strings (SAS) have been proposed. Most of these systems offer specific benefits, e. g., QR codes and NFC do not require users to compare strings, but they also come with specific disadvantages, e. g., they require hardware and software support on all devices. While advances are being made in these areas, the text-based representation is still the dominant form in most applications.

However, due to the recent boom of secure messaging tools, the debate of how to best represent and evaluate textual fingerprints has opened up again and there are many very active discussions among security experts [28, 33]. In April 2016, WhatsApp serving over one billion users enabled end-to-end encryption as default by implementing the Signal protocol. Key verification is optional and can be done by using QR codes or comparing numeric representations, in their case 60-digit numbers [43]. However, it is not clear whether their solution is more usable than traditional representations.

In this paper, we present an evaluation of different textual key-fingerprint representation schemes to aid in the secure messenger discussion. The requirements posed to the developers are as follows:

- The fingerprint representation scheme should provide offline support and work asynchronously. One reason for this is that fingerprints are often printed on business cards or exchanged by third parties.

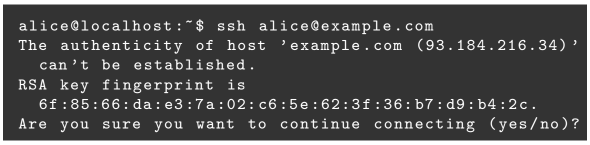
- The fingerprint should be transferable via audio channels, e. g., it should be possible to compare fingerprint over the phone.
- The representation scheme should be as technically inclusive as possible. No special hardware or software should be required to verify the fingerprints: both require a concerted and coordinated effort between many actors to get enough coverage for a comparison mechanism to be worthwhile for users to adopt.
- The representation should be as inclusive as possible, i. e., excluding as few people with sensory impairments (visual, color, audio, etc.) as possible.

The above requirements exclude many proposed representation schemes and offer an explanation why they have not seen any adoption outside of academia. For this reason, we focus exclusively on textual fingerprint representations in our study. Textual key-fingerprints do not require hardware support and work in synchronous and asynchronous scenarios, i. e., they can be compared via voice or printed on business cards. Depending on the scheme, they even could be recalled from memory and exchanged over a voice channel.

This paper presents our study testing the usability of various textual key-fingerprint representation schemes. Our study consists of two parts: (1) an experiment where we measured how fast and accurate participants perform for different schemes, and (2) a survey about their perception and sentiment. These also contained a direct comparison between the representations.

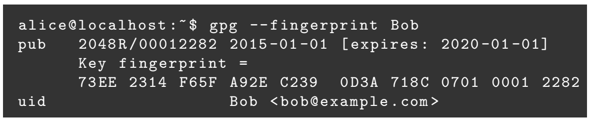
Our findings suggest that the most adopted alphanumeric approaches such as the Hexadecimal and Base32 scheme perform worse than other alternatives: under a realistic threat model, more than 10% of the users failed to detect attacks targeting Hexadecimal representations, whereas our best system had failure rates of less than 3%. While the best system for accuracy is not the fastest, it is the system which received the highest usability rating and is preferred by users.

In the following sections, we discuss related work followed by an analysis of current implementations deploying in-persona key-fingerprint representation techniques and discuss our evaluated representation schemes. Then, we describe our experiment evaluating text-based key-fingerprint verification techniques with regards to their attack-detection accuracy and speed. Our experiment was conducted as an online study with 1047 participants recruited via the Amazon Mechanical Turk (MTurk) platform. We consider the scenario outlined above, where a user compares two key-fingerprint strings encoded by the different representation schemes. In addition to the implicit measurements of accuracy and speed, we also evaluate the self-reported user perception to get feedback about which systems are preferred by end users. Finally, we present our results, discuss their implications and takeaways, and conclude our work.



```
alice@localhost:~$ ssh alice@example.com
The authenticity of host 'example.com (93.184.216.34)'
  can't be established.
RSA key fingerprint is
  6f:85:66:da:e3:7a:02:c6:5e:62:3f:36:b7:d9:b4:2c.
Are you sure you want to continue connecting (yes/no)?
```

(a) OpenSSH: Lowercase Hexadecimal with Colons



```
alice@localhost:~$ gpg --fingerprint Bob
pub   2048R/00012282 2015-01-01 [expires: 2020-01-01]
      Key fingerprint =
      73EE 2314 F65F A92E C239  0D3A 718C 0701 0001 2282
uid                  Bob <bob@example.com>
```

(b) GnuPG: Uppercase Hexadecimal with Spaces

Figure 1: Alphanumeric Fingerprints Used in Practice

2 Related Work

Various key-fingerprint representations have been proposed in academia and industry. Various cryptographic protocol implementations still rely on manual fingerprint comparisons, while the hexadecimal representation is used in most of them. However, previous work suggests that fingerprint verifications are seldom done in practice [17, 37].

2.1 Key-Fingerprint Representations

Previous work has shown that users struggle with comparing long and seemingly “meaningless” fingerprints and it is suspected that they even might perform poorly in this task [19]. While most previous work has focused on the family of visual fingerprints [35, 32, 19, 10], to our knowledge, none of those focused on the differences between various different textual fingerprint representations.

Hsiao et al. have conducted a study with some textual and visual representation methods for hash verification [19]. They compared Base32 and simple word list representations with various algorithms for visual fingerprints and hash representation with Asian character sets (a subset of Chinese, Japanese Hiragana, and Korean Hangul, respectively). A within-subjects online study with 436 participants revealed that visual fingerprints score very well in both accuracy and speed, together with the Base32 text representation. Hsiao et al. conclude that depending on the available computation power and display size, either Base32 or one of the visual fingerprinting schemes should be used. They explicitly did not evaluate hexadecimal representation or digits "because that scheme is similar to Base32 and known to be error-prone" [19]. However, our work shows that numeric representations actually perform significantly better than Base32 and is less error prone. In addition, our results suggest that language-based schemes, e. g., generated sentences achieve excellent results comparable to visual schemes. At the same time, textual approaches are more flexible (can be read out loud) and do not exclude people with sensory impairments.

Another study by Olembo et al. also focused mainly on the topic of visual fingerprints [32]. They developed a new family of visual fingerprints and compared them against a Base32 representation. The Base32 strings were twelve characters long and displayed without chunking. The participants performed better with the visual fingerprints than with Base32, regarding both accuracy and speed. Olembo et al. conclude that the Base32 representation is far away from optimal when it comes to manual key-fingerprint verification. We test this claim by comparing Base32 representation with other textual key-fingerprint representation and eventually prove it wrong.

Regarding chunking, Miller et al. have published *The magical number seven* and succeeding work that shows that most people can recall 7 ± 2 items from their memory span [29]. It has been shown that although there are slight differences between numbers, letters and words (numbers perform slightly better than letters, and letters slightly better than words), they perform similar in studies. More recent studies have shown that human working memory easily remembers up to 6 digits, 5.6 letters and 5.2 words [1, 6, 8]. Adjusting chunk sizes to these numbers can help users when comparing hashes.

While all of the above studies offer interesting insights into different (mainly visual) fingerprint representations, to the best of our knowledge there is not work focusing on which textual representation performs the best. However, this knowledge would be extremely important to help in the current debate in the secure messaging community. The representations currently being put forward and implemented are far from optimal and the results of our study can help improve the accuracy and usability of fingerprint representations. Unlike the above studies we conduct our study with a more realistic attacker strenth, as presented in subsection 4.1).

2.2 Passwords and Passphrases

A passphrase is basically a password consisting of a series of words rather than characters. In academic literature, passphrases are often considered as a potentially more memorable and more secure alternative to passwords and are often recommended by system administrators [23, 40]. In contrast to most passphrase-based systems, key-fingerprints cannot be chosen by the end-user and thus are more related to the system-assigned passphrases field: Bonneau et al. have shown that users are able to memorize 56-bit passwords [4]. miniLock[1] and its commercial successor Peerio[2] use system-assigned passphrases to generate cryptographic key pairs easing key backup and synchronization among multiple devices.

Scheme	Example

Hexadecimal	`18e2 55fd b51b c808`
	`601b ee5c 2d69`

Base32	`ddrf l7nv dpea`
	`qya3 5zoc 22i`

Numeric	`2016 507 6420 1070 394`
	`1136 2973 991 70`

PGP	locale voyager waffle disable Belfast performance slingshot Ohio spearhead coherence hamlet liberty reform hamburger
Peerio	bates talking duke rummy slurps iced farce pound day
Sentences	Your line works for this kind power cruelly. That lazy snow agrees upon our tall offer.

Table 1: Examples for different textual key-fingerprint representations for the same hash value

Contrary to widespread expectations, Shay et al. were not able to find any significant recall differences between system-assigned passphrases and system-assigned passwords [40]. However, they reported reduced usability due to longer submission times due to typing.

Similar to passphrases, the usage of language-based key-fingerprint representations is claimed to provide better memorability than just an arbitrary series of character strings despite the lack of empirical evidence. In our study, we measure the performance of the different approaches and also collect perception and feedback from end users.

3 Background

In the past years, various textual key-fingerprint representations have been proposed. In this section, we analyze currently practised in-persona key verification techniques in well-known applications. For comparison, Table 1 lists the approaches we used in our evaluation generated from the same hash value.

Only applications requiring manual key-fingerprint

---

[1] `https://minilock.io`

[2] `https://peerio.com`

verification are considered. In mechanisms like S/MIME or X.509, fingerprints play only a secondary role because certificates are verified via certificate chains.

In the following, $SHA\text{-}1(x)^{16}$ defines the execution of 16 rounds of nested *SHA-1* on x, a truncation to the leftmost 16 bits is defined by $x[0,\dots,16]$, and pk is used as an abbreviation for the values of a public key (differs for RSA, DSA, or ECC).

3.1 Numeric

Numeric representation describes the notation of data using only numeric digits (0-9). The primary advantage of a such system is that Arabic numerals are universally understood, and in addition, numeric key-fingerprints show a similarity to phone numbers. The encoding is achieved by splitting a binary hash into chunks of equal length and expressing each chunk as a decimal number, e. g., by simply switching the representation base from 2 to 10.

The messaging and data exchange application SafeSlinger[3] implements this as a fallback scheme for unsupported languages [14]. A 24 bit SAS in SafeSlinger (cf. Figure 2a) can be expressed by three decimal encoded 8-bit numbers.

In the messaging platform WhatsApp, a fingerprint is calculated by $SHA\text{-}256(pk)^{5200}[0,\dots,240]$. This fingerprint is split up into six chunks, where each chunk is represented by a five digits long number modulo 100,000 [43]. Concatenating this fingerprint with the fingerprint of the communication partner results in the displayed representation, e. g.,

```
77658 87428 72099 51303
34908 23247 95615 27317
09725 59699 62543 54320
```

3.2 Alphanumeric

Alphanumeric approaches use numbers and letters to represent data. Depending on the representation type and its parameters, the letters can be presented either in lower-case or in upper-case. The string can be chunked into groups of characters, which are usually of equal length. Chunking does not alter the information contained, while changing lower-case letters to upper-case letters (and vice versa) may does, depending on the coding scheme. Commonly used representations are Hexadecimal, Base32, and Base64.

3.2.1 Hexadecimal

Hexadecimal digits use the letters A-F in addition to numerical digits and are a common representation for key-fingerprints and primarily used in SSH and OpenPGP. Note that the case of the letters do not make any difference. Regarding chunking, both spaces (cf. Figure 1b) and colons (cf. Figure 1a) are commonly used as separation characters.

Key fingerprints in OpenPGP version 4 are defined in RFC 4880 [7] by

$$Hex(SHA\text{-}1(0x99 \,\|\, len \,\|\, 4 \,\|\, creation\_time \,\|\, algo \,\|\, pk))$$

where *len* is the length of the packet, *creation_time* is the time the key has been created and *algo* is unique identifier for the public-key algorithm. While the inclusion of *creation_time* makes sure that even two keys with the same key material have different fingerprints, it allows an attacker to iterate through possible past times to generate similar fingerprints skipping the key generation step [5]. The actual representation of OpenPGP fingerprints is not defined in RFC 4880, but most implementations chose to encode them in hexadecimal form, e. g., GnuPG displays them uppercase in 16 bit blocks separated by whitespaces with an additional whitespace after 5 blocks (cf. Figure 1b), e. g.,

```
73EE 2314 F65F A92E C239  0D3A 718C 0701 0001 2282
```

Other implementations, such as OpenKeychain, deviate only slightly, for example by displaying them lowercase or with colored letters to ease comparison but still provide compatibility with GnuPG.

SSH fingerprint strings, as defined in RFC 4716 and RFC 4253 [15, 44], are calculated by

$$Hex(MD5(Base64(algo \,\|\, pk)))$$

where *algo* is a string indicating the algorithm, for example “ssh-rsa”. Fingerprints are displayed as “hexadecimal with lowercase letters and separated by colons” [15] (cf. Figure 1a), e. g.,

```
6f:85:66:da:e3:7a:02:c6:5e:62:3f:36:b7:d9:b4:2c
```

3.2.2 Base32

Base32 uses the Latin alphabet (A-Z) without the letters O and I (due to the confusion with numbers 1 and 0). There is no difference between lower-case letters and upper-case letters. In addition, a special padding character “=” is used, since the conversion algorithm processes blocks of 40 bit (5 Byte) in size. The source string is padded with zeroes to achieve a compatible length and sections containing only zeroes are represented by “=” [20, 21].

The ZRTP key exchange scheme for real-time applications is based on a Diffie-Hellman key exchange extended by a preceding hash commitment that allows for very short fingerprints, called Short Authentication

---

[3] `https://www.cylab.cmu.edu/safeslinger`

Strings (SAS) without compromising security [45]. The Base32 encoding used in ZRTP uses a special alphabet to produce strings that are easier to read out loud. VoIP applications such as CSipSimple[4] use this Base32 option, usually named “B32” inside the protocol. Here, the leftmost 20 bits of the 32 bit SAS value are encoded as Base32. , e. g.,

`5 e m g`

3.2.3 Base64

There exist a number of specifications for encoding data into the Base64 format, which uses the Latin alphabet in both lower-case and upper-case (a-z, A-Z) as well as the digits 0-9 and the characters “+”, “/”, and “=” to represent text data. Again, the character “=” is used to encode padded input [20]. Starting with OpenSSH 6.8 a new fingerprint format has been introduced that uses SHA-256 instead of MD5 and Base64 instead of hexadecimal representation. In addition the utilized hash algorithm is prepended, e. g.,

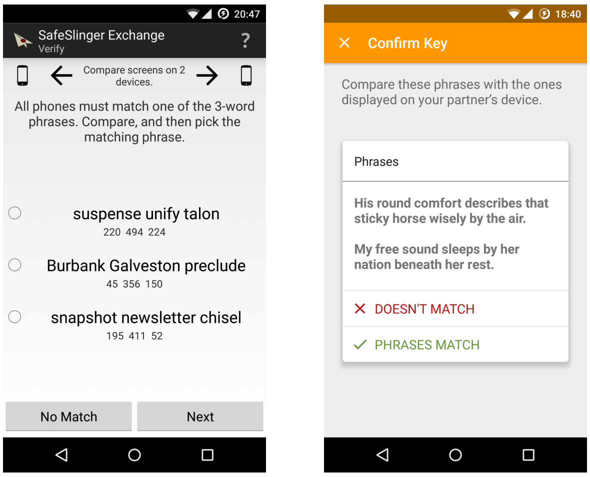
`SHA256:mVPwvezndPv/ARoIadVY98vAC0g+P/5633yTC4d/wXE`

3.3 Unrelated Words

Instead of (alpha)numeric representation, fingerprints can be mapped to lists of words. Here, the binary representation is split into chunks, where each possible value of a chunk is assigned to a word in a dictionary. To increase readability, such a dictionary usually contains no pronouns, articles, prepositions and such. Word lists, such as the PGP Word List [22] and the Basic English word list compiled by K.C. Ogden [31], are primarily used for verification mechanisms based on SAS. Key-Fingerprints represented by words have been implemented for VoIP applications based on the ZRTP key exchange and other real-time communication protocols. Examples are Signal[5], and the messaging and contact sharing application SafeSlinger [14] (cf. Figure 2). Besides their use in SAS based mechanisms, miniLock and Peerio utilize unrelated words for passphrase generation.

An example for a modern VoIP implementation that utilizes ZRTP for key exchange over Secure Real-Time Transport Protocol (SRTP) is Signal’s private calling feature, previously distributed as Redphone. The developers chose to implement only a specific subset of the ZRTP specification [45], namely Diffie-Hellmann key exchange via P-256 elliptic curves using “B256” SASs, i. e., Base256 encoding that maps to the leftmost 16 bits of the 32 bit SAS values to the previously introduced PGP Word List [22], e. g.,

`quota holiness`



All phones must match one of the 3-word phrases. Compare, and then pick the matching phrase.

○ suspense unify talon
220 494 224

○ Burbank Galveston preclude
45 356 150

○ snapshot newsletter chisel
195 411 52

Compare these phrases with the ones displayed on your partner’s device.

Phrases

His round comfort describes that sticky horse wisely by the air.

My free sound sleeps by her nation beneath her rest.

(a) SafeSlinger: List of words (b) OpenKeychain: Sentences

Figure 2: Language-based fingerprint representations

The messaging application SafeSlinger is based on a Group Diffie-Hellman protocol [14] implementing a key verification with SASs for up to 10 participants. In SafeSlinger the leftmost 24 bits of a SHA-1 hash is used to select 3 words from the PGP Word List, e. g.,

`suspense unify talon.`

Besides this, two other 3 word triples are selected to force users to make a selection before proceeding (cf. Figure 2a).

In contrast to Signal and SafeSlinger, Peerio (based on miniLock) does not use any SAS based verification mechanism. It uses pictures for verification and word lists for code generation. The word list is generated from most occurring words in movie subtitles. Besides key verification, these are also used to generate so called passphrases, which are used to derive their ECC private keys.

3.4 Generated Sentences

The words from the previous dictionaries can also be used to generate syntactically correct sentences as proposed by previous research: Goodrich et al. proposed to use a “syntactically-correct English-like sentence” representation for exchanging hash-derived fingerprints over audio by using *text-to-speech* (TTS) [16]. Michael Rogers et al. implemented a simple deterministic sentence generator [16, 38][6] Though the sentences from both approaches rarely make sense in a semantic fashion, they are syntactically correct and are claimed to pro-

---

[4] https://github.com/r3gis3r/CSipSimple

[5] https://github.com/WhisperSystems/Signal-Android

[6] https://github.com/akwizgran/basic-english

vide good memorability. In our study, we used Michael Roger's approach for our sentence generator.

We implemented this method for PGP fingerprints in OpenKeychain 3.6[7] (cf. Figure 2b). To the best of the authors' knowledge, to this date, it is the first integration of key verification via sentences although other projects are considering to change their fingerprint encoding scheme [38, 36].

4 Methodology

In order to evaluate the effect and perception of the different textual key-fingerprint representations, we conducted an online study on Amazon's Mechanical Turk (MTurk) crowdsourcing service. Our Universities do not have an IRB, but the study conformed to the strict data protection law of Germany and informed consent was gathered from all participants. Our online study is divided into two parts: The experiment for performance evaluation followed by a survey extracting self-reported data from users. The survey ended with demographic questions.

4.1 Security Assumptions

In this section, we define the underlying security assumptions of our study, such as fingerprint method, length, and strength against an adversary. The fingerprint method and parameters are utilized consistently for all experiments in our study to offer comparability between all possible fingerprint representations. This attack model is important for the usability since an unrealistically strong or weak attacker could skew the results. Obviously, if the fingerprint strength is not kept equal between the systems this would also skew the results.

4.1.1 Fingerprint Method

To decide upon a fingerprint method for humanly verifiable fingerprints in our study, we first have to differentiate between human and machine verification to illustrate their differences. While a full fingerprint comparison can be implemented for machine verification, humans can fall for fingerprints that match only partially. Additionally, machine comparison can work with long values, whereas for human verification the length must be kept short enough to fit on business cards and to keep the time needed for comparison low.

For machine comparison, full SHA-256 hashes should be calculated binding a unique *ID* to the public key material. The probability of finding a preimage or collision attack is obviously negligible, but the fingerprints can still be computed fast in an ad-hoc manner when needed.

It is important to note that collision resistance is not required for our scenarios. It is required for infrastructure-based trust models such as X.509, where certificates are verified by machines and trust is established by authority. In these schemes, a signature generated by a trusted authority can be requested for a certificate by proving the control over a domain, but then reused maliciously for a different certificate/domain. This is already possible with a collision attack, without targeting a full preimage. In contrast, the direct human-based trust schemes considered in this study only need to be protected against preimage attacks, because no inherently trusted authority is involved here.

While machine comparison needs to be done fast, e. g., on key import, manual fingerprint verification by humans is done asynchronously in person or via voice. Thus, we can use a key derivation function to provide a proof-of-work, effectively trading calculation time for a shorter fingerprint length. Secure messaging applications such as Signal or OpenPGP-based ones could pre-calculate the fingerprints after import and cache these before displaying them for verification later.

Thus, modern memory-hard key derivation functions such as *scrypt* [34] or *Argon2* [3] can be utilized to shorten the fingerprint length. These key derivation functions are parametrized to allow for different work factors. Suitable parameters need to be chosen by implementations based on their targeted devices and protocol.

As discussed in Section 3.2.1, while the generation of new fingerprints consists of the creation of a new key pair and the key derivation step, an attacker can potentially skip the key creation. Thus, in the following we only consider the key derivation performance as the limiting factor for brute force attacks.

When utilizing a properly parametrized key derivation function for bit stretching, the security of a 112 bit long fingerprint can be increased to require a brute force attack comparable to a classical 2^{128} brute force attacker. Consequently, a fingerprint length of 112 bit is assumed throughout our study.

4.1.2 Attacker Strength for Partial Preimages

In our user study, we assume an average attacker trying to impersonate an existing *ID* using our fingerprint method. Thus, an attacker would need to find a 112 bit preimage for this existing fingerprint using a brute force search executing the deployed key derivation function in each step. Due to the work factor, we consider this to be infeasible and instead concentrate on partial preimages. For comparability and to narrow the scope of our study, an attacker is assumed that can control up to 80 bits of the full 112 bit fingerprint.

Attackers might aim to find partial preimages where

---

[7] `https://www.openkeychain.org`

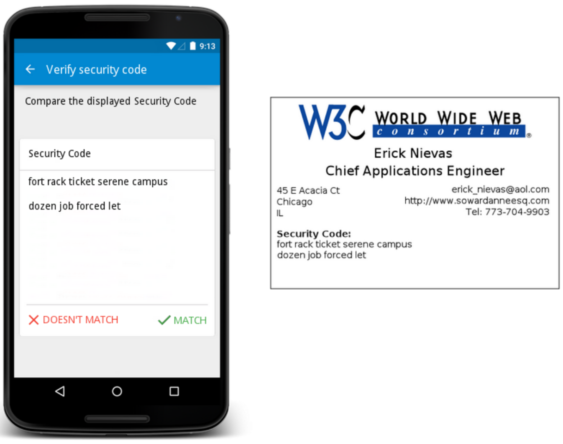
the uncontrolled bits occur at positions that are more easily missed by inattentive users. First, the bits at the beginning and the end should be fixed as users often begin their comparison with these bits. Thus, we assume that, for any representation method, the first 24 and last 24 bits are controlled by the attacker and thus the same as in the existing fingerprint. Based on the feedback from our pre-study participants and reports from related work, this can be considered best-practice [17, 37]. Second, of the remaining 64 bits in the middle of our 112 bit fingerprint, we assume that 32 bits are controlled by the attacker in addition to the first 24 and last 24 bits. In total, we assume that 80 bits are controlled by the attacker, i. e., are the same as in the existing fingerprint, and 32 bit are uncontrolled.

The probability of finding such a partial preimage for a fingerprint when executing 2^{49} brute force steps is calculated approximately by

$$1-\left(\frac{2^{112}-\sum_{k=1}^{32}\binom{64}{k}}{2^{112}}\right)^{2^{49}} \approx 0.66.$$

The inner parentheses of this equation define the probability that no partial preimage exists for one specific bit permutation. Instead of using $\binom{64}{32}$, a sum over 32 variations has been inserted to include permutations with more than the uncontrolled 32 bit that are also valid partial preimages. Finally, the probability to find a partial preimage is defined by the inverse of the exponentiation. Assuming the scrypt key derivation function parametrized with $(N, r, p) = (2^{20}, 8, 1)$, Percival calculates the computational costs of a brute force attack against 2^{38} ($\approx 26^8$) hashed passwords with \$610k and 2^{53} ($\approx 95^8$) with \$16B [34]. These costs can be considered a lower and upper bound for our attacker, which we assume to have average capabilities and resources. While 2^{38} has a probability of finding a partial preimage of only 0.05%, with 2^{42} the probability reaches nearly 1%, and with 2^{49}, as in our example, a partial preimage is found with over 50%.

In our study, we simulate attacks by inverting the bits from the existing fingerprint which are uncontrolled by the attacker, while the controlled bits are unchanged. For our theoretical approximation, we assume that the first 24 and last 24 bits should be controlled as well as 32 bits from the middle. In our study, we simulate an even more careful selection of appropriate fingerprints from the ones that an attacker would brute force. A general criteria here is to minimize the influence of uncontrolled bits on the entire fingerprint: For numeric and alphanumeric representations all bits affecting a character or digit are inverted together. For unrelated words, all bits affecting a word are changed. Sentences are never changed in a way that would alter the sentence structure.



Compare the displayed Security Code

Security Code
fort rack ticket serene campus
dozen job forced let

Erick Nievas
Chief Applications Engineer
45 E Acacia Ct
Chicago
IL
erick_nievas@aol.com
http://www.sowardanneesq.com
Tel: 773-704-9903

Security Code:
fort rack ticket serene campus
dozen job forced let

Figure 3: A screenshot of the actual task a user had to perform in the experiment. A user rates whether the *security codes* match, in this case with the Peerio word list approach, by clicking on the corresponding buttons shown on the phone.

4.2 Pre-Study

To get additional feedback from participants and evaluate our study design for flaws and misunderstandings, we conducted two small pre-studies: A lab study with 15 participants and an MTurk experiment with 200 participants, all required to perform 10 comparisons for each representation scheme (totally 60 comparisons in a randomized order). In our lab-study, we mainly focused on qualitative feedback, whereas the main goal of the MTurk pre-study was to find flaws in the presentation and task descriptions, as well as to check whether our proposed methodology is received as expected.

The biggest problem we found regarding the study design was that participants were uncertain if they should check for spelling mistakes in the words and sentence-based representation or if the all attacks would change entire words. To clarify this, a speech bubble was included in the task description that the participants do not have to look for spelling mistakes for language-based approaches.

We tested different rates of attack during the pre-study. The results showed that participants who were exposed to frequently occurring attacks were more aware and had a much higher attack detection rate. For our main study, we reduced the number of attacks to 40 comparisons with 4 attacks to have a good balance between true positives and false negatives. We received feedback that attacks on anchor parts of the strings, i. e., in the beginning, end, and at line breaks could be easily detected. Many users had problems with distinguishing the hexadecimal from the Base32 representation as well as distinguishing different word list approaches (Peerio vs. OpenPGP word list). Thus, we opted for a mixed factorial study design where users test *only one* scheme of each type. We grouped the hexadecimal and Base32 scheme for the alphanumeric type and the PGP and Peerio for the wordlist type together. These two groups were tested between-subjects in a split-plot design, i. e., the participants test either hexadecimal or Base32 for the alphanumeric type. See Table 2 for a graphical representation of our condition assignment design.

4.3 Experiment Design

The main part of our online study is the experiment part where users perform actual fingerprint comparisons. Here, we conducted two separate experiments with a distinct set of participants: (1) our main experiment testing different textual high-level representation schemes against each other and (2) a secondary experiment testing different chunk sizes for the hexadecimal representation. We opted for two distinct experiments due to the exponential growth of experiment conditions, as described in Section 4.3.1.

Before letting the participants start our experiment, we explained the scenario:

> "With this HIT, we are conducting an academic usability study followed by a short survey about different types of security codes used in the IT world. Security codes are often used in encrypted communications to identify the participants in a communication. If the security codes match, you are communicating securely. If they don't match, an eavesdropper may be intercepting your communication".

On MTurk, the term *Human Intelligence Task*, or *HIT* stands for a self-contained task that a worker can work on, submit answers, and get a reward for completing. Since our participants might not be familiar with the *key-fingerprint representation* term, we replaced it with *security codes* for the sake of the study.

We opted not to obfuscate the goal of the study since our research aims at finding the best possible representation for the comparison of key-fingerprints in a security context. This is closest to how users interact with fingerprints in the real world — their secure messaging applications also ask them to compare the strings for security purposes. The question how to motivate users to compare fingerprints is an entirely different research question. So in our case, we believe it was not necessary or desirable to use deception and since deception should be used as sparingly as possible we opted for the "honest" approach.

After agreeing the terms, participants are shown a fictitious business card next to a mobile phone, both displaying a security code (as shown in Figure 3). To become more familiar with the task, the experiment is started with 4 training tasks (each method once) not considered in the evaluation. The user's only task is to rate whether the shown fingerprints match by clicking on *Match* or *Doesn't Match* on the phone. Based on the condition assignment, participants see different approaches in a *randomized order*. We measure whether their answer was correct and their speed, i. e., the amount of time spent on the comparison. The experiment is concluded with a survey collecting feedback on the used approaches and the tasks and demographic information discussed in the "Results" section.

Type (Within-Group)	**Scheme (Between-Group)**
Alphanumeric	Hexadecimal `XOR` Base32
Numeric	Numeric
Unrelated Words	PGP `XOR` Peerio
Generated Sentences	Generated Sentences

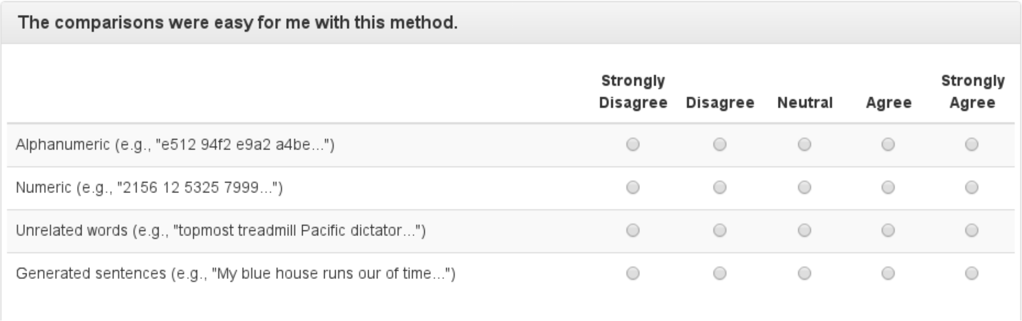
Table 2: To avoid confusion between too similar approaches (cf. Section 4.2), in our condition assignment, scheme types (left column) can consist of multiple representation schemes (right column). Each participant tests *only one* randomly assigned scheme of each type in a randomized order. .

4.3.1 Variables and Conditions

In the main experiment, the used *representation scheme* is our controlled independent variable whereas its values define our experiment conditions. In our additional chunking experiment, the *chunking size* is our controlled independent variable instead of the representation algorithm. During all tasks, we measure how fast participants perform with their given conditions and whether they are able to detect attacks by rating "incorrect" (*speed* and *accuracy* as our measured dependent variables).

In both experiments, each user had to perform 46 comparisons in total. To detect users clicking randomly, 2 obviously distinct comparisons were added to test a participant's attention. Training comparisons and attention tests are not included in the evaluation. Based on the feedback in our pre-study, we added tooltips during the training comparisons giving hints for language-based approaches telling the user that spelling attacks would not occur. We set the number of attacks to six: two obvious attacks where all bits are altered serving as control questions and 4 actual attacks with partial 80-bit preimages (one for each representation scheme). Participants failing at the control attacks are not considered in the evaluation but still received a payment if finishing all tasks. The major challenge in the study design is a high attack detection rate in general: most users perform comparisons correctly for the given attacker strength.

To avoid side effects, we chose fixed font size, color



The comparisons were easy for me with this method.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Alphanumeric (e.g., "e512 94f2 e9a2 a4be...")	○	○	○	○	○
Numeric (e.g., "2156 12 5325 7999...")	○	○	○	○	○
Unrelated words (e.g., "topmost treadmill Pacific dictator...")	○	○	○	○	○
Generated sentences (e.g., "My blue house runs our of time...")	○	○	○	○	○

Figure 4: A screenshot showing a statement rating in the post-experiment survey. Since the participants might not distinguish the different types, we have provided an example from their previous task.

and style, i. e., the same typeface for all fingerprint representations. In addition, we set fixed line breaks for sentences and word lists. In the main experiment, the same chunking style was used for all representations: For (alpha)numeric approaches a chunk consists of four characters separated by spaces. For word lists, we opted for a line break every four words. In the generated sentences representation, one sentence per line is displayed. We are aware that all these design decisions can have an effect on the comparison of the representations. However, our pre-study results show a significantly lower effect size. More importantly, we are mainly interested in comparing the concepts, therefore we did not vary any of the visual attributes like font size or style. In particular, differences resulting from the font's typeface have not been evaluated. Lund showed in his meta-analysis that there are no significant legibility differences between serif and sans serif typefaces [25].

Chunk-Size Testing A question was raised whether the chunking of a hexadecimal string plays a greater role in comparison to the different approaches. Thus, in addition to the main experiment testing different representation types, we conducted a second experiment with new participants testing different chunk sizes for the hexadecimal representation. Here, we used chunk-sizes ranging from 2 to 8 in addition to "zero-chunk size" (8 cases). The zero-chunk size means that no spaces have been included. To make the results more comparable, we opted for a similar design as done in the major experiment, i. e., we required the same amount of comparisons, used the same font settings, and had the same amount of attacks. For each participant, we assigned 4 out of 8 different chunk-sized randomly. Same as in the major experiment, all participants had to compare 46 fingerprints whereas the first 4 are considered as training comparisons, 4 attacks (one for each chunk size), and 2 control attacks with obviously distinct fingerprints.

The major experiment is followed by a survey focusing on self-reported user perception and opinions about the different approaches. This is the main reason we opted to compare as much as possible in a within-groups fashion and only selected a small number of conditions in total. Since users might not notice the difference between the various dictionary or alphabet approaches, we designed a mixed factorial design where the users would only get one of the alphabets/dictionaries (between-subjects) but they would test all different high-level systems (within-group) as depicted in Table 2. The between-group conditions have been assigned randomly with a uniform distribution. Since participants from our pre-study had difficulties to distinguish the different chunking approaches, we skipped the survey part in the chunk-size experiment.

4.3.2 Online Survey

The experiment was followed by an online survey gathering self-reported data and demographics from participants. To measure perception, we asked the participants whether they agreed with statements discussed in subsection 5.2 on a 5 point Likert scale: from strongly disagree to neural strongly agree as shown in Figure 4. Participants had to rate each representation type for all statements. Since users might not distinguish the different representation schemes, we provide an example from their previously finished task.

4.3.3 Statistical Testing

We opted for the common significance level of $\alpha = 0.05$. To counteract the multiple comparisons problem, we use the Holm-Bonferronicorrection for our statistical significance tests [18]. Consequently, all our p-values are reported in the corrected version.

We test the comparison duration with the Mann-Whitney-Wilcoxon (MWW) test (two-tailed). We opt for this significance test due to a few outliers, consequently a

Scheme	**Speed**				**Accuracy**			**Total**		
	mean [s]	med [s]	stdev	p-val	fail-rate	p-val	f-pos	fails	attacks	tests
Hexadecimal	11.2	10.0	6.4		10.44		0.49	50	479	4765
Hexadecimal – Base32	1.0	1.1	0.0	<0.001	−1.94	0.690	−2.09	12	32	269
Hexadecimal – Numeric	0.6	0.5	0.6	<0.001	−4.10	0.048	0.21	−9	−452	−4527
Hexadecimal – PGP	−1.8	−1.2	−1.0	<0.001	−1.65	0.690	−0.01	11	35	340
Hexadecimal – Peerio	2.5	2.7	0.8	<0.001	−4.69	0.048	0.08	22	−8	−91
Hexadecimal – Sentences	−1.1	−0.7	−0.6	<0.001	−7.45	<0.001	−0.99	22	−457	−4518
Base32	10.2	8.9	6.4		8.50		2.58	38	447	4496
Base32 – Hexadecimal	−1.0	−1.1	−0.0	<0.001	1.94	0.690	2.09	−12	−32	−269
Base32 – Numeric	−0.4	−0.6	0.6	<0.001	−2.16	0.404	2.30	−21	−484	−4796
Base32 – PGP	−2.8	−2.3	−1.0	<0.001	0.28	0.714	2.08	−1	3	71
Base32 – Peerio	1.5	1.6	0.8	<0.001	−2.75	0.404	2.17	10	−40	−360
Base32 – Sentences	−2.1	−1.8	−0.6	<0.001	−5.51	<0.001	1.10	10	−489	−4787
Numeric	10.6	9.5	5.8		6.34		0.28	59	931	9292
Numeric – Hexadecimal	−0.6	−0.5	−0.6	<0.001	4.10	0.048	−0.21	9	452	4527
Numeric – Base32	0.4	0.6	−0.6	<0.001	2.16	0.404	−2.30	21	484	4796
Numeric – PGP	−2.4	−1.7	−1.6	<0.001	2.45	0.404	−0.22	20	487	4867
Numeric – Peerio	1.9	2.2	0.2	<0.001	−0.59	0.714	−0.13	31	444	4436
Numeric – Sentences	−1.7	−1.2	−1.2	<0.001	−3.35	0.004	−1.20	31	−5	9
PGP	13.0	11.2	7.4		8.78		0.50	39	444	4425
PGP – Hexadecimal	1.8	1.2	1.0	<0.001	1.65	0.690	0.01	−11	−35	−340
PGP – Base32	2.8	2.3	1.0	<0.001	−0.28	0.714	−2.08	1	−3	−71
PGP – Numeric	2.4	1.7	1.6	<0.001	−2.45	0.404	0.22	−20	−487	−4867
PGP – Peerio	4.3	3.9	1.8	<0.001	−3.03	0.337	0.09	11	−43	−431
PGP – Sentences	0.7	0.5	0.4	<0.001	−5.79	<0.001	−0.98	11	−492	−4858
Peerio	8.7	7.3	5.6		5.75		0.41	28	487	4856
Peerio – Hexadecimal	−2.5	−2.7	−0.8	<0.001	4.69	0.048	−0.08	−22	8	91
Peerio – Base32	−1.5	−1.6	−0.8	<0.001	2.75	0.404	−2.17	−10	40	360
Peerio – Numeric	−1.9	−2.2	−0.2	<0.001	0.59	0.714	0.13	−31	−444	−4436
Peerio – PGP	−4.3	−3.9	−1.8	<0.001	3.03	0.337	−0.09	−11	43	431
Peerio – Sentences	−3.6	−3.4	−1.4	<0.001	−2.76	0.075	−1.07	0	−449	−4427
Sentences	12.3	10.7	7.0		2.99		1.48	28	936	9283
Sentences – Hexadecimal	1.1	0.7	0.6	<0.001	7.45	<0.001	0.99	−22	457	4518
Sentences – Base32	2.1	1.8	0.6	<0.001	5.51	<0.001	−1.10	−10	489	4787
Sentences – Numeric	1.7	1.2	1.2	<0.001	3.35	0.004	1.20	−31	5	−9
Sentences – PGP	−0.7	−0.5	−0.4	<0.001	5.79	<0.001	0.98	−11	492	4858
Sentences – Peerio	3.6	3.4	1.4	<0.001	2.76	0.075	1.07	0	449	4427

Table 3: Our experiment results showing the differences between the representation schemes. The top rows of each row group separated by a rule, show the raw performance of a baseline scheme, followed by italic rows showing a direct comparison delta. Greyed-out values are not backed by statistical significance. The columns *fail-rate* (undetected attacks) and *false-pos* (same string rated as an attack) display percentage values.

slightly skewed normal distribution, and a large amount of collected data. The *common language effect size* is shown by mean and median comparisons [26].

The attack detection rate is tested with a pairwise Holm-Bonferroni-corrected Barnard's exact test (`Exakt` package in `R`) achieving one of highest statistical power for 2x2 contingency tables [2].

Survey ratings are, again, tested by using the MWW significance test (two-tailed test). As has been shown in previous research [9], it is most suitable for 5-point Likert scales, especially if not multimodal distributed as in our survey results. In case two fingerprint representation schemes are statistically tested against each other, only participants encountering both schemes were considered.

5 Results

In this section, we present our results: our online study with 1047 participants has been conducted in August and September 2015. The study for testing the chunk size has been conducted in February 2016 with 400 participants. Starting with our online experiment evaluation showing the raw performance of users, we then present user perception results from the follow-up survey. Finally, we discuss the demographics of our participants.

5.1 Online Experiment

Participants who have not finished all comparisons or failed the attention tests were excluded from our eval-

Scheme	**Speed**			**Accuracy**			**Total**		
	mean [s]	med [s]	p-val	fail-rate	p-val	false-pos	fails	*attacks*	*tests*
Hexadecimal (4)	12.3	10.4		6.78		0.38	16	236	2360
hex (4) – hex (0)	−2.4	−2.6	<0.001	0.33	1.000	−0.28	−2	−17	−170
hex (4) – hex (2)	−0.3	−0.9	<0.001	1.37	1.000	0.00	−3	3	30
hex (4) – hex (3)	−0.3	0.1	0.362	−0.64	1.000	0.09	2	8	80
hex (4) – hex (5)	−1.4	−1.2	<0.001	1.01	1.000	−0.40	−2	5	50
hex (4) – hex (6)	−1.9	−1.8	<0.001	2.43	1.000	0.09	−5	8	80
hex (4) – hex (7)	−1.7	−1.8	<0.001	3.35	1.000	0.19	−8	−1	−10
hex (4) – hex (8)	−2.8	−3.2	<0.001	1.35	1.000	−0.12	−4	−10	−100

Table 4: Comparison of the chunking experiment results showing the differences between the representation schemes. The top row shows the raw performance of the hexadecimal scheme with a four-character chunking, followed by italic rows showing a direct comparison delta. Greyed-out values are not backed by statistical significance. The columns *fail-rate* (undetected attacks) and *false-pos* (same string rated as an attack) display percentage values.

uation: all participant compared 46 security codes in a randomized order, whereas 40 (10 of each scheme) were considered in the evaluation. The four training samples and the control questions are excluded. Few comparisons done in less than 2 seconds and more than one minute have been excluded. The reason for such can either be multiple clicks during the page load, or external interruptions of the participants. None of the attack could be successfully detected in under 4 seconds.

Our experiment results, summarized in Table 3, show the raw performance of all schemes regarding their speed, accuracy and false-positive rate. The top rows of each row group, separated by a rule, show the raw performance of a representation scheme as baseline (negative values indicate lower values than the baseline). The following rows show a direct comparison delta between between two schemes. The speed column group consists of the mean and median (in seconds), the standard deviation and the according p-values for a direct comparison. The fail-rate column shows the rate of the undetected attacks with the according p-values for a direct comparison. The total column group simply shows the total numbers of tests, attacks and undetected attacks.

The results show that the average time spent on comparisons plays only a minor role among the schemes: 4.3 s difference between the best and the worst scheme. Note that the Peerio word-list scheme performed best with 8.7 s mean whereas the PGP word list performed worst with 13 s mean ($p < 0.001$).

However, there is a clear effect regarding the attack detection rate (see Table 3). All alternative key-fingerprint representations performed better than the state-of-the-art hexadecimal representation, where 10.1% of attacks have not been detected by the users. Previous work shows similar numbers for Base32 [19]. To our surprise, the numeric approach performs better in both categories: it features an attack detection rate of 93.57% ($p < 0.01$) and an average speed of 10.6*s* ($p < 0.001$). Generated sentences achieved the highest attack detection rate of 97.97% with a similar average speed as the hexadecimal scheme. On the downside, this scheme has produced a slightly higher false-positive rate. We found that the false positives occurred mostly with longer sentences where there has been a line break on the phone mock-up due to portrait orientation. This is a realistic problem of this system if used with portrait orientation and not a problem with our mock-up in itself. Improvements on making the sentences shorter could mitigate this situation.

Chunk-Size Experiment

Table 4 summarizes the results of our secondary chunk-size experiment. As can be seen, no statistically significant results have been achieved for the *attack detection fail-rate* (undetected attacks by end users). However, we observed that the chunk sizes with 3 and 4 characters performed best in speed, even though the effect sizes were minor: only 3.3 seconds difference with similar standard deviations between the best and worst chunk size setting.

Firstly, we notice that despite the same attack strength as in our major experiment, participants were able to detect more attacks. We suspect that the higher attack detection rate is based on (1) a higher learning effect due to the same scheme for all comparisons and (2) in contrast to our major study, participants had a slightly higher drop-out rate and thus only more motivated participants were considered. This is supported by the numbers in the total tests column of Table 4: here, we can see that for the zero-chunking and chunking with 8 characters less tests have been performed. This is based on the fact that although the chunk sizes have been assigned almost uniformly, participants assigned with harder chunk settings often dropped out before even finishing their entire task.

More importantly, our results also support the claim from our pre-study: The chunking parameter in hexadecimal strings plays only a minor role in the *attack detection fail-rate*.

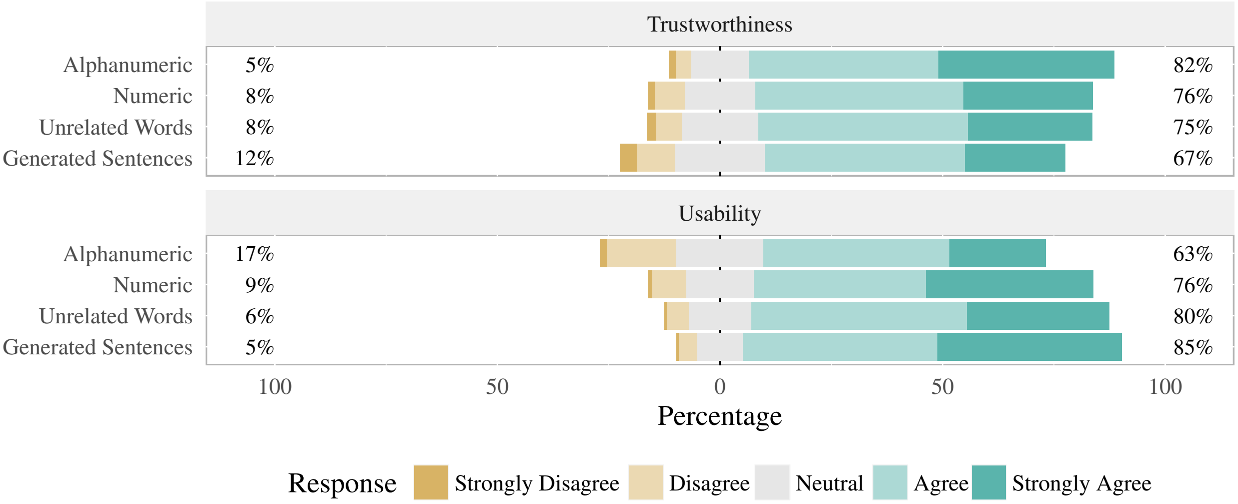


Figure 5: Aggregated survey results for statement rating regarding the usability and trustworthiness.

5.2 Online Survey

To measure the usability and trustworthiness of all representation schemes, we asked our participants whether they agreed with the following statements:

S_1 The comparisons were easy for me with this method

S_2 I am confident that I can make comparisons using this method without making mistakes

S_3 I think making comparisons using this method would help me keep my communications secure

S_4 I was able to do the comparisons very quickly with this method

S_5 I found this method difficult to use

S_6 Overall, I liked this method

We mixed positive and negative statements, e. g., S_1 and S_5, to create a more robust measure. S_6 is used to calculate the overall ranking of the different representation schemes.

Figure 5 shows the aggregated results where the usability statements are grouped to one usability feature and the trustworthiness derived from the rating on the statement S_3. Negative statement ratings have been inverted for a better comparison. Figure 6 shows the rating results for each specific statement in the survey. The order of the tested schemes has been chosen randomly, but was kept consistent across all statements. Same as in our online experiment evaluation, the pairwise statistical tests are Holm-Bonferroni corrected. In case of a direct statistical test between two schemes, only users encountering both schemes have been considered. All in all, the usability perception of the participants is almost consistent with the performance results from the experiment.

To measure the perception of the task difficulty, we asked the participants whether they agreed with the statements S_1, S_2 and S_3 respectively. As illustrated in Figure 6 in the Appendix A, the effect size between the different approaches is low. However, the participants were more likely to agree that language-based representation schemes are easier to use. For instance, we see that in comparison to the alphanumeric schemes (average rating of 3.4), word list (average rating of 3.9, $p < 0.001$) and generated sentence schemes (average rating of 4.2, $p < 0.001$) are rated to be easier by our participants (S_1, S_5). While the experiment results of the sentence generators clearly outperformed all other approaches, they also were rated better by the participants. Same applies for the low-performing hexadecimal and Base32 schemes which clearly received lower ratings. Consistently with the surprising performance results in the experiment, the numeric scheme is also considered to be easier by many participants: average rating of 3.9 and $p < 0.001$.

The sentence generator scheme achieved the highest user confidence rating “making comparisons without any mistakes” (S_2, $p < 0.001$ for all pairwise comparisons). The participants’ perception is consistent with the experiment results where the word-list-based and sentence generator schemes lead to higher attack detection rates.

The ratings for S_4 illustrate that more complex representation schemes from the user's point of view, such as hexadecimal and Base32, are considered to be more secure by participants, even though all approaches provide the same level of security.

5.3 Demographics

A total of 1047 users participated in the online study while only 1001 have been considered in the evaluation due to our two control questions. Out of the evaluated participants, 534 participants were male, 453 were female, 4 chose other while the rest opted to not give any information. No significant difference between genders could be found, with a subtle trend of a higher accuracy for women and higher speed among men. The median age was 34 (34.4 average) years, while 34 participants chose not to answer (no statistically significant differences between ages).

A total of 39 people reported to have "medical conditions that complicated the security code comparisons (e. g., reading disorders, ADHD, visual impairments, etc.)" with a slightly higher undetected attack rate (statistically insignificant due to small sample size and thus low statistical power).

The majority of the participants stated to have a Bachelor's degree (399 of 1047) as their highest education whereas 34% chose not to answer. 931 participants have started our HIT but stopped early during the experiment (mostly after the first few comparisons). 160 users reported the general task to be annoying.

6 Discussion

The results of our study show that while there are subtle speed variations among all approaches, the attack detection rate and user perception for the current state-of-the-art hexadecimal key-fingerprint representation is significantly lower than those of most alternative representation schemes. Language-based representations (with the exception of the PGP word list) show improved user behaviour leading to a higher detection rate of attacks. To improve the usability of key-fingerprints, we propose the following takeaways based on our study results.

6.1 Takeaways

Our results show that all representation schemes achieve a high accuracy (high attack detection rate) and can be performed quickly by users. As expected, language-based fingerprint representations are more resilient against attacks (higher attack detection rate) and achieve better usability scores. Among all conditions, alphanumeric approaches performed worse and have been outperformed. For instance, the *numeric* representation was more suitable than hexadecimal and Base32. The raw performance results suggest a similar speed for the numeric representation with a higher attack detection rate, and it also has received better usability ratings from end-users.

Our chunking experiment has shown that chunk-sizes play only a minor role in improving attack detection rates (we could not find statistically significant differences). However, if a hexadecimal representation is used chunks of 3 and 4 characters perform best.

As shown by the word list representations, the comparison speed can be increased by larger dictionaries leaving room for improvement in this area. Even though all representation schemes provide the same level of security, exotic looking solutions are considered to be more secure by end users.

6.2 Limitations

Most importantly, our study design *does not test* whether end users *are actually willing to compare any fingerprints* in practice. We only aim to study how easy different representations are to compare from the users' point of view.

As with any user study conducted with MTurk, there is concern about the external validity of the results: users in the real world might show different behaviour. This is mainly because of two reasons: (1) in practice fingerprint comparisons will seldom occur in a such frequency, and (2) when performed in practice play a more important role than just participating in an anonymous online study. Additionally, MTurkers have been shown to be more tech-savvy and are better in solving textual and visual tasks in comparison to the average population. Thus, they could have performed better in most of the comparison conditions than the average population. It is also known that some MTurkers just "click through" studies to get the fee and thus distort study results. Our counterbalanced study design with included control questions and statistical significance tests mitigate this effect. For instance, we excluded 46 out of 1047 participants from our main study part based on these questions being answered incorrectly.

Due to the within-group part of our factorial design, many parameter choices such as different fonts, font sizes, attack rates, etc. could not be considered. These are, however, interesting avenues for future work. As shown in our additional chunking experiment, another challenge in testing different parameters is the high attack detection rate, where subtle changes would require a high amount of users to produce statistically significant results.

Due to the anonymous nature of online studies, it is also impossible to reliably tell which languages a participant is fluent in. We specified that we only wanted participants from English-speaking countries, however we had no way of checking compliance except by relying on self-reported data. Language-based representation approaches might induce additional barriers for non-native speakers, e. g., due to unknown or unfamiliar words.

7 Conclusion and Future Work

We evaluated six different key-fingerprint representation types with regards to their comparison speed, attack detection accuracy and usability, which encompasses attack detection but also resilience against human errors in short-term memory. An online study with 1047 participants was conducted to compare numeric, alphanumeric (Hexadecimal and Base32), word lists (PGP and Peerio), as well as generated sentences representation schemes for key-fingerprint verification. All fingerprint representations were configured to offer the same level of security with the same attacker strength.

Our results show that usage of the large word lists (as used in Peerio) lead to the fastest comparison performance, while generated sentences achieved highest attack detection rates. In addition, we found that additional parameters such as chunking of characters plays only a minor role in the overall performance. The widely-used hexadecimal representation scheme performed worst in all tested categories which indicates that it should be replaced by more usable schemes. Unlike proposals which call for radically new fingerprint representations, we studied only textual fingerprint representations, which means that the results of our work can be directly applied to various encryption applications with minimal changes needed. Specifically, no new hardware or complex software is required: applications merely need to replace the strings they output to achieve a significant improvement in both attack-detection accuracy and usability.

There are various interesting areas of future work. Firstly, we chose to study only a selected sample from the design space of fingerprint representations in a within-subjects design, so we could facilitate a direct comparison between the different classes of fingerprints. Further work exploring line breaks, font settings, dictionaries, different attacker strengths, etc. will likely lead to further improvement possibilities.

While this work shows that there are better ways to represent key-fingerprints than currently being used, it does not explore what can be done to motivate more users to actually compare the fingerprints in the first place. Follow-up studies to research this important question are naturally an interesting and vital area of research.

Acknowledgments

The authors would like to thank the anonymous reviewers for their insightful comments, and Trevor Perrin, Jake McGinty, Tom Ritter and Skylar Nagao for their discussion and excellent feedback.

References

[1] BADDELEY, A. Working memory. *Science 255*, 5044 (1992), 556–559.

[2] BARNARD, G. Significance tests for 2×2 tables. *Biometrika* (1947), 123–138.

[3] BIRYUKOV, A., DINU, D., AND KHOVRATOVICH, D. Argon2: the memory-hard function for password hashing and other applications. Tech. rep., Password Hashing Competition (PHC), December 2015.

[4] BONNEAU, J., AND SCHECHTER, S. Towards reliable storage of 56-bit secrets in human memory. In *Proceedings of the 23rd USENIX Security Symposium* (August 2014).

[5] BREITMOSER, V. pgp-vanity-keygen. `https://github.com/Valodim/pgp-vanity-keygen`, 2014.

[6] BUCKNER, R. L., PETERSEN, S. E., OJEMANN, J. G., MIEZIN, F. M., SQUIRE, L., AND RAICHLE, M. Functional anatomical studies of explicit and implicit memory retrieval tasks. *The Journal of Neuroscience 15*, 1 (1995), 12–29.

[7] CALLAS, J., DONNERHACKE, L., FINNEY, H., SHAW, D., AND THAYER, R. OpenPGP Message Format. RFC 4880 (Proposed Standard), Nov. 2007. Updated by RFC 5581.

[8] CRANNELL, C., AND PARRISH, J. A comparison of immediate memory span for digits, letters, and words. *The Journal of Psychology 44*, 2 (1957), 319–327.

[9] DE WINTER, J. C., AND DODOU, D. Five-point Likert items: t test versus Mann-Whitney-Wilcoxon. *Practical Assessment, Research & Evaluation 15*, 11 (2010), 1–12.

[10] DHAMIJA, R. Hash visualization in user authentication. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2000), CHI EA '00, ACM, pp. 279–280.

[11] DIERKS, T., AND RESCORLA, E. The Transport Layer Security Protocol Version 1.2. RFC 5246, Aug. 2008. Updated by RFCs 5746, 5878, 6176.

[12] ELECTRONIC FRONTIER FOUNDATION. Secure Messaging Scorecard. `https://www.eff.org/secure-messaging-scorecard`, 2014.

[13] ELLISON, C., ET AL. Establishing identity without certification authorities. In *USENIX Security Symposium* (1996), pp. 67–76.

[14] FARB, M., LIN, Y.-H., KIM, T. H.-J., MCCUNE, J., AND PERRIG, A. Safeslinger: easy-to-use and secure public-key exchange. In *Proceedings of the 19th annual international conference on Mobile computing & networking* (2013), ACM, pp. 417–428.

[15] GALBRAITH, J., AND THAYER, R. The Secure Shell (SSH) Public Key File Format. RFC 4716 (Informational), Nov. 2006.

[16] GOODRICH, M. T., SIRIVIANOS, M., SOLIS, J., TSUDIK, G., AND UZUN, E. Loud and clear: Human-verifiable authentication based on audio. In *Distributed Computing Systems, 2006. ICDCS 2006. 26th IEEE International Conference on* (2006), IEEE, pp. 10–10.

[17] GUTMANN, P. Do users verify SSH keys? *USENIX;login: 36*, 4 (2011).

[18] Holm, S. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics* (1979), 65–70.

[19] Hsiao, H.-C., Lin, Y.-H., Studer, A., Studer, C., Wang, K.-H., Kikuchi, H., Perrig, A., Sun, H.-M., and Yang, B.-Y. A study of user-friendly hash comparison schemes. In *Computer Security Applications Conference, 2009. ACSAC'09. Annual* (2009), IEEE, pp. 105–114.

[20] Josefsson, S. The Base16, Base32, and Base64 Data Encodings. RFC 3548 (Informational), July 2003.

[21] Josefsson, S. The Base16, Base32, and Base64 Data Encodings. RFC 4648, Oct. 2006.

[22] Juola, P. Whole-word phonetic distances and the PGPFone alphabet. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on* (1996), vol. 1, IEEE, pp. 98–101.

[23] Keith, M., Shao, B., and Steinbart, P. A behavioral analysis of passphrase design and effectiveness. *Journal of the Association for Information Systems 10*, 2 (2009), 2.

[24] Laurie, B., Langley, A., and Kasper, E. Certificate Transparency. RFC 6962 (Experimental), June 2013.

[25] Lund, O. *Knowledge construction in typography: the case of legibility research and the legibility of sans serif typefaces.* PhD thesis, The University of Reading, Department of Typography & Graphic Communication., 1999.

[26] McGraw, K. O., and Wong, S. A common language effect size statistic. *Psychological bulletin 111*, 2 (1992), 361.

[27] Melara, M. S., Blankstein, A., Bonneau, J., Freedman, M. J., and Felten, E. W. CONIKS: A Privacy-Preserving Consistent Key Service for Secure End-to-End Communication. Tech. Rep. 2014/1004, Cryptology ePrint Archive, December 2014.

[28] [messaging] Mailing-List Archive. Usability of Public-Key Fingerprints. `https://moderncrypto.org/mail-archive/messaging/2014/000004.html`, 2014.

[29] Miller, G. A. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review 63*, 2 (1956), 81.

[30] Namecoin Project. Namecoin. `http://namecoin.info`, Nov. 2014.

[31] Ogden, C. K., et al. System of Basic English. *Self-published* (1934).

[32] Olembo, M. M., Kilian, T., Stockhardt, S., Hülsing, A., and Volkamer, M. Developing and testing a visual hash scheme. In *EISMC* (2013), pp. 91–100.

[33] [openpgp] IETF Mail Archive. Fingerprints. `https://mailarchive.ietf.org/arch/msg/openpgp/2C9gTsxTgh29W8VX8x70OYZqfUY`, 2015.

[34] Percival, C. Stronger key derivation via sequential memory-hard functions. *Self-published* (2009).

[35] Perrig, A., and Song, D. Hash visualization: A new technique to improve real-world security. In *International Workshop on Cryptographic Techniques and E-Commerce* (1999), pp. 131–138.

[36] Pingel, I., Irving, A., generalmanager, Wikinaut, tinloaf, Farb, M., and jpopplewell. Fingeprint exchange - issue #826 - whispersystems/textsecure - github.

[37] Plasmoid. Fuzzy Fingerprints: Attacking Vulnerabilities in the Human Brain. `http://www.thc.org/papers/ffp.html`, Oct. 2003.

[38] Rogers, M., and Perrin, T. Key-Fingerprint Poems. `https://moderncrypto.org/mail-archive/messaging/2014/000125.html`, 2014.

[39] Ryan, M. D. Enhanced certificate transparency and end-to-end encrypted mail. In *NDSS* (2014), NDSS.

[40] Shay, R., Kelley, P. G., Komanduri, S., Mazurek, M. L., Ur, B., Vidas, T., Bauer, L., Christin, N., and Cranor, L. F. Correct horse battery staple: Exploring the usability of system-assigned passphrases. In *Proceedings of the Eighth Symposium on Usable Privacy and Security* (2012), ACM, p. 7.

[41] Unger, N., Dechand, S., Bonneau, J., Fahl, S., Perl, H., Goldberg, I., and Smith, M. Sok: Secure messaging. In *Security and Privacy (SP), 2015 IEEE Symposium on* (2015), IEEE, pp. 232–249.

[42] Vasco.com. `http://www.vasco.com/company/about_vasco/press_room/news_archive/2011/news_diginotar_reports_security_incident.aspx`, Sept. 2011.

[43] WhatsApp. Encryption Overview. `https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf`, Apr. 2016.

[44] Ylonen, T., and Lonvick, C. The Secure Shell Transport Layer Protocol. RFC 4253, Jan. 2006. Updated by RFC 6668.

[45] Zimmermann, P., Johnston, A., and Callas, J. ZRTP: Media Path Key Agreement for Unicast Secure RTP. RFC 6189 (Informational), Apr. 2011.

A Appendix

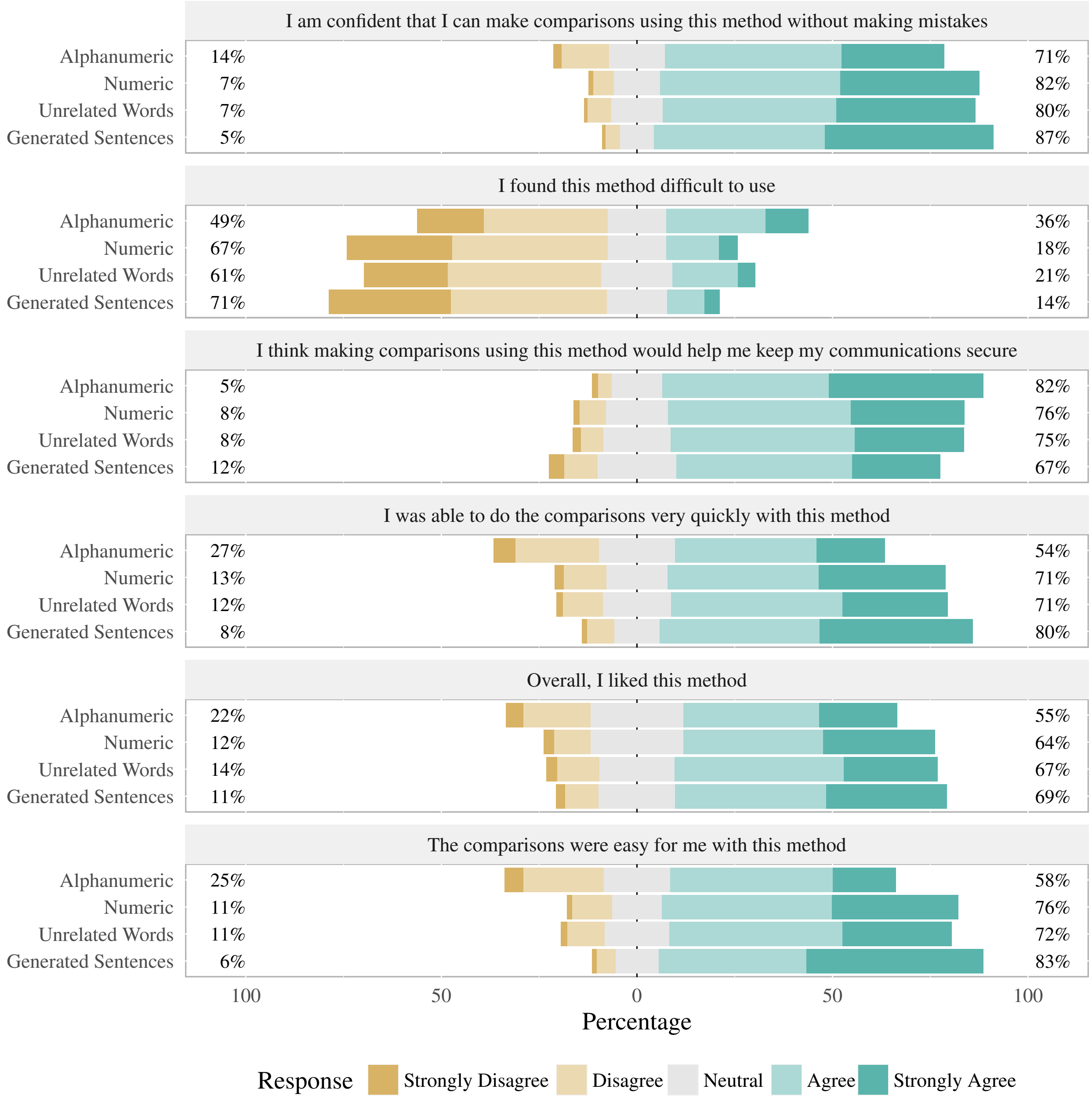


Figure 6: Survey results for all statement ratings