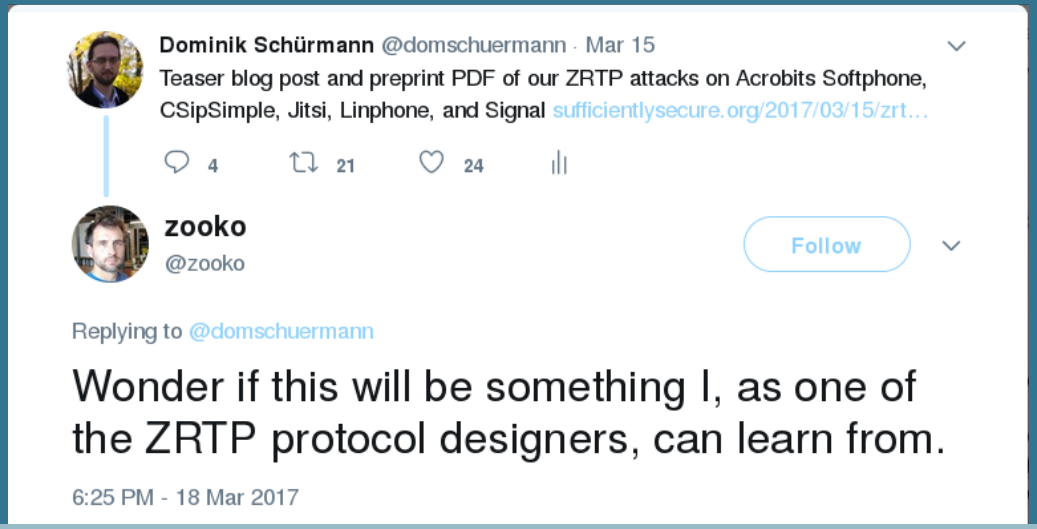
Technische Universität Braunschweig

Institute of Operating Systems and Computer Networks



Wiretapping End-to-End Encrypted VoIP Calls

Real-World Attacks on ZRTP

Dominik Schürmann, Fabian Kabus, Gregor Hildermeier, Lars Wolf,
2017-07-18

End-to-End Security for Voice Calls

No End-to-End Security

- PSTN (Public Switched Telephone Network)
- SIP + (S)RTP (Session Initiation Protocol + Secure Real-Time Transport Protocol)

End-to-End Security for Voice Calls

No End-to-End Security

- PSTN (Public Switched Telephone Network)
- SIP + (S)RTP (Session Initiation Protocol + Secure Real-Time Transport Protocol)

End-to-End Encryption

- SIP + DTLS-SRTP (SIP + Datagram Transport Layer Security-SRTP)

End-to-End Security for Voice Calls

No End-to-End Security

- PSTN (Public Switched Telephone Network)
- SIP + (S)RTP (Session Initiation Protocol + Secure Real-Time Transport Protocol)

End-to-End Encryption

- SIP + DTLS-SRTP (SIP + Datagram Transport Layer Security-SRTP)

End-to-End Encryption & Authentication

- SIP + SRTP + ZRTP

End-to-End Security for Voice Calls

No End-to-End Security

- PSTN (Public Switched Telephone Network)
- SIP + (S)RTP (Session Initiation Protocol + Secure Real-Time Transport Protocol)

End-to-End Encryption

- SIP + DTLS-SRTP (SIP + Datagram Transport Layer Security-SRTP)

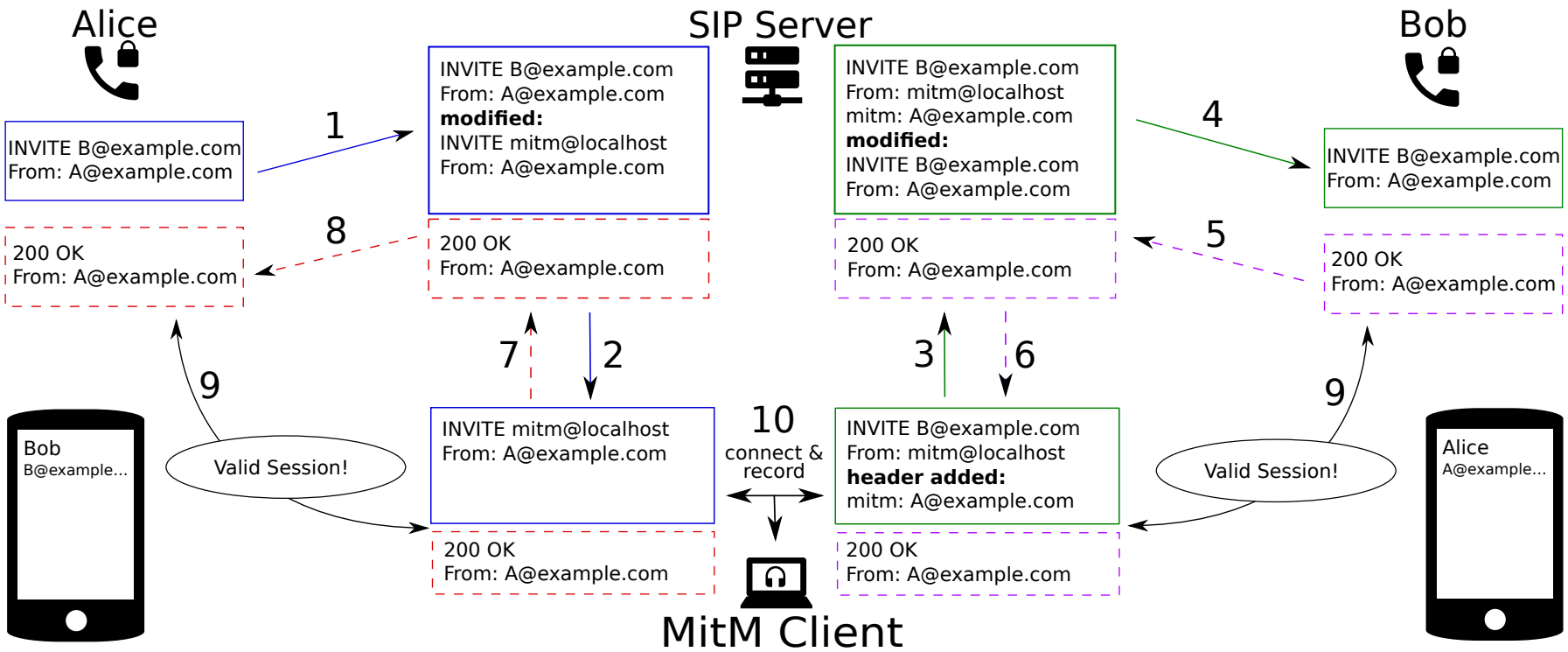
End-to-End Encryption & Authentication

- SIP + SRTP + ZRTP

wiretapping difficulty

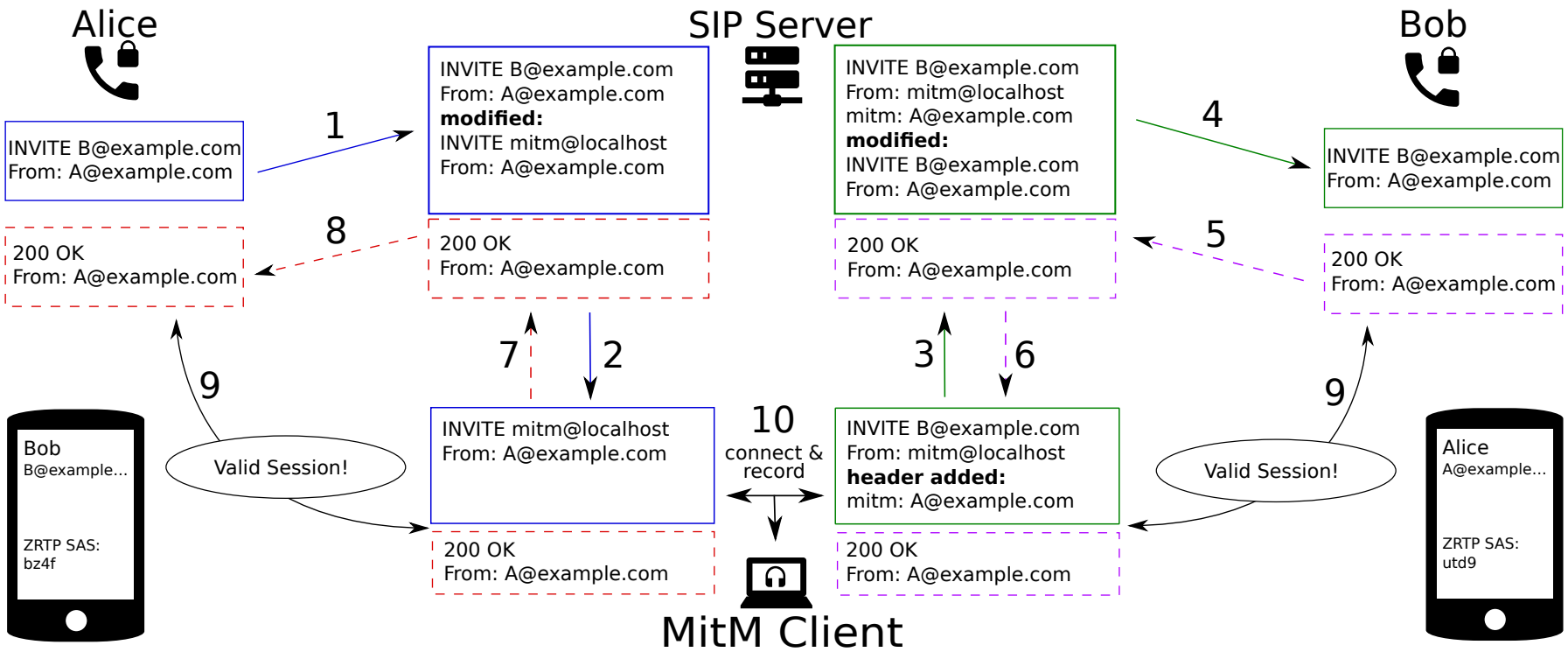
Man-in-the-Middle (Evil Operator)

SIP with Encryption-only:



Man-in-the-Middle (Evil Operator)

Encryption & Authentication with ZRTP:



ZRTP Attacks

ZRTP

- Complex Protocol
- Authenticates Diffie-Hellman key exchange
- Authentication by comparison of Short Authentication Strings (SAS)
- Hash Commitment constraints online-attacker to one try per call

Evaluation of Real-World Implementations

- Excluded closed-network implementations
- Excluded attacks with speech synthesis
- Assume correctly compared SAS

Evaluation

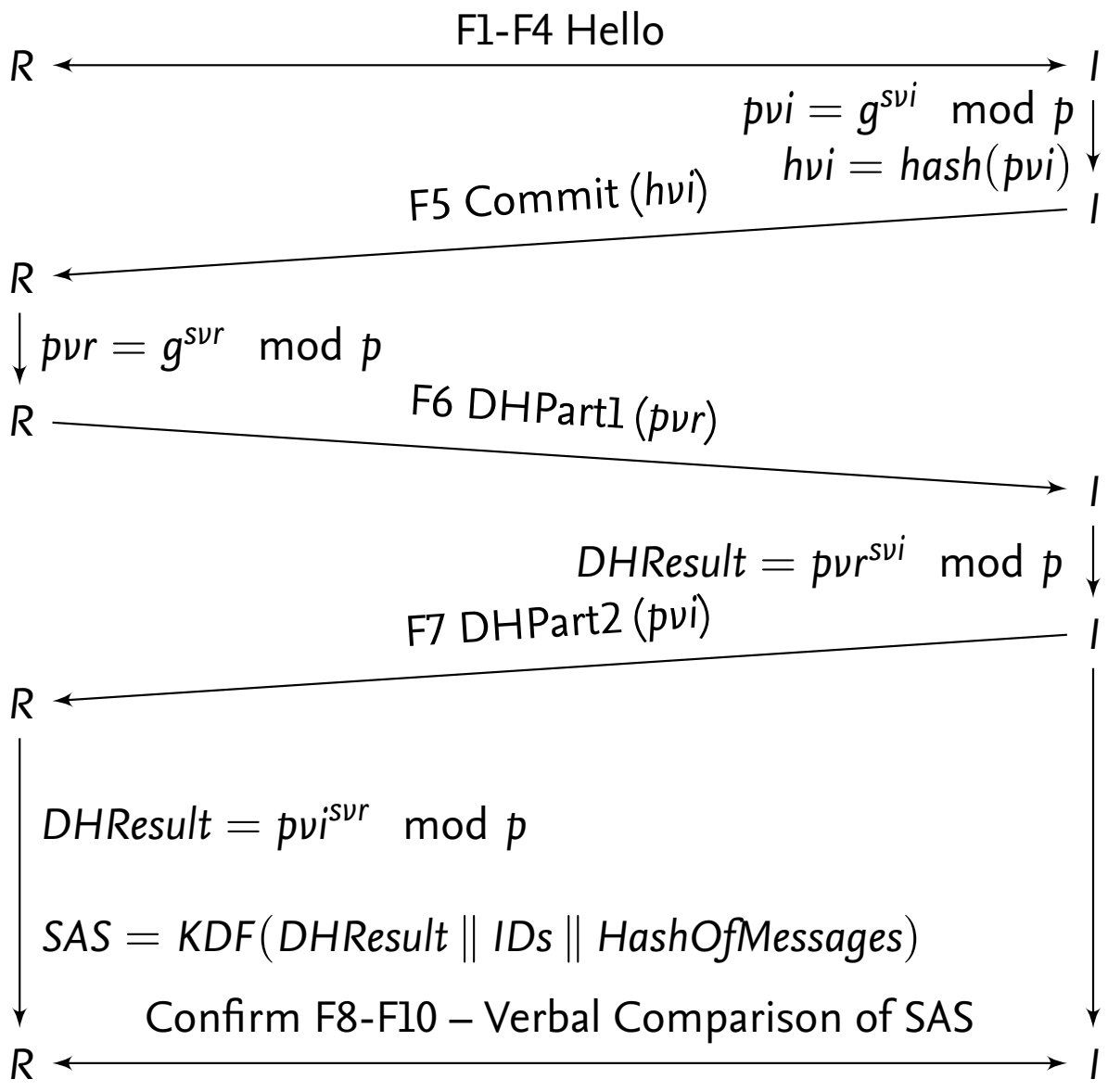
Apps

Application	OS	Version	Library
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
Signal	Android	3.15.2	-
Signal	iOS	2.6.4	-

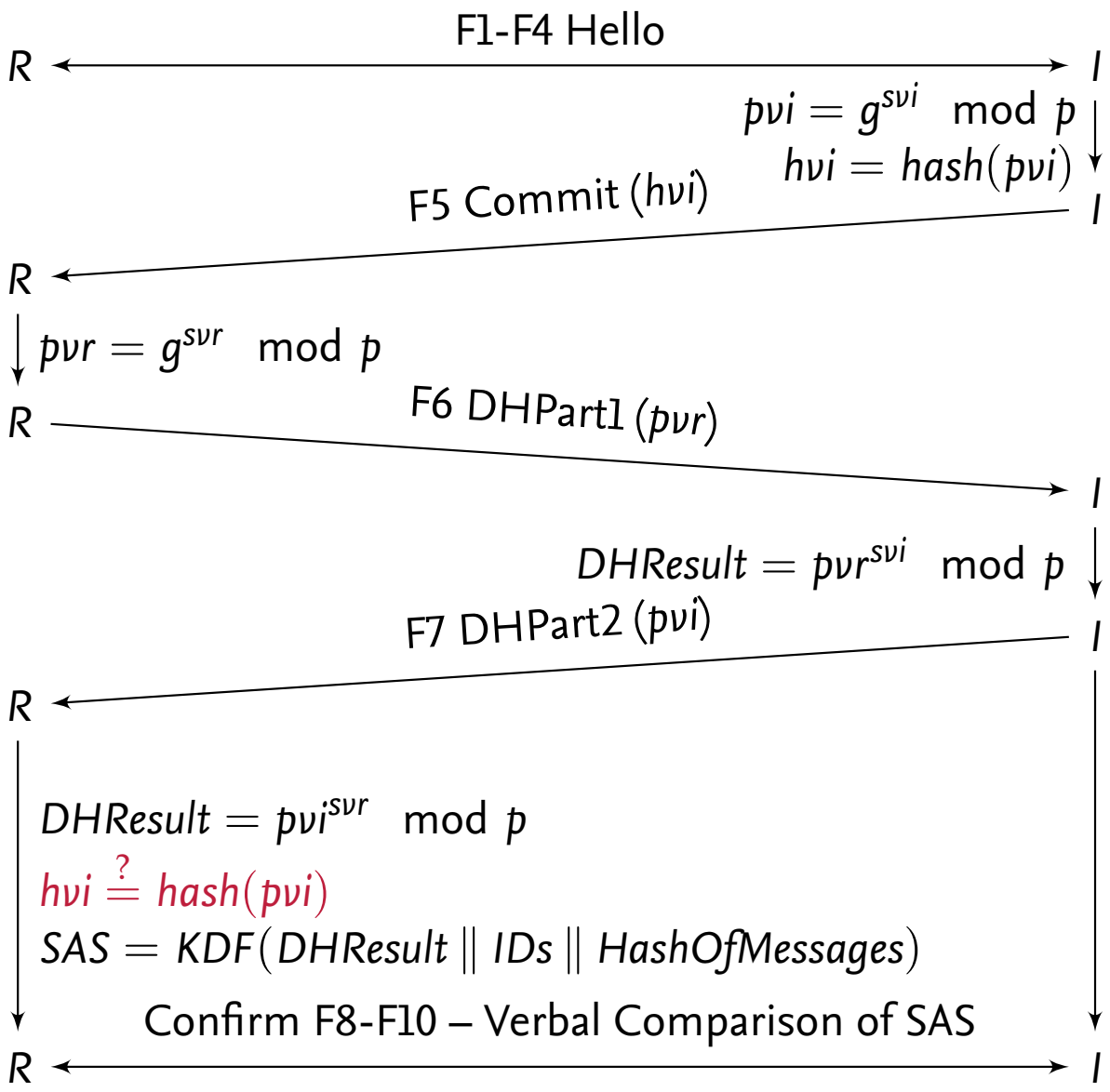
Tests

- Paper: 7 protocol tests, 4 non-protocol tests
- Presentation: Most interesting results

ZRTP in a Nutshell (Highly Simplified)



Check for Invalid Commit



Invalid Commit: Linphone

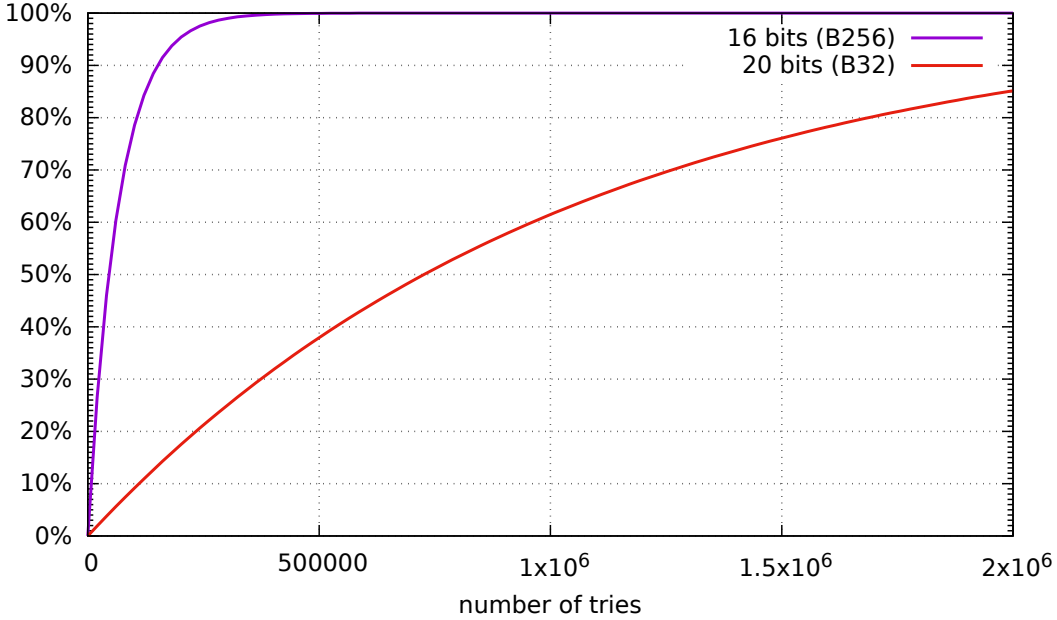


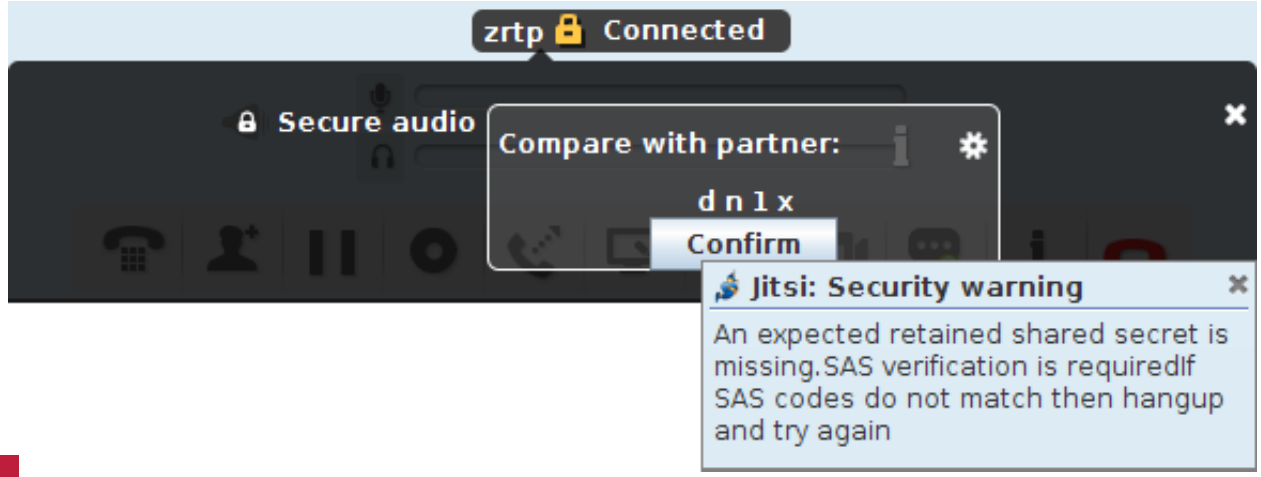
Figure: Linphone CVE-2016-6271: Probability of hitting a targeted SAS

RFC: Error on Invalid Shared Secret

- ZRTP stores secrets when user confirms SAS
- Cache: ZRTP ID assigned to $rs1 = KDF(DHResult)$ (highly simplified)
- Next call no longer requires Diffie-Hellman and no SAS comparison

RFC

"If either party discovers a cache mismatch, the user agent who makes this discovery must treat this as a possible security event and MUST alert their own user that there is a heightened risk of a MiTM attack [...]"



RFC: Error on Invalid Shared Secret

- Questionable requirement in RFC
- CSipSimple, Linphone do not implement this

Bug in Jitsi (ZRTP4J)

- A new cache entry copies the secrets and flags from the last saved one
- Invalid security warning is raised for new clients

1 src/gnu/java/zrtp/zidfile/ZidFile.java

```
@@ -250,6 +250,7 @@ public synchronized ZidRecord getRecord(byte[] zid) {
250 250         // If we reached end of file, then no record with matching ZID
251 251         // found. We need to create a new ZID record.
252 252         if (!numRead) {
    253 +           rec = new ZidRecord();
253 254             rec.setIdentifier(zid);
254 255             rec.setValid();
255 256             try {
```

Shared Man-in-the-Middle

Attack

1. Call between Eve & Alice, confirm SAS $\Rightarrow$ $rs1_A$ for Eve in Alice' cache
2. Call between Eve & Bob, confirm SAS $\Rightarrow$ $rs1_B$ for Eve in Bob's cache
3. Eve conducts MitM attack (evil operator) $\Rightarrow$ No SAS confirmation, Eve has $rs1_A$, $rs1_B$ in her cache
4. SIP addresses shown: Alice: B@example.com, Bob: A@example.com

Shared Man-in-the-Middle

Attack

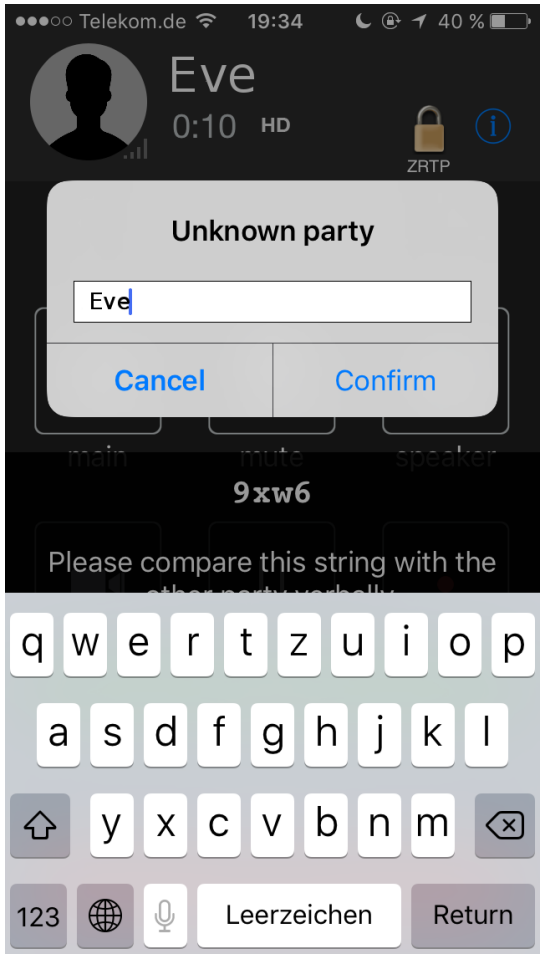
1. Call between Eve & Alice, confirm SAS $\Rightarrow$ $rs1_A$ for Eve in Alice' cache
2. Call between Eve & Bob, confirm SAS $\Rightarrow$ $rs1_B$ for Eve in Bob's cache
3. Eve conducts MitM attack (evil operator) $\Rightarrow$ No SAS confirmation, Eve has $rs1_A$, $rs1_B$ in her cache
4. SIP addresses shown: Alice: B@example.com, Bob: A@example.com

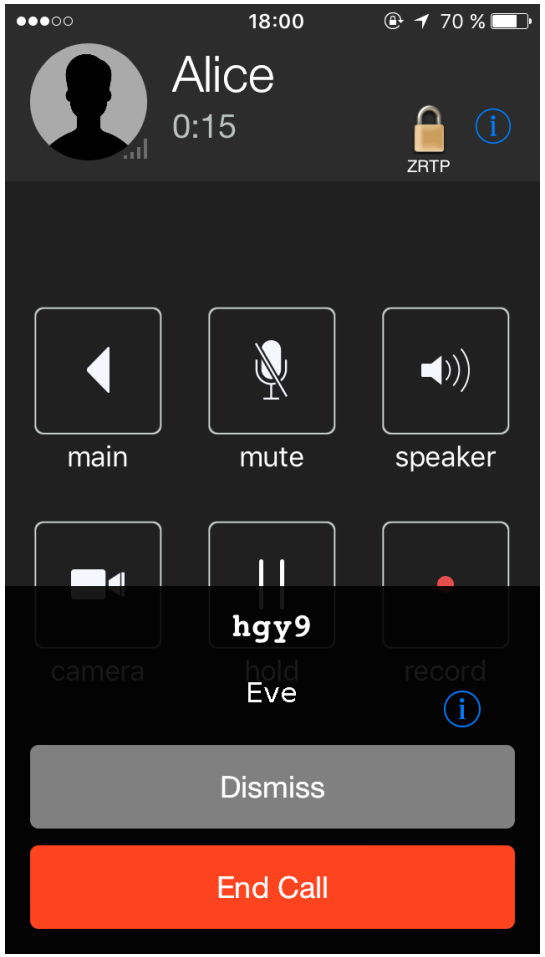
Why Does This Work?

- No ID binding to outer protocol
- ZRTP works independent of SIP addresses with random IDs
  $\Rightarrow$ Cache uses ZRTP ID for lookup
- Alice and Bob's cache lookup by Eve's ZRTP ID

Shared Man-in-the-Middle

- Signal: No cache ⇒ Secure
- Acrobits Softphone: RFC-compliant protection
- Other implementations: Insecure





Conclusion

Current Status

- CVE-2016-6271 responsibly disclosed on 2016-07-05, fixed in Linphone 3.2.04
- Upstream fix for Jitsi always reading the last entry from the ID cache
- Signal no longer uses ZRTP (independent decision)

Future

- Most apps fallback to insecure mode
- Discussion about shared MitM attack
- Discussion about security indicators

Conclusion

Current Status

- CVE-2016-6271 responsibly disclosed on 2016-07-05, fixed in Linphone 3.2.04
- Upstream fix for Jitsi always reading the last entry from the ID cache
- Signal no longer uses ZRTP (independent decision)

Future

- Most apps fallback to insecure mode
- Discussion about shared MitM attack
- Discussion about security indicators

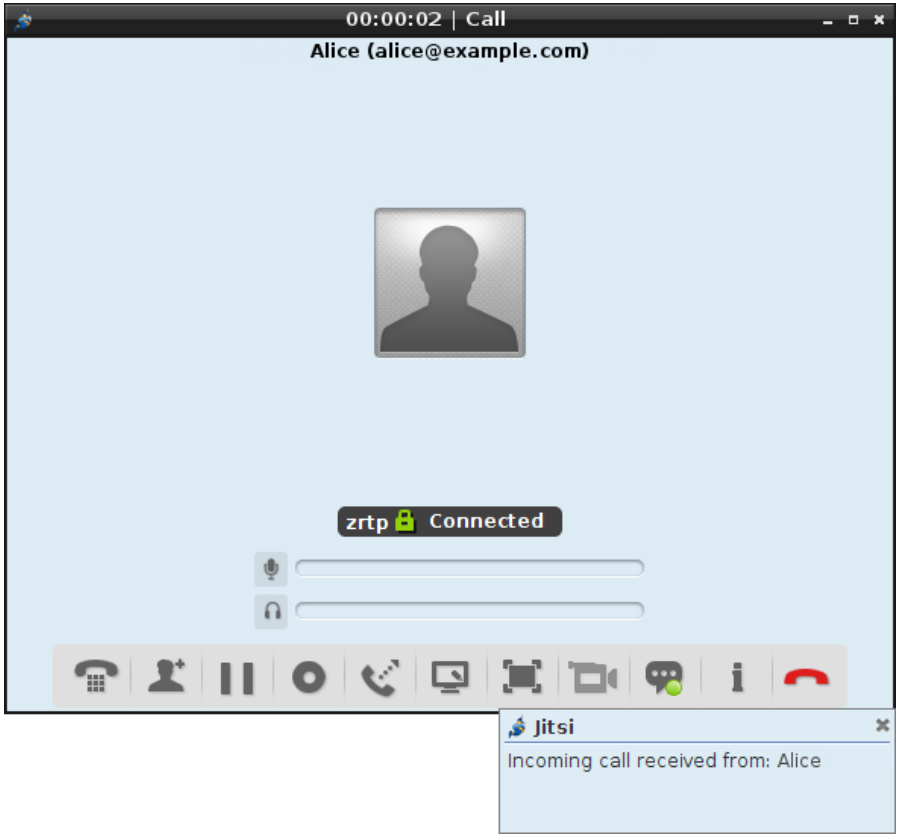
Any questions?

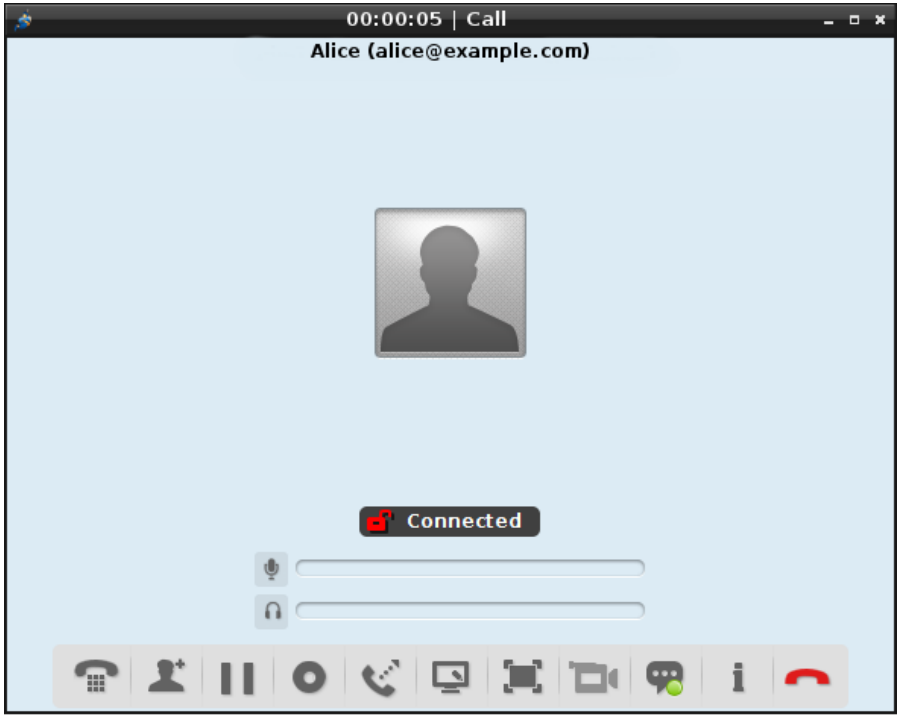
Dominik Schürmann <schuermann@ibr.cs.tu-bs.de>
Twitter: @domschuermann

Quiz Time: Security Indicators

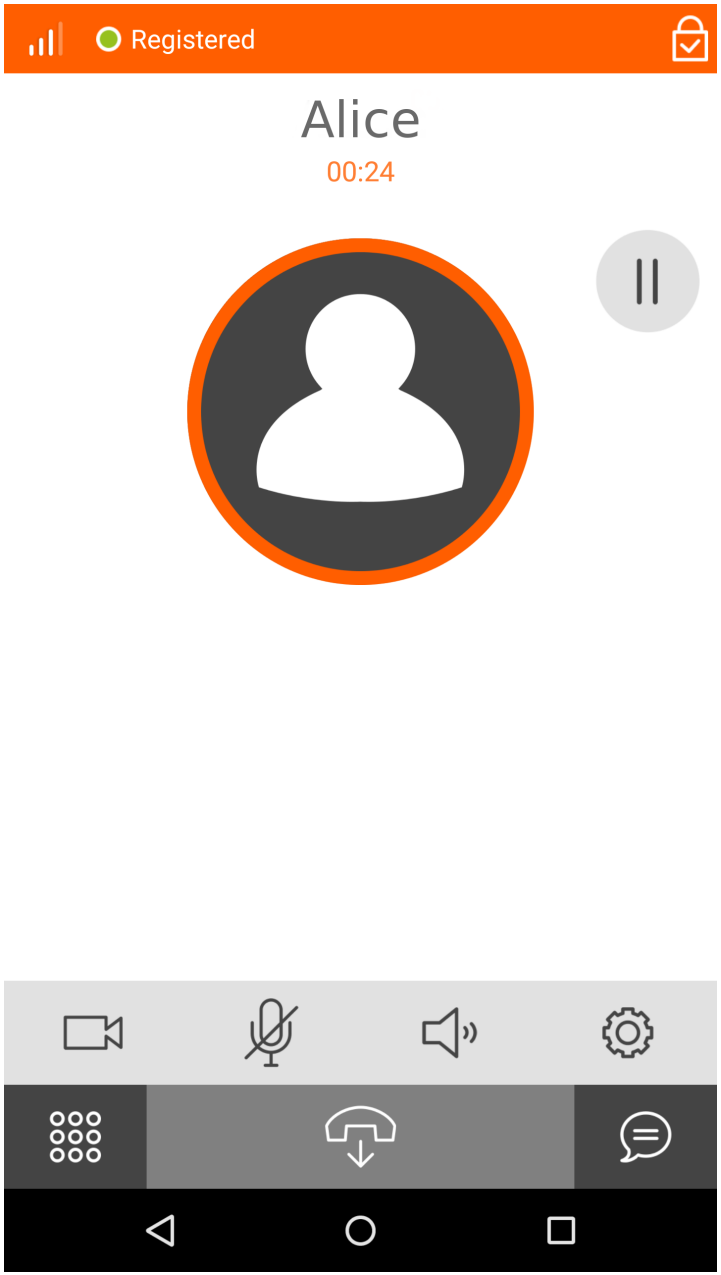
Are you end-to-end secure?

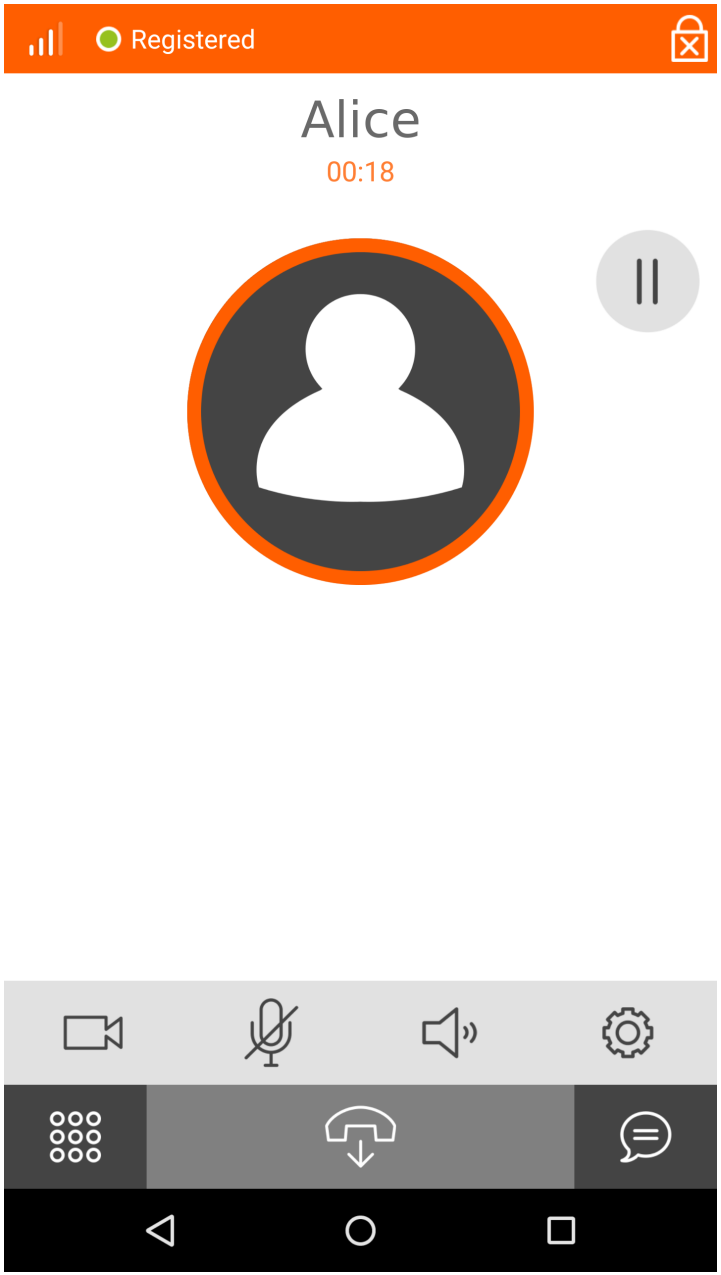
Jitsi


00:00:02 | Call
Alice (alice@example.com)
zrtp Connected
Jitsi
Incoming call received from: Alice

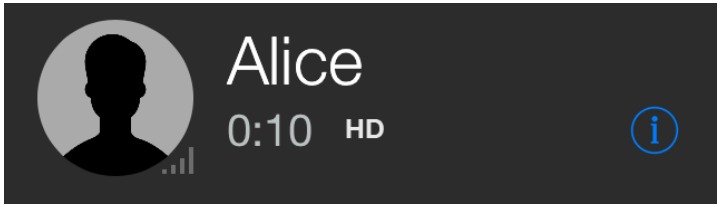

00:00:05 | Call
Alice (alice@example.com)
Connected

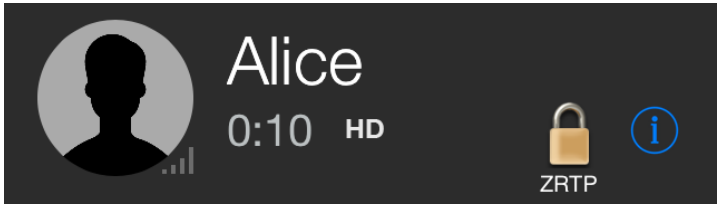
Linphone





Acrobits Softphone





Acrobits Softphone