Technische Universität Braunschweig

Aalto University

# Authenticated Resource Management in Delay-Tolerant Networks using Proxy Signatures

Dominik Schürmann, Jörg Ott, Lars Wolf

March 18, 2013

# Mobile Communication in Rural Areas of Africa

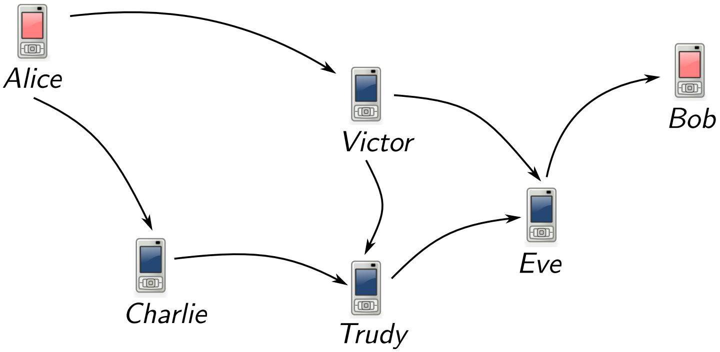
- Unreliable cell infrastructure (power outages)
- Relative high monthly costs: Nigerians living on $ 2 a day or less

⇒ Delay-Tolerant Networks (DTN) (RFC 5050)

---

*Mit Mobile Money gegen "finanzielle Apartheid"*. 2009. URL: http://www.zeit.de/digital/mobil/2009-11/m-money-africa; *Nigeria*. 2012. URL: http://topics.nytimes.com/top/news/international/countriesandterritories/nigeria/index.html

# Hop-by-hop Communication in DTNs




- Unknown meeting times
- Limited buffer space

# Example Attacks on Storage Buffers

## Denial-of-Service

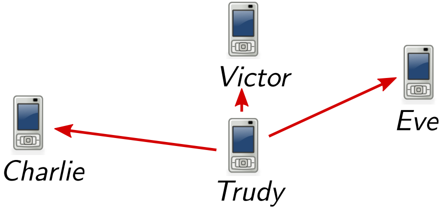
Flooding with big messages, differing in content, and forge source IDs.
Set lifetime of bundle very high.

## Multicast Amplification

Address bundle to multicast EID, set Report-to-EID to multicast EID

## More DTN-Specific Attacks...

"Amplification by Fragmentation", "Amplification by Custody Transfers",...



# Example Attacks on Storage Buffers

## Denial-of-Service

Flooding with big messages, differing in content, and forge source IDs.
Set lifetime of bundle very high.

## Multicast Amplification

Address bundle to multicast EID, set Report-to-EID to multicast EID

## More DTN-Specific Attacks...

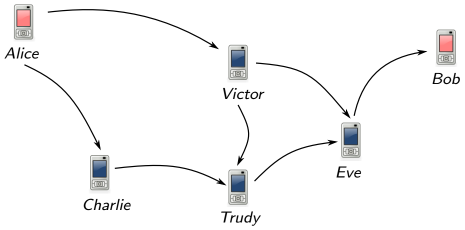
"Amplification by Fragmentation", "Amplification by Custody Transfers",...

**Properties of DTNs make attacks worse!**
**What to do against malicious nodes flooding the network?**

# Preemptive Buffer Management[1]

## Basic Idea

- Sign messages to authenticate their source ID
- Partition storage equally between IDs of incoming messages



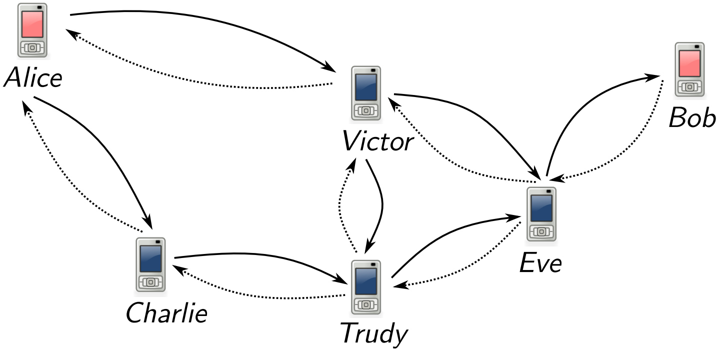

## Example: *Eve*'s Buffer (Max: 6 Messages)

Stores messages coming from Alice, Victor, and Bob

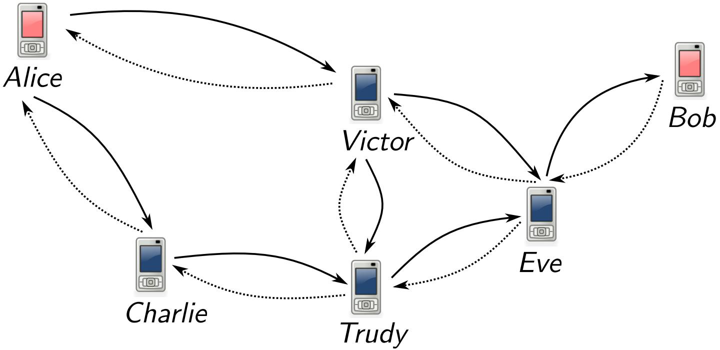
| | | |
|---|---|---|
| $M^1_{Alice}$ $M^2_{Alice}$ | $M^1_{Victor}$ | $M^1_{Bob}$ $M^2_{Bob}$ |

[1] John Solis et al. "Controlling resource hogs in mobile delay-tolerant networks". In: *Computer Communications* 33.1 (May 14, 2010), pp. 2–10.

# Request-Response Scenario



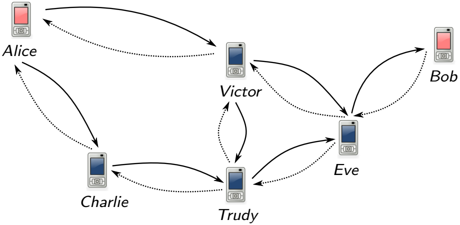
# Request-Response Scenario




## Example: *Eve*'s Buffer (Max: 6 Messages)

| | | | | |
|---|---|---|---|---|
| 1. Request: | $M^1_{Alice}$ $M^2_{Alice}$ | $M^1_{Victor}$ | $M^1_{Bob}$ $M^2_{Bob}$ | |
| 2. Response: | $M^1_{Alice}$ $M^2_{Alice}$ | $M^1_{Victor}$ | $M^2_{Bob}$ $R^1_{Bob}$ | drop $M^1_{Bob}$ |

# Improving Fairness?




## *Eve*'s Buffer: Original Scheme

2. Response:

| | | |
|---|---|---|
| $M^1_{Alice}$ $M^2_{Alice}$ | $M^1_{Victor}$ | $M^2_{Bob}$ $R^1_{Bob}$ |

drop $M^1_{Bob}$

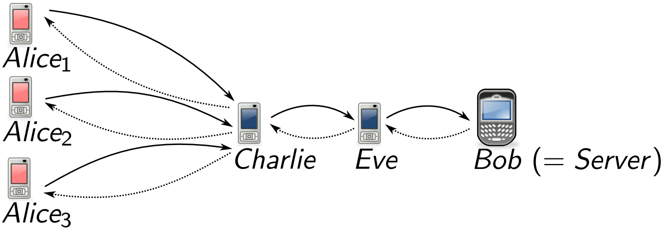
## Changed Affiliation of Response

2. Response:

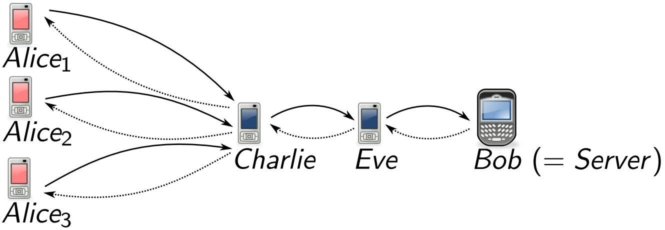
| | | |
|---|---|---|
| $M^2_{Alice}$ $R^1_{A(B)}$ | $M^1_{Victor}$ | $M^1_{Bob}$ $M^2_{Bob}$ |

drop $M^1_{Alice}$

# Scenario with Often Requested Server Node



# Scenario with Often Requested Server Node




## *Eve*'s Buffer: Original Scheme

2. Response:

| | | | |
|---|---|---|---|
| $M^1_{A_1}$ | $M^1_{A_2}$ | $M^1_{A_3}$ | $R^2_B$ $R^3_B$ |

drop $\rightarrow$ $R^1_B, M^1_B$

## Changed Affiliation of Responses

2. Response:

| | | | |
|---|---|---|---|
| $M^1_{A_1} R^1_{A_1(B)}$ | $M^1_{A_2} R^2_{A_2(B)}$ | $M^1_{A_3} R^3_{A_3(B)}$ | $M^1_B$ |

# Cryptographic Background

## Signing

- Every node $i$ has a public/private key pair $\langle pk_i, sk_i \rangle$ and an $ID_i$
- Every node on the path should be able to verify the signature
  $\rightarrow$ **Encrypt-then-Sign**
- Encryption when sending message to *Bob*: $c = Enc_{pk_B}(m)$
- Sign ciphertext by *Alice*: $\sigma = Sign_{sk_A}(c)$

Message to be send: $M = \langle c, \sigma \rangle$

## Verification

- Buffering incoming messages based on source ID
- Verify source ID by verifying signature: $Verify_{pk_A}(c, \sigma)$

# Proxy Signature: "Delegation-by-Certificate"[2]

$A$

warrant $\omega$ defines the input space of the proxy signing function

certificate $pcert = Sign_{sk_A}(00 \parallel ID_B \parallel pk_B \parallel \omega)$

ciphertext $c_1 = Enc_{pk_B}(m_1 \parallel \omega \parallel pcert)$

with $m_1$ as request content

$A$ → $B$: $request = \langle c_1, \sigma_1 = Sign_{sk_A}(11 \parallel c_1)\rangle$

(forwarded hop-by-hop)

$B$

*decrypt and verify signatures*

proxy signing key $psk_B = \langle sk_B, pk_A, (ID_B \parallel pk_B \parallel \omega), pcert\rangle$

ciphertext $c_2 = Enc_{pk_A}(m_2)$ with $m_2$ as response content

$PSign_{psk_B}(c_2) = \langle ID_B, \omega, pk_B, pcert,$

$\sigma_2 = Sign_{sk_B}(01 \parallel pk_A \parallel c_2)\rangle$

$B$ → $A$: $response = \langle c_2, \Sigma = PSign_{psk_B}(c_2)\rangle$

(forwarded hop-by-hop)

$A$

[2] Alexandra Boldyreva et al. "Secure Proxy Signature Schemes for Delegation of Signing Rights". In: *Journal of Cryptology* 25 (1 2012), pp. 57–115.

# Verification of Proxy Signatures

## Verification by Nodes Forwarding the Response

- Verify traditional signature
- Verify proxy signature by $PVerify_{pk_A,pk_B}(c_2, \Sigma)$

$$
\begin{aligned}
& PVerify_{pk_A,pk_B}(c_2, \Sigma) = \\
& \quad Verify_{pk_A}(00 \parallel ID_B \parallel pk_B \parallel \omega, pcert) \\
& \quad \wedge Verify_{pk_B}(01 \parallel pk_A \parallel c_2, \sigma_2) \wedge (c_2 \in \omega).
\end{aligned}
$$

# Application of Proxy Signatures

## *pcert* Restrictions

### Validity Restriction

Certificate is only valid for a specific time frame

### Limited Response

Responses are restricted to specific IDs by warrant $\omega$

## Message Pattern

- One-time request-response
- Publish-subscribe
- Two-way communication

# Simulation with “The ONE” Simulator

**Does our approach improve request/response success probability?**

# Simulation with “The ONE” Simulator

**Does our approach improve request/response success probability?**
**What happens in presence of malicious nodes?**

# Simulation with "The ONE" Simulator

**Does our approach improve request/response success probability?**
**What happens in presence of malicious nodes?**

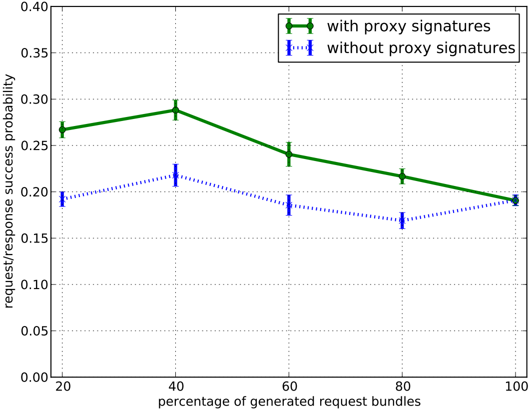
## Scenario with Server Nodes (With and Without Proxy Signatures)

- 95 % nodes with 5 MB storage
- 5 % are "server" nodes with 50 MB storage
- 3 message types: Request, response, unidirectional

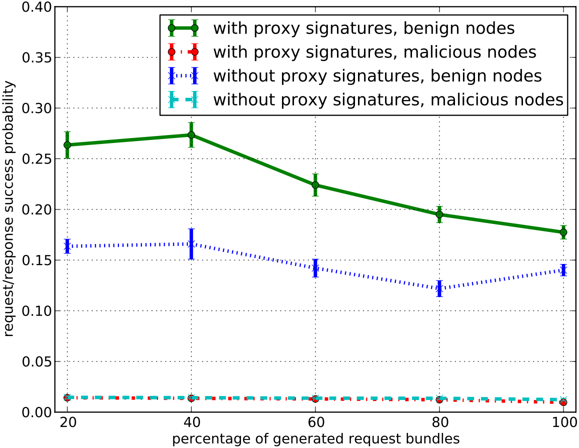
| **Parameter** | **Choice** |
|---|---|
| Movement Model | Shortest Path |
| Connectivity | Bluetooth-like |
| Routing Model | Spray-and-Wait |
| Map | Helsinki city's central area |

# Only Benign Nodes

- Struggle for buffer space between message types
- Request/response success probability as a metric



# 95 % Benign and 5 % Malicious Nodes



# Conclusion

## In proper scenarios, our approach improves. . .

- fairness by affiliating responses to **initiating** peer
- request/response success probability
- performance of mutual communications even in presence of attackers

## Properties

- Cryptographically secured extension to buffer management
- Delegation is done without central authority
- Delegation is delay-tolerant
- No further storage is needed for time based certificate restriction

# Conclusion

## In proper scenarios, our approach improves...

- fairness by affiliating responses to **initiating** peer
- request/response success probability
- performance of mutual communications even in presence of attackers

## Properties

- Cryptographically secured extension to buffer management
- Delegation is done without central authority
- Delegation is delay-tolerant
- No further storage is needed for time based certificate restriction

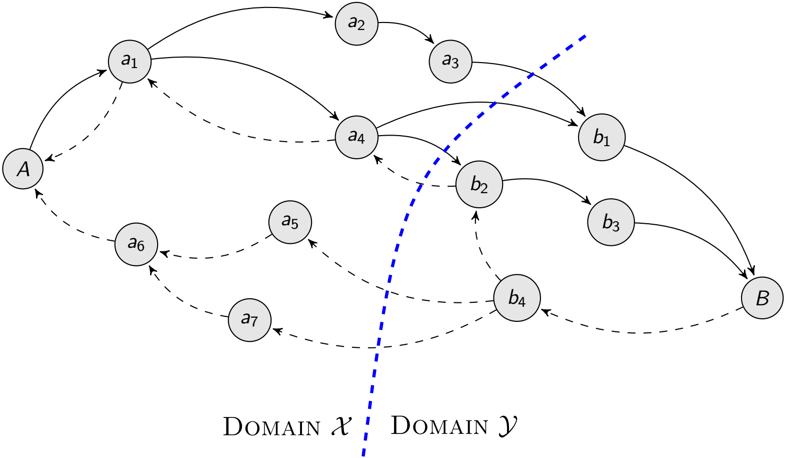
**Questions?**

Request from $A$ to $B$: $request = \langle c_1, \sigma = Sign_{sk_A}(11 \parallel c_1)\rangle$

Response from $B$ to $A$: $response = \langle c_2, \Sigma = PSign_{psk_B}(c_2)\rangle$

Storage buffer on $a_1$:

| | | | | | | |
|---|---|---|---|---|---|---|
| $A$ | $a_2$ | $a_3$ | $a_4$ | $a_6$ | $a_7$ | $\mathcal{Y}$ |



# Only Benign Nodes, 40 % Prob. to Generate Requests