Abuse-resistant pseudonymization and pseudonym change without trusted third parties

Martin Florian, Johannes Walter, Ingmar Baumgart
Context

- Cooperative services
  - Online communities
  - Vehicular networks
  - Crowdsensing
  - ...

- Privacy-relevant data is shared
  - Sensory data, locations, opinions…
  → privacy risk

- Pseudonyms hide identities of users and devices
  → help protect user privacy
Pseudonyms for cooperative services

- Central requirements
  - **Unlinkability** between users and their pseudonyms
    - Otherwise no privacy…
  - Possibility for unlinkable *pseudonym changes*
    - Reduces risk from accidental identification
    - Prevents profiling
  - **Sybil-resistance**
    - Only bounded number of pseudonyms per user
    - Required, e.g., for crowdsensing and voting
Pseudonyms for cooperative services

Basic architecture

- Initial pseudonym creation
- Pseudonym validation and use
- Pseudonym change
Solution #1: using a trusted issuer / CA

- Pseudonym = (blinded) certificate signed by CA

- Initial pseudonym creation
- CA checks user ID, issues pseudonym

- Pseudonym validation and use
- Standard crypto

- Pseudonym change
- Fancy crypto, pseudonym pools or via CA

Problem: who is that trusted third party?
- And how to pay for securing it?
Our alternative

- Use P2P networks providing decentralized consistency
- In the following: Bitcoin
- Completely decentralized – no trusted parties necessary
- Highly resilient
- Can potentially replace CA functions
BitNym

- Pseudonym = Bitcoin address (ECDSA public key)
  - Validation logic ensures that
    # active pseudonyms = # initial pseudonyms
    → prevents sybil attacks on pseudonym changing
  - Unlinkable changes enabled through P2P Bitcoin mixing
A little Bitcoin background

- **Transaction**

- **Multiple transactions + puzzle solution** → **block**

- **Blocks are chained** → **blockchain**

- Longest known chain is “truth” → decentralized consistency

Transactions:

```
<table>
<thead>
<tr>
<th>Transaction</th>
<th>Input (5 mBTC)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Input (9 mBTC)</td>
</tr>
<tr>
<td></td>
<td>Fee (0.1 mBTC)</td>
</tr>
<tr>
<td></td>
<td>Output (10 mBTC)</td>
</tr>
<tr>
<td></td>
<td>Output (3 mBTC)</td>
</tr>
</tbody>
</table>
```

```plaintext
Input (5 mBTC)
Input (9 mBTC)
Fee (0.1 mBTC)
Output (10 mBTC)
Output (3 mBTC)
```
Specially formed *Genesis Pseudonym Transaction* (GPTx)

- **Regular inputs**
- **A marker output**
  - Contains proof that access criteria have been met
  - Current implementation: access control via proof-of-burn (To register a pseudonym users need to „destroy“ Bitcoins)
- **A pseudonym output**
  - Realized as standard output (*pay-to-pubkey-hash*)
  - Encodes pseudonym public key
Pseudonym change

- In essence: by spending pseudonym outputs
- Unlinkability through the formation of mix transactions with other users
  - P2P mixing protocol based on *CoinShuffle* [Ruffing et al., 2014]
Pseudonym validation

Construction of unambiguous validation paths

- Validated via GPTx of Bob
- Might belong to Alice or Bob
Summary

- Unlinkable, changeable, sybil-resistant **pseudonyms**
- For **cooperative services** with requirements to **user privacy**
  - Online services, vehicular networking, IoT, …
- No trusted third parties
  → built upon **blockchain** networks like Bitcoin

Next steps
- Social-graph based initial access control
- Blacklisting via **blockchain**-based broadcast channels

Thank you!
BACKUP
Evaluation of achievable anonymity sets

- Anonymity level grows exponentially with each change
- Possible maximum is reached quickly

Graph showing the relationship between the number of changes and anonymity set size for different values of uc (500, 1000, 2000, 4000, 8000, 16000, 32000).