GI SIG SIDAR & SIG PET Workshop on Privacy Respecting Incident Management

Evaluating the Design of an Audit Data Pseudonymizer Using Basic Building Blocks for Anonymity

Ulrich Flegel

April 2005, Dortmund
Overview

- APES Basic Building Blocks for Anonymity
  - Overview APES Project
  - Motivation for Evaluation
  - Basic Building Blocks
- Example Anonymity System: Pseudo/CoRe
  - Motivation for Audit Data Pseudonymization
  - Overview Pseudo/CoRe
  - Specific Building Block Requirements
- Evaluation of Pseudo/CoRe
  - Decomposition
  - Building Blocks Used
  - Results
- Conclusion
Anonymity and Privacy in Electronic Services

* 

Basic Building Blocks for Anonymity
APES: Anonymity and Privacy in Electronic Services

- surveys state-of-the-art anonymity systems: anonymous connections, web browsing, e-mail, e-payments, e-auctions, . . .

- anonymity systems decomposed into reusable basic build blocks
  - easier to compare similar building blocks than complex anonymity systems
  - can systematically identify deficiencies given list of building blocks
  - can design anonymity systems by systematically composing building blocks

**here: evaluate design of a given anonymity system:**

- decompose into building blocks
- compare building blocks used to all similar building blocks to

  goal 1) identify room for improvement
  goal 2) identify deficiencies
The APES Basic Building Blocks Levels

- building blocks hide or remove identifying information at

  **connection level:** provide anonymous communication channels
  - information may identify individuals
    - **implicitly:** linking information along connection path by
      - **appearance:** content, format, size, ... 
      - **flow:** exploit knowledge about packet processing: order, timing, ... 
    - **explicitly (appearance):** IP address in packet header, ... 
  - compose building blocks to change appearance and flow 

  **application level:** provide anonymity in an application
  - mostly not *basic* building blocks, rather composed of elementary building blocks not offering anonymity alone 

- need to be combined on both levels to achieve anonymity
An Example Anonymity System
*

*Pseudo/CoRe*

Pseudonymization with *Conditional Re*identification
Audit Data Pseudonymization

**audit data:** (=log data)
- can be used to identify individual persons that use a service:
  - performance monitoring, activity profiling

**conflicting security requirements:**
- **accountability** of misuse to protect victims
- individual desire for and right on anonymity / privacy

**balancing conflicting security requirements:**
- replace person identifying features in audit data with **pseudonyms**
- detection of misuse suspicions possible on pseudonymized audit data
- for a given misuse suspicion **accountability** can be established:
  - only the involved **pseudonyms** can be disclosed
Pseudo/CoRe

technical purpose binding

organizational purpose binding
Specific Building Block Requirements

- SSO generally cannot observe user behavior, exception: inspection of pseudonymized audit data

⇒ no connection-level anonymity required

- channel between audit component and pseudonymizer must be protected, easiest if channel is short and local, hence pseudonymize on device providing service and generating audit data

⇒ service responsiveness must not degrade substantially  \(^{(a)}\)

- device may get successfully hacked, hence move audit data to a secure location as soon as possible

⇒ pseudonymization must:  \(^{(b)}\)
  - be performed on the fly
  - introduce no significant delay
  - keep up with audit data volume characteristic for the service

(a) & (b) ⇒ building blocks with low computational complexity and low delay
Evaluation of *Pseudo/CoRe*
Pseudonymization Approach Decomposed

- **Audit Data**
  - Filtering
  - Random String
    - Padding 1
      - Substitution
        - Encryption 1
          - Reordering
            - Dummy Updates
        - One-Way Function 1
          - Dummy Generation
            - Padding 2
          - Random Key
            - One-Way Function 2
            - Secret Sharing Schemes
              - Threshold Cryptosystem
            - Pseudonymity-Layer Data
  - Encryption 1
  - One-Way Function 1
  - Random Key
  - Secret Sharing Schemes
  - Threshold Cryptosystem

- **Pseudonymized Audit Data**
- **Pseudonym Mapping**
## Connection-Level Building Blocks Used

<table>
<thead>
<tr>
<th>building block</th>
<th>connection-level appearance</th>
<th>connection-level flow</th>
<th>application-level</th>
<th>our approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>encryption</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>padding</td>
<td>✓</td>
<td></td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>substitution</td>
<td>✓</td>
<td></td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>compression</td>
<td>✓</td>
<td></td>
<td></td>
<td>—</td>
</tr>
<tr>
<td>reordering</td>
<td>✓</td>
<td>?</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>latency</td>
<td>✓</td>
<td></td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>dummy activity</td>
<td>✓</td>
<td>?</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>no replay</td>
<td>✓</td>
<td></td>
<td></td>
<td>—</td>
</tr>
<tr>
<td>filtering</td>
<td>✓</td>
<td>?</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>caching</td>
<td>✓</td>
<td></td>
<td></td>
<td>—</td>
</tr>
<tr>
<td>broadcast</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>—</td>
</tr>
<tr>
<td>untraceable broadcast</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>—</td>
</tr>
<tr>
<td>multiplexing</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>—</td>
</tr>
<tr>
<td>bulletin board</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>—</td>
</tr>
</tbody>
</table>
### Application-Level Building Blocks Used

<table>
<thead>
<tr>
<th>building block</th>
<th>connection-level</th>
<th>application-level</th>
<th>our approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>appearance</td>
<td>flow</td>
<td></td>
</tr>
<tr>
<td>one-way function</td>
<td>—</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>(fair) blind signature</td>
<td>√</td>
<td>(?) / —</td>
<td></td>
</tr>
<tr>
<td>group signature</td>
<td>√</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>threshold cryptosystem</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>multi-party computation</td>
<td>√</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>homomorphic encryption</td>
<td>√</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>deniable encryption</td>
<td>√</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>secret sharing schemes</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>zero-knowledge</td>
<td>√</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>pseudonyms</td>
<td>√</td>
<td>? / √</td>
<td></td>
</tr>
<tr>
<td>trusted third party</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>
Evaluation Results

ad goal 1) identify room for improvement

- in the conceptual design under specific circumstances a more efficient building block could be used to hide pseudonym mapping updates
- six build blocks could be used to
  - reduce the power of the TTP
  - replace the threshold cryptosystem
  - provide exploitable properties in of protected pseudonymity layer data
- probably none of the candidate building blocks will either satisfy the specific requirements of audit data pseudonymization wrt. computational complexity or delay

⇒ improvement possible only if requirements are relaxed to trade off stronger mechanisms against computational complexity or delay

ad goal 2) identify deficiencies

- none found
Conclusions About the APES Approach

- it is **feasible** to decompose the design of a given anonymity system
- informally analyzing the decomposed design can identify weaknesses and/or room for improvement
- the given building blocks for conditional anonymity were sufficient for our design; may be sufficient to build many systems for conditional anonymity
- the classification of building blocks is **incomplete**
- the list of basic building blocks for anonymity is **not exhaustive**

⇒ analysis results merely give **strong indications** based on the **current state of knowledge**
Contact

Software

Site: http://ls6-www.cs.uni-dortmund.de/pseudocore
Support: pseudo-support@ls6.cs.uni-dortmund.de

Contact

Ulrich Flegel
WWW: http://ls6-www.cs.uni-dortmund.de/~flegel
Email: ulrich.flegel/at/udo.edu