Privacy for Distributed Databases via (Un)linkable & Auditable Pseudonyms

Anja Lehmann
IBM Research – Zurich

based on joint work with Jan Camenisch
How to maintain related yet distributed data?

- examples: social security system, ehealth
  - different entities maintain data of citizens
  - eventually data needs to be exchanged or correlated

<table>
<thead>
<tr>
<th>ID</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice.1210</td>
<td></td>
</tr>
<tr>
<td>Bob.0411</td>
<td></td>
</tr>
<tr>
<td>Carol.2503</td>
<td></td>
</tr>
</tbody>
</table>

- simple solution: data gets associated with globally unique identifiers (e.g., US, Belgium, Sweden, ...)
- unique identifiers are security & privacy risk
  - no control about data exchange & usage
  - if associated data is lost, all pieces can be linked together
  - linkability of data allows re-identification of “anonymized” data (e.g. Netflix challenge)
Local Pseudonyms & *Trusted* Converter

- user data is associated with (unlinkable) server-local identifiers aka “pseudonyms”
- only central converter can link & convert pseudonyms

+ control about data exchange
+ if records are lost, pieces cannot be linked together

<table>
<thead>
<tr>
<th>Main ID</th>
<th>Doctor A</th>
<th>Hospital</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice.1210</td>
<td>Hba02</td>
<td>7twnG</td>
</tr>
<tr>
<td>Bob.0411</td>
<td>P89dy</td>
<td>ML3m5</td>
</tr>
<tr>
<td>Carol.2503</td>
<td>912uj</td>
<td>sD7Ab</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hba02</td>
<td></td>
</tr>
<tr>
<td>P89dy</td>
<td></td>
</tr>
<tr>
<td>912uj</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML3m5</td>
<td></td>
</tr>
<tr>
<td>sD7Ab</td>
<td></td>
</tr>
<tr>
<td>y2B4m</td>
<td></td>
</tr>
</tbody>
</table>
Local Pseudonyms & *Trusted* Converter

- user data is associated with (unlinkable) server-local identifiers aka “pseudonyms”
- only central converter can link & convert pseudonyms

+ control about data exchange
+ if records are lost, pieces cannot be linked together
+ converter can provide audit logs to users (*GDPR*-requirement)
  - converter learns all request & knows all correlations
Our Work: Local Pseudonyms & Oblivious Converter

- user data is associated with (unlinkable) server-local identifiers aka “pseudonyms”
- only central converter can link & convert pseudonyms

+ control about data exchange
+ if records are lost, pieces cannot be linked together
+ converter can provide audit logs to users (GDPR-requirement)
  - converter learns all requests & knows all correlations
(Un)linkable Pseudonyms | Pseudonym Generation

- user, converter & server jointly derive pseudonyms from unique identifiers

- [CL15] generation triggered by converter, knows unique IDs
- [CL17] oblivious pseudonym generation triggered by user
### (Un)linkable Pseudonyms | Pseudonym Conversion

- only converter can link & convert pseudonyms, but does so in a blind way

<table>
<thead>
<tr>
<th>ID</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hba02</td>
<td>![Folder Icon]</td>
</tr>
<tr>
<td>P89dy</td>
<td>![Folder Icon]</td>
</tr>
<tr>
<td>912uj</td>
<td>![Folder Icon]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML3m5</td>
<td>![Folder Icon]</td>
</tr>
<tr>
<td>sD7Ab</td>
<td>![Folder Icon]</td>
</tr>
<tr>
<td>y2B4m</td>
<td>![Folder Icon]</td>
</tr>
</tbody>
</table>

**Converter**
- Record of **ML3m5** at Hospital
- Record of **P89dy** at Hospital

**Doctor A**
- Record of **P89dy** at Hospital
- Blind conversion request
- Unblinding conversion response

**Hospital**
- Record of **ML3m5**
- Record of **P89dy**
(Un)linkable Pseudonyms | Consistency

- pseudonym generation is deterministic & consistent with blind conversion
(Un)linkable Pseudonyms | Consistency

- pseudonym conversions are transitive, unlinkable data can be aggregated
[CL17] every pseudonym conversion triggers blind generation of audit log entry

Audit Bulletin Board

Converter

Doctor A

Hospital

ML3m5

P89dy

ID

Data

Hba02

P89dy

912uj

ID

Data

ML3m5

sD7Ab

y2B4m

Doctor A → Hospital. 02/26/2017

Unique ID

Bob.0411
servers and users can be fully corrupt
converter at most honest-but-curious (w/o audits even fully corrupt [CL15])
Our Protocol

- high-level idea of convertible pseudonyms
- adding (efficient) auditability
- security against active adversaries
During the generation of a pseudonym, the Core Idea involves a converter that blindly computes $\text{nym}_{i,A} \leftarrow \text{PRF}(k, \text{uid}_i)^{x_A}$. Here, $k$ represents a key for each server: $x_A, x_B, x_C, \ldots$.

1. The converter $\mathcal{X}$ and the user $U_i$ jointly compute $z_i \leftarrow \text{OPRF}(k, \text{uid}_i)$.
2. User $U_i$ encrypts $z_i$ for server $S_A$ as $C_{\text{nym}} \leftarrow \text{Enc}(pk_A, z_i)$.
3. The converter $\mathcal{X}$ blindly computes $C'_{\text{nym}} \leftarrow C_{\text{nym}}^{x_A}$.
4. Server $S_A$ decrypts the pseudonym $\text{nym}_{i,A} \leftarrow \text{Dec}(sk_A, C'_{\text{nym}})$. 
Core Idea

Generation: \( X \) blindly computes \( \text{nym}_{i,A} \leftarrow \text{PRF}(k,\text{uid}_i)^{x_A} \)

Conversion: \( X \) blindly computes \( \text{nym}_{i,B} \leftarrow \text{nym}_{i,A}^{x_B} / x_A \)

[1] \( S_A \) encrypts \( \text{nym}_{i,A} \) under \( S_B \)'s key

\[ C \leftarrow \text{Enc}(\text{pk}_B, \text{nym}_{i,A}) \]

[2] \( X \) blindly transforms encrypted pseudonym

\[ C' \leftarrow C^\Delta \quad \text{with} \quad \Delta = x_B / x_A \]

\[ C' = \text{Enc}(\text{pk}_B, \text{nym}_{i,A})^{x_B} / x_A \]

\[ = \text{Enc}(\text{pk}_B, \text{PRF}(k,\text{uid}_i)^{x_A})^{x_B} / x_A \]

\[ = \text{Enc}(\text{pk}_B, \text{PRF}(k,\text{uid}_i)^{x_B}) \]

\[ = \text{Enc}(\text{pk}_B, \text{nym}_{i,B}) \]

[3] \( S_B \) decrypts converted pseudonym

\[ \text{nym}_{i,B} \leftarrow \text{Dec}(\text{sk}_B, C') \]
High-level Idea | Overview

Generation
Conversion

Converter \( \chi \)

NymRequest

Converter \( \chi \)

ConvRequest

ConvResponse

Server A

nym_{i,A}

Server A

nym_{i,A}

Server B

nym_{i,B}
High-level Idea | Adding Auditability

decrypt all audit ciphertexts: $info \leftarrow \text{Dec}(usk,C^*)$?

`ConvRequest, upk''`

`ConvResponse, upk'''`

`Converter X`

`NymRequest, upk'`

`NymResponse, upk'`

`Server A`

`nym_{i,A}, upk'`

`Server B`

`nym_{i,B}, upk'''`

`Converter X`

`C^* \leftarrow \text{Enc}(upk', info)`

`Audit Bulletin Board`

`C^*`
High-level Idea | Adding Efficient Auditability (via Audit Tags)

```
High-level Idea | Adding Efficient Auditability (via Audit Tags)

decrypt ciphertext for $T_A$:
```

```
info ← Dec(usk, C*)
```

```
C_T ← Enc(pk_A, T_A) \ldots \text{for random } T_A
```

```
NymRequest, upk', C_T
```

```
Converter X
```

```
NymResponse, upk', C_T
```

```
Server A
```

```
nym_{i,A}, upk', T_A
```

```
T_A ← Dec(sk_A, C_T)
```

```
Generation Conversion
```

```
ConvertRequest, upk'', T_A
```

```
Server A
```

```
nym_{i,A}, upk', T_A
```

```
ConvResponse, upk'''
```

```
Server B
```

```
nym_{i,B}, upk'''
```

Audit Bulletin Board

```
T_A, C*
```

```
\ldots
```

```
C* ← Enc(upk'', info)
```

\[ 17 \]
High-level Idea | Adding Efficient Auditability (via Audit Tags)

- decrypt ciphertext for $T_A$: $\text{info} \leftarrow \text{Dec(usk, C*)}$
- get new audit tags for $T_A$: $T_B \leftarrow \text{Dec(usk, C*_{TB})}$

Converter $\mathcal{X}$

NymRequest, upk', $C_T$

NymResponse, upk', $C_T$

Server A

nym_{i,A}, upk', $T_A$

$T_A \leftarrow \text{Dec(sk}_A, C_T)$

Generation

Conversion

Audit Bulletin Board

$T_A$, C*; $T_B$, C**

Tag Chain:

$T_A$, C*_{TB}

Converter $\mathcal{X}$

ConvRequest, upk'', $T_A$

ConvResponse, upk'''

$C^*_{TB} \leftarrow \text{Enc(upk''', } T_B \text{) ... for random } T_B$

Server A

nym_{i,A}, upk', $T_A$

$nym_{i,B}, upk''', T_B$
High-level Idea | Adding Efficient Auditability (via Audit Tags)

decrypt ciphertext for $T_A$:
$\text{info} \leftarrow \text{Dec}(\text{usk}, C^*)$

get new audit tags for $T_A$:
$T_B \leftarrow \text{Dec}(\text{usk}, C^*_{TB})$
$T'_A \leftarrow \text{Dec}(\text{usk}, C^*_{TA})$

$T_A \leftarrow \text{Dec}(\text{sk}_A, C_T)$

Generation

Conversion

Audit Bulletin Board
$T_A$, $C^*$, $T_B$, $C^**$

Tag Chain:
$T_A$, $C^*_{TB}$
$T_A$, $C^*_{TA}$

Converter $\mathcal{X}$
$C^* \leftarrow \text{Enc}(\text{upk}'', \text{info})$

ConvRequest, upk'', $T_A$, $C^*_{TA}$

Server A

$nym_{i,A}$, upk', $T_A$

$nym_{i,B}$, upk''', $T_B$

Converter $\mathcal{X}$

ConvResponse, upk'''

$C^*_B \leftarrow \text{Enc}(\text{upk'''}, T_B)$ ... for random $T_B$

$C^*_{TA} \leftarrow \text{Enc}(\text{upk''', } T'_A)$ ... for random $T'_A$

$T'_A \leftarrow \text{Dec}(\text{usk}, C^*_TA)$

$T_A \leftarrow \text{Dec}(\text{sk}_A, C_T)$

$C_T \leftarrow \text{Enc}(\text{pk}_A, T_A)$ ... for random $T_A$

$\text{usk, upk}, \{T_A, T_B, T'_A, \ldots\}$

NymRequest, upk', $C_T$

NymResponse, upk', $C_T$

Server A

$nym_{i,A}$, upk', $T_A$
High-level Idea | Security against Active Adversaries

Audit Bulletin Board

Tag Chain:
- $T_A$, $C^*$, $T_B$, $C''$
- $T_A$, $C^*_{TB}$
- $T_A$, $C^*_{TA}$

Converter $\mathcal{X}$

$C_T \leftarrow \text{Enc}(pk_A, T_A)$ ... for random $T_A$

Get new audit tags for $T_A$:
- $T_A \leftarrow \text{Dec}(usk, C^*)$
- $T'_A \leftarrow \text{Dec}(usk, C^*_{TA})$

Decrypt ciphertext for $T_A$:
- $\text{info} \leftarrow \text{Dec}(usk, C^*)$

Signature scheme for homomorphic encodings

Converter $\mathcal{X}$

ConvRequest, $upk'$, $C_T$

Server A

NymRequest, $upk'$, $C_T$

NymResponse, $upk'$, $C_T$

Server B

ConvResponse, $upk''$

$C^*_{TB} \leftarrow \text{Enc}(upk'', T_B)$ ... for random $T_B$

$C^*_{TA} \leftarrow \text{Enc}(upk'', T'_A)$ ... for random $T'_A$

Audit Bulletin Board

$T_A$, $C^*$, $T_B$, $C''$

Tag Chain:
- $T_A$, $C^*_{TB}$
- $T_A$, $C^*_{TA}$

Converter $\mathcal{X}$

ConvRequest, $upk''$, $T_A$, $C^*_{TA}$, $\pi_A$

Server A

NymResponse, $upk'$, $C_T$

Server B

ConvResponse, $upk''$

$C^*_{TB} \leftarrow \text{Enc}(upk''', T_B)$ ... for random $T_B$
(Un)linkable & Auditable Pseudonyms | Summary

- provably secure in the Universal Composability (UC) framework
  - converter honest-but-curious, users & servers actively corrupt
- concrete instantiation ~50ms computational time per party for conversion
  - ElGamal-based encryption, Groth+ signature scheme for encoded messages, Dodis-Yampolskiy-based OPRF

Summary

- pseudonym scheme for (un)linkable data storage with controlled & auditable data exchange
- pseudonyms can only be linked via a central converter
- conversions & audit logs are done in a blind way → converter must not be a trusted entity

→ paradigm shift: unlinkability per default, linkability only when necessary
Thanks!

Questions?

(Un)linkable Pseudonyms for Governmental Databases. CCS15.
anj@zurich.ibm.com
(Un)linkable & Auditable Pseudonyms | Security & Efficiency

- provably secure construction based on
  - homomorphic encryption scheme (ElGamal encryption)
  - homomorphic encryption scheme with re-randomizable public keys (ElGamal-based)
  - oblivious pseudorandom function (based on Dodis-Yampolskiy-PRF)
  - signature scheme for homomorphic encoding functions (Groth+)
  - zero-knowledge proofs (Fiat-Shamir NIZKs)
  - commitment scheme (ElGamal based)