Privacy-Preserving Personal Information Management

Mohamed Layouni

PhD Oral Defense
School of Computer Science, McGill University
Designing protocols that are:

- Secure
- Privacy-preserving
- User-centric
Main Contributions of this Thesis (1/2)

- Studied/Surveyed **Privacy-Preserving Credentials**
  - Compared the most complete/elaborate ones
  - Proposed an *extension* to the Camenisch-Lysyanskaya credential system*

- Proposed two privacy-preserving protocols for **controlling access to remotely-stored DB records**, where **access** is performed according to policies defined by the owners of those records.
Main Contributions of this Thesis (2/2)

- Proposed protocols to solve real-world problems using privacy-preserving credentials:
  - **Prescription-handling** for the Belgian Healthcare System* (e.g., protecting patients’ privacy from administrative entities involved in the processing of insurance claims)
  - **Tele-monitoring** of patients’ health outside Hospital (Protocol for collecting patients’ health measurements in a user-centric and privacy-preserving way)
1. Introduction

2. Accredited Symmetrically Private Information Retrieval (ASPIR)

3. Multi-Authorizer ASPIR

4. Conclusion
Figure: Setting of the ASPIR Protocol
Privacy for Receiver: DB Server *should not be able to compute the index* of the retrieved record (and hence the ID of data-subject)

Privacy for DB Server: For each query, the Receiver can compute *information only on one record* (defined in the query), and nothing about the other records in DB.

Privacy for Data Subject:
- DB records *cannot be retrieved without authorization*
- It should be *intractable* for a quorum of players to *forge an authorization* for a record that none of them owns.
- DB Server should be able to *verify the validity of an authorization* presented by the Receiver, *without learning the identity* of the Data-Subject who issued it.
Solution combines **two main building blocks**:

- Privacy-Preserving Credential System (Brands’00)
- Symmetrically Private Information Retrieval System (Lipmaa’05)
Solution combines **two main building blocks:**

- Privacy-Preserving Credential System (Brands’00)
- Symmetrically Private Information Retrieval System (Lipmaa’05)
Symmetrically Private Information Retrieval (SPIR)

Figure: A Simple Database Query
Symmetrically Private Information Retrieval (SPIR)

Figure: Symmetrically Private Information Retrieval
Solution combines **two main building blocks**:

- Privacy-Preserving Credential System (Brands’00)
- Symmetrically Private Information Retrieval System (Lipmaa’05)
  - Similar to an **Oblivious Transfer** scheme,
  - Higher efficiency, *but*
  - Weaker security.
Solution combines **two main building blocks**:

- Privacy-Preserving Credential System (Brands’00)

- Symmetrically Private Information Retrieval System (Lipmaa’05)
  - Similar to an **Oblivious Transfer** scheme,
  - Higher efficiency, *but*
  - Weaker security.
Privacy-Preserving Credentials

**Figure:** Privacy-Preserving Credentials Issuing, Showing, and Depositing
Building Blocks

Privacy-Preserving Credentials

Properties of Privacy-Preserving Credentials

- Selective disclosure (in the sense of Zero Knowledge)
- Soundness (no false claims)
- Untraceability (showings unlinkable to user’s identity)
- Unlinkability (between showings)
- ...

Constructions from the Literature

- Camenisch and Lysyanskaya (IBM’s IDEMIX)
- Brands (Microsoft’s U-Prove)
Solution Overview

**Figure:** Accredited SPIR Protocol: High-Level Overview

Data Subject i

Receiver

Database Server

Q := Query(i, Rec−Public−Key)

Auth = SPK{ (i, j) : Cred.ID = j ^
Inv(Q) = i ^ i = j } (RecID, Policy...)

Q + Auth

Q + Auth + RecID + Policy

Check Auth, RecID,
if Policy is satisfied
SPIR−Process Q

Response R

DB[i] := Recover(Rec−Secret−Key, R)
Multi-Authorizer ASPIR is:

1. A new approach to constructing ASPIR schemes (also useful for single-Authorizer ASPIR)

2. An extension of ASPIR to a setting where:
   - A DB record belongs to multiple owners simultaneously
   - Receiver can recover a DB record only if he:
     - Complies with privacy policy defined by record owners.
     - Has authorizations from:
       - All owners of target record,
       - Any subset of owners of size larger than a threshold,
       - Certain subsets of owners (general access structure)
Figure: Setting of the Multi-Authorizer ASPIR Protocol
Requirements

- **Privacy for Receiver:** DB Server *cannot compute the index* of the retrieved record (and hence the IDs of its owners)

- **Privacy for DB Server:** For each query, the Receiver *learns information only on one record* (defined in the query), and nothing about the other records in DB.

- **Privacy for Data Subject:**
  - DB records cannot be recovered without the *necessary authorizations*
  - It should be *intractable* for a quorum of players to *forge an authorization* for a record that none of them owns.
Multi-Authorizer ASPIR is a completely new construction:

- We use different building blocks:

- We use SPIR schemes in a black-box fashion;
  Construction works with any SPIR scheme, not only Lipmaa’s SPIR scheme as in ASPIR.

- The new scheme is more efficient than previous ASPIR.
Auth_i = F_i(s, RecID, Policy)

Auth_{1,2,3} = \bigcup_i Auth_i

s = index(ID_{1,2,3})

Q = Query_{SPIR}(s)

DKey = F(Auth_{1,2,3}, R)

DB[ID_{1,2,3}] = Recover(DKey, R)

Figure: Multi-Authorizer ASPIR Protocol (Basic Construction)
The proposed protocols have the following extra functionalities:

- Receiver can retrieve *multiple records* belonging to a tuple of data-subjects (2 Constructions)

**Idea 1:** Change the way the SPIR query is processed (Technique similar to the one used in the General and Threshold Access Structure variants)

**Idea 2:** Two Databases: one for Keys, one for Ciphertexts. Retrieve key with MASPIR, and use it to decrypt all records of owners’ tuple being considered.
Summary:

1. Proposed two privacy-preserving protocols for controlling access to remotely-stored DB records, where access is performed according to policies defined by the owners of those records.

2. Proposed Privacy-Preserving eHealth protocols (e.g., Prescription-handling for the Belgian Healthcare System)

3. Surveyed the State of the Art in Privacy-Preserving Credential Systems, and provided a Comparison of the most elaborate/complete ones.
Possible Extensions:
Accredited Privately-Searchable Encryption

Same setting as ASPIR, except that:

- Data records are stored in encrypted form, with each record labelled by a set of keywords (also encrypted),
- Querying by keywords instead of by indices,
- Data-subjects control who can search their records, what keywords can be queried, terms & conditions.

The solution should be such that:

- Receiver can only retrieve records matching the authorized search keywords,
- DB Manager does not learn: ID of data-subject, search keywords, access pattern, or search results.
Thank you!
Accredited SPIR Protocol – Detailed Description

Public Info
\[ p, q, (g_i)_{0 \leq i \leq \ell}, h_0, (g_i^\sigma)_{0 \leq i \leq \ell}, h_0^\sigma, H, k, pk^{(R)}, \mathcal{R}, g_{db}, \]
\[ pk_{ElG}^{(R)} := (g_{ElG}, y_{ElG}), G_q := \langle g_i \rangle := \langle g_{db} \rangle, n := |DB| \leq q, \lambda_1, \cdots, \lambda_\alpha. \]

Authorizer
\[(c_1, c_2) := E_{pk_{ElG}}^{|DB|}((g_{db})^{ID_A}) \]

Receiver
\[ h, \sigma_{CA}(h) := (z', r'_0, c'_0, (c_1, c_2)) \]

Authorization
\[ \text{Authorization} \]
\[ \text{For } j := 1 \text{ to } \alpha \text{ do :} \]
\[ \text{For } t := 0 \text{ to } \lambda_j - 1 \text{ do :} \]
\[ r_{jt} \in R \mathcal{R} \]
\[ \beta_{jt} := \text{HomEnc}_{pk^{(R)}}(b_{jt}, r_{jt}), \]
\[ \text{where } b_{jt} := 1 \text{ if } t = ID_A^j, \]
\[ \text{and } b_{jt} := 0 \text{ otherwise.} \]

Sender (Database DB)
\[ \text{Check Authorization validity.} \]
\[ \text{For } j := 1 \text{ to } n \text{ do :} \]
\[ \delta_j \in R [1, q - 1] \]
\[ DB_0[j] := ((E_{pk_{ElG}}(g_{db}^{ID_A}) \otimes g_{db}^{-j})^{\delta_j} \otimes DB[j]) \]
\[ \text{For } j := 1 \text{ to } \alpha - 1 \text{ do :} \]
\[ \text{For } i_{j+1} := 0 \text{ to } \lambda_{j+1} - 1, \cdots, \]
\[ i_\alpha := 0 \text{ to } \lambda_\alpha - 1 \text{ do :} \]
\[ DB_j(i_{j+1}, \cdots, i_\alpha) := \prod_{t \in \mathbb{Z}_{\lambda_t}} (\beta_{jt})^{DB_{j-1}(t,i_{j+1}, \cdots, i_\alpha)} \]
\[ DB_{\alpha} := \prod_{t \in \mathbb{Z}_{\lambda_t}} (\beta_{\alpha t})^{DB_{(\alpha-1)}(t)} \]
\[ DB_\alpha := DB_\alpha \]
\[ \text{Output } DB[|ID_A|] := D_{sk_{ElG}}(|DB_0|) \]

Figure: Accredited SPIR Protocol (DLog-Based Construction)
Accredited SPIR Protocol – Detailed Description

**Public Info**

\[ p, q, (g_i)_{0 \leq i \leq \ell}, h_0, (g_i^{x_0})_{0 \leq i \leq \ell}, h_0^{x_0}, H, k, pk^{(R)}, R, g_{db}, \]

\[ pk^{(R)} := (g_{EIG}, y_{EIG}), G_q := \langle g_i \rangle := \langle g_{EIG} \rangle := \langle g_{db} \rangle, n := |DB| \leq q, \lambda_1, \ldots, \lambda_\alpha. \]

**Authorizer**

\[
(c_1, c_2) := E_{pk^{(R)}_{EIG}}((g_{db})^{ID,A})
\]

**Receiver**

\[
h, \sigma_{CA}(h) := (z', r'_0, c'_0), (c_1, c_2) \]

\[
\{ \epsilon_1, \ldots, \epsilon_\ell, \mu, \nu : h = g_1^{\epsilon_1} \cdots g_\ell^{\epsilon_\ell} h_0 \\
\land c_2 = y_\mu^{\epsilon_\ell} g_{db}^{\nu} \land \epsilon_1 = \nu \} (m)
\]

**Figure:** Accredited SPIR Protocol – Detailed description – Part I
**Accredited SPIR Protocol – Detailed Description**

<table>
<thead>
<tr>
<th>Receiver</th>
<th>Sender (Database DB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>For $j := 1$ to $\alpha$ do :</td>
<td>Authorization, ${\beta_{jt}}_{0 \leq t &lt; \lambda_j}$</td>
</tr>
<tr>
<td>For $t := 0$ to $\lambda_j - 1$ do :</td>
<td>Check Authorization validity.</td>
</tr>
<tr>
<td>$r_{jt} \in_R R$</td>
<td>For $j := 1$ to $n$ do :</td>
</tr>
<tr>
<td>$\beta_{jt} := \text{HomEnc}<em>{pk_R}(b</em>{jt}, r_{jt}),$</td>
<td>$\delta_j \in_R [1, q - 1]$</td>
</tr>
<tr>
<td>where $b_{jt} := 1$ if $t = \text{ID}_A(j),$</td>
<td>$\text{DB}_0[j] :=$</td>
</tr>
<tr>
<td>and $b_{jt} := 0$ otherwise.</td>
<td>$((E_{pk_{\text{ElG}}}^R(g_{db}^{\text{ID}<em>A}) \otimes g</em>{db}^{-j})^{\delta_j} \otimes \text{DB}[j])$</td>
</tr>
</tbody>
</table>

... 

**Figure:** Accredited SPIR Protocol – Detailed description – Part II
**Accredited SPIR Protocol – Detailed Description**

<table>
<thead>
<tr>
<th><strong>Receiver</strong></th>
<th><strong>Sender (Database DB)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Check Authorization validity.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>For</strong> $j := 1$ <strong>to</strong> $n$ <strong>do</strong> :</td>
<td></td>
</tr>
<tr>
<td>$\delta_j \in_R [1, q - 1]$</td>
<td></td>
</tr>
<tr>
<td>$\text{DB}_0[j] :=$</td>
<td></td>
</tr>
<tr>
<td>$((E_{pk}^{(R)}(g^{</td>
<td>ID</td>
</tr>
<tr>
<td><strong>For</strong> $j := 1$ <strong>to</strong> $\alpha - 1$ <strong>do</strong> :</td>
<td></td>
</tr>
<tr>
<td><strong>For</strong> $i_{j+1} := 0$ <strong>to</strong> $\lambda_{j+1} - 1$, $\cdots$, $i_{\alpha} := 0$ <strong>to</strong> $\lambda_{\alpha} - 1$ <strong>do</strong> :</td>
<td></td>
</tr>
<tr>
<td>$\text{DB}<em>j(i</em>{j+1}, \cdots, i_{\alpha}) :=$</td>
<td></td>
</tr>
<tr>
<td>$\prod_{t \in \Bbb{Z}<em>{\lambda_j}} (\beta</em>{jt})^{\text{DB}<em>{j-1}(t, i</em>{j+1}, \cdots, i_{\alpha})}$</td>
<td></td>
</tr>
<tr>
<td>$\text{DB}<em>\alpha := \prod</em>{t \in \Bbb{Z}<em>{\lambda</em>{\alpha}}} (\beta_{\alpha t})^{\text{DB}(\alpha - 1)(t)}$</td>
<td></td>
</tr>
</tbody>
</table>

$\text{DB}_\alpha' := \text{DB}_\alpha$  

$\text{DB}_\alpha := \frac{\text{DB}_\alpha'}{\prod_{t \in \Bbb{Z}_{\lambda_{\alpha}}} (\beta_{\alpha t})^{\text{DB}(\alpha - 1)(t)}}$

**For** $j := \alpha$ **downto** 1 **do** :  

$\text{DB}_{j-1}' := \text{HomDec}_{sk_{Eig}^{(R)}}(\text{DB}_j')$

**Output** $\text{DB}[ID|_A] := D_{sk_{Eig}^{(R)}}(\text{DB}_0')$

**Figure:** Accredited SPIR Protocol – Detailed description – Part III
Multi-Authorizer ASPIR Protocol – Detailed Overview

**Receiver (RecID)**

\((P_m, \sigma_u(P_m)), u \in \{A, B, C\}, \) for \(P_m := H(s, \text{RecID}, \mathcal{P})\),
where \(s := \text{index}(ID_A, ID_B, ID_C)\), and \(\mathcal{P} := \{\text{usage policy}\}\)

\[
\{pk_u\}_{u \in \{A, B, C\}} \triangleq \{pk_s, i\}_{1 \leq i \leq 3},
\]

\(e(\cdot, \cdot), P, G_1 = \langle P \rangle, G_2, q, \) SPIR scheme

\[
\text{Sig}(P_m) = \prod_{u \in \{A, B, C\}} \sigma_u(P_m) = \prod_{u \in \{A, B, C\}} (P_m)^{x_u} = (P_m)^{\sum_u x_u}
\]

\(Q = Q_{\text{SPIR}}(s)\)

**Public Info**

**Sender (Database DB)**

\[\text{If } \mathcal{P} \text{ satisfied continue}\]
\[\text{else abort}\]
\[\text{Choose } \delta \in_R \mathbb{Z}_q^*\]
\[\text{For } j := 1 \text{ to } N \text{ do :}\]
\[P_{mj} = H(j, \text{RecID}, \mathcal{P})\]
\[DB'[j] = DB[j] \times e \left( P_{mj}, \prod_{u=1}^{3} pk_{j,u} \right)^{\delta}\]

\[\text{Execute SPIR scheme on DB}' \text{ and } Q\]

\[\text{Let } Res = R_{\text{SPIR}}(Q, DB')\]

**SPIR-recover DB'[s] from Res**

**Output DB_0[s] := DB'[s] / e(Sig(P_m), P^\delta)\]

**Figure:** Multi-Authorizer ASPIR (Basic Construction)