Size Hiding in Private Set Intersection: Existential Results and Constructions

Paolo D’Arco (U. Salerno), María Isabel González Vasco (URJC), Angel L. Pérez del Pozo (URJC) and Claudio Soriente (ETH Zürich)
INTRODUCTION
Our setting

- PSI protocols are dedicated constructions for a special case of the general problem known as Secure Function Evaluation [M. Ben-Or, S. Goldwasser, A. Wigderson, STOC 1988].

\[
\begin{array}{c}
U_1 \\
X_1 \\
\downarrow \\
\end{array} \quad \ldots \quad \begin{array}{c}
U_i \\
X_i \\
\downarrow \\
\end{array} \quad \ldots \quad \begin{array}{c}
U_n \\
X_n \\
\downarrow \\
\end{array}
\]

SFE PROTOCOL
Our setting

- PSI protocols are dedicated constructions for a special case of the general problem known as **Secure Function Evaluation** [M.Ben-Or, S. Goldwasser, A. Wigderson, STOC 1988].

\[ y = f(X_1, \ldots, X_n) \]
The PSI Problem: motivation

Situation: two mutually distrusting entities hold non-disjoint sets of private data and want to take some actions on the intersection.

European banks
I do not trust

European banks
I do not trust
The PSI Problem: motivation

- Two national law enforcement bodies (FBI and MI5) want to compare their respective databases of terrorist suspects.
- A tax authority wants to learn whether any suspected tax evaders have accounts with a certain foreign bank.
- An on-line dating system allows you to search for people with shared hobbies.
A PSI Protocol: formulation

- Players: Client (C) and Server (S), each holding a private set $S = \{s_1, \ldots, s_w\}$ and $C = \{c_1, \ldots, c_v\}$ from a ground set $U$. We consider them to be HBC (honest-but-curious).

- Algorithms:
  - Setup: selects all global parameters
  - Interaction: protocol played by C and S, on (private) input their respective sets $S$ and $C$, provides private output to both parties.
A PSI Protocol: formulation

- Requirements (informal — one-sided):
  - **Correctness**: once Interaction is executed, with overwhelming probability, C gets as private output the intersection $C \cap S$ and (maybe) $|S|$ (w).
  - **Server Privacy**: the execution of Interaction leaks no information about $C \cap S^c$ (apart from its size)
  - **Client Privacy**: no information about $C$, apart from its size $v$, is leaked from Interaction.
Size-Hiding in PSI

- For some applications it can be important to hide the cardinality of $C$ and $S$.
- Most existing PSI protocols reveal the size of $C$ and $S$.
- In PKC 2011, Ateniese, De Cristofaro and Tsudik consider the problem and provide an RSA-based construction, secure in the ROM, which hides only the size of $C$.
- It was unclear if hiding the size of both $C$ and $S$ was even possible.
- **This work**: we further explore what can and cannot be done wrt Size-Hiding in PSI (one or two sided).
Security Models

- **Unconditional setting:**
  no computational assumption.

- **Cryptographic setting:**
  we add computational assumptions, consider pptm participants, and may also:
  - Rely on hardness assumptions related to number theoretical problems
  - Assume the existence of ideal hash functions (random oracles)

- **Further relaxations (of both models)**
  - Assume the presence of a Trusted Third Party (present in a Setup phase but not available in the Interaction)
UNCONDITIONAL SETTING
Unconditional setting

- PSI is impossible in the plain model, as it is equivalent to Oblivious Transfer
- Two side size hiding PSI, with a TTP, is possible in an unconditional setting
PSI versus OT

- An OT protocol involves two participants, Sender and Chooser, so that:
  - Sender’s input consists of two bits $b_0$ and $b_1$
  - Chooser’s input consists of a bit $\sigma$
  - At the end of the protocol the Chooser learns $b_\sigma$, and nothing else, while the Sender learns nothing.
PSI versus OT

- An OT protocol involves two participants, Sender and Chooser, so that:
  - Sender’s input consists of two bits $b_0$ and $b_1$
  - Chooser’s input consists of a bit $\sigma$
  - At the end of the protocol the Chooser learns $b_\sigma$, and nothing else, while the Sender learns nothing.
- Unconditionally secure OT is impossible
PSI versus OT

- An OT protocol involves two participants, Sender and Chooser, so that:
  - Sender’s input consists of two bits $b_0$ and $b_1$
  - Chooser’s input consists of a bit $\sigma$
  - At the end of the protocol the Chooser learns $b_\sigma$, and nothing else, while the Sender learns nothing.

- Unconditionally secure OT is impossible

- PSI implies OT:
  - Sender generates a set as $\{0|b_0, 1|b_1\}$
  - Chooser generates a set as $\{\sigma|0, \sigma|1\}$
  - After running a PSI protocol, Chooser acting as Client gets $\{\sigma \mid b_\sigma\}$
PSI versus OT

- An OT protocol involves two participants, Sender and Chooser, so that:
  - Sender’s input consists of two bits $b_0$ and $b_1$
  - Chooser’s input consists of a bit $\sigma$
  - At the end of the protocol the Chooser learns $b_{\sigma}$, and nothing else, while the Sender learns nothing.
- Unconditionally secure OT is impossible
- PSI implies OT:
  - Sender generates a set as $\{0|b_0, 1|b_1\}$
  - Chooser generates a set as $\{\sigma|0, \sigma|1\}$
  - After running a PSI protocol, Chooser acting as Client gets $\{\sigma | b_{\sigma}\}$

Unconditionally secure PSI is impossible
Unconditional setting

- PSI is impossible in the plain model, as it is equivalent to Oblivious Transfer.
- Two side size hiding PSI, with a TTP, is possible in an unconditional setting.
Unconditionally Secure SH-PSI with a TTP

Setup: the TTP chooses two random bijections $f, g: \mathcal{U} \mapsto \{0,1\}^{|U|}$
Unconditionally Secure SH-PSI with a TTP

Setup: the TTP chooses two random bijections \( f, g : \wp(U) \rightarrow \{0,1\}^{|U|} \)

\[ C = \{c_1, \ldots, c_v\} \]

\[ R, L \]

\[ R = f(C'), L = \{(g(D), D) : D \subseteq C'\} \]
Unconditionally Secure SH-PSI with a TTP

Setup: the TTP chooses two random bijections \( f, g: \mathcal{U} \rightarrow \{0, 1\}^{\mid \mathcal{U} \mid} \)

\[ C = \{c_1, \ldots, c_v\} \]

\[ R = f(C'), L = \{(g(D), D) : D \subseteq C\} \]

\[ T = \{(f(E), g(E \cap S)) : E \subseteq U\} \]

\[ S = \{s_1, \ldots, s_w\} \]
Unconditionally Secure SH-PSI with a TTP

Setup: the TTP chooses two random bijections \( f, g: \mathcal{U} \rightarrow \{0,1\}^{|U|} \)

- \( C = \{c_1, \ldots, c_v\} \)
- \( R, L \)
- \( R = f(C'), L = \{(g(D), D) : D \subseteq C\} \)
- \( S = \{s_1, \ldots, s_w\} \)
- \( T = \{(f(E), g(E \cap S)) : E \subseteq U\} \)

Interaction:

Search for \((R', D)\) in \(L\)
Output \(D\)

Search for \((R, *)\) in \(T\), define \(R'\) as \(*\)
We get:

- **Correctness**: once Interaction is executed, with overwhelming probability, C gets as private output the intersection $C \cap S$

- **Server Privacy**: the execution of Interaction leaks no information about $C \cap S^c$ - as $g$ is a random bijection, $R'$ leaks nothing

- **Client Privacy**: no information about $C$, apart from its size $v$, is leaked from Interaction - as $f$ is a random bijection, $R$ leaks nothing

- Both are $v$ and $w$ are hidden.
CRYPTOGRAPHIC SETTING
Security Models

- **Unconditional setting:**
  no computational assumption.

- **Cryptographic setting:**
  we add computational assumptions, consider pptm participants, and may also:
  - Rely on hardness assumptions related to number theoretical problems
  - Assume the existence of ideal hash functions (random oracles)

- **Further relaxations (of both models)**
  - Assume the presence of a Trusted Third Party
Technique I: AND Protocol (two side SH)
AND protocol

- Let \( U = \{u_1, \ldots, u_n\} \) be the ground set.
- C encodes his set as the characteristic vector of its set, i.e. a binary vector \( I_C \) of length \( n \) where
  \[ I_C[j] = 1 \text{ iff } u_j \text{ is in } C \]
- S does the same.
- For \( j = 1, \ldots, n \):
  - C and S run a secure evaluation protocol for the function \( \text{AND}(I_C[j], I_S[j]) \).
  - If the result is 1, C learns that \( u_j \) is in the intersection.
AND Protocol

- We get:
  - **Correctness**: once Interaction is executed, with overwhelming probability, C gets as private output the intersection $C \cap S \checkmark$
  - **Server Privacy**: the execution of Interaction leaks no information about $C \cap S^c \checkmark$
  - **Client Privacy**: no information about $C$, apart from its size $v$, is leaked from Interaction \( v \checkmark\)
- Both are $v$ and $w$ are hidden.
Technique II: Oblivious Polynomial Evaluation (two side SH*)
A seminal work: Freedman et al [Eurocrypt 2004]

Setup: run by C, selects an encoding of $U$ as a subset of $\mathbb{Z}_n^*$, runs a key generation algorithm for Paillier encryption which outputs $(s_k, p_k)$.
A seminal work: Freedman et al [Eurocrypt 2004]

Setup: run by C, selects an encoding of $U$ as a subset of $\mathbb{Z}_n^*$, runs a key generation algorithm for Paillier encryption which outputs $(s_k, p_k)$

Interaction:

$C = \{c_1, \ldots, c_v\}$, $s_k$

$S = \{s_1, \ldots, s_w\}$, $\pi \in_R S_w$

$P(x) = \prod_1^v (x - c_i) = \sum_0^k \alpha_i t^i \quad \{\text{Enc}(\alpha_i)\}_{i=0}^k$
A seminal work: Freedman et al [Eurocrypt 2004]

Setup: run by C, selects an encoding of $U$ as a subset of $\mathbb{Z}_n^*$, runs a key generation algorithm for Paillier encryption which outputs $(s_k, p_k)$

$p_k$, Encoding

Interaction:

$C = \{c_1, \ldots, c_v\}$, $s_k$

$P(x) = \prod_1^v (x - c_i) = \sum_0^k \alpha_i t^i$

$\{\text{Enc}(\alpha_i)\}_{i=0}^k$

$S = \{s_1, \ldots, s_w\}$, $\pi \in_{R} S_w$

for i=1 to w

$r_i \in_{R} \mathbb{Z}_n^*$

$e_i = \text{Enc}(r_i P(s_i) + s_i)$

$\{\pi(e_i)\}_{i=1}^w$
A seminal work: Freedman et al [Eurocrypt 2004]

Setup: run by C, selects an encoding of $U$ as a subset of $Z_{n^*}$, runs a key generation algorithm for Paillier encryption which outputs $(s_k, p_k)$

$P(x) = \prod_{1}^{v} (x- c_i) = \sum_{0}^{k} \alpha_i t^i \quad \{ \text{Enc}(\alpha_i) \}_{i=0}^{k}$

$C \cap S := \{c_1, \ldots, c_v\} \cap \{ \text{Dec}(e_i) \}_{i=1}^{w}$

for $i=1$ to $w$

$r_i \in_R Z_{n^*}$

$e_i = \text{Enc}(r_i P(s_i) + s_i)$

$S = \{s_1, \ldots, s_w\}$

$\pi \in_R S_w$
Extension

- Size Hiding: two-side, if an upper bound M on the sizes of both client and server’s set is known (only of interest if $M<<|U|$)
"Bounded" - Freedman et al

Setup: run by Client, selects an encoding of $U$ as a subset of $\mathbb{Z}_n^*$, runs a key generation algorithm for Paillier encryption which outputs $(s_k, p_k)$

$p_k$, Encoding

Interaction:

$C = \{c_1, \ldots, c_v\}, s_k$

$S = \{s_1, \ldots, s_w\}$,

$\pi \in_R S_w$

$P(x) = x^{(M-v)} \Pi_{i=1}^v (x - c_i)$

$= \sum_{i=0}^M \alpha_i t^i$

$\{\text{Enc}(\alpha_i)\}_{i=0}^M$
“Bounded”- Freedman et al

Setup: run by Client, selects an encoding of $U$ as a subset of $\mathbb{Z}_n^*$, runs a key generation algorithm for Paillier encryption which outputs $(s_k, p_k)$.

$p_k$, Encoding

Interaction:

$C = \{c_1, \ldots, c_v\}, s_k$

$P(x) = x^{(M-v)} \prod_1^v (x-c_i)$

$= \sum_{0}^{M} \alpha_i t^i$

{ $\{\text{Enc}(\alpha_i)\}$ } $i=0^M$

{ $\{\pi(e_i)\}$ } $i=1^M$

$S = \{s_1, \ldots, s_w\}$,

$\pi \in_R S_w$

for $i=1$ to $w$

$r_i \in_R \mathbb{Z}_n^*$

$e_i = \text{Enc}(r_i P(s_i) + s_i)$

for $i=w+1$ to $M$

$e_i \in_R \mathbb{Z}_n^2$
"Bounded"- Freedman et al

Setup: run by Client, selects an encoding of $U$ as a subset of $\mathbb{Z}_n^*$,
runs a key generation algorithm for Paillier encryption which outputs $(s_k, p_k)$

$p_k$, Encoding

Interaction:

$C = \{c_1, ..., c_v\}, s_k$

$P(x) = x^{(M-v)} \Pi_{1}^{v} (x - c_i) = \sum_{0}^{M} \alpha_i t^i$

$\{ \text{Enc}(\alpha_i) \}_{i=0}^{M}$

$\{ \pi(e_i) \}_{i=1}^{M}$

$S = \{s_1, ..., s_w\}$

$\pi \in \mathbb{R} S_w$

for $i=1$ to $w$

$r_i \in \mathbb{R} \mathbb{Z}_n^*$

$e_i = \text{Enc}(r_i P(s_i) + s_i)$

for $i=w+1$ to $M$

$e_i \in \mathbb{R} \mathbb{Z}_n^2$

$C \cap S := \{c_1, ..., c_v\} \cap \{ \text{Dec}(e_i) \}_{i=1}^{M}$
“Bounded”-Freedman SH-PSI

- We get:
  - **Correctness**: once Interaction is executed, with overwhelming probability, C gets as private output the intersection $C \cap S \checkmark$
  - **Server Privacy**: the execution of Interaction leaks no information about $C \cap S^c \checkmark$
  - **Client Privacy**: no information about $C$, apart from its size $v$, is leaked from Interaction. $\checkmark$
  - Both are $v$ and $w$ are hidden (up to the bound $M$).
Technique III: (Oblivious) Pseudorandom Function Evaluation (One Side SH with TTP)
What is OPRFE?

Any two party protocol, run between C and S so that:

- **Input:**
  - Client: x in the appropriate domain
  - Server: a key k designating a pseudorandom function $f_k$ from a public family $F = \{f_k\}_{k \in K}$

- **Output:**
  - Client: $f_k(x)$
  - Server: nothing

**Example:**

An example: Hazay and Lindell [TCC 2008]

Setup: run by S, selects a secret random key $k$ designating a pseudorandom function $f_k$ from a PRF family $F = \{f_k\}_{k \in K}$. He computes and publishes $R = \{r_1, \ldots, r_w\}$ where $r_i = f_k(s_i)$.

$$ R = \{r_1, \ldots, r_w\}, \quad F = \{f_k\}_{k \in K} $$
**An example: Hazay and Lindell [TCC 2008]**

Setup: run by $S$, selects a secret random key $k$ designating a pseudorandom function $f_k$ from a PRF family $F = \{f_k\}_{k \in K}$. He computes and publishes $R = \{r_1, \ldots, r_w\}$ where $r_i = f_k(s_i)$.

Interaction:

- $C = \{c_1, \ldots, c_v\}$
- $S = \{s_1, \ldots, s_w\}$
- $f_k(c_i)$
- $k$
- OPRF

$R = \{r_1, \ldots, r_w\}$, $F = \{f_k\}_{k \in K}$
An example: Hazay and Lindell [TCC 2008]

Setup: run by S, selects a secret random key $k$ designating a pseudorandom function $f_k$ from a PRF family $F = \{f_k\}_{k \in K}$. He computes and publishes $R = \{r_1, \ldots, r_w\}$ where $r_i = f_k(s_i)$.

Interaction:

$C = \{c_1, \ldots, c_v\}$

$S = \{s_1, \ldots, s_w\}$

$C \cap S$ from $R \cap \{f_k(c_i)\}_{i=1}^v$
Extension

- Size Hiding: one side: we may hide client’s set size and boost efficiency using a TTP.

\[ C = \{c_1, \ldots, c_v\} \]
\[ S = \{s_1, \ldots, s_w\} \]
\[ k \in_R K \]
\[ R = \{r_1, \ldots, r_w\} \text{ where } r_i = f_k(s_i) \]

Gets \( C \cap S \) from \( R \cap \{f_k(c_i)\}_{i=1}^{v} \)
(O)PRF based SH-PSI with a TTP

- We get:
  - **Correctness**: once Interaction is executed, with overwhelming probability, C gets as private output the intersection $C \cap S$.
  - **Server Privacy**: the execution of Interaction leaks no information about $C \cap S^c$.
  - **Client Privacy**: no information about $C$, apart from its size $v$, is leaked from Interaction.

- Only $v$ is hidden.

- We can use oblivious evaluation or not depending on the level of confidence on the TTP.
Technique IV:
RSA-based construction
(One Side SH with TTP)
RSA-based One-side Size Hiding

- Based on RSA construction in the ROM, Ateniese et al. [PKC 2011]:

- Our construction: RSA construction with a TTP, without random oracles
Our construction: Set up.

TTP executes an RSA key generation algorithm, keeps $N, e, d$, private. Further, selects an encoding of $U$ as a subset of $\mathbb{Z}_N^*$, fixes a group $G$ of order $p$ prime ($p$ smallest larger than $N$), and selects a generator $g$ of $G$. Finally, selects a strongly universal hash function $H : \mathbb{Z}_N^* \mapsto \mathbb{Z}_p^*$.

Encoding, $G$, $g$, $p$
Our construction: Set up.

\[ C = \{c_1, \ldots, c_v\} \]
\[ S = \{s_1, \ldots, s_w\} \]

\[ \{c^*_{1}, \ldots, c^*_{v}\} \]
\[ X^* = H(x^d) \]
\[ \{s^*_{1}, \ldots, s^*_{w}\} \]
Our construction: Interaction

\[ R_c \in_{R} Z_p \]
\[ \text{PCH} = \prod_{i=1}^{v} c_i^* \]
\[ X = g^{R_c \cdot \text{PCH}} \mod N \]

\[ S = \{s_1, \ldots, s_w\} \]
\[ \pi \in_{R} S_w \]
Our construction: Interaction

\[ R_c \in_{R} Z_p \]
\[ \text{PCH} = \Pi_{l=1}^{v} c^*_{l} \]
\[ X = g^{R_c \ PCH} \mod N \]

\[ S' = \{s_1, \ldots, s_w\} \]
\[ \pi \in_{R} S_w \]

\[ Z, \{t_1, \ldots, t_w\} \]

\[ R_s \in_{R} Z_p, Z = g^{R_s} \mod N \]

For \( j=1 \) to \( w \),
\[ Y_j = X^{R_s \ (1/s^*_j)} \mod N \]

\[ \{t_1, \ldots, t_w\} = \pi(Y_1, \ldots, Y_w) \]
Our construction: Interaction

\( R_c \in_{R} Z_p \)

\( \text{PCH} = \prod_{i=1}^{\text{v}} c_{i}^{*} \)

\( X = g^{R_c \cdot \text{PCH}} \mod N \)

\( S = \{s_1, \ldots, s_w\} \)

\( \pi \in_{R} S_w \)

For\(i=1\) to \(\text{v}\),

\( \text{PCH}_i = \prod_{l \neq i, l=1}^{\text{v}} c_{l}^{*} \)

\( Z_i = Z^{R_c \cdot \text{PCH}_i} \mod N \)

If \( Z_i \in \{t_1, \ldots, t_w\} \), \( c_i \in C \cap S \)

\( R_s \in_{R} Z_p, Z = g^{R_s} \mod N \)

For\(j=1\) to \(\text{w}\),

\( Y_j = X^{R_s \cdot (1/s_{j}^{*})} \mod N \)

\( \{t_1, \ldots, t_w\} = \pi(Y_1, \ldots, Y_w) \)
We get:

- **Correctness**: once *Interaction* is executed, with overwhelming probability, C gets as private output the intersection $C \cap S$ and $|S|$. $\surd$ - this comes from the fact that $H$ is a bijection.

- **Server Privacy**: the execution of *Interaction* leaks no information about $C \cap S^c$ (apart from its size) $\surd$ - similar as client privacy + the fact that $H$ is strongly universal.

- **Client Privacy**: no information about $C$, apart from its size $v$, is leaked from *Interaction* $\surd$ -- $x$ is statistically indistinguishable from a random element in $G$.

- Only $w$ is leaked.
Final Remarks
### What is possible

<table>
<thead>
<tr>
<th>TTP</th>
<th>SH</th>
<th>ASSUMPTION</th>
<th>EFFICIENCY</th>
<th>RNDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>C/S</td>
<td>NONE</td>
<td>NO</td>
<td>2</td>
</tr>
<tr>
<td>NO</td>
<td>C/S</td>
<td>STANDARD</td>
<td>NO</td>
<td>2</td>
</tr>
<tr>
<td>NO</td>
<td>C/S*</td>
<td>STANDARD</td>
<td>YES</td>
<td>2</td>
</tr>
<tr>
<td>YES</td>
<td>C</td>
<td>STANDARD</td>
<td>YES</td>
<td>1</td>
</tr>
<tr>
<td>YES</td>
<td>C</td>
<td>STANDARD</td>
<td>YES</td>
<td>2</td>
</tr>
<tr>
<td>YES</td>
<td>C</td>
<td>ROM</td>
<td>YES</td>
<td>3</td>
</tr>
<tr>
<td>YES</td>
<td>C</td>
<td>ROM</td>
<td>YES</td>
<td>1</td>
</tr>
</tbody>
</table>

**Notes:**
- **RANDOM FUNCTIONS AND PROTOCOL**
- **POLYNOMIALS**
- **OPRFE**
- **RSA – BASED.**
- **NOT IN THIS TALK**
Open Problems

- Efficient One Side SH protocol without TTP in the standard model.
- Efficient Two Side SH protocol without TTP in the standard model.
- Several executions: unlinkability.
- Malicious Insiders.
Thank you!