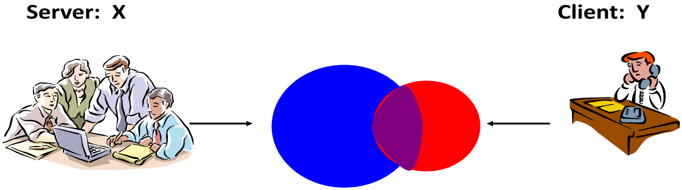
Efficient Private Set Intersection Cardinality in the Presence of Malicious Adversaries

Sumit Kumar Debnath and Ratna Dutta

Department of Mathematics
Indian Institute of Technology Kharagpur
Kharagpur-721302, India

Outline

Private Set Intersection (PSI) Protocol



- At the end of the protocol, either one of them gets the intersection, yielding-one-way PSI, or both of them get the intersection yielding-mutual PSI (mPSI)

Private Set Intersection Cardinality(PSI-CA)

This is a variant of PSI, where the participants wish to learn the cardinality of the intersection rather than the content.

Private Set Intersection (PSI) Protocol

The applications of PSI and PSI-CA protocols are as follows:

- Two real estate companies would like to identify customers (e.g., home owners) who are double-dealing, i.e., have signed exclusive contracts with both companies to assist them in selling their properties.
- Two different health organizations want to know the number of common villagers who are suffering from a particular disease in a village. None of the organizations will reveal their list of suspects but they may learn the number of common suspects by running an PSI-CA.

Cryptographic Building Blocks

- Bloom Filter of [1]
- Homomorphic Encryption of [2]

[1]: B. H. Bloom, Communications of the ACM 1970.
[2]: T. ElGamal, In Advances in Cryptology, Springer, 1985.

Bloom Filter (BF)

Bloom filter (BF) is a data structure that represents a set $X = \{x_1, ..., x_v\}$ of v elements by an array of m bits and uses k independent hash functions $H = \{h_0, h_1, ..., h_{k-1}\}$ with $h_i : \{0,1\}^* \to \{0, 1, ..., m-1\}$ for $i = 0, 1, ..., k-1$. Bloom filter of X is denoted by BF_X.

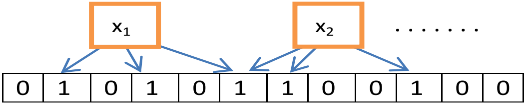
Bloom Filter (BF)

Choose $m = 12$ and $k = 3$.

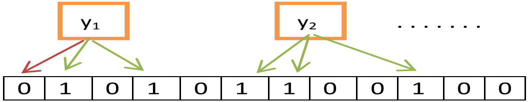
Initialization:

0	0	0	0	0	0	0	0	0	0	0	0

Add step: Suppose $(h_0(x_1) = 5, h_1(x_1) = 1, h_2(x_1) = 3)$, $(h_0(x_2) = 9, h_1(x_2) = 6, h_2(x_2) = 5)$......



Check step: Suppose $(h_0(y_1) = 0, h_1(y_1) = 3, h_2(y_1) = 1)$, $(h_0(y_2) = 9, h_1(y_2) = 6, h_2(y_2) = 5)$......



ElGamal encryption

This is a homomorphic encryption under the modulo multiplication and consists the algorithms $(\mathcal{EL}.\mathrm{Setup}, \mathcal{EL}.\mathrm{KGen}, \mathcal{EL}.\mathrm{Enc}, \mathcal{EL}.\mathrm{Dec})$:

• $\mathrm{par} = (\mathbf{p}, \mathbf{q}, \mathbf{g}) \leftarrow \mathcal{EL}.\mathrm{Setup}(\mathbf{1}^{\kappa})$, where p, q are primes such that q divides $p-1$ and g is a generator of the unique cyclic subgroup $\mathbb{G}$ of $\mathbb{Z}_p^*$ of order q.

• $(\mathbf{pk_U} = \mathbf{h}, \mathbf{sk_U} = \mathbf{x}) \leftarrow \mathcal{EL}.\mathrm{KGen}(\mathrm{par})$, where $x \longleftarrow \mathbb{Z}_q$ and $y = g^x$.

• $\mathbf{c} \leftarrow \mathcal{EL}.\mathrm{Enc}(\mathbf{m}, \mathbf{pk_U}, \mathrm{par}, \mathbf{r})$, where $c = E_{pk_U}(m) = (\alpha, \beta) = (g^r, mh^r)$ and $r \longleftarrow \mathbb{Z}_q$.

• $\mathbf{m} \leftarrow \mathcal{EL}.\mathrm{Dec}(\mathbf{E_{pk_U}}(\mathbf{m}), \mathbf{sk_U})$, where m can be computed as $\frac{\beta}{\alpha^x} = \frac{m(g^x)^r}{(g^r)^x} = m$.

Decisional Diffie-Hellman (DDH) Assumption

Let the algorithm $(n, g) \leftarrow gGen(1^\kappa)$, where g is a generator of a multiplicative group $\mathbb{G}$ of order n. Suppose $a, b, c \leftarrowtail \mathbb{Z}_n$. Then the DDH assumption states that no PPT algorithm $\mathcal{A}$ can distinguish between the two distributions $\langle g^a, g^b, g^{ab} \rangle$ and $\langle g^a, g^b, g^c \rangle$ i.e., $|Prob[\mathcal{A}(g, g^a, g^b, g^{ab}) = 1] - Prob[\mathcal{A}(g, g^a, g^b, g^c) = 1]|$ is negligible function of κ.

Zero-Knowledge Proof of Knowledge PoK$\{\alpha \mid X = g^{\alpha}\}$

- The prover chooses $v \leftarrow \mathbb{Z}_q$ and sends the commitments $\overline{X} = g^v$ to the verifier.
- The verifier chooses $c \leftarrow \mathbb{Z}_q$ and gives c as challenge to the prover.
- The prover sets $r = v + c\alpha$ and sends the response r to the verifier.
- The verifier checks whether the relations $g^r = \overline{X}X^c$ hold. If this holds, then the verifier accepts it, otherwise rejects it.

PSI-CA-I

C's private input $X = \{x_1, ..., x_v\}$

Common input: $\mathsf{par} = (p, q, g)$

S's private input $Y = \{y_1, ..., y_w\}$

$(pk_C, sk_C) \leftarrow \mathcal{EL}.\mathsf{KGen}(\mathsf{par})$
for $i = 1, ..., v,$
$r_{x_i} \leftarrow\!\!\prec \mathbb{Z}_q,$
$E_{pk_C}(x_i) = (c_{x_i} = g^{r_{x_i}}, d_{x_i} = x_i h^{r_{x_i}})$
$\leftarrow \mathcal{EL}.\mathsf{Enc}(x_i, pk_C, \mathsf{par}, r_{x_i});$
$\pi_1 = \mathsf{PoK}\{(r_{x_1}, ..., r_{x_v}) | \wedge_{i=1}^{v} (c_{x_i} = g^{r_{x_i}})\}$
$R_1 = \langle \{E_{pk_C}(x_1), ..., E_{pk_C}(x_v)\}, pk_C, \pi_1 \rangle$

$\xrightarrow{R_1}$

verifies the validity of π_1 by interacting with C as discussed in the previous slide

$r \leftarrow\!\!\prec \mathbb{Z}_q;$
$\widehat{Y} = \{t_1 = (y_1)^r, ..., t_w = (y_w)^r\};$
for $i = 1, ..., v,$
$(E_{pk_C}(x_i))^r = (\hat{c}_{x_i} = (c_{x_i})^r, \hat{d}_{x_i} = (d_{x_i})^r$
$\mathsf{Perm}\{(E_{pk_C}(x_1))^r, ..., (E_{pk_C}(x_v))^r\}$
$= \{(E_{pk_C}(\bar{x}_1))^r, ..., (E_{pk_C}(\bar{x}_v))^r\} = \overline{X};$
$\pi_2 = \mathsf{IPoK}\{(r) | (\Pi_{i=1}^{v} \hat{c}_{\bar{x}_i} = (\Pi_{i=1}^{v} c_{x_i})^r)$
$\wedge (\Pi_{i=1}^{v} \hat{d}_{\bar{x}_i} = (\Pi_{i=1}^{v} d_{x_i})^r)\}$
$R_2 = \langle \widehat{Y} = \{t_1, ..., t_w\}, \overline{X}, \pi_2 \rangle.$

$\xleftarrow{R_2}$

checks the validity of π_2 by interacting with S as discussed in the previous slide

for $i = 1, ..., v,$
$s_i = (\bar{x}_i)^r \leftarrow \mathcal{EL}.\mathsf{Dec}((E_{pk_C}(\bar{x}_i))^r, sk_C);$
sets $|X \cap Y| = |\{s_1, ..., s_v\} \cap \{t_1, ..., t_w\}|$

PSI-CA-I contd...

Correctness: As the set $\{\bar{x}_1, ..., \bar{x}_v\}$ is same as $\{x_1, ..., x_v\}$ in some order, the set $\{\bar{x}_1^r, ..., \bar{x}_v^r\}$ is same as $\{x_1^r, ..., x_v^r\}$ in that order. Thus we have the following:

$$
\begin{aligned}
|\{s_1, ..., s_v\} \cap \{t_1, ..., t_w\}| &= |\{\bar{x}_1^r, ..., \bar{x}_v^r\} \cap \{y_1^r, ..., y_w^r\}| \\
&= |\{x_1^r, ..., x_v^r\} \cap \{y_1^r, ..., y_w^r\}| \\
&= |\{x_1, ..., x_v\} \cap \{y_1, ..., y_w\}| \\
&= |X \cap Y|
\end{aligned}
$$

PSI-CA-II

C's private input $X = \{x_1, ..., x_v\}$

Common input:
$\mathsf{par} = (p, q, g)$

S's private input $Y = \{y_1, ..., y_w\}$

$(pk_C, sk_C) \leftarrow \mathcal{EL}.\mathsf{KGen}(\mathsf{par})$;
for $i = 1, ..., v$,
$r_{x_i} \leftarrow\!\!\!\!\!\!\shortmid\; \mathbb{Z}_q$
$E_{pk_C}(x_i) = (c_{x_i} = g^{r_{x_i}}, d_{x_i} = x_i h^{r_{x_i}})$
$\leftarrow \mathcal{EL}.\mathsf{Enc}(x_i, pk_C, \mathsf{par}, r_{x_i})$;
$\pi_1 = \mathsf{PoK}\{(r_{x_1}, ..., r_{x_v}) | \wedge_{i=1}^{v} (c_{x_i} = g^{r_{x_i}})\}$
$R_1 = \langle \{E_{pk_C}(x_1), ..., E_{pk_C}(x_v)\}, pk_C, \pi_1 \rangle$

$\xrightarrow{R_1}$

verifies the non-interactive proof π_1
$r \leftarrow\!\!\!\!\!\!\shortmid\; \mathbb{Z}_q$;
$\widehat{Y} = \{t_1 = (y_1)^r, ..., t_w = (y_w)^r\}$;
for $i = 1, ..., v$,
$(E_{pk_C}(x_i))^r = (\hat{c}_{x_i} = (c_{x_i})^r, \hat{d}_{x_i} = (d_{x_i})^r$
$\mathsf{Perm}\{(E_{pk_C}(x_1))^r, ..., (E_{pk_C}(x_v))^r\}$
$= \{(E_{pk_C}(\bar{x}_1))^r, ..., (E_{pk_C}(\bar{x}_v))^r\} = \overline{X}$;
constructs $\mathsf{BF}_{\widehat{Y}}$;
$\pi_2 = \mathsf{PoK}\{(r) | (\Pi_{i=1}^{v} \hat{c}_{\bar{x}_i} = (\Pi_{i=1}^{v} c_{x_i})^r)$
$\wedge (\Pi_{i=1}^{v} \hat{d}_{\bar{x}_i} = (\Pi_{i=1}^{v} d_{x_i})^r)\}$
$R_2 = \langle \mathsf{BF}_{\widehat{Y}}, \overline{X}, \pi_2 \rangle$.

$\xleftarrow{R_2}$

verifies the non-interactive proof π_2
sets $\mathsf{card} = 0$;

for $i = 1, ..., v$,
$s_i = (\bar{x}_i)^r \leftarrow \mathcal{EL}.\mathsf{Dec}((E_{pk_C}(\bar{x}_i))^r, sk_C)$,
if $\mathsf{BF}_{\widehat{Y}}[h_j(s_i)] = 1 \ \forall j = 0, ..., k-1$
then $\mathsf{card} = \mathsf{card} + 1$;
outputs card as $|X \cap Y|$

Security

The security definition is based on a comparison between the ideal model and real model.

Security Requirements

- **Privacy:** Each party should learn whatever prescribed in the protocol, not more than that.
- **Correctness:** At the end of interaction, each party should receive correct output.

Theorems

Theorem

If the encryption scheme $\mathcal{EL}$ is semantically secure, the associated proof protocols are zero knowledge proof and the associated permutation is random, then our PSI-CA-I is a secure computation protocol for the functionality $\mathcal{F}_{\text{card}} : (X, Y) \to (|X \cap Y|, \perp)$ against malicious adversaries in standard model.

Theorems contd...

Theorem

If the encryption scheme $\mathcal{EL}$ is semantically secure, the associated proof protocols are zero knowledge proof and the associated permutation is random, then our PSI-CA-II is a secure computation protocol for the functionality $\mathcal{F}_{\text{card}} : (X, Y) \rightarrow (|X \cap Y|, \perp)$ against malicious adversaries in ROM except with negligible probability $\frac{1}{2^k}$.

Efficiency

Table: : Comparison of PSI-CA protocols

Protocol	Security model	Adv. model	Security assumption	Comm.	Comp.	Based on
[1]	Std	Mal	HE	$O(t^2 v)$	$O(v^2)$	OPE
[2]	Std	SH	SD and SC	$O(w+v)$	$O(w \log\log v)$	OPE
Sch. 1 of [3]	ROM	SH	DDH and GOMDH	$O(w+v)$	$O(w+v)$	
Sch. 2 of [3]	ROM	MS, SHC	GOMDH	$O(w+v)$	$O(w+v)$	
PSI-CA-I	Std	Mal	DDH	$O(w+v)$	$O(w+v)$	
PSI-CA-II	ROM	Mal	DDH	$O(w+v)$	$O(w+v)$	BF

[1] L. Kissner and D. Song. Privacy-preserving set operations. In Advances in Cryptology 2005.
[2] S. Hohenberger and S. A. Weis, In Privacy Enhancing Technologies 2006.

[3] E. De Cristofaro, P. Gasti, and G. Tsudik, In Cryptology and Network Security 2012.

Conclusion

- This paper consists of two flavors of PSI-CA, one is secure in standard model and the other one is secure in ROM. Both are secure against malicious parties with linear computation complexity under DDH assumption.
- In contrast to PSI-CA-I, PSI-CA-II requires at most $5v + 4$ group elements instead of $6v + w + 4$.
- Our PSI-CA constructions are the *first* to achieve *linear complexity* in the presence of *malicious* adversaries.
- Furthermore, each of our PSI-CA construction can be converted to efficient PSI protocol by removing the associated permutation.


ThaNK
YOU

For any query mail at sd.iitkgp@gmail.com