Seminar V: Private Set Intersection Protocols

Sven Laur
Helsinki University of Technology
swen@math.ut.ee, slaur@tcs.hut.fi
**Known results**

Let $\mathcal{X}$ be the universal set of all possible elements and $N = |\mathcal{X}|$.

- Private equality tests $x \in \mathcal{X}$?
  - Yao’s circuit evaluation $O(\log N)$ gates (oblivious transfers).
  - Special PET protocols have one round, but the asymptotic complexity is same $O(\log N)$.

- Disjointness and cardinality tests of $X \cap Y$
  - The lower communication complexity bound is $\Omega(\min \{|X|, |Y|\})$.
  - The good approximation still requires $\Omega(\min \{|X|, |Y|\})$. 
Various scenarios of private set intersection

A client Alice has a set $X = \{x_1, \ldots, x_k\}$.
A server Bob has a set $Y = \{y_1, \ldots, y_\ell\}$.

Different tasks

- Private matching (PM) — Alice learns $X \cap Y$.
- Private cardinality (PC) — Alice learns $|X \cap Y|$.
- Private threshold test (PT) — Alice learns $|X \cap Y| > \tau$.

Attack scenarios

- Semi-honest model
- Malicious Alice and Bob.
A basic tool—an indicator polynomial

Consider a set \( X = \{x_1, \ldots, x_k\} \subseteq \mathbb{F}_q \) then the indicator polynomial

\[
P_X(y) = \prod_{i=1}^{k} (x_i - y) = \sum_{i=0}^{k} c_i y^i
\]

has a trivial property

\[
P_X(y) r = 0 \iff P_X(y) = 0 \iff y \in X
\]

The property (LZ) holds in residue rings \( \mathbb{Z}_m \) if

- \( x_i, y \in [0, \kappa/2) \), where \( \kappa \) is the smallest zero-divisor

\[
\kappa = \min \{ a : \exists b \neq 0 \land ab \equiv 0 \mod m \}.
\]
Corresponding PM protocol

**Input:** Private input sets $X$ and $Y$ such that $k = |X|, \ell = |Y| \ll m$.

**Output:** Alice learns $X \cap Y$ and Bob $\perp$.

**Step Setup phase**
- Alice chooses a private key of homomorphic encryption scheme.
- Alice sends the public key to Bob.

**Step 1**
- Alice constructs the indicator polynomial $P_X$ and encrypts coefficients $c_i$.
- Alice sends coefficients $(E(c_0), \ldots, E(c_k))$ to Bob.

**Step 2**
- for $y \in Y$ do
  - Bob evaluates $m_i = E(rP_X(y) + y)$ with a fresh random number $r \neq 0$.
  - Bob sends randomly permuted $m_i$ to Alice.

**Step 3**
- for $i = 1$ to $\ell$ do
  - if $D(m_i) \in X$ then Alice outputs $D(m_i)$.
**Correctness**

The error probability is negligible.

- If $y \in X$ then (LZ) property assures $D(m_i) = P_X(y)r + y = y \in X$.

- If $y \notin X$ and $r$ is invertible $rP_X(y)$ has uniform distribution and
  $\Pr[D(m_i) \in X] = \frac{|X|}{\varphi(m)} \approx \frac{k}{m} < 2^{-1000}$

- The probability that $r$ is zero-divisor is negligible $2^{-500}$.

Alternatively, we could use a large factor of $m$
Security

- Since Bob manipulates with encryptions the privacy guarantee of Alice computational.

- If $y \notin X$ then Alice receives $zr + y$, where $z$ is invertible element. Hence, the security guarantee of Bob is information theoretical, iff the statistical difference

$$\Delta_1 = \left( \frac{1}{\varphi(m)} - \frac{1}{m} \right) \varphi(m) + (m - \varphi(m)) \frac{1}{m} = 2 \left( 1 - \frac{\varphi(m)}{m} \right)$$

is small. Otherwise we get a vague computational guarantee.

- The probability $r$ is not invertible is negligible.
Complexity analysis

- Alice sends $k + 1$ and Bob sends $\ell$ ciphertexts.
- Alice computes $k + 1$ coefficients. The naive complexity is $O(k^2)$.
- Alice computes $k + 1$ encryptions and $\ell$ decryptions.
- Bob evaluates $P_X$ at $\ell$ different locations, it takes $O(k\ell)$ exponentiations.

Computations are dominated by $k\ell$ exponentiations!
The first hack. Applying Horner’s rule

- There is a big computational difference between $E(z)^y$ and $E(z)^{y^i}$.

- Bob should compute

\[
E(c_0 + c_1 y + \cdots + c_k y^k) = E(c_0 + y(c_1 + y(c_2 \cdots + yc_k))) \\
= E(c_0) \cdot ((E(c_1) \cdot (E(c_2) \cdot (\cdots E(c_k) y \cdots )y)y)y) y
\]

- Bob does $k$ short exponentiations.

- The optimization makes the process approximately 50 times faster.
The second hack. Divide and conquer technique

- The computation complexity of Bob depends on the degree of $P_X$. A smaller degree reduces amount of computations.

- If we divide $X = X_1 \cup X_2$ and publish corresponding supersets $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$, the degree and consequently the number of exponentiations decreases twofold.

- But this is not a secure and efficient solution. We could use random hash function $h : \mathcal{X} \rightarrow \{1, 2\}$ instead and define

$$\mathcal{X}_i = \{x \in \mathcal{X} : h(x) = i\}, \quad i = 1, 2.$$  

Then with a high probability

$$|X \cap \mathcal{X}_1| \approx |X \cap \mathcal{X}_2|.$$
Balanced hashing. Tradeoff between complexities

Consider two hash functions $h_1, h_2 : X \to \{1, \ldots, B\}$. Let $C(i)$ denote the dynamic number of elements of $X$ with $h(x) = i$. Then the balanced hash function

$$h(x_i) = \begin{cases} 
    h_1(x_i), & \text{if } C(h_1(x_i)) < C(h_2(x_i)), \\
    h_2(x_i), & \text{otherwise}.
\end{cases}$$

The maximum number of elements of $X$ in the bins

$$M = \Theta(k/B) + (1 + o(1)) \frac{\ln \ln B}{\ln 2}$$

with high probability.

- Setting $B = k/\ln \ln k$, we get $M = O(\ln \ln k)$.
- In practice $M \leq 5$ with probability $10^{-58}$.
Implementation details

Alice and Bob use keyed fast (non-)cryptographic hash to divide elements of $X$ and $Y$ into $B$ bins. Let $M$ be the degree bound.

Alice must send $M + 1$ coefficients of $B = \frac{k}{\ln \ln k}$ polynomials

$$P_j(y) = \prod_{x \in X \cap \chi_i} (x - y) = \sum_{i=0}^{M} c_{ij}y^i.$$

For each $y \in Y$ Bob must evaluate both polynomials

$$m_j = E(P_j(y)r + y), \quad j = h_1(y), h_2(y).$$

- The communication complexity increases about 4 times.
- The workload of Alice doubles.
- The workload of Bob decreases rapidly $O(2M\ell) \ll O(k\ell)$. 

Special Course in Cryptology, 17.02.2004 Seminar V: Private Set Intesection Protocols, Sven Laur
What about security?

- If keys of $h_1$ and $h_2$ are chosen randomly, then the probability that there are more than $M$ elements in one bucket is small, say $10^{-58}$.

  The protocol fails or something leaks only if $M$ is too small.

- Since the value of $P_j(y)r + y$ is still garbage, when $y \notin X_j$ or $y$ the privacy guarantee of Bob is still information theoretical.

  In other words, $m_i$ that corresponds to a wrong bin reveals nothing about $y$. 

What about PC and PT?

- The protocol allows easily to compute private cardinality. Bob must evaluate $E(rP(y))$ instead.

- The generalization to private threshold test reduces circuit complexity.
  - Basically, we can compute shares $s_i, t_i \in \mathbb{Z}_m$ such that
    \[
    s_i + t_i \equiv 0 \pmod{m} \iff y_i \in X.
    \]
  - Thus the corresponding Yao’s circuit has lower complexity, since each pair of shares encodes predicate $y_i \in X$.
  - This is not a major breakthrough.
Protection against malicious Alice

If Alice sets $P_X \equiv 0$, she will learn $Y$. Bob needs a guarantee $|X| = k$. First assume that we have only one bin.

To prove that $\deg P_X = k$ Alice reveals all coefficients. But this violates the privacy of Alice.

Hence, Alice has to mask his entries with keyed cryptographic pseudo-random function $f$. Then values $f(s, x_i)$ do not reveal $x_i$ provided $s$ is secret.

There is no point in cheating if either Alice gets caught or she cannot cheat.

The aim: Alice passes a test, only if she is honest with extremely high probability.
Almost perfect protection mechanism

Alice chooses $2L$ random keys $s_1, \ldots, s_{2L}$ and generates indicator polynomials

$$P_j(y) = \prod_{i=1}^{k} \left[ f(x_i, s_j) - y \right] = \sum_{i=0}^{k} c_{ij} y^i$$

and sends encryptions $\mathcal{E}(c_{ij})$ to Bob.

Bob asks to reveal coefficients $c_{ij}$ and $f(x_i, s)$ of $L$ polynomials. Alice gets caught with an extremely high probability if she lied about $L$ polynomials.

Alice reveals keys $s_j$ of other $L$ polynomials. Bob forces all or nothing behavior by setting

$$\mathcal{E}(P_j(F(y, s_j))^r + u_j), \quad \bigoplus_{j \in \mathcal{J}} u_j = y.$$ 

Alice gets something useful only if $y$ is the root of all polynomials.
Alice can still cheat!

Alice might choose weak keys $s$ so that

$$f(s, x_i) \neq f(s, x_j), \quad i \neq j$$

but

$$\forall y \exists i : f(s, x_i) = f(s, y)$$

To eliminate this threat Bob chooses a collision resistant hash function $g$ and compose a fair keyed hash $f'(s, \cdot) = f(s, g(\cdot))$.

Alternatively we could use keyed pseudo-random permutations (block-ciphers). It is possible if block-size is less than $\log m$. 
A trouble with bins

If bins contain at most $M$ elements of $X$ then some bins are under-fulled.

We cannot reveal how many elements of $X$ belong to the $i$th bin, since the superset $\mathcal{X}$ might be small enough to use brute force search algorithms.

We can use false roots to increase the degree of under-fulled polynomials. Now two options exist:

- We take different elements — Alice cannot prove to Bob that $|X| = k$.
- Alice takes repeating elements — finding “greatest common divisor” allows Bob to reveal bin counts of $h_1$ and $h_2$.

Hence, Alice can securely prove only that $|X| \leq MB = O(k)$!!
Can we prove if Bob lies?

Bob can trivially lie by replacing $E(rP(y) + y)$ with $E(y^*)$. Thus Alice should force Bob to prove that he computed $rP(y) + y$.

**Proof by random witness**

Bob chooses a random $s \in \mathbb{Z}_m$. Asks from a random oracle enough randomness $(r, r') = H_1(s)$ and computes $e_1 = E(rP(y) + y)$ and $e_2 = E(rP(y) + s)$.

To complete the proof he asks from an other random oracle $h = H_2(r', y)$ and sends triple $(e_1, e_2, h)$ to Alice.

Decoding procedure

- Set $s' = D(e_2)$ and $y' = D(e_1)$. Compute $(r, r') = H_1(s)$.
- Reject if $y' \notin X$ or $h \neq H_2(r', y')$, otherwise output $y'$. 

Special Course in Cryptology, 17.02.2004

Seminar V: Private Set Intersection Protocols, Sven Laur
Approximate solution of PC

Both parties compute indicator strings $X$ and $Y$.

They random sample $\mathcal{I}$ yields an (unbiased?) statistical estimate

$$\delta = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} x_i y_i.$$ 

If the sample has statistically significant size $|X \cap Y| \approx \delta N$.

The sampling is done with oblivious indexing. The communication complexity is asymptotically optimal.
The multi-party case

Let $A_n$ be the leader. The leader creates shares

$$y_i = \bigoplus_{j=1}^{n-1} u_{ij}, \quad i = 1, \ldots, \ell.$$ 

Parties $A_1, \ldots, A_{n-1}$ use two-party protocol, where leader computes

$$m_{\pi(i)j} = E(rP(y_i) + u_{ij}).$$

For each candidate $v_{ij} = D(m_{ij})$ parties $A_1, \ldots, A_{n-1}$ use Benaloh protocol to securely compute

$$v_i = v_{i1} \oplus \cdots \oplus v_{i,n-1}$$

All parties accept $v$ if it belongs to their sets.
Secure fuzzy matching of $n$ component vectors

Can Alice retrieve all fuzzy matches

$$\mathcal{F}_k(X, Y) = \{ z \in X : \exists x \in X \ \exists y \in Y \ H(z, x) \leq k \land H(z, y) \leq k \}$$

where $H$ is Hamming weight?

Choose indicator polynomials $P_j$ for each component so that

$$\sum_{j \in \mathcal{J}} P_j(x_{ij}) + a_{\mathcal{J}} = 0, \quad |\mathcal{J}| = k$$

Then again Bob can compute

$$E \left( \left( \sum_{j \in \mathcal{J}} P_j(y_i + a_{\mathcal{J}}) \right) r + y \right), \quad \text{forall } |\mathcal{J}| = k$$

and send them back in a randomly permuted fashion.