Multi-party Private Set Operations with an External Decider

Sara Ramezanian, Tommi Meskanen and Valtteri Niemi
{sara.ramezanian, tommi.meskanen, valtteri.niemi@helsinki.fi}

- Private Set Operation (PSO)
  - A cryptographic protocol for two or more parties.
  - All / some of the parties have an input set of private elements.
  - All / some of the parties want to compute the output of one or more set operations of the input sets.
- Goal is to compute the output without revealing anything about the elements that are not in the output.

External Decider (D) is a special party that does not have an input set, and is the only party who learns the output of the protocol [1].

Examples:
- Secure electronic voting.
- Privacy-preserving parental control [2].
- Decentralized social networking platform such as HELIOS [3], to obtain common interests between different groups of friends.

PSO-Lim Protocol for parties $P_1, \ldots, P_n$ and a decider D: Private sets $S_1, \ldots, S_n$ are subsets of $U = \{a_1, \ldots, a_u\}$. The decider wants to learn $S_T = (A_{1,1} \cup \cdots \cup A_{1,\alpha_1}) \cap \cdots \cap (A_{\beta,1} \cup \cdots \cup A_{\beta,\alpha_\beta})$ where $1 \leq \alpha_1 \leq \alpha_\beta \leq \beta \in N$, and each $A_{i,j} \in \{S_1, \ldots, S_n, \tilde{S}_1, \ldots, \tilde{S}_n\}$.

Set-up phase
1. D creates public and private keys for Paillier cryptosystem, and sends public keys and $U$ to parties. Parties create a shared repository.
2. Each $P_i$ creates a set containing many instances of $enc(0)$, and another set containing many instances of $enc(r)$, where $r$ is a random number chosen for that instance.
3. Parties create $\beta$ vectors $W^k$ of length $u$, where $W^k = (enc(r_{1,k}), \ldots, enc(r_{u,k}))$, when $1 \leq k \leq \beta$.

On-line phase
1. For every vector $W^k$ each $P_i$ modifies the vector as follows. If $u_j \in S_i$ then $P_i$ replaces $W^k_j$ with $enc(0)$. Otherwise, $P_i$ multiplies $W^k_j$ with $enc(0)$.
2. After all the vectors $W^k$ have been computed, one of the parties (e.g., $P_n$) creates a vector $Z$ where $Z_j = \prod_{k=1}^\beta W^k_j$. Party $P_n$ sends vector $Z$ to D.
3. D decrypts $Z$. If $\text{dec}(Z_j) = 0$, then $a_j \in S_T$. Otherwise, $a_j$ is not in $S_T$.

Performance: If public key in Paillier is of length 4096 bits and we assume that $\alpha = \beta = n$, the numbers in the table show the required time for each party to modify $Z$ with a single thread. When $u = 2^2, 2^5, 2^7, 2^{10}$ the decider needs 0.02, 0.17, 0.68, 5.51 seconds respectively, to decrypt this vector with 32 threads.

<table>
<thead>
<tr>
<th></th>
<th>$n = 3$</th>
<th>$n = 5$</th>
<th>$n = 10$</th>
<th>$n = 15$</th>
<th>$n = 20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u = 2^2$</td>
<td>0.001</td>
<td>0.002</td>
<td>0.003</td>
<td>0.005</td>
<td>0.007</td>
</tr>
<tr>
<td>$u = 2^5$</td>
<td>0.008</td>
<td>0.013</td>
<td>0.025</td>
<td>0.039</td>
<td>0.05</td>
</tr>
<tr>
<td>$u = 2^7$</td>
<td>0.031</td>
<td>0.05</td>
<td>0.1</td>
<td>0.15</td>
<td>0.2</td>
</tr>
<tr>
<td>$u = 2^{10}$</td>
<td>0.237</td>
<td>0.391</td>
<td>0.786</td>
<td>1.178</td>
<td>1.56</td>
</tr>
</tbody>
</table>