On optimal threshold defender structures of resharing-based oblivious shuffle protocols for secret-shared secure multi-party computations

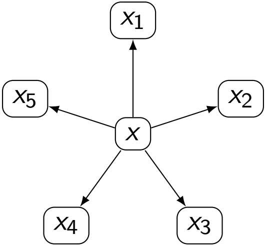
Jan Willemson

Cybernetica

Tõrve Theory Days
October 7th-9th, 2011

Secret Shared Databases

- If we need to compute with a dataset in a privacy-preserving manner, we can share the values between independent computing nodes using a *secret sharing scheme*.



- E.g. Sharemind uses additive secret sharing scheme, where

$$x_1 + x_2 + \ldots + x_m \equiv x \bmod 2^{32}$$

Adversary structures

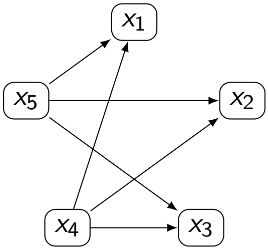
- Let X be the set of computing nodes. The secret sharing scheme is characterized by the *tolerable adversary structure* $\mathcal{A} \subseteq \mathcal{P}(X)$; i.e. for any $A \in \mathcal{A}$, the nodes of A should not be able to learn anything about the shared values.
  - We assume that the tolerable adversary structure is *monotone*, i.e. if $A \in \mathcal{A}$ and $B \subseteq A$ then $B \in \mathcal{A}$.
  - A t-threshold adversary structure is defined as

$$\{A \subseteq X : |A| \leq t\}$$

  - Sharemind additive sharing can resist value reconstruction attacks by $m - 1$ corrupt parties
  - Shamir secret sharing scheme can be tweaked to work for any t

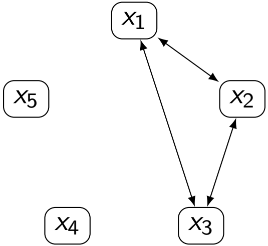
Database shuffle problem

- Many database manipulation operations can leak some information about the entries
  - E.g. their relative order, origin, etc.
- To fight this, the database needs to be shuffled in an oblivious manner
- One way to do it is to reshare the database among a subset of nodes and let them shuffle it, then repeat it with other subsets
  - Essentially, we have a mix-net



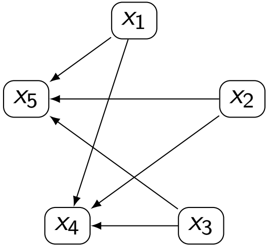
Database shuffle problem

- Many database manipulation operations can leak some information about the entries
  - E.g. their relative order, origin, etc.
- To fight this, the database needs to be shuffled in an oblivious manner
- One way to do it is to reshare the database among a subset of nodes and let them shuffle it, then repeat it with other subsets
  - Essentially, we have a mix-net



Database shuffle problem

- Many database manipulation operations can leak some information about the entries
  - E.g. their relative order, origin, etc.
- To fight this, the database needs to be shuffled in an oblivious manner
- One way to do it is to reshare the database among a subset of nodes and let them shuffle it, then repeat it with other subsets
  - Essentially, we have a mix-net



Security requirements

- We call the set of all reshuffling consortia $\mathcal{D} \subseteq \mathcal{P}(X)$ a *defender structure*
- No adversarial set should be able to learn all the shares of the values of the database, i.e.

$$\forall A \in \mathcal{A}\ \forall D \in \mathcal{D}\ D \not\subseteq A \tag{1}$$

  - For t-threshold case this reads as $\forall D \in \mathcal{D}\ |D| \geq t+1$
- No adversarial set should learn all the permutations, i.e.

$$\forall A \in \mathcal{A}\ \exists D \in \mathcal{D}\ A \cap D = \emptyset \tag{2}$$

- For both requirements, it is enough to consider only maximal adversarial and minimal defender sets (in terms of set inclusion)
- However, there can be several different defender structures

Research questions

- Given an adversary structure $\mathcal{A}$, find the least possible cardinality of the corresponding defender structures $\mathcal{D}$
  - Describe the defender structures explicitly if you can
- For m computing nodes and a t-threshold adversary structure $\mathcal{A}$, let $d(m, t)$ denote this minimal cardinality
  - Tabulate as many values of $d(m, t)$ as you can
  - Give good bounds for others
- For a given threshold t, find the optimal number m of the computing nodes so that the overall complexity of the shuffle protocol would be decreased

Some observations concerning $d(m, t)$

- $d(m, t)$ is well-defined iff $m \geq 2t + 1$
- For $m = 2t + 1$ we have $d(m, t) = \binom{m}{t}$
- $d(m, t)$ is monotonously decreasing as a function of m
- $d(m, t) \geq t + 1$
- $d((t + 1)^2, t) = t + 1$
- The last three observations imply

$$\lim_{m \to \infty} d(m, t) = t + 1$$

- For $t = 1$, the table looks like

m	1	2	3	4	5	6	...
$d(m, 1)$	-	-	3	2	2	2	...

A lower bound

Theorem

$$d(m,t) \geq \frac{\binom{m}{t}}{\binom{m-t-1}{t}}$$

Proof.

There are $\binom{m}{t}$ maximal adversarial sets. Each defender set D has at least $t+1$ elements, hence at most $m-t-1$ elements are left over from D. Thus, at most $\binom{m-t-1}{t}$ maximal adversarial sets satisfy the condition (2) for a given D. Consequently, each defender structure must have at least $\frac{\binom{m}{t}}{\binom{m-t-1}{t}}$ sets, including the minimal ones. $\square$

The case $t = 2$

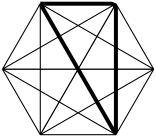
- We know $d(5, 2) = 10$
- From the Theorem we know that $d(6, 2) \geq \frac{\binom{6}{2}}{\binom{3}{2}} = \frac{15}{3} = 5$.
  Equality would mean that we can cover all the edges of the graph K_6 exactly with 5 triangles, but this is impossible, since

The case $t = 2$

- We know $d(5,2) = 10$
- From the Theorem we know that $d(6,2) \geq \frac{\binom{6}{2}}{\binom{3}{2}} = \frac{15}{3} = 5$. Equality would mean that we can cover all the edges of the graph K_6 exactly with 5 triangles, but this is impossible, since the vertex degrees of K_6 are odd. Hence $d(6,2) \geq 6$.

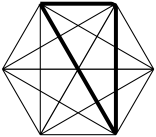
The case $t = 2$

- We know $d(5, 2) = 10$
- From the Theorem we know that $d(6, 2) \geq \frac{\binom{6}{2}}{\binom{3}{2}} = \frac{15}{3} = 5$. Equality would mean that we can cover all the edges of the graph K_6 exactly with 5 triangles, but this is impossible, since the vertex degrees of K_6 are odd. Hence $d(6, 2) \geq 6$.
- It is doable with 6 triangles. Just rotate this figure 6 times:



The case $t = 2$

- We know $d(5, 2) = 10$
- From the Theorem we know that $d(6, 2) \geq \frac{\binom{6}{2}}{\binom{3}{2}} = \frac{15}{3} = 5$.
  Equality would mean that we can cover all the edges of the graph K_6 exactly with 5 triangles, but this is impossible, since the vertex degrees of K_6 are odd. Hence $d(6, 2) \geq 6$.
- It is doable with 6 triangles. Just rotate this figure 6 times:



- For $t = 2$, the table looks like

m	1	2	3	4	5	6	7	8	9	10	...
$d(m, 2)$	-	-	-	-	10	6	5	4	3	3	...

On communication complexity of the shuffle protocol

- For $t = 2$ and $m = 5$, in total total

$$2 \cdot 2 \cdot 3 \cdot 10 = 120$$

  messages are sent in 10 rounds (not counting the messages exchanged between the defenders)

- For $t = 2$ and $m = 6$, we have to send

$$2 \cdot 3 \cdot 3 \cdot 6 = 108$$

  messages in 6 rounds

- Hence we see that increasing the number of computing nodes, the actual communication complexity may drop!

That's as far as I've got

- You can ask a question and then answer it yourself