“Secure” Multiparty Computation – Multi-round protocols

Li Xiong

CS573 Data Privacy and Security
Secure multiparty computation

- General circuit based secure multiparty computation methods
- Specialized secure multiparty computation protocols
  - Decision tree mining across horizontally partitioned data
  - Secure sum, secure union
  - Association rule mining across horizontally partitioned data
- Most of them rely on cryptographic primitives and are still expensive
- Multi-round protocols as an alternative
Multi-round protocols

- Max/min, top k
  - k-th element protocol using secure comparison (Aggarwal ’04)
  - Multi-round probabilistic protocols (Xiong ’07)
- OR (Union)
  - Commutative encryption based
  - Multi-round probabilistic protocols (Bawa ’03)

Secure computation of the k-ranked element, Aggarwal, 2004
Preserving Privacy for outsourcing aggregation services, Xiong, 2007
Privacy Preserving Indexing of Documents on the Network, Bawa, 2003
K-th element (Aggarwal 04)

Input
- $D_i$, $i = 1, 2, \ldots, s$
- $k$
- Range of data values: $[\alpha, \beta]$
- Size of the union of database: $n$

Output
- The $k$-th ranked elements in the union of $D_i$

Secure computation of the $k$-ranked element, Aggarwal, 2004
kth element protocol

- Initialize
  - Each party ranks its elements in ascending order. Initialize current range \([a, b]\) to \([\alpha, \beta]\), set \(n = \text{sum } |D_i|\)

- Repeat until done
  - Set \(m = (a+b)/2\)
  - Each party computes \(l_i\), number of elements less than \(m\), and \(g_i\), number of elements greater than \(m\)
  - If \(\text{sum}(l_i) \leq k-1\) and \(\text{sum}(g_i) \leq n-k\), done
  - If \(\text{sum}(l_i) \geq k\), set \(b = m-1\), output 0
  - If \(\text{sum}(g_i) \geq n-k+1\), set \(a = m+1\), output 1
Cost

- Number of rounds: $\log M$ where $M$ is the range size
- Each round requires two secure sums and two secure comparisons
Multi-round protocols

- Can we get away from cryptographic primitives?
- Multi-round protocols idea
  - Use randomizations (random response)
  - Utilize inherent network anonymity of multiple nodes
- Multi-round protocols
  - May not be completely secure
  - May not be completely accurate
Multi-round protocols

- Multi-round probabilistic protocols for max/min and top k (Xiong ‘07)
- Multi-round OR (union) protocol (Aggarwal ’04)
Protocol Structure

- Random response (Warner 1965)
- Multi-round randomized protocol
  - Randomized local computation
  - Multi-node anonymity
- Assumption: semi-honest model

Preserving Privacy for outsourcing aggregation services, Xiong, 2007
A Naïve Max/Min Protocol

\[ g_{i-1} \geq v_i \quad g_{i-1} < v_i \]

\[
\begin{array}{|c|c|}
\hline
& g_{i-1} \geq v_i & g_{i-1} < v_i \\
\hline
g_i & g_{i-1} & v_i \\
\hline
\end{array}
\]

- Add in randomization – how, when, and how much?
Max Protocol – Random response

Random response at node $i$:

\[
\begin{array}{|c|c|}
\hline
\text{gi-1} & \text{gi} \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|}
\hline
\text{gi(r)} & \text{gi-1(r)} \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|}
\hline
\text{gi-1} \geq vi & \text{gi-1(r)} < vi \\
\hline
\end{array}
\]

w/ prob Pr:
random number
w/ prob 1-Pr:
$v_i$
Max Protocol – multi-round random response

- Multiple rounds
- Randomization Probability at round $r$:
  - $Pr(r) = P_0 \times d^{r-1}$

- Local algorithm at round $r$ and node $i$:

\[
\begin{array}{c|c|c}
  g_{i-1}(r) & g_{i-1}(r) & \text{w/ prob Pr:} \\
  \text{g}_i(r) & \text{rand} [g_{i-1}(r), v_i] & \text{w/ prob 1-Pr:} \\
  \hline
  g_i(r) & g_{i-1}(r) & v_i
\end{array}
\]
Max Protocol - Illustration
PrivateTopK Protocol

\[ G_i'(r) = \text{topk}(G_{i-1}(r) \cup V_i); \]
\[ V_i' = G_i'(r) - G_{i-1}'(r); \]
\[ m = |V_i'|; \]
\[ \text{if } m=0 \text{ then} \]
\[ G_i(r) = G_{i-1}(r); \]
\[ \text{else} \]
\[ \text{with probability } 1-Pr(r): \]
\[ G_i(r) = G_{i-1}(r); \]
\[ \text{with probability } Pr: \]
\[ G_i(r)[1:k-m] = G_{i-1}(r)[1:k-m] \]
\[ G_i(r)[k-m+1:k] = \text{a sorted list of } m \]
\[ \text{random values generated from} \]
\[ [\min(G_i'(r)[k]-\delta, G_{i-1}(r-1)[k-m+1]), \]
\[ G_i'(r)[k]) \]
end
Min/Max Protocol - Correctness

Precision bound:
- Converges with \( r \)
  \[
  1 - \prod_{j=1}^{r} \Pr(j) \geq 1 - P_0^r \cdot d \cdot \frac{r(r-1)}{2}
  \]
- Smaller \( p_0 \) and \( d \) provides faster convergence
Min/Max Protocol - Cost

Communication cost
- single round: $O(n)$
- Minimum # of rounds given precision guarantee $(1-e)$:
Min/Max Protocol - Security

- Probability/confidence based metric: $P(C|IR,R)$
- Different types of exposures based on claim
  - Data value: $v_i = a$
  - Data ownership: $V_i$ contains $a$

- **Loss of Privacy** ($\text{LoP}$) = $| P(C|IR,R) - P(C|R) |$

- Information entropy based metric:
  - Loss of privacy as a measure of randomness of information: $H(D|R) - H(D|IR,R)$
Min/Max Protocol – Security (Analysis)

- Upper bound for average expected LoP:
  \[ \max_r \frac{1}{2^{r-1}} \times (1 - P_0 \times d^{r-1}) \]
- Larger \( p_0 \) and \( d \) provides better privacy
Min/Max Protocol – Security (Experiments)

- Loss of privacy decreases with increasing number of nodes
- Probabilistic protocol achieves better privacy (close to 0)
- When \( n \) is large, anonymous protocol is actually okay!
Union

- Commutative encryption based approach
  - Number of rounds: 2 rounds
  - Each round: encryption and decryption
- Multi-round random-response approach?
Vector

Each database has a boolean vector of the data items

Union vector is a logical OR of all vectors

Privacy Preserving Indexing of Documents on the Network, Bawa, 2003
Group Vector Protocol

Processing of $V_G'$ at $p_s$ of round $r$

- $P_{ex}=1/2^r$, $P_{in}=1-P_{ex}$
- for $i=1; i<L; i++$
  - if ($V_s[i]=1$ and $V_G'[i]=0$)
    - Set $V_G'[i]=1$ with prob. $P_{in}$
  - if ($V_s[i]=0$ and $V_G'[i]=1$)
    - Set $V_G'[i]=0$ with prob. $P_{ex}$

$r=1, P_{ex}=1/2, P_{in}=1/2$

$r=2, P_{ex}=1/4, P_{in}=3/4$
Open issues

- Tradeoff between accuracy, efficiency, and security
  - How to quantify security
  - How to design adjustable protocols
- Can we generalize the algorithms for a set of operators based on their properties
  - Operators: sum, union, max, min …
  - Properties: commutative, associative, invertible, randomizable