Using Fully Homomorphic Encryption for Statistical Analysis of Categorical, Ordinal and Numerical Data

Wen-jie Lu, Shohei Kawasaki, Jun Sakuma

1. University of Tsukuba, Japan
2. JST CREST
3. RIKEN Center for AIP
Statistical Analysis on the Cloud

Cloud computing is useful for statistical analysis

- Gather distributed data, and reduce hardware cost.
- Minimal interactions between data providers and the cloud.
- The cloud does most of the work for the analyst.
Cloud Computing with Sensitive Data

• Using outside cloud servers raises privacy concerns.
  ○ E.g, medical records, federal data.
• We want to calculate statistics on the cloud while keeping the data secret.
Secure Multiparty Computation (SMC)

- Off-the-shelf tools for SMC protocols
  - Yao’s garbled circuit (GC).
  - Fully homomorphic encryption (FHE).
- But development cost and efficiency hinder applications of GC and FHE in the cloud.

GC on the Cloud Environment

GC requires a large development cost

- Multiple servers are needed.
  - Assume no collusion between servers.
- Fast network is necessary for computation.
  - E.g., 10Gbps bandwidth.
FHE on the Cloud Environment

- Less development cost
  - Single server is enough.
  - Rapid network is not necessary.
- But might be inefficient in practice
  - Encrypt bits one by one.
  - 1~10 ms per evaluation.
  - 1~10 megabytes per ciphertext.

Observation

• Purpose of encrypting bits separately
  o To evaluate any Boolean function.

• But to do statistical analysis, we can use
  o matrix arithmetic operation.
  o comparison operation.
Our Result

• Two new FHE-based primitives:
  o *Matrix Operations*
  o *Batch Greater-than*

• Secure statistical protocols:
  o histogram (count),
  o order of counts,
  o *contingency table* (with cell-suppression),
  o percentile,
  o principal component analysis (PCA),
  o linear regression.

• Source codes: https://github.com/fionser/CODA
Preliminaries: Fully Homomorphic Encryption

• Public-private key scheme.
  o Data providers & cloud share the public key.
  o The analyst holds the private key.

• Allow addition (subtraction) and multiplication on encrypted integers.
  o Analogy: black box with gloves

Preliminaries: Packing (Batching)

- Enable to encrypt and process **vectors** at no extra cost.

Single homomorphic operation

- Fewer ciphertexts
- Faster computation

Preliminaries: Slot Manipulation

*Rotate* slots of the encrypted vector.

![Diagram showing rotation of slots]

*Replicate* a specific slot.

![Diagram showing replication of slots]
Part II Technical Details

• Data preprocessing.
• Efficient matrix multiplication on ciphertexts.
• Comparing two encrypted integers.
• Example of two protocols:
  o Contingency table with cell-suppression
  o Linear regression
    (for other protocols, refer to our paper).
Data Preprocessing

• Numerical data: fixed-point representation
  o $3.14159 \rightarrow \lceil 3.14159 \times 1000 \rceil = 3142$
  o Precision (e.g., 1000) determined in advance

• Categorical data: 1-of-k representation
  o Gender (i.e., $k = 2$). Female $\rightarrow [1, 0]$ and Male $\rightarrow [0, 1]$

• Ordinal data: stair-case encoding
Proposed Matrix Primitive

• Used for adding & multiplying encrypted matrices
• Encrypt each row separately by packing.
  o Row-wise encryption.
  o Horizontally partitioned data
• Efficient and layout consistent.
  o $O(N^2)$ homomorphic operations.
Matrix Multiplication[1/2]

- Encrypt the matrix row by row with packing.

\[
\begin{bmatrix}
1 & 2 \\
3 & 4
\end{bmatrix} \times \begin{bmatrix} a & b \\
c & d \end{bmatrix} = \begin{bmatrix} 1a + 2c & 1b + 2d \\
3a + 4c & 3b + 4d \end{bmatrix}
\]
Matrix Multiplication[1/2]

• Encrypt the matrix row by row with packing.

\[
\begin{bmatrix}
1 & 2 \\
3 & 4 \\
\end{bmatrix} \times
\begin{bmatrix}
a & b \\
c & d \\
\end{bmatrix} =
\begin{bmatrix}
1a + 2c & 1b + 2d \\
3a + 4c & 3b + 4d \\
\end{bmatrix}
\]

• \(N^2\) replications, multiplications and additions
  ○ \(O(N^2)\) complexity compared to \(O(N^3)\) (no packing).

• Also row-wisely encrypted resulting matrix.
Matrix Multiplication[2/2]

• Layout consistency is important for developing efficient statistical protocols.
  o Statistical algorithms need iterative matrix multiplications

Efficient for single multiplication

No

Layout consistent ??

Yes

Efficient for iterative multi.

Heavy layout adjustment

Inefficient for iterative multi.
Experimental Settings of Matrix Primitive

- **Implementations:**
  - FHE: HElib (C++ based)
  - GC: ObliVM (java based)
- **Evaluated on 32-bit integers**
- **Networks:**
  - LAN (about 88 Mbps)
  - WAN (about 48 Mbps)

Evaluation of Matrix Primitive

- When do iterative multiplications, FHE-based primitive can offer better performance.
  - Save communication cost between each iteration
Greater-than (GT) Primitive

\[
\text{GT}(e(x), e(y)) \rightarrow e(x > y) \text{ s.t. } 0 \leq x, y \leq D
\]

- [Golle06] based on Paillier cryptosystem:
  \[\text{if } x > y \text{ then } \exists k \in [1, D] \rightarrow x - y - k = 0\]
- Combination with packing gives great improvements:
  \[e([x, \ldots, x]) - e([y, \ldots, y]) - [1, 2, \ldots, D] \rightarrow e(\eta)\]
  \[\text{Replicated } D \text{ times}\]
  - \[0 \in \eta \iff x > y\] (i.e., decryption is needed)
  - Complexity from \(D\) to \([D/\ell]\).

Experimental Settings for GT Primitive

• Implementations:
  o FHE: HElib (C++ based)
  o GC : ObliVM (java based)
• Domain $D = 2^4 \sim 2^{24}$
• Number of slots $\ell \approx 1700$.
• Networks:
  o LAN (about 88 Mbps)
  o WAN (about 48 Mbps)

Evaluation of Greater-than Primitive

Works for small domains, which is enough for ordinal statistics.
Secure Statistical Protocols

• Contingency table with cell-suppression protocol:
  o Use the greater-than primitive.
  o One round protocol between cloud and analyst.

• Linear regression protocol:
  o Use the matrix primitive.
  o Two rounds protocol.
  o Use a Plaintext Precision Expansion technique (discuss it latter).
Contingency Table

<table>
<thead>
<tr>
<th>Gender</th>
<th>Smoke</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>Smoker</td>
</tr>
<tr>
<td>Female</td>
<td>Non-smoker</td>
</tr>
<tr>
<td>Male</td>
<td>Non-Smoker</td>
</tr>
</tbody>
</table>

Categorical data

- **Indicator encoding:**
  - Male $\rightarrow [1, 0]$, Female $\rightarrow [0, 1]$
  - Smoker $\rightarrow [1, 0]$, Non-smoker $\rightarrow [0, 1]$

- **Basic Idea:** **multiply & rotate**
  - $[a_1, a_2] \times [b_1, b_2]$ counts Male-Smoker, and Female-Nonsmoker
  - $[a_1, a_2] \times ([b_1, b_2] >> 1) = [a_1, a_2] \times [b_2, b_1]$ gives other two counts.

- Improvement with no extra preprocessing
  - $O(\max(k_1, k_2)) \Rightarrow O(\log k_1 k_2)$. 

\[K_1 = 2\]

\[K_2 = 2\]
Contingency Table: Cell Suppression

<table>
<thead>
<tr>
<th></th>
<th>Smoker</th>
<th>Non-smoker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>20</td>
<td>11</td>
</tr>
<tr>
<td>Female</td>
<td>3</td>
<td>12</td>
</tr>
</tbody>
</table>

- Protect the privacy of rare individuals.
- Given a ciphertext $e(x)$, to compute $e(y)$ where if $x > \text{threshold}$ then $y = x$ else $y =$ some random value.
- $GT(e(x), \text{threshold}) = e(\eta)$. iff $x > \text{threshold}$, then $0 \in \eta$.
- To compute $\{e(x + r), e(\eta + r), e(\eta \times r')\}$
  - Non-zero random vectors $r, r'$
  - If $0 \in \eta$, we have $0 \in \eta \times r'$, then we can get $r$ and know $x$. 

Origin Table

<table>
<thead>
<tr>
<th></th>
<th>Smoker</th>
<th>Non-smoker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>20</td>
<td>11</td>
</tr>
<tr>
<td>Female</td>
<td>3</td>
<td>12</td>
</tr>
</tbody>
</table>

Suppressed Table

<table>
<thead>
<tr>
<th></th>
<th>Smoker</th>
<th>Non-smoker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>20</td>
<td>11</td>
</tr>
<tr>
<td>Female</td>
<td>0</td>
<td>12</td>
</tr>
</tbody>
</table>
Contingency Table Performance Evaluation

- Complexity increases logarithmically with the table sizes.
- Most of the work (>90%) done by the cloud.

#records = 4000
Linear Regression (LR)

- From data \( \{(x_i, y_i)\}_i \), computes a model \( w \) s.t.
  \[
  w = (X^T X)^{-1} X^T y
  \]

- The inversion of an encrypted matrix.

**Division-free Matrix Inversion \((Q, \lambda)\):**

Set \( A^{(1)} = Q, R^{(1)} = I, a^{(1)} = \lambda \), and iterate

\[
\begin{align*}
R^{(t+1)} &= 2a^{(t)} R^{(t)} - R^{(t)} A^{(t)} \\
A^{(t+1)} &= 2a^{(t)} A^{(t)} - A^{(t)} A^{(t)} \\
ap^{(t+1)} &= a^{(t)} a^{(t)}
\end{align*}
\]

[Guo06] \( R^{(t)} \) gives a good approximation to \( \lambda^{2t} Q^{-1} \) if \( \lambda \) is close to largest eigenvalue of \( Q \) (use PCA to compute \( \lambda \)).

Plaintext Precision Expansion (PPE)

• Division-free algorithms introduce large integers. \((\lambda^{2t})\)
  o But the current FHE library allows at most 60-bit integers.
• Allows division-free algorithms without changing the FHE library.
• Uses \(K\) different FHE parameters (each \(b\)-bit < 60)
  o Achieves an equivalent \(Kb\)-bit parameter.
  o Increases the time by \(K\) times, but naturally parallelizable.
• Direct application of the Chinese Remainder Theorem.
Experiments: Linear Regression

- Negligible decryption time (less than 2 s).
- 20x faster than previous FHE solution [Wu et al. 12]
  - 5 dimensions (400+ mins).
- Good scalability (reduced execution using more cores).
Summary

• Secure statistical analysis in the cloud with multiple data providers.

• Two primitives
  o Matrix operation and greater-than

• Two protocols.
  o Contingency table and linear regression.

• Encoding and packing can improve FHE's balance between generality and efficiency.