Secure Computation System
– How it works

2018.8
NTT Secure Platform laboratories
Secure computation

- Secure Computation
- Benefits
- History
Secure Computation

• computes while keeping data encrypted

<table>
<thead>
<tr>
<th></th>
<th>data encrypted process</th>
</tr>
</thead>
<tbody>
<tr>
<td>typical encryption</td>
<td>transformation and storage</td>
</tr>
<tr>
<td>secure computation</td>
<td>transformation and storage</td>
</tr>
</tbody>
</table>
Benefits

- keep the data secret except for results
- a novel “inter-organization” data analysis

Data from company A
Data from company B
Data from company C

Data is never disclosed among companies
Data is never leaked even to the system
Only the result can be accessed

Secure Computation System
Computation while keeping data encrypted
Analysis results

Personal Data, Trade Secret, …
History

• Cryptology “Secure multi-party computation” has been studied since 1980’s
• Practical issue is the performance
  • too slow!
• More modern studies are made for speed-up and implementation
• Industries start to pay attention for Secure Computation as a new data utilization method
Secure Computation based on Secret Sharing

- Secret Sharing as an encryption mechanism
- Multi-party Computation based on Secret Sharing
- Security Condition

- Secure Computation denotes Secret Sharing based Secure Computation in this document
Secret Sharing

• Secret Sharing as an encryption mechanism

Secret Sharing is to protect data confidentiality by dividing data into pieces called "shares"
1. Individual shares are no use
2. If some shares are lost, data can be recovered

• We use ISO conformance Secret Sharing

ISO/IEC 19592-2:
Information technology - Security techniques - Secret sharing - Part 2: Fundamental mechanisms
NTT contributed to this ISO as editors
Multi-party Computation

- We use multi-party Computation based on Secret Sharing

Multi-party Computation
1. performs data operations and exchange among multiple servers according to defined procedures
2. data is processed in an encrypted fashion called share
Security Condition

• One share has no information
  
  Single server can not reconstruct the data or results

• Two shares can be enough for reconstruction
  
  Prevention of data acquisition from two servers is the security condition
  • compatible with typical encryption (right table)

<table>
<thead>
<tr>
<th>Condition for data reconstruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>typical encryption</td>
</tr>
<tr>
<td>ciphertext and key</td>
</tr>
<tr>
<td>secure computation</td>
</tr>
<tr>
<td>two shares</td>
</tr>
</tbody>
</table>
System Model

- Client-Multiservers Model
- Data Registration
- Computation (data analysis)
Client-Multiservers Model

- Multiple servers work together to perform Secure Computation
Data Registration

- Client protects data as shares and register them with each server
Computation (data analysis)

- Client requests computation to each server and obtains results from them

1. “Calculate the average of income”
   - Multi-party computation
   - Shares of the value xxxx are returned
   - Client reconstructs the value from shares

Client

Multiservers
How it works

• Secret Sharing
• Addition on Secret Sharing
  ADVANCED
• Multiplication on Secret Sharing
  ADVANCED
• Secure Sort
Secret Sharing

“2” into two shares

Secret Sharing

Throw the dice

The number and the reverse are shares

Reconstruction

Forward 5 from 7

Reverse the number from the secret

Single Share (the number of dice, reverse of the roulette) is no use
Addition on Secret Sharing

“2+3=5” Calculation

Alice

Share 1

Alice

5

7

Add share 1s

“Share” the secret

Bob

Share 2

Bob

2

1

Add Share 2s

Reconstructing results

Forward 7 from 8

Share 1

Share 1

5

2

7

Share 2

Share 2

7

1

8

© 2018 NTT corp. 16
Multiplication on Secret Sharing (1/2)

“2” into three shares

Secret Sharing

1. Throw two dice
2. Reverse two numbers from the secret

Select share-pair out of three*

Reconstruction

Get two numbers from two share pair

Forward (5+3) steps from 4

* three shares: two numbers of dice and reverse of the roulette

© 2018 NTT corp. 17
Multiplication on Secret Sharing

“2x3=6” Calculation

Evaluate the formula below with each pair of share
\[ ab = (a_0 + a_1 + a_2)(b_0 + b_1 + b_2) = a_0b_0 + a_0b_1 + a_1b_0 + a_1b_1 + a_2b_1 + a_2b_2 + a_2b_0 + a_0b_2 \]

Forward 3 results steps (ignoring the ten’s place)

Reconstructing results

IGNORING THE TEN’S PLACE

Forward (7+6) steps from 3
(ignoring the ten’s place)
Secure Sort

Using “radix sort” with our “secure one-bit-sort”

Secure one-bit-sort

Outline
Calculate the numerical order for each input while keeping data encrypted

<table>
<thead>
<tr>
<th>input</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

Reconstruct only the order and sort the encrypted input

Detailed

1. NOT of input

<table>
<thead>
<tr>
<th>input</th>
<th>not</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

2. Sub total of series of not and input

<table>
<thead>
<tr>
<th>not</th>
<th>sum1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

3. Multiply not/input by mul1/mul2

<table>
<thead>
<tr>
<th>not</th>
<th>sum1</th>
<th>mul1</th>
<th>mul2</th>
<th>add=order</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

4. Add mul1 and mul2

<table>
<thead>
<tr>
<th>mul1</th>
<th>mul2</th>
<th>add=order</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Boxed numbers are all encrypted
Performance of Secure Computation

• Does it work?
• Performance
• Behind our exclusive performance
Q: I heard that secure computation is too slow to use?

A: No, we have achieved highly efficient data processing in secure computation.

<table>
<thead>
<tr>
<th>functions</th>
<th>execution time</th>
<th>10 million times</th>
</tr>
</thead>
<tbody>
<tr>
<td>addition</td>
<td>0.014 sec</td>
<td></td>
</tr>
<tr>
<td>multiplication</td>
<td>0.473 sec</td>
<td></td>
</tr>
<tr>
<td>sort</td>
<td>12.2 sec</td>
<td>10 million records</td>
</tr>
</tbody>
</table>

12 seconds for secure sort 10 million records
about 1 second for plain sort (with single thread)

The performance gap between our secure computation and ordinary plain one reaches one to ten level
## Performance

Execution time of typical function of our system

<table>
<thead>
<tr>
<th>functions</th>
<th>number of data</th>
<th>$10^3$</th>
<th>$10^4$</th>
<th>$10^5$</th>
<th>$10^6$</th>
<th>$10^7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>Multiplication</td>
<td></td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>39</td>
<td>473</td>
</tr>
<tr>
<td>Sort</td>
<td></td>
<td>10</td>
<td>23</td>
<td>133</td>
<td>1,274</td>
<td>12,255</td>
</tr>
<tr>
<td>Sum Total</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>Sum of Products</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>Quantity Table</td>
<td></td>
<td>22</td>
<td>46</td>
<td>255</td>
<td>2,252</td>
<td>22,676</td>
</tr>
<tr>
<td>Shuffle</td>
<td></td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>60</td>
<td>731</td>
</tr>
<tr>
<td>Table Join</td>
<td></td>
<td>19</td>
<td>65</td>
<td>518</td>
<td>4,965</td>
<td>53,205</td>
</tr>
<tr>
<td>Data Filter with prefix match</td>
<td></td>
<td>6</td>
<td>6</td>
<td>14</td>
<td>91</td>
<td>813</td>
</tr>
<tr>
<td>Data Filter with numerical data</td>
<td></td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>35</td>
<td>413</td>
</tr>
</tbody>
</table>

PC 3台 (CPU: Intel Core i7 6900K, MEM: 32GB, SSD: 525GB, OS: CentOS 7.2) with 10Gbps networks
Behind our exclusive performance

• Secret Sharing based
  • small data size
  • dedicated algorithm for addition and multiplication

• Implementations
  We work hard for:
  • secret sharing data procession
  • multi-party communication
  • fast secure sort algorithm
Secure Computation System
San-shi®

• What is San-shi?
• Features of San-shi
• Typical Functions

San-shi® is a trade mark of NTT
What is San-shi?

• NTT’s Secure Computation System
  • consist of server/client software
  • multi-party computation based on secret sharing by 3 or 4 computers

Client Software
• data input and output with secret sharing manner
• analysis requests procession

Server Software
• multi-party computation based on secret sharing
Features of San-shi

Secure data management on secret sharing manner
• Schema definition
• Table creation
• Data registration

Rich secure statistical functions
• Sum, Average, Variance
• Max, Min, Median
• Cross Tabulation

A novel “inter-organization” data analysis while keeping data secret
• Data join
• Cross-sectional data analysis

Practical performance with 10 million x 100 attributes data
## Typical Functions

<table>
<thead>
<tr>
<th>Statistical Functions</th>
<th>管理機能</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Sum</td>
<td>Table management</td>
</tr>
<tr>
<td>Average</td>
<td>Schema management</td>
</tr>
<tr>
<td>Variance</td>
<td>Transaction (for table)</td>
</tr>
<tr>
<td>Sum of Products</td>
<td>Rollback (for table)</td>
</tr>
<tr>
<td>Max</td>
<td>User/Tenant Administration</td>
</tr>
<tr>
<td>Min</td>
<td></td>
</tr>
<tr>
<td>Median</td>
<td></td>
</tr>
<tr>
<td>P-quartile</td>
<td></td>
</tr>
<tr>
<td>Quantity Table</td>
<td></td>
</tr>
<tr>
<td>Frequency Table (cross tabulation, histogram)</td>
<td></td>
</tr>
<tr>
<td>Threshold / Dominance Rule for Frequency Table Output</td>
<td></td>
</tr>
<tr>
<td>t-test</td>
<td></td>
</tr>
<tr>
<td>Kaplan-Meier’s graph</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>データ操作機能</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Update</td>
<td></td>
</tr>
<tr>
<td>Table Join</td>
<td></td>
</tr>
<tr>
<td>NULL Filter</td>
<td></td>
</tr>
<tr>
<td>Duplicate Filter</td>
<td></td>
</tr>
<tr>
<td>Data Filter with date/strings comparison</td>
<td></td>
</tr>
<tr>
<td>Categorization for numerical value</td>
<td></td>
</tr>
<tr>
<td>Shuffle</td>
<td></td>
</tr>
</tbody>
</table>
Related Papers

- Understanding our technology
- Reviewed Paper/Conferences
- Awards
Understanding our technology

Note that some Japanese documents are included.


Understanding our technology

Note that some Japanese documents are included.


9. 五 十 嵐 大, 濱 田 浩 気, 菊 池 亮, 千田 浩 司: "超高速秘密計算ソートの設計と実装：秘密計算がスクリプト言語に並んだ日," コンピュータセキュリティシンポジウム(CSS), 2017 (CSS論文賞).

10. 桐淵 直人, 五 十 嵐 大, 諸 橋 玄 武, 濱 田 浩 気: "属性情報と履歴情報の秘匿統合分析に向けた秘密計算による高速な等結合アルゴリズムとその実装," コンピュータセキュリティシンポジウム(CSS), 2016 (CSS論文賞).

11. 五 十 嵐 大, 菊 池 亮, 高 橋 克 巳: "MEVAL2 vs. CCS Best paper on MPC-AES,"暗号と情報セキュリティシンポジウム(SCIS), 2017 (SCISイノベーション論文賞).


Awards

Note that some Japanese documents are included.

1. [HLI and Baidu Award for 2017 iDASH Genome Privacy & Security Computation] Koki Hamada, Dai Ikarashi, Satoshi Hasegawa, Koji Chida
Thank you!

• version August 2018