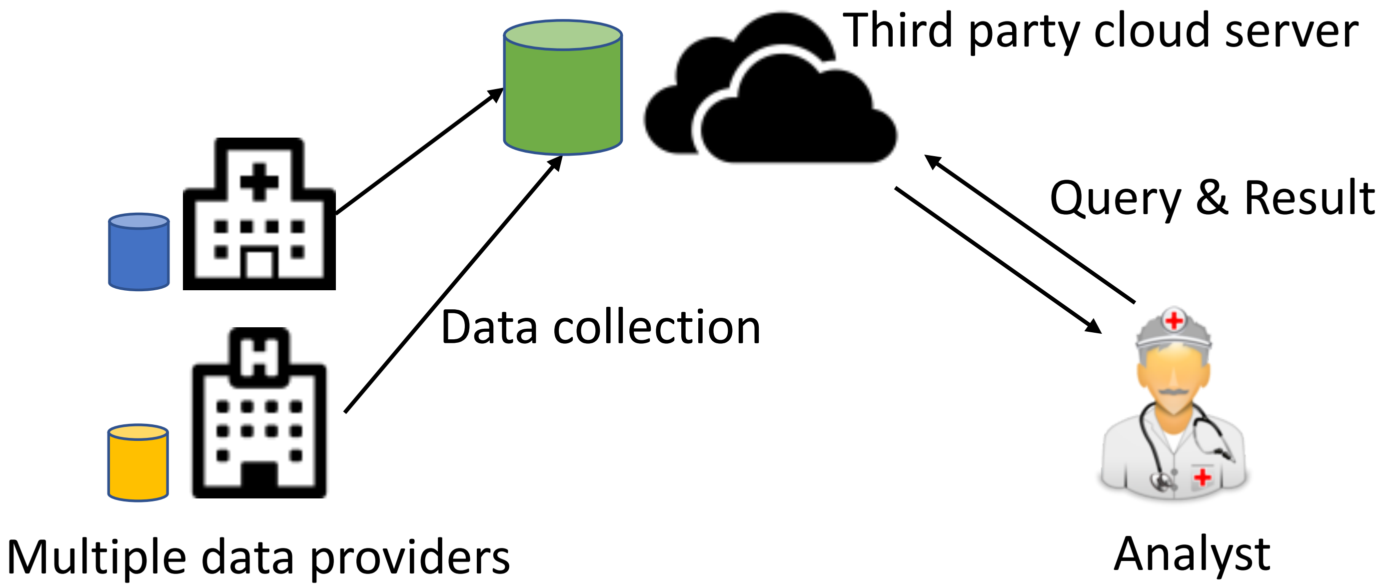
# Using Fully Homomorphic Encryption for Statistical Analysis of Categorical, Ordinal and Numerical Data

**Wen-jie Lu**[1], Shohei Kawasaki[1], Jun Sakuma[1,2,3]

1. University of Tsukuba, Japan
2. JST CREST
3. RIKEN Center for AIP

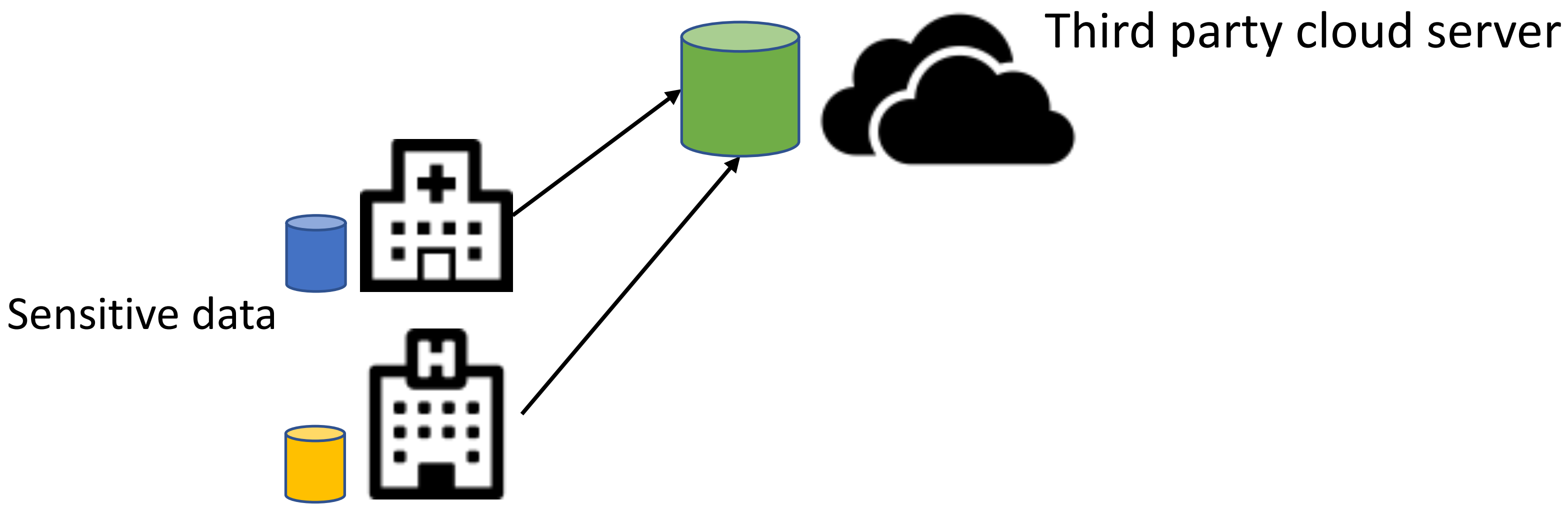
# Statistical Analysis on the Cloud




Cloud computing is useful for statistical analysis

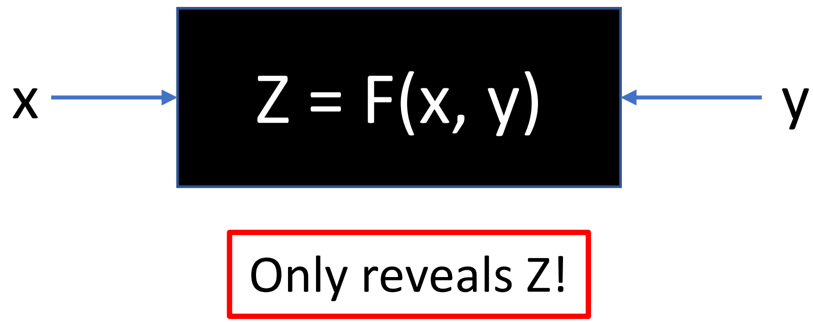
- Gather distributed data, and reduce hardware cost.
- Minimal interactions between data providers and the cloud.
- The cloud does most of the work for the analyst.

# 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)



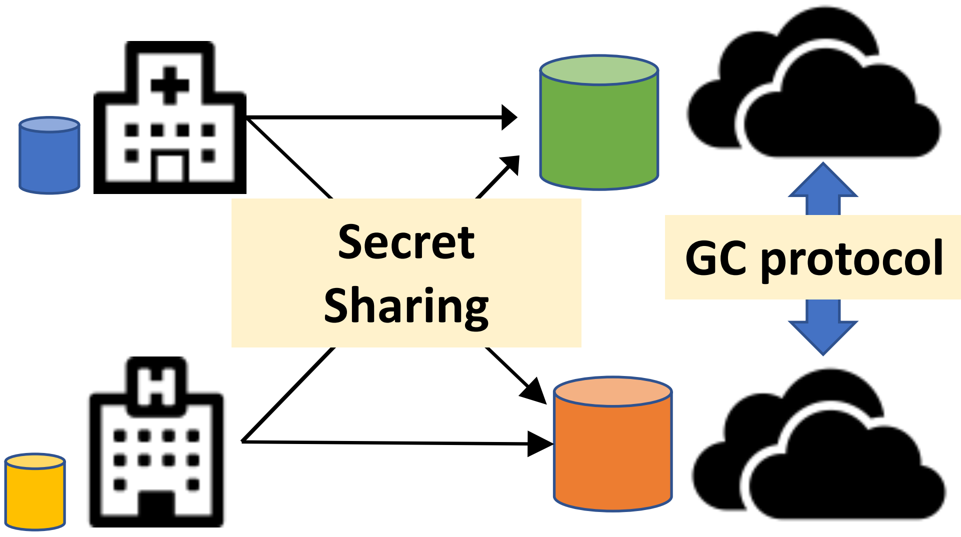

x, y: private input

F: public function

- 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.

Yao Andrew. *Protocols for secure secure computation.* 1982.
Gentry. *Fully homomorphic encryption using ideal lattices*. 2009.

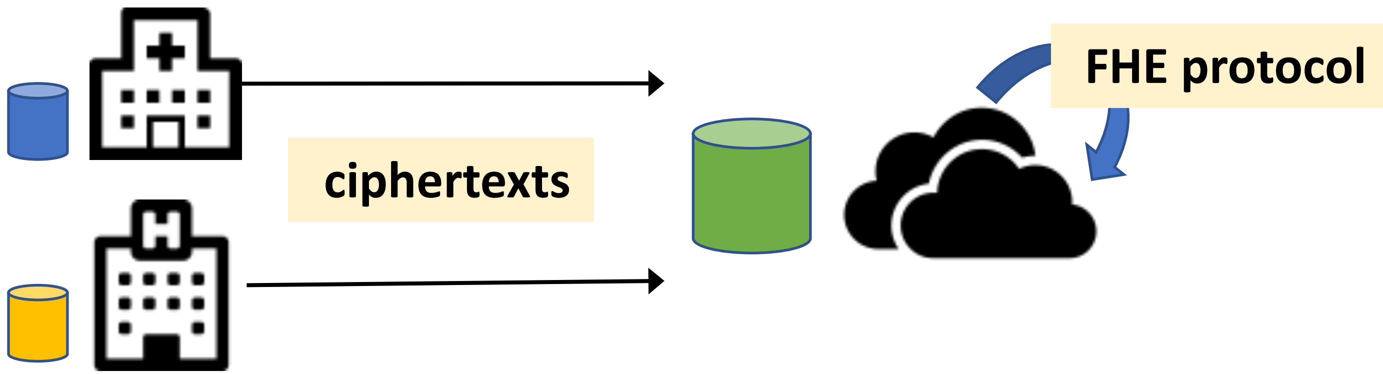
# GC on the Cloud Environment




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.

Gentry et al. *Homomorphic Evaluation of the AES Circuit*. 2012.

# Observation

- Purpose of encrypting bits separately
  - To evaluate any Boolean function.
- But to do statistical analysis, we can use
  - *matrix arithmetic* operation.
  - *comparison* operation.

# Our Result

- Two new FHE-based primitives:
  - *Matrix Operations*
  - *Batch Greater-than*
- Secure statistical protocols:
  - histogram (count),
  - order of counts,
  - contingency table (with cell-suppression),
  - percentile,
  - principal component analysis (PCA),
  - linear regression.
- Source codes: https://github.com/fionser/CODA

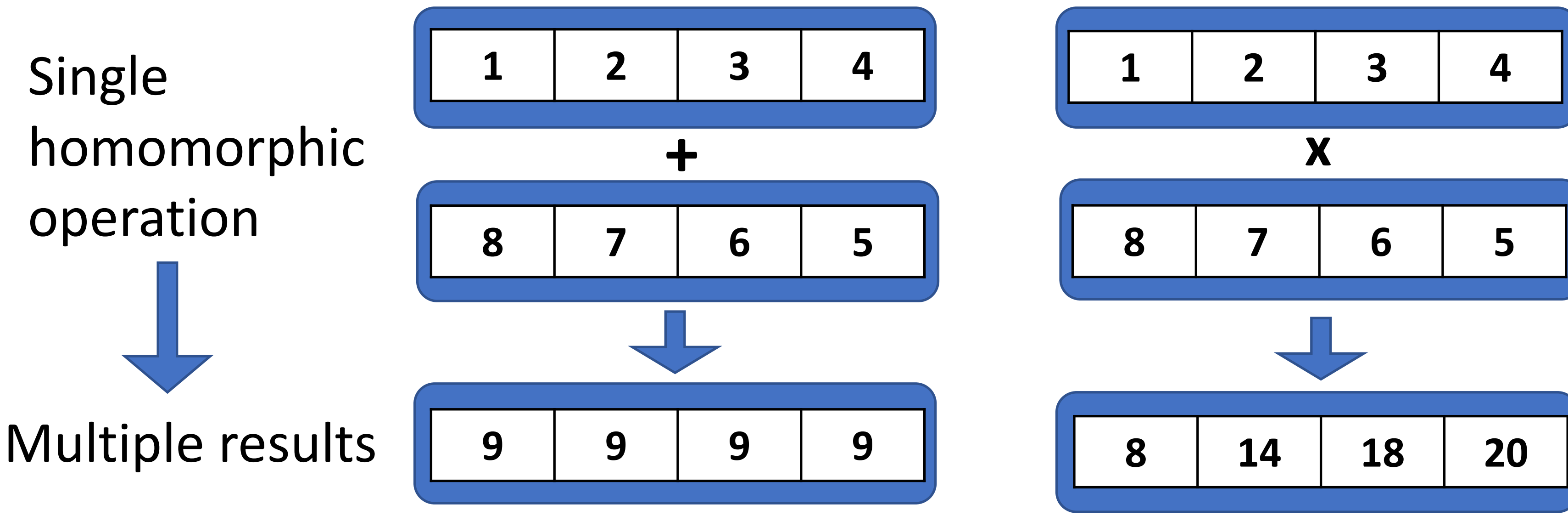
# Preliminaries: Fully Homomorphic Encryption

- Public-private key scheme.
  - Data providers & cloud share the public key.
  - The analyst holds the private key.
- Allow *addition (subtraction) and multiplication* on encrypted integers.
  - Analogy: black box with gloves

Brakerski et al. Fully Homomorphic Encryption without Bootstrapping. 2012.

# Preliminaries: Packing (Batching)

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

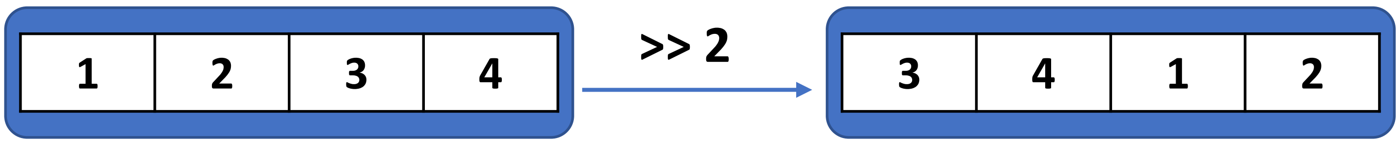



  - Fewer ciphertexts
  - Faster computation

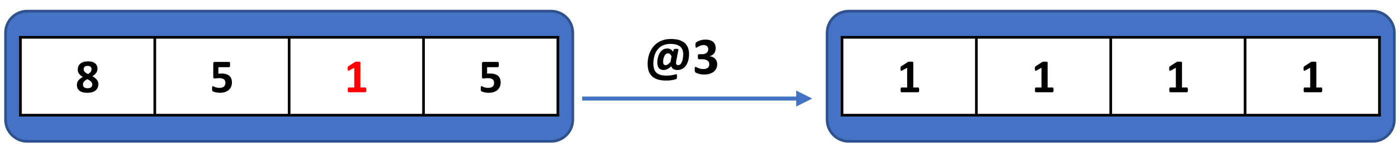
N.P. Smart et al. *Fully homomorphic SIMD operations*. 2011.

# Preliminaries: Slot Manipulation

*Rotate* slots of the encrypted vector.




*Replicate* a specific slot.




Halevi et al. *Algorithms in Helib*. 2014.

# Part II Technical Details

- Data preprocessing.
- Efficient matrix multiplication on ciphertexts.
- Comparing two encrypted integers.
- Example of two protocols:
    - Contingency table with cell-suppression
    - Linear regression

  (for other protocols, refer to our paper).

# Data Preprocessing

- Numerical data: fixed-point representation
  - $3.14159 \rightarrow \lceil 3.14159 \times 1000 \rfloor = 3142$
  - Precision (e.g., 1000) determined in advance
- Categorical data: 1-of-k representation
  - Gender (i.e., k = 2). Female → [1, 0] and Male → [0, 1]
- Ordinal data: stair-case encoding

# Proposed Matrix Primitive

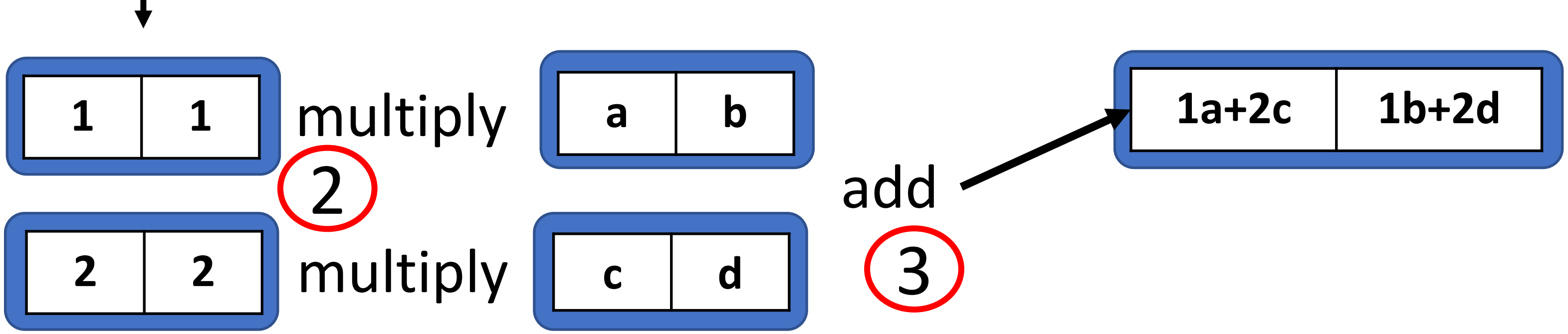
- Used for adding & multiplying encrypted matrices
- Encrypt each row separately by packing.
  - Row-wise encryption.
  - Horizontally partitioned data
- Efficient and layout consistent.
  - $O(N^2)$ homomorphic operations.

# Matrix Multiplication[1/2]

- Encrypt the matrix row by row with packing.

1 Replicate @1 @2

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

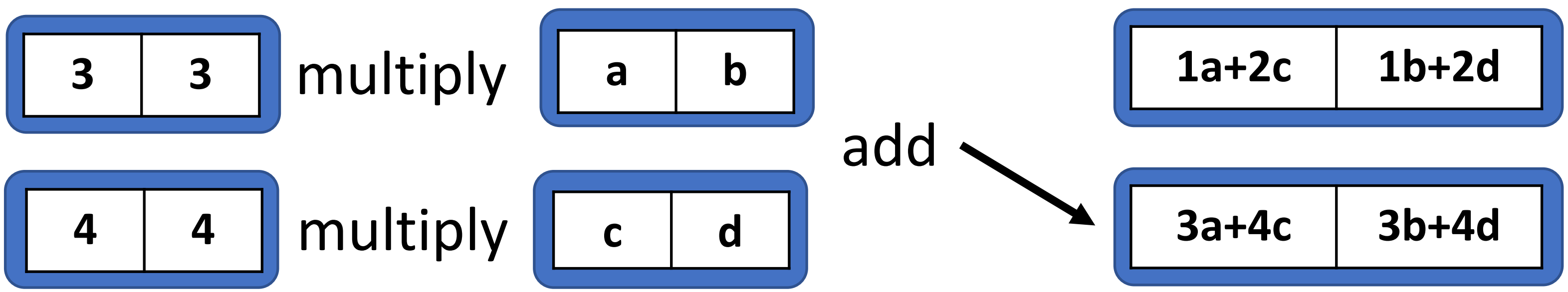


# Matrix Multiplication[1/2]

- Encrypt the matrix row by row with packing.

Replicate
@1 @2

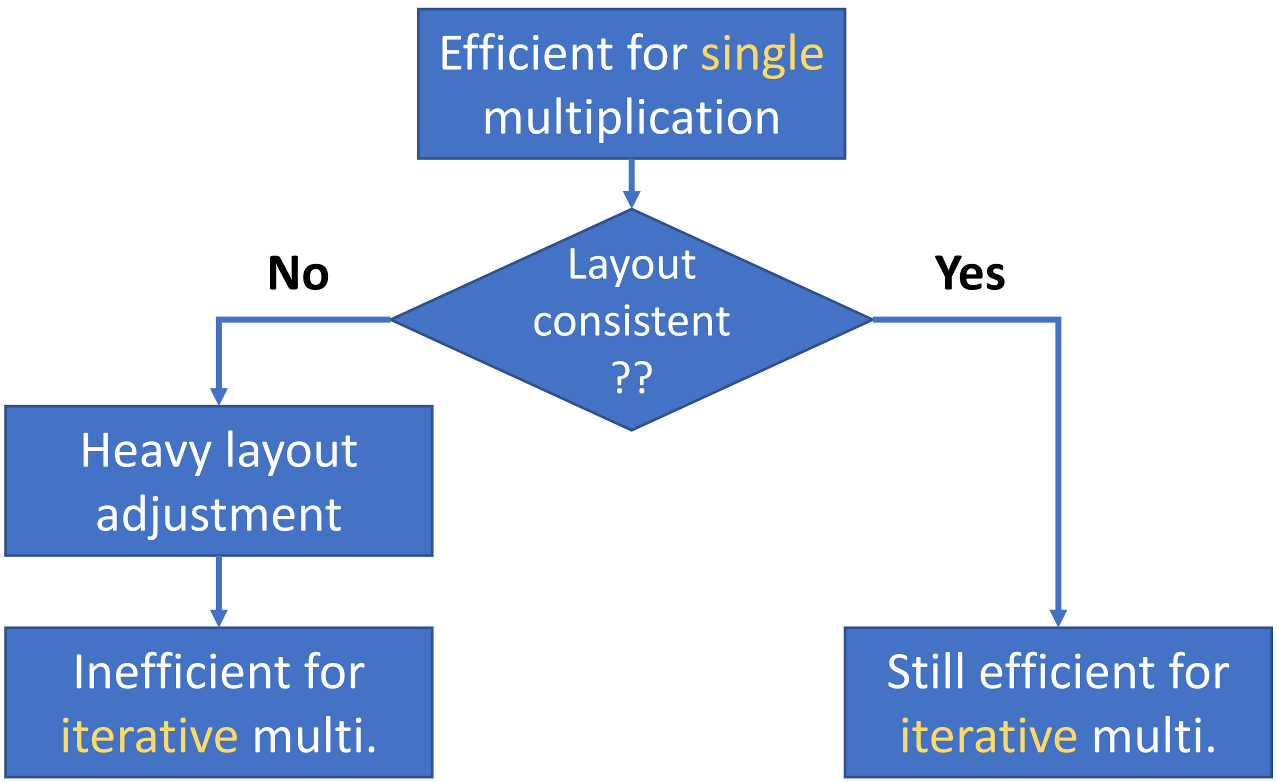
$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1a + 2c & 1b + 2d \\ 3a + 4c & 3b + 4d \end{bmatrix}$$




- $N^2$ 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.
  - Statistical algorithms need iterative matrix multiplications



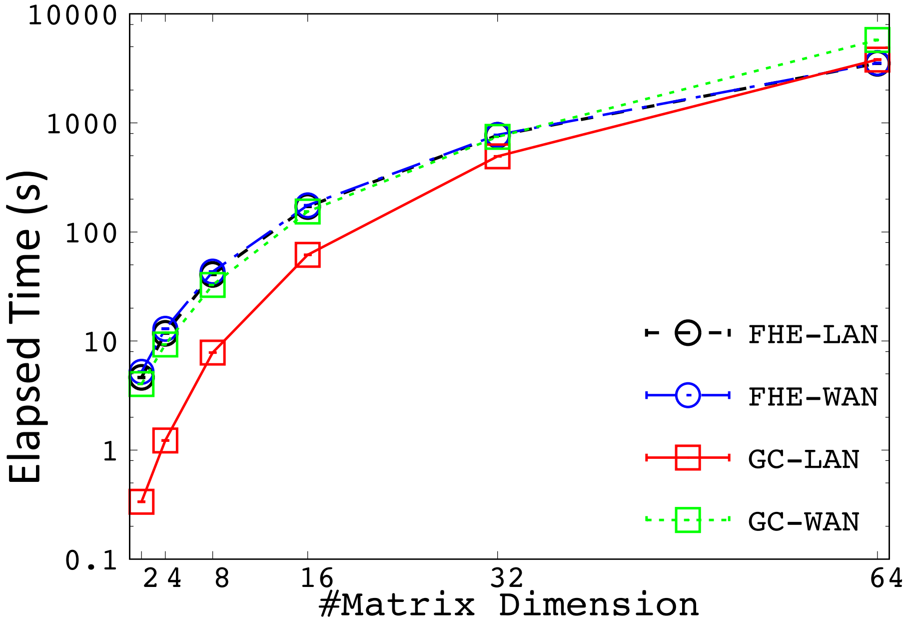
# Experimental Settings of Matrix Primitive

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

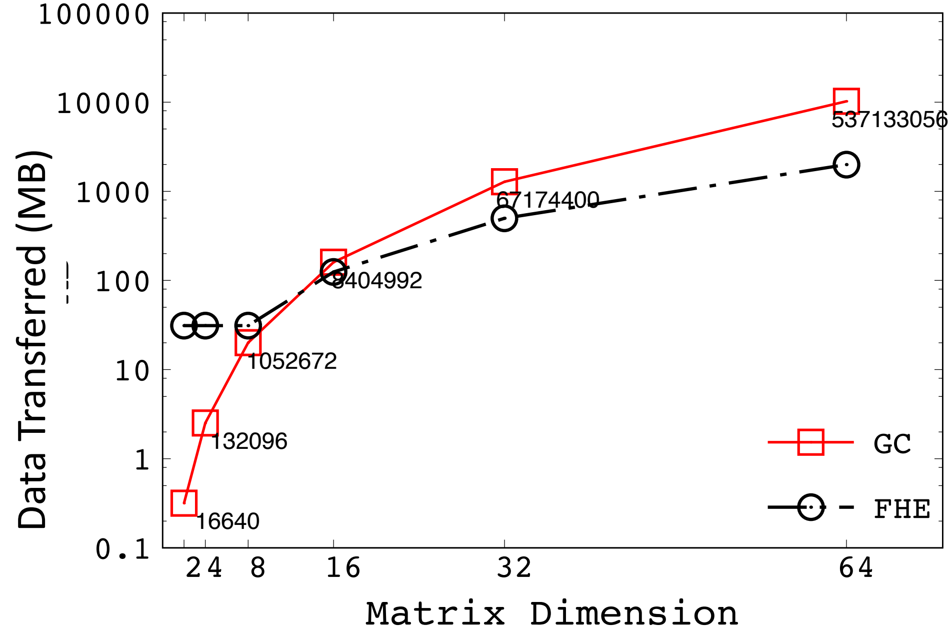
HElib. https://github.com/shaih/HElib.
Liu et al. *ObliVM: A programming framework for secure computation*. 2015.

# Evaluation of Matrix Primitive

**Execution Time**




**Communication Cost**




- When do iterative multiplications, FHE-based primitive can offer better performance.
  - Save communication cost between each iteration

# Greater-than (GT) Primitive

$$\boxed{\mathrm{GT}\big(\mathrm{e}(x), e(y)\big) \rightarrow e(x >^{?} y) \text{ s.t. } 0 \leq x, y \leq \mathrm{D}}$$

- [Golle06] based on Paillier cryptosystem:

$$if\ x > y\ then\ \exists k \in [1, D] \rightarrow x - y - k = 0$$

- Combination with packing gives great improvements:

$$e(\underbrace{[x, \dots, x]}_{\text{Replicated D times}}) - e(\underbrace{[y, \dots, y]}_{\text{Replicated D times}}) - [1, 2, \dots, D] \rightarrow e(\boldsymbol{\eta})$$

  - $0 \in \boldsymbol{\eta} \Leftrightarrow x > y$ (i.e., decryption is needed)
  - Complexity from $D$ to $\lceil \mathrm{D}/\ell \rceil$.

Golle. *A private stable matching algorithm*. 2006.

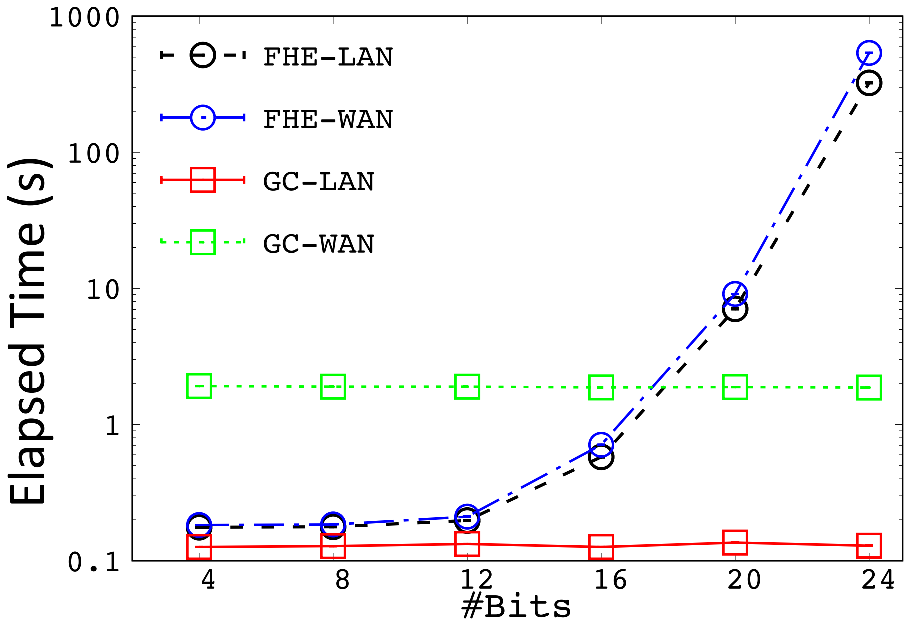
# Experimental Settings for GT Primitive

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

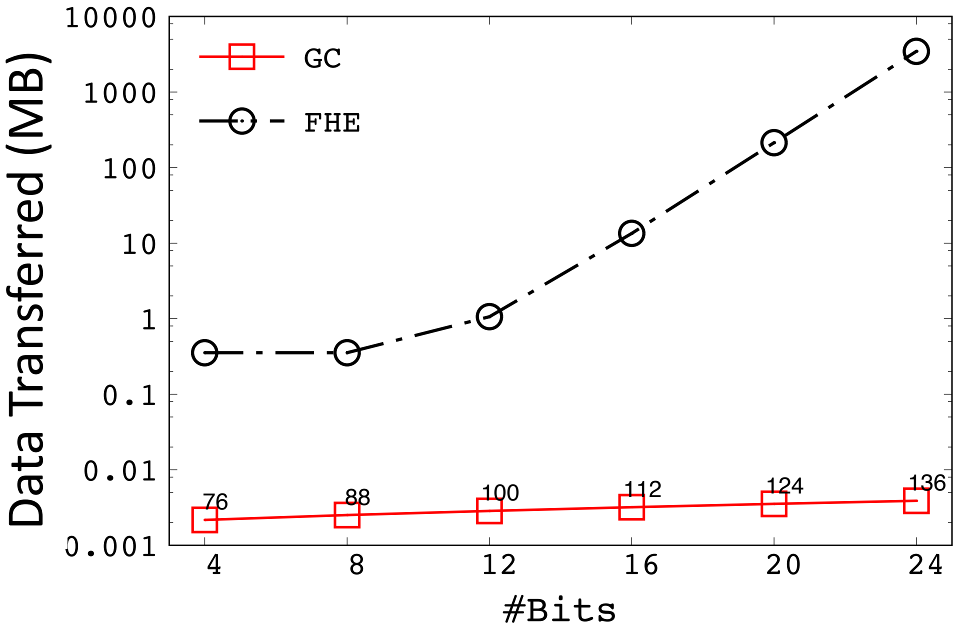
HElib. https://github.com/shaih/HElib.
Liu et al. *ObliVM: A programming framework for secure computation*. 2015.

# Evaluation of Greater-than Primitive

**Execution Time**




**Communication Cost**




Works for small domains, which is enough for ordinal statistics.

# Secure Statistical Protocols

- Contingency table with cell-suppression protocol:
  - Use the greater-than primitive.
  - One round protocol between cloud and analyst.
- Linear regression protocol:
  - Use the matrix primitive.
  - Two rounds protocol.
  - Use a Plaintext Precision Expansion technique (discuss it latter).

# Contingency Table

| Gender | Smoke |
|---|---|
| Male | Smoker |
| Female | Non-smoker |
| Male | Non-Smoker |

Categorical data

$K_1 = 2$

$K_2 = 2$

| | Smoker | Non-smoker |
|---|---|---|
| Male | 1 | 1 |
| Female | 0 | 1 |

Contingency Table

- Indicator encoding:

  Male → [1, 0], Female → [0, 1]

  Smoker → [1, 0], Non-smoker → [0, 1]

- Basic Idea: **multiply & rotate**

  $[a_1, a_2] \times [b_1, b_2]$ counts Male-Smoker, and Female-Nonsmoker

  $[a_1, a_2] \times ([b_1, b_2] \gg 1) = [a_1, a_2] \times [b_2, b_1]$ gives other two counts.

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

# Contingency Table: Cell Suppression

Origin Table

| | Smoker | Non-smoker |
|---|---|---|
| Male | 20 | 11 |
| Female | 3 | 12 |

if < 10

zero out

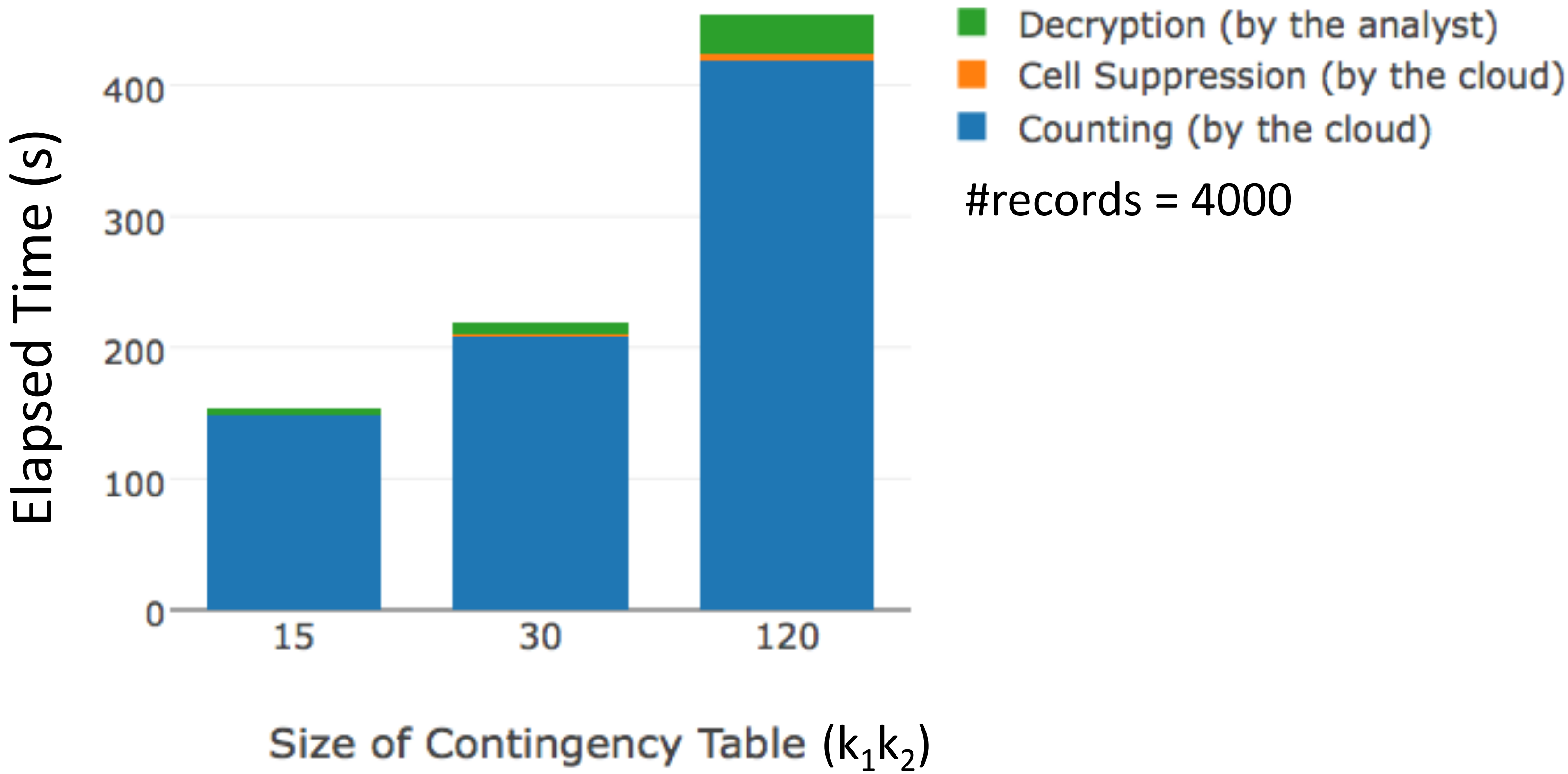
Suppressed Table

| | Smoker | Non-smoker |
|---|---|---|
| Male | 20 | 11 |
| Female | 0 | 12 |

- Protect the privacy of rare individuals.
- Given a ciphertext $e(x)$, to compute $e(y)$ where

  if $x$ > threshold then $y$ = $x$ else $y$ = some random value
- $GT(e(x), \text{threshold}) = e(\boldsymbol{\eta})$. iff $x$ > threshold, then $0 \in \boldsymbol{\eta}$.
- To compute $\{e(x+\boldsymbol{r}), e(\boldsymbol{\eta}+\boldsymbol{r}), e(\boldsymbol{\eta}\times\boldsymbol{r}')\}$
  - Non-zero random vectors $\boldsymbol{r}$, $\boldsymbol{r}'$
  - If $0 \in \boldsymbol{\eta}$, we have $0 \in \boldsymbol{\eta}\times\boldsymbol{r}'$, then we can get $\boldsymbol{r}$ and know $x$.

# Contingency Table Performance Evaluation




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

# Linear Regression (LR)

- From data $\{(\boldsymbol{x}_i, y_i)\}_i$ , computes a model $\boldsymbol{w}$ s.t.

$$\boldsymbol{w} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$$

- The inversion of an encrypted matrix.

*Division-free Matrix Inversion* $(\boldsymbol{Q}, \lambda)$:

set $\boldsymbol{A}^{(1)} = \boldsymbol{Q}, \boldsymbol{R}^{(1)} = \boldsymbol{I}, a^{(1)} = \lambda$, and *iterate*

$$\boldsymbol{R}^{(t+1)} = 2a^{(t)}\boldsymbol{R}^{(t)} - \boldsymbol{R}^{(t)}\boldsymbol{A}^{(t)}$$

$$\boldsymbol{A}^{(t+1)} = 2a^{(t)}\boldsymbol{A}^{(t)} - \boldsymbol{A}^{(t)}\boldsymbol{A}^{(t)}$$

$$a^{(t+1)} = a^{(t)}a^{(t)}$$

Layout consistency leads to efficient iterative protocols.

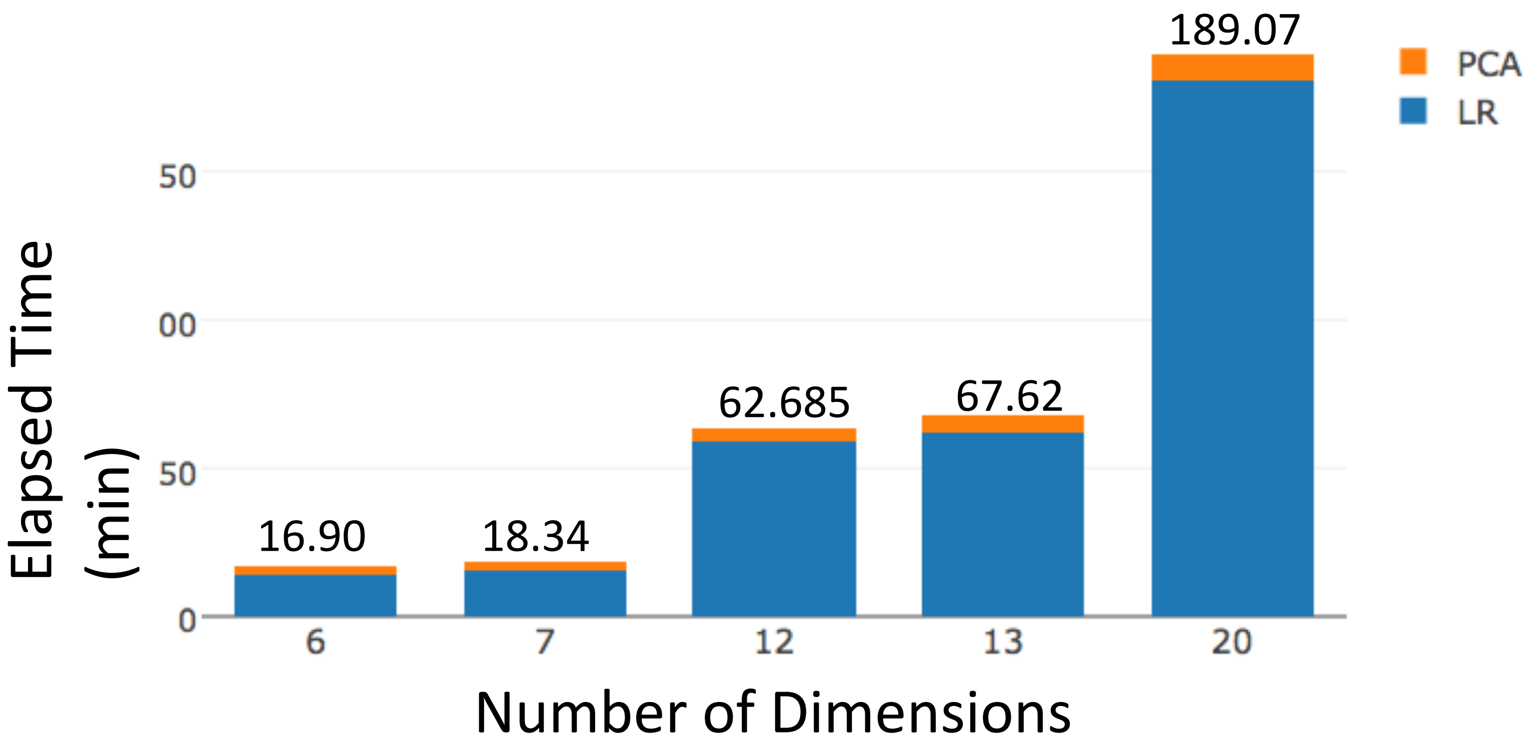
[Guo06] $\boldsymbol{R}^{(t)}$ gives a good *approximation* to $\lambda^{2^t}\boldsymbol{Q}^{-1}$ if $\lambda$ is close to largest eigenvalue of $\boldsymbol{Q}$ (use PCA to compute $\lambda$).

Guo et al. *A Schur-Newton method for the matrix pth root and its inverse*. 2006.

# Plaintext Precision Expansion (PPE)

- Division-free algorithms introduce large integers. ($\lambda^{2^t}$)
    - 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)
    - Achieves an equivalent $Kb$-bit parameter.
    - 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
    - Matrix operation and greater-than
- Two protocols.
    - Contingency table and linear regression.
- Encoding and packing can improve FHE's balance between generality and efficiency.