

# Efficient Secure Outsourcing of Genome-wide Association Studies

Wenjie Lu<sup>1</sup>, Yoshiji Yamada<sup>2</sup>, Jun Sakuma<sup>1,3</sup>



1 Dept. of Computer Science, Univ. of Tsukuba



2 Life Science Research Center, Mie Univ.



3 JST CREST

# Motivations

- GWAS

To find genetic variations associated with a particular disease.

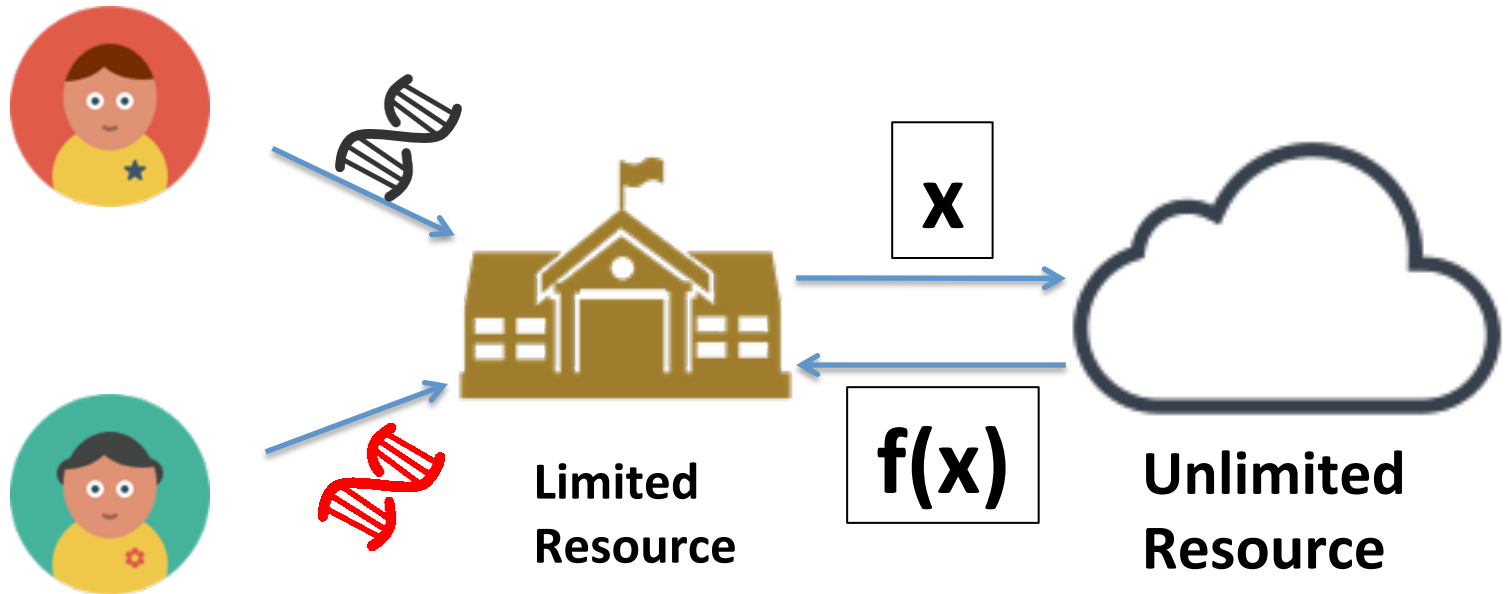
- Outsourcing

To use the cloud resources to conduct large-scale GWAS computations.

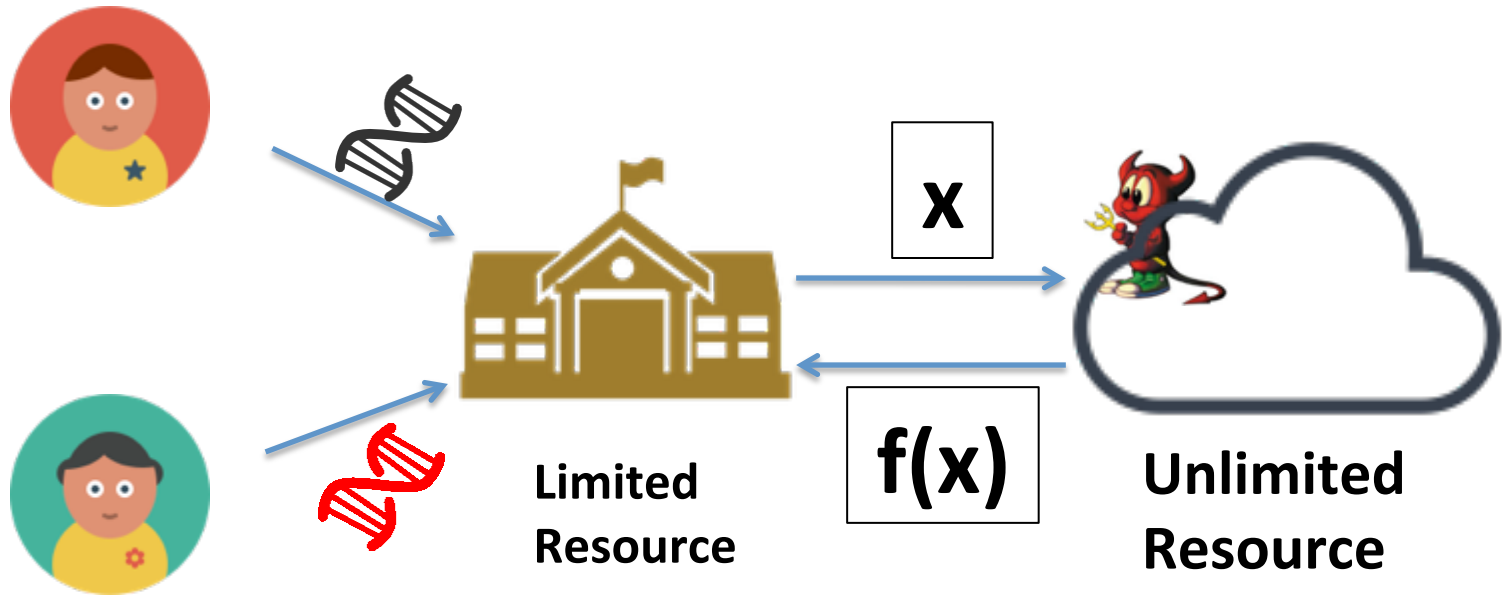
- Personal privacy

Genetic/clinical data is very sensitive.

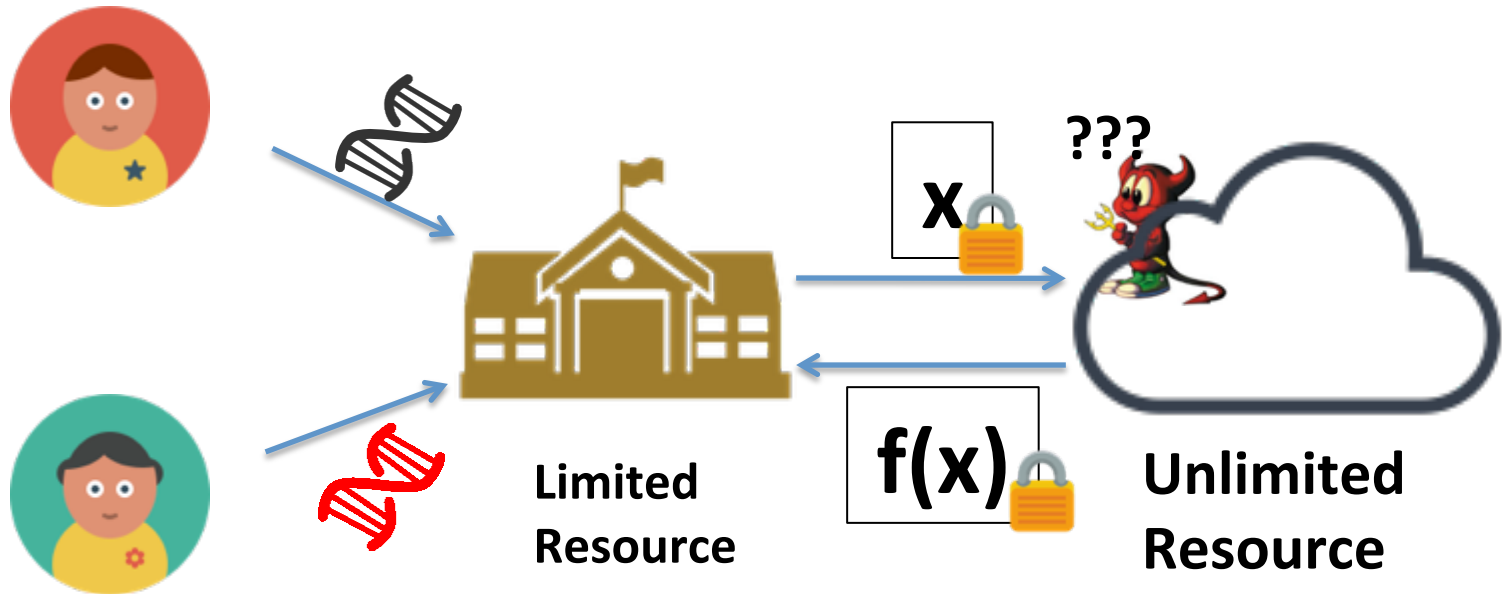
# Outsourcing GWAS



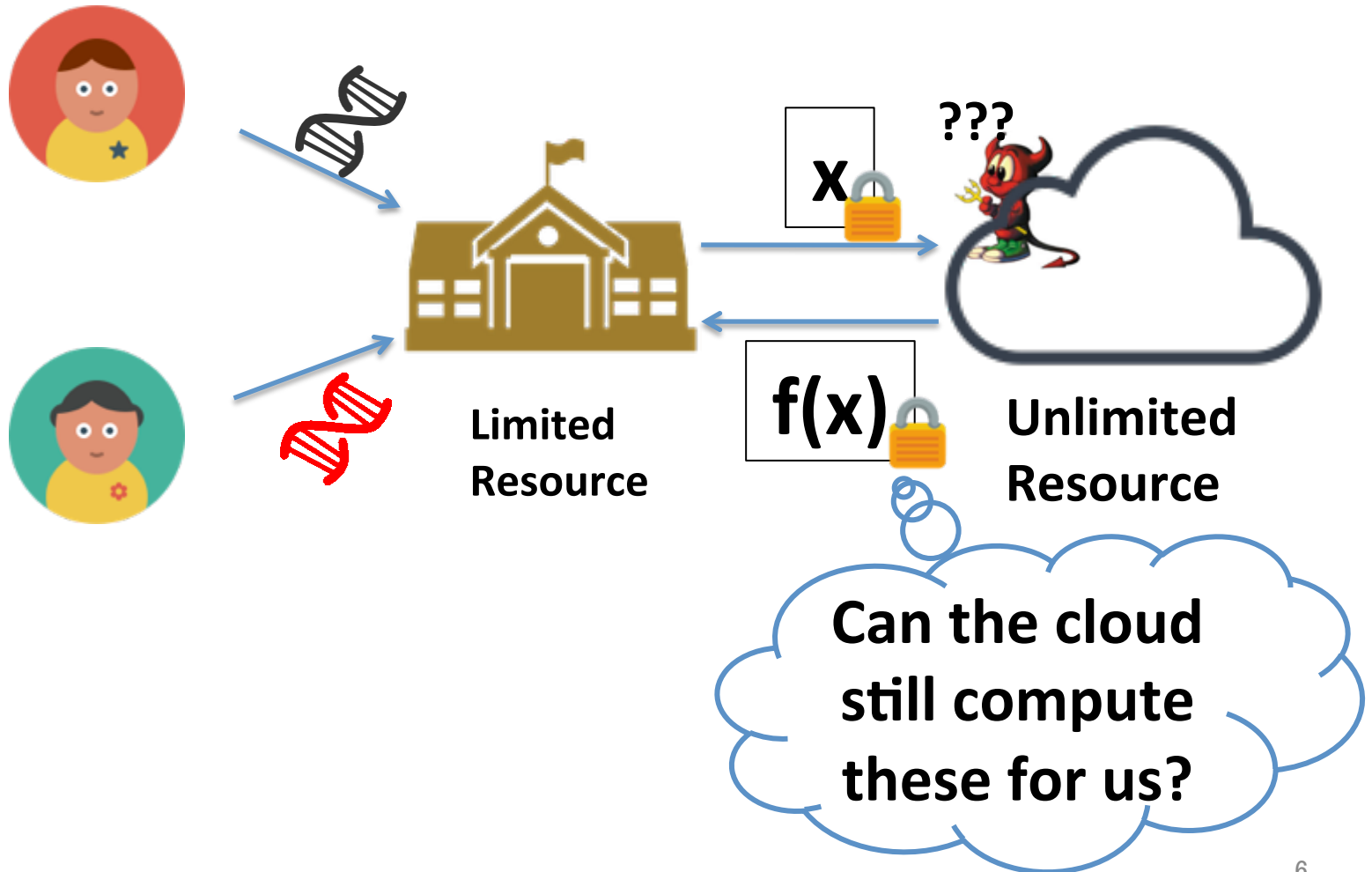
# The evils in the detail



# Protection from Cryptosystem

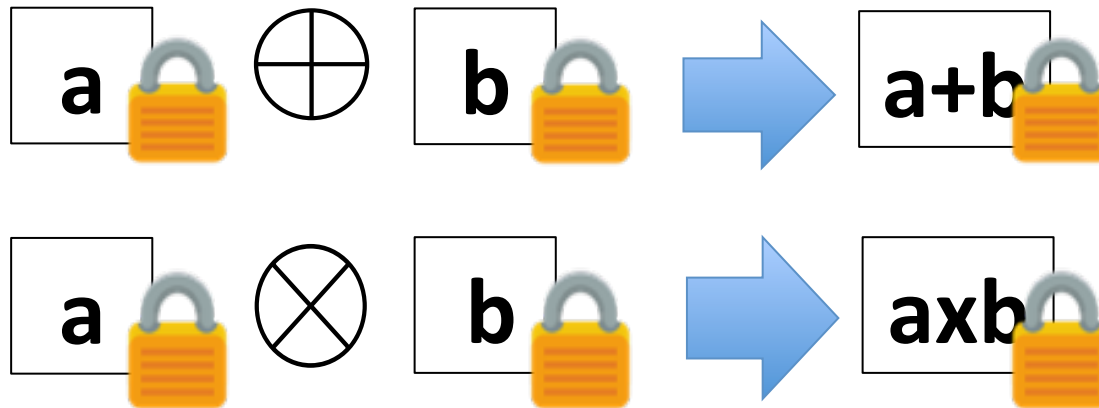


# Protection from Cryptosystem



# Fully Homomorphic Encryption(FHE)

- Mathematic operations can be carried out on encrypted values *without disclosing* these values



Gentry Craig, “A fully homomorphic encryption scheme”,  
Doctoral dissertation, Stanford University, 2009

# Ring Learning With Error(RLWE)

- Fully homomorphic encryption
- *A plaintext is a polynomial*

$$m \in \mathbb{Z}_t[x] / (x^N + 1)$$

P.S.: An integer in  $\mathbb{Z}_t$  can be seen as a degree-0 polynomial

Brakerski Zvika et al., “Leveled fully homomorphic encryption without bootstrapping”, Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, ACM, 2012.



# Outsourcing Statistical Test

For a *single nucleotide polymorphisms*(SNP) and a disease, e.g. diabetes.

- Genotype: [AA, aa, Aa, AA, .....]
- Phenotype: [case, control, case, case, .....]

**N people**



Case: with diabetes

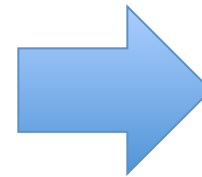
Control: without diabetes

# Outsourcing Statistical Test

- Genotype: [AA, aa, Aa, AA, .....]<sub>N people</sub>
- Phenotype: [case, control, case, case, .....]

## Observation

Genotype	A	a	Count
Case	$o_1$	$o_2$	$n_3$
Control	$o_3$	$o_4$	$n_4$
Count	$n_1$	$n_2$	$2N$



## Expectation

A	a
$e_1$	$e_2$
$e_3$	$e_4$

# Outsourcing Statistical Test

- Genotype: [AA, aa, Aa, AA, .....]<sub>N people</sub>
- Phenotype: [case, control, case, case, .....]

**Observation**

**Expectation**

Genotype				a
Case	$\chi^2 = \sum_{i=1}^4 \frac{(o_i - e_i)^2}{e_i}$			$e_2$
Contrc.	$o_3$	$\chi^2 \geq 3.84; 95\%$		$e_4$
Count	$n_1$	$n_2$	$2N$	

# Our Encoding for SNP data

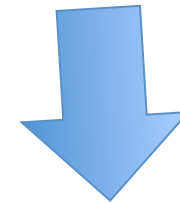
- Genotype: [AA, aa, Aa, AA, .....]
- Phenotype: [case, control, case, case, .....]

$$x_i = \begin{cases} 2, & \text{AA} \\ 1, & \text{Aa} \\ 0, & \text{o.w} \end{cases} \quad \Rightarrow \quad \mathbf{x} \quad [2, 0, 1, 2, \dots]$$

$$y_i = \begin{cases} 1, & \text{case} \\ 0, & \text{control} \end{cases} \quad \Rightarrow \quad \mathbf{y} \quad [1, 0, 1, 1, \dots]$$

# Compute the contingency table

$$x_i = \begin{cases} 2, AA \\ 1, Aa \\ 0, \text{o.w} \end{cases} \quad y_i = \begin{cases} 1, \text{case} \\ 0, \text{control} \end{cases}$$



Genotype	A	a	Count
Case	$o_1$	$o_2$	$n_3$
Control	$o_3$	$o_4$	$n_4$
Count	$n_1$	$n_2$	$2N$

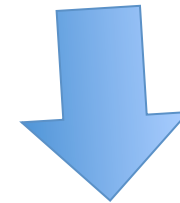
$$o_1 = \langle x, y \rangle$$

$$n_1 = \langle x, 1 \rangle$$

$$n_3 = \langle y, 1 \rangle$$

# Compute the contingency table

$$x_i = \begin{cases} 2, AA \\ 1, Aa \\ 0, \text{o.w} \end{cases} \quad y_i = \begin{cases} 1, \text{case} \\ 0, \text{control} \end{cases}$$



Genotype	A	a	Count
Case	$o_1$	$o_2$	$n_3$
Control	$o_3$	$o_4$	$n_4$
Count	$n_1$	$n_2$	$2N$

$$o_1 = \langle x, y \rangle$$

$$n_1 = \langle x, 1 \rangle$$

$$n_3 = \langle y, 1 \rangle$$

# Compute the contingency table

$$x_i = \begin{cases} 2, AA \\ 1, Aa \\ 0, o.w \end{cases} \quad y_i = \begin{cases} 1, \text{case} \\ 0, \text{control} \end{cases}$$

Genotype			
AA			4
Aa			
Count	$n_1$	$n_2$	$2N$

How to efficiently  
compute the  
scalar product on  
encrypted data ??

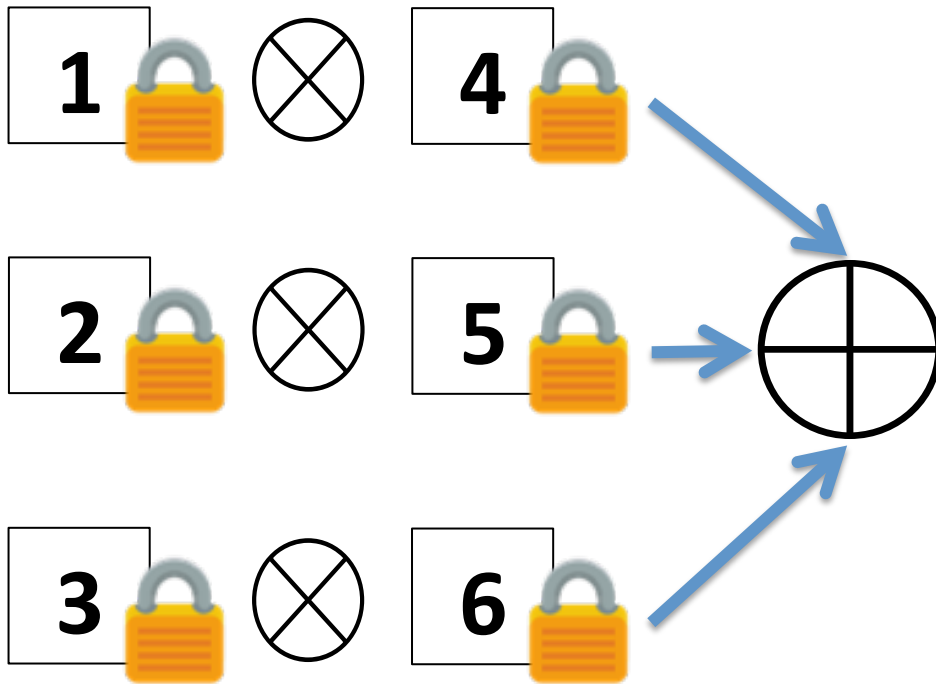
$$o_1 = \langle x, y \rangle$$

$$n_1 = \langle x, 1 \rangle$$

$$n_3 = \langle y, 1 \rangle$$

# Scalar product: A naïve way

$$\mathbf{v} = [1, 2, 3] \quad \mathbf{u} = [4, 5, 6]$$



$$\|\mathbf{v}\| = d$$
$$\#\text{ciphertext} = 2d$$

$$\text{Mul.} = d$$

$$\text{Add} = d - 1$$



# Scalar product: more efficient way

Plaintext space of RLWE :  $\mathbb{Z}_t[x]/(x^N + 1)$

$$\mathbf{v} = [1, 2, 3] \rightarrow V(x) = 1 + 2x + 3x^2$$

$$\mathbf{u} = [4, 5, 6] \rightarrow U(x) = 6 + 5x + 4x^2$$

$$\boxed{V(x)} \otimes \boxed{U(x)}$$

$$\rightarrow \boxed{6 + 17x + 32x^2 + 27x^3 + 12x^4}$$

Only need *ONE* multiplication ! ( $\|\mathbf{v}\| < N$ )

# Scalability

- Plaintext space:  $m \in \mathbb{Z}_t[x]/(x^N + 1)$
- $\|\mathbf{v}\| \geq N$  ? To partition into smaller parts

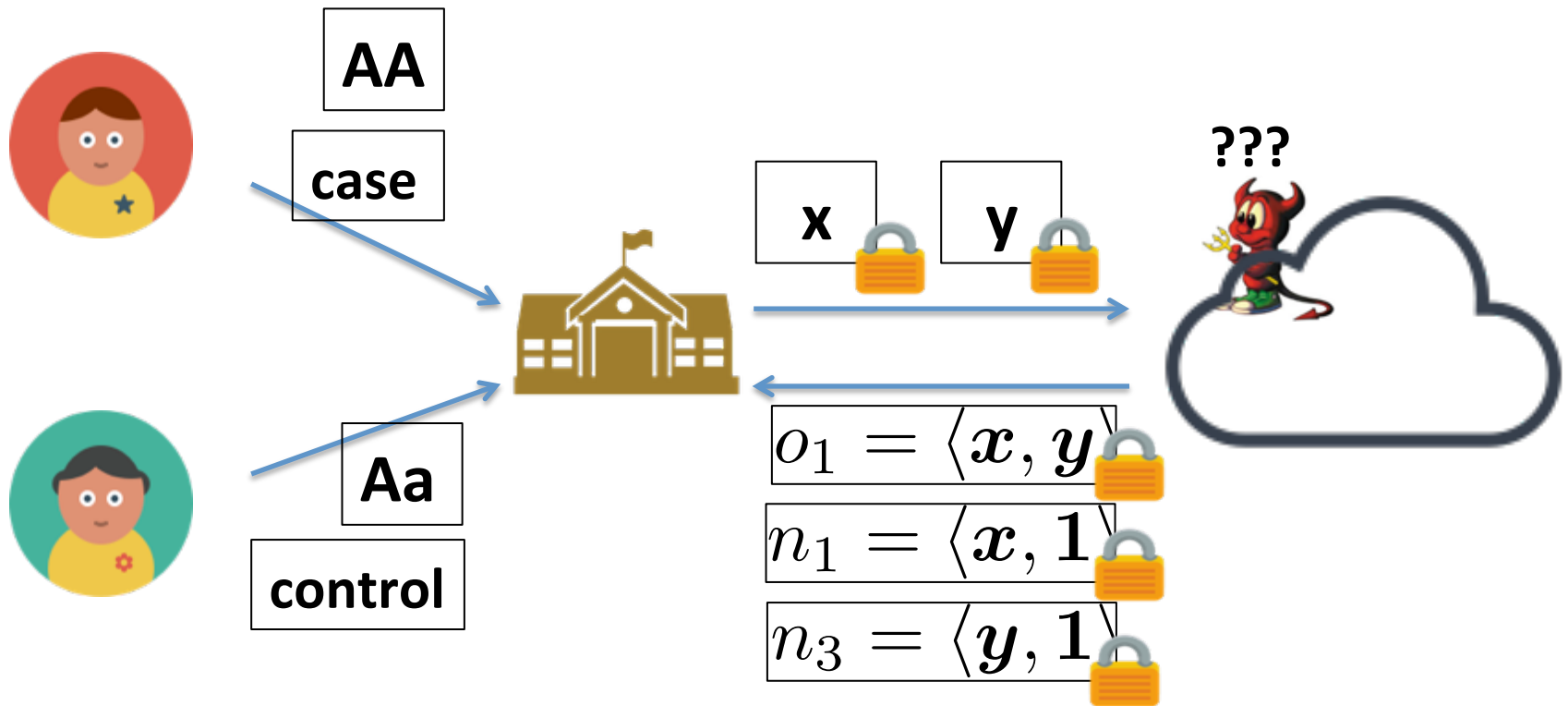
$$\mathbf{v} = [\mathbf{v}_1 || \cdots || \mathbf{v}_k] \quad \mathbf{u} = [\mathbf{u}_1 || \cdots || \mathbf{u}_k]$$

$$\langle \mathbf{u}, \mathbf{v} \rangle := \sum_{i=1}^k \langle \mathbf{u}_i, \mathbf{v}_i \rangle$$

Mul. :  $k$   
Add :  $k - 1$

For example:  $N = 8192$ , to conduct  $\|\mathbf{v}\| = 10000$ ;  
 $k = 2$

# The whole image



# Comparison Method

- Genotype: [AA, aa, Aa, AA, .....]

Genotype  
Encoding

AA	→	[1], [0], [0]
Aa	→	[0], [1], [0]
aa	→	[0], [0], [1]

- Phenotype: [case, control, case, case, .....]

Phenotype  
Encoding

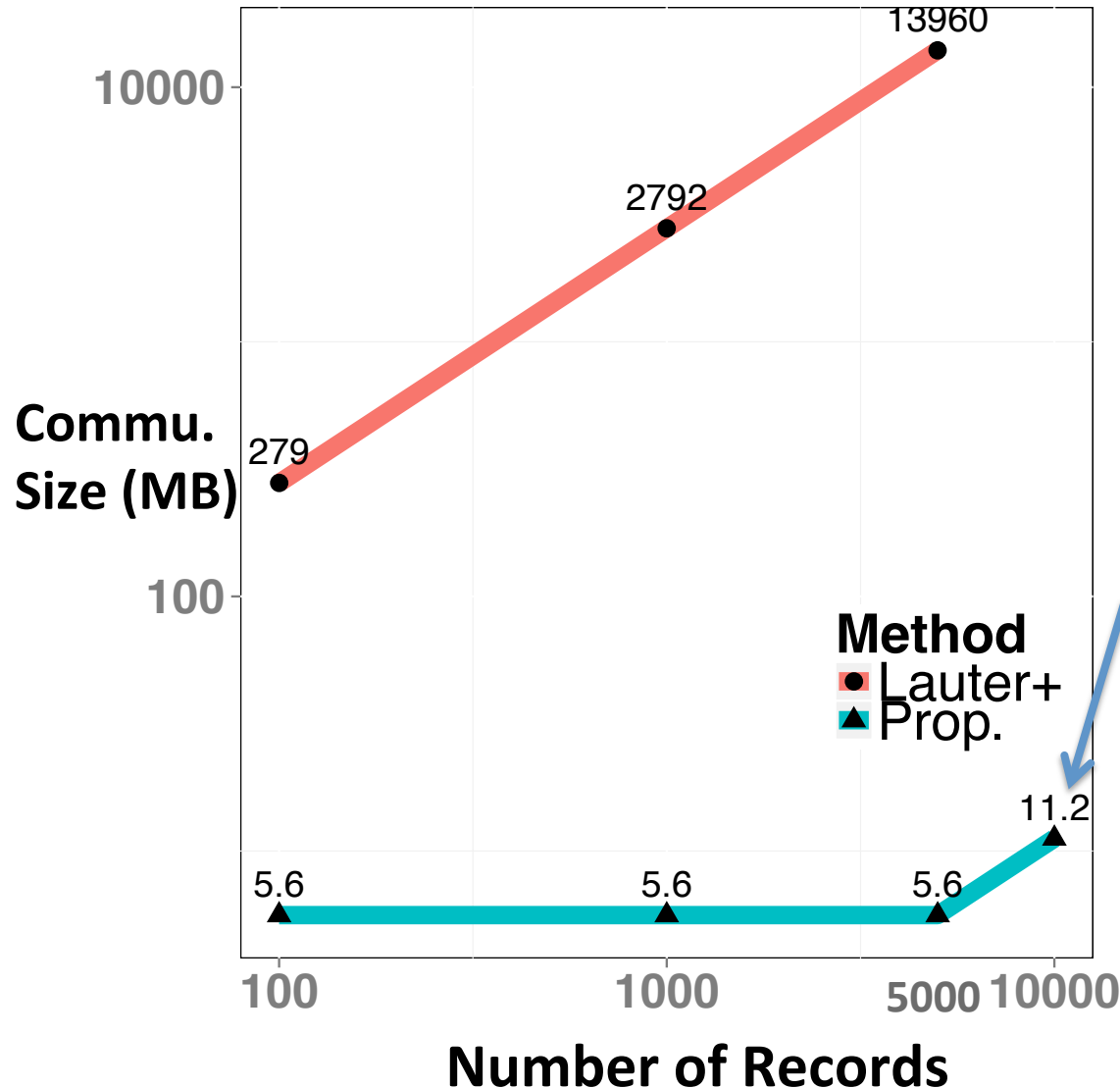
case	→	[1], [0]
control	→	[0], [1]

# Experiment Settings

- Encryption Implementation: HElib
- The maximum degree of the polynomial:  $N = 8192$
- Security parameter:  $> 80$ bits
- CPU 2.3GHz; RAM 16G

[<https://github.com/shaih/HElib>]

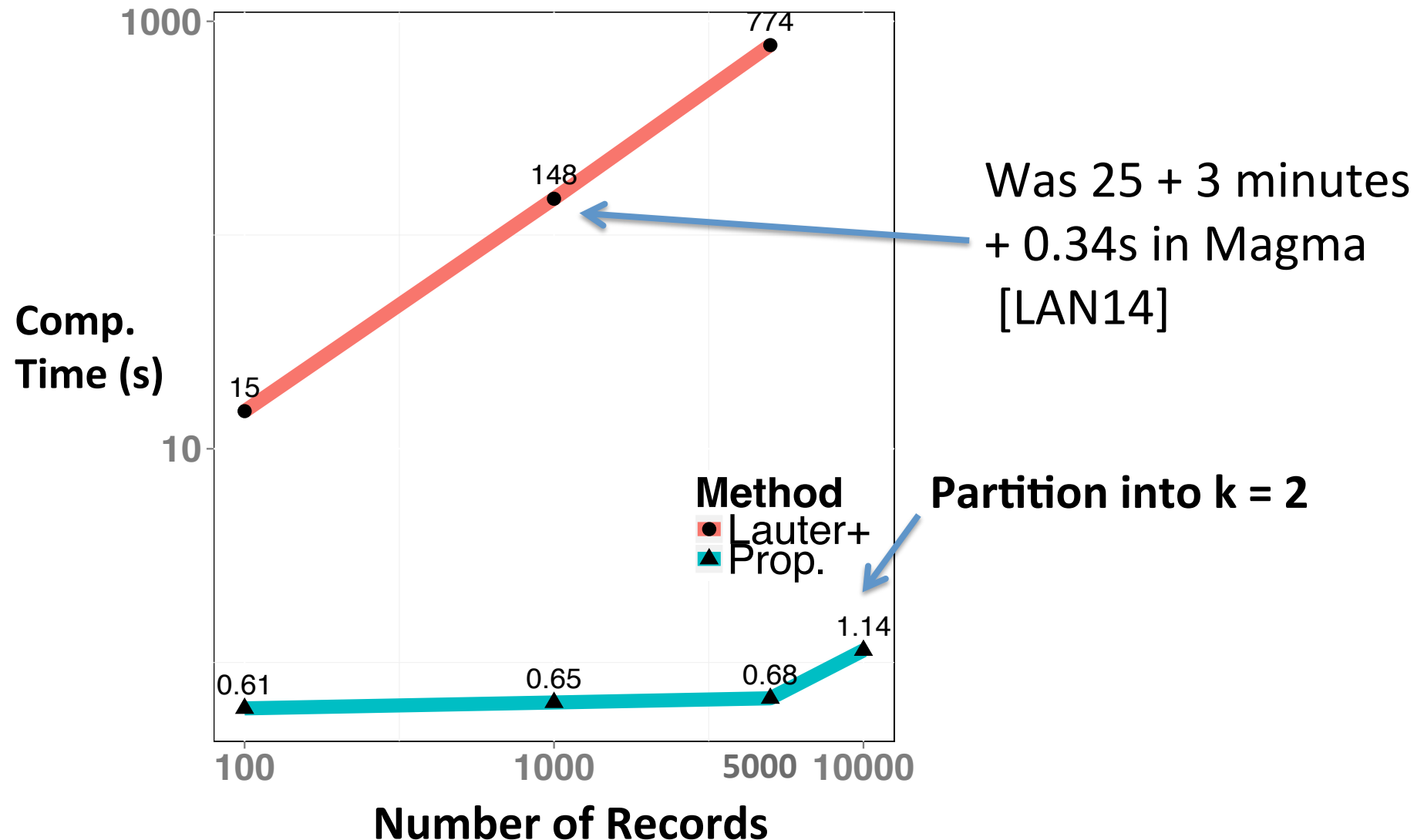
# Experimental Result: Communication Size



**Comparison Encoding:**  
5000 Records => 25000 ciphers  
**Proposal Encoding:**  
5000 Records => only 2 ciphers

**Partition into  $k = 2$**

# Experimental Result: Computation Time



# Conclusion

1. With suitable data arrangement, efficient computation is achievable.
2. Our method helps space/time complexity.



Thank you!