Efficient Secure Outsourcing of Genome-wide Association Studies

Wenjie Lu[1], Yoshiji Yamada[2], Jun Sakuma[1,3]

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.

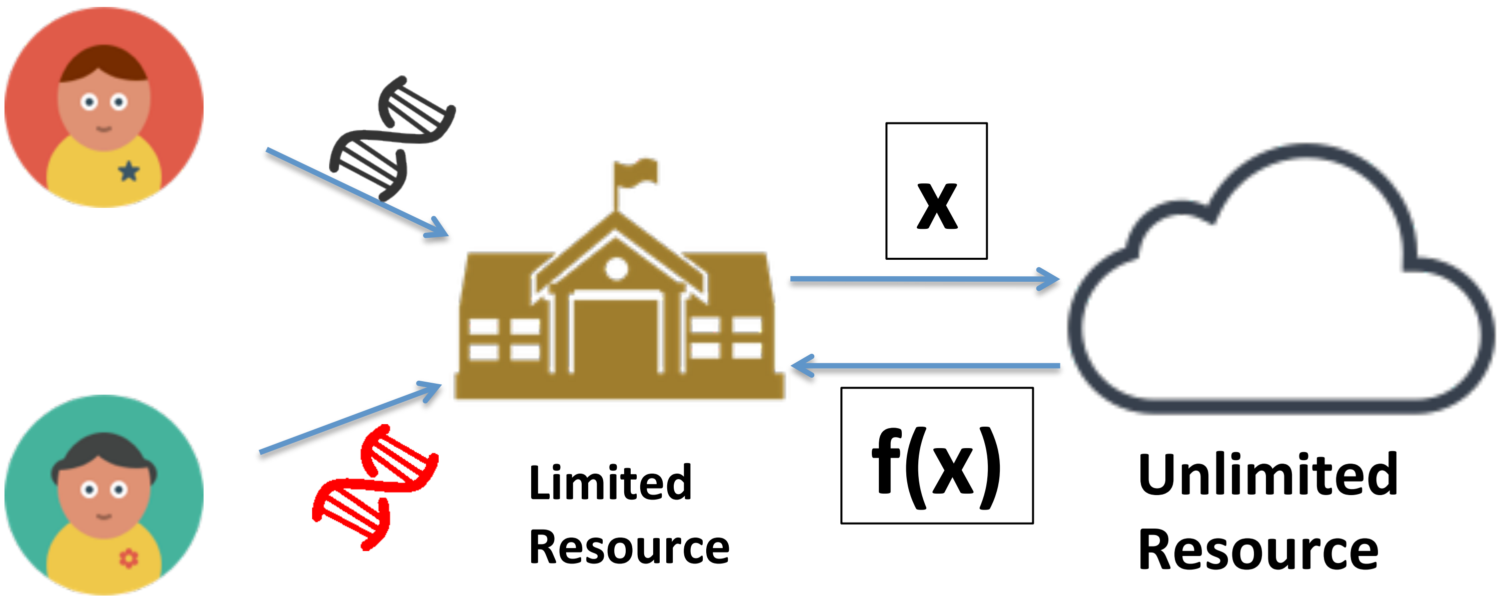
- Outsourcing

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

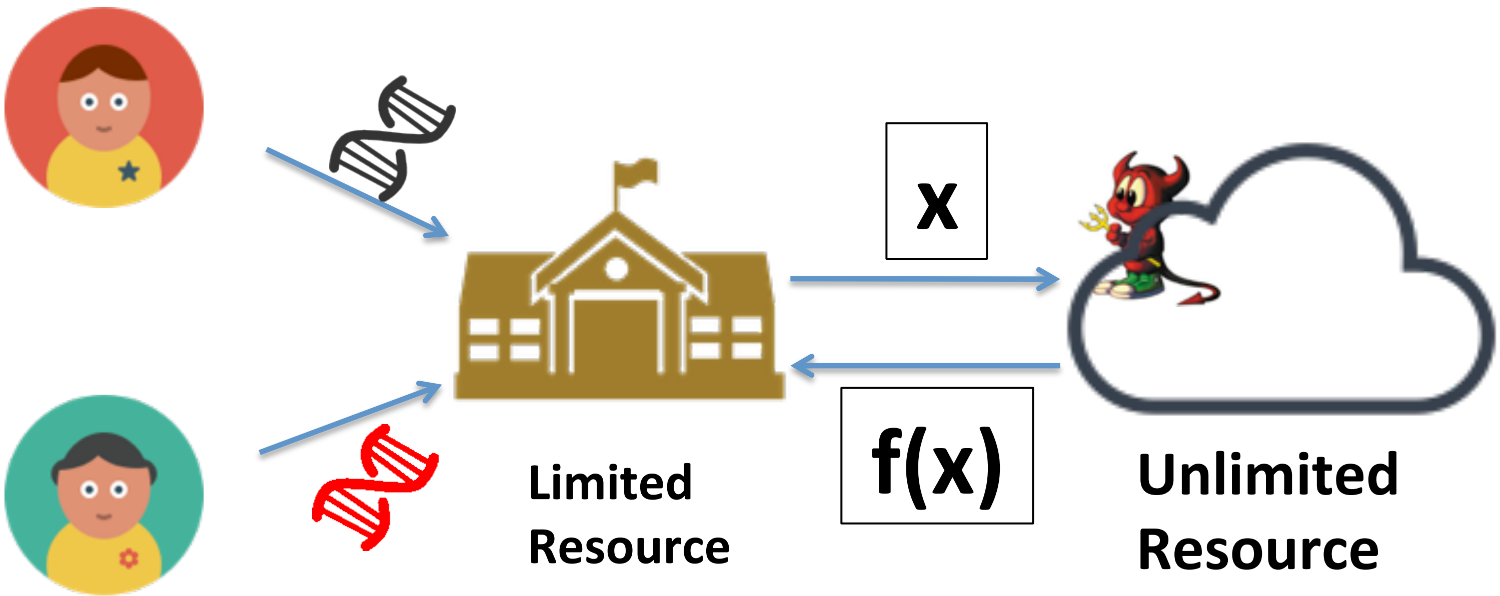
- Personal privacy

  Genetic/clinical data is very sensitive.

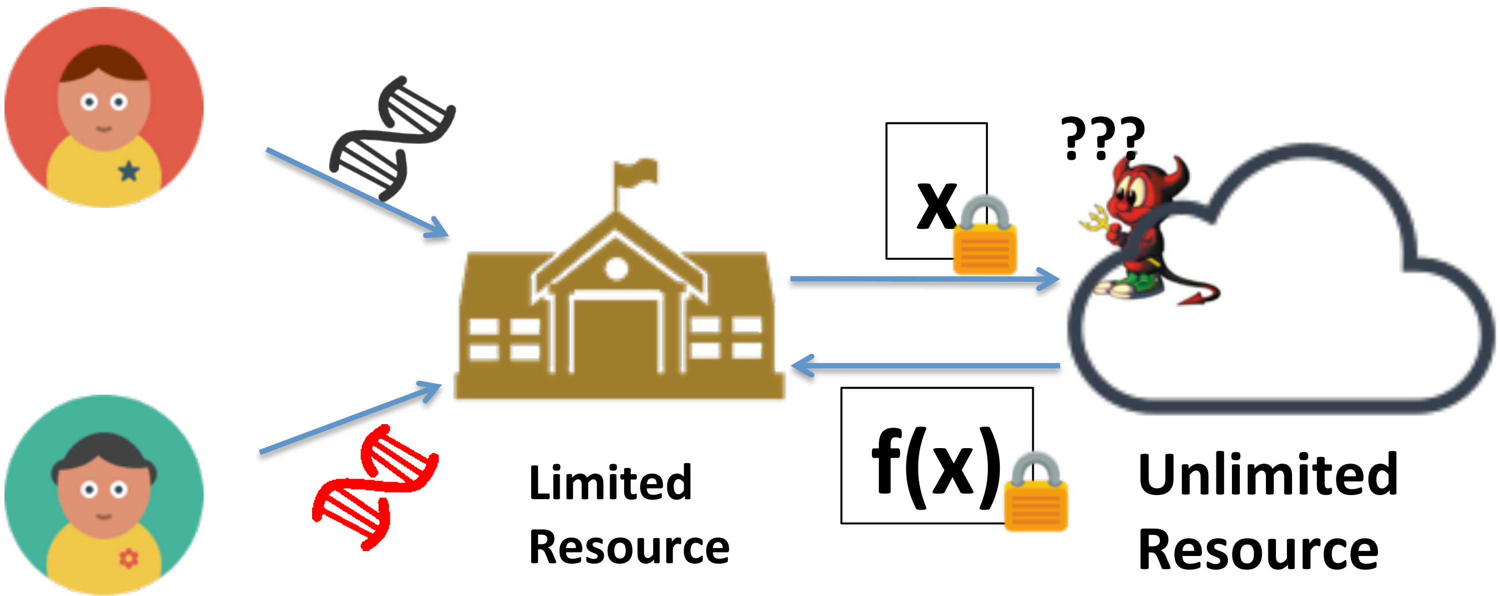
Outsourcing GWAS



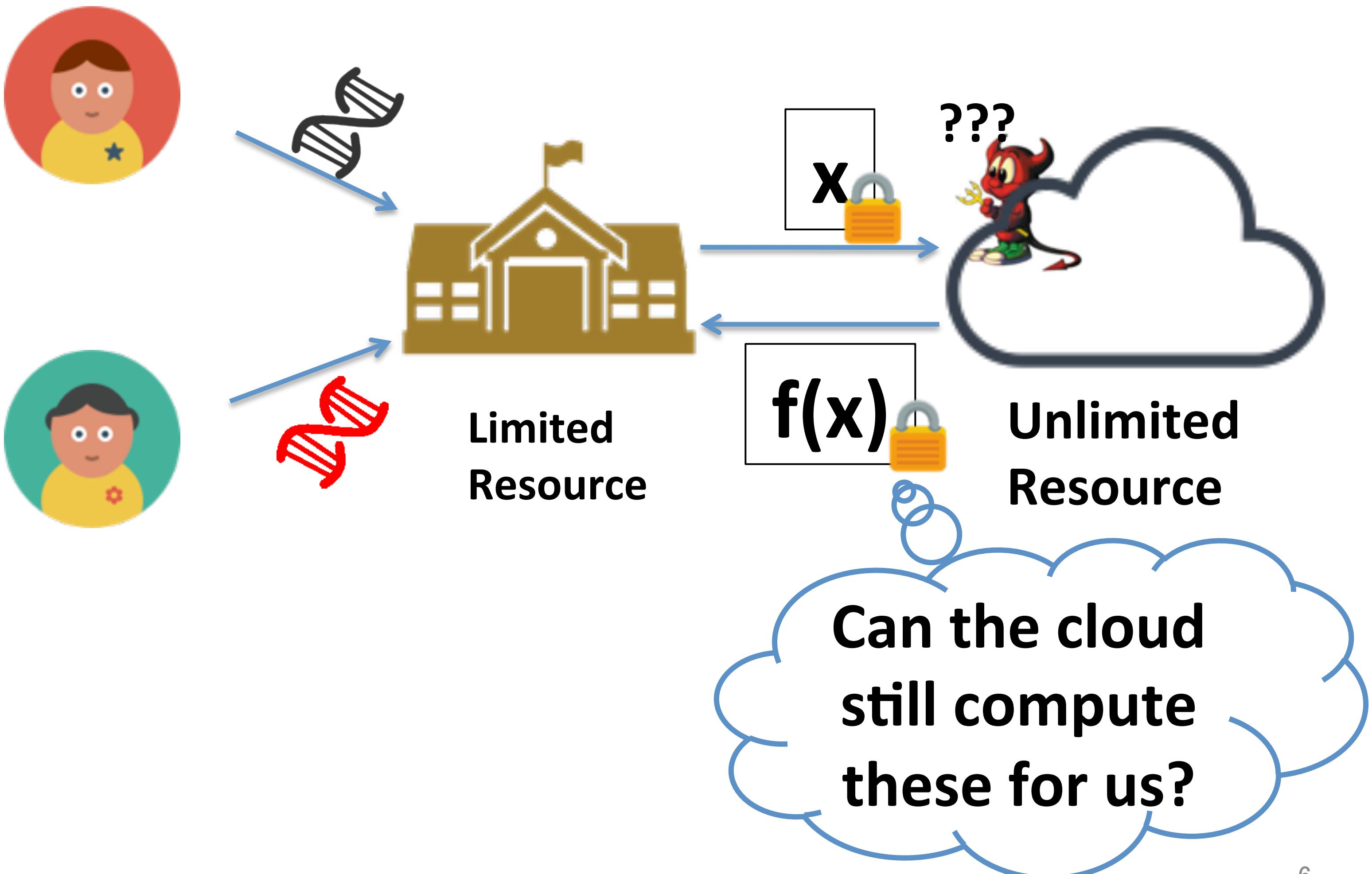
The evils in the detail



Protection from Cryptosysmtem

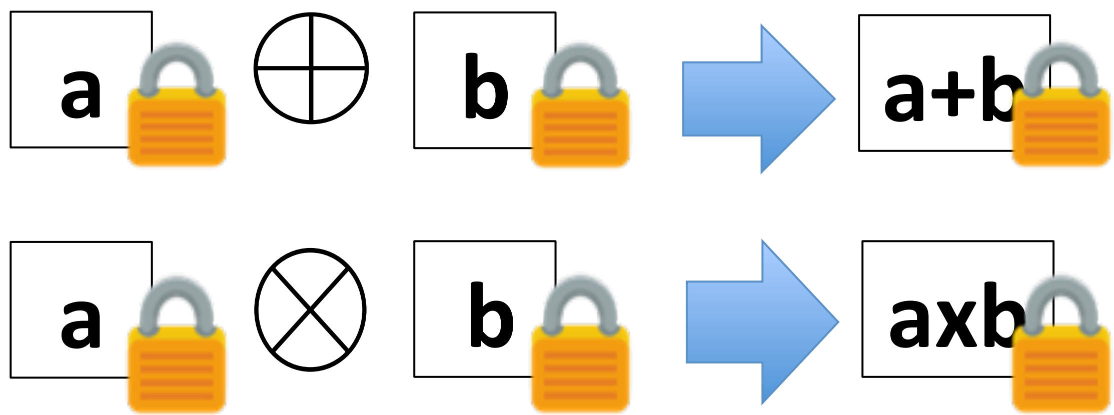


Protection from Cryptosysmtem



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
- *<u>A plaintext is a polynomial</u>*

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

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

N people

Observation

Expectation

$$\chi^2 = \sum_{i=1}^{4} \frac{(o_i - e_i)^2}{e_i}$$

$$\chi^2 \geq 3.84; 95\%$$

Genotyp			
Case			
Contr	o_3		
Count	n_1	n_2	$2N$

	a
	e_2
e_3	e_4

Our Encoding for SNP data

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

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

$$y_i = \begin{cases} 1, \mathrm{case} \\ 0, \mathrm{control} \end{cases} \Rightarrow \boldsymbol{y} \quad \mathbf{[1, 0, 1, 1, ...]}$$

Compute the contingency table

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

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

$$o_1 = \langle \boldsymbol{x}, \boldsymbol{y} \rangle$$

$$n_1 = \langle \boldsymbol{x}, \mathbf{1} \rangle$$

$$n_3 = \langle \boldsymbol{y}, \mathbf{1} \rangle$$

Compute the contingency table

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

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

$$o_1 = \langle \boldsymbol{x}, \boldsymbol{y} \rangle$$

$$n_1 = \langle \boldsymbol{x}, \mathbf{1} \rangle$$

$$n_3 = \langle \boldsymbol{y}, \mathbf{1} \rangle$$

Compute the contingency table

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

Genot			
Co			4
Count	n_1	n_2	$2N$

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

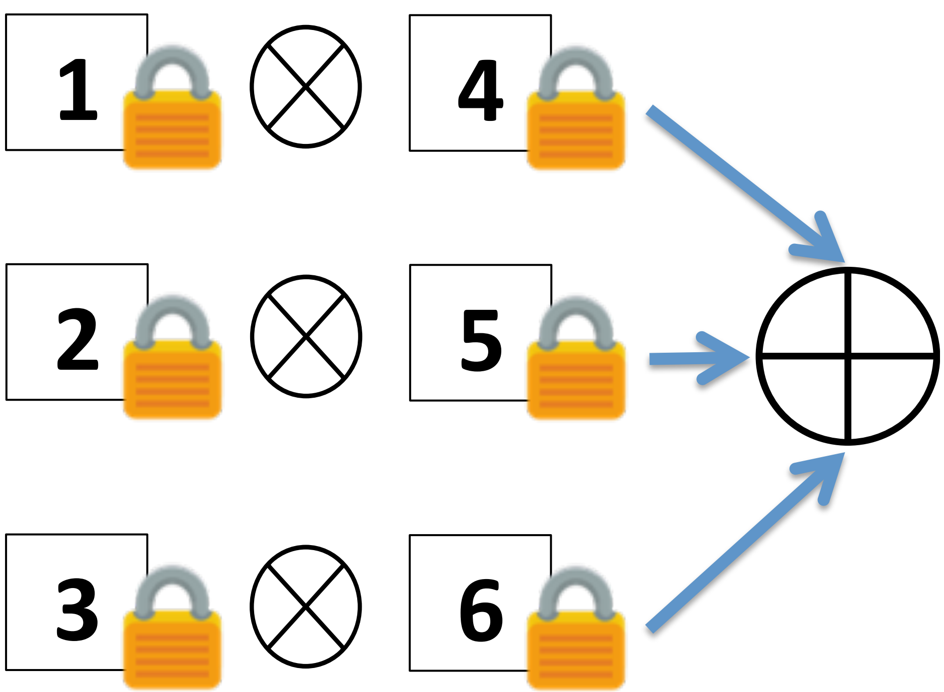
$$o_1 = \langle \boldsymbol{x}, \boldsymbol{y} \rangle$$

$$n_1 = \langle \boldsymbol{x}, \mathbf{1} \rangle$$

$$n_3 = \langle \boldsymbol{y}, \mathbf{1} \rangle$$

Scalar product: A naïve way

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



$$\|\boldsymbol{v}\| = d$$

$$\#\text{ciphertext} = 2d$$

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

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

Scalar product: more efficient way

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

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

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

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

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

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

Yasuda Masaya, et al. "Secure pattern matching using somewhat homomorphic encryption." Proceedings of the 2013 ACM workshop on Cloud computing security workshop. ACM, 2013.

Scalability

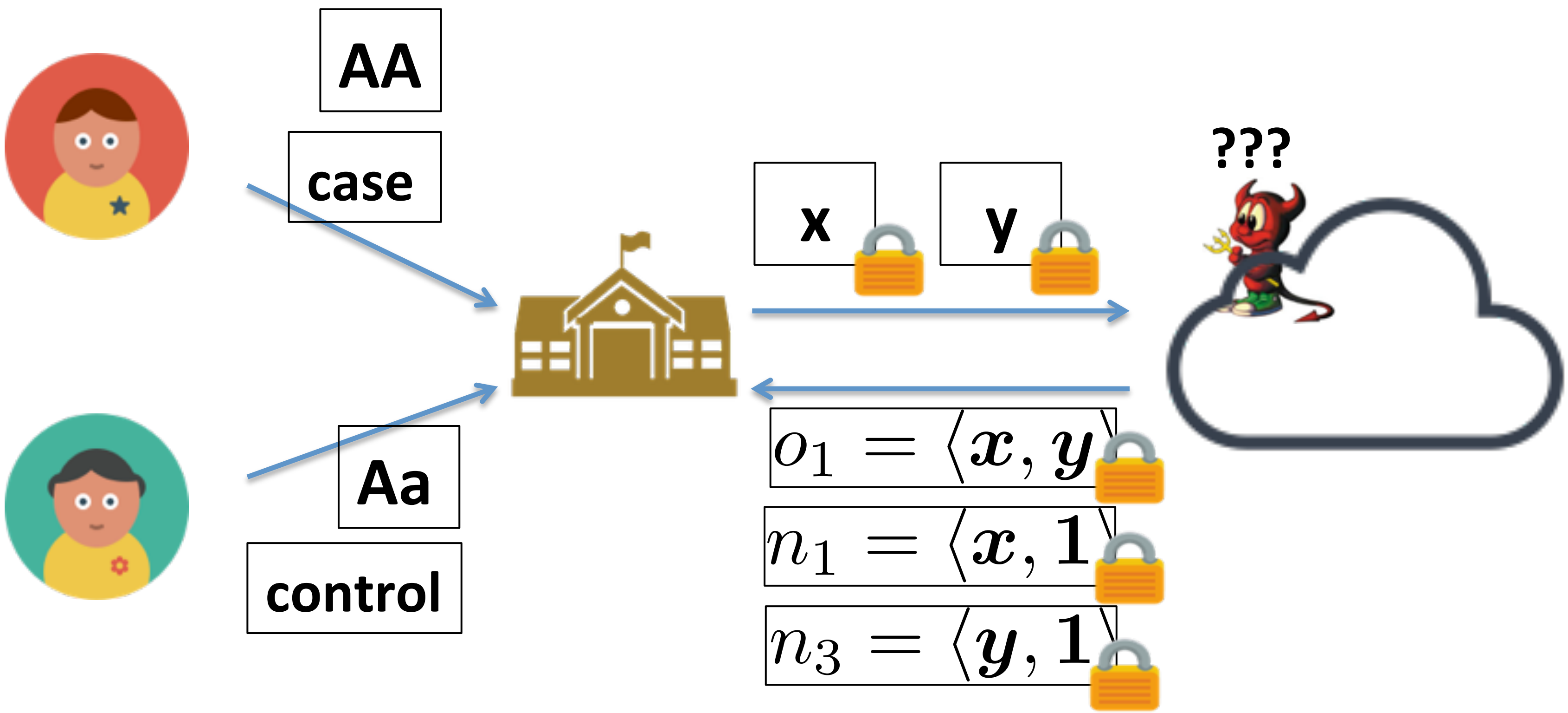
- Plaintext space: $m \in \mathbb{Z}_t[x]/(x^N + 1)$
- $\|\boldsymbol{v}\|$ >= N ? To partition into smaller parts

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

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle := \sum_{i=1}^{k} \langle \boldsymbol{u}_i, \boldsymbol{v}_i \rangle \qquad \begin{array}{r} \text{Mul.} : k \\ \text{Add} : k - 1 \end{array}$$

For example: N = 8192, to conduct $\|\boldsymbol{v}\|$ = 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]

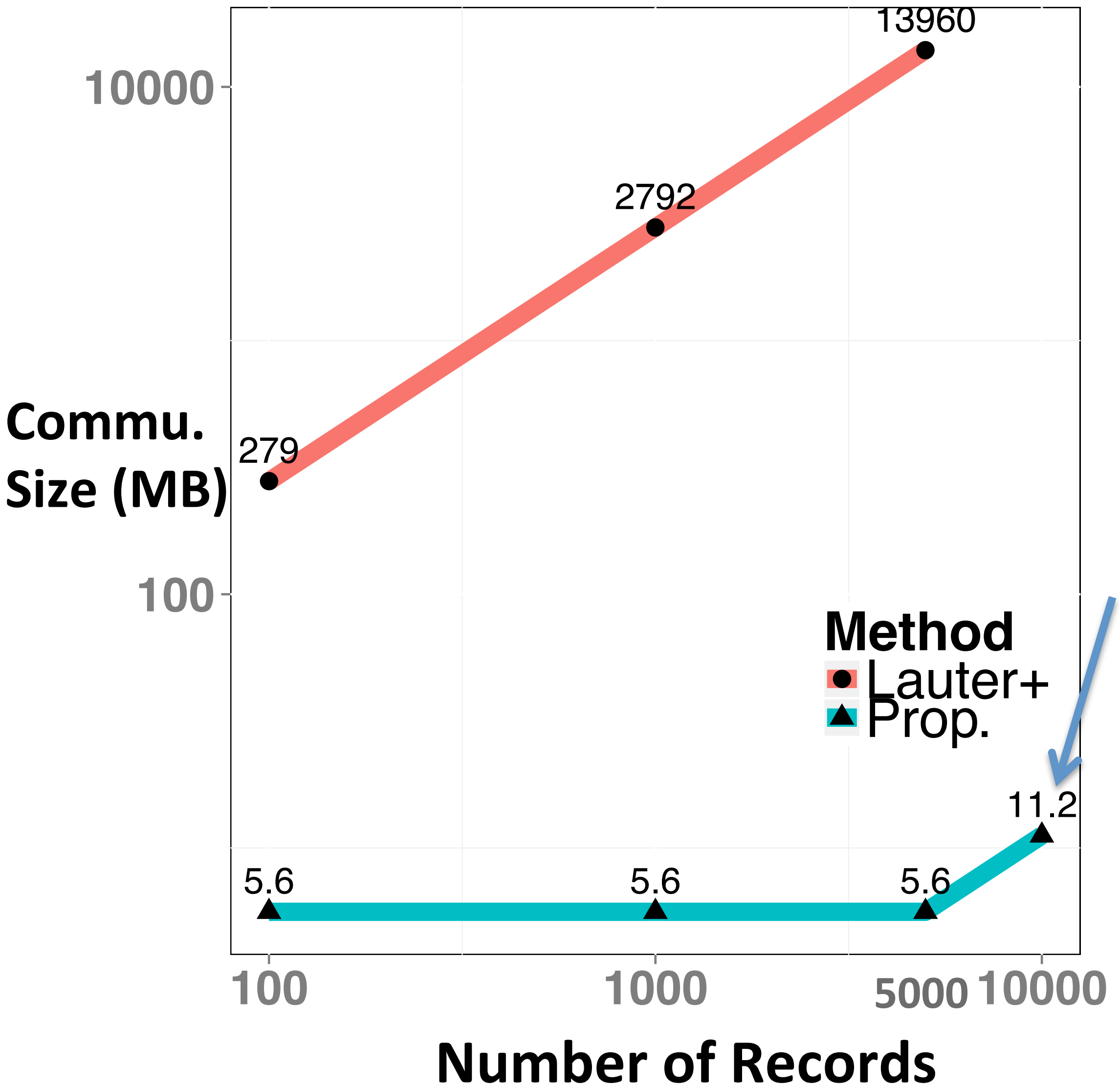
Kristin Lauter and Adriana Lopez-Alt and Michael Naehrig "Private computation on encrypted genomic data" 14th Privacy Enhancing Technologies Symposium, Workshop on Genome Privacy. 2014

Experiment Settings

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

[https://github.com/shaih/Helib]

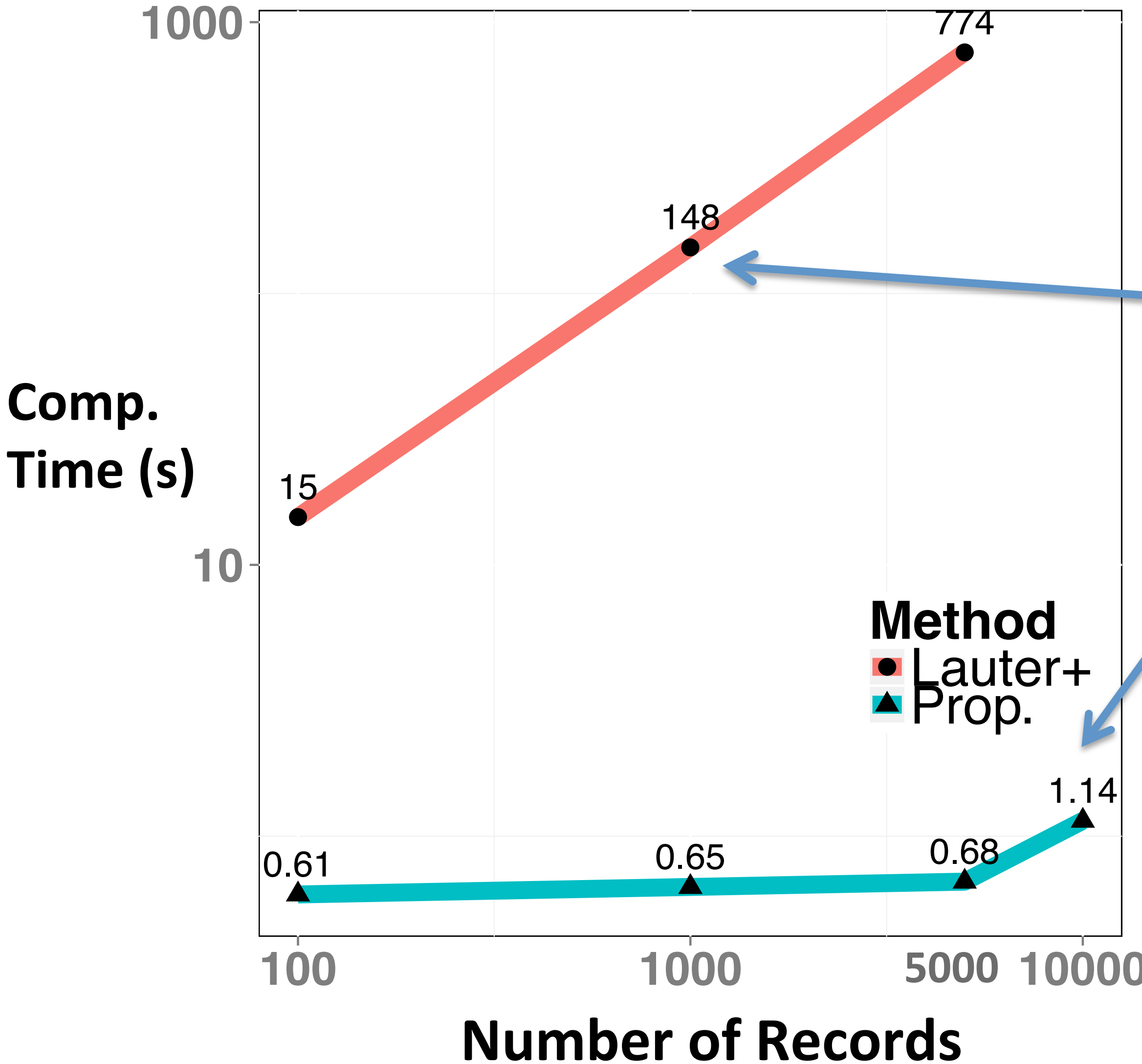
Experimental Result: Communication Size



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

Partition into k = 2

Experimental Result: Computation Time



Was 25 + 3 minutes + 0.34s in Magma [LAN14]

Partition into k = 2

Conclusion

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

Thank you!