Private Computation on Encrypted Genomic Data

Michael Naehrig
Cryptography Research Group
Microsoft Research

Joint work with Adriana Lopez-Alt (New York University) and Kristin Lauter (MSR)

Workshop on Genome Privacy 2014
Amsterdam, 15 July 2014

Encrypt everything?

- Protect outsourced data by encrypting everything
- “Conventional” encryption methods do not allow any computation on the encrypted data
without using the secret key and decrypting it
- Homomorphic encryption schemes allow specific operations on encrypted data with only public information

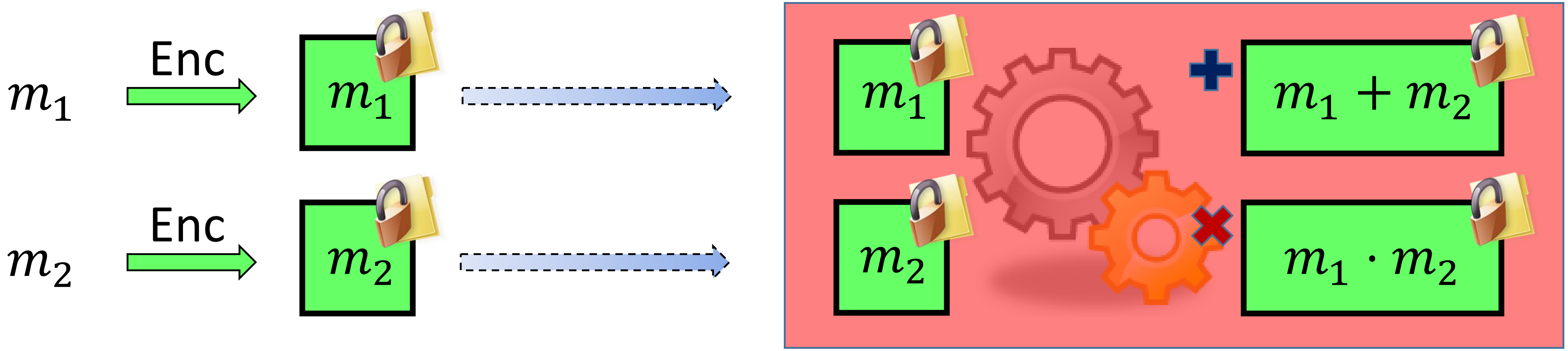
Fully Homomorphic Encryption (FHE)

FHE enables unlimited computation on encrypted data

- Public operations on ciphertexts:

$+$ $(\text{Enc}(m_1), \text{Enc}(m_2)) \rightarrow \text{Enc}(m_1 + m_2)$

$\times$ $(\text{Enc}(m_1), \text{Enc}(m_2)) \rightarrow \text{Enc}(m_1 \cdot m_2)$



Fully Homomorphic Encryption (FHE)

FHE enables unlimited computation on encrypted data

- Public operations on ciphertexts:

  ✚ (Enc(m_1), Enc(m_2)) $\rightarrow \mathrm{Enc}(m_1 + m_2)$

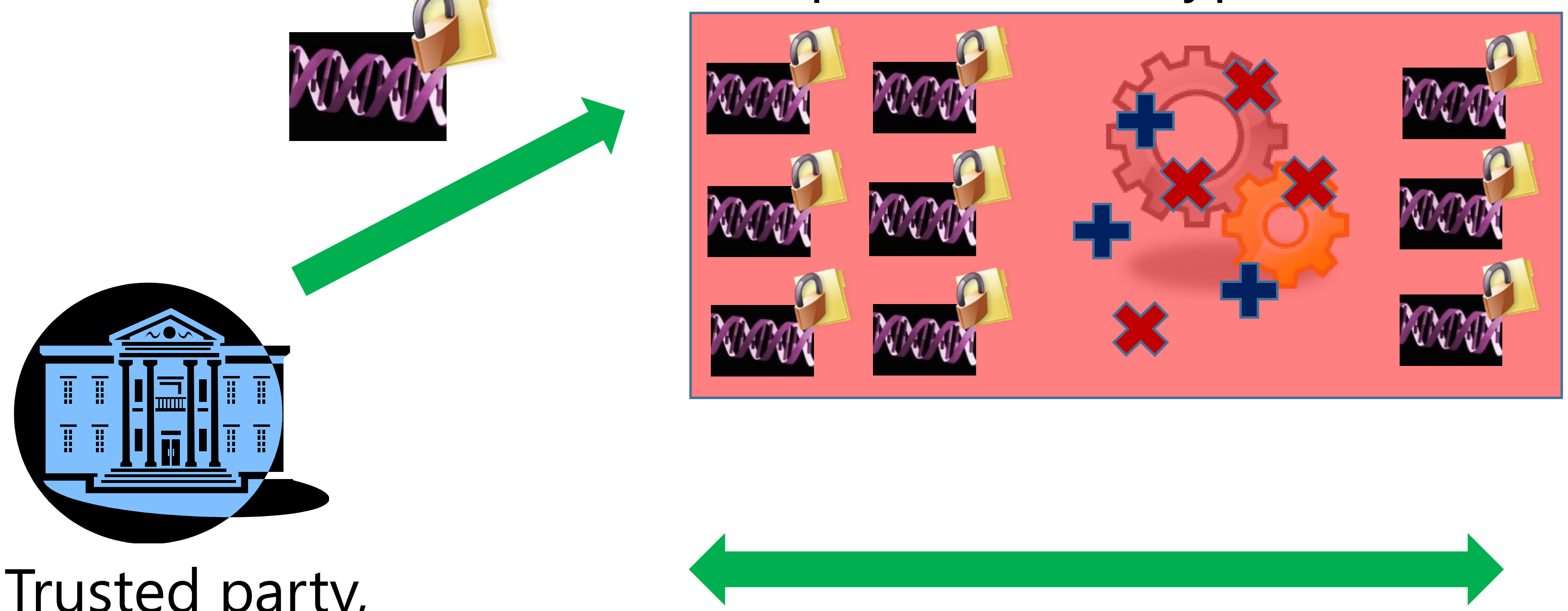
  ✖ (Enc(m_1), Enc(m_2)) $\rightarrow \mathrm{Enc}(m_1 \cdot m_2)$

- For data encrypted bitwise $(m_1, m_2 \in \{0,1\})$, operations $m_1 + m_2$ and $m_1 \cdot m_2$ are bitwise (XOR and AND)

- Get arbitrary operations via binary circuits.

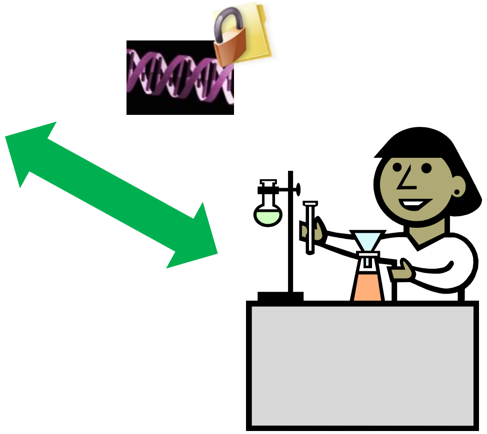
A possible scenario for genomic data?

Untrusted cloud service, stores and computes on encrypted data



Trusted party, hosts data and regulates access

Requests for decryption of certain results



Requests encrypted results of specific computation

This might not be a solution to your problem!

Here’s the caveat

- FHE schemes do exist!

- BUT FHE on binary circuits with bitwise encryption is extremely inefficient:

    - huge ciphertexts,
    - costly noise handling,
    - large overhead in storage space and computation time

Ways to slightly ease the pain

- Pack more data into ciphertexts
- Use so-called leveled homomorphic schemes
- Use arithmetic circuits and restrict to computations with low multiplicative depth

This comes at a cost: restrictions on the type of computations that can be done!

Homomorphic Encryption from RLWE

- Uses polynomial rings as plaintext and ciphertext spaces

$$R = \mathbf{Z}[X]/(X^n + 1), \quad n = 2^k$$

Example: $n = 8$

$$a_1 = 2x^7 + x^5 - 11x^4 + x^2 + 5x + 7$$
$$a_2 = x^6 - 4x^5 - 3x^3 + 12x^2 + 3$$

$a_1 =$	$2x^7$		$+x^5$	$-11x^4$		$+x^2$	$+5x$	$+7$
$a_2 =$		x^6	$-4x^5$		$-3x^3$	$+12x^2$		$+3$
$a_1 + a_2 =$	$2x^7$	$+x^6$	$-3x^5$	$-11x^4$	$-3x^3$	$+13x^2$	$+5x$	$+10$
$a_1 \cdot a_2 =$	$52x^7$	$-145x^6$	$-30x^5$	$-28x^4$	$+38x^3$	$+108x^2$	$-53x$	$+23$

Homomorphic Encryption from RLWE

- Uses polynomial rings as plaintext and ciphertext spaces

$$R = \mathbf{Z}[X]/(X^n + 1), \quad n = 2^k$$

- Work with polynomials in R modulo some $q \in \mathbf{Z}$

- Homomorphic operations (✚ / ✖) correspond to polynomial operations (add/mult) in R
- ✚ is relatively efficient, ✖ is costly

- Use this structure to encode and work with your data

Homomorphic Encryption from RLWE

- Encode an integer $z \in \mathbf{Z}$ as a polynomial $m \in R$ with $m(2) = z$.

Example: $n = 8$

$z = 13, (z)_2 = 1101$
Use the polynomial
$m_{13} = x^3 + x^2 + 1$

$z = 11, (z)_2 = 1011$
$m_{13} = x^3 + x + 1$

Addition
$m_{13} + m_{11} = 2x^3 + x^2 + x + 2$
$(m_{13}+m_{11})(2) = 2 \cdot 8 + 4 + 2 + 2 = 24$

Multiplication
$m_{13} \cdot m_{11}$
$= x^6 + x^5 + x^4 + 3 \cdot x^3 + x^2 + x + 1$
$(m_{13} \cdot m_{11})(2)$
$= 64 + 32 + 16 + 3 \cdot 8 + 4 + 2 + 1 = 143$

HE Performance

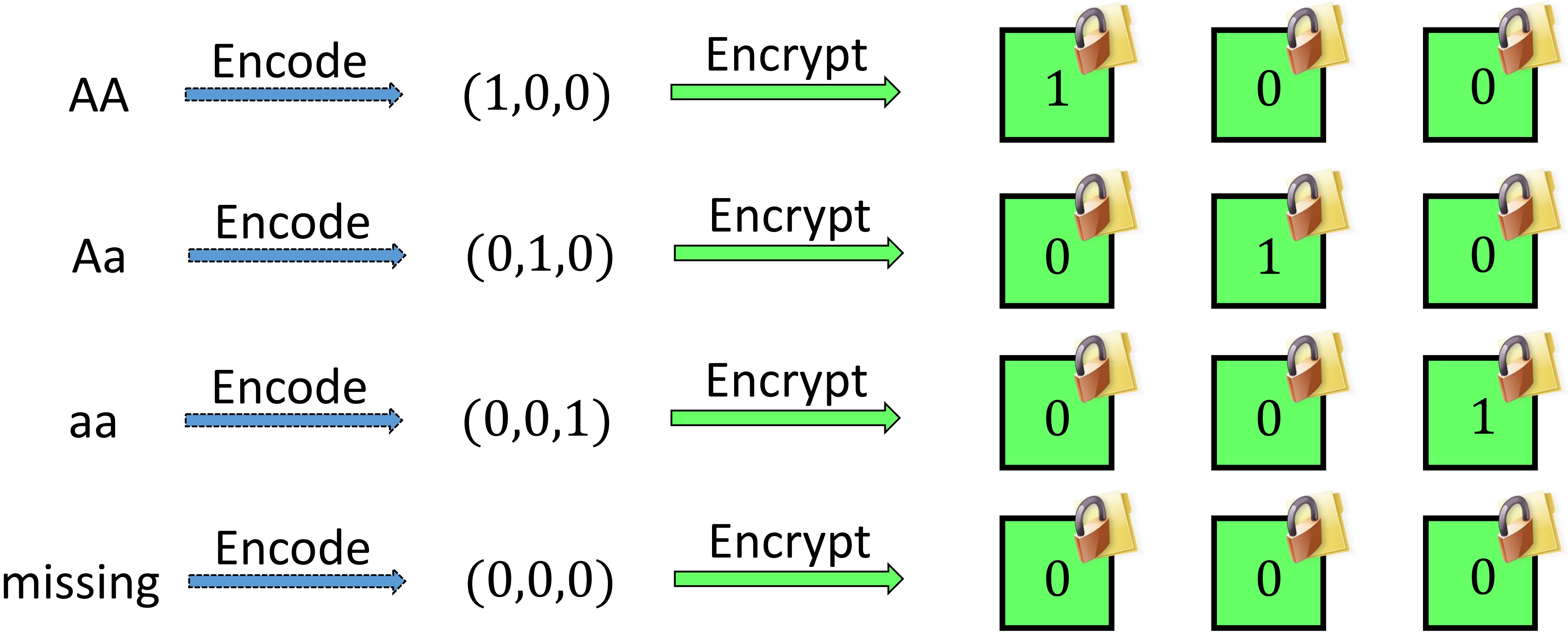
80-bit security

- Parameter set I: $n = 4096,\ q \approx 2^{192}$, ciphertext $\approx 100\text{KB}$
- Parameter set II: $n = 8192,\ q \approx 2^{384}$, ciphertext $\approx 400\text{KB}$

Operation	KeyGen	Encrypt	Add	Mult	Decrypt
Parameters I	3.6s	0.3s	0.001s	0.05s	0.04s
Parameters II	18.1s	0.8s	0.003s	0.24s	0.26s

Proof-of-concept implementation: computer algebra system Magma,
Intel Core i7 @ 3.1GHz, 64-bit Windows 8.1

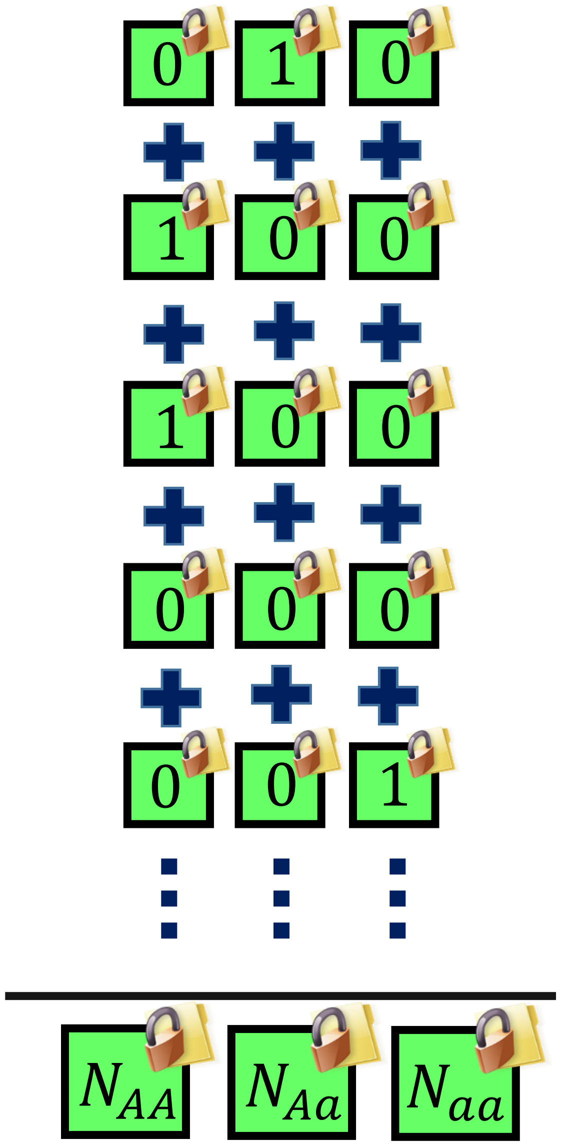
Encoding and encrypting of genotype data



Computing genotype counts

$$N_{AA} + N_{Aa} + N_{aa} = N$$

$$N_{AA} \quad N_{Aa} \quad N_{aa} \quad N$$



- Only homomorphic additions
- Cost linear in size of data sample

Pearson goodness-of-fit test

Tests for Hardy-Weinberg Equilibrium, i.e. whether allele frequencies are statistically independent

$$p_{AA} = p_A^2, p_{Aa} = 2p_A p_a, p_{aa} = p_a^2$$

- $p_{AA} = \frac{N_{AA}}{N}, p_{Aa} = \frac{N_{Aa}}{N}, p_{aa} = \frac{N_{aa}}{N}$
- Observed counts: $N_{AA}, N_{Aa}, N_{aa},$

$$p_A = \frac{2N_{AA}+N_{Aa}}{2N}, p_a = 1 - p_A$$

- Expected counts: $E_{AA} = Np_A^2, E_{Aa} = 2Np_A p_a, E_{aa} = Np_a^2$

Pearson goodness-of-fit test

- Compute the X^2 test statistic

$$X^2 = \frac{(N_{AA} - E_{AA})^2}{E_{AA}} + \frac{(N_{Aa} - E_{Aa})^2}{E_{Aa}} + \frac{(N_{aa} - E_{aa})^2}{E_{aa}}$$

- Problem: Arithmetic circuits over R do not allow divisions
- Rewrite the formula to avoid divisions

Modified algorithm

It turns out that

$$X^2 = \frac{\alpha}{2N}\left(\frac{1}{\beta_1} + \frac{1}{\beta_2} + \frac{1}{\beta_3}\right),$$

where

$$\alpha = (4N_{AA}N_{aa} - N_{Aa}^2)^2, \qquad \beta_1 = 2(2N_{AA} + N_{Aa})^2,$$

$$\beta_2 = (2N_{AA} + N_{Aa})(2N_{aa} + N_{Aa}), \qquad \beta_3 = 2(2N_{aa} + N_{Aa})^2$$

- Return encryptions of values $\alpha, \beta_1, \beta_2, \beta_3, N$
- X^2 is computed after decryption

Other algorithms

Other than the Pearson test for Hardy-Weinberg equilibrium, we implemented:

- Estimation Maximization for haplotyping (EM),
  1,2,3 iterations,
- Test for Linkage Disequilibrium (LD),
- Cochran-Armitage Test for Trend (CATT),
  case control studies.

Genetic algorithm performance

80-bit security

- Parameter set I: $n = 4096$, $q \approx 2^{192}$, ciphertext $\approx 100\text{KB}$
- Parameter set II: $n = 8192$, $q \approx 2^{384}$, ciphertext $\approx 400\text{KB}$

Algorithm	Pearson	EM (iterations)			LD	CATT
		1	2	3		
Parameters I	0.3s	0.6s	1.1s	-	0.2s	1.0s
Parameters II	1.4s	2.3s	4.5s	6.9s	0.7s	3.6s

Proof-of-concept implementation: computer algebra system Magma,
Intel Core i7 @ 3.1GHz, 64-bit Windows 8.1

Private Computation on Encrypted Genomic Data

Michael Naehrig
Cryptography Research Group
Microsoft Research

Joint work with Adriana Lopez-Alt (New York University) and Kristin Lauter (MSR)

Workshop on Genome Privacy 2014
Amsterdam, 15 July 2014