

CHIFFREMENT (COMPLÈTEMENT) HOMOMORPHE: DE LA THÉORIE À LA PRATIQUE

Tancrède Lepoint

CryptoExperts

Séminaire sur la Confiance Numérique – Jeudi 9 Octobre 2014

Outline

1. Introduction

1.1 What is Fully Homomorphic Encryption? Use Cases?

1.2 Somewhat Homomorphic Encryption over the Integers

2. Implementations and Cloud Communications

2.1 Pointers to Implementations and Libraries

2.2 Cloud Communication Issues

Outline

1. Introduction

1.1 What is Fully Homomorphic Encryption? Use Cases?

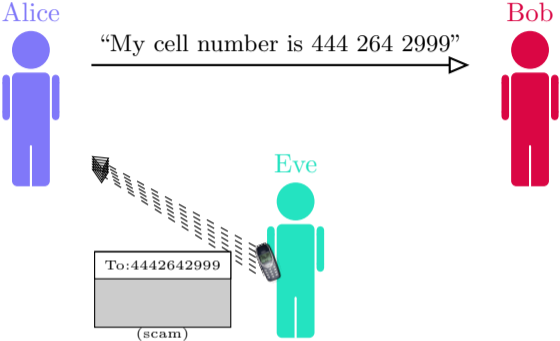
1.2 Somewhat Homomorphic Encryption over the Integers

2. Implementations and Cloud Communications

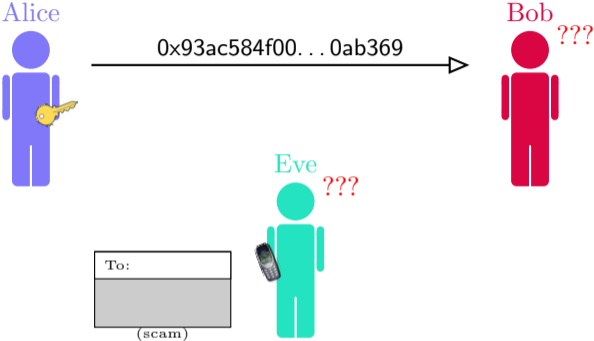
2.1 Pointers to Implementations and Libraries

2.2 Cloud Communication Issues

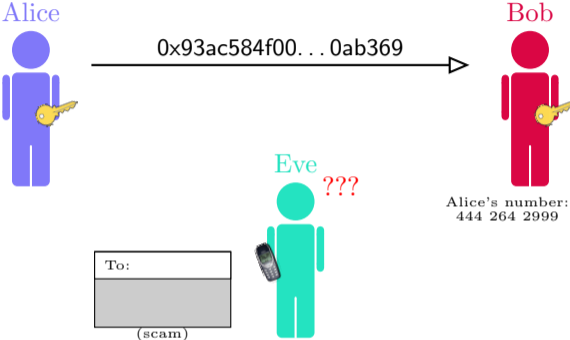
Encryption



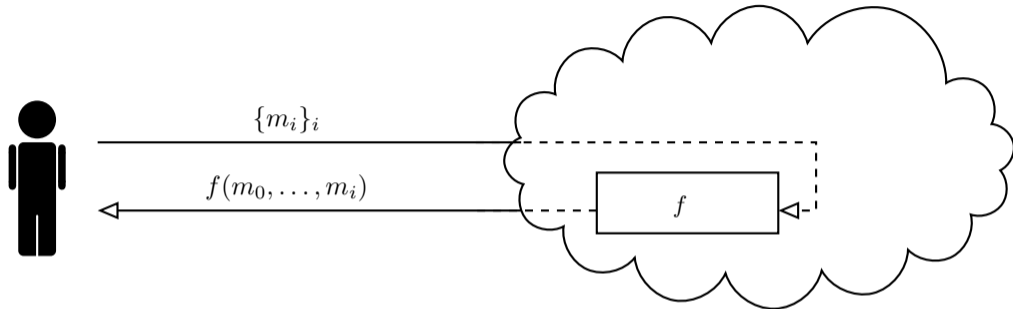
Encryption



Encryption

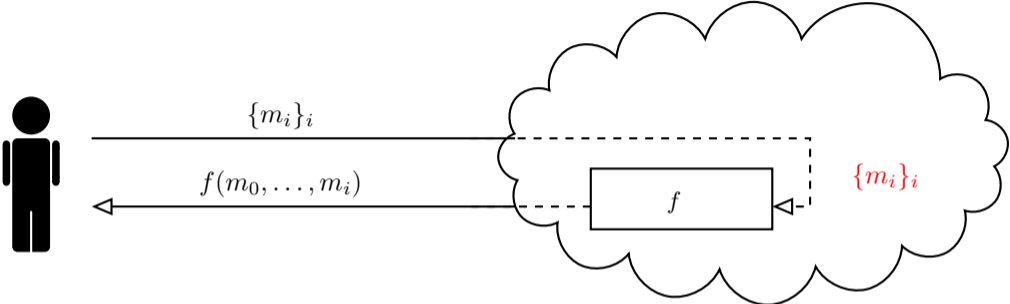


Modelization



f is the service provided by the Cloud on your data m_i

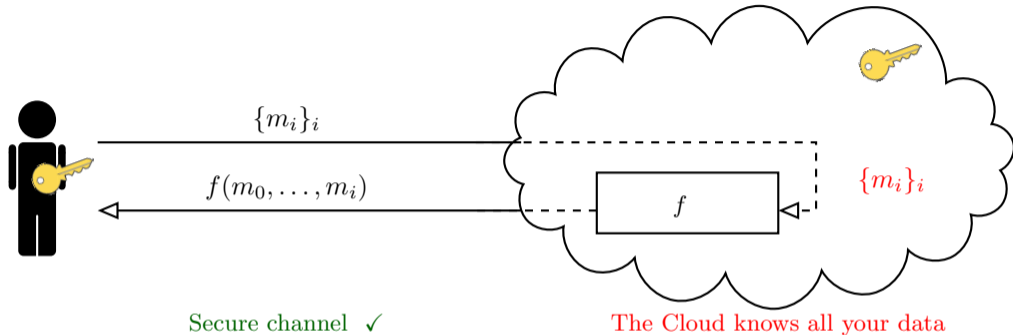
Confidentiality of Your Data



The Cloud knows all your data

Confidentiality of your data in the Cloud?

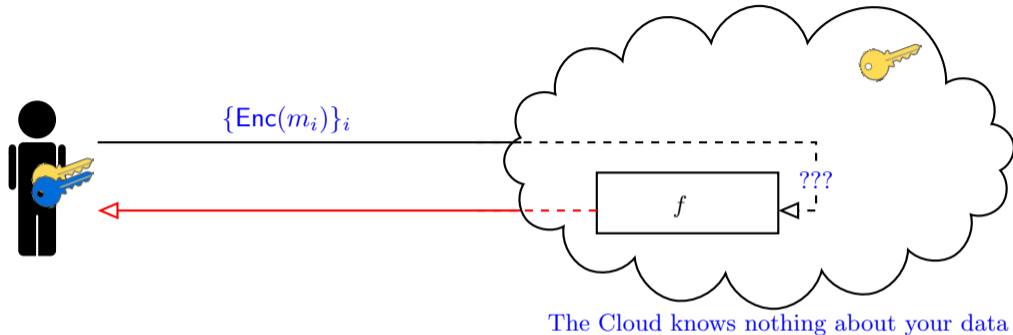
Confidentiality of Your Data



Confidentiality of your data in the Cloud?

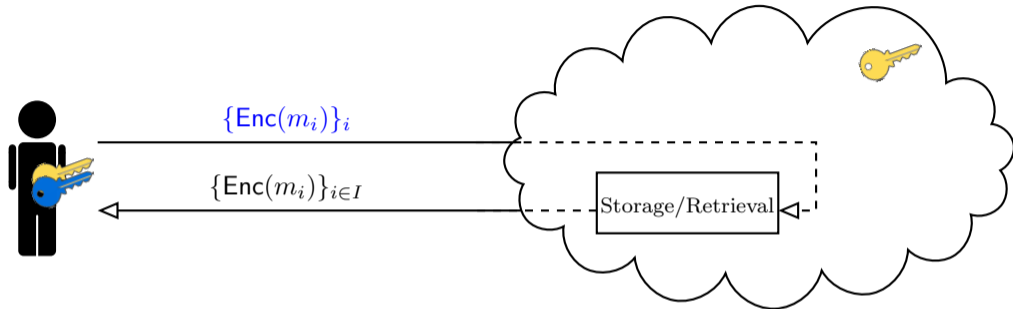
- ▶ We assume communication with the Cloud is **secure** ✓ (e.g. HTTPS)

Confidentiality w.r.t. The Cloud



- For confidentiality, we use **encryption**

Confidentiality w.r.t. The Cloud

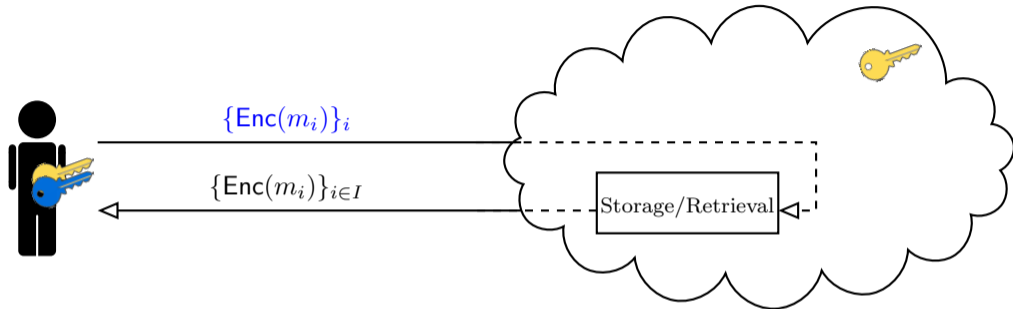


The Cloud knows nothing about your data

- ▶ For confidentiality, we use **encryption**
 - ▶ Now... limited to **storage/retrieval**



Confidentiality w.r.t. The Cloud



The Cloud knows nothing about your data

- ▶ For confidentiality, we use **encryption**
 - ▶ Now... limited to **storage/retrieval**
 - ▶ This is not even what Dropbox/Google Drive/Microsoft OneDrive/Amazon S2/iCloud Drive/etc. are doing
 - ▶ Allow access control and sharing, interaction with whole app universe, etc.

Operating on Encrypted Data

[RivestAdlemanDertouzos78]

Going beyond the storage/retrieval of encrypted data by permitting **encrypted data to be operated on** for interesting operations, **in a public fashion?**

Operating on Encrypted Data

[RivestAdlemanDertouzos78]

Going beyond the storage/retrieval of encrypted data by permitting **encrypted data to be operated on** for interesting operations, **in a public fashion?**

- ▶ **Additive** Homomorphic Encryption:

$$E = \text{Enc}(a) + \text{Enc}(b) \Rightarrow \text{Dec}(E) = a + b$$

e.g. Paillier's cryptosystem [Paillier99]

$$\begin{aligned} c &= g^m \cdot r^N \bmod N^2 \\ c' &= g^{m'} \cdot r'^N \bmod N^2 \end{aligned} \Rightarrow c \cdot c' = g^{m+m'} \cdot (r \cdot r')^N \bmod N^2$$

Operating on Encrypted Data

[RivestAdlemanDertouzos78]

Going beyond the storage/retrieval of encrypted data by permitting **encrypted data to be operated on** for interesting operations, **in a public fashion?**

- ▶ **Additive** Homomorphic Encryption:

$$E = \text{Enc}(a) + \text{Enc}(b) \Rightarrow \text{Dec}(E) = a + b$$

- ▶ **Multiplicative** Homomorphic Encryption:

$$E = \text{Enc}(a) \times \text{Enc}(b) \Rightarrow \text{Dec}(E) = a \times b$$

e.g. 'textbook ElGamal'

$$\begin{aligned} c &= \left(g^y, m \cdot (g^x)^y \right) \\ c' &= \left(g^{y'}, m' \cdot (g^x)^{y'} \right) \end{aligned} \Rightarrow c \odot c' = \left(g^{y+y'}, (m \cdot m') \cdot (g^x)^{y+y'} \right)$$

Operating on Encrypted Data

[RivestAdlemanDertouzos78]

Going beyond the storage/retrieval of encrypted data by permitting **encrypted data to be operated on** for interesting operations, **in a public fashion?**

- ▶ **Additive** Homomorphic Encryption:

$$E = \text{Enc}(a) + \text{Enc}(b) \Rightarrow \text{Dec}(E) = a + b$$

- ▶ **Multiplicative** Homomorphic Encryption:

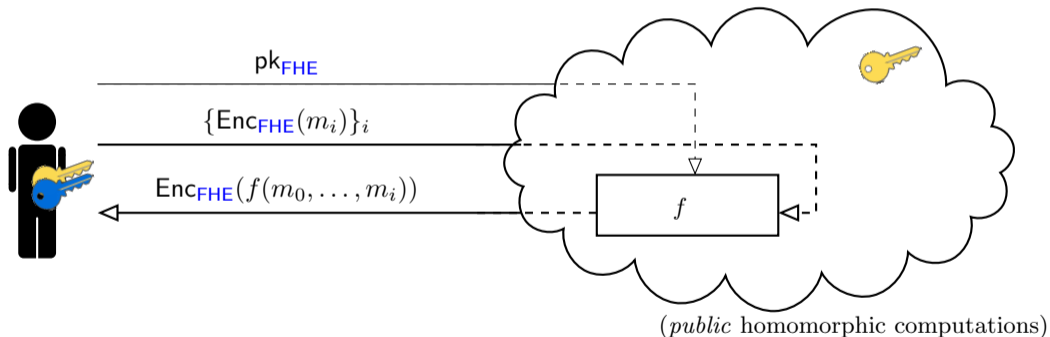
$$E = \text{Enc}(a) \times \text{Enc}(b) \Rightarrow \text{Dec}(E) = a \times b$$

FULLY Homomorphic Encryption: Additive and Multiplicative on $\{0, 1\}$

Fully Homomorphic Encryption

Enable **unlimited computation on encrypted data**

(w.l.o.g. m_i 's are bits and f Boolean circuit)



Towards Fully Homomorphic Encryption

- ▶ [RivestAdlemanDertouzos78]: notion of privacy homomorphism
- ▶ [GoldwasserMicali84]: XOR of bits
- ▶ [ElGamal84]: multiplication mod p
- ▶ [Paillier98]: addition mod $N = pq$
- ▶ [BonehGohNissim05]: additions and **one** multiplication mod p

Towards Fully Homomorphic Encryption

- ▶ [RivestAdlemanDertouzos78]: notion of privacy homomorphism
- ▶ [GoldwasserMicali84]: XOR of bits
- ▶ [ElGamal84]: multiplication mod p
- ▶ [Paillier98]: addition mod $N = pq$
- ▶ [BonehGohNissim05]: additions and **one** multiplication mod p
- ▶ [Gentry09]: additions and multiplications mod 2!

Awesome! Can We Use It?

- ▶ In **theory**, plentiful of applications
 - ▶ Everything can be viewed as a circuit
 - ▶ Humongous potential
 - ▶ Solve many problems on privacy

Awesome! Can We Use It?

- ▶ In **theory**, plentiful of applications
 - ▶ Everything can be viewed as a circuit
 - ▶ Humongous potential
 - ▶ Solve many problems on privacy
- ▶ In **practice**... problem because of sequential **homomorphic multiplications**!
 - ▶ State-of-the-art in 2011: 30 minutes after each bit-multiplication



Awesome! Can We Use It?

- ▶ In **theory**, plentiful of applications
 - ▶ Everything can be viewed as a circuit
 - ▶ Humongous potential
 - ▶ Solve many problems on privacy
- ▶ In **practice**... problem because of sequential **homomorphic multiplications**!
 - ▶ State-of-the-art in 2011: 30 minutes after each bit-multiplication
 - ▶ State-of-the-art in 2014: **not much better**... for **fully** homomorphic encryption
 - ▶ (But I heard about exciting new results to come...)



(Fully ?) Homomorphic Encryption

Question [NaehrigLauterVaikuntanathan12]:

Do we really need **fully** homomorphic encryption?

(Fully ?) Homomorphic Encryption

Question [NaehrigLauterVaikuntanathan12]:

Do we really need **fully** homomorphic encryption?

- ▶ Work over bits?
 - ▶ e.g. computing $\sum_{i=1}^{10} t_i$ where t_i are 8-bit values:
 - ▶ 135 '×' and '× depth' = 8 if working over bits [FauSirdeyFontaineAguilar-MelchorGogniat13]
 - ▶ 0 '×' if plaintext space is ≥ 2560

(Fully ?) Homomorphic Encryption

Question [NaehrigLauterVaikuntanathan12]:

Do we really need **fully** homomorphic encryption?

- ▶ Work over bits?
 - ▶ e.g. computing $\sum_{i=1}^{10} t_i$ where t_i are 8-bit values:
 - ▶ 135 '×' and '× depth' = 8 if working over bits [FauSirdeyFontaineAguilar-MelchorGogniat13]
 - ▶ 0 '×' if plaintext space is ≥ 2560
- ▶ “Real World”: **limited number of multiplications**
 - ▶ **Statistics** on medical data: mean, variance, linear regression, etc.
 - ▶ Geolocalization (Euclidean distance, etc.)

Somewhat Homomorphic Encryption

- ▶ Somewhat Homomorphic Encryption (SHE): **limited number** of homomorphic operations
- ▶ **Know in advance** the \times depth of the circuit to be evaluated

SHE is sufficient for many applications,
and this is on what we (& the community) focus on

Somewhat Homomorphic Encryption

- ▶ Somewhat Homomorphic Encryption (SHE): **limited number** of homomorphic operations
- ▶ **Know in advance** the \times depth of the circuit to be evaluated

SHE is sufficient for many applications,
and this is on what we (& the community) focus on

- ▶ Interestingly enough: FHE = (SHE that evaluates its decryption circuit) [Gentry09]
 - ▶ If $c = \text{Enc}(m)$, run **homomorphically Dec**:

$$c_{\text{result}} = \text{Enc}(\text{Dec}(c)) = \text{Enc}(\text{Dec}(\text{Enc}(m))) = \text{Enc}(m)$$

Use-Cases?

Information and Communications Technologies call for projects (H2020)

Construction of “Resource efficient, real-time, highly secure fully homomorphic cryptography” is a key challenge

- ▶ We need to focus on applications driven by **real use-cases** having **small multiplicative depth**
- ▶ Statistical Computations
 - ▶ Mean
 - ▶ Standard deviation
- ▶ Genomics (e.g. χ^2 test: statistical tests)
- ▶ Machine learning
- ▶ ...

Mean

- ▶ Cloud want to compute the **mean** on private values $\{x_1, \dots, x_n\}$

$$\bar{x} = \left(\sum_{i=1}^n x_i \right) / n$$

- ▶ SHE encryption scheme Enc (with decryption Dec)

Mean

- ▶ Cloud want to compute the **mean** on private values $\{x_1, \dots, x_n\}$

$$\bar{x} = \left(\sum_{i=1}^n x_i \right) / n$$

- ▶ SHE encryption scheme Enc (with decryption Dec)
1. We can assume that n is public, so we only need to compute $\sum_{i=1}^n x_i$

Mean

- ▶ Cloud want to compute the **mean** on private values $\{x_1, \dots, x_n\}$

$$\bar{x} = \left(\sum_{i=1}^n x_i \right) / n$$

- ▶ SHE encryption scheme Enc (with decryption Dec)
 1. We can assume that n is public, so we only need to compute $\sum_{i=1}^n x_i$
 2. The cloud has $\text{Enc}(x_1), \dots, \text{Enc}(x_n)$

Mean

- ▶ Cloud want to compute the **mean** on private values $\{x_1, \dots, x_n\}$

$$\bar{x} = \left(\sum_{i=1}^n x_i \right) / n$$

- ▶ SHE encryption scheme Enc (with decryption Dec)
 1. We can assume that n is public, so we only need to compute $\sum_{i=1}^n x_i$
 2. The cloud has $\text{Enc}(x_1), \dots, \text{Enc}(x_n)$
 3. The cloud can **homomorphically** compute and send back to me

$$X = \text{Enc}(x_1) + \dots + \text{Enc}(x_n)$$

Mean

- ▶ Cloud want to compute the **mean** on private values $\{x_1, \dots, x_n\}$

$$\bar{x} = \left(\sum_{i=1}^n x_i \right) / n$$

- ▶ SHE encryption scheme Enc (with decryption Dec)

1. We can assume that n is public, so we only need to compute $\sum_{i=1}^n x_i$
2. The cloud has $\text{Enc}(x_1), \dots, \text{Enc}(x_n)$
3. The cloud can **homomorphically** compute and send back to me

$$X = \text{Enc}(x_1) + \dots + \text{Enc}(x_n)$$

4. I can decrypt the result V :

$$\text{Dec}(X) = x_1 + \dots + x_n = \sum_{i=1}^n x_i$$

Variance

- ▶ Cloud want to compute the **variance** on private values $\{x_1, \dots, x_n\}$

$$v = \left(\sum_{i=1}^n (x_i - \bar{x})^2 \right) / n$$

- ▶ SHE encryption scheme Enc (with decryption Dec)

Variance

- ▶ Cloud want to compute the **variance** on private values $\{x_1, \dots, x_n\}$

$$v = \left(\sum_{i=1}^n (x_i - \bar{x})^2 \right) / n$$

- ▶ SHE encryption scheme Enc (with decryption Dec)
1. We can assume that n is public, so we only need to compute

$$n^3 \cdot v = n^2 \cdot \sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n \left(n \cdot x_i - \sum_{j=1}^n x_j \right)^2$$

Variance

- ▶ Cloud want to compute the **variance** on private values $\{x_1, \dots, x_n\}$

$$v = \left(\sum_{i=1}^n (x_i - \bar{x})^2 \right) / n$$

- ▶ SHE encryption scheme Enc (with decryption Dec)

1. We can assume that n is public, so we only need to compute

$$n^3 \cdot v = n^2 \cdot \sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n \left(n \cdot x_i - \sum_{j=1}^n x_j \right)^2$$

2. The cloud has $\text{Enc}(x_1), \dots, \text{Enc}(x_n)$

Variance

- ▶ Cloud want to compute the **variance** on private values $\{x_1, \dots, x_n\}$

$$v = \left(\sum_{i=1}^n (x_i - \bar{x})^2 \right) / n$$

- ▶ SHE encryption scheme Enc (with decryption Dec)

1. We can assume that n is public, so we only need to compute

$$n^3 \cdot v = n^2 \cdot \sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n \left(n \cdot x_i - \sum_{j=1}^n x_j \right)^2$$

2. The cloud has $\text{Enc}(x_1), \dots, \text{Enc}(x_n)$
3. The cloud can **homomorphically** compute and send back to me

$$V = \sum_{i=1}^n \left(\sum_{j=1}^n (\text{Enc}(x_i) - \text{Enc}(x_j)) \right) \times \left(\sum_{j=1}^n (\text{Enc}(x_i) - \text{Enc}(x_j)) \right)$$

Variance

- ▶ Cloud want to compute the **variance** on private values $\{x_1, \dots, x_n\}$

$$v = \left(\sum_{i=1}^n (x_i - \bar{x})^2 \right) / n$$

- ▶ SHE encryption scheme Enc (with decryption Dec)

1. We can assume that n is public, so we only need to compute

$$n^3 \cdot v = n^2 \cdot \sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n \left(n \cdot x_i - \sum_{j=1}^n x_j \right)^2$$

2. The cloud has $\text{Enc}(x_1), \dots, \text{Enc}(x_n)$
3. The cloud can **homomorphically** compute and send back to me

$$V = \sum_{i=1}^n \left(\sum_{j=1}^n (\text{Enc}(x_i) - \text{Enc}(x_j)) \right) \times \left(\sum_{j=1}^n (\text{Enc}(x_i) - \text{Enc}(x_j)) \right)$$

4. I can decrypt the result V and recover $\text{Dec}(V) = n^3 \cdot v$

Genomics

- ▶ Application for genomic data
Private Computation on Encrypted Genomic Data
Lauter, López-Alt, Naehrig, 2014

Global Alliance

A global alliance of government agencies, research institutes, and hospitals wants to pool all their patients' genomic data to make available for research.
<http://www.broadinstitute.org/files/news/pdfs/GAWhitePaperJune3.pdf>

- ▶ In the following: **Pearson Goodness-of-Fit to test for deviation from Hardy-Weinberg equilibrium**

Hardy-Weinberg Equilibrium (HWE)

- ▶ Population of $N = N_{AA} + N_{Aa} + N_{aa}$ people with genotypes AA , Aa or aa
- ▶ Probabilities

$$p_{AA} = \frac{N_{AA}}{N} \quad ; p_{Aa} = \frac{N_{Aa}}{N} \quad ; p_{aa} = \frac{N_{aa}}{N} \quad ; p_A = \frac{2N_{AA} + N_{Aa}}{2N} \quad ; \quad p_a = \frac{2N_{aa} + N_{Aa}}{2N}$$

Hardy-Weinberg Equilibrium (HWE)

- ▶ Population of $N = N_{AA} + N_{Aa} + N_{aa}$ people with genotypes AA , Aa or aa
- ▶ Probabilities

$$p_{AA} = \frac{N_{AA}}{N} \quad ; \quad p_{Aa} = \frac{N_{Aa}}{N} \quad ; \quad p_{aa} = \frac{N_{aa}}{N} \quad ; \quad p_A = \frac{2N_{AA} + N_{Aa}}{2N} \quad ; \quad p_a = \frac{2N_{aa} + N_{Aa}}{2N}$$

A gene is said to be in **HWE** if its allele frequencies are independent

- ▶ HWE:

$$p_{AA} = p_A^2 \quad ; \quad p_{Aa} = p_A p_a \quad ; \quad p_{aa} = p_a^2$$

Pearson Goodness-Of-Fit Test: χ^2 test

- ▶ If the alleles are independent (i.e. HWE), then

$$\mathbb{E}_{AA} = N \cdot p_A^2 \quad ; \quad \mathbb{E}_{Aa} = 2N \cdot p_A p_a \quad ; \quad \mathbb{E}_{aa} = N \cdot p_a^2$$

Pearson Goodness-Of-Fit Test: χ^2 test

- ▶ If the alleles are independent (i.e. HWE), then

$$\mathbb{E}_{AA} = N \cdot p_A^2 \quad ; \quad \mathbb{E}_{Aa} = 2N \cdot p_A p_a \quad ; \quad \mathbb{E}_{aa} = N \cdot p_a^2$$

- ▶ Compare the X^2 test-statistic below to the χ^2 -statistic with 1 degree of freedom

$$X^2 = \sum_{i \in \{AA, Aa, aa\}} \frac{(N_i - \mathbb{E}_i)^2}{\mathbb{E}_i}$$

- ▶ Can be rewritten as previously so that the multiplicative depth is 2
 - ▶ **Can be done homomorphically in an efficient manner!**

Pearson Goodness-Of-Fit Test: χ^2 test

- ▶ If the alleles are independent (i.e. HWE), then

$$\mathbb{E}_{AA} = N \cdot p_A^2 \quad ; \quad \mathbb{E}_{Aa} = 2N \cdot p_A p_a \quad ; \quad \mathbb{E}_{aa} = N \cdot p_a^2$$

- ▶ Compare the X^2 test-statistic: $\sum_{i \in \{AA, Aa, aa\}} \frac{(O_i - \mathbb{E}_i)^2}{\mathbb{E}_i}$ with the degree of freedom

**Rough timing:
1 second for 1'000 encrypted genotypes**

- ▶ Can be rewritten as previously so that the multiplicative depth is 2
 - ▶ **Can be done homomorphically in an efficient manner!**

Lots of consequences on the privacy, and how this interacts with the European laws.



Questions before the first (conceptually simple) construction?

Simple SHE: DGHV Scheme [vDGHV10]

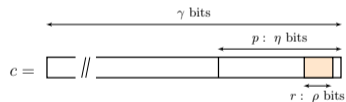
- ▶ Public error-free element: $x_0 = q_0 \cdot p$
- ▶ Secret key $sk = p$

Simple SHE: DGHV Scheme [vDGHV10]

- ▶ Public error-free element: $x_0 = q_0 \cdot p$
- ▶ Secret key $sk = p$
- ▶ Ciphertext for $m \in \{0, 1\}$:

$$c = q \cdot p + 2 \cdot r + m$$

where q large random, r small random

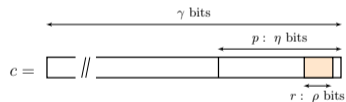


Simple SHE: DGHV Scheme [vDGHV10]

- ▶ Public error-free element: $x_0 = q_0 \cdot p$
- ▶ Secret key $sk = p$
- ▶ Ciphertext for $m \in \{0, 1\}$:

$$c = q \cdot p + 2 \cdot r + m$$

where q large random, r small random



- ▶ Decryption of c :

$$m = (c \bmod p) \bmod 2$$

Homomorphic Properties

- ▶ How to Add and Multiply Encrypted Bits:

- ▶ Add/Mult two near-multiples of p gives a near-multiple of p

- ▶ $c_1 = q_1 \cdot p + 2 \cdot r_1 + m_1, \quad c_2 = q_2 \cdot p + 2 \cdot r_2 + m_2$

- ▶ $c_1 + c_2 = p \cdot (q_1 + q_2) + \underbrace{2 \cdot (r_1 + r_2) + m_1 + m_2}_{\text{mod } 2 \rightarrow m_1 \text{ XOR } m_2}$

- ▶ $c_1 \cdot c_2 = p \cdot (c_2 q_1 + c_1 q_2 - q_1 q_2) + \underbrace{2 \cdot (2 r_1 r_2 + r_2 m_1 + r_1 m_2) + m_1 \cdot m_2}_{\text{mod } 2 \rightarrow m_1 \text{ AND } m_2}$

Homomorphic Properties

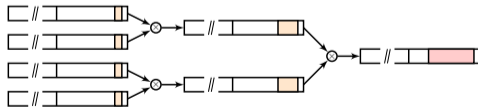
► How to Add and Multiply Encrypted Bits:

► Add/Mult two near-multiples of p gives a near-multiple of p

► $c_1 = q_1 \cdot p + 2 \cdot r_1 + m_1, \quad c_2 = q_2 \cdot p + 2 \cdot r_2 + m_2$

► $c_1 + c_2 = p \cdot (q_1 + q_2) + \underbrace{2 \cdot (r_1 + r_2) + m_1 + m_2}_{\text{mod } 2 \rightarrow m_1 \text{ XOR } m_2}$

► $c_1 \cdot c_2 = p \cdot (c_2 q_1 + c_1 q_2 - q_1 q_2) + \underbrace{2 \cdot (2r_1 r_2 + r_2 m_1 + r_1 m_2) + m_1 \cdot m_2}_{\text{mod } 2 \rightarrow m_1 \text{ AND } m_2}$



Correctness for multiplicative depth of $L: \log_2 p = \eta \approx 2^L \cdot (\rho + 1)$

Numerical Example

- ▶ $p = 541, q_0 = 809 \Rightarrow x_0 = 437669$
- ▶ noise size: $\rho = 4$

Numerical Example

- ▶ $p = 541, q_0 = 809 \Rightarrow x_0 = 437669$
- ▶ noise size: $\rho = 4$

Encryption:

- ▶ $c_1 = 737 \cdot 541 + 2 \cdot 6 + 1 = 398730$
- ▶ $c_2 = 368 \cdot 541 + 2 \cdot 9 + 0 = 199106$

Numerical Example

- ▶ $p = 541, q_0 = 809 \Rightarrow x_0 = 437669$
- ▶ noise size: $\rho = 4$

Encryption:

- ▶ $c_1 = 737 \cdot 541 + 2 \cdot 6 + 1 = 398730$
- ▶ $c_2 = 368 \cdot 541 + 2 \cdot 9 + 0 = 199106$

Addition and Multiplication:

- ▶ $c_3 = c_1 + c_2 \bmod x_0 = (398730 + 199106) \bmod 437669 = 160167$
- ▶ $c_4 = c_1 \cdot c_2 \bmod x_0 = (398730 \cdot 199106) \bmod 437669 = 317801$

Numerical Example

- ▶ $p = 541, q_0 = 809 \Rightarrow x_0 = 437669$
- ▶ noise size: $\rho = 4$

Encryption:

- ▶ $c_1 = 737 \cdot 541 + 2 \cdot 6 + 1 = 398730$
- ▶ $c_2 = 368 \cdot 541 + 2 \cdot 9 + 0 = 199106$

Addition and Multiplication:

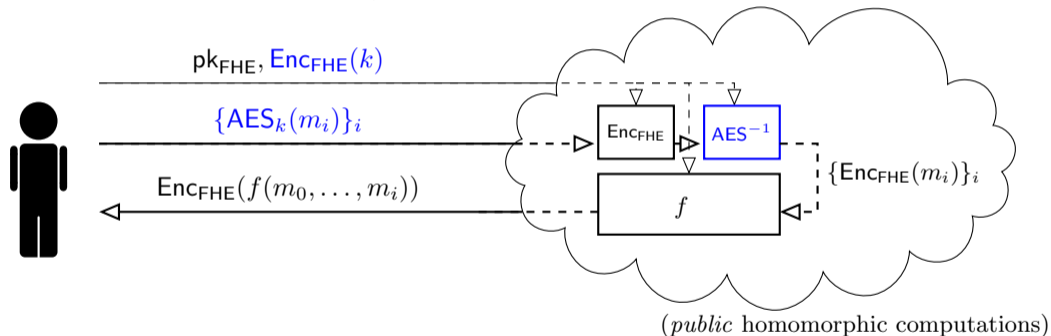
- ▶ $c_3 = c_1 + c_2 \bmod x_0 = (398730 + 199106) \bmod 437669 = 160167$
- ▶ $c_4 = c_1 \cdot c_2 \bmod x_0 = (398730 \cdot 199106) \bmod 437669 = 317801$

Decryption:

- ▶ $c_3 \bmod p = 160167 \bmod 541 = 31 = 2 \cdot 10 + 1 = 2 \cdot 10 + (1 \text{ XOR } 0)$
- ▶ $c_4 \bmod p = 317801 \bmod 541 = 234 = 2 \cdot 117 + 0 = 2 \cdot 10 + (1 \text{ AND } 0)$

Implementations

- ▶ Implementation of bit-encryption scheme:
<https://github.com/coron/fhe>
- ▶ Benchmark on a nontrivial, not astronomical circuit: AES



Implementations

- ▶ Implementation of bit-encryption scheme:
<https://github.com/coron/fhe>
- ▶ Benchmark on a nontrivial, not astronomical circuit: AES
- ▶ Batch DGHV (with bootstrapping) [CCKLLTY13]

λ	γ	ℓ	Mult	Bootstrapping	AES	Relative time
72	2.9MB	544	0.68 s	225 s	113 h	768 s
80	–	–	–	–	–	–

Implementations

- ▶ Implementation of bit-encryption scheme:
<https://github.com/coron/fhe>
- ▶ Benchmark on a nontrivial, not astronomical circuit: AES
- ▶ Batch DGHV (with bootstrapping) [CCKLLTY13]

λ	γ	ℓ	Mult	Bootstrapping	AES	Relative time
72	2.9MB	544	0.68 s	225 s	113 h	768 s
80	–	–	–	–	–	–

- ▶ Scale-Invariant DGHV (without bootstrapping) [CLT14]

λ	γ	ℓ	Mult	Convert	AES	Relative time
72	2MB	569	0.1 s	33 s	3.6 h	23 s
80	4.5MB	1875	0.3 s	277 s	102 h	195 s

Implementations

- ▶ Implementation of bit-encryption scheme:
<https://github.com/coron/fhe>
- ▶ Benchmark on a nontrivial, not astronomical circuit: AES
- ▶ Batch DGHV (with bootstrapping) [CCKLLTY13]

λ	γ	ℓ	Mult	Bootstrapping	AES	Relative time
72	2.9MB	544	0.68 s	225 s	113 h	768 s
80	–	–	–	–	–	–

- ▶ Scale-Invariant DGHV (without bootstrapping) [CLT14]

λ	γ	ℓ	Mult	Convert	AES	Relative time
72	2MB	569	0.1 s	33 s	3.6 h	23 s
80	4.5MB	1875	0.3 s	277 s	102 h	195 s

- ▶ Lattice-Based Scheme [GHS12]

λ	Ciphertext size	ℓ	AES	Relative time
80	0.3 MB	720	65 h	300 s

Outline

1. Introduction

1.1 What is Fully Homomorphic Encryption? Use Cases?

1.2 Somewhat Homomorphic Encryption over the Integers

2. Implementations and Cloud Communications

2.1 Pointers to Implementations and Libraries

2.2 Cloud Communication Issues

Some Libraries for C/C++ implementations

- ▶ **GMP:** GNU Multiple Precision Arithmetic Library
<https://gmplib.org/>
- ▶ **NTL:** A Library for doing Number Theory
<http://www.shoup.net/ntl/>
 - ▶ Not thread safe...
 - ▶ Fork of NTL: newNTL
(<http://www.prism.uvsq.fr/~gama/newntl.html>)
- ▶ **FLINT:** Fast Library for Number Theory
<http://www.flintlib.org/>
 - ▶ LOTS of dependencies...
- ▶ **OpenMP:** library for easy parallelization
<http://openmp.org/>
 - ▶ Does not work easily with **clang** yet...

Do It Yourself?

Table: YASHE with parameters $R = \mathbf{Z}[x]/(x^{4096} + 1)$, $q = 2^{127} - 1$, $w = 2^{32}$, $t = 2^{10}$ on an Intel Core i7-2600 at 3.4 GHz with hyper-threading turned off and over-clocking ('turbo boost') disabled

	KeyGen	Encrypt	Add	Mult	KeySwitch	Decrypt
[LN14] (FLINT)	3.4s	16ms	0.7ms	18ms	31ms	15ms
[BLLN13] (Home-made)	?	23ms	0.020ms	27ms		4.3ms

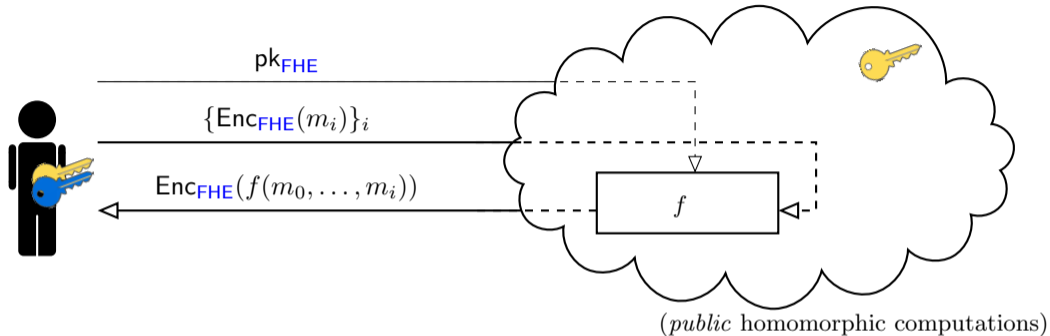
- ▶ **Might be interesting:** not too many functions to implement
 - ▶ If $q \equiv 1 \pmod{2n}$ prime and $n = 2^k$: very efficient FFT
 - ▶ More work for general rings $R = \mathbf{Z}[X]/(\phi_d(X))$ with cyclotomic polynomial ϕ_d

Public Implementations of FHE?

Unfortunately, **few implementations** are available to play with...

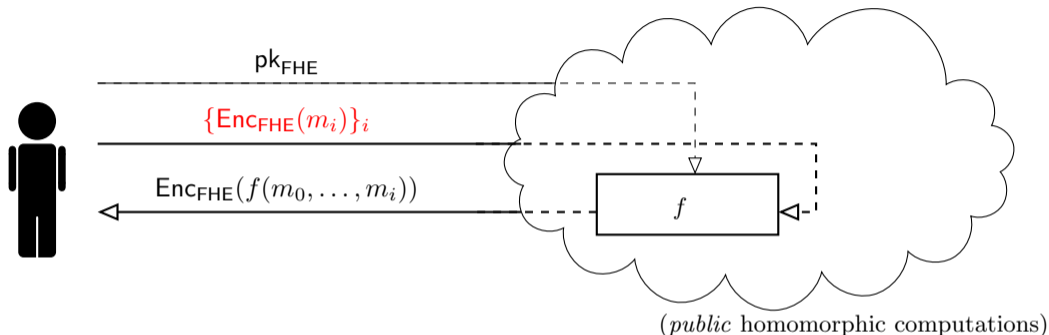
- ▶ **SV** [SV10]: <http://www.hcrypt.com>
 - ▶ Quite inefficient...
- ▶ **DGHV** [CNT12]: <https://github.com/coron/fhe>
 - ▶ In SAGE
- ▶ **BGV** [BGV12]: <https://github.com/shaih/HElib>
 - ▶ Uses NTL
- ▶ **YASHE** and **FV** [LN14]:
<https://github.com/tlepoint/homomorphic-simon>
 - ▶ Uses FLINT

Reducing Communication with the Cloud



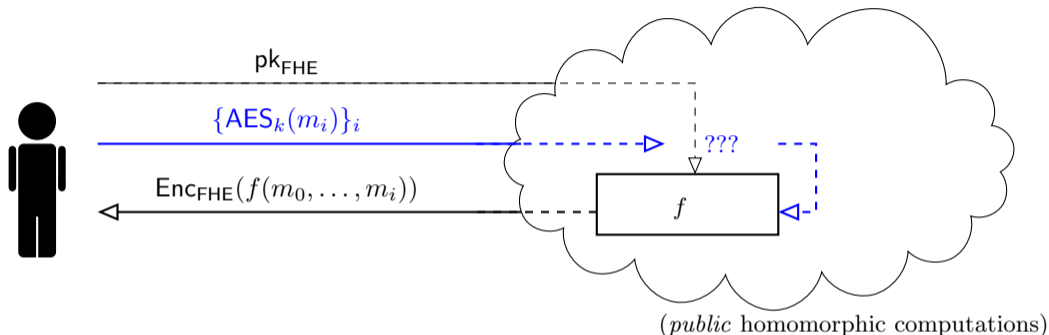
- ▶ Typical high-level FHE use-case

Reducing Communication with the Cloud



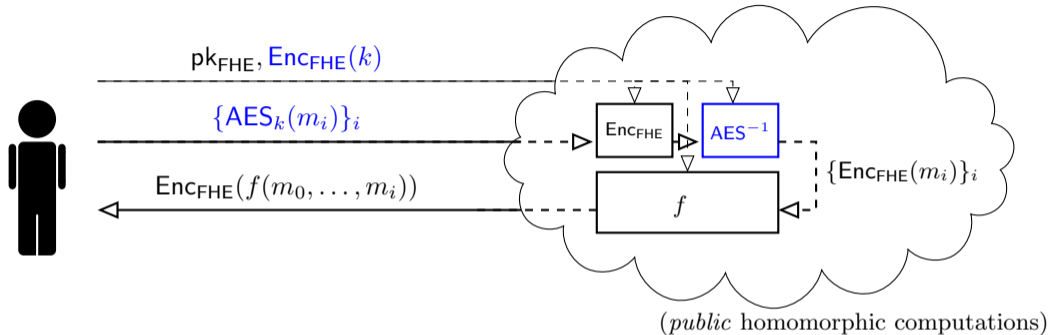
- ▶ Typical high-level FHE use-case
- ▶ ... wait a sec! The ciphertext expansion is HUGE (prohibitive!)
 - ▶ If m_i is a 4MB image, using previous schemes, the user would have to send around **200/300GB** of encrypted data

Reducing Communication with the Cloud



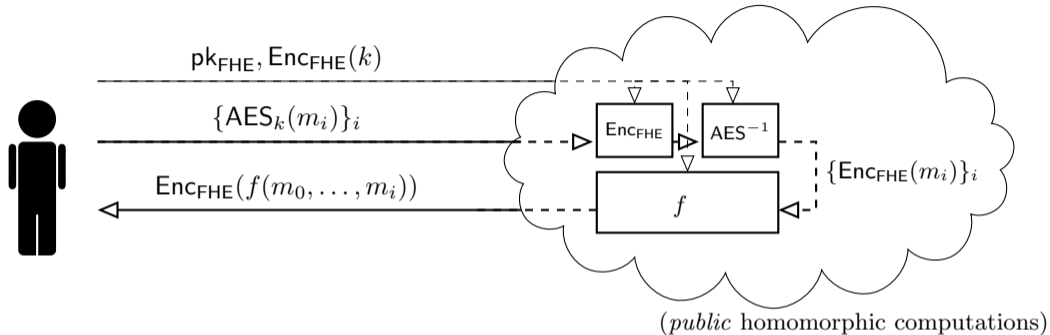
- ▶ Typical high-level FHE use-case
- ▶ ... wait a sec! The ciphertext expansion is HUGE (prohibitive)!
- ▶ What if we use hybrid encryption? [NaehrigLauterVaikuntanathan12]
 - ▶ e.g. AES does not have ciphertext expansion

Reducing Communication with the Cloud



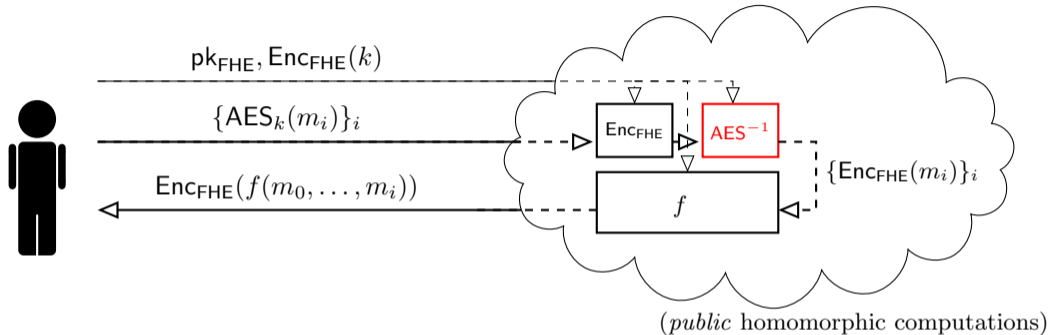
- ▶ Typical high-level FHE use-case
- ▶ ... wait a sec! The ciphertext expansion is HUGE (prohibitive)!
- ▶ What if we use hybrid encryption? [NaehrigLauterVaikuntanathan12]
 - ▶ e.g. AES does not have ciphertext expansion
 - ▶ It works :)
 - ▶ Network communication from user to cloud essentially optimal

Latency of Homomorphic AES



- **Latency of homomorphic eval.:** time to get the result

Latency of Homomorphic AES



- ▶ **Latency** of homomorphic eval.: time to get the result
- ▶ **Latency of homomorphic AES: dozens of hours**
 - ▶ I'm not even considering the function f ...

Replacing AES?

- ▶ Three implementations published [GentryHaleviSmart12, CheonCoronKimLeeLTibouchiYun13, CoronLTibouchi14]
 - ▶ Perform ℓ AES in parallel (several plaintexts in one ciphertext)
 - ▶ Running times: ≈ 100 hours
 - ▶ Time per AES block: ≤ 5 minutes

Replacing AES?

- ▶ Three implementations published [GentryHaleviSmart12, CheonCoronKimLeeLTibouchiYun13, CoronLTibouchi14]
 - ▶ Perform ℓ AES in parallel (several plaintexts in one ciphertext)
 - ▶ Running times: ≈ 100 hours
 - ▶ Time per AES block: ≤ 5 minutes
- ▶ AES is not too complicated, but is **not a trivial circuit!**
 - ▶ Multiplicative depth of the binary circuit: **40** (4 per S-box)
 - ▶ Non-linear part: $b \mapsto b^{2^{54}}$ in $GF(2^8)$

Replacing AES?

- ▶ Three implementations published [GentryHaleviSmart12, CheonCoronKimLeeLTibouchiYun13, CoronLTibouchi14]
 - ▶ Perform ℓ AES in parallel (several plaintexts in one ciphertext)
 - ▶ Running times: ≈ 100 hours
 - ▶ Time per AES block: ≤ 5 minutes
- ▶ AES is not too complicated, but is **not a trivial circuit!**
 - ▶ Multiplicative depth of the binary circuit: **40** (4 per S-box)
 - ▶ Non-linear part: $b \mapsto b^{2^{54}}$ in $GF(2^8)$

We know the constraints of FHE/SWHE: can we choose something better than AES? (with small multiplicative depth)

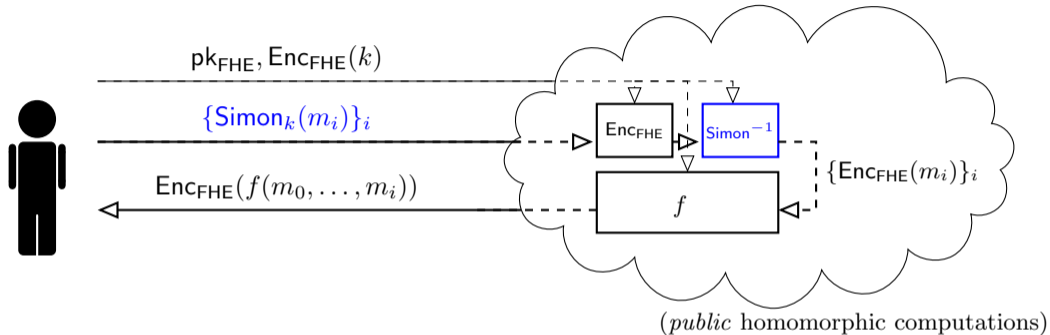
Replacing AES?

- ▶ Three implementations published [GentryHaleviSmart12, CheonCoronKimLeeLTibouchiYun13, CoronLTibouchi14]
 - ▶ Perform ℓ AES in parallel (several plaintexts in one ciphertext)
 - ▶ Running times: ≈ 100 hours
 - ▶ Time per AES block: ≤ 5 minutes
- ▶ AES is not too complicated, but is **not a trivial circuit!**
 - ▶ Multiplicative depth of the binary circuit: **40** (4 per S-box)
 - ▶ Non-linear part: $b \mapsto b^{2^{54}}$ in $GF(2^8)$

We know the constraints of FHE/SWHE: can we choose something better than AES? (with small multiplicative depth)

- ▶ Resemble some hardware/masking constraints (but is different): reduce the number of multiplications

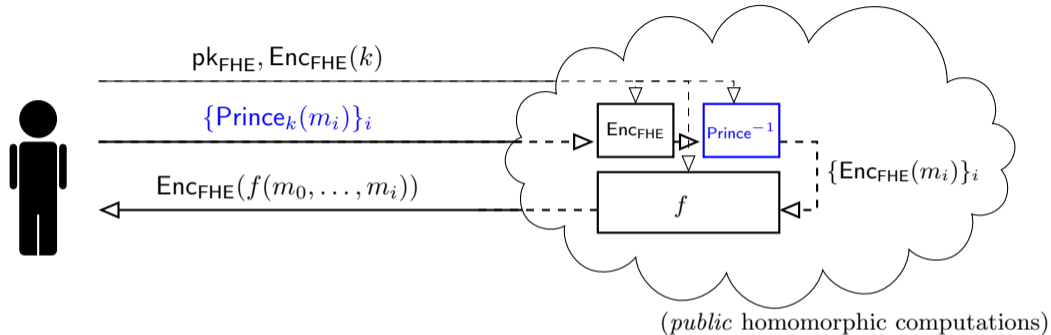
Lightweight Block Ciphers?



Maybe we could consider lightweight block ciphers?

- ▶ Independently done for **Simon** [LNaehrig14] and Prince [DorözShahverdiEisenbarthSunar14]

Lightweight Block Ciphers?



Maybe we could consider lightweight block ciphers?

- ▶ Independently done for Simon [LNaehrig14] and **Prince** [DorözShahverdiEisenbarthSunar14]

Benchmarks

- ▶ Hard to compare (not same schemes/same computers/same programming languages)

Rough idea:

Scheme	Block Size	Number of cores	Latency
AES	128	4	30-100h
Simon	64	4	3 min
Simon	64	1	12 min
Simon	128	4	1h
Prince	128	1	1h

- ▶ Some parallelization is possible
 - ▶ AES easily up to 16 cores
 - ▶ Simon easily up to block size/2 cores
 - ▶ Prince up to 32 cores

Benchmarks

- ▶ Hard to compare (not same schemes/same computers/same programming languages)

Rough idea:

Scheme	Block Size	Number of cores	Latency
AES	128	4	30-100h
Simon Simon Simon	☺ PoC Implementation available at https://github.com/tlepoint/homomorphic-simon		
Prince	128	1	1h

- ▶ Some parallelization is possible
 - ▶ AES easily up to 16 cores
 - ▶ Simon easily up to block size/2 cores
 - ▶ Prince up to 32 cores

Mainstream Subject & Lots of Open Questions

- ▶ Current best choice: Prince (multiplicative depth of 24)
- ▶ The community is working on the subject

Mainstream Subject & Lots of Open Questions

- ▶ Current best choice: Prince (multiplicative depth of 24)
- ▶ The community is working on the subject

Lots of open questions

- ▶ **Do we really need a block cipher?** (wrt to PK scheme, RNG?)

Mainstream Subject & Lots of Open Questions

- ▶ Current best choice: Prince (multiplicative depth of 24)
- ▶ The community is working on the subject

Lots of open questions

- ▶ **Do we really need a block cipher?** (wrt to PK scheme, RNG?)
- ▶ **What is the security/attack models?** (who attacks? What do we want to avoid?)

Mainstream Subject & Lots of Open Questions

- ▶ Current best choice: Prince (multiplicative depth of 24)
- ▶ The community is working on the subject

Lots of open questions

- ▶ **Do we really need a block cipher?** (wrt to PK scheme, RNG?)
- ▶ **What is the security/attack models?** (who attacks? What do we want to avoid?)
- ▶ **What are the conditions we want on the block cipher?** (e.g. resistance to related key does not seem required?)

Mainstream Subject & Lots of Open Questions

- ▶ Current best choice: Prince (multiplicative depth of 24)
- ▶ The community is working on the subject

Lots of open questions

- ▶ **Do we really need a block cipher?** (wrt to PK scheme, RNG?)
- ▶ **What is the security/attack models?** (who attacks? What do we want to avoid?)
- ▶ **What are the conditions we want on the block cipher?** (e.g. resistance to related key does not seem required?)
- ▶ **How to exploit FHE constraints?** (It is not only the multiplicative depth that is interesting to reduce)

Mainstream Subject & Lots of Open Questions

- ▶ Current best choice: Prince (multiplicative depth of 24)
- ▶ The community is working on the subject

Lots of open questions



















- ▶ **Do we really need a block cipher?** (wrt to PK scheme, RNG?)
- ▶ **What is the security/attack models?** (who attacks? What do we want to avoid?)
- ▶ **What are the conditions we want on the block cipher?** (e.g. resistance to related key does not seem required?)
- ▶ **How to exploit FHE constraints?** (It is not only the multiplicative depth that is interesting to reduce)
- ▶ **Reciprocally, can we design FHE schemes specially adapted to certain schemes/algorithms?**



<https://www.cryptoexperts.com/tlepoint>

CRYPTOEXPERTS 

(Sparse) Bibliography

-  [Gen09] [Fully Homomorphic Encryption using Ideal Lattices](#)
-  [DGHV10] [Fully Homomorphic Encryption over the Integers](#)
-  [BV11] [Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages](#)
-  [CMNT11] [Fully Homomorphic Encryption over the Integers with Shorter Public Keys](#)
-  [CNT12] [Public Key Compression and Modulus Switching for Fully Homomorphic Encryption over the Integers](#)
-  [BGV12] [\(Leveled\) Fully Homomorphic Encryption without Bootstrapping](#)
-  [FV12] [Somewhat Practical Fully Homomorphic Encryption](#)
-  [GHS12] [Homomorphic Evaluation of the AES Circuit](#)
-  [LTV12] [On-the-fly Multiparty Computation on the Cloud via multikey Fully Homomorphic Encryption](#)
-  [NLV12] [Can Homomorphic Encryption be Practical?](#)
-  [BLLN13] [Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme](#)
-  [LP13] [On the Minimal Number of Bootstrappings in Homomorphic Circuits](#)
-  [CCKLLTY13] [Batch Fully Homomorphic Encryption over the Integers](#)
-  [GSW13] [Homomorphic Encryption from Learning With Errors: Conceptually-simpler, Asymptotically-faster, Attribute-based](#)
-  [CLT14] [Scale-Invariant Fully Homomorphic Encryption over the Integers](#)
-  [LN14] [A Comparison of the Homomorphic Encryption Schemes FV and YASHE](#)
-  [DSES14] [Toward Practical Homomorphic Evaluation of Block Ciphers using Prince](#)
-  [BV14] [Lattice-Based FHE as Secure as PKE](#)