FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

A survey on fully homomorphic encryption with statistical applications

Rener Oliveira
Rodrigo Targino

CNMAC / Bonito, MS / September 18th, 2023

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Author/Advisor

Rener Oliveira

- Applied Math BSc, EMAp/FGV (2022)
- Quantitative Research Analyst, BOCOM BBM Bank (2021-)
- OpenFHE Developer (2023-)

Rodrigo Targino

- PhD in Statistics, UCL (2017)
- Assistant Professor EMAp/FGV (2017-)
- Bayesian Statistics
- Financial and Actuarial Risk Management

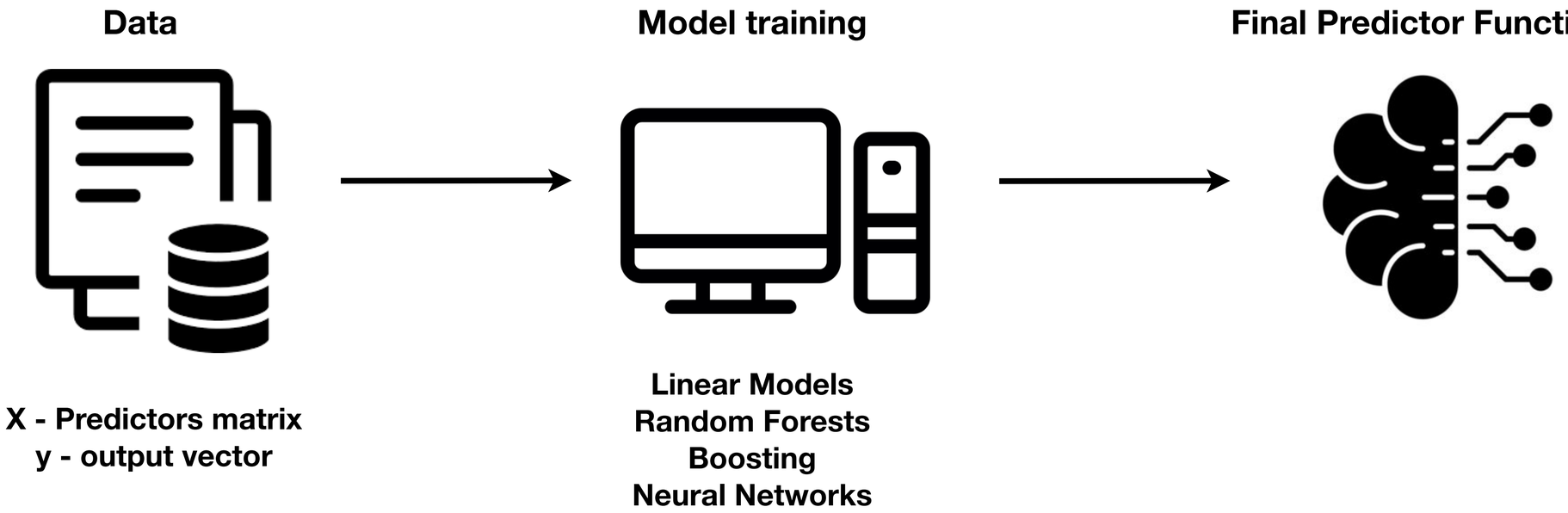
FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Table of Contents

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Introduction

Supervised machine learning setup:



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Introduction

Supervised machine learning setup:

Data

X - Predictors matrix
y - output vector

What if Data is sensitive?

- Medical records
- Financial data
- Genetic data

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Introduction

Supervised machine learning setup:

Encrypted Data

%$*#!7E% - Encrypted predictors
$&h%$8 - Encrypted target vector

What if Data is sensitive?

- Medical records
- Financial data
- Genetic data

Just encrypt it ;-)

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Introduction

Supervised machine learning setup:

Encrypted Data

%$*#!7E% - Encrypted predictors
$&h%$8 - Encrypted target vector

What if Data is sensitive?

- Medical records
- Financial data
- Genetic data

Just encrypt it ;-)

But, can we still train the models over encrypted data?

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Introduction

Supervised machine learning setup:

Encrypted Data

%$*#!7E% - Encrypted predictors
$&h%$8 - Encrypted target vector

What if Data is sensitive?

- Medical records
- Financial data
- Genetic data

Just encrypt it ;-)

But, can we still train the models over encrypted data?

Maybe :)

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Privacy Homomorphisms, 1978

ON DATA BANKS AND PRIVACY HOMOMORPHISMS

Ronald L. Rivest
Len Adleman
Michael L. Dertouzos

Massachusetts Institute of Technology
Cambridge, Massachusetts

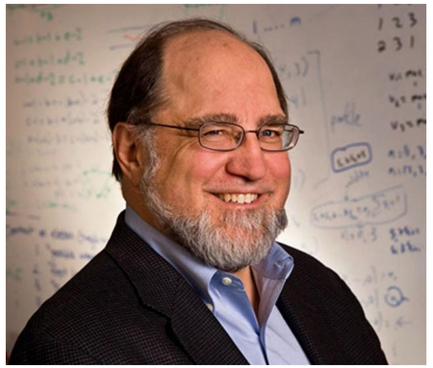
FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Privacy Homomorphisms, 1978

ON DATA BANKS AND PRIVACY HOMOMORPHISMS

Ronald L. Rivest
Len Adleman
Michael L. Dertouzos

Massachusetts Institute of Technology
Cambridge, Massachusetts



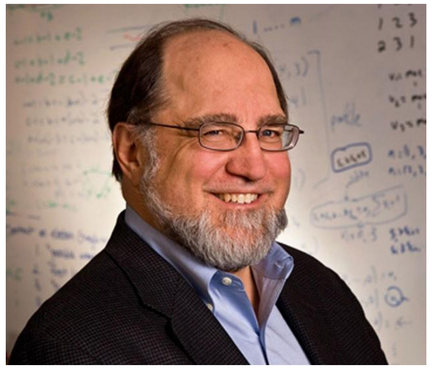
FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Privacy Homomorphisms, 1978

ON DATA BANKS AND PRIVACY HOMOMORPHISMS

Ronald L. Rivest
Len Adleman
Michael L. Dertouzos

Massachusetts Institute of Technology
Cambridge, Massachusetts



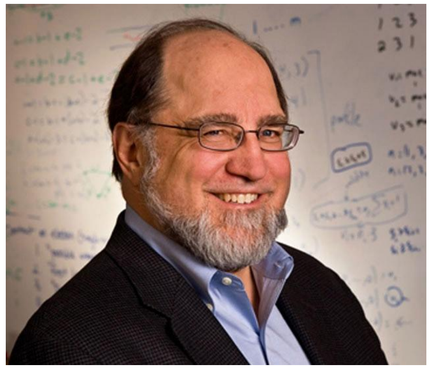
FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Privacy Homomorphisms, 1978

ON DATA BANKS AND PRIVACY HOMOMORPHISMS

Ronald L. Rivest
Len Adleman
Michael L. Dertouzos

Massachusetts Institute of Technology
Cambridge, Massachusetts



limitations on what can be accomplished, we shall see that it appears likely that there exist encryption functions which permit encrypted data to be operated on without preliminary decryption of the operands, for many sets of interesting operations. These special encryption functions we call "privacy homomorphisms"; they form an interesting subset of arbitrary encryption schemes

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Privacy Homomorphisms, 1978

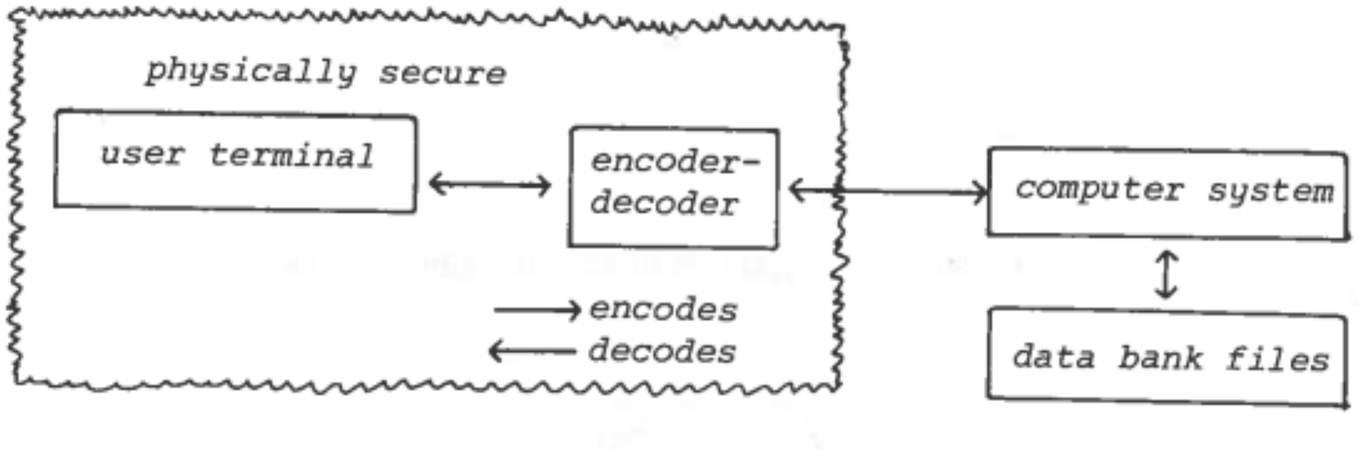


Figure 1

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

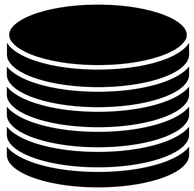
Fully Homomorphic Encryption

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Fully Homomorphic Encryption

How to <u>securely</u> compute f(m1,m2,...) ?

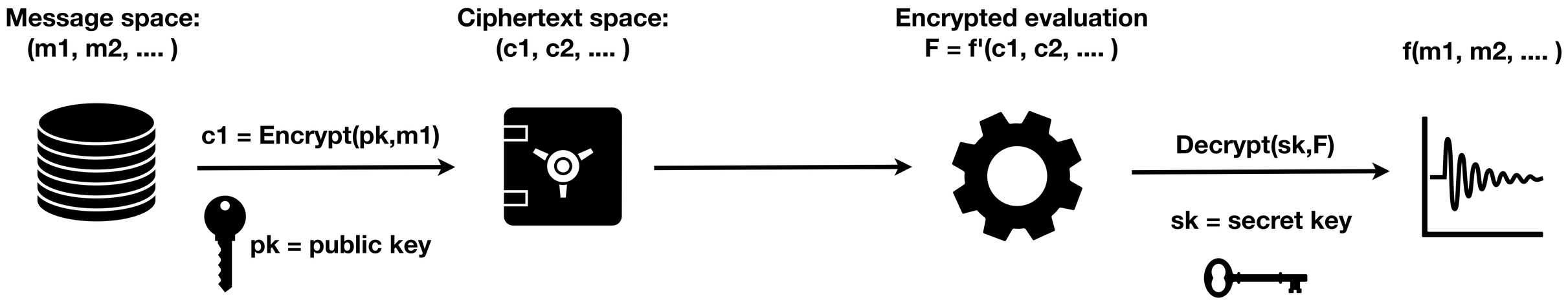
Message space:
(m1, m2, ...)



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Fully Homomorphic Encryption

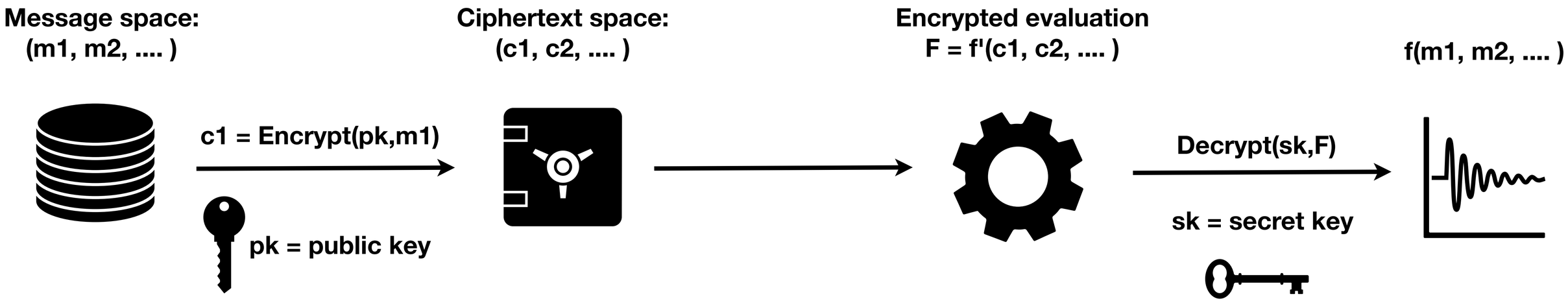
How to <u>securely</u> compute f(m1,m2,...) ?



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Fully Homomorphic Encryption

How to <u>securely</u> compute f(m1,m2,...) ?



**An encryption scheme is called Fully Homomorphic,
if it can handle <u>any desired function f !</u>**

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Fully Homomorphic Encryption

Craig Gentry built the first FHE sheme, on his Ph.D. Thesis in 2009

Message space: {0,1}
Permitted functions:
additions and multiplications

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Semantically secure encryption

- Encrypt the same message 2+ times must generate different ciphertexts
- Encryption must be random

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Semantically secure encryption

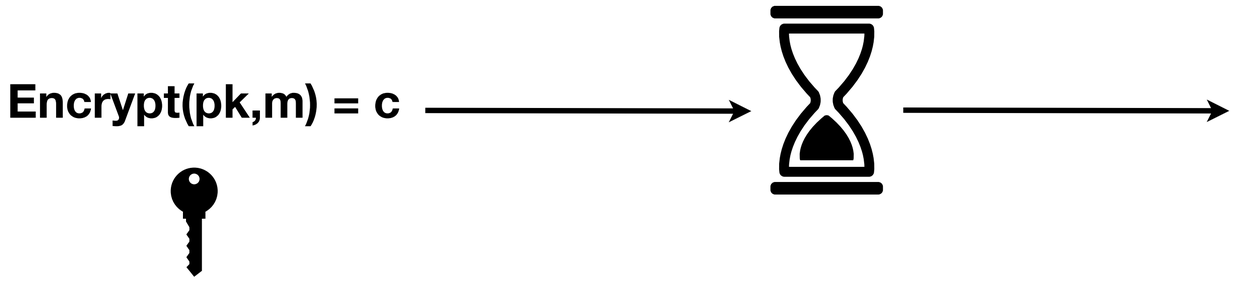
- Encrypt the same message 2+ times must generate different ciphertexts
- Encryption must be random

Encrypt(pk,m) = c

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Semantically secure encryption

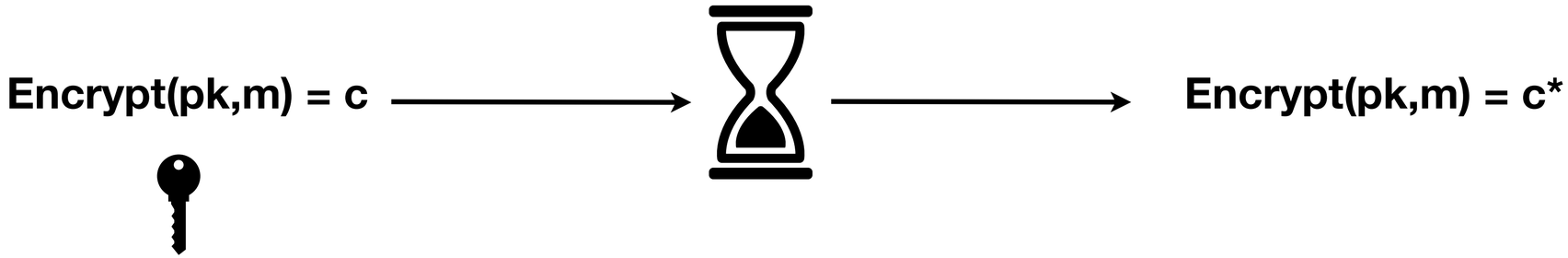
- Encrypt the same message 2+ times must generate different cyphertexts
- Encryption must be random



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Semantically secure encryption

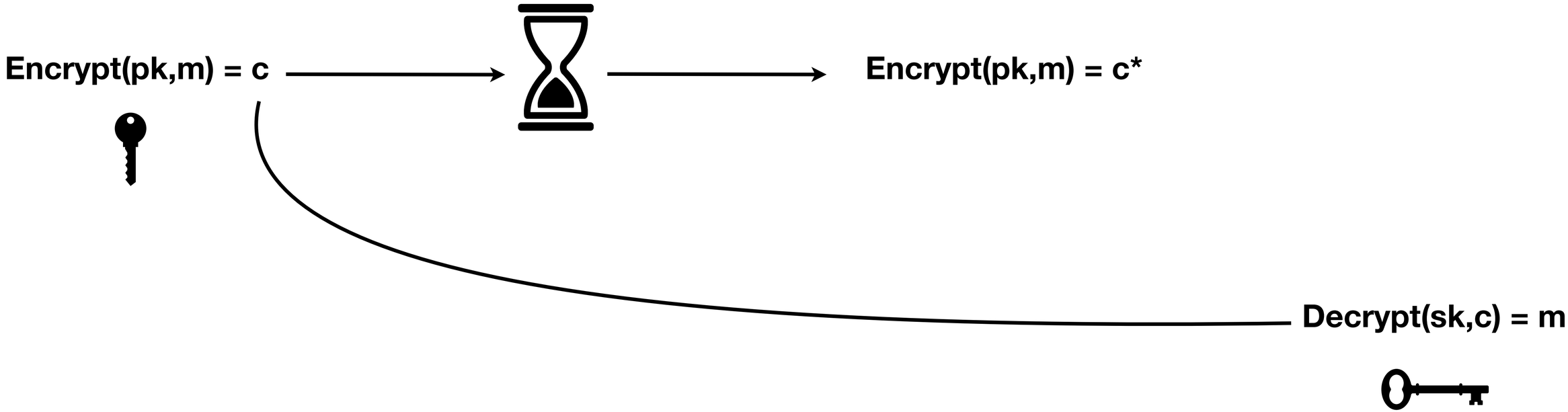
- Encrypt the same message 2+ times must generate different cyphertexts
- Encryption must be random



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Semantically secure encryption

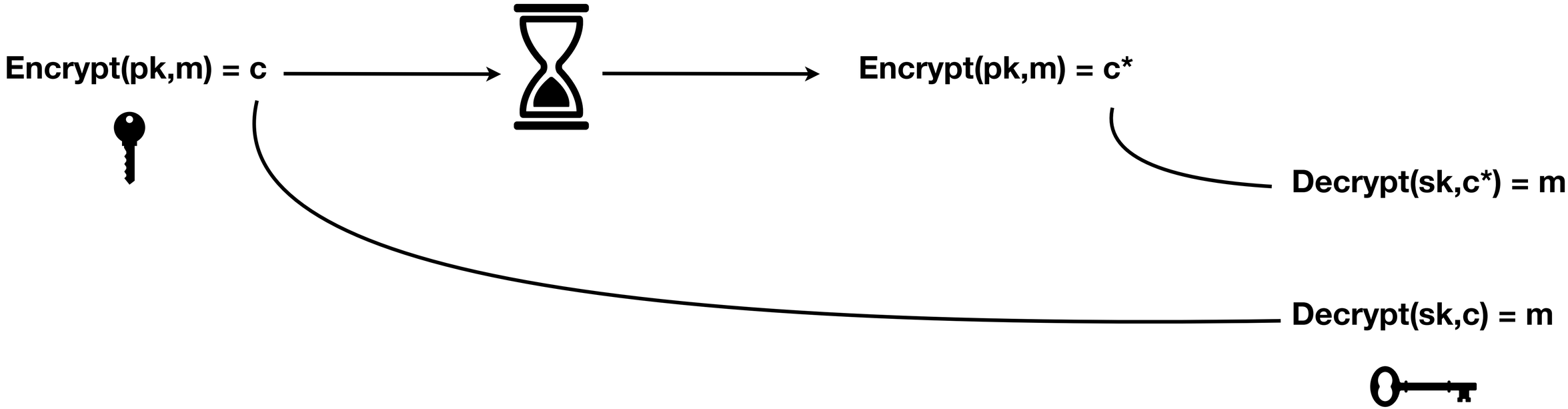
- Encrypt the same message 2+ times must generate different cyphertexts
- Encryption must be random



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Semantically secure encryption

- Encrypt the same message 2+ times must generate different cyphertexts
- Encryption must be random



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Somewhat Homomorphic Encryption (SHE)

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Somewhat Homomorphic Encryption (SHE)

A SHE scheme can homomorphically evaluate functions f with a limited depth

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Somewhat Homomorphic Encryption (SHE)

A SHE scheme can homomorphically evaluate functions f with a limited depth

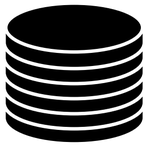
A simple SHE:

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Somewhat Homomorphic Encryption (SHE)

A SHE scheme can homomorphically evaluate functions f with a limited depth

A simple SHE:



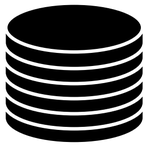
Message space:
{0,1}

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Somewhat Homomorphic Encryption (SHE)

A SHE scheme can homomorphically evaluate functions f with a limited depth

A simple SHE:



Message space:
{0,1}

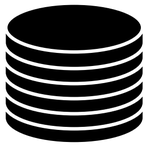
Key:
p, odd integer

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Somewhat Homomorphic Encryption (SHE)

A SHE scheme can homomorphically evaluate functions f with a limited depth

A simple SHE:



Message space:
{0,1}

Key:
p, odd integer

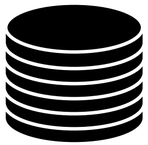
Ciphertext space:
integers

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Somewhat Homomorphic Encryption (SHE)

A SHE scheme can homomorphically evaluate functions f with a limited depth

A simple SHE:



Message space:
{0,1}

Encryption function:
c = pq + 2r + m

Key:
p, odd integer

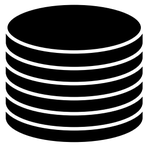
Ciphertext space:
integers

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Somewhat Homomorphic Encryption (SHE)

A SHE scheme can homomorphically evaluate functions f with a limited depth

A simple SHE:



Message space:
{0,1}

Encryption function:
c = pq + 2r + m
(q,r random)

Key:
p, odd integer

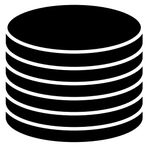
Ciphertext space:
integers

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Somewhat Homomorphic Encryption (SHE)

A SHE scheme can homomorphically evaluate functions f with a limited depth

A simple SHE:

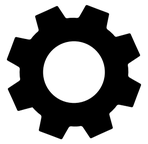


Message space:
{0,1}

Key:
p, odd integer

Ciphertext space:
integers

Encryption function:
c = pq + 2r + m
(q,r random)



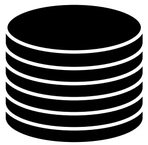
Decryption function:
(c mod p) mod 2

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Somewhat Homomorphic Encryption (SHE)

A SHE scheme can homomorphically evaluate functions f with a limited depth

A simple SHE:

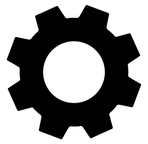


Message space:
{0,1}

Key:
p, odd integer

Ciphertext space:
integers

Encryption function:
$c = pq + 2r + m$
(q,r random)



Decryption function:
$(c \bmod p) \bmod 2$

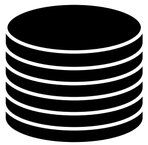
If $2r < p$, the decryption is correct:

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Somewhat Homomorphic Encryption (SHE)

A SHE scheme can homomorphically evaluate functions f with a limited depth

A simple SHE:

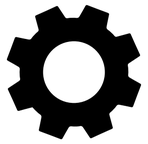


Message space:
{0,1}

Key:
p, odd integer

Ciphertext space:
integers

Encryption function:
c = pq + 2r + m
(q,r random)



Decryption function:
(c mod p) mod 2

If 2r < p, the decryption is correct:

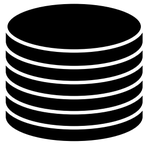
$$
\begin{aligned}
&(c \bmod p) \bmod 2 \\
&= ((pq + 2r + m) \bmod p) \bmod 2 \\
&= ((2r + m) \bmod p) \bmod 2 \\
&= (2r + m) \bmod 2 \\
&= m
\end{aligned}
$$

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Somewhat Homomorphic Encryption (SHE)

A SHE scheme can homomorphically evaluate functions f with a limited depth

A simple SHE:

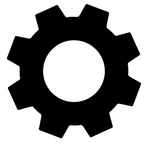


Message space:
{0,1}

Key:
p, odd integer

Ciphertext space:
integers

Encryption function:
c = pq + 2r + m
(q,r random)



Decryption function:
(c mod p) mod 2

If 2r < p, the decryption is correct:

$$
\begin{aligned}
&(c \bmod p) \bmod 2 \\
&= ((pq + 2r + m) \bmod p) \bmod 2 \\
&= ((2r + m) \bmod p) \bmod 2 \\
&= (2r + m) \bmod 2 \\
&= m
\end{aligned}
$$

If we add/multiply two ciphertexts c1 and c2, the noise will be added/multiplied too

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Bootstrapping

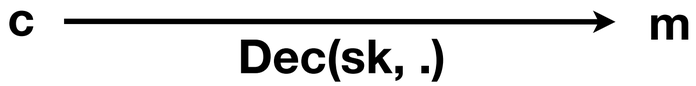
Let's suppose we have a ciphertext c
of a message m, and want to reduce its noise

c

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Bootstrapping

Let's suppose we have a ciphertext c
of a message m, and want to reduce its noise

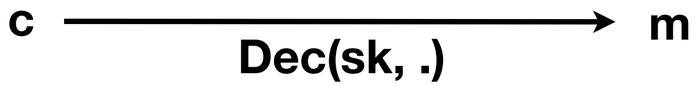


FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Bootstrapping

Let's suppose we have a ciphertext c
of a message m, and want to reduce its noise

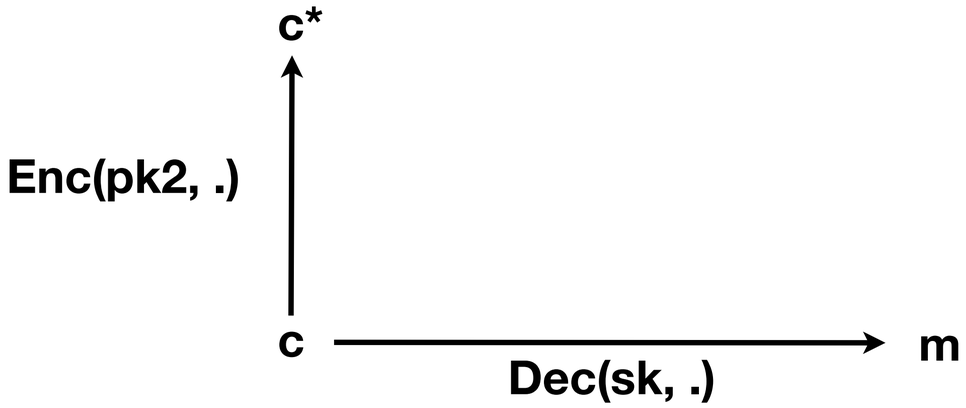
The Decryption function completely removes any noise !!
But reveals the message and uses the private key :(



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Bootstrapping

Let's suppose we have a ciphertext c
of a message m, and want to reduce its noise



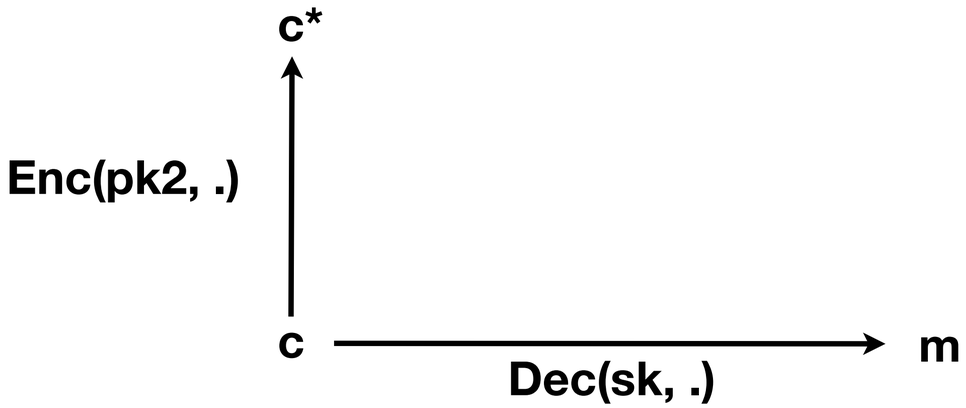
The Decryption function completely removes any noise !!
But reveals the message and uses the private key :(

c* is a double encryption of m

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Bootstrapping

Let's suppose we have a ciphertext c
of a message m, and want to reduce its noise



The Decryption function completely removes any noise !!
But reveals the message and uses the private key :(

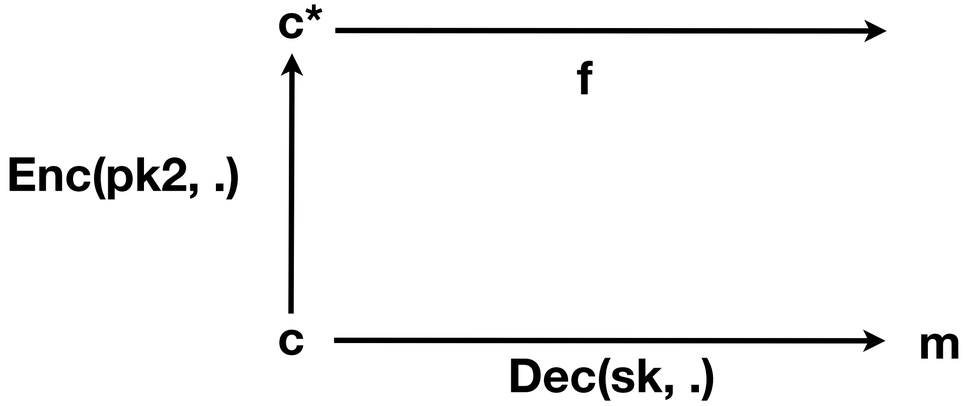
c* is a double encryption of m

We can homomorphically apply permitted functions f to c*

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Bootstrapping

Let's suppose we have a ciphertext c
of a message m, and want to reduce its noise



The Decryption function completely removes any noise !!
But reveals the message and uses the private key :(

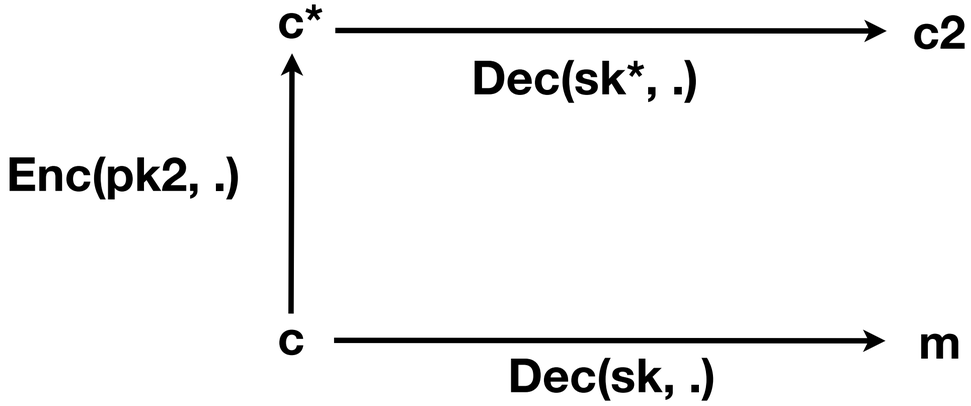
c* is a double encryption of m

We can homomorphically apply permitted functions f to c*

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Bootstrapping

Let's suppose we have a ciphertext c
of a message m, and want to reduce its noise



pk2 = second public key
sk* = Enc(pk2,sk)

The Decryption function completely removes any noise !!
But reveals the message and uses the private key :(

c* is a double encryption of m

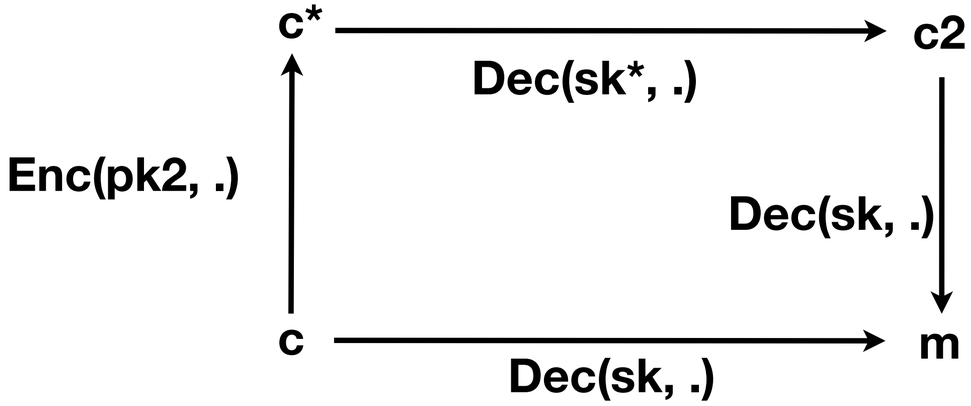
We can homomorphically apply permitted functions f to c*

So, apply f(.) = Dec(sk*, .)

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Bootstrapping

Let's suppose we have a ciphertext c
of a message m, and want to reduce its noise



pk2 = second public key
sk* = Enc(pk2,sk)

The Decryption function completely removes any noise !!
But reveals the message and uses the private key :(

c* is a double encryption of m

We can homomorphically apply permitted functions f to c*

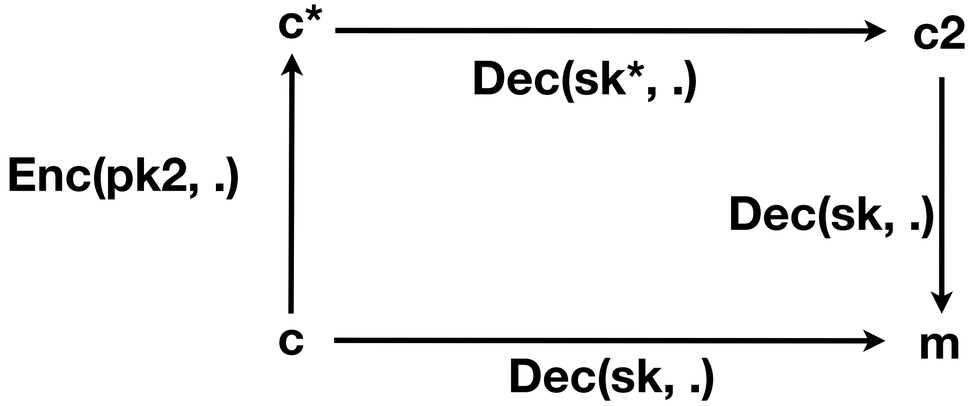
So, apply f(.) = Dec(sk*, .)

c2 is a new ciphertext that encrypts m, with less noise!

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Bootstrapping

Let's suppose we have a ciphertext c
of a message m, and want to reduce its noise



pk2 = second public key
sk* = Enc(pk2,sk)

The Decryption function completely removes any noise !!
But reveals the message and uses the private key :(

c* is a double encryption of m

We can homomorphically apply permitted functions f to c*

So, apply f(.) = Dec(sk*, .)

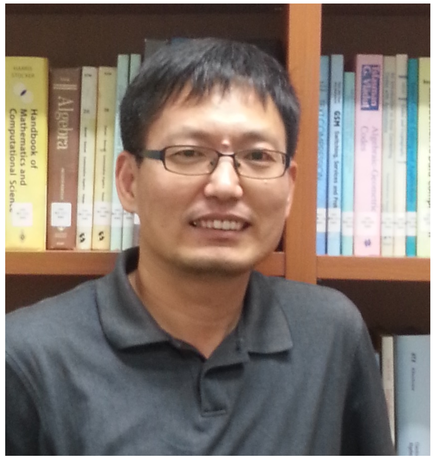
c2 is a new ciphertext that encrypts m, with less noise!

A SHE scheme is called bootstrappable,
if the Decryption function is a permitted function

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

CKKS scheme

Cheon, Kim, Kim and Song (CKKS, 2017) - Homomorphic Encryption for arithmetic of approximate numbers



Jun Hee Cheon
Seoul National University

Andrey Kim
Samsung Advanced Institute of Technology

Miran Kim
Hanyang University

Yongsoo Song
Seoul National University

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

CKKS scheme

Plaintext Space: Polynomials

$$\mathcal{R}_q = \mathbb{Z}_q[X]/(X^N + 1)$$

Ciphertext Space: Polynomials tuples

$$\mathcal{R}_q \times \mathcal{R}_q.$$

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

CKKS scheme

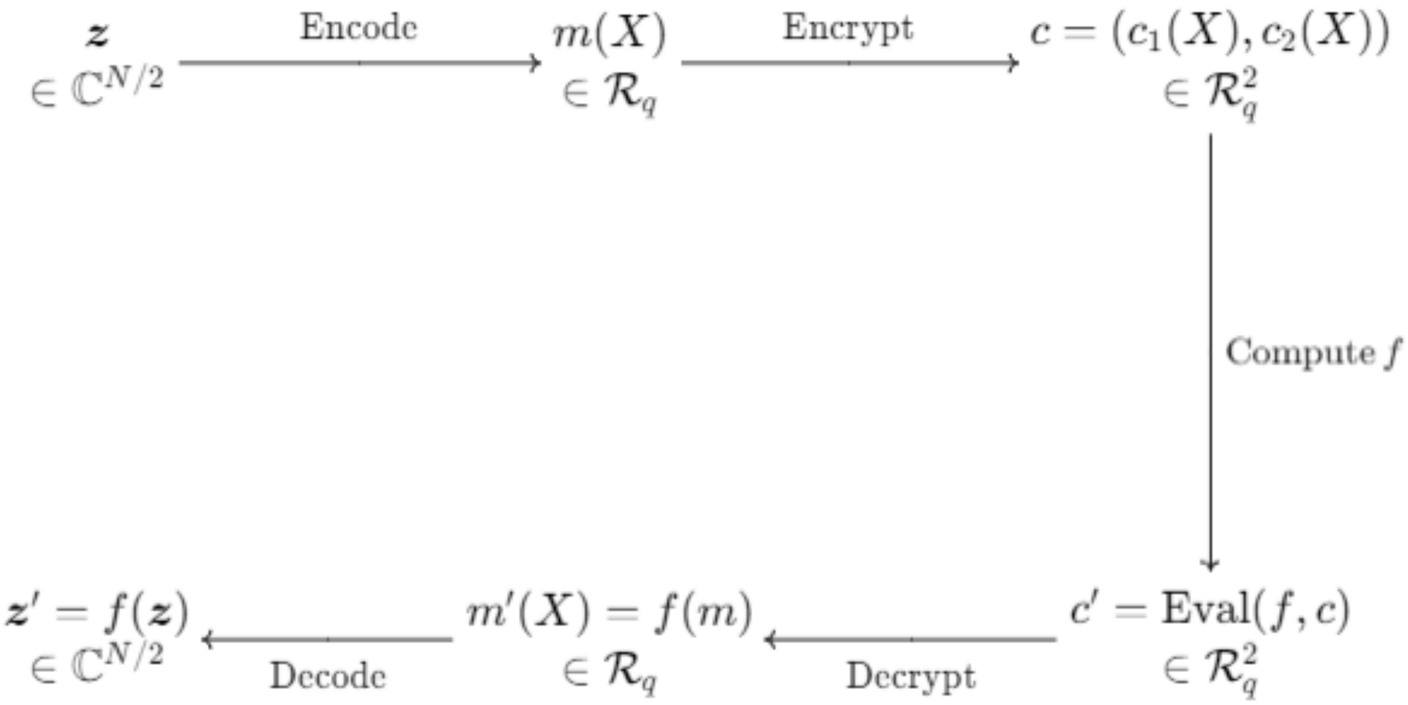
Plaintext Space: Polynomials

$$\mathcal{R}_q = \mathbb{Z}_q[X]/(X^N + 1)$$

Ciphertext Space: Polynomials tuples

$$\mathcal{R}_q \times \mathcal{R}_q.$$

Complex Numbers are supported through encoding.



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

CKKS scheme

Plaintext Space: Polynomials

$$\mathcal{R}_q = \mathbb{Z}_q[X]/(X^N + 1)$$

Ciphertext Space: Polynomials tuples

$$\mathcal{R}_q \times \mathcal{R}_q.$$

Secret key: (1,s)
s - random polynomial from Rq

Public Key:

$$\mathrm{pk} = (b, a), \text{ with } b = [-a \cdot s + e]_{q_L}$$

$$a \xleftarrow{\mathcal{U}} \mathcal{R}_{q_L} \text{ and } e \xleftarrow{\mathcal{DG}(\sigma^2)} \mathcal{R}.$$

Encryption:

$$\mathrm{Enc}(\mathrm{pk}, m) = [v \cdot \mathrm{pk} + (m + e_0, e_1)]_{q_L}$$

Decryption:

$$\mathrm{Dec}(\mathrm{sk}, \boldsymbol{c}) = [\langle \boldsymbol{c}, \mathrm{sk} \rangle]_{q_\ell} = [c_1 + c_2 \cdot s]_{q_\ell}, \text{ for } \boldsymbol{c} = (c_1, c_2)$$

Allowed operations:
Additions, Multiplications, Rotations

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Security - Learning with Errors (LWE)

$$\boldsymbol{a}_i \xleftarrow{u} \mathbb{Z}_q^n$$

$$b_i = \langle \boldsymbol{a}_i, \boldsymbol{s} \rangle + e_i \pmod{q}$$

$$e_i \xleftarrow{\chi} \mathbb{Z}_q$$

LWE Problem: Guess s, given (ai,bi)

If ei = 0, we could easily find s, solving As = b

With ei > 0, this is a NP-hard problem

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Ring Learning with Errors (RLWE)

$$a_i \xleftarrow{\mathcal{U}} \mathcal{R}_q$$

$$b_i = a_i \cdot s + e_i \pmod{q}$$

$$e_i \xleftarrow{\chi} \mathcal{R}$$

RLWE Problem: Guess s, given (ai,bi)

CKKS public key security is based on RLWE:

$$\mathrm{pk} = (b, a), \text{ with } b = [-a \cdot s + e]_{q_L}$$

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Logistic Regression

$$\boldsymbol{X} \in \mathbb{R}^{n \times d}$$

$$\boldsymbol{y} \in \{0, 1\}^n$$

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Logistic Regression

$$\boldsymbol{X} \in \mathbb{R}^{n \times d}$$

$$\boldsymbol{y} \in \{0, 1\}^n$$

$$\ln\left(\frac{p(\boldsymbol{x})}{1 - p(\boldsymbol{x})}\right) = \boldsymbol{w}^T \boldsymbol{x}'$$

$$\boldsymbol{w} = (w_0, w_1, \ldots, w_d)^T \in \mathbb{R}^{d+1}$$

$$\boldsymbol{x}' = (1; \boldsymbol{x})$$

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Logistic Regression

$$\boldsymbol{X} \in \mathbb{R}^{n \times d}$$

$$\boldsymbol{y} \in \{0, 1\}^n$$

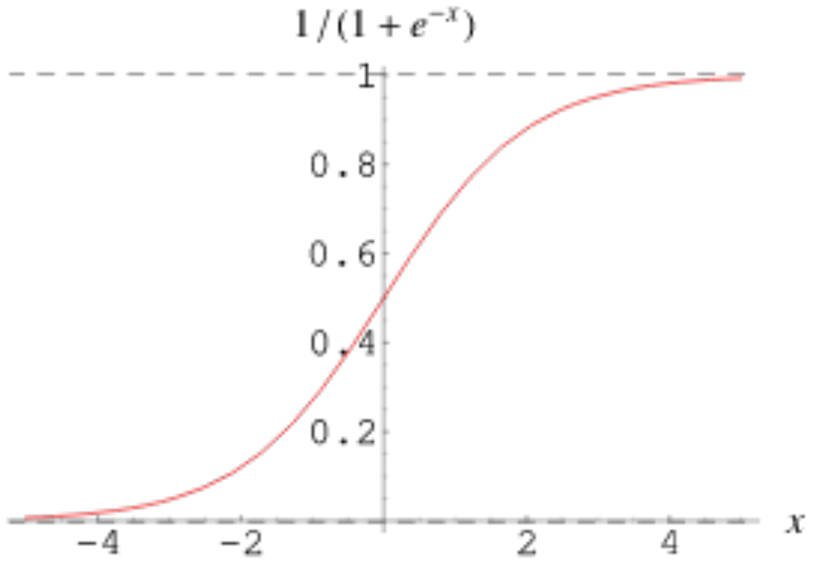
$$\ln\left(\frac{p(\boldsymbol{x})}{1 - p(\boldsymbol{x})}\right) = \boldsymbol{w}^T \boldsymbol{x}'$$

$$\boldsymbol{w} = (w_0, w_1, \ldots, w_d)^T \in \mathbb{R}^{d+1}$$

$$\boldsymbol{x}' = (1; \boldsymbol{x})$$

$$p(\boldsymbol{x}) = \sigma(\boldsymbol{w}^T \boldsymbol{x}')$$

$$\sigma(x) = 1/(1 + e^{-x})$$



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Logistic Regression - Training

Training algorithm: Find w that maximizes the likelihood ...

$$\mathcal{L}(\boldsymbol{w}) = \prod_{i=1}^{n} \sigma(\boldsymbol{w}^T \boldsymbol{x}_i')^{y_i} (1 - \sigma(\boldsymbol{w}^T \boldsymbol{x}_i'))^{1-y_i}$$

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Logistic Regression - Training

Training algorithm: Find w that maximizes the likelihood ...

$$\mathcal{L}(\boldsymbol{w}) = \prod_{i=1}^{n} \sigma(\boldsymbol{w}^T \boldsymbol{x}_i')^{y_i} (1 - \sigma(\boldsymbol{w}^T \boldsymbol{x}_i'))^{1-y_i}$$

... or minimizes the negative log-likelihood

$$\ell(\boldsymbol{w}) = -\frac{1}{n} \ln(\mathcal{L}(\boldsymbol{w}))$$

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Logistic Regression - Training

Training algorithm: Find w that maximizes the likelihood ...

$$\mathcal{L}(\boldsymbol{w}) = \prod_{i=1}^{n} \sigma(\boldsymbol{w}^T \boldsymbol{x}_i')^{y_i} (1 - \sigma(\boldsymbol{w}^T \boldsymbol{x}_i'))^{1-y_i}$$

... or minimizes the negative log-likelihood

$$\ell(\boldsymbol{w}) = -\frac{1}{n} \ln(\mathcal{L}(\boldsymbol{w}))$$

Gradient Descent Method:

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i - \alpha \nabla \ell(\boldsymbol{w}_i),$$

$$\text{where } \nabla \ell(\boldsymbol{w}_i) = -\frac{1}{n} \sum_{j=1}^{n} \sigma(-\boldsymbol{w}_i^T \boldsymbol{z}_j) \boldsymbol{z}_j$$

Two inner products involved!
exponential function in sigmoid!

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Logistic Regression - Private Training

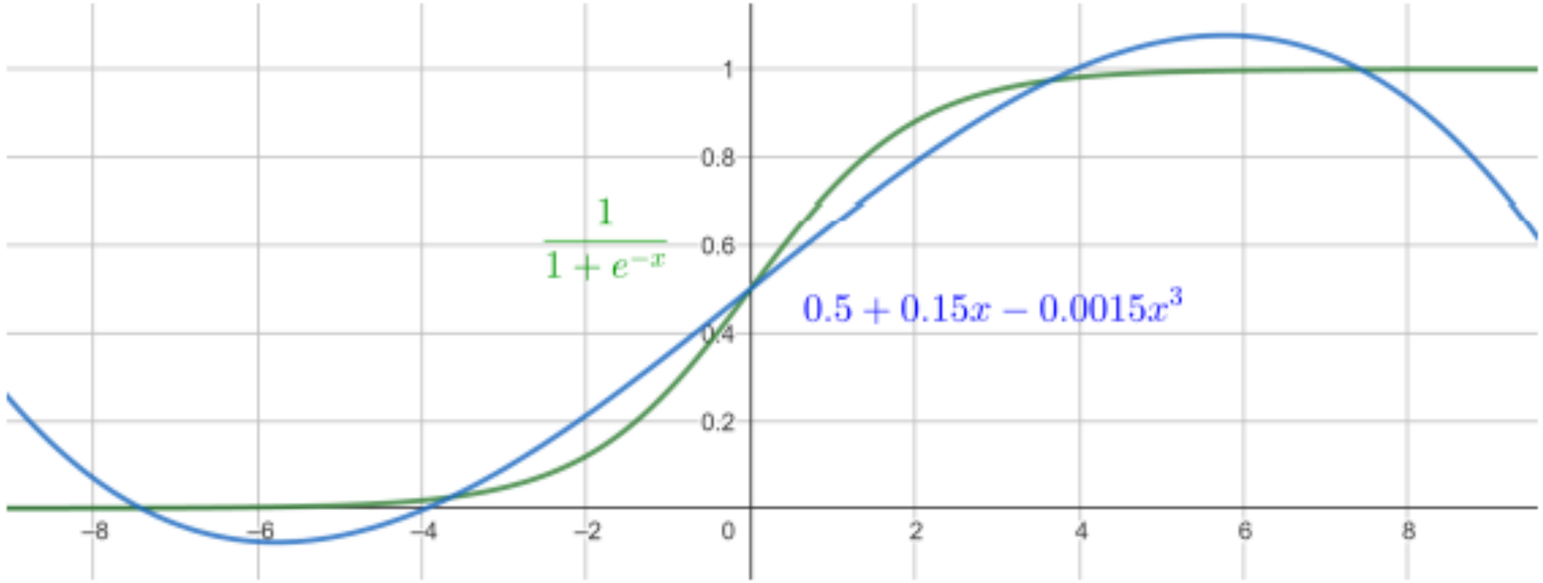
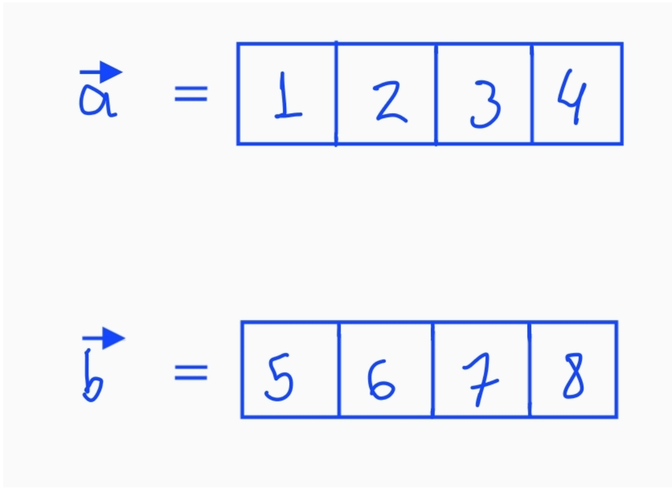


Figure 6 – Sigmoid polynomial approximation

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Logistic Regression - Private Training

Calculating Inner products
Naive approach: encrypt each entry as a ciphertext



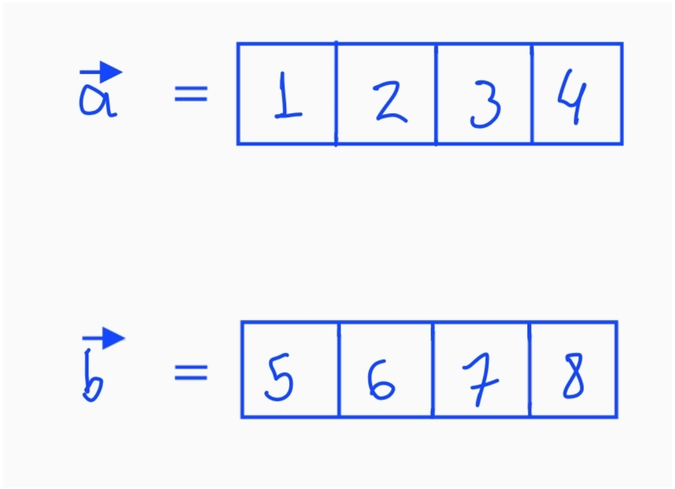
a1.b1 + a2.b2 + a3.c3 + a4.c4 = 70

4 multiplications, 3 additions required

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Logistic Regression - Private Training

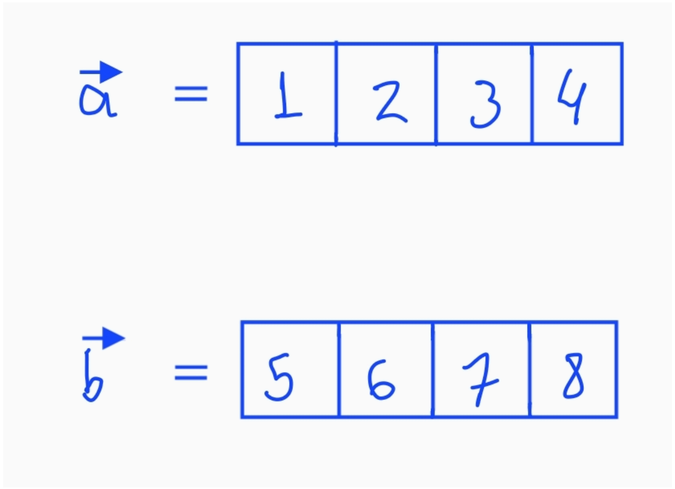
Calculating Inner products
SIMD approach: encrypt each vector as a ciphertext



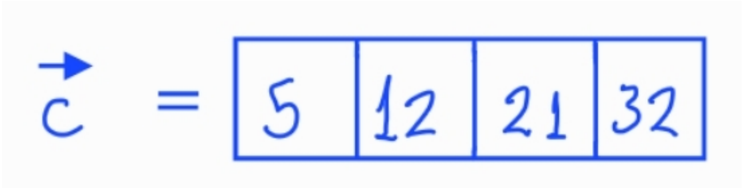
FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Logistic Regression - Private Training

1st Step: Do 1 SIMD multiplication



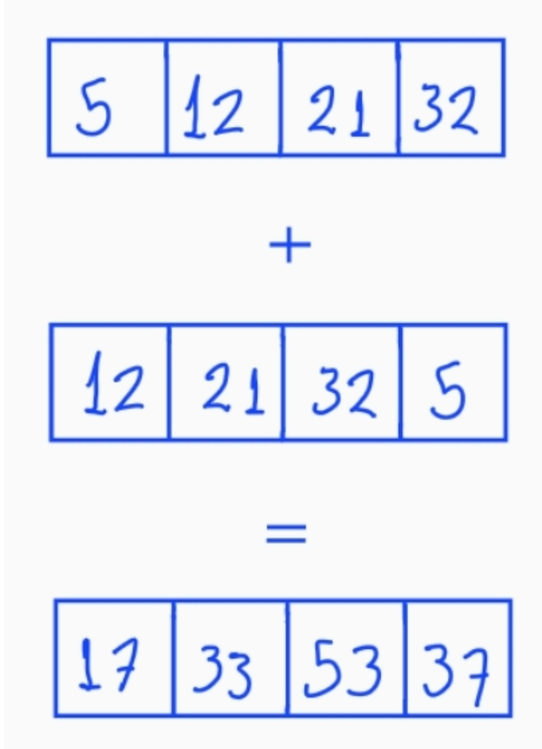
c = a.b



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Logistic Regression - Private Training

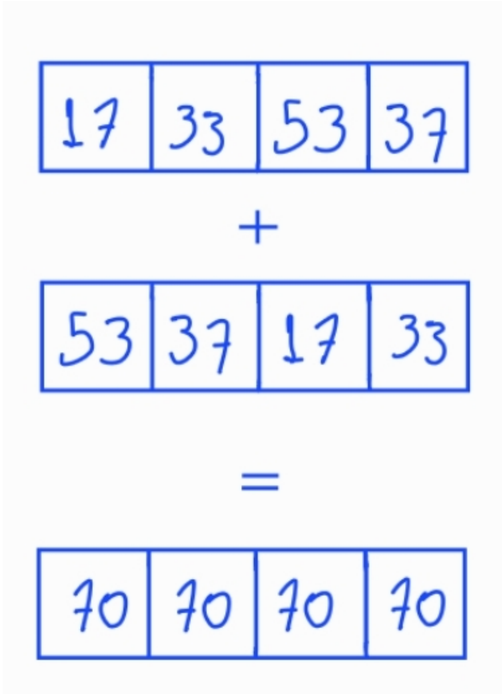
2nd Step: c = c + left_rotate(c,1)



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Logistic Regression - Private Training

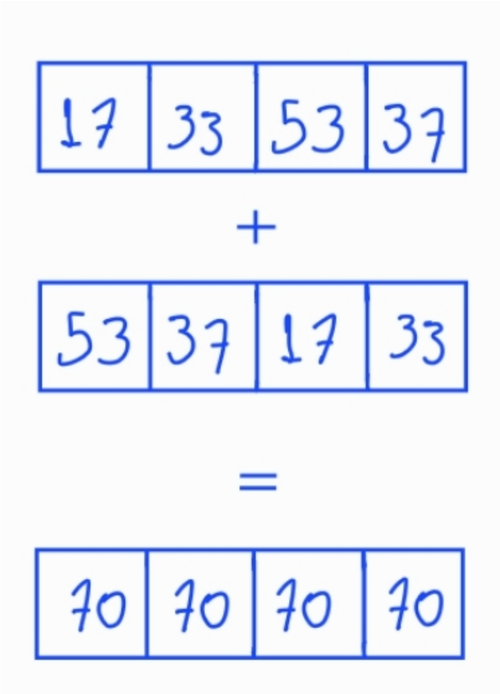
3nd Step: c = c + left_rotate(c,2)



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Logistic Regression - Private Training

3nd Step: c = c + left_rotate(c,2)



Total cost: 1 multiplications + 2 additions

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Logistic Regression - Private Training

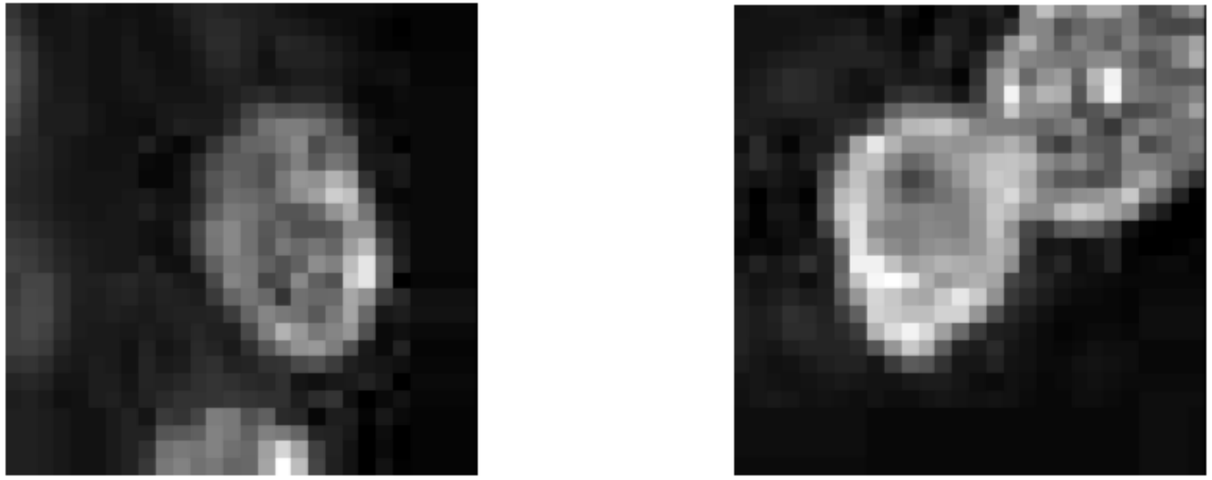
In general a d-dimensional inner product requires:

- Naive approach: d multiplications, d-1 additions
- SIMD approach: 1 multiplication, log(d) additions

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Simulation Results

- HEAAN + HELR C++ packages;
- Dataset: Subset of the TissueMNIST dataset; 92672 rows and 196 columns;
- Machine: 64-bit quad-core Intel Core i5-6200U 2.3GHz CPU, 16GB of RAM;

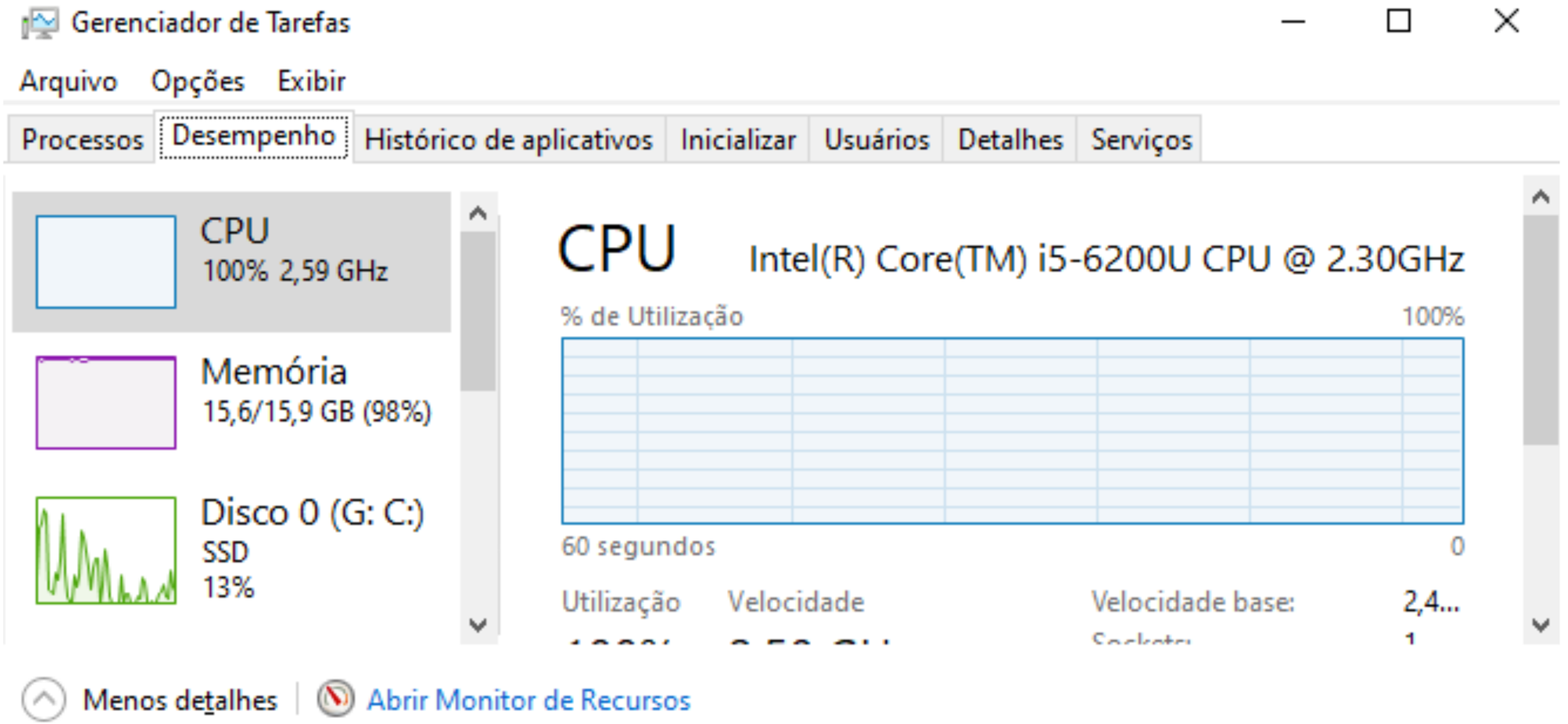


(a) Collecting Duct (b) Proximal Tubule Segments

Figure 7 – TissueMNIST

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Simulation Results



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Simulation Results

KeyGen time	8.52 min
Encrypt time	19.17 min
Training time	6.19 hours
Public key size	2.62 GB

	Encrypted	Unencrypted
Accuracy	64.6211%	64.6363%
AUC	81.7039%	81.6996%

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Text organization

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

References

Rivest, 1978 - ON DATA BANKS AND PRIVACY HOMOMORPHISMS;

Gentry, 2009 - Fully Homomorphic Encryption using Ideal Lattices;

Gentry, 2010 - Fully Homomorphic Encryption over the Integers;

CHEON, Jung Hee; KIM, Andrey, et al., 2017 - Homomorphic Encryption
for Arithmetic of Approximate Numbers;

HAN, Kyoohyung et al., 2018 Efficient Logistic Regression on Large Encrypted Data.

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

FHE libraries - Overview

Library/ Scheme or Extension	BGV	BGV Bootstr.	BFV	CKKS	CKKS Bootstr.	DM	CGGI	Threshold FHE (MP)	PRE (MP)
Concrete							✓		
FHEW						✓			
HEAAN				✓	✓				
HELib	✓	✓		✓					
Lattigo			✓	✓	✓			✓	
OpenFHE	✓	*	✓	✓	✓	✓	✓	✓	✓
PALISADE	✓		✓	✓		✓	✓	✓	✓
SEAL	✓		✓	✓					
TFHE							✓		

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

FHE libraries - Overview

Library/ Scheme or Extension	BGV	BGV Bootstr.	BFV	CKKS	CKKS Bootstr.	DM	CGGI	Threshold FHE (MP)	PRE (MP)
Concrete							✓		
FHEW						✓			
HEAAN				✓	✓				
HELib	✓	✓		✓					
Lattigo			✓	✓	✓			✓	
OpenFHE	✓	*	✓	✓	✓	✓	✓	✓	✓
PALISADE	✓		✓	✓		✓	✓	✓	✓
SEAL	✓		✓	✓					
TFHE							✓		

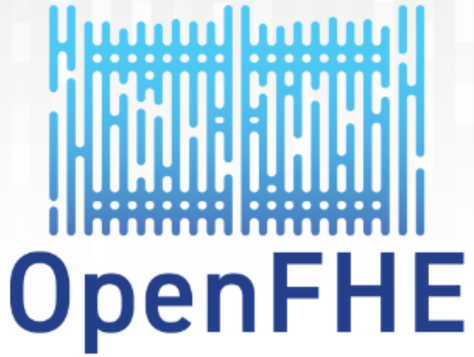
FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

FHE libraries - OpenFHE

Library/ Scheme or Extension	BGV	BGV Bootstr.	BFV	CKKS	CKKS Bootstr.	DM	CGGI	Threshold FHE (MP)	PRE (MP)
Concrete							✓		
FHEW						✓			
HEAAN				✓	✓				
HELib	✓	✓		✓					
Lattigo			✓	✓	✓			✓	
OpenFHE	✓	*	✓	✓	✓	✓	✓	✓	✓
PALISADE	✓		✓	✓		✓	✓	✓	✓
SEAL	✓		✓	✓					
TFHE							✓		

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

FHE libraries - OpenFHE



Community Growth:

OpenFHE is an open-source project that provides efficient extensible implementations of the leading post-quantum Fully Homomorphic Encryption (FHE) schemes.

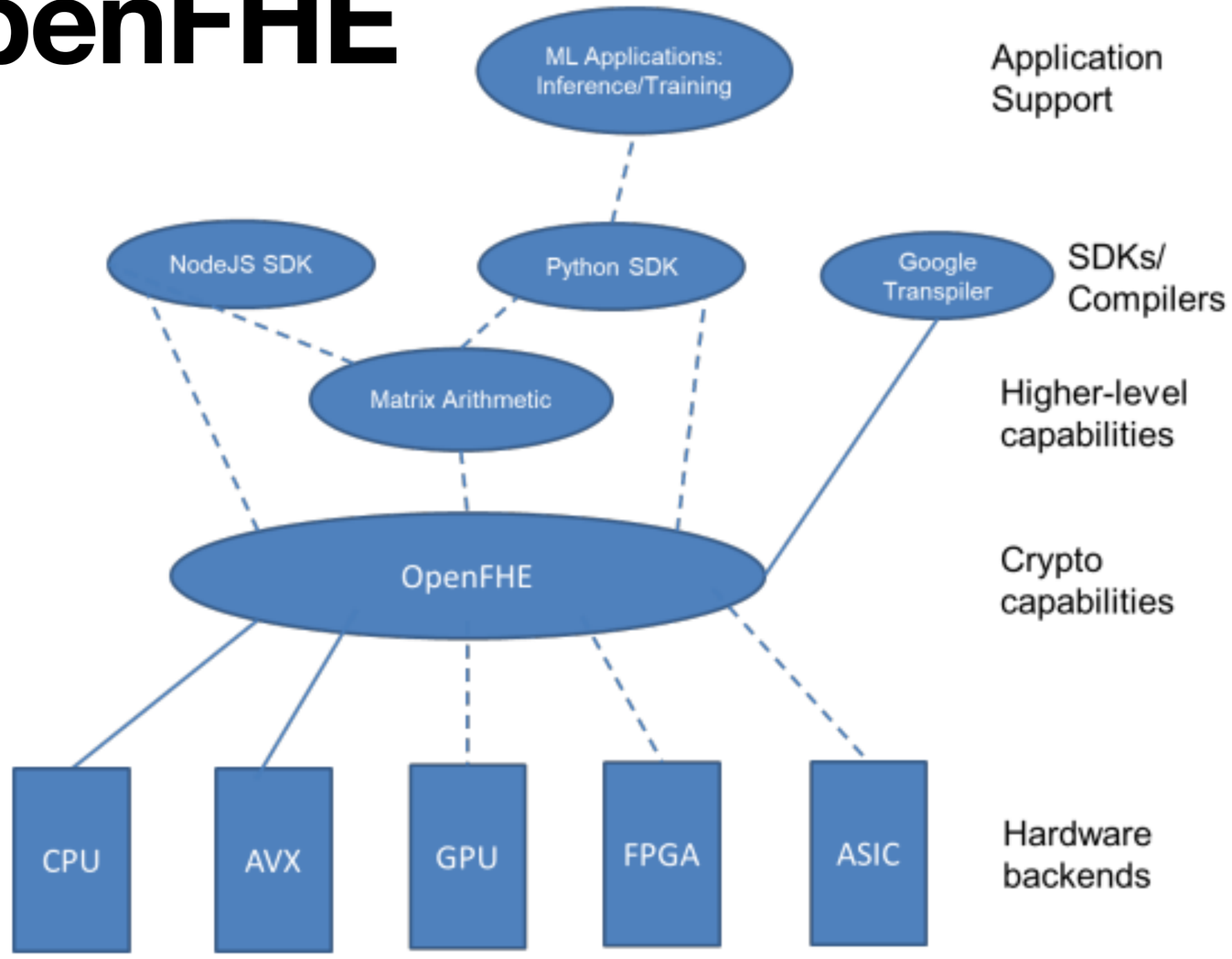
Paper 2022/915

OpenFHE: Open-Source Fully Homomorphic Encryption Library

Ahmad Al Badawi, Duality Technologies
Jack Bates, Duality Technologies
Flavio Bergamaschi, Intel Corporation
David Bruce Cousins, Duality Technologies
Saroja Erabelli, Duality Technologies
Nicholas Genise, Duality Technologies
Shai Halevi, Algorand Foundation
Hamish Hunt, Intel Corporation
Andrey Kim, Samsung Advanced Institute of Technology
Yongwoo Lee, Samsung Advanced Institute of Technology
Zeyu Liu, Duality Technologies
Daniele Micciancio, University of California, San Diego, Duality Technologies
Ian Quah, Duality Technologies
Yuriy Polyakov, Duality Technologies
Saraswathy R.V., Duality Technologies
Kurt Rohloff, Duality Technologies
Jonathan Saylor, Duality Technologies
Dmitriy Suponitsky, Duality Technologies
Matthew Triplett, Duality Technologies
Vinod Vaikuntanathan, Massachusets Institute of Technology, Duality Technologies
Vincent Zucca, DALI, Universite de Perpignan Via Domitia, LIRMM, University of Montpellier

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

FHE libraries - OpenFHE



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

FHE libraries - OpenFHE Python

Goals:

- **Make FHE more accessible for non-C++ users and ML community**
- **Keep C++ efficiency**

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

FHE libraries - OpenFHE Python

Goals:

- Make FHE more accessible for non-C++ users and ML community
- Keep C++ efficiency

Development Team:

- Yuriy Polyakov
- Ian Quah
- Matthew Tripplet
- Rener Oliveira

Paper 2022/915

OpenFHE: Open-Source Fully Homomorphic Encryption Library

Ahmad Al Badawi, Duality Technologies
Jack Bates, Duality Technologies
Flavio Bergamaschi, Intel Corporation
David Bruce Cousins, Duality Technologies
Saroja Erabelli, Duality Technologies
Nicholas Genise, Duality Technologies
Shai Halevi, Algorand Foundation
Hamish Hunt, Intel Corporation
Andrey Kim, Samsung Advanced Institute of Technology
Yongwoo Lee, Samsung Advanced Institute of Technology
Zeyu Liu, Duality Technologies
Daniele Micciancio, University of California, San Diego, Duality Technologies
Ian Quah, Duality Technologies
Yuriy Polyakov, Duality Technologies
Saraswathy R.V., Duality Technologies
Kurt Rohloff, Duality Technologies
Jonathan Saylor, Duality Technologies
Dmitriy Suponitsky, Duality Technologies
Matthew Triplett, Duality Technologies
Vinod Vaikuntanathan, Massachusets Institute of Technology, Duality Technologies
Vincent Zucca, DALI, Universite de Perpignan Via Domitia, LIRMM, University of Montpellier

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

FHE libraries - OpenFHE Python

Goals:
- Make FHE more accessible for non-C++ users and ML community
- Keep C++ efficiency

Development Team:
- Yuriy Polyakov
- Ian Quah
- Matthew Tripplet
- Rener Oliveira

Project History
- Feb/2023 - July/2023: First pre-release v0.8.0
- Ago/2023 - Now: working on stable release

Sponsorship:
- NumFocus (also finances Numpy, scikit-learn, pandas, etc)

Paper 2022/915

OpenFHE: Open-Source Fully Homomorphic Encryption Library

Ahmad Al Badawi, Duality Technologies
Jack Bates, Duality Technologies
Flavio Bergamaschi, Intel Corporation
David Bruce Cousins, Duality Technologies
Saroja Erabelli, Duality Technologies
Nicholas Genise, Duality Technologies
Shai Halevi, Algorand Foundation
Hamish Hunt, Intel Corporation
Andrey Kim, Samsung Advanced Institute of Technology
Yongwoo Lee, Samsung Advanced Institute of Technology
Zeyu Liu, Duality Technologies
Daniele Micciancio, University of California, San Diego, Duality Technologies
Ian Quah, Duality Technologies
Yuriy Polyakov, Duality Technologies
Saraswathy R.V., Duality Technologies
Kurt Rohloff, Duality Technologies
Jonathan Saylor, Duality Technologies
Dmitriy Suponitsky, Duality Technologies
Matthew Triplett, Duality Technologies
Vinod Vaikuntanathan, Massachusets Institute of Technology, Duality Technologies
Vincent Zucca, DALI, Universite de Perpignan Via Domitia, LIRMM, University of Montpellier

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

FHE libraries - OpenFHE Python

openfhe-python

Search docs

API REFERENCE:

- CryptoContext
- Ciphertext
- Plaintext
- Public Key
- Private Key
- KeyPair
- PKE enums
- BinFHEContext
- LWECiphertext
- LWEPrivateKey
- BinFHE enums

Welcome to OpenFHE - Python's documentation! View page source

Welcome to OpenFHE - Python's documentation!

OpenFHE - Python

Fully Homomorphic Encryption (FHE) is a powerful cryptographic primitive that enables performing computations over encrypted data without having access to the secret key. OpenFHE is an open-source FHE library that includes efficient implementations of all common FHE schemes: BFV, BGV, CKKS, DM and CGGI.

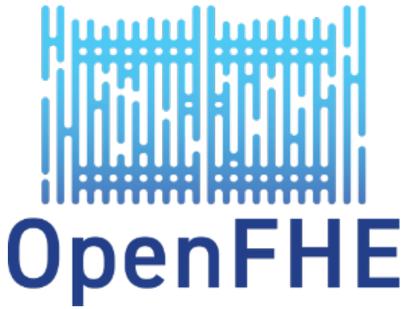
`openfhe-python` is a Python library built as a wrapper for the main capabilities of OpenFHE C++ library. It provides a more user-friendly interface for Python developers, while keeping the efficiency of C++ FHE operations.

API Reference:

- CryptoContext
- Ciphertext

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

FHE libraries - OpenFHE Python



New contributors are welcome :)
Visit our websites:

openfhe.org
github.com/openfheorg
github.com/reneroliveira