FGV EMAp
ESCOLA DE
MATEMÁTICA
APLICADA

Machine learning over encrypted data

Rener Oliveira
Rodrigo Targino

June 27th, 2022

Table of Contents

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Introduction

Supervised machine learning setup:

Data | Model training | Final Predictor Function

X - Predictors matrix
y - output vector

Linear Models
Random Forests
Boosting
Neural Networks

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 :)

Privacy Homomorphisms, 1978

ON DATA BANKS AND PRIVACY HOMOMORPHISMS

Ronald L. Rivest
Len Adleman
Michael L. Dertouzos

Massachusetts Institute of Technology
Cambridge, Massachusetts

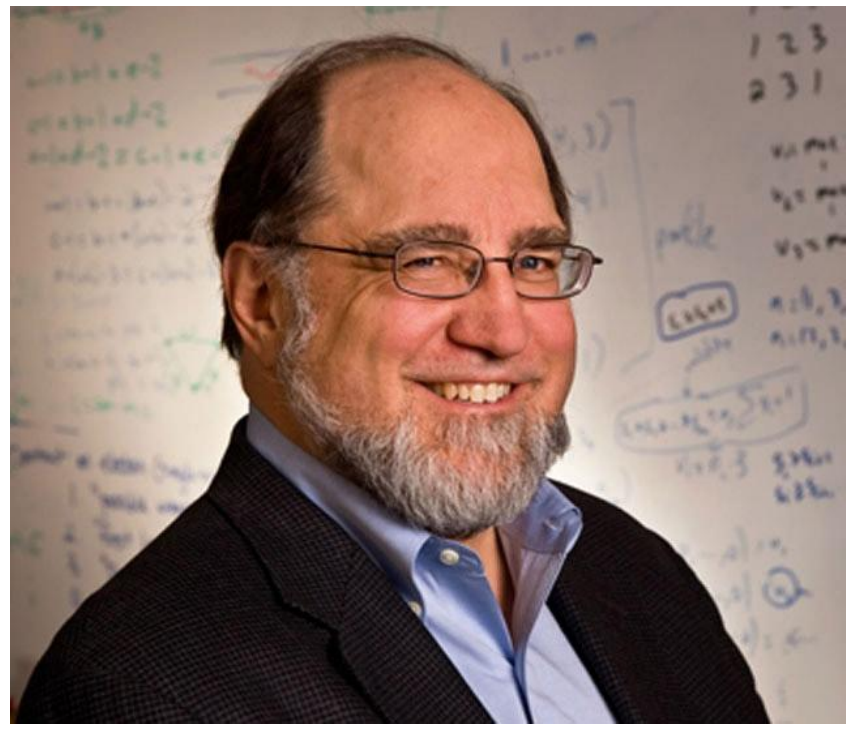
FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Privacy Homomorphisms, 1978

ON DATA BANKS AND PRIVACY HOMOMORPHISMS

Ronald L. Rivest
Len Adleman
Michael L. Dertouzos

Massachusetts Institute of Technology
Cambridge, Massachusetts

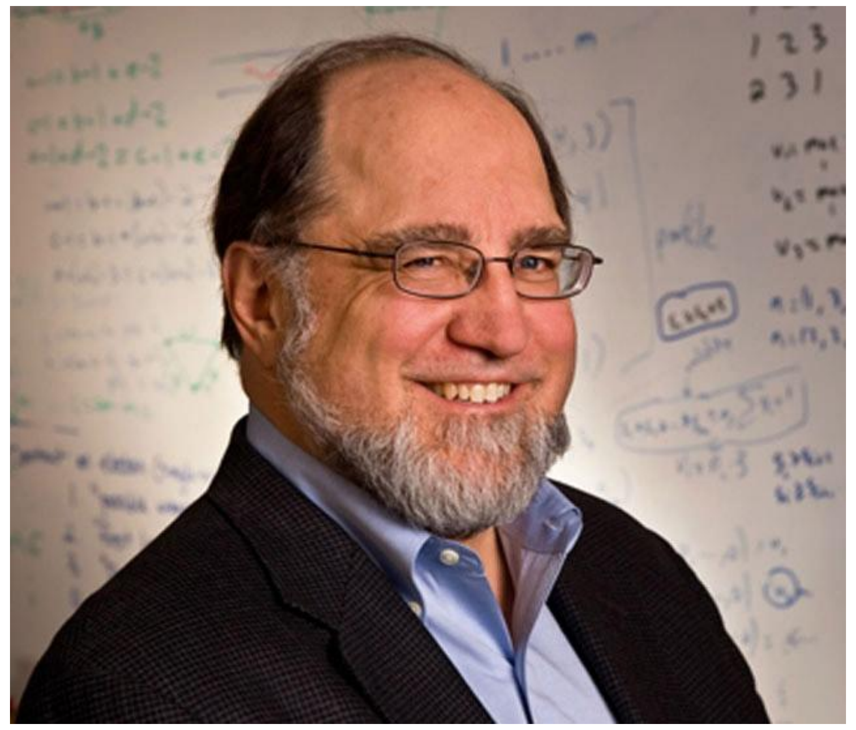


Privacy Homomorphisms, 1978

ON DATA BANKS AND PRIVACY HOMOMORPHISMS

Ronald L. Rivest
Len Adleman
Michael L. Dertouzos

Massachusetts Institute of Technology
Cambridge, Massachusetts



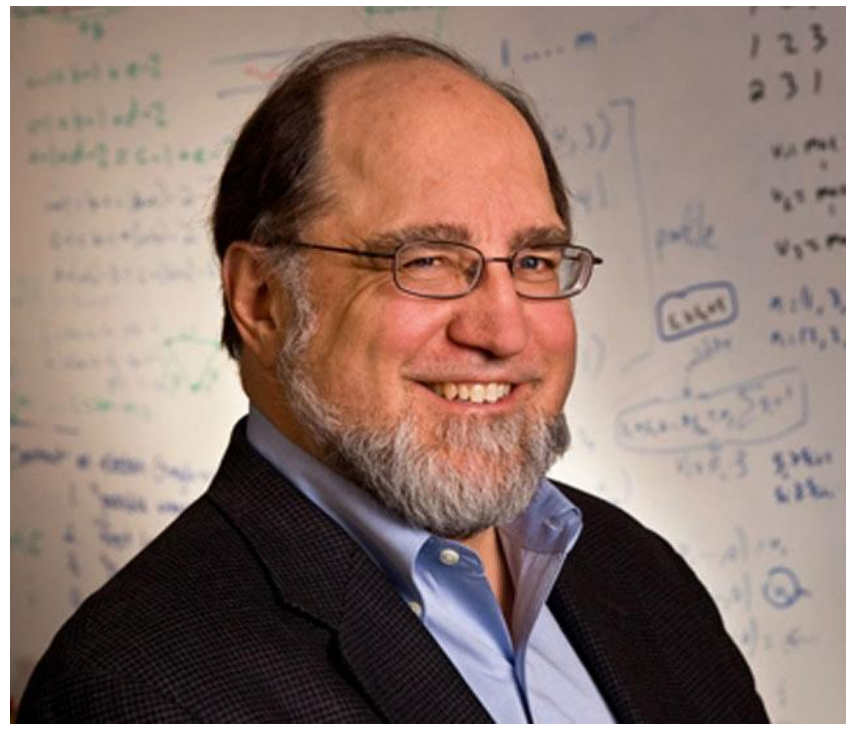
FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Privacy Homomorphisms, 1978

ON DATA BANKS AND PRIVACY HOMOMORPHISMS

Ronald L. Rivest
Len Adleman
Michael L. Dertouzos

Massachusetts Institute of Technology
Cambridge, Massachusetts



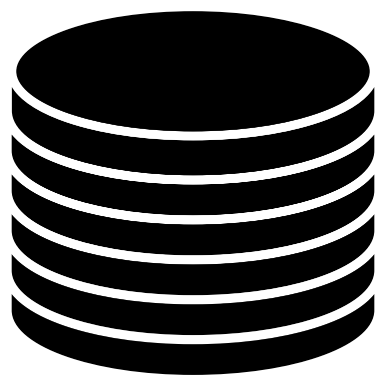
limitations on what can be accomplished, we shall see that it appears likely that there exist encryption functions which permit encrypted data to be operated on without preliminary decryption of the operands, for many sets of interesting operations. These special encryption functions we call "privacy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

Fully Homomorphic Encryption

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Fully Homomorphic Encryption

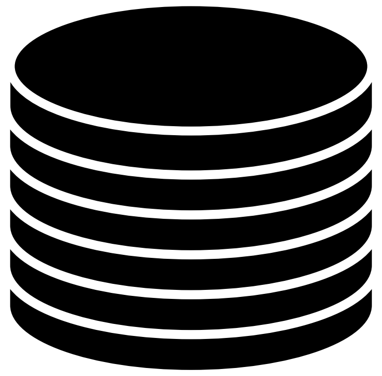
Message space:
(m1, m2, ...)



Fully Homomorphic Encryption

How to securely compute f(m1,m2,...) ?

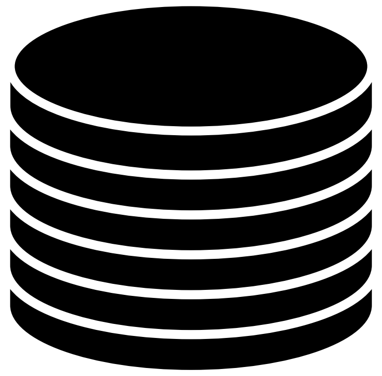
Message space:
(m1, m2, ...)



Fully Homomorphic Encryption

How to securely compute f(m1,m2,...) ?

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

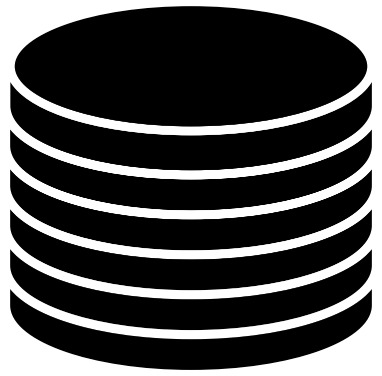


pk = public key

Fully Homomorphic Encryption

How to securely compute f(m1,m2,...) ?

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



c1 = Encrypt(pk,m1)

pk = public key

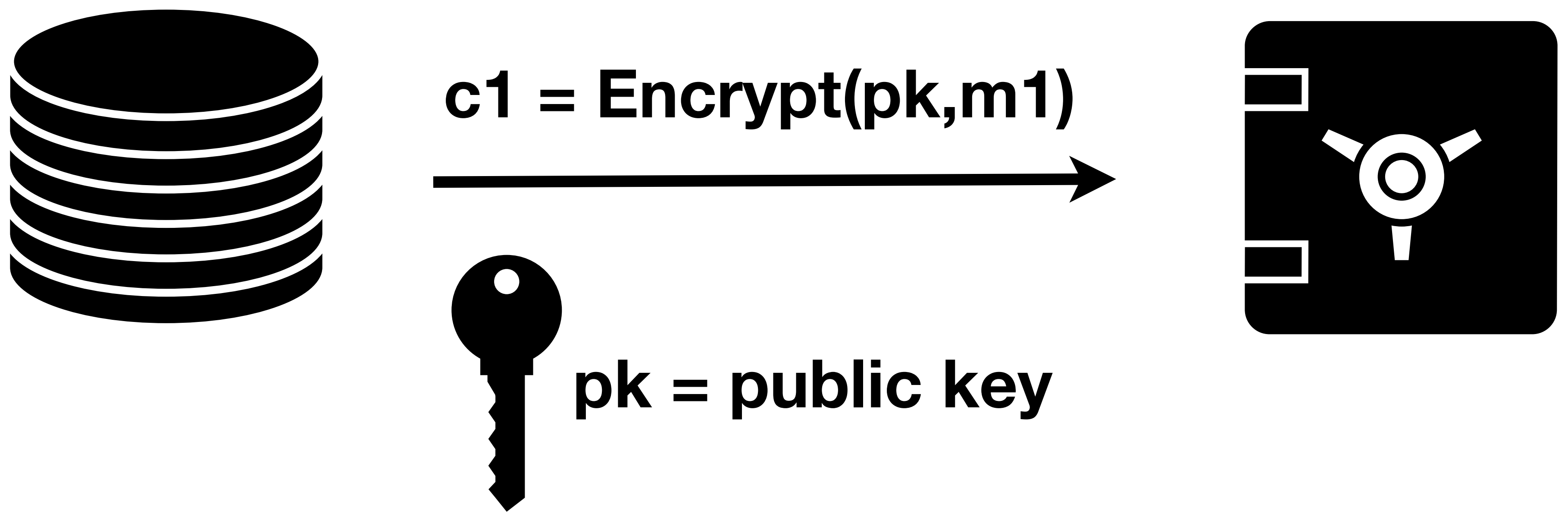
FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Fully Homomorphic Encryption

How to securely compute f(m1,m2,...) ?

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

Cyphertext space:
(c1, c2, ...)



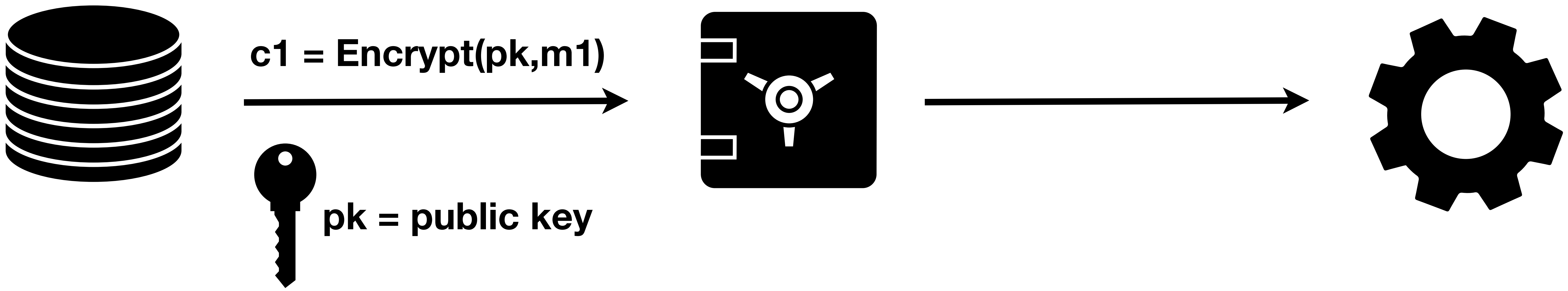
Fully Homomorphic Encryption

How to securely compute f(m1,m2,...) ?

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

Cyphertext space: (c1, c2, ...)

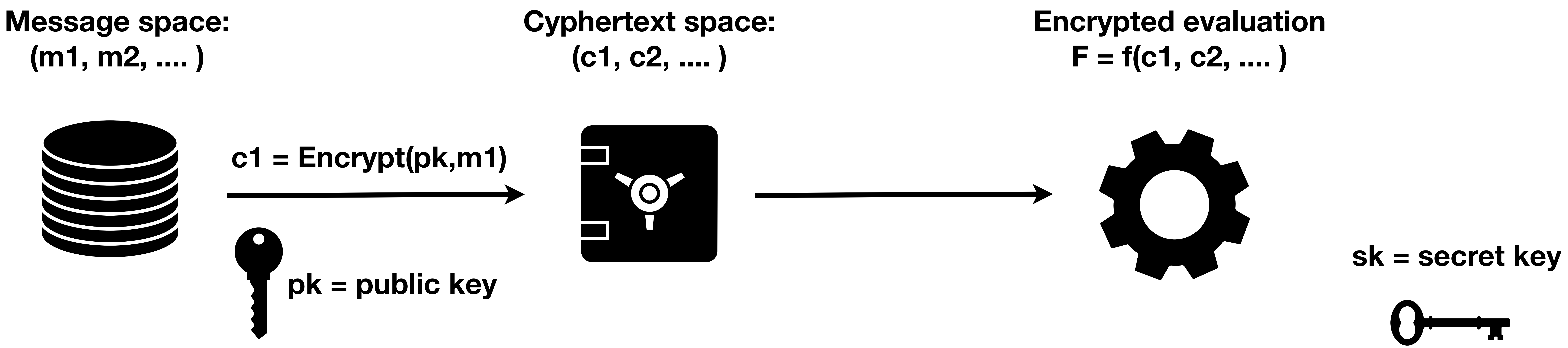
Encrypted evaluation F = f(c1, c2, ...)



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Fully Homomorphic Encryption

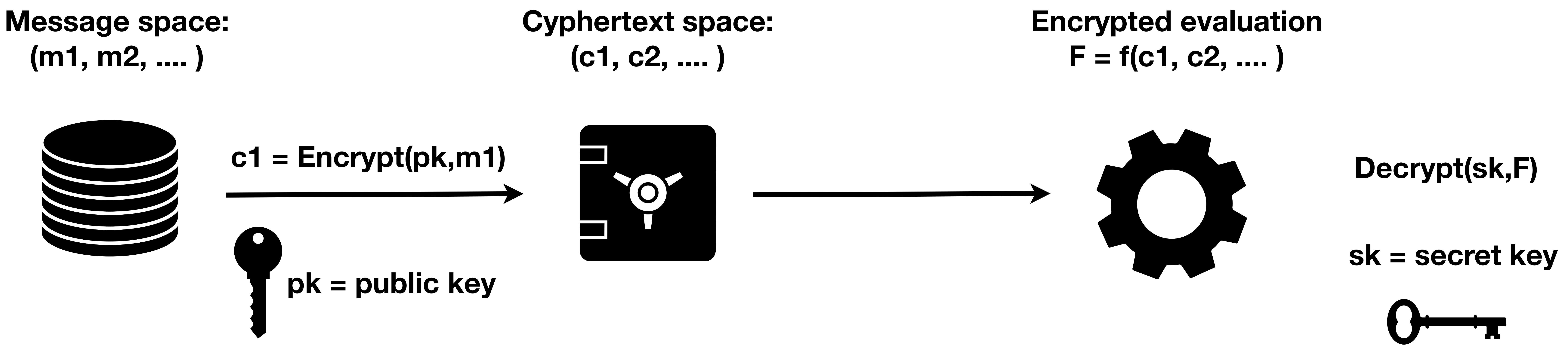
How to securely compute f(m1,m2,...) ?



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Fully Homomorphic Encryption

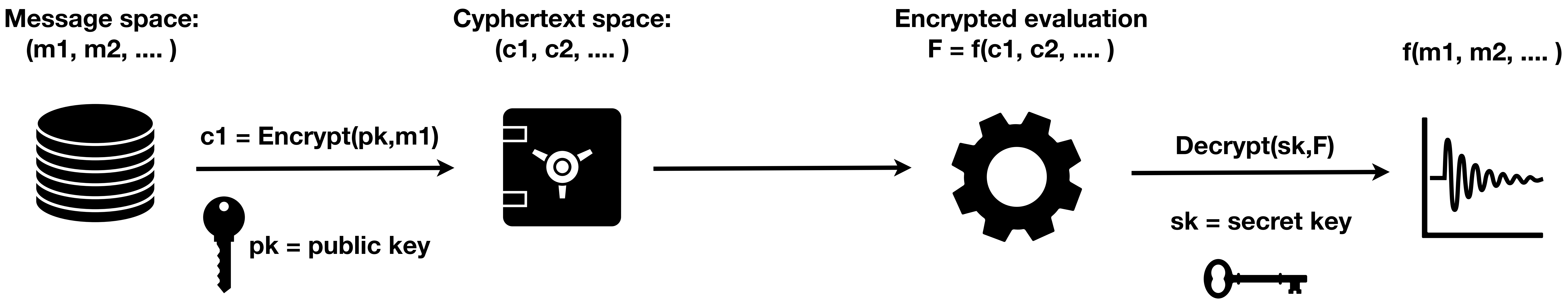
How to securely compute f(m1,m2,...) ?



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Fully Homomorphic Encryption

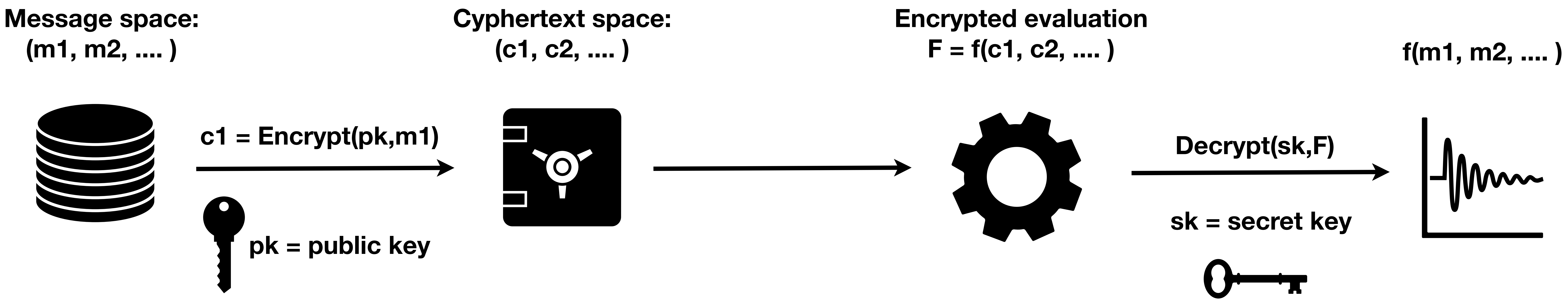
How to <u>securely</u> compute f(m1,m2,...) ?



FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Fully Homomorphic Encryption

How to securely compute f(m1,m2,...) ?

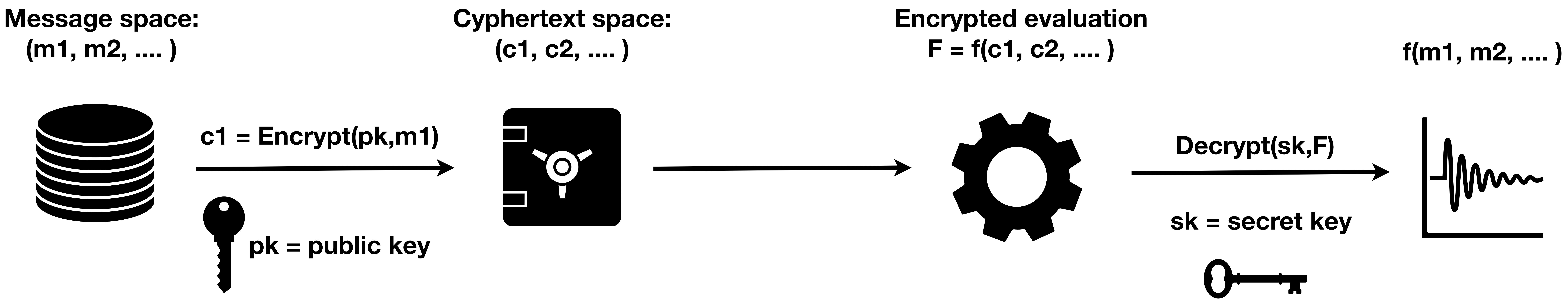


An encryption scheme is called Fully Homomorphic, if it can handle any desired function f !

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, on his Ph.D. Thesis in 2009

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

Semantically secure encryption

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

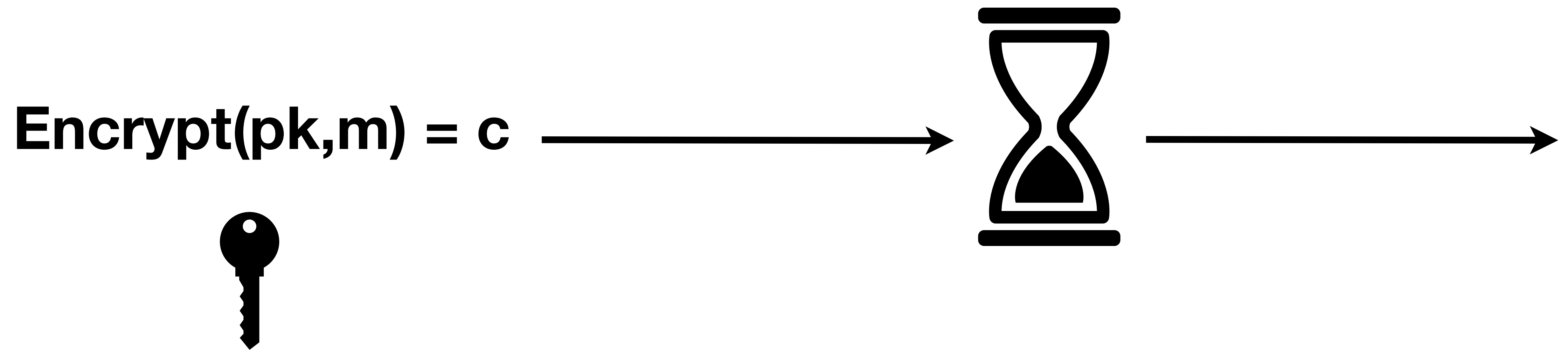
Semantically secure encryption

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

Encrypt(pk,m) = c

Semantically secure encryption

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



ESCOLA DE MATEMÁTICA APLICADA

Semantically secure encryption

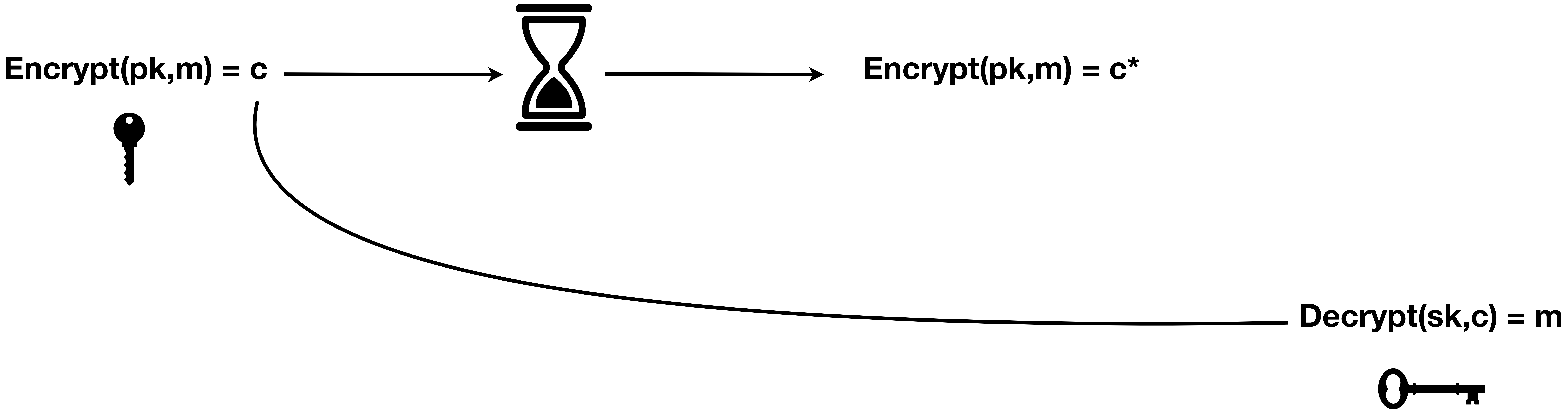
- Encrypt the same message 2+ times must generate different cyphertexts
- Encryption must be random

Encrypt(pk,m) = c $\longrightarrow$  $\longrightarrow$ Encrypt(pk,m) = c*

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

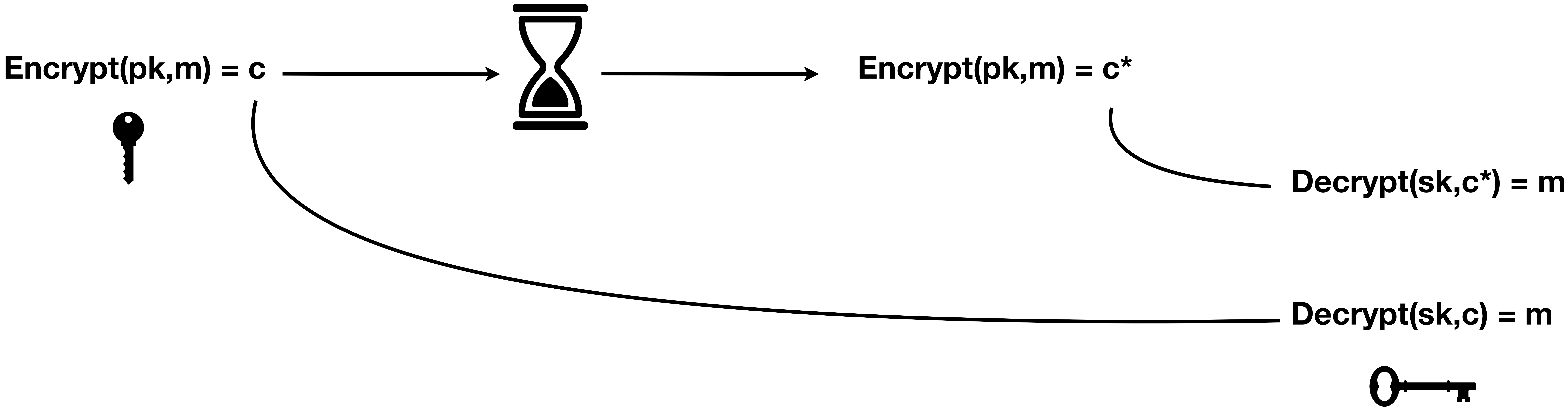
Semantically secure encryption

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



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)

Somewhat Homomorphic Encryption (SHE)

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

Somewhat Homomorphic Encryption (SHE)

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

A simple SHE:

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

Cyphertext 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

Cyphertext space:
integers

Encryption function:
$c = pq + 2r + m$

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

Cyphertext space:
integers

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

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

Cyphertext space:
integers

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

Decryption function:
(c mod p) mod 2

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

Cyphertext 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:

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

$(c \bmod p) \bmod 2$

Message space:
{0,1}

Key:
p, odd integer

Cyphertext space:
integers

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

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

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

Cyphertext 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:

$$(c \bmod p) \bmod 2$$
$$= ((pq + 2r + m) \bmod p) \bmod 2$$

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

Cyphertext 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:

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

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

Cyphertext 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:

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

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

Cyphertext 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:

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

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

Cyphertext 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:

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

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

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Bootstrapping

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

$$c \xrightarrow[\text{Dec}(sk,\ .)]{} m$$

Bootstrapping

Let's suppose we have a cyphertext 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 :(

$$c \xrightarrow[\text{Dec(sk, .)}]{} m$$

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Bootstrapping

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

$$
\begin{tikzcd}
c^* & \\
c \arrow[u, "\mathrm{Enc}(pk2,\ .)"] \arrow[r, "\mathrm{Dec}(sk,\ .)"'] & m
\end{tikzcd}
$$

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 cyphertext c
of a message m, and want to reduce its noise

$$\begin{tikzcd}
c^* & \\
c \arrow[u, "\mathrm{Enc}(pk2,\ .)"] \arrow[r, "\mathrm{Dec}(sk,\ .)"'] & m
\end{tikzcd}$$

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*

Bootstrapping

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

$$
\begin{tikzcd}
c^* \arrow[r, "f"'] & {} \\
c \arrow[u, "{Enc(pk2,\ .)}"] \arrow[r, "{Dec(sk,\ .)}"'] & m
\end{tikzcd}
$$

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 cyphertext c
of a message m, and want to reduce its noise

$$\begin{tikzcd}
c^* \arrow[r, "\mathrm{Dec}(sk^*,\, .)"'] & c2 \\
c \arrow[u, "\mathrm{Enc}(pk2,\, .)"] \arrow[r, "\mathrm{Dec}(sk,\, .)"'] & m
\end{tikzcd}$$

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*, .)

Bootstrapping

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

$$
\begin{tikzcd}
c^* \arrow[r, "\text{Dec}(sk^*, .)"'] & c2 \arrow[d, "\text{Dec}(sk, .)"'] \\
c \arrow[u, "\text{Enc}(pk2, .)"] \arrow[r, "\text{Dec}(sk, .)"'] & m
\end{tikzcd}
$$

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 cyphertext that encrypts m, with less noise!

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

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Bootstrapping

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

$$
\begin{tikzcd}
c^* \arrow[rr, "\mathrm{Dec}(sk^*,\ .)"'] & & c2 \arrow[d, "\mathrm{Dec}(sk,\ .)"'] \\
c \arrow[u, "\mathrm{Enc}(pk2,\ .)"] \arrow[rr, "\mathrm{Dec}(sk,\ .)"'] & & m
\end{tikzcd}
$$

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 cyphertext that encrypts m, with less noise!

A SHE scheme is called <u>bootstrappable</u>,
if the Decryption function is a permitted function

Literature overview

4 Generations of FHE

1st Gen.	C. Gentry, 2009	FHE Using Ideal Lattices	Solved 1978 Rivest Problem
1st Gen.	V. Dijk, C. Gentry, S. Halevi, V. Vaikuntanathan (DGHV, 2010)	FHE Over the Integers	Simplification of 2009 scheme
2nd Gen.	Z. Brakerski, Fan, Vercauteren (BFV, 2012)	Somewhat Practical FHE	More eficient SHE
3rd Gen.	Gentry, A. Sahai, and B. Waters (GSW, 2013)	HE from LWE: Conceptually-Simpler, Asymptotically-Faster	Faster bootstrapping, more efficient multiplications
4th Gen.	Cheon, Kim, Kim and Song (CKKS, 2016)	HE for arithmetic of approximate numbers	Built over complex/real numbers

Calendar

MONTH	TASK
Mar - Jun	Mapped existing papers and studied Gentry's work (1st gen.)
July	Study next generation main papers/Map existing library's/packages
Ago - Set	Implementation of private regression models, tests and benchmarks
Out - Nov	Write

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