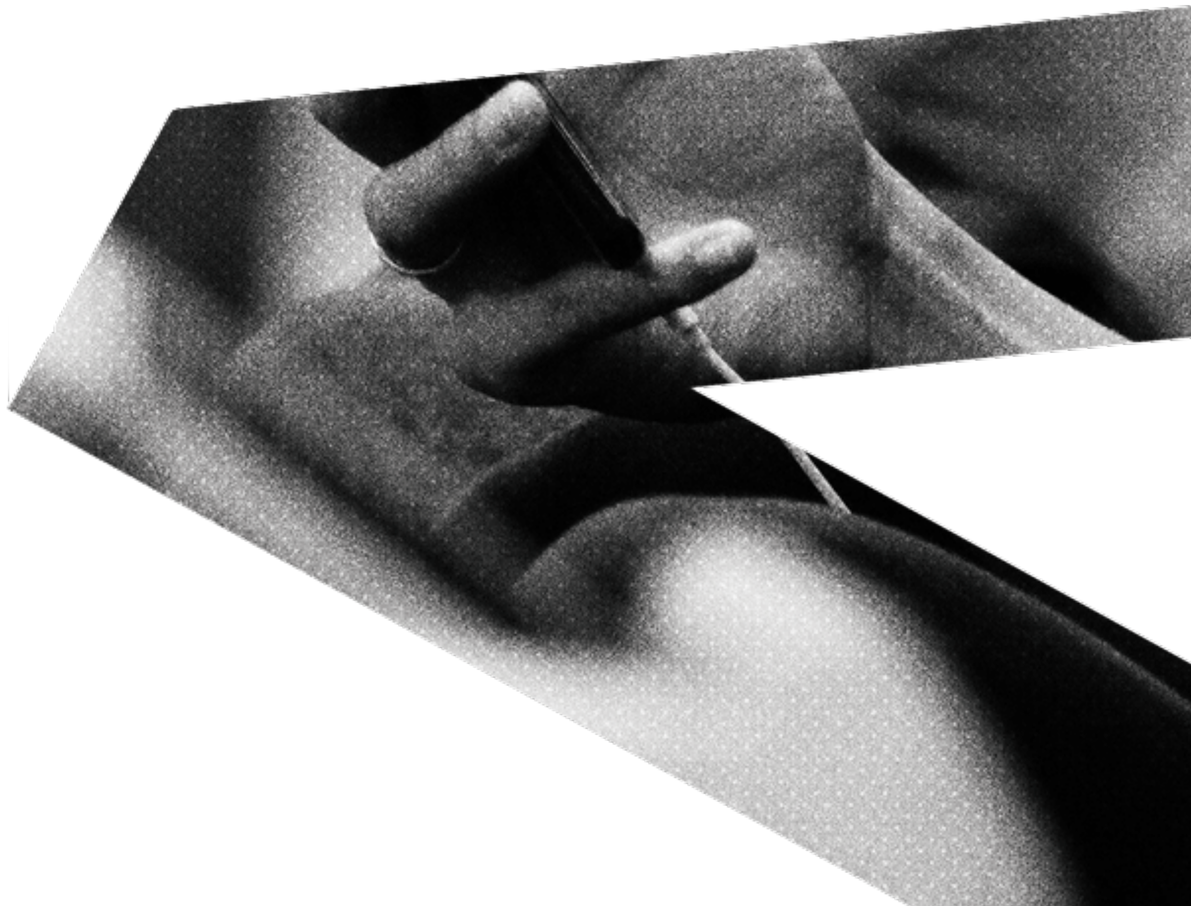
ZAMA

Introduction to Fully Homomorphic Encryption

FHE



ILARIA CHILLOTTI - ILARIA.CHILLOTTI@ZAMA.AI

DAMIEN LIGIER - DAMIEN.LIGIER@ZAMA.AI

JEAN-BAPTISTE ORFILA - JB.ORFILA@ZAMA.AI

SAMUEL TAP - SAMUEL.TAP@ZAMA.AI

ZAMA

agenda

- Introduction
- Toward Homomorphic Encryption
- Homomorphic Computation of any Circuit
- TFHE Bootstrap
- Conclusion

ZAMA

Introduction

What's Cryptography?

Scytale

From Ancient Greek:

- "*kryptós*" = secret
- "*graphein*" = to write

Enigma

Initially used for **military** reasons to avoid the enemy to intercept messages during wars

https://

Goals of a Classical Encryption Scheme

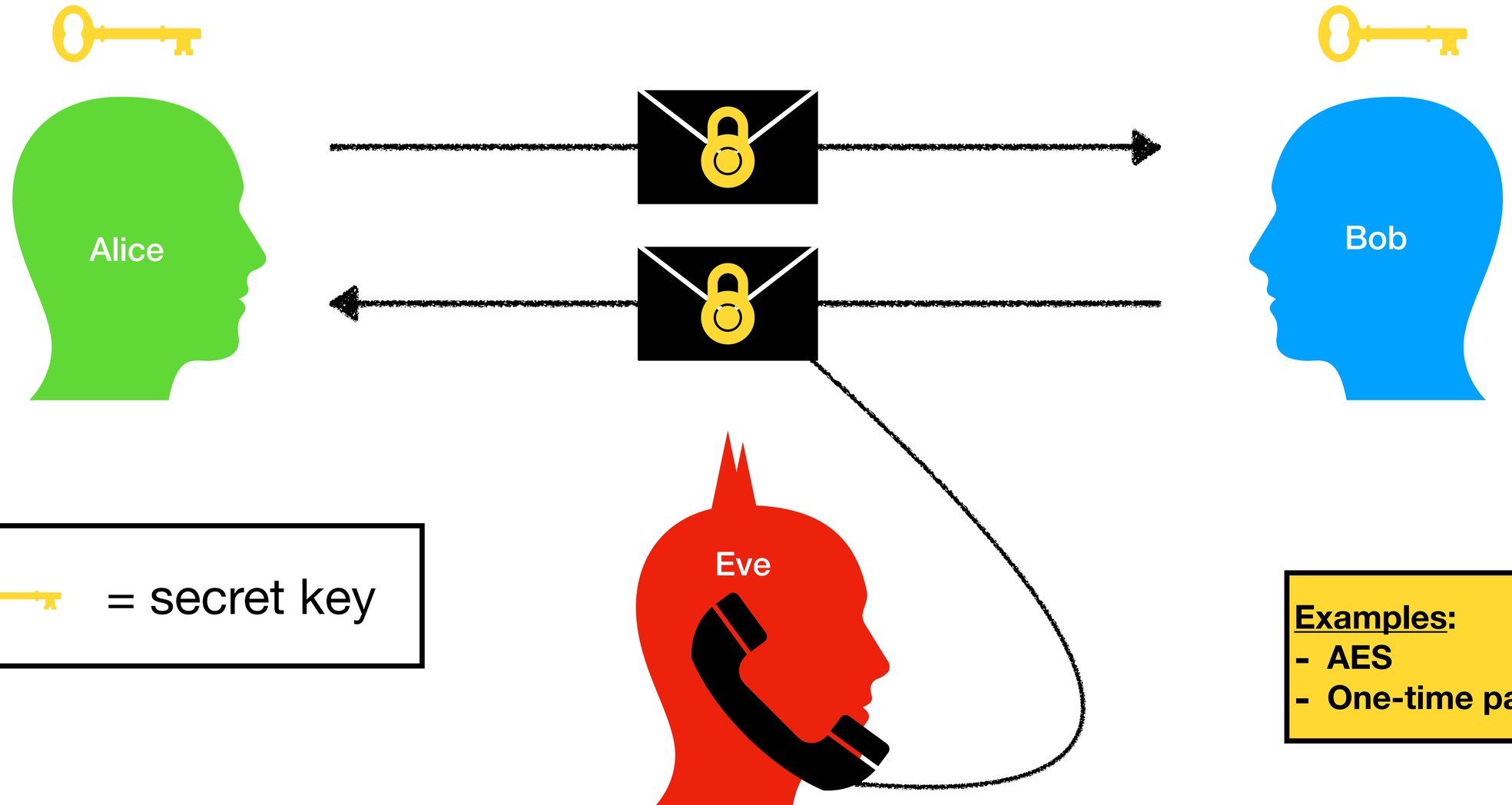
Authentication

CONFIDENTIALITY

Integrity

Kerckhoffs's principle: ¬ (security through obscurity)

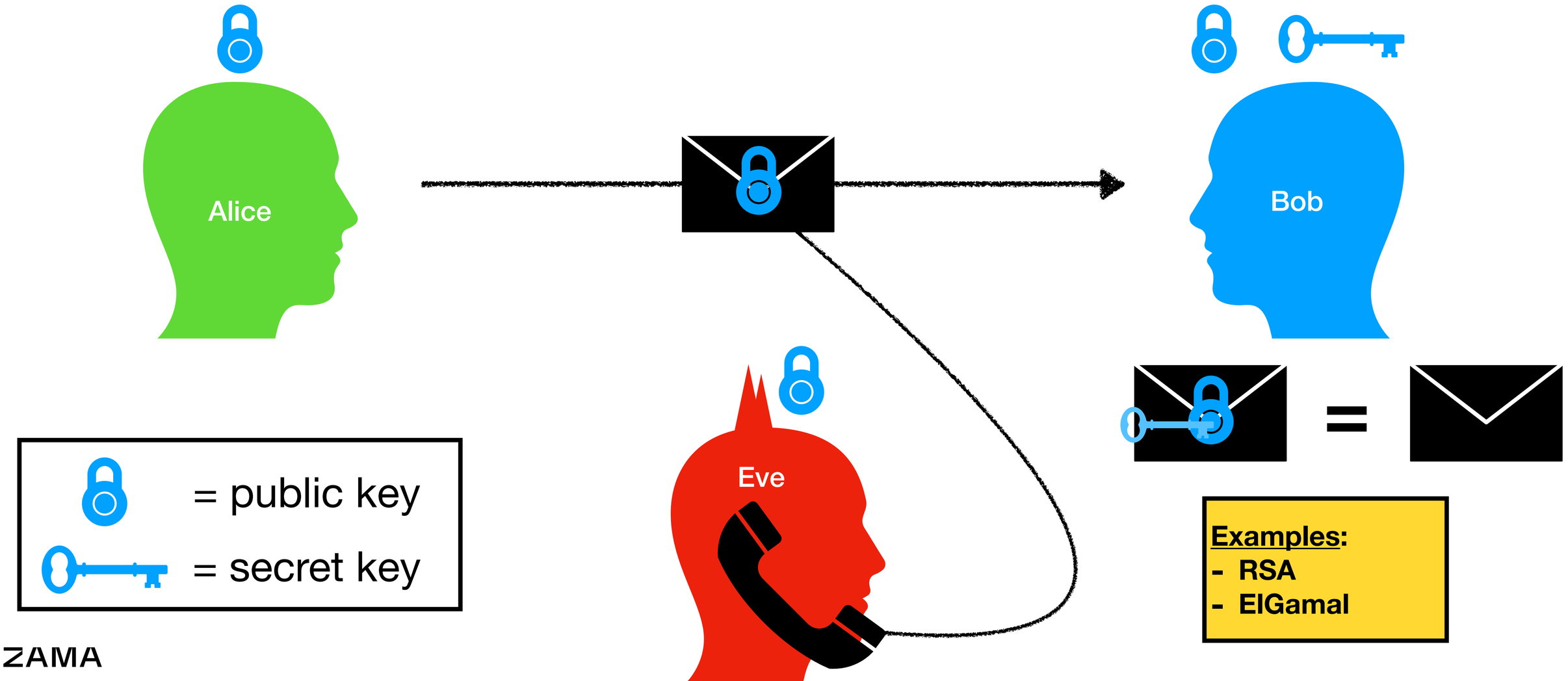
Private Key / Symmetric Encryption



= secret key

Examples:
- **AES**
- **One-time pad**

Public Key / Asymmetric Encryption



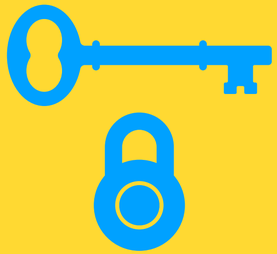
ZAMA

Toward Homomorphic Encryption

RSA Primitive [RSA78], an Example

Key Generation

- p,q: large prime numbers
- $n = pq$; $\phi(n) = (p-1)(q-1)$
- e s.t. $\gcd(e, \phi(n)) = 1$
- d s.t. $ed \equiv 1[\phi(n)]$



$= \{p, q, d\}$

$= \{n, e\}$

Encryption

$$c \equiv m^e[n]$$

Decryption

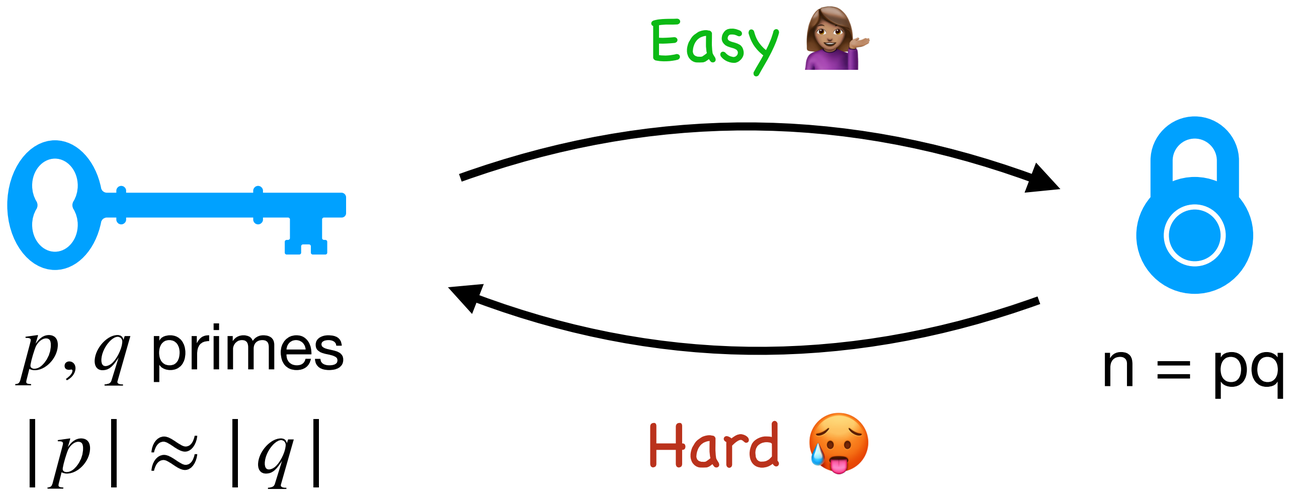
$$c^d \equiv (m^e)^d \equiv m[n]$$

RSA

How can it be secure ?

Hardness assumption

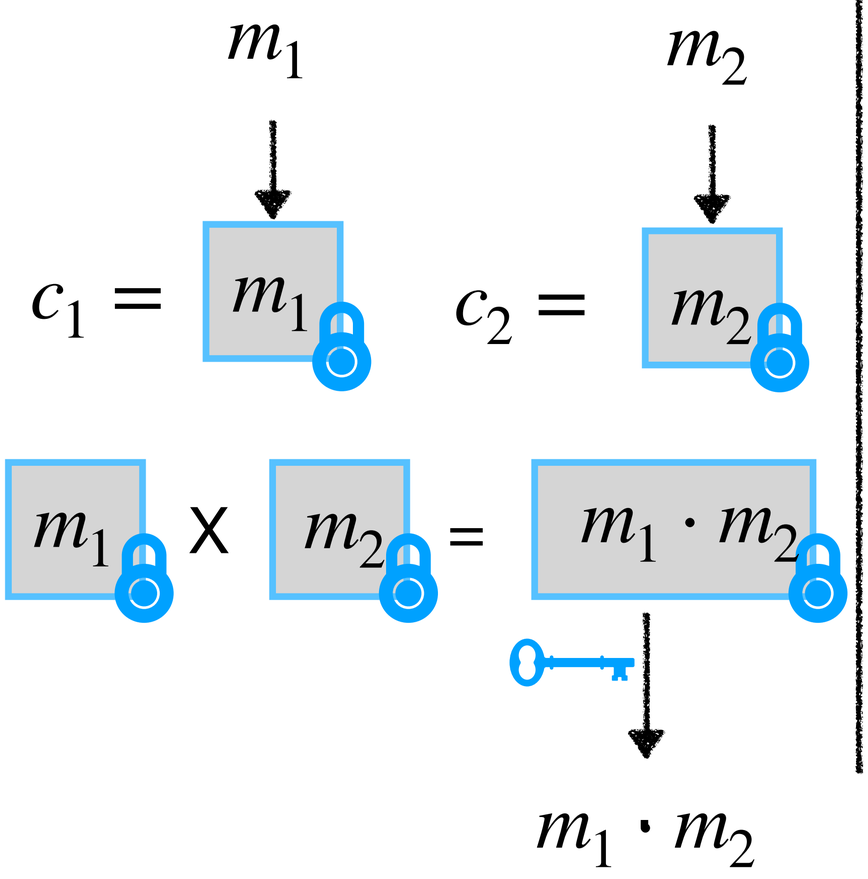
Factoring is hard



In practice: $|n| \geq 4096$ bits

Homomorphic Property of RSA

Encryption:



Product on ciphertexts

Decryption:

$= \{p, q, d\}$

$= \{n, e\}$

$$c_i \equiv m_i^e [n]$$

$$c_3 \equiv c_1 \times c_2 \equiv (m_1 \cdot m_2)^e [n]$$

$$c_3^d \equiv (m_1 \cdot m_2)^{ed} \equiv m_1 \cdot m_2 [n]$$

RSA primitive for FHE ?

- RSA Primitive is not **fully** homomorphic:

Only multiplications, not additions!

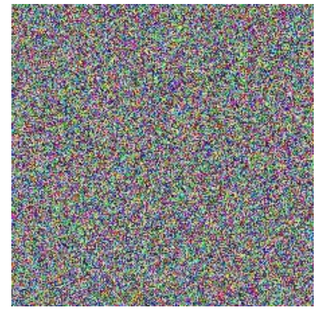
- RSA Primitive is not secure:

RSA Primitive

Encrypt

Initial Image

Encrypt



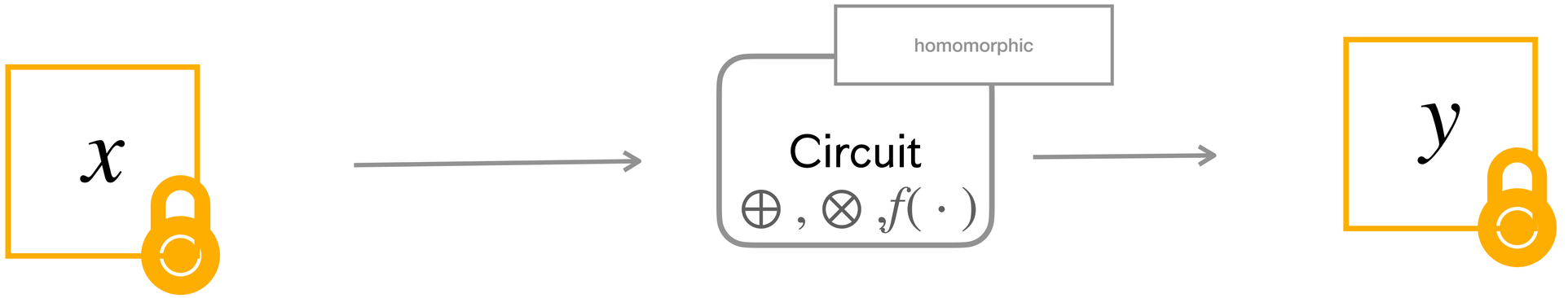
RSA-OAEP

⚠ RSA-OAEP: no homomorphism

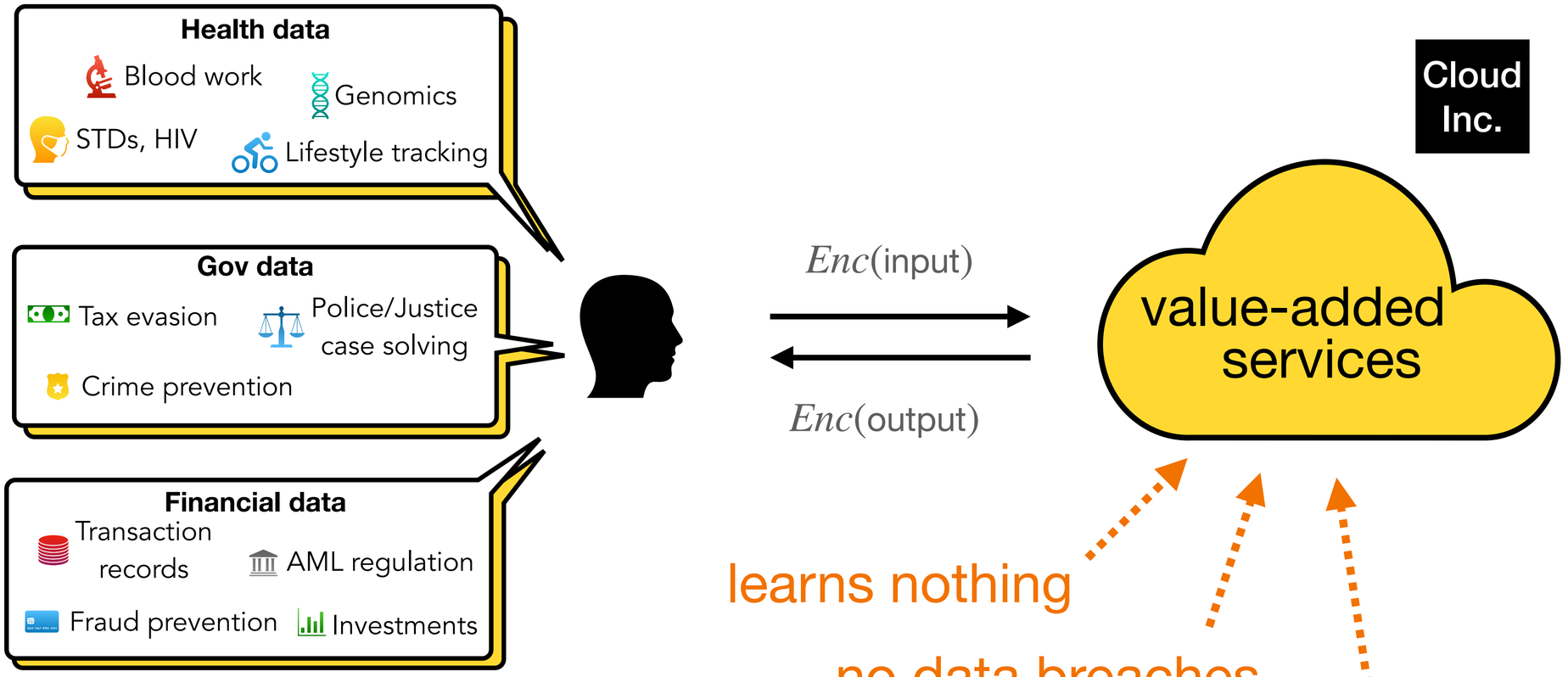
Goals of Homomorphic Encryption Scheme

CONFIDENTIALITY

Computation

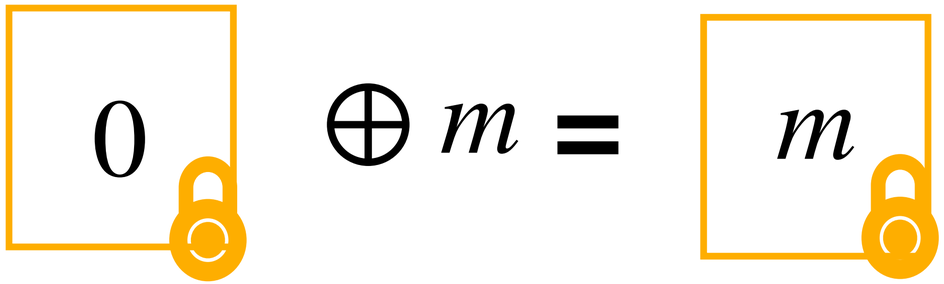


Where FHE Could Be Used IRL?



Symmetric vs Asymmetric Encryption in FHE

Symmetric Scheme ⇒Asymmetric Scheme



Solving the $\mathrm{LWE}_{q,\chi}$ Problem

for an integer q and a distribution χ - [Reg05]

$$\begin{cases} b_1 \equiv a_{1,1}s_1 + \cdots + a_{1,n}s_n & [q] \\ b_2 \equiv a_{2,1}s_1 + \cdots + a_{2,n}s_n & [q] \\ \vdots \\ b_l \equiv a_{l,1}s_1 + \cdots + a_{l,n}s_n & [q] \end{cases}$$

- uniform $a_{i,j} \in \mathbb{Z}/q\mathbb{Z}$
- $\mathbf{s} \in (\mathbb{Z}/q\mathbb{Z})^n$

LWE Problem:

- hard to find $\mathbf{s}$
- $\{b_i\}$ look like uniform

Hardness depends on the:

- size of the key **n**
- error distribution χ
- type of $\mathbf{s}$ *(uniform/gaussian/binary)*

Regev Cryptosystem [Reg05] - 1

Key Generation

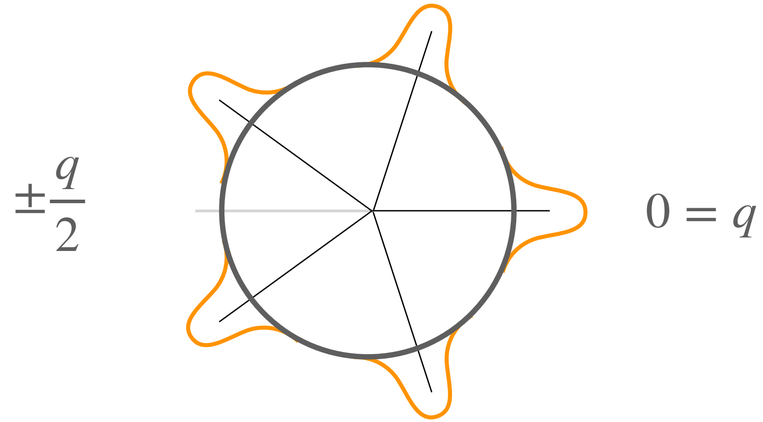
- Choose $(n, \sigma) \in (\mathbb{N}, \mathbb{R})$
- Draw $\mathbf{s}$ *s.t.* $\mathbf{s} \sim \mathcal{U}(\{0,1\}^n)$

Encryption

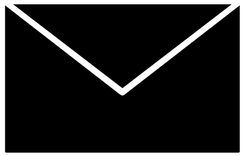
- Draw $\mathbf{a} \sim (\mathbb{U}(\mathbb{Z}/q\mathbb{Z}))^n$
- Draw $\epsilon \sim \mathcal{N}(0, q^2\sigma^2)$
- Compute $b = [\mathbf{a} \cdot \mathbf{s} + \dot{m} + \epsilon]_q$

Decryption

Compute $\lceil [b - \mathbf{a} \cdot \mathbf{s}]_q \rfloor = \lceil \dot{m} + \epsilon \rfloor = \dot{m}$



 $= \mathbf{s}$

 $= m$

 $= (\mathbf{a}, b)$

Regev Cryptosystem [Reg05] - 2

Addition

Inputs:

- $c_1 = (\mathbf{a}_1, b_1)$
- $c_2 = (\mathbf{a}_2, b_2)$

Outputs:

$$c = c_1 \oplus c_2 = ([\mathbf{a}_1 + \mathbf{a}_2]_q, [b_1 + b_2]_q)$$

c is a valid encryption of $[\dot{m}_1 + \dot{m}_2]_q$

Scalar multiplication

Inputs:

- $c_{in} = (\mathbf{a}, b)$
- $w \in \mathbb{Z}$

Outputs:

$$c_{out} = w \otimes c_{in} = ([w \cdot \mathbf{a}]_q, [w \cdot b]_q)$$

c_{out} is a valid encryption of $[w \cdot \dot{m}_{in}]_q$

Regev Cryptosystem [Reg05] - 3

Noise analysis *(scalar mult)*

Inputs:

- $c_{in} = (\mathbf{a}, b = \mathbf{a} \cdot \mathbf{s} + \dot{m} + \epsilon_{in})$ with
  $\epsilon_{in} \sim \mathcal{N}(0,(q\sigma)^2)$
- $w \in \mathbb{Z}$

Outputs:

$c_{out} = w \otimes c_{in} = ([w \cdot \mathbf{a}]_q, [w \cdot b]_q)$
with $\epsilon_{out} \sim \mathcal{N}(0,w(q\sigma)^2)$

The noise grows with each operation

RLWE [SSTX09, LPR10]

LWE variant over Rings

Instead of working in $\mathbb{Z}/q\mathbb{Z}$, we work on $\mathscr{R} = \left(\mathbb{Z}/q\mathbb{Z}[X]\right)/(X^N + 1)$

- We choose N a power of 2 $\Rightarrow X^N + 1$ is the $2N$ cyclotomic polynomial (irreducible)

Secret Key: $s \in \mathscr{R}$ with random binary coefficients

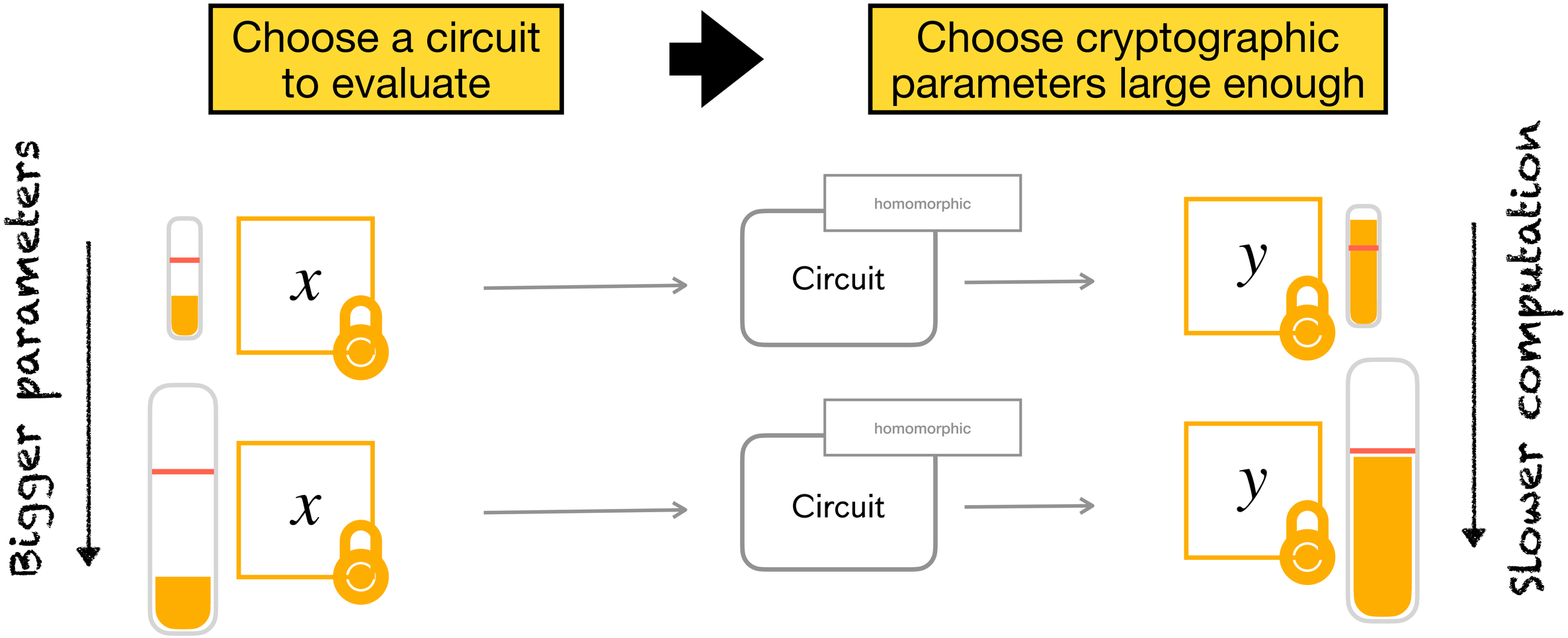
Encryption: a ciphertext is a pair $(a, b) \in \mathscr{R}^2$ such that $\begin{cases} a \text{ uniformly random} \\ b = a \cdot s + m + e \\ e \sim \mathcal{N}(0, q^2\sigma^2) \end{cases}$

Decryption: $\lceil [b - a \cdot s]_q \rfloor = \lceil m + e \rfloor = m$

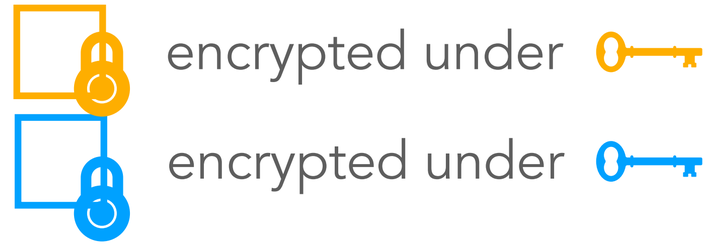
ZAMA

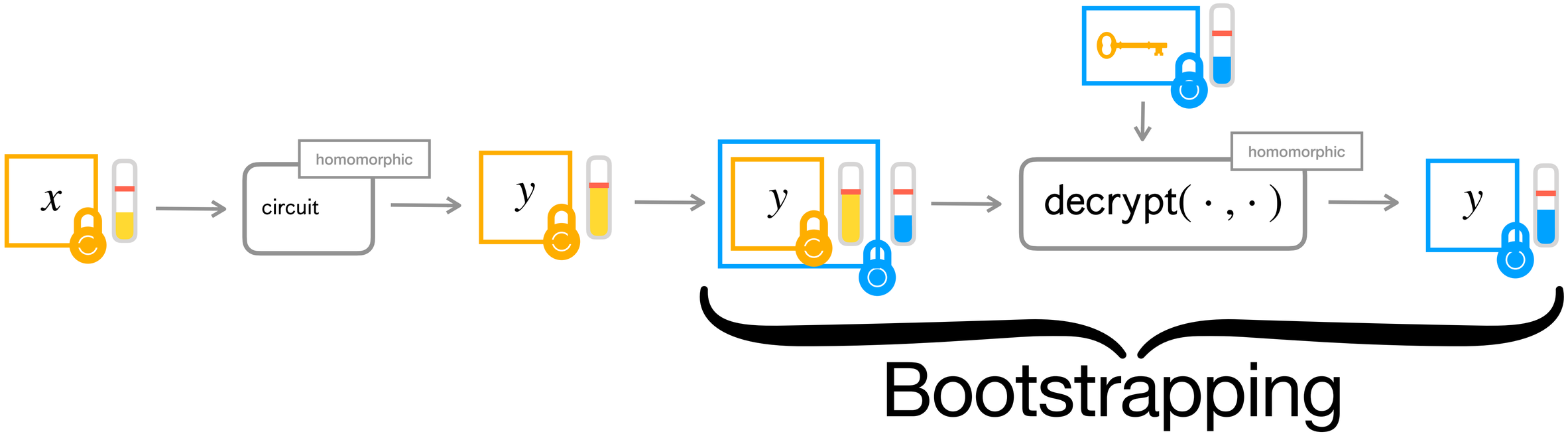
Towards Homomorphic Computation of Any Circuit

Somewhat HE Approach



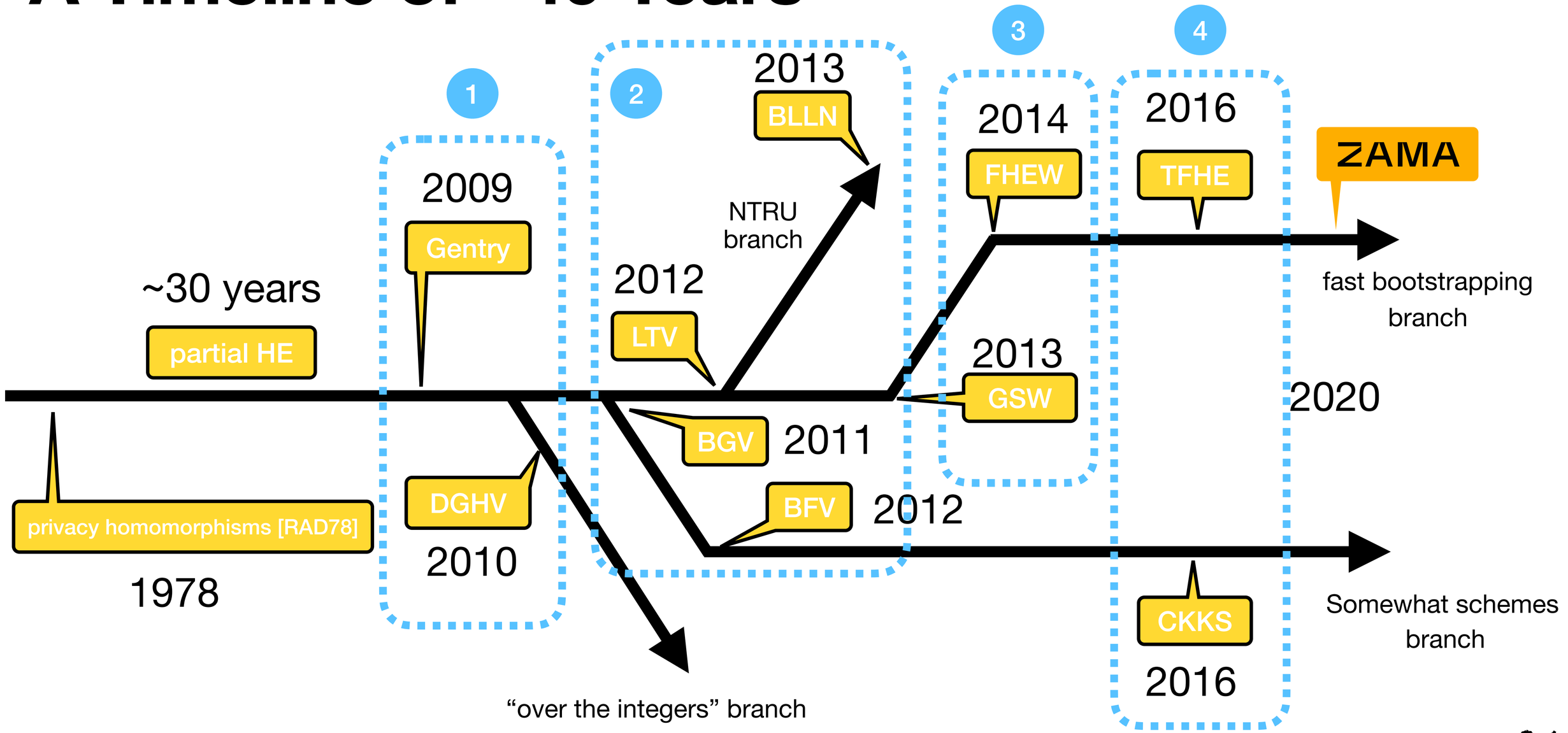
Bootstrapping [Gen09]





No need to know the circuit beforehand anymore

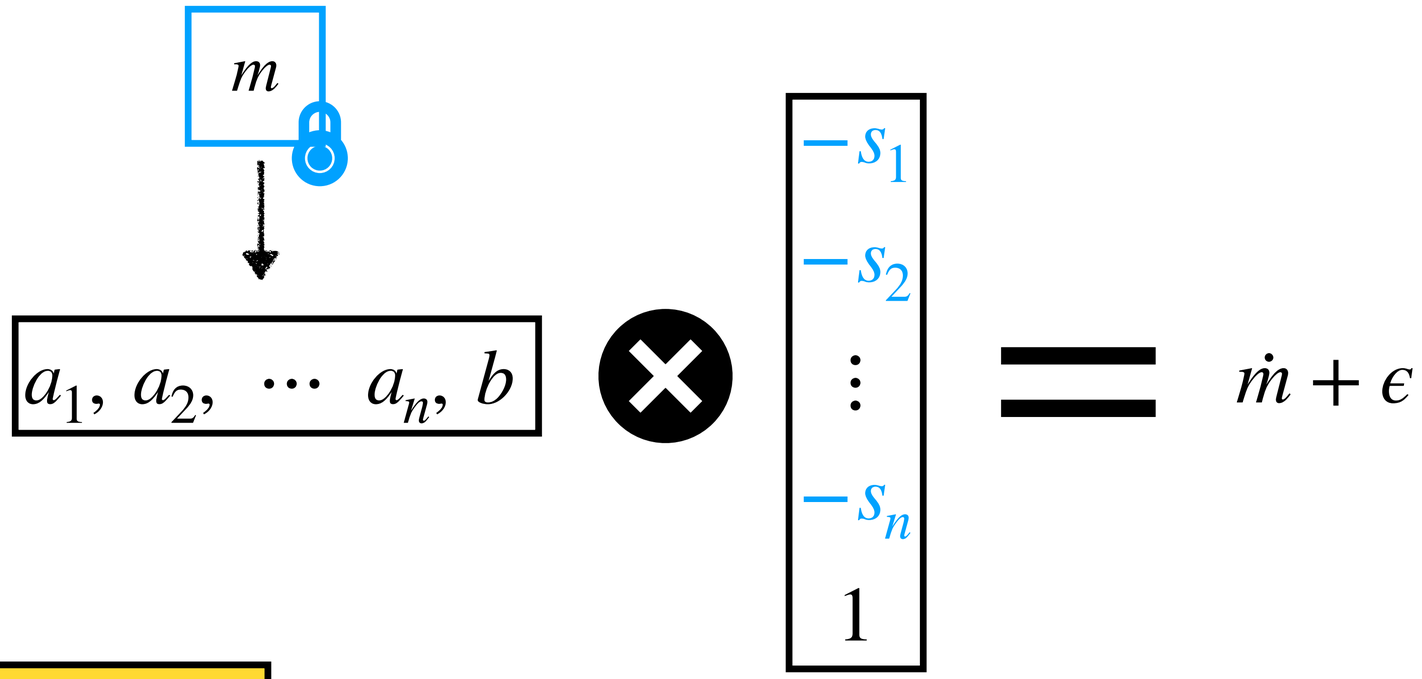
A Timeline of ~40 Years



ZAMA

TFHE Bootstrap [CGGI16]

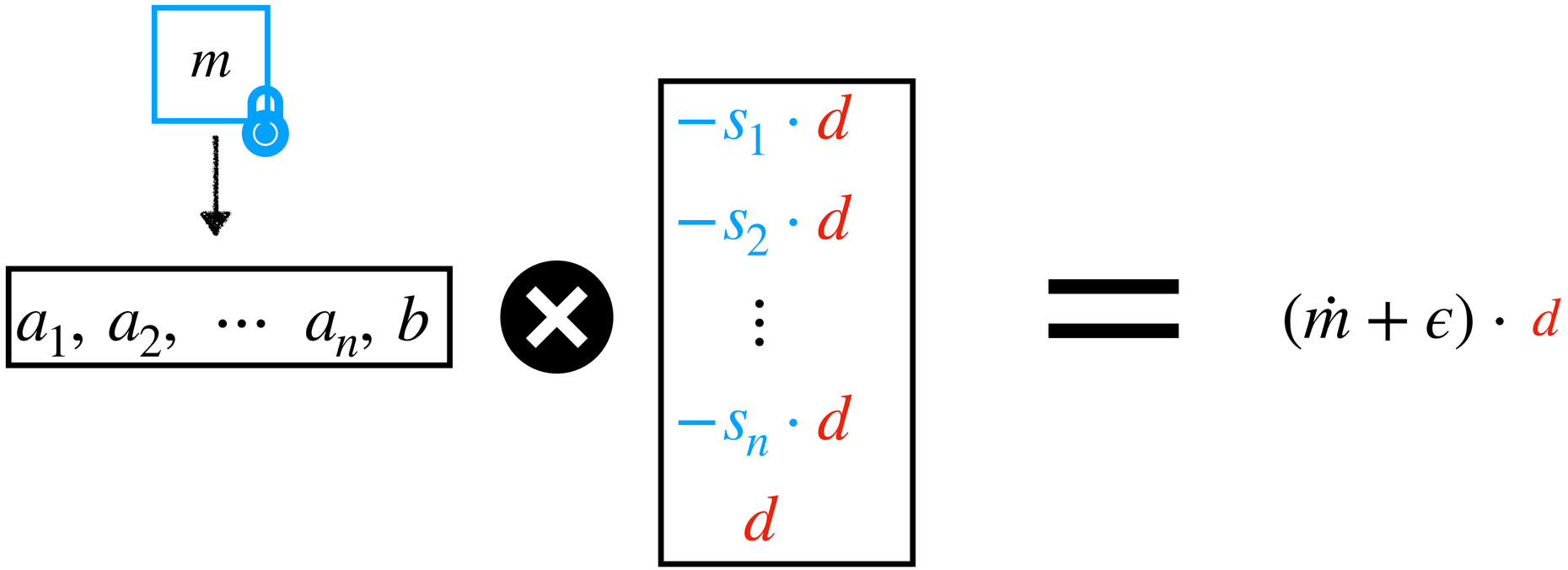
External Product - 1



Reminder:
$b = [\mathbf{a} \cdot \mathbf{s} + \dot{m} + \epsilon]_q$

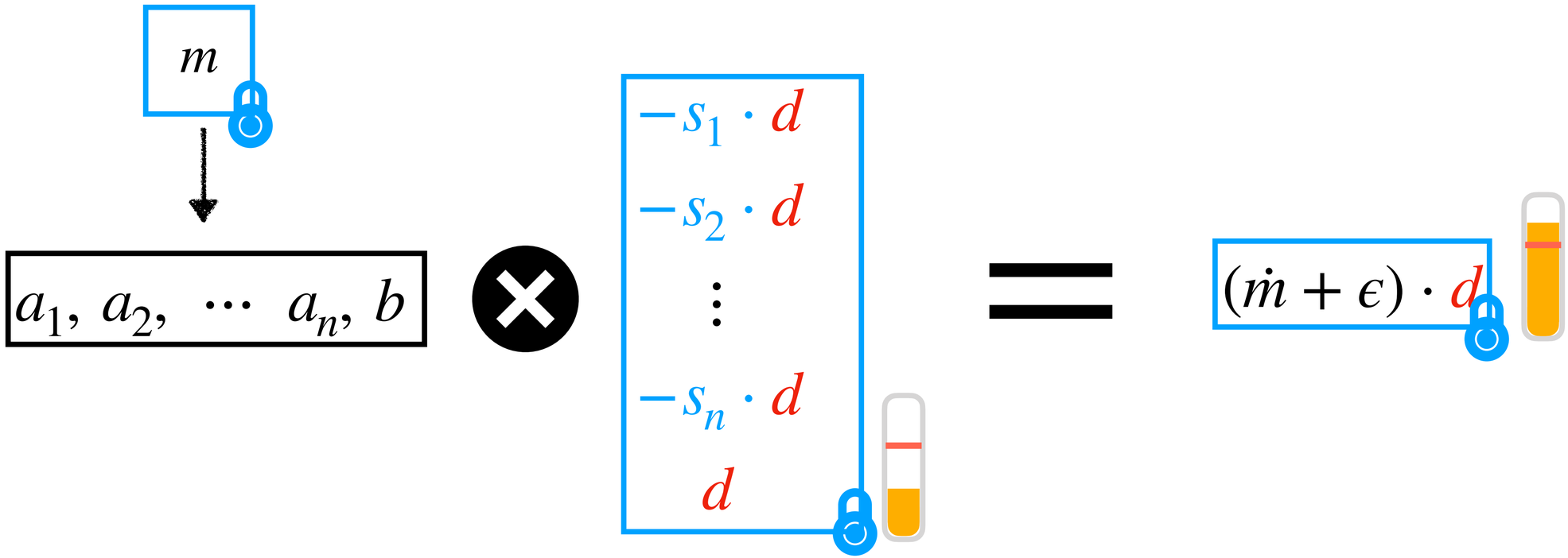
Decryption

External Product - 2



Multiplication and decryption

External Product - 3



Homomorphic Multiplication and Decryption

Multiply with Less Noise

$$a \cdot s \approx \langle \text{decomp}(a), \text{powers(s)} \rangle$$

a base B

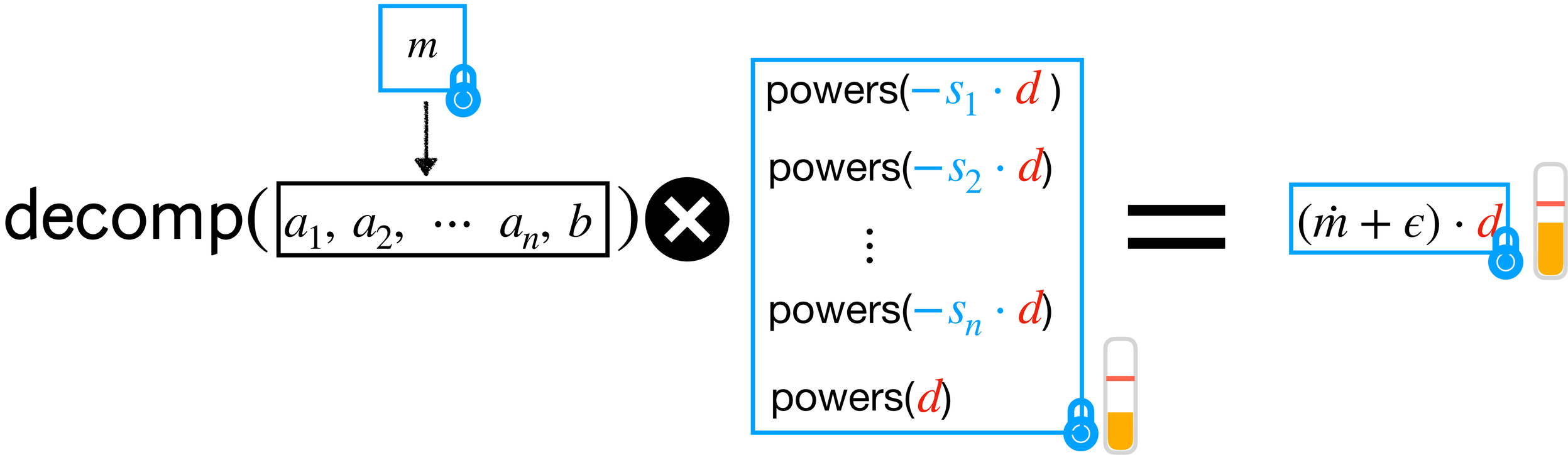
$$a \approx \sum_{i=0}^{l} a_i \cdot B^i$$

$$0 \leq a_i < B$$

$$\text{decomp}(a) = (a_l, \cdots, a_0)$$

$$\text{powers}(s) = (s \cdot B^l, \cdots, s \cdot B^0)$$

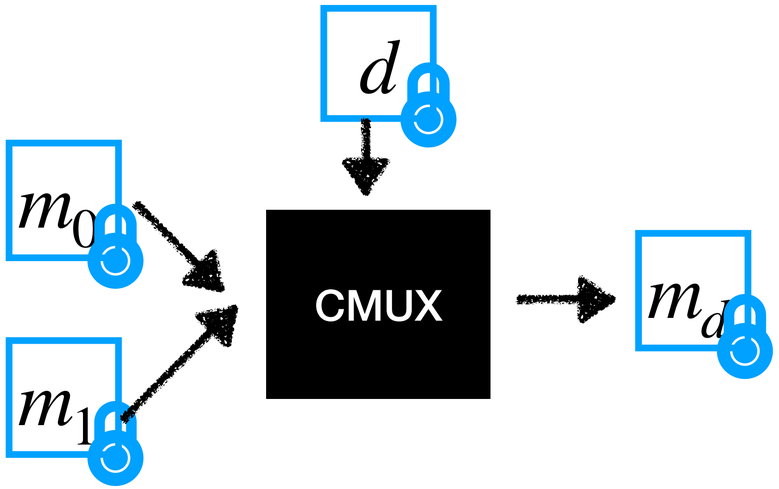
External Product - 4



Homomorphic Multiplication with less noise

but requires more space and more computation

CMUX



$$\mathsf{CMUX}(E(m_0), E(m_1), \mathscr{E}(d)) = E(m_d)$$

$$= (E(m_1) - E(m_0)) \odot \mathscr{E}(d) + E(m_0)$$

Homomorphic selection

Some Facts on $\mathbb{Z}[X]/(X^N + 1)$

$$P(X) = \begin{array}{|c|c|c|c|c|c|c|}\hline f_0 & f_1 & \dots & \dots & \dots & \dots & f_{N-1} \\ \hline \end{array} = \sum_{i=0}^{N-1} f_i X^i$$

$$P(X) \cdot X^{-1} = \begin{array}{|c|c|c|c|c|c|c|}\hline f_1 & f_2 & \dots & \dots & \dots & \dots & -f_0 \\ \hline \end{array}$$

$$P(X) \cdot X^{-N} = \begin{array}{|c|c|c|c|c|c|c|}\hline -f_0 & -f_1 & \dots & \dots & \dots & \dots & -f_{N-1} \\ \hline \end{array}$$

$$P(X) \cdot X^{-2N} = \begin{array}{|c|c|c|c|c|c|c|}\hline f_0 & f_1 & \dots & \dots & \dots & \dots & f_{N-1} \\ \hline \end{array}$$

$$X^{2N} = 1$$

$$X^m \mod 2N = X^m$$

Modulus Switching

$$\text{ms} : \mathbb{Z}/q\mathbb{Z} \to \mathbb{Z}/2N\mathbb{Z}$$
$$x \mapsto \lceil x \cdot 2N/q \rfloor$$

Reminder:
$b = [\mathbf{a} \cdot \mathbf{s} + \dot{m} + \epsilon]_q$

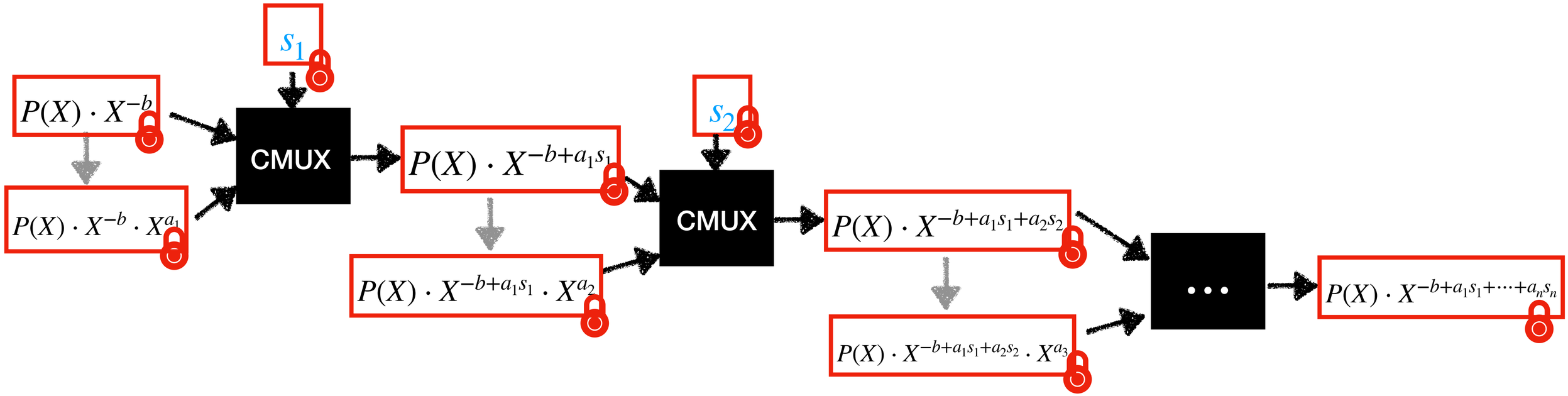
$$\text{ms}(b) - \sum \text{ms}(a_i) \cdot s_i = \text{ms}(\dot{m}) + \underbrace{\epsilon'}$$

with $\epsilon' > \epsilon$

Noise

Bootstrap

Inputs: $(a_1, a_2, \cdots a_n, b) =$ m , s_1 , $\cdots$, s_n



Step 0

ACC $\leftarrow P(X) \cdot X^{-b}$

Step 1

ACC $\leftarrow P(X) \cdot X^{-b+a_1 \cdot s_1}$

Step n

ACC $\leftarrow P(X) \cdot X^{-b+\sum a_i \cdot s_i}$

$\approx P(X) \cdot X^{-m}$

Lookup Table

$P(X) = \sum p_i \cdot X^i$ is actually a *lookup table* 🤯

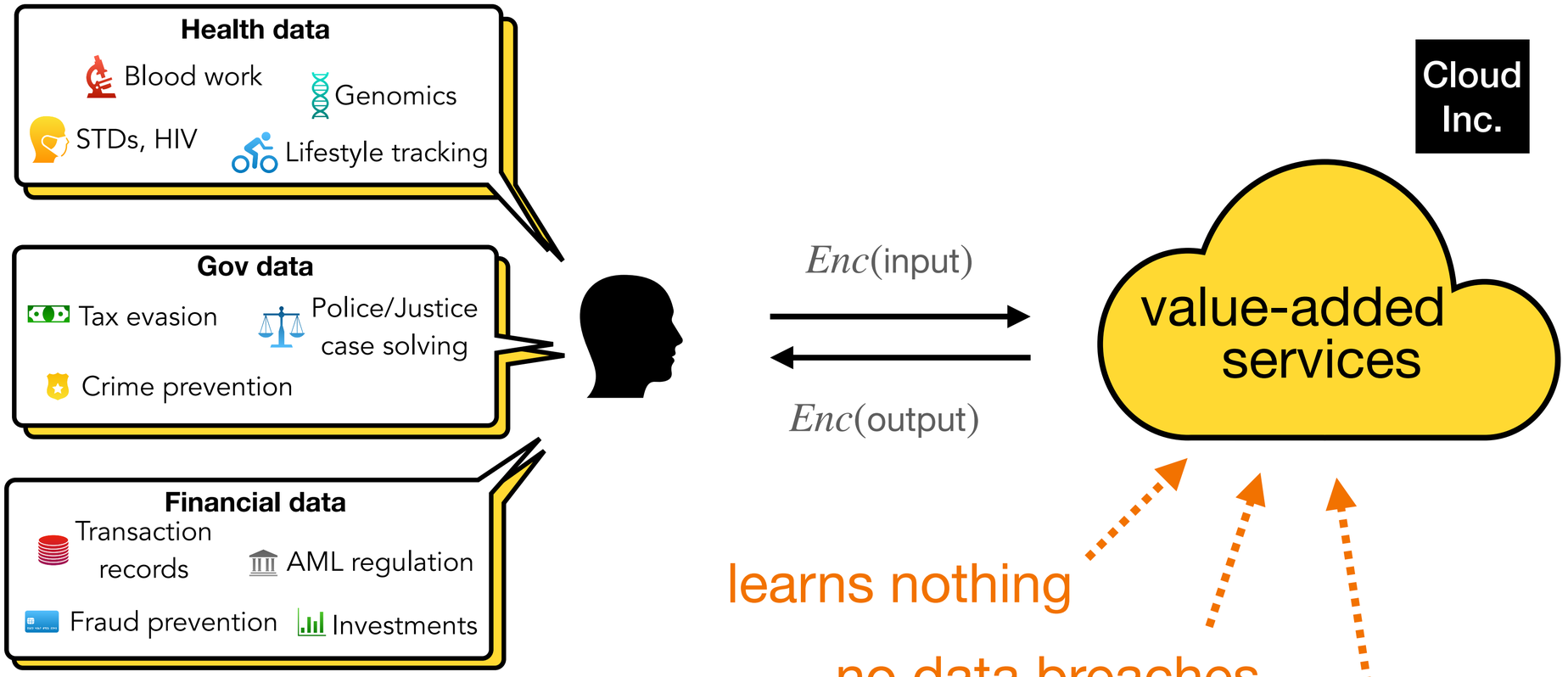
i	0	1	2	...	$N-1$
p_i	$f(0)$	$f(1)$	$f(2)$	...	$f(N-1)$

the constant term of $P(X) \cdot X^{-m}$ is: $p_m = f(m)$ 🥳

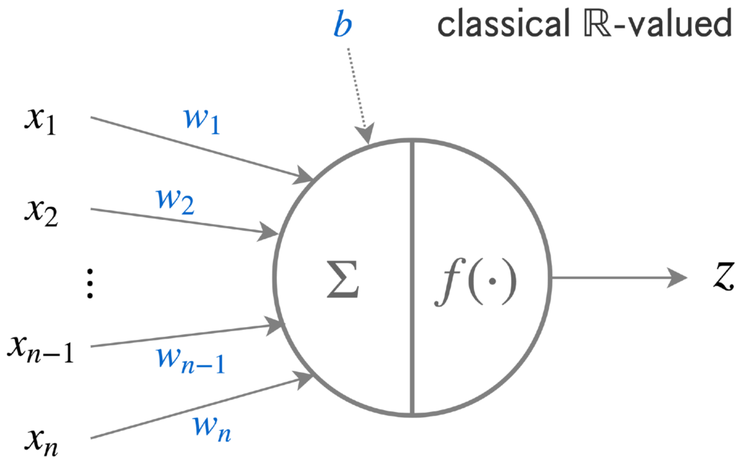
ZAMA

Conclusion

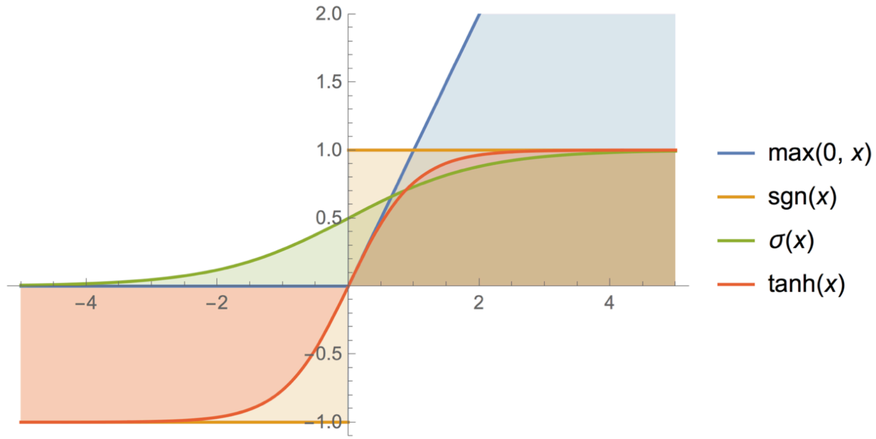
Where FHE Could Be Used IRL?



Application to Machine Learning



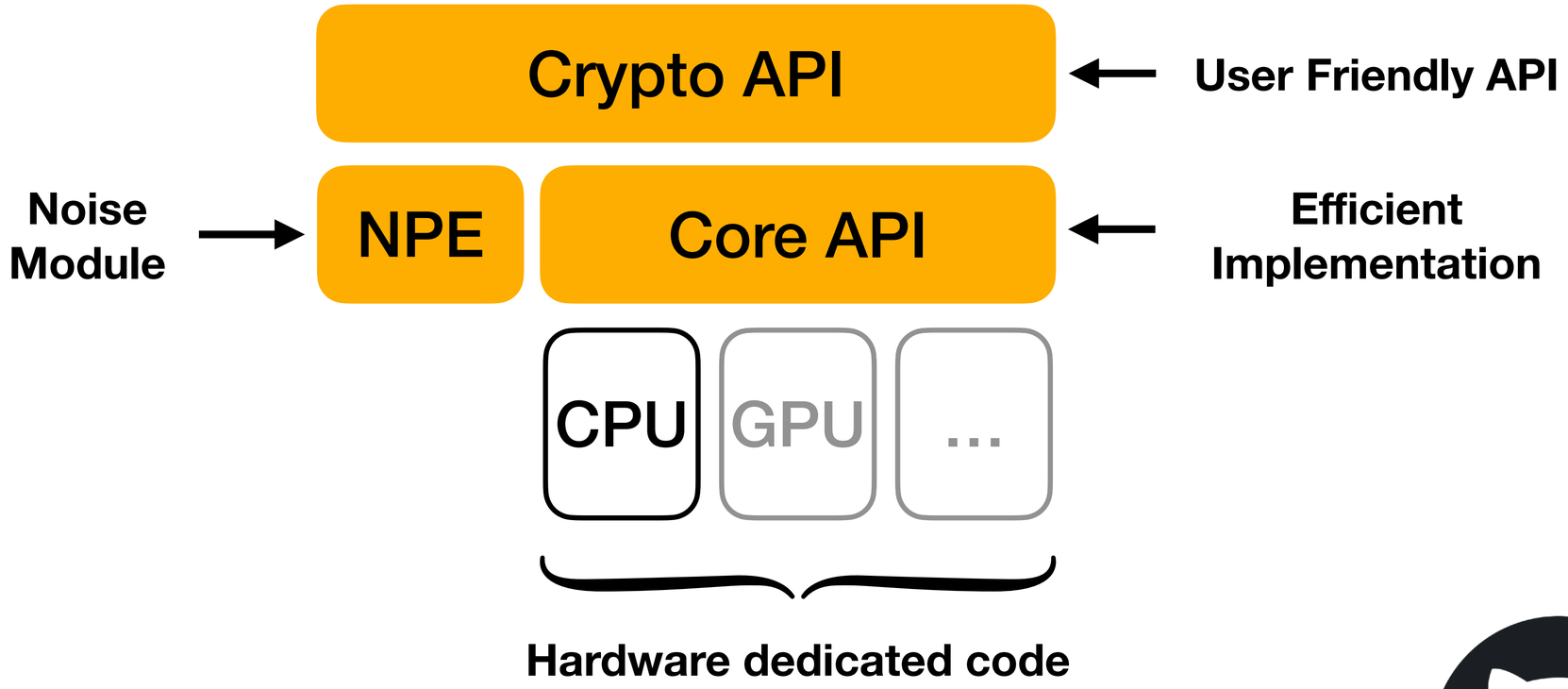
$$y = \sum_{i=1}^{n} w_i x_i + b \qquad z = f(y)$$



activation function

use PBS instead

Concrete Library



https://github.com/zama-ai/concrete

ZAMA

Thank you!

Questions?

Bibliography

[RSA78] R. L. Rivest, A. Shamir, L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 1978.

[Reg05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. STOC 2005.

[SSTX09] D. Stehlé, R. Steinfeld, K. Tanaka, K. Xagawa. Efficient public key encryption based on ideal lattices. ASIACRYPT 2009.

[LPR10] V. Lyubashevsky, C. Peikert, O. Regev. On ideal lattices and learning with errors over rings. EUROCRYPT 2010.

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. STOC 2009.

[RAD78] R. L. Rivest, L. Adleman, M. L. Dertouzos. On data banks and privacy homomorphisms. Foundations of secure computation 1978.

[DGHV10] M. van Dijk, C. Gentry, S. Halevi, V. Vaikuntanathan. Fully homomorphic encryption over the integers. EUROCRYPT 2010.

[BGV12] Z. Brakerski, C. Gentry, V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. ITCS 2012.

[Bra12] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. CRYPTO 2012.

[FV12] J. Fan, F. Vercauteren. Somewhat practical fully homomorphic encryption. IACR Cryptology ePrint Archive, 2012.

[CKKS17] J. H. Cheon, A. Kim, M. Kim, Y. Song. Homomorphic encryption for arithmetic of approximate numbers. ASIACRYPT 2017.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. CRYPTO 2013.

[DM15] L. Ducas, D. Micciancio. FHEW: bootstrapping homomorphic encryption in less than a second. EUROCRYPT 2015.

[CGGI16] I. Chillotti, N. Gama, M. Georgieva, M. Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. ASIACRYPT 2016.