ZAMA

FULLY HOMOMORPHIC ENCRYPTION OVER THE [DISCRETIZED] TORUS

CHES 2022 · Leuven, September 18–21, 2022

Marc Joye

THE CLOUD NEEDS BETTER DATA SECURITY

Even the best companies sometimes make mistakes

OUTLINE

Fully Homomorphic Encryption

Gentry's Recryption

(Programmable) Bootstrapping

Functional Circuits

Numerical Experiments

OUTLINE

Fully Homomorphic Encryption

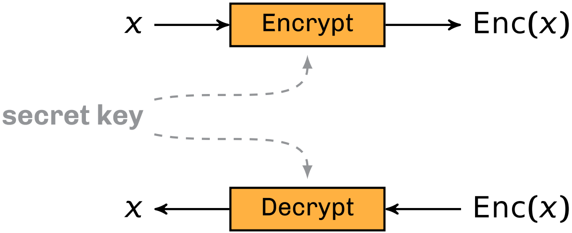
Gentry’s Recryption

(Programmable) Bootstrapping

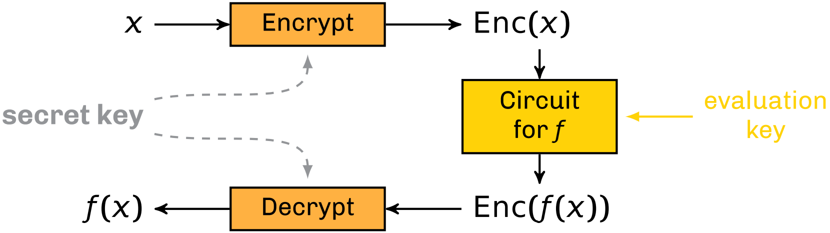
Functional Circuits

Numerical Experiments

WHAT IS FULLY HOMOMORPHIC ENCRYPTION?

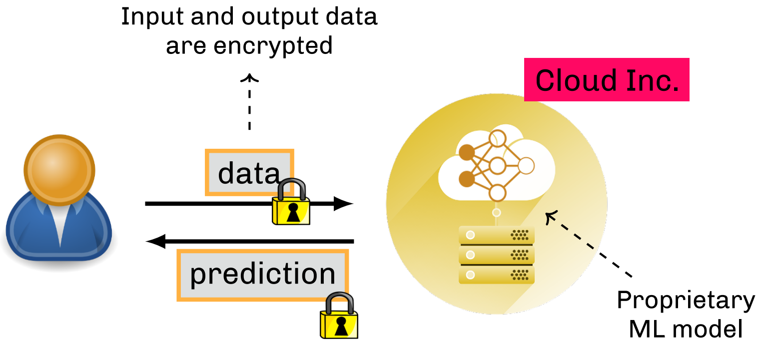


WHAT IS FULLY HOMOMORPHIC ENCRYPTION?

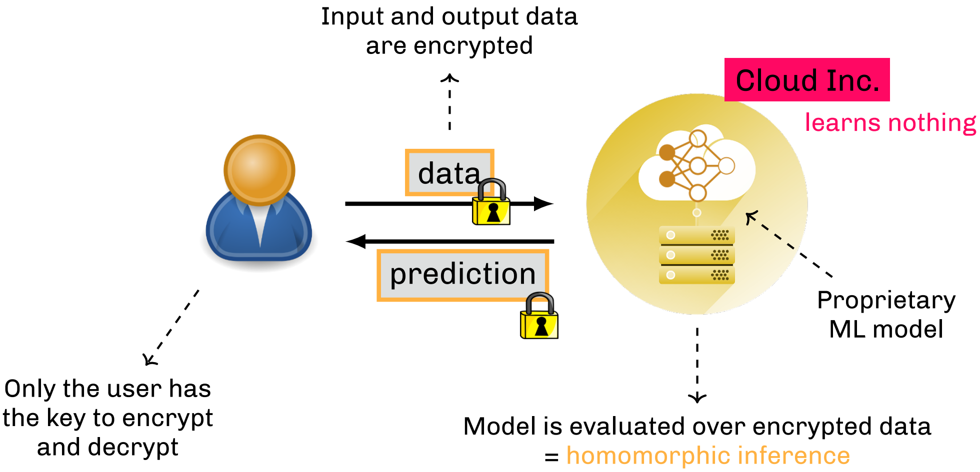


Remark: Any private-key FHE scheme can easily be turned into a public-key FHE scheme

EMPOWERING MACHINE LEARNING WITH FHE



EMPOWERING MACHINE LEARNING WITH FHE



FIRST GENERATION FHE (2009)

PERFORMANCE

$x, y \in \{0, 1\}$	$\text{Enc}(x), \text{Enc}(y) \rightsquigarrow \text{Enc}(x \oplus y)$	pretty fast
	$\text{Enc}(x), \text{Enc}(y) \rightsquigarrow \text{Enc}(x \wedge y)$	super slow

$\oplus$ and $\wedge$ = all operations

FIRST GENERATION FHE (2009)

PERFORMANCE

$x, y \in \{0, 1\}$

$\text{Enc}(x), \text{Enc}(y) \rightsquigarrow \text{Enc}(x \oplus y)$ pretty fast

$\text{Enc}(x), \text{Enc}(y) \rightsquigarrow \text{Enc}(x \wedge y)$ super slow

$\oplus$ and $\wedge$ = all operations

NOISE PROPAGATION

$x, y \in \{0, 1\}$

$\text{Enc}(x), \text{Enc}(y) \rightsquigarrow \text{Enc}(x \oplus y)$ noise size $\sim$ the same

$\text{Enc}(x), \text{Enc}(y) \rightsquigarrow \text{Enc}(x \wedge y)$ noise size doubles

If noise exceeds a threshold, the ciphertext looses “decryptability”

$\Longrightarrow$ One must resort to bootstrapping, a very slow noise-cleaning operation

OUTLINE

Fully Homomorphic Encryption

Gentry's Recryption

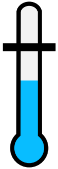
(Programmable) Bootstrapping

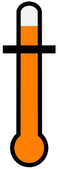
Functional Circuits

Numerical Experiments

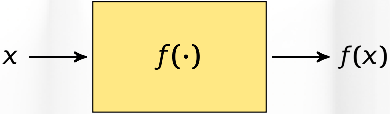
CONTROLLING THE NOISE

There is a notion of noise in ciphertexts

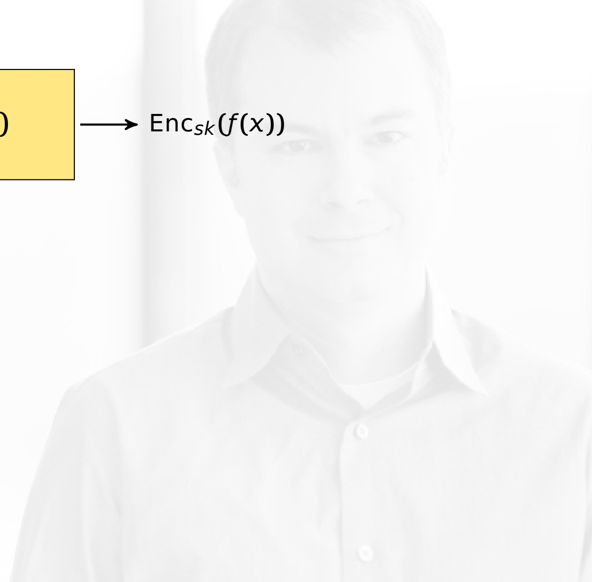
Enc(x)  ✔ valid ciphertext

Enc(x)  ✖ incorrect decryption

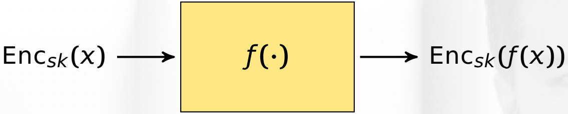
Noise accumulates over time


x
$f(\cdot)$
$f(x)$

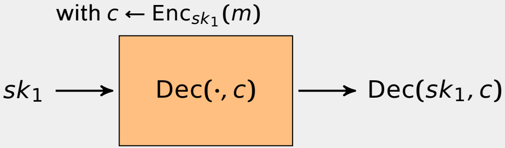
$$\text{Enc}_{sk}(x) \longrightarrow \boxed{f(\cdot)} \longrightarrow \text{Enc}_{sk}(f(x))$$



GENTRY'S RECRYPTION



APPLICATION: RECRYPTION



GENTRY'S RECRYPTION

$$\mathsf{Enc}_{sk}(x) \longrightarrow \boxed{f(\cdot)} \longrightarrow \mathsf{Enc}_{sk}(f(x))$$

APPLICATION: RECRYPTION

with $c \leftarrow \mathsf{Enc}_{sk_1}(m)$

$$\mathsf{Enc}_{sk_2}(sk_1) \longrightarrow \boxed{\mathsf{Dec}(\cdot, c)} \longrightarrow \mathsf{Enc}_{sk_2}(\mathsf{Dec}(sk_1, c))$$

GENTRY'S RECRYPTION

$$\mathsf{Enc}_{sk}(x) \longrightarrow \boxed{f(\cdot)} \longrightarrow \mathsf{Enc}_{sk}(f(x))$$

APPLICATION: RECRYPTION

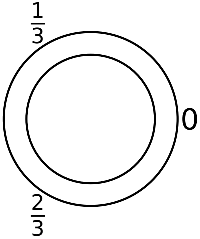
with $c \leftarrow \mathsf{Enc}_{sk_1}(m)$

$$\mathsf{Enc}_{sk_2}(sk_1) \longrightarrow \boxed{\mathsf{Dec}(\cdot, c)} \longrightarrow \mathsf{Enc}_{sk_2}(\mathsf{Dec}(sk_1, c))$$

$$= \mathsf{Enc}_{sk_2}(m)$$

TORUS FHE a.k.a. TFHE

secret key: $s \in \mathbb{B}^n$

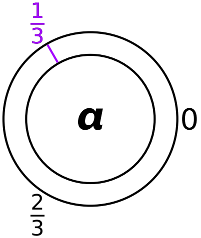


ENCRYPTION

DECRYPTION

TORUS FHE a.k.a. TFHE

secret key: $\boldsymbol{s} \in \mathbb{B}^n$



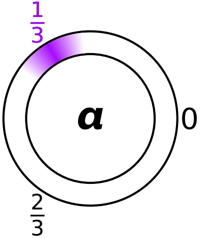
ENCRYPTION

1 $\boldsymbol{a} \xleftarrow{\$} \mathbb{T}^n$ (mask)

DECRYPTION

TORUS FHE a.k.a. TFHE

secret key: $\boldsymbol{s} \in \mathbb{B}^n$

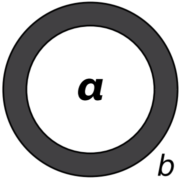


ENCRYPTION

1. $\boldsymbol{a} \xleftarrow{\$} \mathbb{T}^n$ (mask)
2. $\mu^* := \mu + e$ with $e \leftarrow \mathcal{N}(0, \sigma^2)$

DECRYPTION

secret key: $\mathbf{s} \in \mathbb{B}^n$



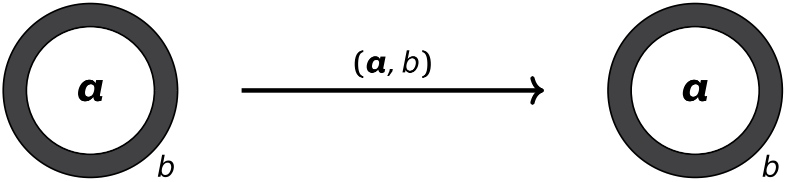
ENCRYPTION

1. $\mathbf{a} \xleftarrow{\$} \mathbb{T}^n$ (mask)
2. $\mu^* := \mu + e$ with $e \leftarrow \mathcal{N}(0, \sigma^2)$
3. $b \leftarrow \mu^* + \langle \mathbf{s}, \mathbf{a} \rangle$ (body)

DECRYPTION

TORUS FHE a.k.a. TFHE

secret key: $\boldsymbol{s} \in \mathbb{B}^n$



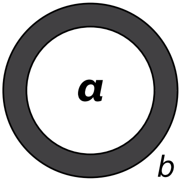
ENCRYPTION

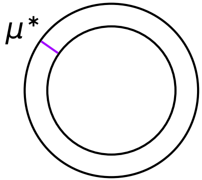
1. $\boldsymbol{a} \xleftarrow{\$} \mathbb{T}^n$ (mask)
2. $\mu^* := \mu + e$ with $e \leftarrow \mathcal{N}(0, \sigma^2)$
3. $b \leftarrow \mu^* + \langle \boldsymbol{s}, \boldsymbol{a} \rangle$ (body)

DECRYPTION

TORUS FHE a.k.a. TFHE

secret key: $\mathbf{s} \in \mathbb{B}^n$





ENCRYPTION

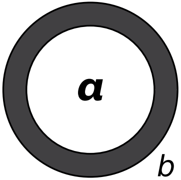
1. $\mathbf{a} \xleftarrow{\$} \mathbb{T}^n$ (mask)
2. $\mu^* := \mu + e$ with $e \leftarrow \mathcal{N}(0, \sigma^2)$
3. $b \leftarrow \mu^* + \langle \mathbf{s}, \mathbf{a} \rangle$ (body)

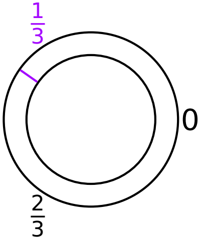
DECRYPTION

1. $\mu^* \leftarrow b - \langle \mathbf{s}, \mathbf{a} \rangle$

TORUS FHE a.k.a. TFHE

secret key: $\boldsymbol{s} \in \mathbb{B}^n$





ENCRYPTION

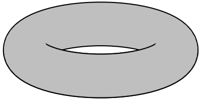
1. $\boldsymbol{a} \xleftarrow{\$} \mathbb{T}^n$ (mask)
2. $\mu^* := \mu + e$ with $e \leftarrow \mathcal{N}(0, \sigma^2)$
3. $b \leftarrow \mu^* + \langle \boldsymbol{s}, \boldsymbol{a} \rangle$ (body)

DECRYPTION

1. $\mu^* \leftarrow b - \langle \boldsymbol{s}, \boldsymbol{a} \rangle$
2. round μ^* to the closest value in $\mathcal{P}$ (plaintext space)

IN PRACTICE...

$$\mathbb{T} = \mathbb{R}/\mathbb{Z} = \{\text{real numbers modulo 1}\}$$

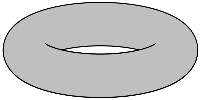


IN THEORY

- $t \in \mathbb{T}$
  $= \sum_{i=1}^{\infty} t_i 2^{-i}$
  $= 0.t_1 t_2 t_3 t_4 \ldots$

IN PRACTICE…

$$\mathbb{T} = \mathbb{R}/\mathbb{Z} = \{\text{real numbers modulo } 1\}$$



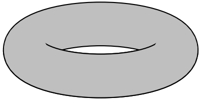
IN THEORY

- $t \in \mathbb{T}$
  $= \sum_{i=1}^{\infty} t_i \, 2^{-i}$
  $= 0.t_1 t_2 t_3 t_4 \ldots$

FINITE PRECISION (ℓ BITS)

- $t = \sum_{i=1}^{\ell} t_i \, 2^{-i}$
  $= \dfrac{\sum_{i=0}^{\ell-1} t_{\ell-i} \, 2^{i}}{q}$ where $q = 2^{\ell}$

$\mathbb{T} = \mathbb{R}/\mathbb{Z} = \{\text{real numbers modulo } 1\}$



$\rightsquigarrow$ subset $\mathbb{T}_q := \frac{1}{q}\mathbb{Z}/\mathbb{Z}$

with representatives $\{0, \frac{1}{q}, \ldots, \frac{q-1}{q}\}$

IN THEORY

- $t \in \mathbb{T}$
  $= \sum_{i=1}^{\infty} t_i\, 2^{-i}$
  $= 0.t_1 t_2 t_3 t_4 \ldots$

FINITE PRECISION (ℓ BITS)

- $t = \sum_{i=1}^{\ell} t_i\, 2^{-i}$
  $= \dfrac{\sum_{i=0}^{\ell-1} t_{\ell-i}\, 2^i}{q}$ where $q = 2^\ell$

OUTLINE

Fully Homomorphic Encryption

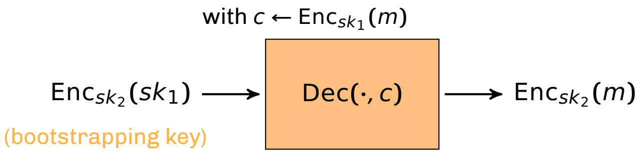
Gentry’s Recryption

(Programmable) Bootstrapping

Functional Circuits

Numerical Experiments

PROBLEM TO SOLVE



- Only known way to bootstrap is Gentry’s recryption technique

PROBLEM TO SOLVE

with $c \leftarrow \mathrm{Enc}_{sk_1}(m)$

$\mathrm{Enc}_{sk_2}(sk_1) \longrightarrow$ $\mathrm{Dec}(\cdot, c)$ $\longrightarrow \mathrm{Enc}_{sk_2}(m)$

(bootstrapping key)

- Only known way to bootstrap is Gentry's recryption technique

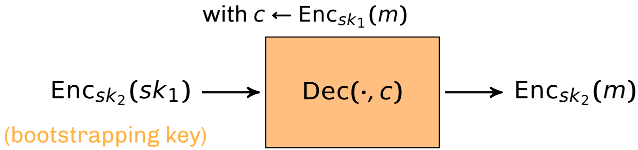
TLWE ENCRYPTION

1. $\boldsymbol{a} \overset{\$}{\leftarrow} \mathbb{T}_q^n$
2. $\mu^* := \mu + e \in \mathbb{T}_q$
3. $b \leftarrow \mu^* + \langle \boldsymbol{s}, \boldsymbol{a} \rangle$

TLWE DECRYPTION

1. $\mu^* \leftarrow b - \langle \boldsymbol{s}, \boldsymbol{a} \rangle$
2. round μ^*

PROBLEM TO SOLVE



- Only known way to bootstrap is Gentry's recryption technique
- **How to round over encrypted data?**

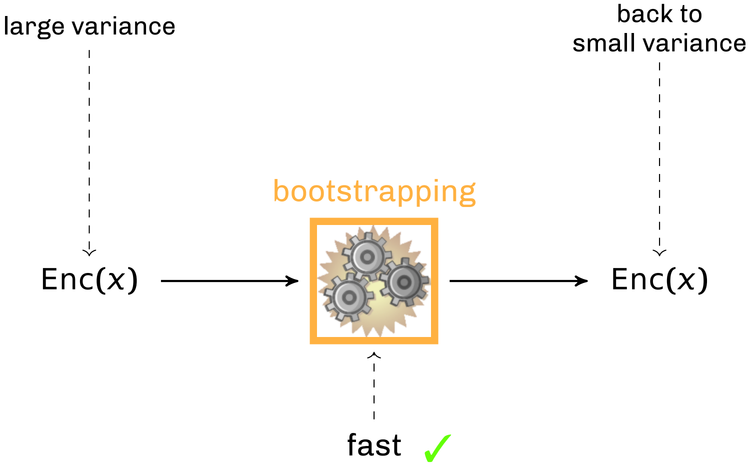
TLWE ENCRYPTION

1. $\boldsymbol{a} \xleftarrow{\$} \mathbb{T}_q^n$
2. $\mu^* := \mu + e \in \mathbb{T}_q$
3. $b \leftarrow \mu^* + \langle \boldsymbol{s}, \boldsymbol{a} \rangle$

TLWE DECRYPTION

1. $\mu^* \leftarrow b - \langle \boldsymbol{s}, \boldsymbol{a} \rangle$
2. round μ^*

BOOTSTRAPPING



PROGRAMMABLE BOOTSTRAPPING

large variance

$\text{Enc}(x)$

programmable
bootstrapping

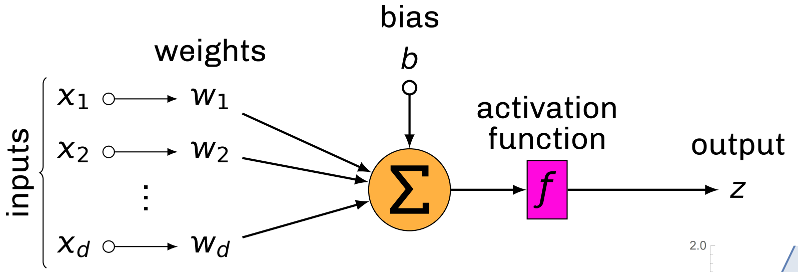
$f(\cdot)$

back to
small variance

$\text{Enc}(f(x))$

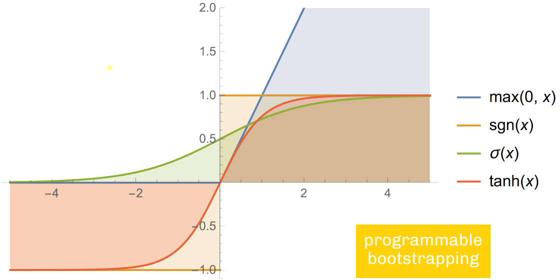
- function f is programmed during the bootstrapping
- essentially no extra cost

ARTIFICIAL NEURON



$$y = \sum_{i=1}^{d} w_i x_i + b$$

$$z = f(y)$$



programmable bootstrapping

PERFORMANCE

Programmable bootstrapping in milliseconds*

	$N = 1024$		$N = 2048$		$N = 4096$	
# bits	32	64	32	64	32	64
$n = 630$	15.49	18.08	33.28	39.54	73.22	84.01
$n = 800$	19.23	22.98	42.33	50.53	93.12	107.3
$n = 1024$	24.54	29.16	54.14	64.18	117.9	135.2

*2.6 GHz 6-Core Intel® Core™ i7 processor

OUTLINE

Fully Homomorphic Encryption

Gentry's Recryption

(Programmable) Bootstrapping

Functional Circuits

Numerical Experiments

PROGRAMMABLE BOOTSTRAPPING IS POWERFUL

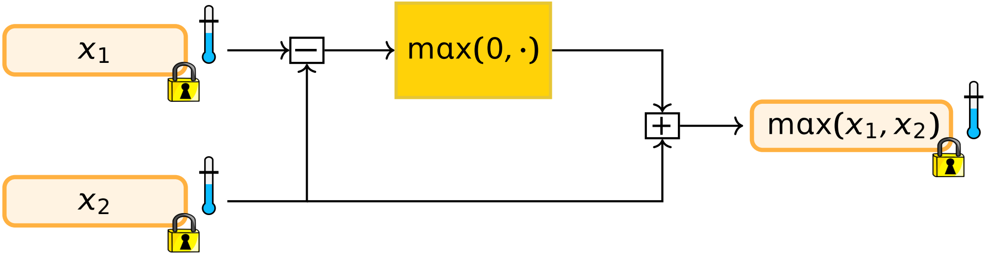
COMPUTING A MAXIMUM: $\max(x_1, x_2, \ldots, x_n)$

- $\max(x_1, x_2) = \max(0, x_1 - x_2) + x_2$
- $\max(x_1, x_2, x_3) = \max\big(\max(x_1, x_2), x_3\big)$

PROGRAMMABLE BOOTSTRAPPING IS POWERFUL

COMPUTING A MAXIMUM: $\max(x_1, x_2, \ldots, x_n)$

- $\max(x_1, x_2) = \max(0, x_1 - x_2) + x_2$
- $\max(x_1, x_2, x_3) = \max\big(\max(x_1, x_2), x_3\big)$



ALL YOU NEED: ADDITIONS AND PBS's

Kolmogorov Superposition Theorem (KST) 1957

$$f(x_1,...,x_n) = \sum_i g_i\left(\sum_j f_{i,j}(x_j)\right)$$

univariate

Ridge decomposition or approximation

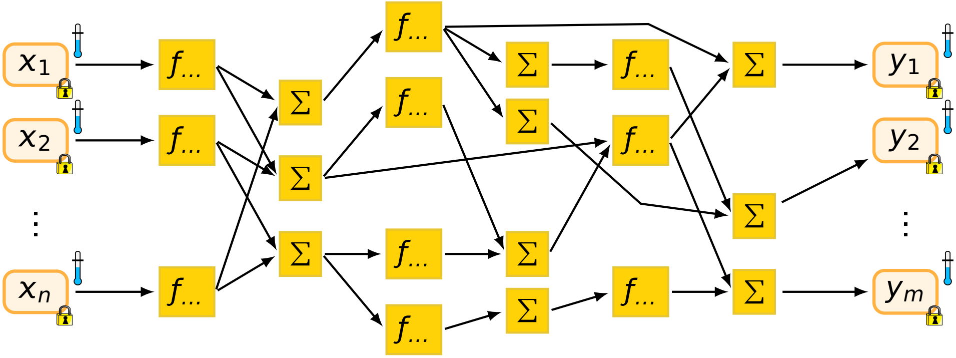
$$f(x_1,...,x_n) \approx \sum_i g_i\left(\sum_j a_{i,j}\, x_j\right)$$

univariate

$a_{i,j} \in \mathbb{Z}$

A NEW COMPUTATIONAL PARADIGM

Circuit of univariate functions



Graph mixing univariate functions and linear combinations

OUTLINE

Fully Homomorphic Encryption

Gentry's Recryption

(Programmable) Bootstrapping

Functional Circuits

Numerical Experiments

Let's be Concrete

https://github.com/zama-ai

NUMERICAL EXPERIMENTS

- MNIST dataset

- Three neural networks:
  - NN-x where x is the number of layers with $x \in \{20, 50, 100\}$
  - networks all include dense and convolution layers with activation functions
  - every hidden layer possesses at least 92 active neurons

NUMERICAL EXPERIMENTS

	In the clear	Encrypted
NN-20	$0.17\,ms$	$115.52\,s$
NN-50	$0.20\,ms$	$233.55\,s$
NN-100	$0.33\,ms$	$481.61\,s$

*2.6 GHz 6-Core Intel® Core™ i7 processor

SUMMARY

- Programmable bootstrapping is a powerful tool
  - enables evaluation of any function
  - runs relatively fast
  - accommodates every use-case

- Try out the Concrete library!