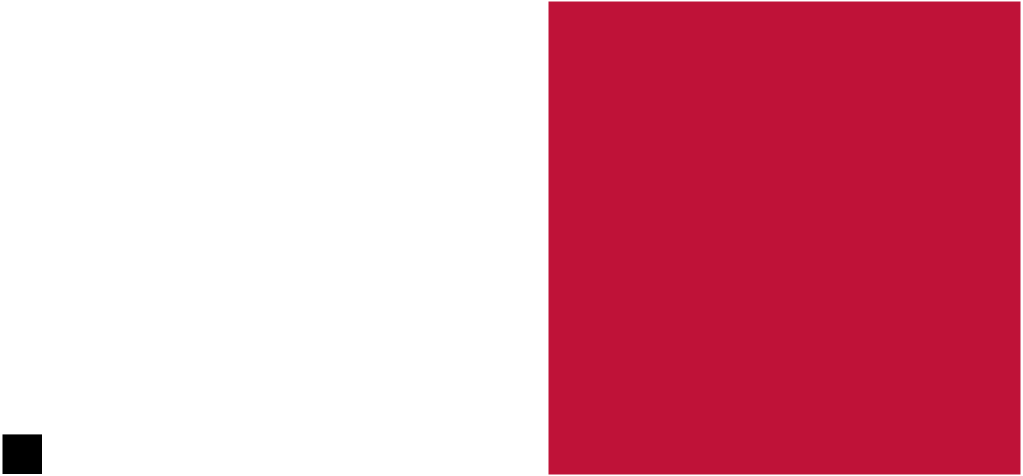
TOWARDS CASE-OPTIMIZED HYBRID HOMOMORPHIC ENCRYPTION

Featuring the Elisabeth Stream Cipher

Orel Cosseron Clément Hoffmann Pierrick Méaux François-Xavier Standaert

INRIA · ENS de Lyon · Télécom Paris · UCLouvain · Luxembourg University

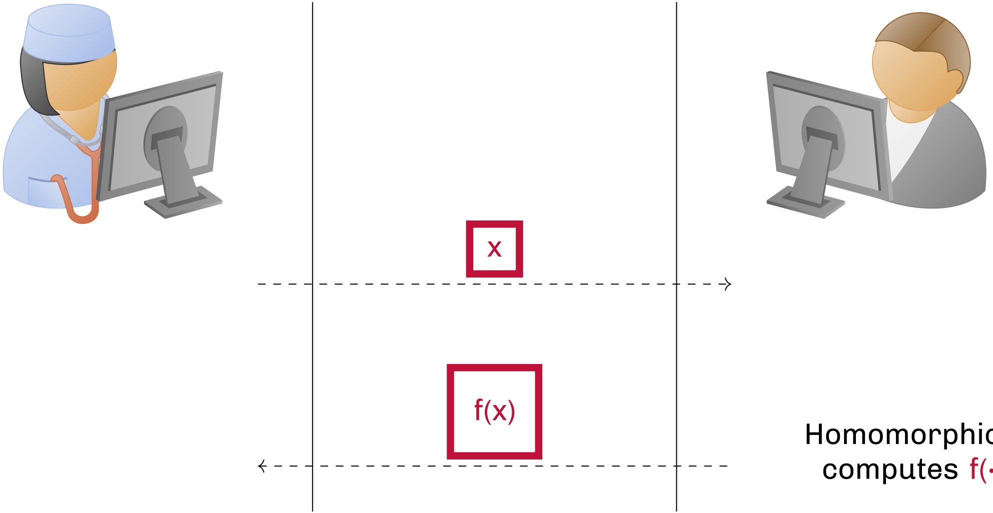
EXPANSION RATIO



Plaintext

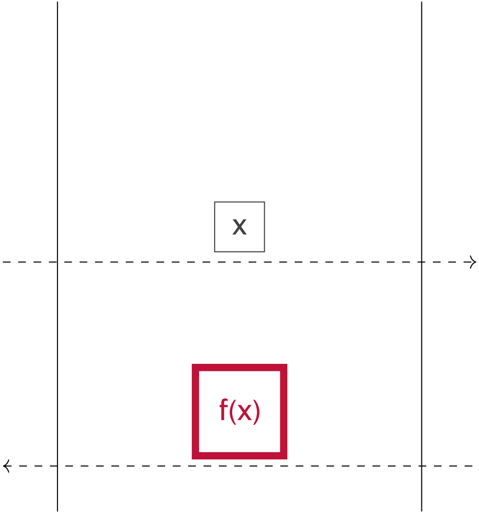
Ciphertext

HYBRID HOMOMORPHIC ENCRYPTION 101



Homomorphically computes $f(\cdot)$

HYBRID HOMOMORPHIC ENCRYPTION 101

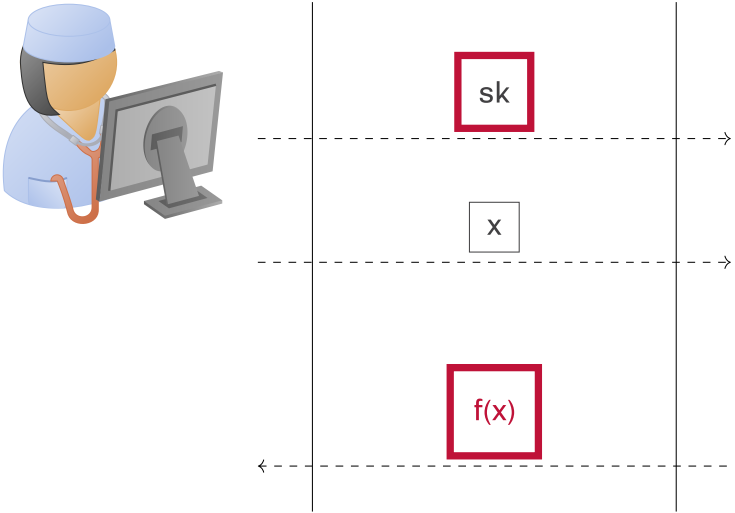


Homomorphically computes Dec(sk,·)

to get $\boxed{x}$

Homomorphically computes f(·)

HYBRID HOMOMORPHIC ENCRYPTION 101



$\boxed{sk}$

$\boxed{x}$

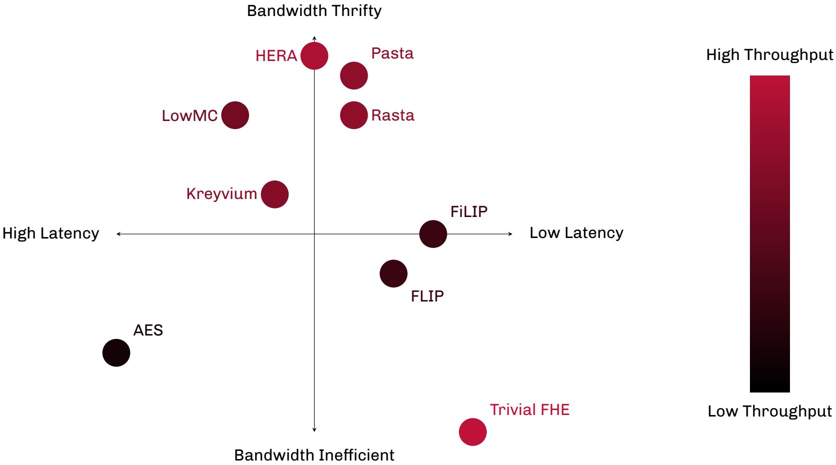
$\boxed{f(x)}$

Homomorphically computes Dec(sk,·)

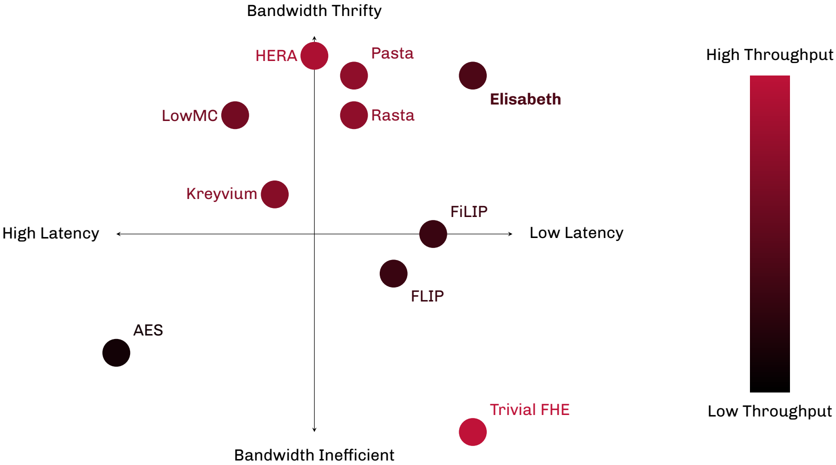
to get $\boxed{x}$

Homomorphically computes f(·)

STATE OF THE ART



STATE OF THE ART



CASE OPTIMIZATION

Current Status:

- Design of symmetric scheme for **HHE**.
- Optimized to be used **as a stand-alone scheme**.

CASE OPTIMIZATION

Current Status:

- Design of symmetric scheme for **HHE**.
- Optimized to be used **as a stand-alone scheme**.

This Paper: Case Optimization

- Design of symmetric scheme for **a concrete case study**.
- Optimized to be used **in combination with that case**.

CASE OPTIMIZATION

Usecase:

CASE OPTIMIZATION

Usecase: **Machine Learning**

CASE OPTIMIZATION

Usecase: **Machine Learning**

- As many bits of message as possible

CASE OPTIMIZATION

Usecase: **Machine Learning**

- As many bits of message as possible
- Fast evaluation

CASE OPTIMIZATION

Usecase: **Machine Learning**

- As many bits of message as possible
- Fast evaluation

Constraints:

CASE OPTIMIZATION

Usecase: **Machine Learning**

- As many bits of message as possible
- Fast evaluation

Constraints: **TFHE**

CASE OPTIMIZATION

Usecase: **Machine Learning**

- As many bits of message as possible
- Fast evaluation

Constraints: **TFHE**

- No Packing

CASE OPTIMIZATION

Usecase: **Machine Learning**

- As many bits of message as possible
- Fast evaluation

Constraints: **TFHE**

- No Packing
- Modular Additions

CASE OPTIMIZATION

Usecase: **Machine Learning**

- As many bits of message as possible
- Fast evaluation

Constraints: **TFHE**

- No Packing
- Modular Additions
- Negacyclic Look-Up Tables

CASE OPTIMIZATION

Usecase: **Machine Learning**

- As many bits of message as possible → **4-bit messages**
- Fast evaluation

Constraints: **TFHE**

- No Packing
- Modular Additions
- Negacyclic Look-Up Tables

CASE OPTIMIZATION

Usecase: **Machine Learning**

- As many bits of message as possible → **4-bit messages**
- Fast evaluation → **Multithreading**

Constraints: **TFHE**

- No Packing
- Modular Additions
- Negacyclic Look-Up Tables

CASE OPTIMIZATION

Usecase: **Machine Learning**

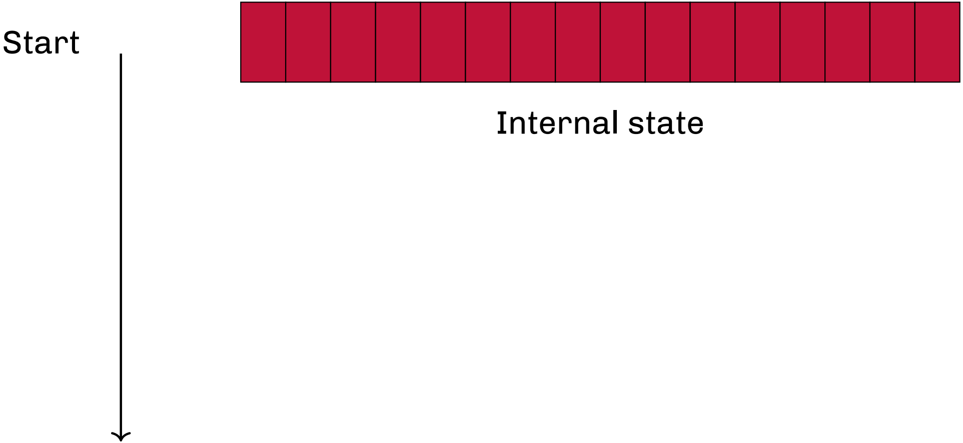
- As many bits of message as possible → **4-bit messages**
- Fast evaluation → **Multithreading**

Constraints: **TFHE**

- No Packing
- Modular Additions
- Negacyclic Look-Up Tables

Toolbox

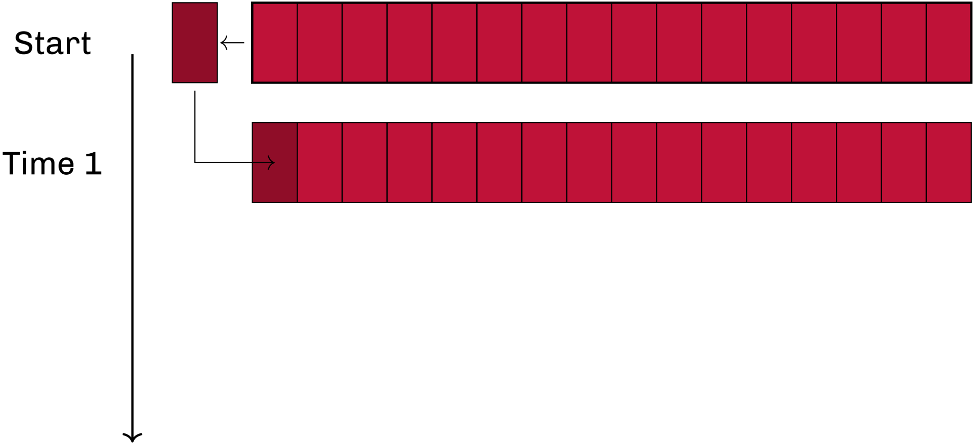
THE FILTER PERMUTATOR PARADIGM



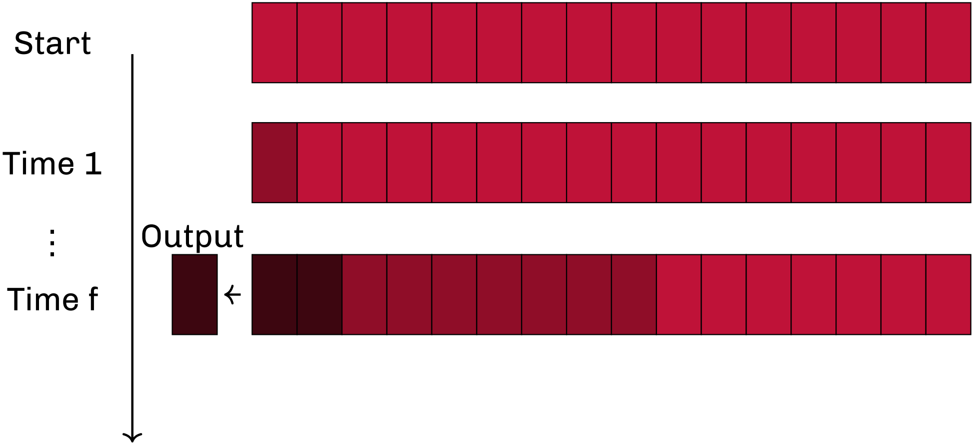
Start

Internal state

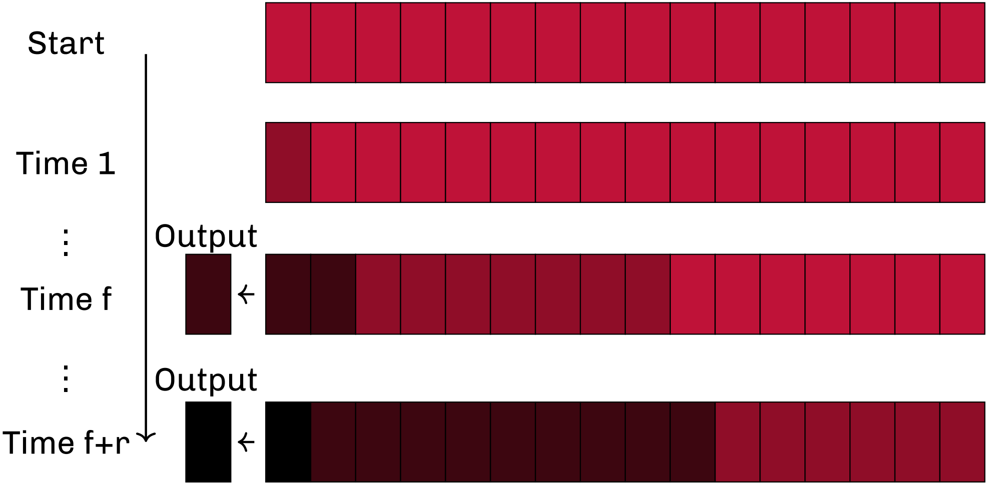
THE FILTER PERMUTATOR PARADIGM



THE FILTER PERMUTATOR PARADIGM



THE FILTER PERMUTATOR PARADIGM



Start

Time 1

⋮

Output

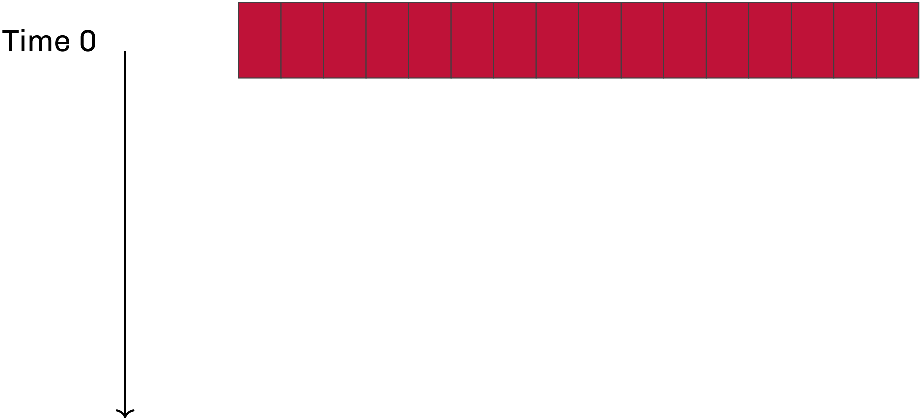
Time f

⋮

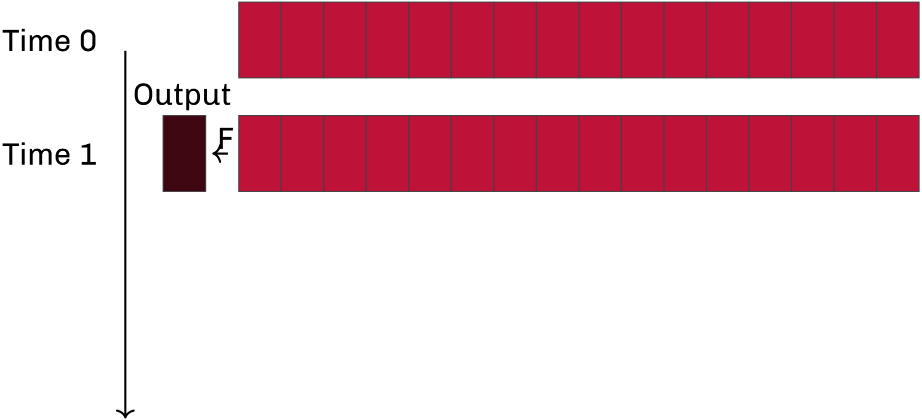
Output

Time f+r

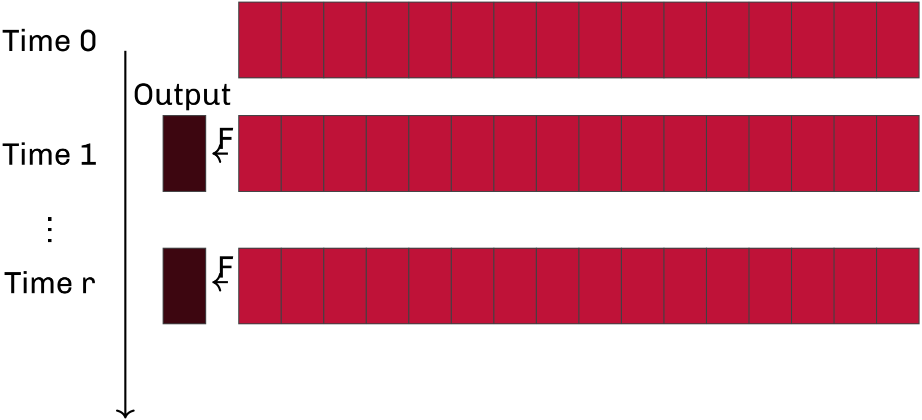
THE FILTER PERMUTATOR PARADIGM



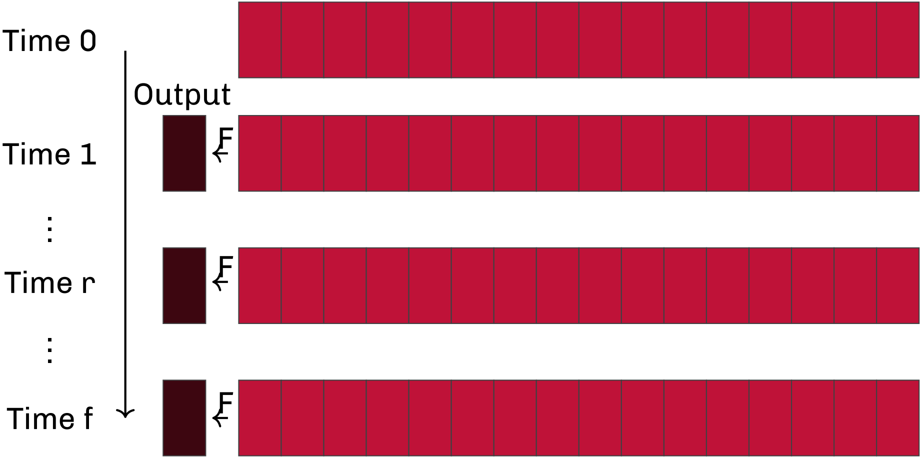
THE FILTER PERMUTATOR PARADIGM



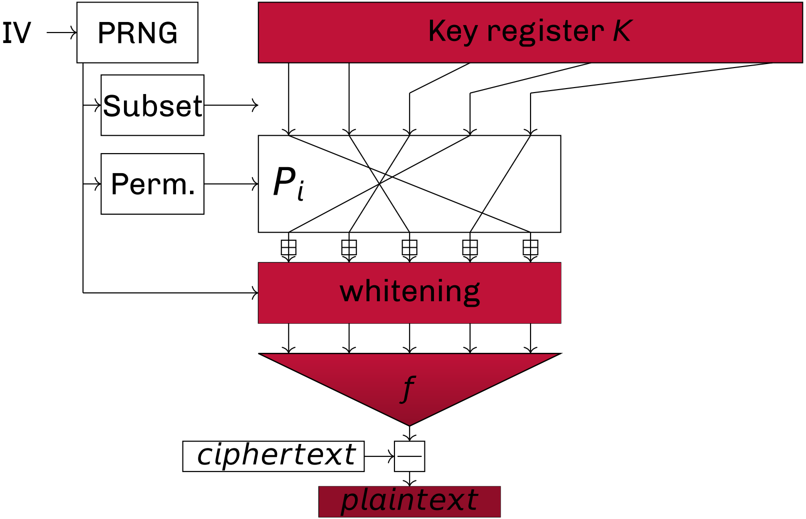
THE FILTER PERMUTATOR PARADIGM



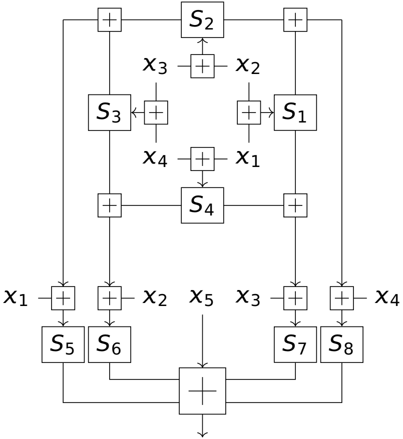
THE FILTER PERMUTATOR PARADIGM



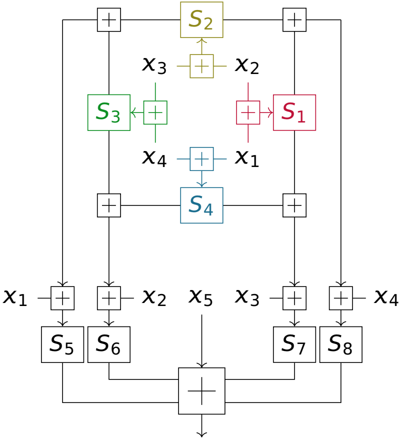
THE GROUP FILTER PERMUTATOR PARADIGM



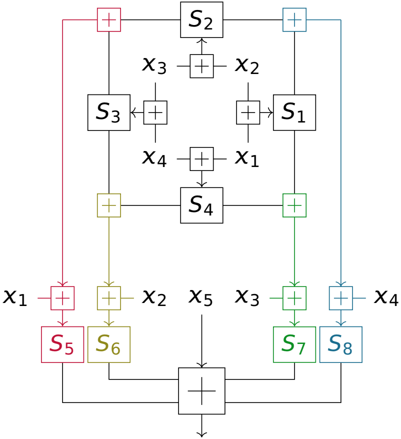
ELISABETH'S FILTER



ELISABETH'S FILTER



ELISABETH'S FILTER



NEURAL NETWORK

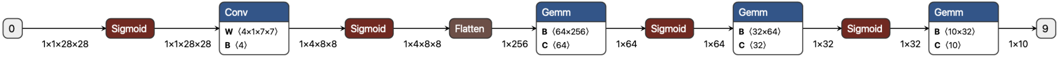


Figure 1: Our Neural Network

FASHION-MNIST

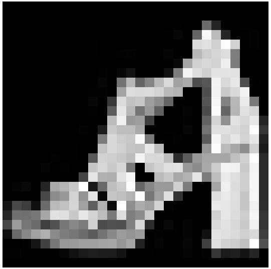
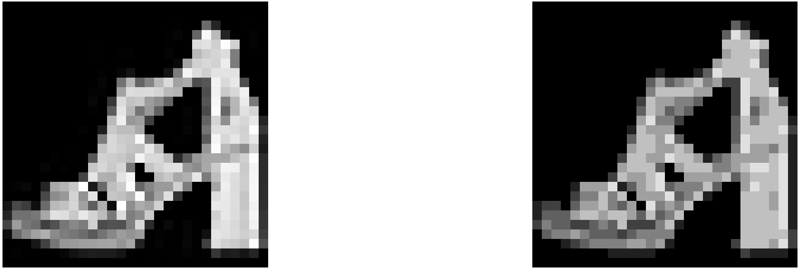


Figure 2: Example of a 784-pixels Fashion-MNIST picture

FASHION-MNIST



(a) Original data (8-bits shades)

(b) Quantized data (3-bits shades)

Figure 2: Example of a 784-pixels Fashion-MNIST picture

PERFORMANCES*

Mode		Latency (ms)	Throughput (ms/b)	Key size (kB)
Stand-alone	Single KeySwitch	104	26	8
	Two KeySwitches	91	22.75	20
Usecase	Single KeySwitch	537.6	134.4	8

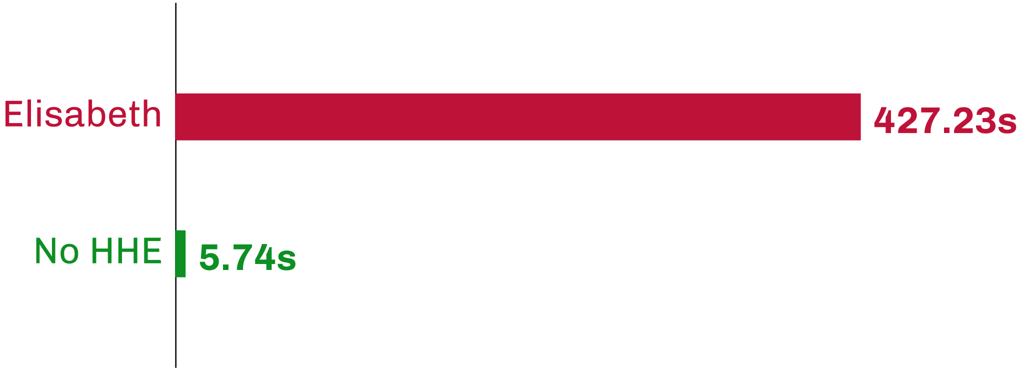
* For 128 bits of security on a computer equipped with an AMD Ryzen Threadripper 3990X 64-Core Processor

PERFORMANCES*

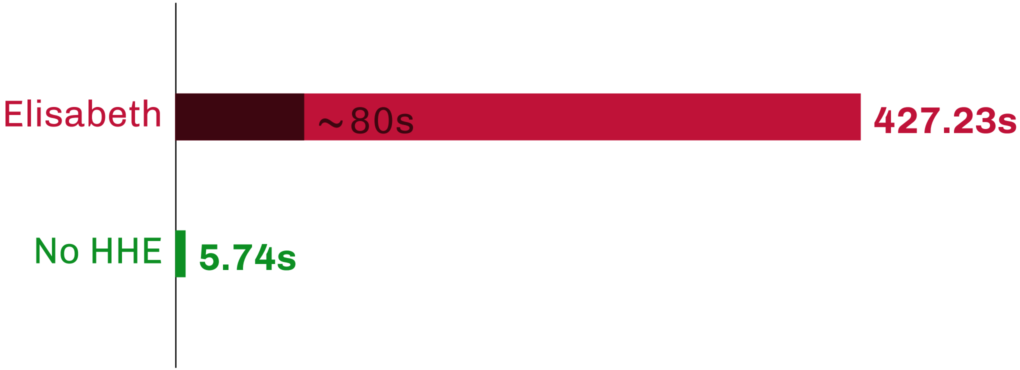
Mode		Latency (ms)	Throughput (ms/b)	Key size (kB)
Stand-alone	Single KeySwitch	**104**	26	8
	Two KeySwitches	**91**	22.75	20
Usecase	Single KeySwitch	**537.6**	134.4	8

* For 128 bits of security on a computer equipped with an AMD Ryzen Threadripper 3990X 64-Core Processor

Inference Time



Inference Time



PERFORMANCES

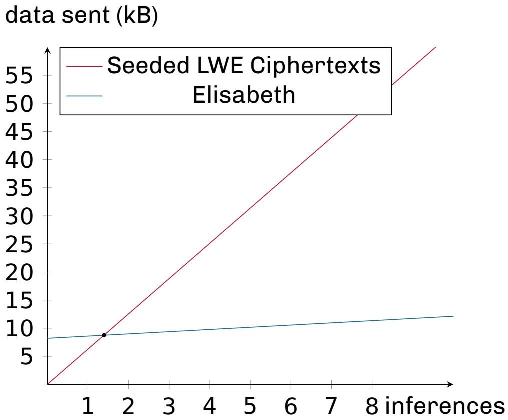


Figure 3: Bandwith consumption

PERFORMANCES

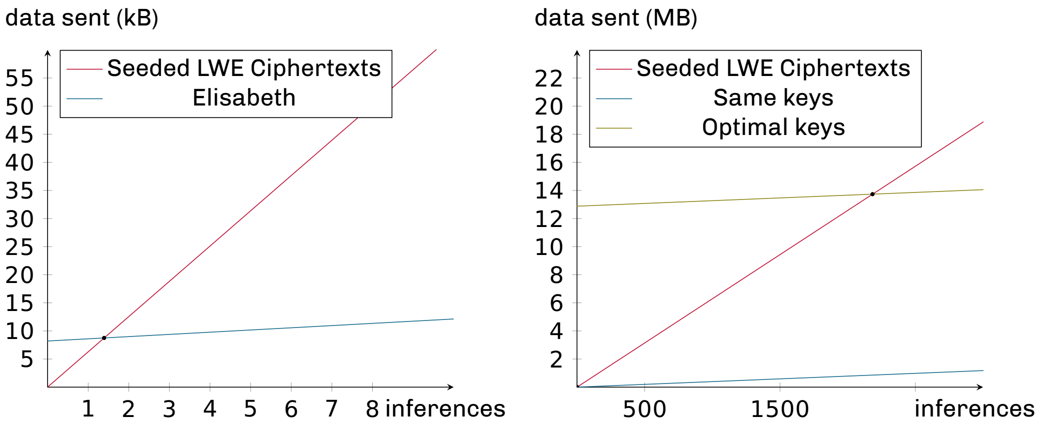


Figure 3: Bandwith consumption

OPEN PROBLEMS

- Bigger messages
- Better ML management in clear
- More relevant usecases

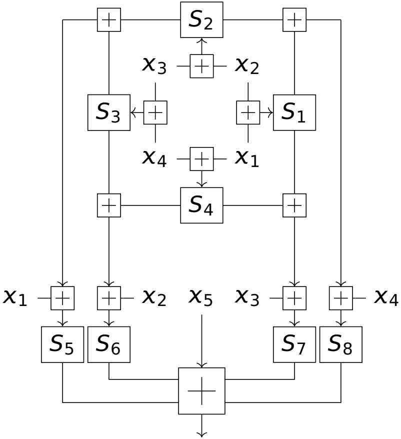
THANK YOU FOR YOUR ATTENTION

Q & A

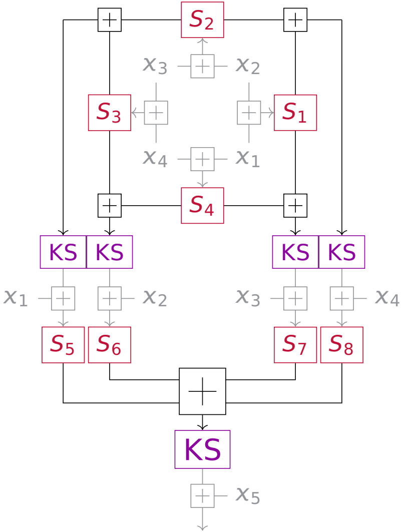
STAND-ALONE COMPARISONS

Cipher	Homomorphic library	Time per ciphertext (s)	Time per bit (ms)
LowMC	TFHE (C)	4283.678	16733
Kreyvium	TFHE (C)	208.255	208255
RASTA 6	TFHE (C)	2424.503	6907
FiLIP 144	Concrete	0.134	134
FiLIP 1216	Concrete	0.586	586
FiLIP 1280	Concrete	0.627	627
DASTA 6	TFHE (C)	2387.674	6802
Elisabeth-4 (two KS)	Concrete	0.091	22.75
Elisabeth-4 (single KS)	Concrete	0.104	26
LowMC	HELib	853.302	3333.21
Kreyvium	HELib	8.222	8222
RASTA 6	HELib	163.131	464.76
DASTA 6	HELib	156.935	447.11
MASTA 5	HELib	22.096	20.31
PASTA 4	HELib	9.827	18.06
HERA	CKKS	14.747	0.01

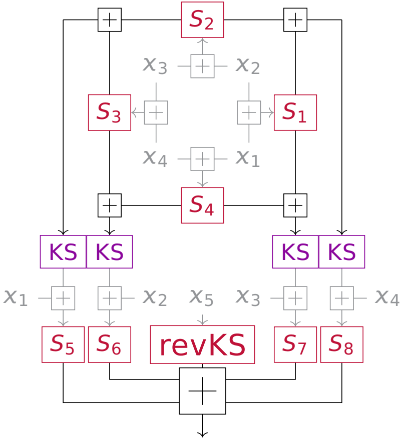
ELISABETH’S FILTER (HOMOMORPHICALLY)



ELISABETH’S FILTER (HOMOMORPHICALLY)



ELISABETH'S FILTER (HOMOMORPHICALLY)



S_2
x_3
x_2
S_3
S_1
x_4
x_1
S_4
KS
KS
KS
KS
x_1
x_2
x_5
x_3
x_4
S_5
S_6
revKS
S_7
S_8