Transistor: a TFHE-friendly Stream Cipher

Christina Boura
IRIF, Université Paris Cité

(joint work with Jules Baudrin, Sonia Belaïd, Nicolas Bon, Anne Canteaut, Gaëtan Leurent, Pascal Paillier, Léo Perrin, Matthieu Rivain, Yann Rotella, and Samuel Tap)

Overview

Université Paris Cité

1. Motivation
2. The TFHE Scheme
3. Our Design
4. Security Analysis

Plan

Université Paris Cité

Motivation

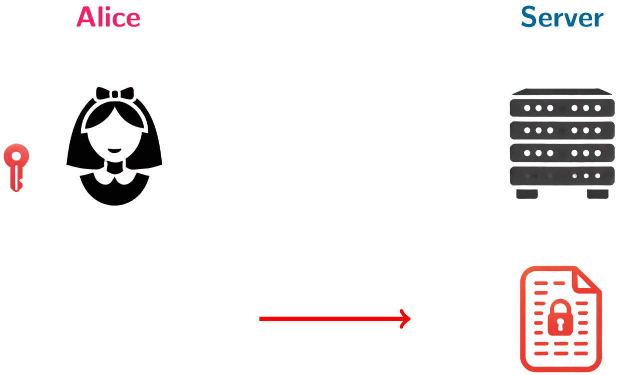
Alice wants to perform operations on her data but needs to **outsource** the processing.

Motivation

Alice   Server

Alice **encrypts** her data to protect its confidentiality.

Motivation



The server receives the encrypted data.

Motivation

The server needs to **decrypt** to process the data, which breaks confidentiality!

Fully Homomorphic Encryption (FHE)

FHE allows arbitrary computations to be performed directly on encrypted data **without needing to decrypt it**:

$$\text{Decrypt}(\text{Evaluate}(f, \text{Encrypt}(m))) = f(m)$$

Fully Homomorphic Encryption (FHE)

FHE allows arbitrary computations to be performed directly on encrypted data **without needing to decrypt it**:

$$\mathsf{Decrypt}(\mathsf{Evaluate}(f, \mathsf{Encrypt}(m))) = f(m)$$

Alice can now homomorphically encrypt her data and send it to the cloud.

Fully Homomorphic Encryption (FHE)

Université Paris Cité

FHE allows arbitrary computations to be performed directly on encrypted data **without needing to decrypt it**:

$$\mathsf{Decrypt}(\mathsf{Evaluate}(f, \mathsf{Encrypt}(m))) = f(m)$$

Alice can now homomorphically encrypt her data and send it to the cloud.

But:

- FHE operations are computationally intensive.
- Homomorphic ciphertexts are significantly larger than plaintexts, leading to increased communication costs.

Transciphering[1]

Alice wants to learn the application of f on her data m, by outsourcing the computation of $f(m)$ to the server.

Alice **Server**

[1] M. Naehrig, K. Lauter, V. Vaikuntanathan. *Can homomorphic encryption be practical?* ACM 2011.

Transciphering[1]

Alice wants to learn the application of f on her data m, by outsourcing the computation of $f(m)$ to the server.

Alice **Server**

$$m \longrightarrow \boxed{E_K^S}$$

[1] M. Naehrig, K. Lauter, V. Vaikuntanathan. *Can homomorphic encryption be practical?* ACM 2011.

Transciphering[1]

Alice wants to learn the application of f on her data m, by outsourcing the computation of $f(m)$ to the server.

Alice **Server**

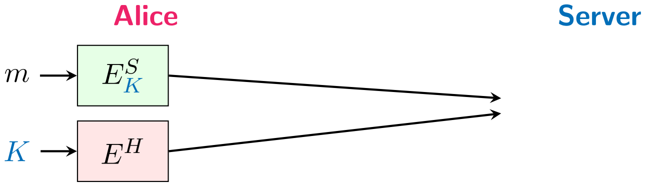
$$m \longrightarrow \boxed{E^S_K}$$

$$K \longrightarrow \boxed{E^H}$$

[1] M. Naehrig, K. Lauter, V. Vaikuntanathan. *Can homomorphic encryption be practical?* ACM 2011.

Transciphering[1]

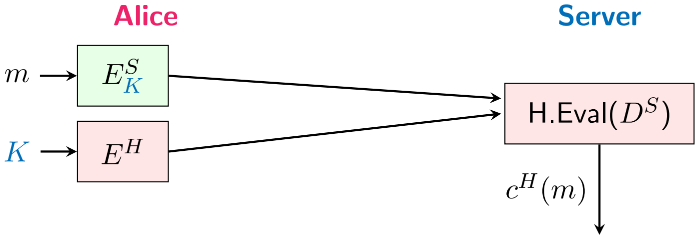
Alice wants to learn the application of f on her data m, by outsourcing the computation of $f(m)$ to the server.



[1] M. Naehrig, K. Lauter, V. Vaikuntanathan. *Can homomorphic encryption be practical?* ACM 2011.

Transciphering[1]

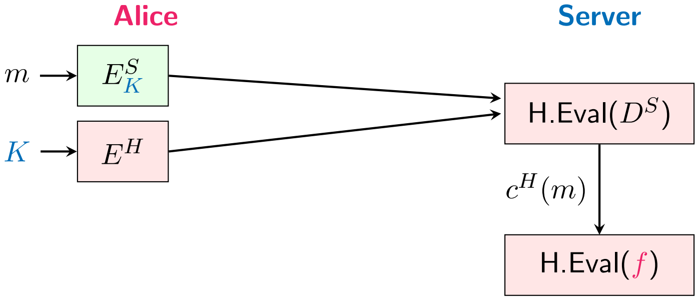
Alice wants to learn the application of f on her data m, by outsourcing the computation of $f(m)$ to the server.



[1] M. Naehrig, K. Lauter, V. Vaikuntanathan. *Can homomorphic encryption be practical?* ACM 2011.

Transciphering[1]

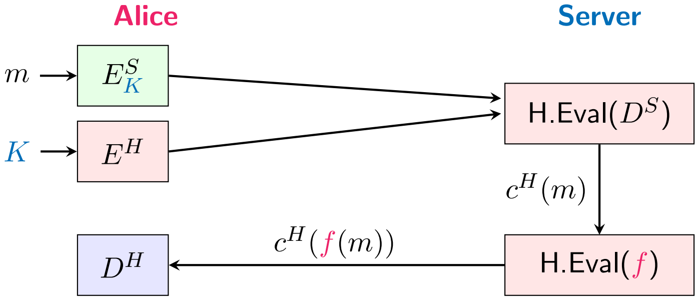
Alice wants to learn the application of f on her data m, by outsourcing the computation of $f(m)$ to the server.



[1] M. Naehrig, K. Lauter, V. Vaikuntanathan. *Can homomorphic encryption be practical?* ACM 2011.

Transciphering[1]

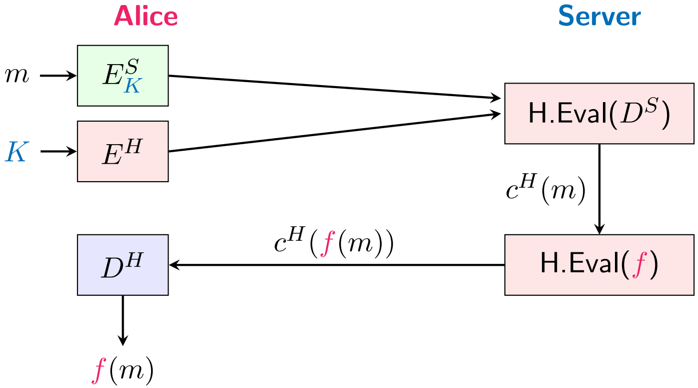
Alice wants to learn the application of f on her data m, by outsourcing the computation of $f(m)$ to the server.



[1] M. Naehrig, K. Lauter, V. Vaikuntanathan. *Can homomorphic encryption be practical?* ACM 2011.

Transciphering[1]

Alice wants to learn the application of f on her data m, by outsourcing the computation of $f(m)$ to the server.



[1] M. Naehrig, K. Lauter, V. Vaikuntanathan. *Can homomorphic encryption be practical?* ACM 2011.

Plan

TFHE (2016)

TFHE: Fast Fully Homomorphic Encryption over the Torus

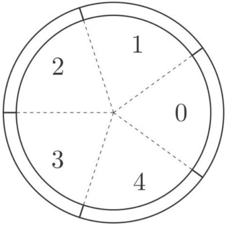
Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, Malika Izabachène

- Based on the Learning With Errors (LWE) problem.
- Very fast for the FHE standards.
- Programmable bootstrapping: Evaluation of encrypted look-up tables (LUT) while resetting the noise level.
- **But**: Operations should be limited on small plaintexts (typically less than 6 bits).

TFHE: Description of the Scheme

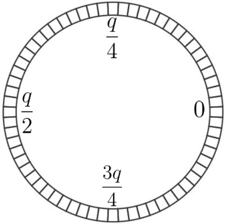
Discretized torus $\mathbb{T}_p = \{\frac{a}{p} \mid a \in \mathbb{Z}_p\}$.

Plaintext space $\mathbb{T}_p$



The size of $p \in \mathbb{N}$ is only a few bits.

Ciphertext space $\mathbb{T}_q$

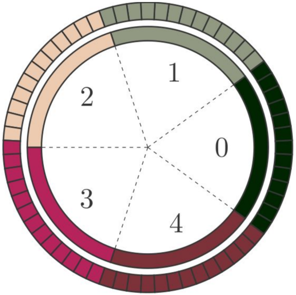


$q = 2^{32}$ or $q = 2^{64}$

Acknowledgment: Thanks to Nicolas Bon for the figures of TFHE.

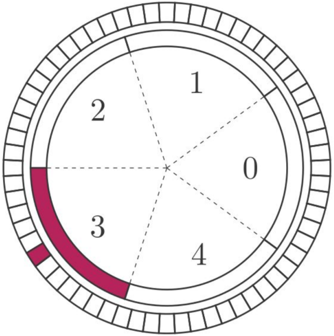
TFHE: Description of the Scheme

Natural embedding of $\mathbb{T}_p$ into $\mathbb{T}_q$: $m \mapsto \left\lfloor \frac{mq}{p} \right\rceil$.



Encryption Process

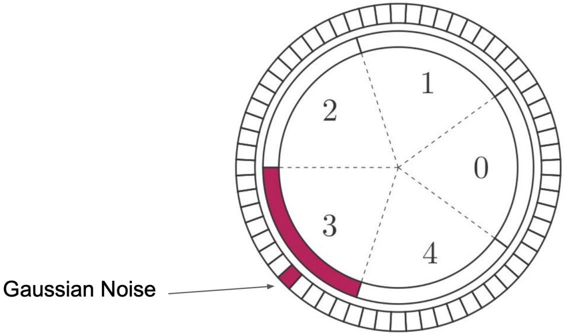
Encoding: $m \in \mathbb{T}_p \mapsto \widetilde{m} \in \mathbb{T}_q$



Encryption Process

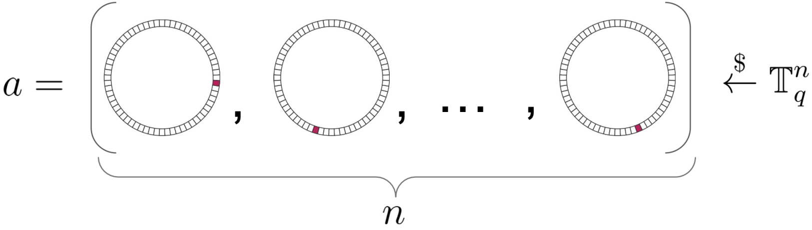
Encoding: $m \in \mathbb{T}_p \mapsto \widetilde{m} \in \mathbb{T}_q$

Adding noise: $\widetilde{m} + e$, with $e \xleftarrow{\$} \chi_\sigma$



Encryption Process

Mask $a = (a_1, \ldots, a_n) \in \mathbb{Z}_q^n$.

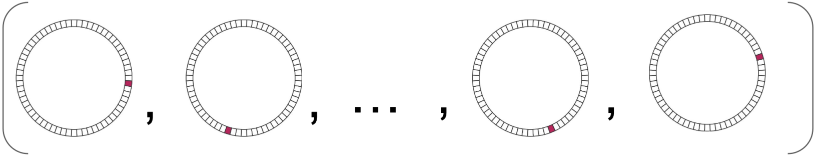


Secret key $\mathsf{sk} = (s_1, \ldots, s_n) \in \{0,1\}^n$.

Encryption Process

Ciphertext: $c = (a_1, \ldots, a_n, b)$

$c =$



where $b = \sum_{i=1}^{n} a_i \cdot s_i + \tilde{m} + e$.

Homomorphic Operations

Sum of ciphertexts

Let c_1 and c_2 be two ciphertexts encrypting m_1 and m_2 with noise levels σ_1^2 and σ_2^2, respectively. The **noise level of the ciphertext encrypting** $m_1 + m_2$ is $\sigma_1^2 + \sigma_2^2$.

Product with a cleartext

Let c be a ciphertext encrypting m with noise σ^2. Multiplying each coordinate of c by a constant $\lambda \in \mathbb{Z}$ produces a valid ciphertext c', which encrypts $m' = \lambda \cdot m$ with **noise** $\lambda^2 \cdot \sigma^2$.

Programmable Bootstrapping (PBS)

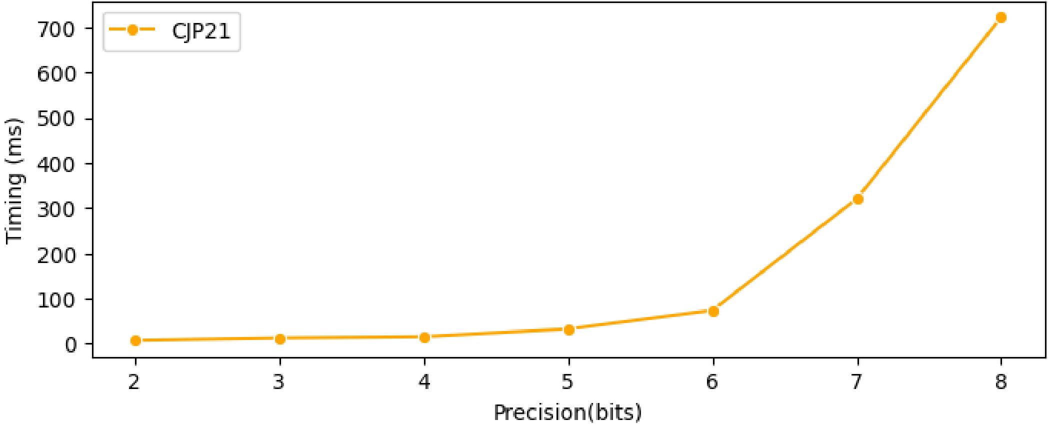
Université Paris Cité

Bootstrapping

Generic technique introduced by Gentry that allows the noise of a ciphertext to be reset to a nominal level.

- In **TFHE**, bootstrapping is implemented in a programmable manner: while the noise is being reset, any arbitrary function can be evaluated on the ciphertext.

Timing of a PBS



Transciphering: State of the Art

Université Paris Cité

- LowMC (2016)
- Kreyvium (2016)
- FLIP (2016)
- FiLIP (2020)
- ~~Elisabeth~~ (2022)
- Elisabeth-b, Gabriel and Margrethe (2023) (patches of Elisabeth)
- FRAST (2024)

- Elisabeth (and its successors) as well as FRAST were designed specifically for **TFHE**.
- Trivium and Kreyvium provide good performance within the **TFHE** transciphering framework.

Plan

Design Choices

Plaintext space: $\mathbb{Z}_p$ with $p = 17$

- p is **odd** (avoid dealing with negacyclicity)
- p is close to 2^4 (convenient for **encoding nibbles**)
- p is **prime**: $\mathbb{Z}_p = \mathbb{F}_p$ has a field structure

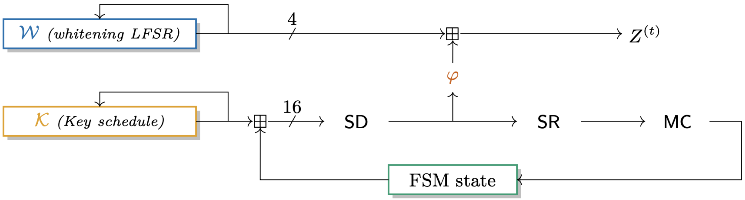
Non-linearity: S-box layer applying in parallel S: $\mathbb{Z}_{17} \to \mathbb{Z}_{17}$

- One PBS per S-box
- Minimize # PBS per element of the output stream

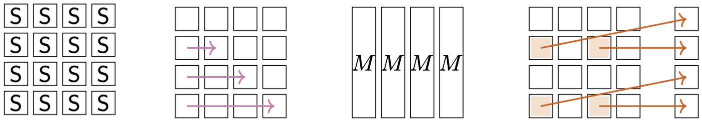
Transistor = Transciphering + Torus

- `Transistor` is a stream cipher that generates a keystream composed of elements from $\mathbb{F}_{17}$.
- It generates tuples of 4 digits at once.

A High Level View of `Transistor`



(a) General structure (rectangles correspond to registers).



(b) SD. (c) SR. (d) MC. (e) φ.

- $|\mathcal{K}| = 64,\ |\mathcal{W}| = 32.$

The S-box

Table representation

$$\mathsf{S} = [1, 12, 6, 11, 14, 3, 15, 5, 10, 9, 13, 16, 7, 8, 0, 2, 4]$$

Polynomial representation

$$\begin{aligned}\mathsf{S}(x) &= 1 + 4x^1 + 13x^2 + 7x^3 + 16x^4 + 15x^5 + 5x^7 + 5x^8 \\ &\quad + 11x^9 + 13x^{10} + 12x^{11} + 13x^{12} + 15x^{14} + x^{15}\ .\end{aligned}$$

Cryptographic properties

- Almost Perfect Nonlinear (APN) permutation:
  - $\mathsf{S}(x+a) = \mathsf{S}(x) + b$ has at most 2 solutions x for all $a \neq 0$ and all b.
- Maximum algebraic degree.

The Linear Layer (MC)

$$M = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & -1 & 1 & -2 \\ 1 & 1 & -2 & -1 \\ 1 & -2 & -1 & 1 \end{pmatrix}$$

Design criterion

M should be **MDS** and its ℓ_2-norm should be **as low as possible**.

The ℓ_2-norm of M is defined as:

$$\ell_2(M) = \max_{i=1,2,3,4} \sqrt{M(i,1)^2 + M(i,2)^2 + M(i,3)^2 + M(i,4)^2}.$$

The Silent LFSR

Homomorphic implementation of the LFSRs:

The naive approach: Maintain an encrypted state, and update it by computing a linear combination with the feedback coefficients.

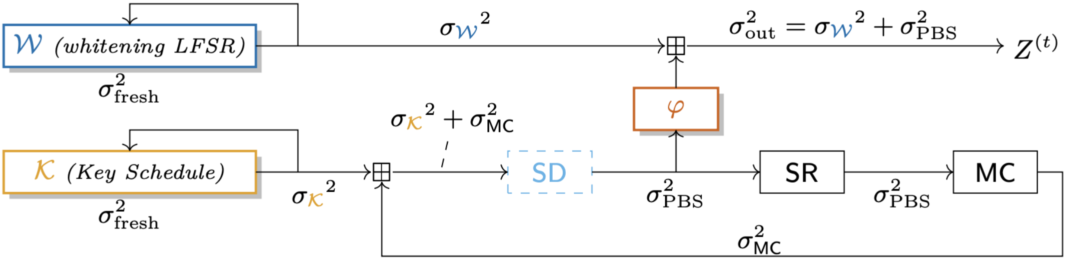
- Noise accumulates over time and needs periodic use of PBS operations to be refreshed.

The silent approach: Computing on the fly the coefficients of the linear combinations in clear, without updating an encrypted version of the internal state.

- The noise level remains stable over time.

Similar approach as in `FLIP` where a key state is queried **without being updated**.

Noise Evolution



Plan

Université Paris Cité

1. Motivation
2. The TFHE Scheme
3. Our Design
4. **Security Analysis**

Security Analysis

Security Claim

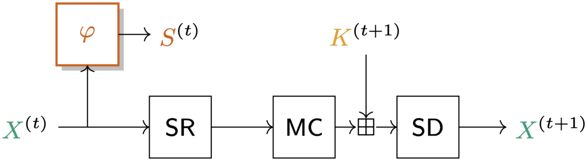
Transistor provides 128 bits of security, assuming no more than 2^{31} digits are generated with a single master key/IV pair.

We analyzed the security of the cipher against:

- Time-Memory-Data trade-off attacks
- Guess and determine attacks
- Fast correlation attacks
- Algebraic attacks

Guess and Determine Attack

The attacker links the FSM state $X^{(t)}$ to the filter output $S^{(t)}$ by guessing digits of $K^{(t)}$.



1. Observe $S^{(t)} = \varphi(X^{(t)}) = \mathsf{SD}\left(K^{(t)} + \left(\mathsf{MC} \circ \mathsf{SR}(X^{(t-1)})\right)\right)_{[4,6,12,13]}$.
2. Guess the 12 missing digits of $K^{(t)}$ to compute $X^{(t)}$.

Complexity: $p^{\frac{3}{4}|\mathcal{K}|+|\mathcal{W}|}$ ($p = 17$, $|\mathcal{K}| = 64$ and $|\mathcal{W}| = 32$).

(Fast) Correlation Attacks

Université Paris Cité

Objective

Recover information about the initial state from the knowledge of the keystream.

Question: What is the smallest length of output sequence $(S^{(t)})$ that can provide information on the key-LFSR?

Theorem

The output of

$$F^{(3)} : (X^{(t)}, K^{(t+1)}, K^{(t+2)}) \mapsto (S^{(t)}, S^{(t+1)}, S^{(t+2)})$$

is statistically independent from (i.e., not correlated to) the key sequence.

Four consecutive outputs

Question: What is the data complexity to recover the internal state from the knowledge of at least four consecutive outputs $S^{(t)}, S^{(t+1)}, S^{(t+2)}, S^{(t+3)}$?

Proposition (Xiao-Massey lemma over $\mathbb{F}_p$)

If the output of

$$F^{(n)} : (X^{(t)}, K^{(t+1)}, \ldots, K^{(t+n-1)}) \mapsto (S^{(t)}, S^{(t+1)}, \ldots, S^{(t+n-1)})$$

is correlated to its key-input, then there exists a biased linear relation between the key-inputs and the outputs of $F^{(n)}$.

Data complexity of fast correlation attacks

Université Paris Cité

The **data complexity** of the best correlation attack based on a linear approximation

$$\sum_{i=1}^{n-1} \alpha_i \cdot K^{(t+i)} + \sum_{i=0}^{n-1} \beta_i \cdot S^{(t+i)}, \forall t \geq 0$$

is the inverse of

$$\Delta = \frac{p}{64 \ln p} \left(\frac{\mathcal{L}(\mathsf{S})}{p} \right)^{2w_n}$$

where

- $\mathcal{L}(\mathsf{S})$: maximal modulus of the Fourier coefficients of S
- $w_n = \sum_{i=1}^{n-1} wt(\alpha_i)$: # active S-boxes in the linear trail.

With MILP-based search

$w_4 \geq 13$, $w_5 \geq 20$, $w_6 \geq 25$, and $w_n \geq 26$ for $n \geq 7$.

Benchmarks

Cipher	p_{err}	Setup	Latency	Throughput
`FRAST`	2^{-80}	25 s (8 threads)	6.2 s	20.66 bits/s
`Transistor`	2^{-128}	No	251 ms	65.10 bits/s

Benchmarks

Université Paris Cité

Cipher	p_{err}	Setup	Latency	Throughput
FRAST	2^{-80}	25 s (8 threads)	6.2 s	20.66 bits/s
Transistor	2^{-128}	No	251 ms	65.10 bits/s

Thanks for your attention!