

Minimize the Randomness in Rasta-Like Designs: How Far Can We Go?

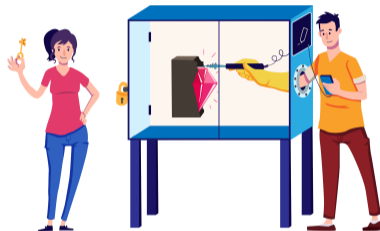
Application to PASTA

Lorenzo Grassi, Fukang Liu, Christian Rechberger,
Fabian Schmid, Roman Walch, and Qingju Wang

28.08.2024

Homomorphic Encryption

- HE scheme \mathcal{E} is set of functions:
 - Setup, Enc, Dec, KeyGen, Eval
- Outsourcing computation on encrypted data
- \mathcal{E} introduces noise and Ciphertext Expansion
 - Depending on $\mathcal{E}.Eval$
- Applications are faced with complex trade-offs:
 - Plaintext precision
 - Evaluation complexity
 - Security
 - Performance (Computation, Communication)



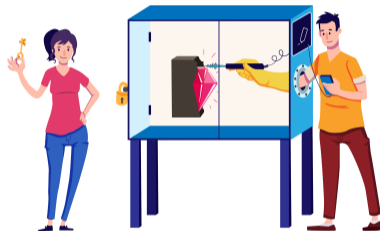
Homomorphic Encryption

- HE scheme \mathcal{E} is set of functions:
 - Setup, Enc, Dec, KeyGen, Eval
- Outsourcing computation on encrypted data
- \mathcal{E} introduces noise and Ciphertext Expansion
 - Depending on $\mathcal{E}.Eval$
- Applications are faced with complex trade-offs:
 - Plaintext precision
 - Evaluation complexity
 - Security
 - Performance (Computation, Communication)



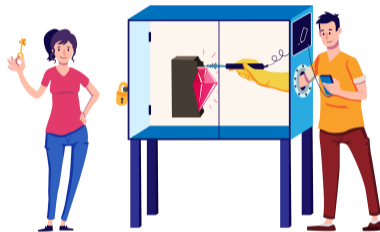
Homomorphic Encryption

- HE scheme \mathcal{E} is set of functions:
 - Setup, Enc, Dec, KeyGen, Eval
- Outsourcing computation on encrypted data
- \mathcal{E} introduces noise and Ciphertext Expansion
 - Depending on $\mathcal{E}.Eval$
- Applications are faced with complex trade-offs:
 - Plaintext precision
 - Evaluation complexity
 - Security
 - Performance (Computation, Communication)



Homomorphic Encryption

- HE scheme \mathcal{E} is set of functions:
 - Setup, Enc, Dec, KeyGen, Eval
- Outsourcing computation on encrypted data
- \mathcal{E} introduces noise and Ciphertext Expansion
 - Depending on $\mathcal{E}.Eval$
- Applications are faced with complex trade-offs:
 - Plaintext precision
 - Evaluation complexity
 - Security
 - Performance (Computation, Communication)



Homomorphic Encryption

- HE scheme \mathcal{E} is set of functions:
 - Setup, Enc, Dec, KeyGen, Eval
- Outsourcing computation on encrypted data
- \mathcal{E} introduces noise and Ciphertext Expansion
 - Depending on $\mathcal{E}.Eval$
- Applications are faced with complex trade-offs:
 - Plaintext precision
 - Evaluation complexity
 - Security
 - Performance (Computation, Communication)



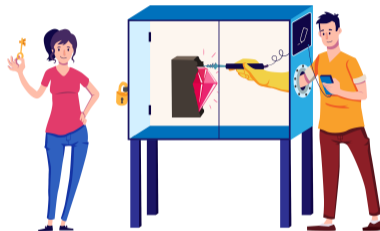
Homomorphic Encryption

- HE scheme \mathcal{E} is set of functions:
 - Setup, Enc, Dec, KeyGen, Eval
- Outsourcing computation on encrypted data
- \mathcal{E} introduces noise and Ciphertext Expansion
 - Depending on $\mathcal{E}.Eval$
- Applications are faced with complex trade-offs:
 - Plaintext precision
 - Evaluation complexity
 - Security
 - Performance (Computation, Communication)



Homomorphic Encryption

- HE scheme \mathcal{E} is set of functions:
 - Setup, Enc, Dec, KeyGen, Eval
- Outsourcing computation on encrypted data
- \mathcal{E} introduces noise and Ciphertext Expansion
 - Depending on $\mathcal{E}.Eval$
- Applications are faced with complex trade-offs:
 - Plaintext precision
 - Evaluation complexity
 - Security
 - Performance (Computation, Communication)



Parameters and Ciphertext Expansion

- Consider the polynomial ring $R = \mathbb{Z}[X]/(X^n + 1)$
 - Ciphertext and Plaintext spaces R_q and R_t , where $q \gg t$
- With perfect parallelization, the expansion factor is at least $2 \cdot \lceil \frac{q}{t} \rceil$
 - Can be $\geq 100x$ for complex use cases
 - Much worse without parallelization
- **Solution:** Encrypt data with a symmetric cipher, expand after transmission

Parameters and Ciphertext Expansion

- Consider the polynomial ring $R = \mathbb{Z}[X]/(X^n + 1)$
 - Ciphertext and Plaintext spaces R_q and R_t , where $q \gg t$
- With perfect parallelization, the expansion factor is at least $2 \cdot \lceil \frac{q}{t} \rceil$
 - Can be $\geq 100x$ for complex use cases
 - Much worse without parallelization
- **Solution:** Encrypt data with a symmetric cipher, expand after transmission

Parameters and Ciphertext Expansion

- Consider the polynomial ring $R = \mathbb{Z}[X]/(X^n + 1)$
 - Ciphertext and Plaintext spaces R_q and R_t , where $q \gg t$
- With perfect parallelization, the expansion factor is at least $2 \cdot \lceil \frac{q}{t} \rceil$
 - Can be $\geq 100x$ for complex use cases
 - Much worse without parallelization
- **Solution:** Encrypt data with a symmetric cipher, expand after transmission

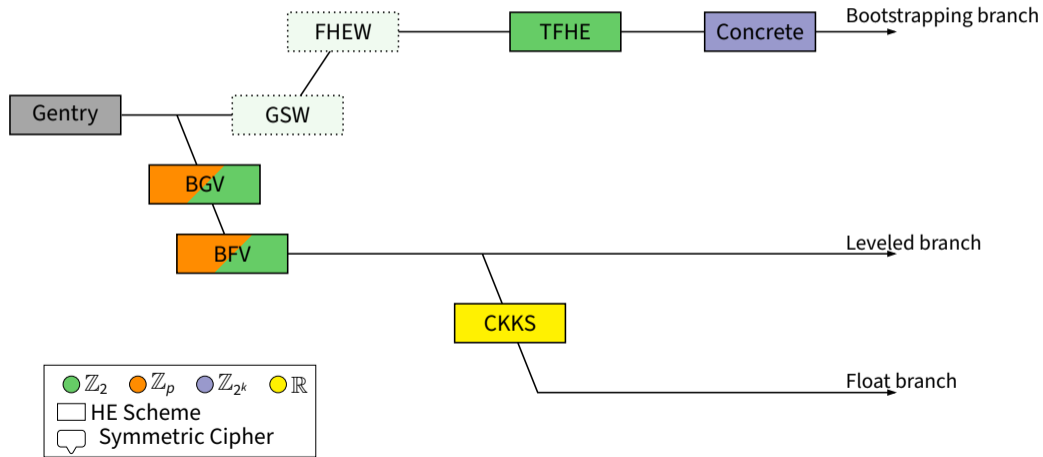
Parameters and Ciphertext Expansion

- Consider the polynomial ring $R = \mathbb{Z}[X]/(X^n + 1)$
 - Ciphertext and Plaintext spaces R_q and R_t , where $q \gg t$
- With perfect parallelization, the expansion factor is at least $2 \cdot \lceil \frac{q}{t} \rceil$
 - Can be $\geq 100x$ for complex use cases
 - Much worse without parallelization
- **Solution:** Encrypt data with a symmetric cipher, expand after transmission

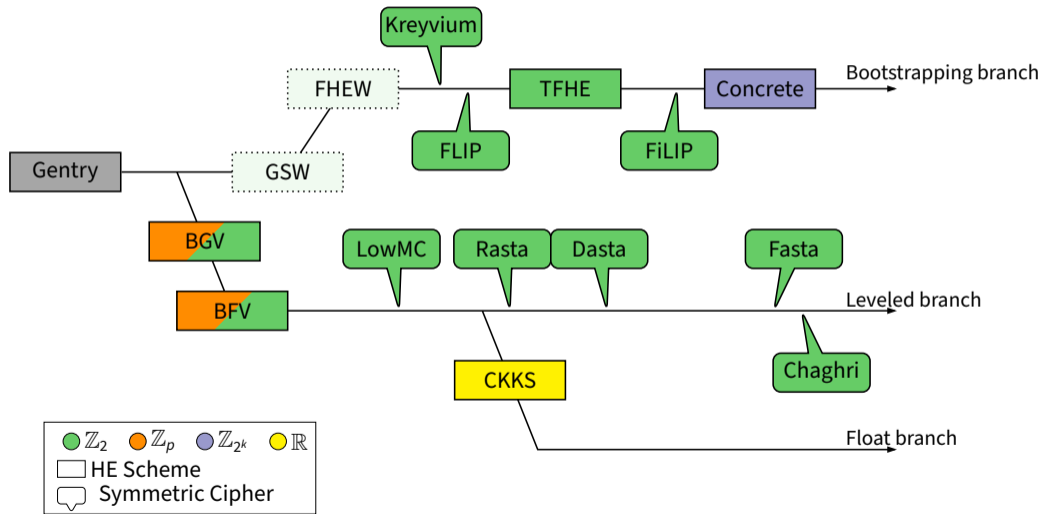
Ciphers for HHE



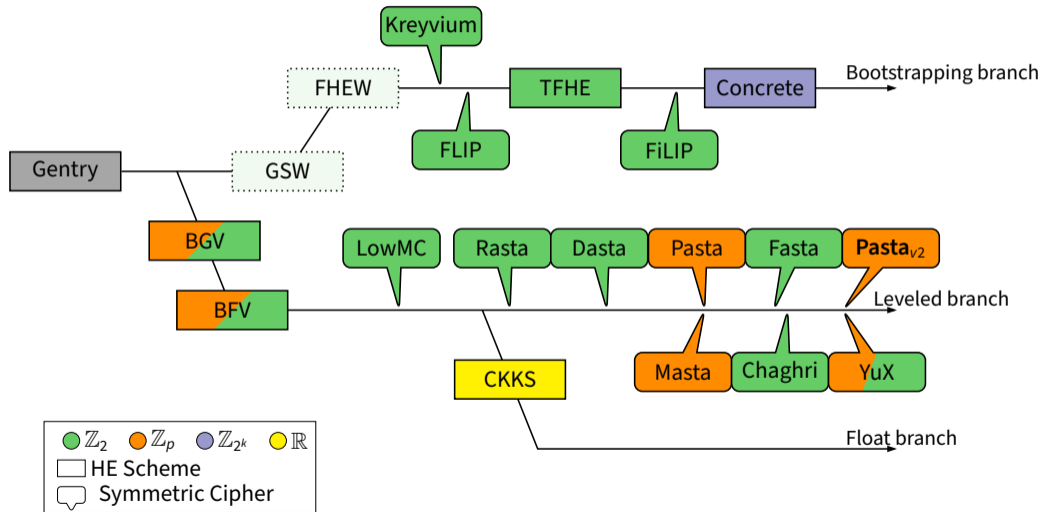
A Zoo of Ciphers for HHE



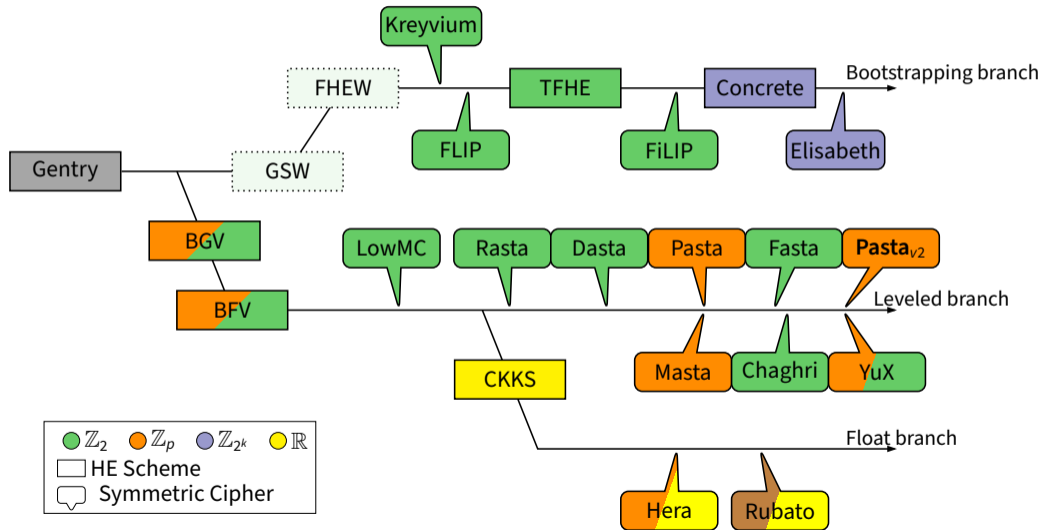
A Zoo of Ciphers for HHE



A Zoo of Ciphers for HHE



A Zoo of Ciphers for HHE



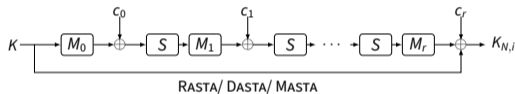
Design of PASTA_{v2}



Background: Design of RASTA-like Ciphers

Randomized Stream-Ciphers

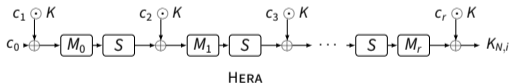
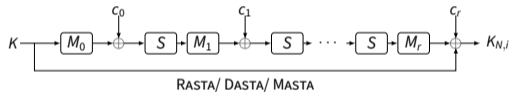
- **RASTA:**
 - Random invertible matrices
 - Random round constants
- **DASTA:**
 - Improved matrix generation
- **MASTA:**
 - RASTA strategy applied to \mathbb{F}_p



Background: Design of RASTA-like Ciphers

Randomized Stream-Ciphers

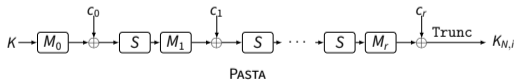
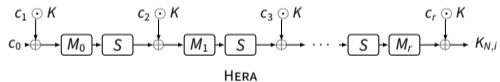
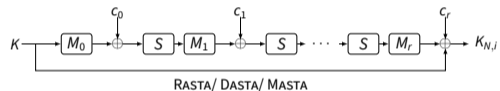
- HERA:
 - Fixed matrices
 - Randomized round keys
 - Small statesize



Background: Design of RASTA-like Ciphers

Randomized Stream-Ciphers

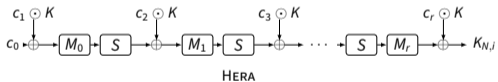
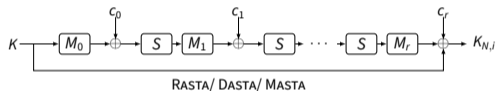
- PASTA:
 - Matrices with **high** branch number
 - **Truncation** of output
 - Geared towards **HE** evaluation



Background: Design of RASTA-like Ciphers

Randomized Stream-Ciphers

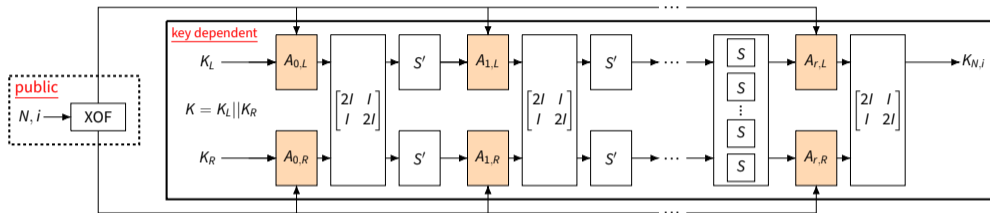
- PASTA:
 - Matrices with **high** branch number
 - **Truncation** of output
 - Geared towards **HE** evaluation



Randomization dominates encryption cost



The PASTA Design Strategy – Linear Layer

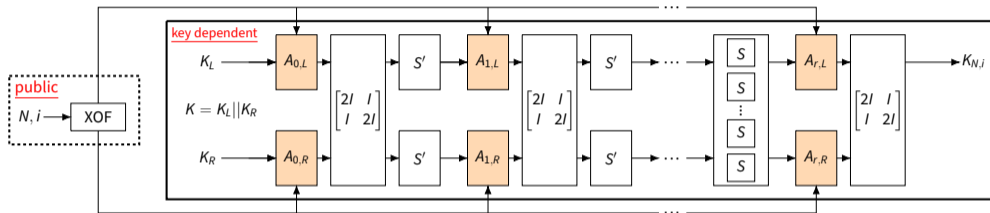


Linear Layer:

$$\begin{bmatrix} \vec{y}_L \\ \vec{y}_R \end{bmatrix} = \begin{bmatrix} 2 \cdot I & I \\ I & 2 \cdot I \end{bmatrix} \times \begin{bmatrix} M_{j,L,N,i}(\vec{x}_L) + C_{j,L,N,i} \\ M_{j,R,N,i}(\vec{x}_R) + C_{j,R,N,i} \end{bmatrix}$$

- Different **random** matrices and constants in each round.

The PASTA Design Strategy – Linear Layer

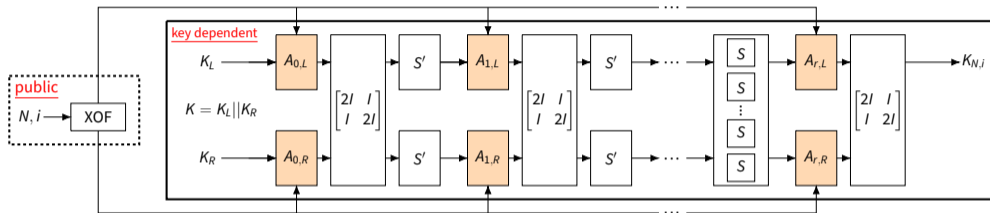


Linear Layer:

$$\begin{bmatrix} \vec{y}_L \\ \vec{y}_R \end{bmatrix} = \begin{bmatrix} 2 \cdot I & I \\ I & 2 \cdot I \end{bmatrix} \times \begin{bmatrix} M_{j,L,N,i}(\vec{x}_L) + C_{j,L,N,i} \\ M_{j,R,N,i}(\vec{x}_R) + C_{j,R,N,i} \end{bmatrix}$$

- Different random matrices and constants in each round.

The PASTA Design Strategy – Linear Layer

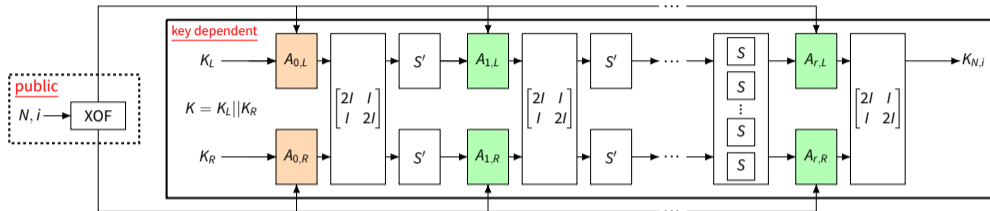


Linear Layer:

$$\begin{bmatrix} \vec{y}_L \\ \vec{y}_R \end{bmatrix} = \begin{bmatrix} 2 \cdot I & I \\ I & 2 \cdot I \end{bmatrix} \times \begin{bmatrix} M_{j,L,N,i}(\vec{x}_L) + C_{j,L,N,i} \\ M_{j,R,N,i}(\vec{x}_R) + C_{j,R,N,i} \end{bmatrix}$$

- Different **random matrices** and **constants** in each round.

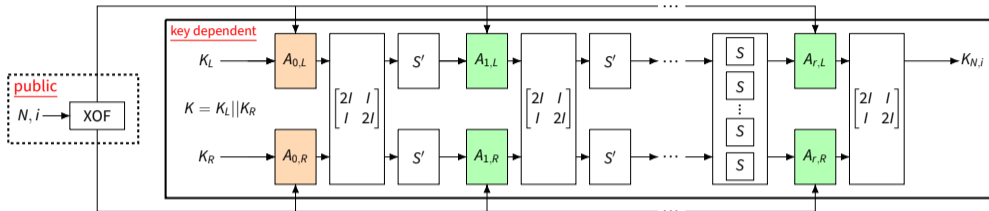
The Birth of PASTA_{v2}



The PASTA_{v2} design

- Replace some **random** with **fixed** affine layers

The Birth of PASTA_{v2}

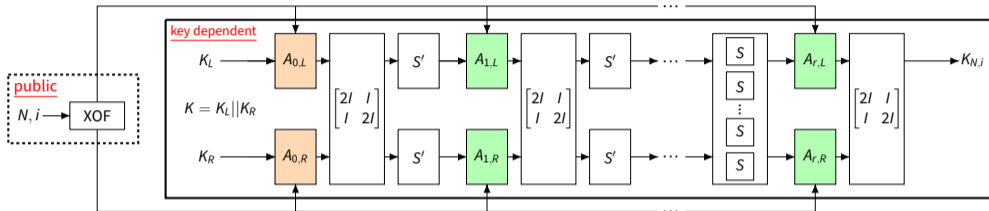


The PASTA_{v2} design

	PASTA	PASTA _{v2}
One-time Setup	-	244 052
Affine Gen	23 550	6 200
Setup/Block	246 995	13 099

Table: Setup generation cost in CPU cycles

The Birth of PASTA_{v2}



The PASTA_{v2} design

$$\text{PASTA}_{v2}\text{-}\pi(x, N, i) = A_r \circ S \circ A_{r-1} \circ S' \circ \dots \circ A_1 \circ S' \circ A_{0,N,i}(x)$$

Randomized Linear Layer

- Define fixed $M_{f,L}$ and $M_{f,R}$ as in PASTA
- During encryption, sample $2t$ random elements $(\beta_1, \dots, \beta_{2t})$ and generate:

$$M_{0,L,N,i} = M_{f,L} \times \text{diag}(\beta_1, \dots, \beta_t)$$

$$M_{0,R,N,i} = M_{f,R} \times \text{diag}(\beta_{t+1}, \dots, \beta_{2t})$$

- Only $4t$ random elements per encryption
- Reduced to $2t$ field multiplications

Randomized Linear Layer

- Define fixed $M_{f,L}$ and $M_{f,R}$ as in PASTA
- During encryption, sample $2t$ random elements $(\beta_1, \dots, \beta_{2t})$ and generate:

$$M_{0,L,N,i} = M_{f,L} \times \text{diag}(\beta_1, \dots, \beta_t)$$

$$M_{0,R,N,i} = M_{f,R} \times \text{diag}(\beta_{t+1}, \dots, \beta_{2t})$$

- Only $4t$ random elements per encryption
- Reduced to $2t$ field multiplications

Randomized Linear Layer

- Define fixed $M_{f,L}$ and $M_{f,R}$ as in PASTA
- During encryption, sample $2t$ random elements $(\beta_1, \dots, \beta_{2t})$ and generate:

$$M_{0,L,N,i} = M_{f,L} \times \text{diag}(\beta_1, \dots, \beta_t)$$

$$M_{0,R,N,i} = M_{f,R} \times \text{diag}(\beta_{t+1}, \dots, \beta_{2t})$$

- Only $4t$ random elements per encryption
- Reduced to $2t$ field multiplications

Fixed Linear Layer

- We define the **fixed affine layers** as:

$$A_j(x) = \begin{bmatrix} 2 \cdot I & I \\ I & 2 \cdot I \end{bmatrix} \times \begin{bmatrix} M(x_L) + c_{j,L} \\ M(x_R) + c_{j,R} \end{bmatrix}$$

- The fixed MDS matrix **M** is a random cauchy matrix

Fixed Linear Layer

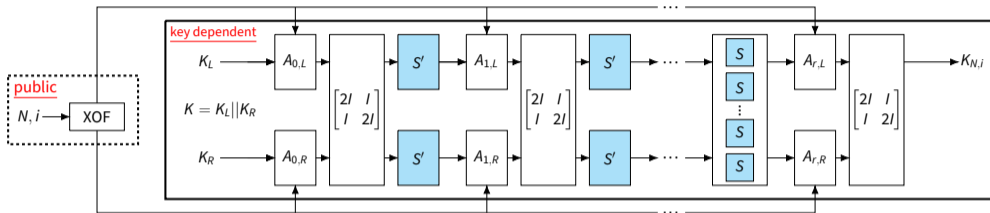
- We define the **fixed affine layers** as:

$$A_j(x) = \begin{bmatrix} 2 \cdot I & I \\ I & 2 \cdot I \end{bmatrix} \times \begin{bmatrix} M(x_L) + c_{j,L} \\ M(x_R) + c_{j,R} \end{bmatrix}$$

- The fixed MDS matrix **M** is a random cauchy matrix

We provide a proof that **the branch number of A_j is $t + 2$**

The Non-Linear Layers



- Feistel-like S-box:

- Low-degree \Rightarrow low depth

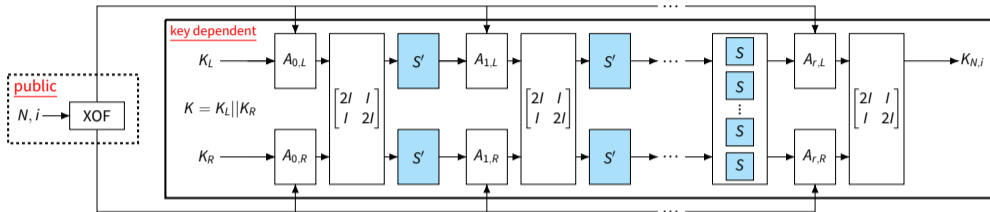
$$[S'(\vec{x})]_i = \begin{cases} x_0 & \text{if } i = 0 \\ x_i + (x_{i-1})^2 & \text{else} \end{cases}$$

- Cube S-box:

- Higher degree
- Only last round

$$S(x) = x^3$$

The Non-Linear Layers



- Feistel-like S-box:

- Low-degree \Rightarrow low depth

$$[S'(\vec{x})]_i = \begin{cases} x_0 & \text{if } i = 0 \\ x_i + (x_{i-1})^2 & \text{else} \end{cases}$$

- Cube S-box:

- Higher degree
 - Only last round

$$S(x) = x^3$$

Security Analysis

- Randomization provides **resistance against:**
 - Differential, truncated differential, and impossible differential attacks
 - Cube attacks and higher order differentials
- Linear Cryptanalysis breaking $PASTA_{v2}$ reduced to LWE
 - High minimum of active non-linear operations
- Algebraic Attacks set up independent variables for all monomials
 - Experiments showed a high number of monomials
 - Randomizing only the first linear layer suffices
 - Peeling off the first layer would affect HERA and PASTA

Security Analysis

- Randomization provides **resistance against:**
 - Differential, truncated differential, and impossible differential attacks
 - Cube attacks and higher order differentials
- **Linear Cryptanalysis** breaking $PASTA_{v2}$ reduced to **LWE**
 - High minimum of **active non-linear operations**
- **Algebraic Attacks** set up independent variables for all monomials
 - Experiments showed a high number of monomials
 - Randomizing **only the first linear layer** suffices
 - Peeling off the first layer would affect **HERA and PASTA**

Security Analysis

- Randomization provides **resistance against**:
 - Differential, truncated differential, and impossible differential attacks
 - Cube attacks and higher order differentials
- **Linear Cryptanalysis** breaking $PASTA_{v2}$ reduced to **LWE**
 - High minimum of **active non-linear operations**
- **Algebraic Attacks** set up independent variables for all monomials
 - Experiments showed a high number of monomials
 - Randomizing **only the first** linear layer suffices
 - Peeling off the first layer would **affect HERA and PASTA**

PASTA_{v2} Instances

- We specify instances with the same security level as PASTA

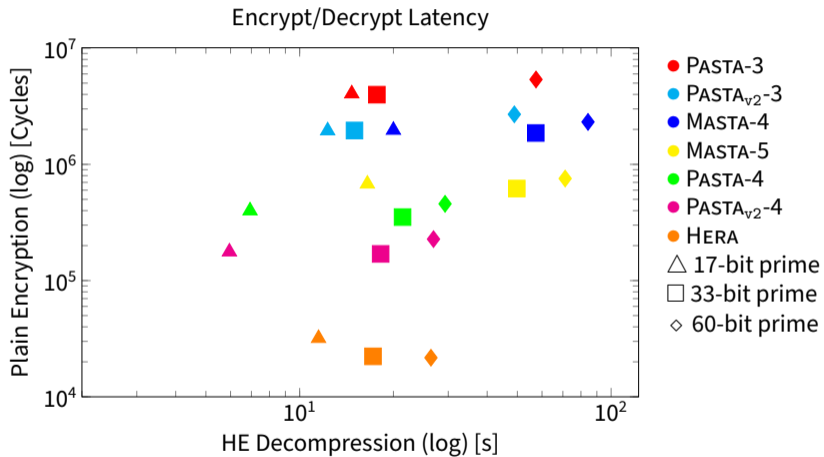
Instance	r	# Key Words	# Plain/Cipher Words	# random words
PASTA _{v2} -3	3	256	128	512
PASTA _{v2} -4	4	64	32	128
PASTA-3	3	256	128	2048
PASTA-4	4	64	32	640

Table: 128 bit security instances of PASTA_{v2} and PASTA

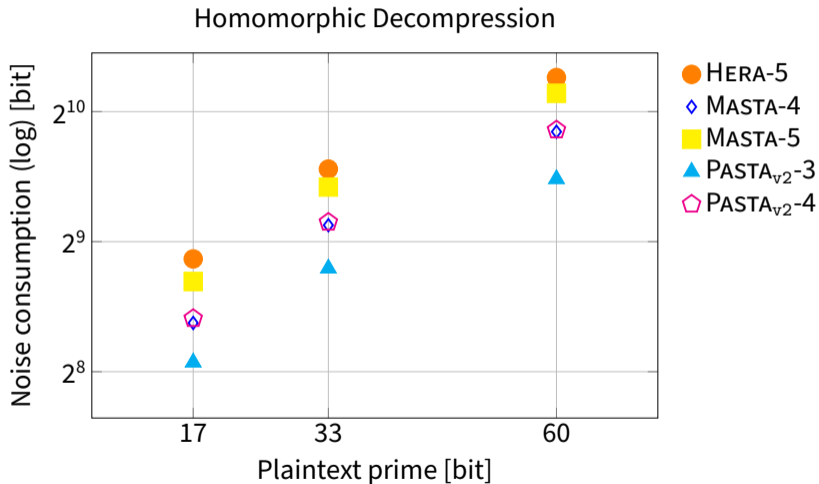
Benchmarks



Overall Performance



Noise development



Software Implementation - Overview

- We provide **open-source** implementation
 - Integration with HHE **benchmarking framework**¹
 - HE Decompression implementation in **SEAL** and **HElib**
 - **C++** plaintext implementation for **encryption**
- More complex use case evaluation in the paper
 - Similar results for respective **PASTA** and **PASTA_{v2}** instances
 - Less noise leads to smaller parameters and better performance

¹<https://github.com/IAIK/hybrid-HE-framework/>

Software Implementation - Overview

- We provide **open-source** implementation
 - Integration with HHE **benchmarking framework**¹
 - HE Decompression implementation in **SEAL** and **HElib**
 - C++ plaintext implementation for **encryption**
- More complex **use case** evaluation in the paper
 - Similar results for respective **PASTA** and **PASTA_{v2}** instances
 - Less noise leads to **smaller parameters** and **better performance**

¹<https://github.com/IAIK/hybrid-HE-framework/>

Summary

- $PASTA_{v2}$ improves PASTA
 - Faster Encryption and slightly faster Homomorphic decompression
 - Provably high branch number in fixed linear layers
 - Same security level for a fraction of required random words
- This strategy can be applied to RASTA
- We minimize randomness in $PASTA_{v2}$
 - We encourage further cryptanalysis of $PASTA_{v2}$
 - Additional analysis helps understanding RASTA-like designs

Summary

- $PASTA_{v2}$ improves PASTA
 - Faster Encryption and slightly faster Homomorphic decompression
 - Provably high branch number in fixed linear layers
 - Same security level for a fraction of required random words
- This strategy can be applied to RASTA
- We minimize randomness in $PASTA_{v2}$
 - We encourage further cryptanalysis of $PASTA_{v2}$
 - Additional analysis helps understanding RASTA-like designs

Summary

- $PASTA_{v2}$ improves PASTA
 - Faster Encryption and slightly faster Homomorphic decompression
 - Provably high branch number in fixed linear layers
 - Same security level for a fraction of required random words
- This strategy can be applied to RASTA
- We minimize randomness in $PASTA_{v2}$
 - We encourage further cryptanalysis of $PASTA_{v2}$
 - Additional analysis helps understanding RASTA-like designs

Minimize the Randomness in Rasta-Like Designs: How Far Can We Go?

Application to PASTA

Lorenzo Grassi, Fukang Liu, Christian Rechberger,
Fabian Schmid, Roman Walch, and Qingju Wang

28.08.2024