TU Graz

SCIENCE
PASSION
TECHNOLOGY

First Order Masking for Hybrid Homomorphic Encryption Schemes

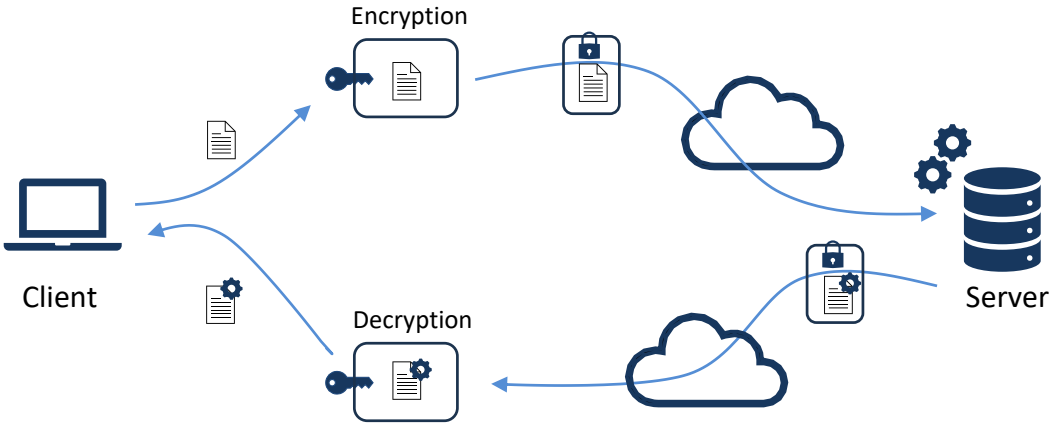
Nedžma Mušović

Master Project Presentation

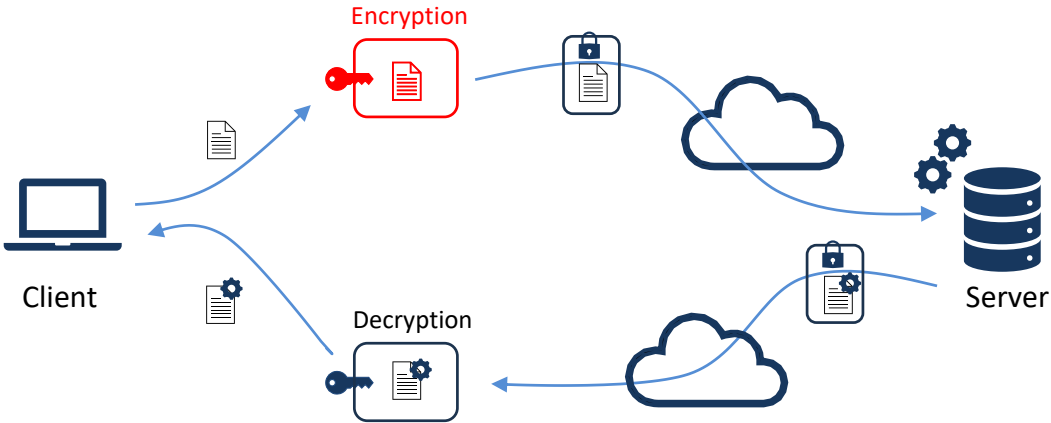
Outline

- Introduction
- **HERA** - First Order Masking
- Application to **PASTA**
- Higher Order Masking and Conclusion

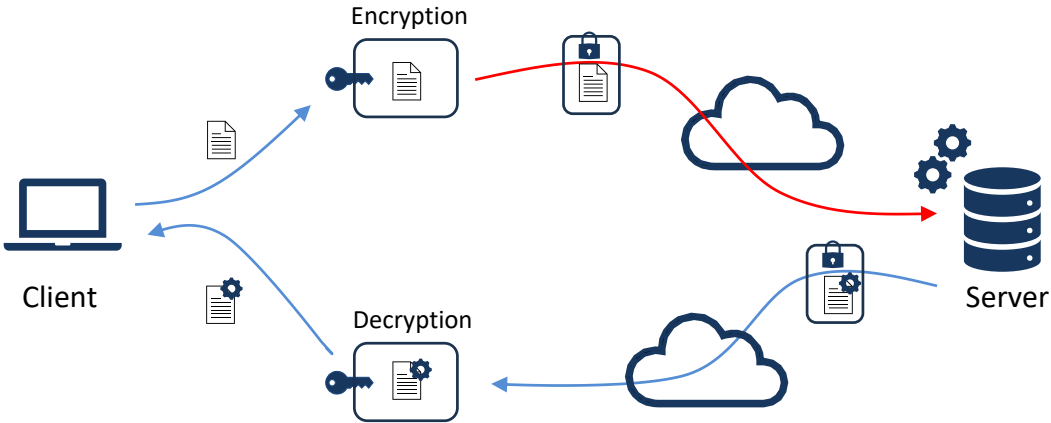
Homomorphic Encryption



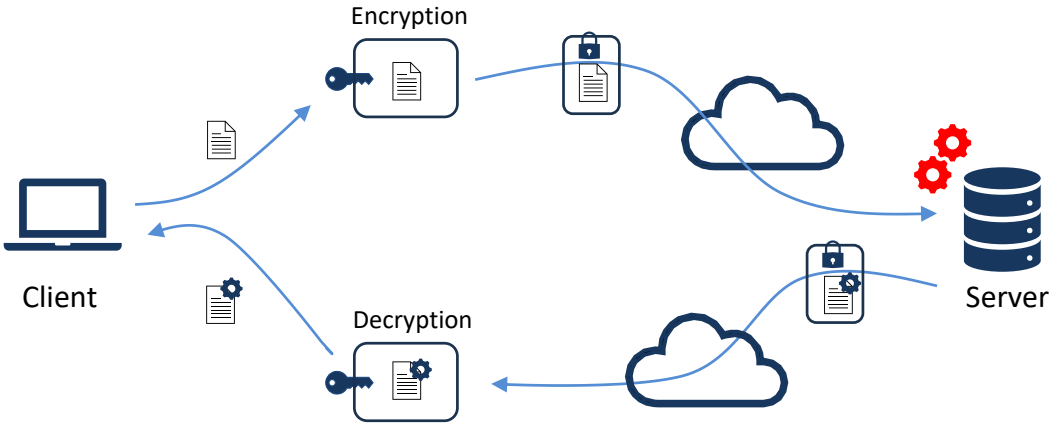
Homomorphic Encryption



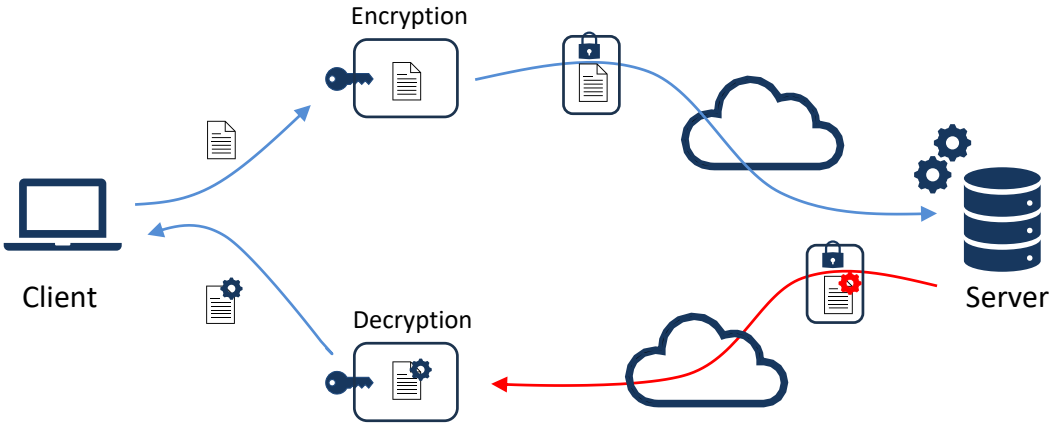
Homomorphic Encryption



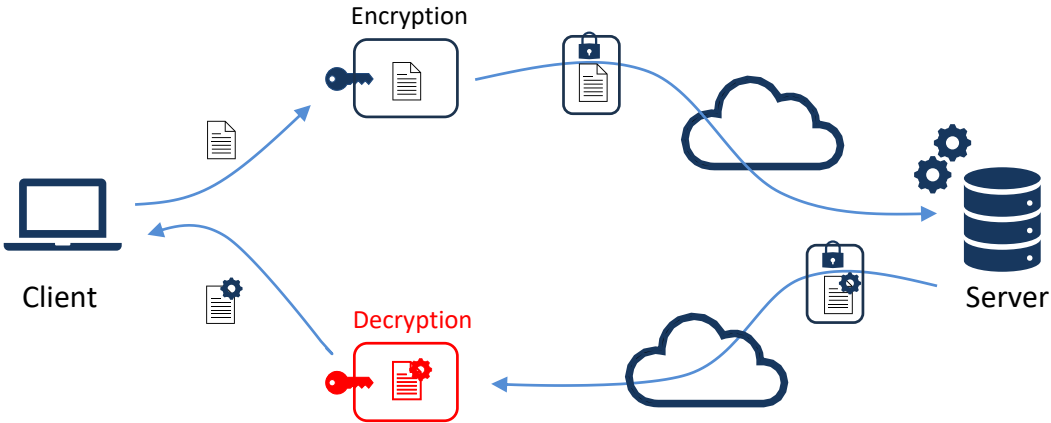
Homomorphic Encryption



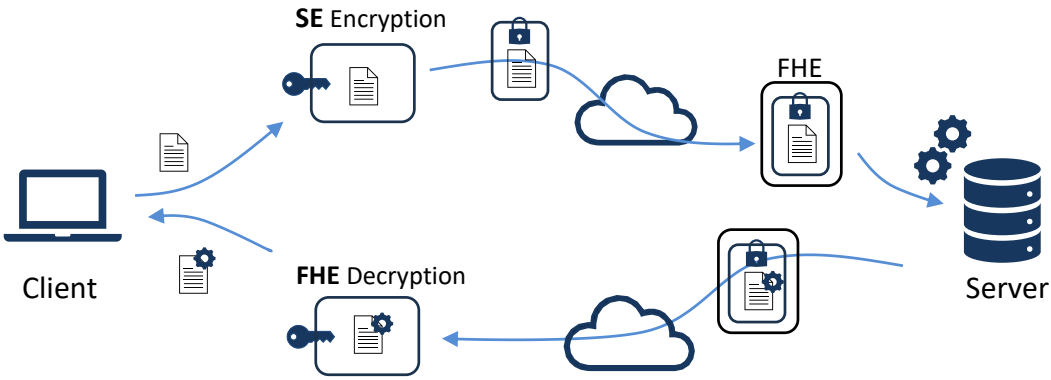
Homomorphic Encryption



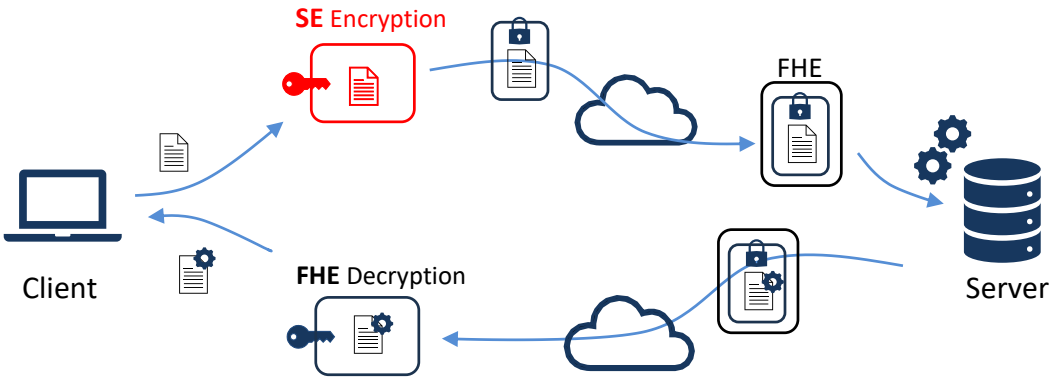
Homomorphic Encryption



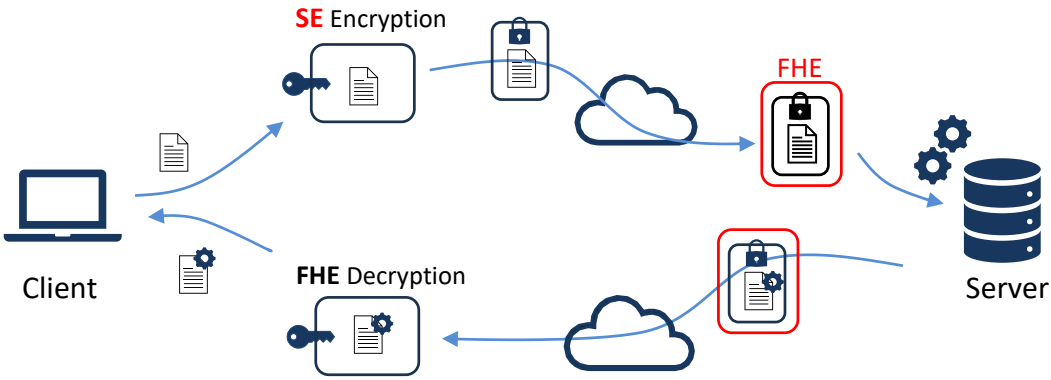
Hybrid Homomorphic Encryption



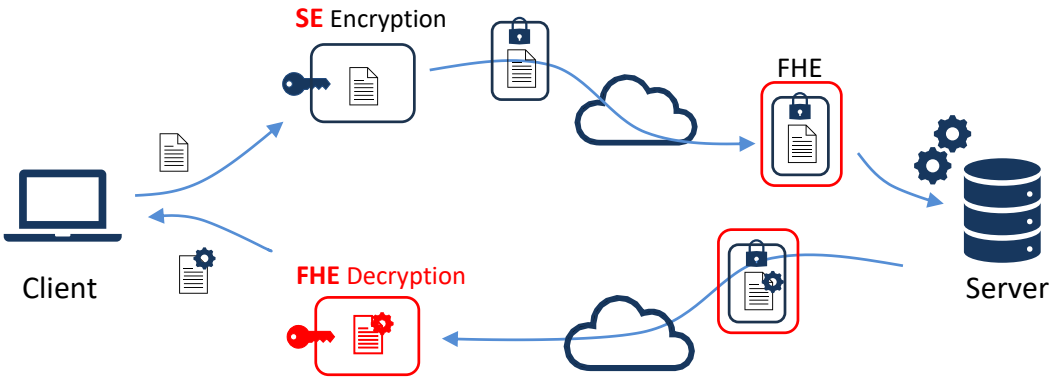
Hybrid Homomorphic Encryption



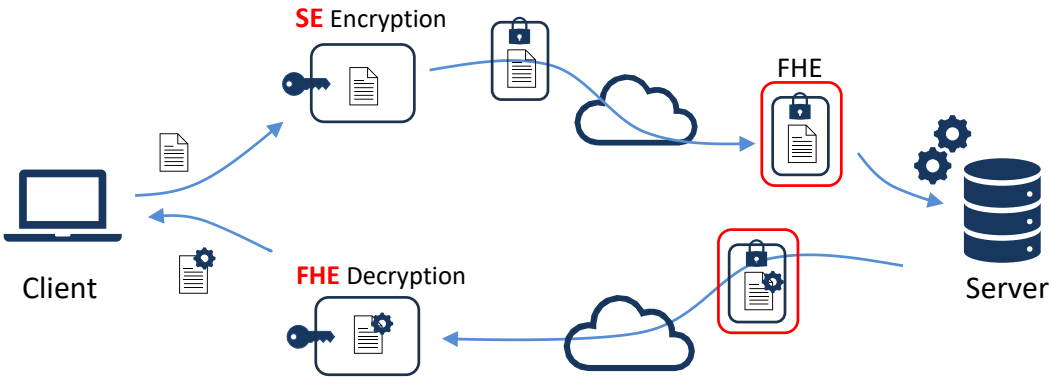
Hybrid Homomorphic Encryption



Hybrid Homomorphic Encryption



Hybrid Homomorphic Encryption



HERA Encryption

- Prime-field arithmetic; 16 to 60 bits prime number p.
- Generates a keystream from the input key *K* and adds it to the plaintext.
- Typical **HERA** permutation consists out of 5 rounds.
- State size: $n = 16$; hence state $X \in \mathbb{Z}_p^n$.

HERA Encryption

- Prime-field arithmetic; 16 to 60 bits prime number p.
- Generates a keystream from the input key *K* and adds it to the plaintext.
- Typical **HERA** permutation consists out of 5 rounds.
- State size: $n = 16$; hence state $X \in \mathbb{Z}_p^n$.

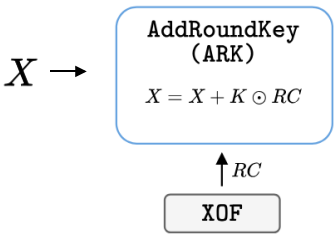
HERA Encryption

- Prime-field arithmetic; 16 to 60 bits prime number p.
- Generates a keystream from the input key *K* and adds it to the plaintext.
- Typical **HERA** permutation consists out of 5 rounds.
- State size: $n = 16$; hence state $X \in \mathbb{Z}_p^n$.

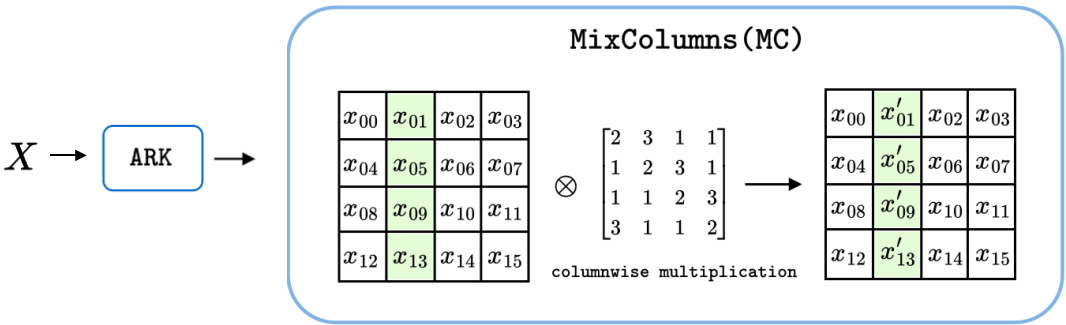
HERA Encryption

- Prime-field arithmetic; 16 to 60 bits prime number p.
- Generates a keystream from the input key *K* and adds it to the plaintext.
- Typical **HERA** permutation consists out of 5 rounds.
- State size: $n = 16$; hence state $X \in \mathbb{Z}_p^n$.

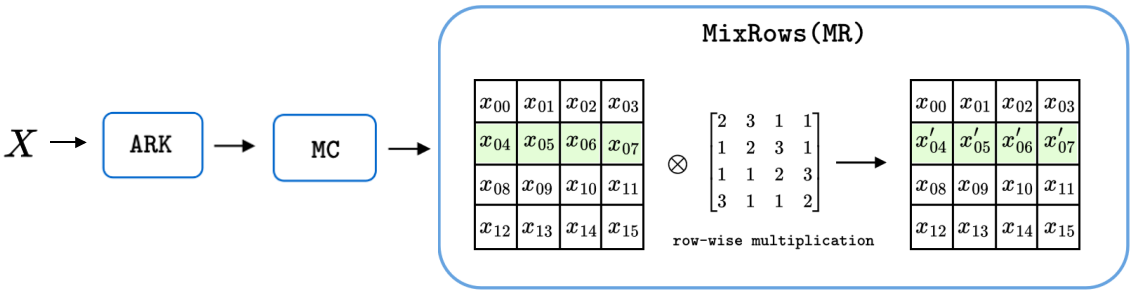
HERA Encryption



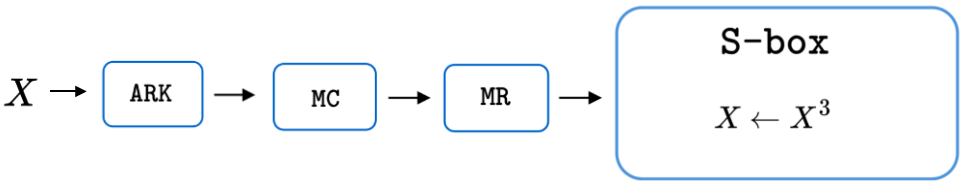
HERA Encryption



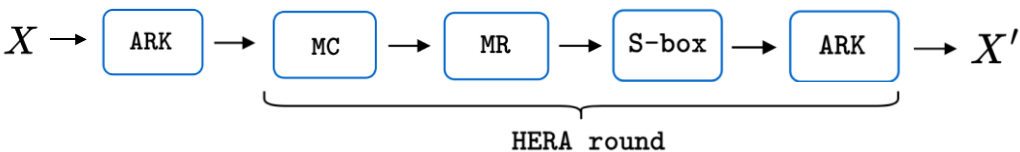
HERA Encryption



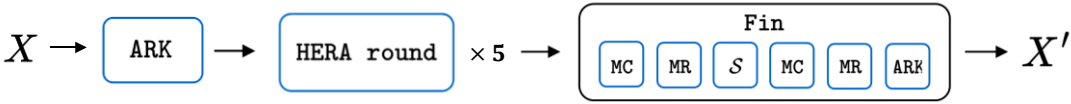
HERA Encryption



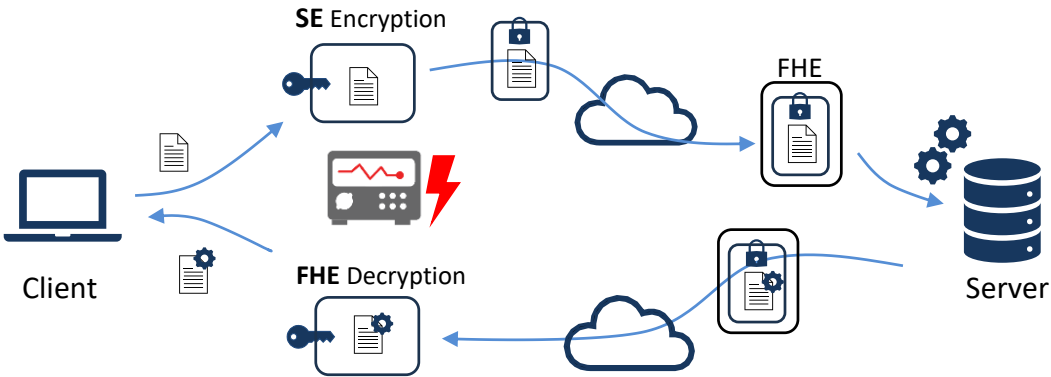
HERA Encryption



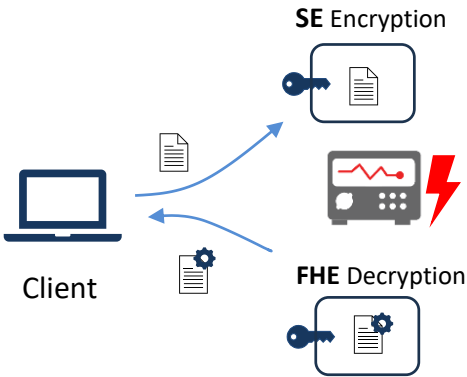
HERA Encryption



HERA Side-Channels Potential

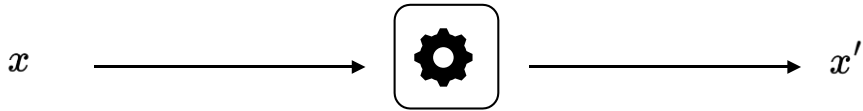


HERA Side-Channels Potential

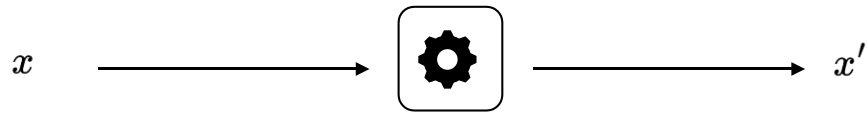


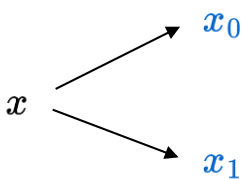
- Exploit relation between power consumption and processed data.
- Statistical methods to guess secret key bits - **DPA/CPA** possible.
- Modular arithmetics spreads side-channel leakage.

First Order Masking

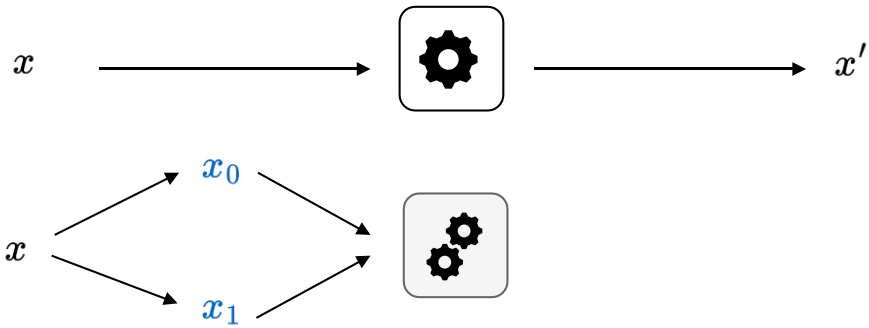


First Order Masking

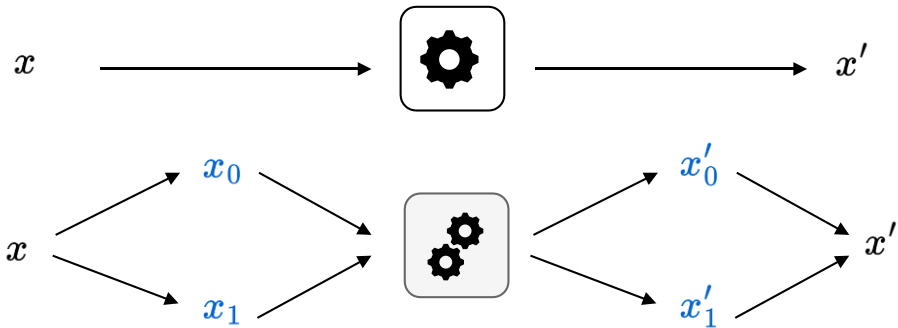




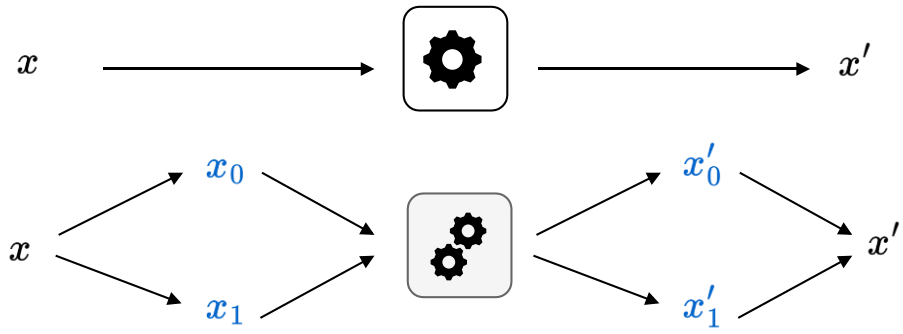
First Order Masking



First Order Masking

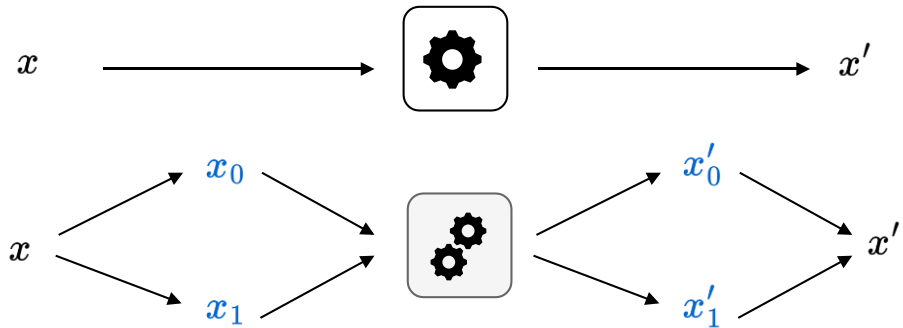


First Order Masking



Boolean sharing

First Order Masking



Arithmetic sharing

$$x = x_0 + x_1 \qquad x' = x'_0 + x'_1$$

Test Vector Leakage Assessment (TVLA)

- Standard statistical method to detect **side-channel leakage**.
- Compares device traces for **fixed vs random inputs**.
- Applies **Welch's t-test** at each time point.
- Evidence of leakage: $|t| > 4.5$.

Test Vector Leakage Assessment (TVLA)

- Standard statistical method to detect **side-channel leakage**.
- Compares device traces for **fixed vs random inputs**.
- Applies **Welch's t-test** at each time point.
- Evidence of leakage: $|t| > 4.5$.

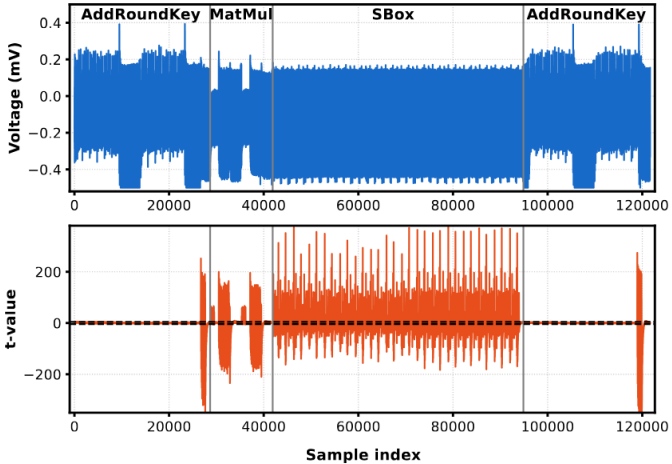
Test Vector Leakage Assessment (TVLA)

- Standard statistical method to detect **side-channel leakage**.
- Compares device traces for **fixed vs random inputs**.
- Applies **Welch's t-test** at each time point.
- Evidence of leakage: $|t| > 4.5$.

Test Vector Leakage Assessment (TVLA)

- Standard statistical method to detect **side-channel leakage**.
- Compares device traces for **fixed vs random inputs**.
- Applies **Welch's t-test** at each time point.
- Evidence of leakage: $|t| > 4.5$.

TVLA Results for HERA



Masking a HERA round

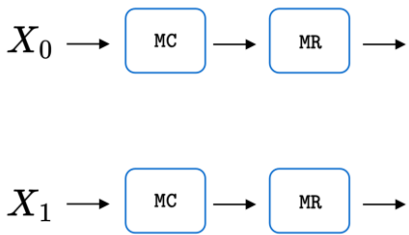
$$X = X_0 + X_1 \pmod{p}$$

$X_0 \longrightarrow$

$X_1 \longrightarrow$

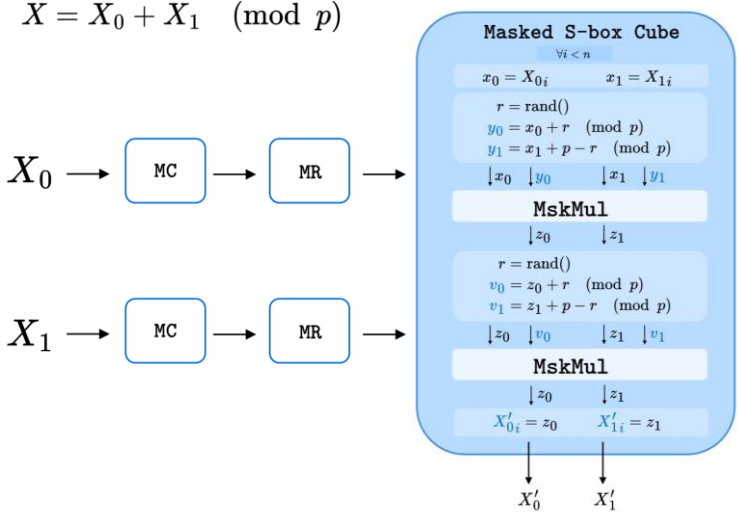
Masking a HERA round

$$X = X_0 + X_1 \pmod{p}$$



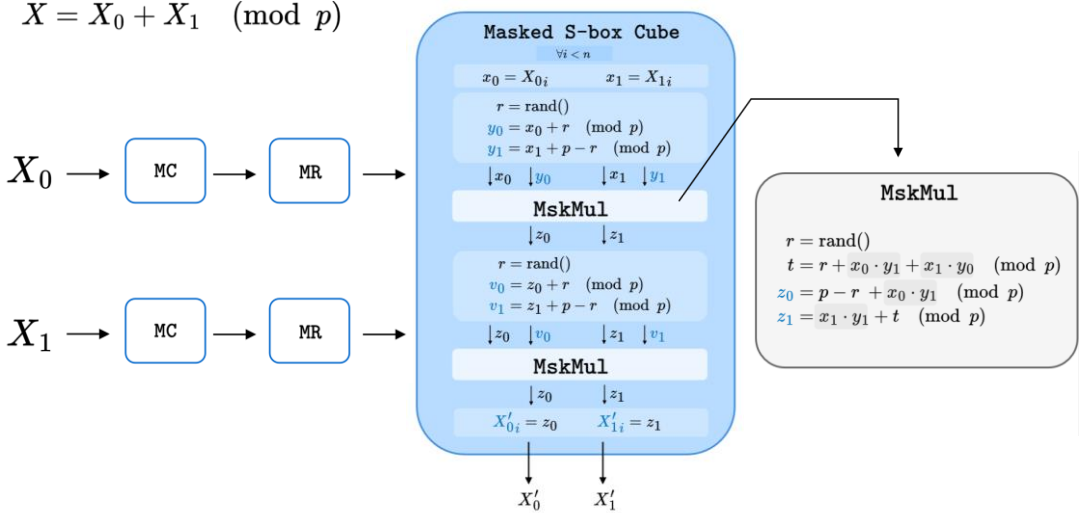
Masking a HERA round

$$X = X_0 + X_1 \pmod{p}$$



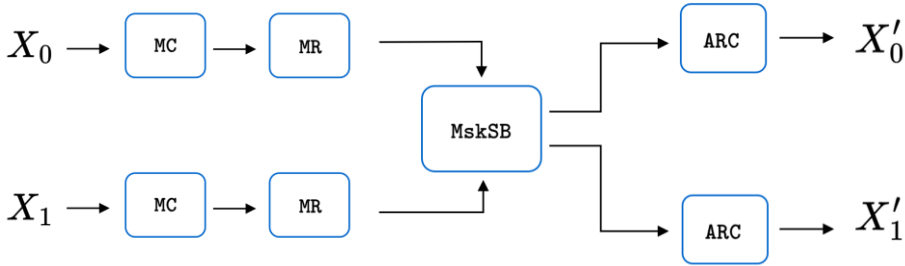
Masking a HERA round

$$X = X_0 + X_1 \pmod{p}$$

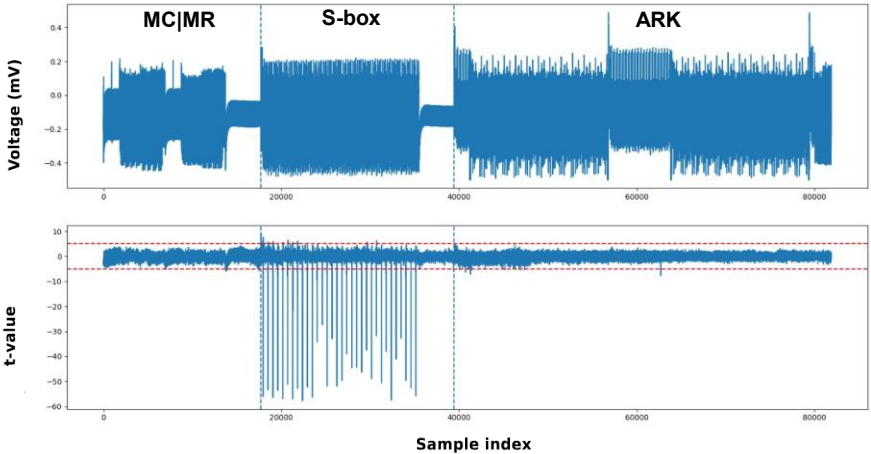


Masking a HERA round

$$X = X_0 + X_1 \pmod{p}$$

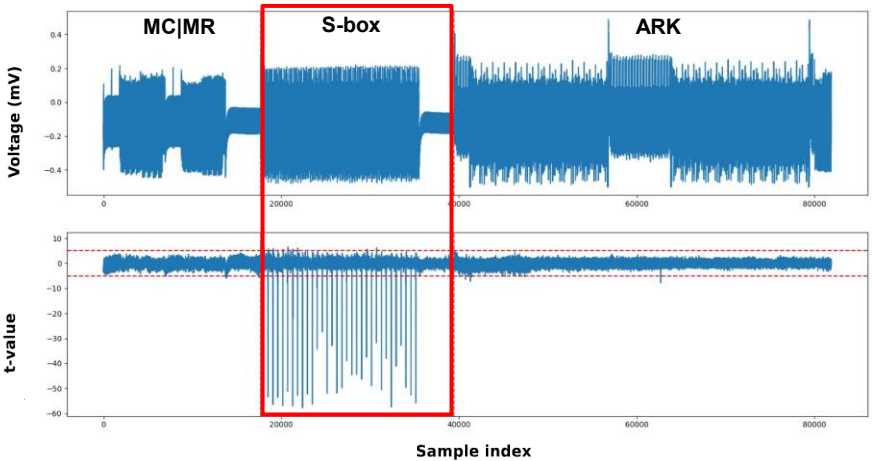


TVLA Results



Preliminary Results

TVLA Results



Implementation Heuristic

- Ensure the random source is random indeed.
- Separate shares usage within the C code.
- Employ assembly code for full register control.
- Address critical points which could lead to the statistical leakage despite the masked model.

Implementation Heuristic

- Ensure the random source is random indeed.
- Separate shares usage within the C code.
- Employ assembly code for full register control.
- Address critical points which could lead to the statistical leakage despite the masked model.

Implementation Heuristic

- Ensure the random source is random indeed.
- Separate shares usage within the C code.
- Employ assembly code for full register control.
- Address critical points which could lead to the statistical leakage despite the masked model.

Implementation Heuristic

- Ensure the random source is random indeed.
- Separate shares usage within the C code.
- Employ assembly code for full register control.
- Address critical points which could lead to the statistical leakage despite the masked model.

Implementation Heuristic

$r_4 \leftarrow x_0$

$r_4 \leftarrow x_1$

Implementation Heuristic

$$r_4 \leftarrow x_0$$
$$r_4 \leftarrow x_1$$

$$\mathtt{HD}(x_0, x_1) = \mathtt{HW}(x_0 \oplus x_1) = \mathtt{HW}(x)$$

leakage through Hamming distance

Implementation Heuristic

$$r_4 \leftarrow x_0$$
$$r_4 \leftarrow x_1$$

$$\mathtt{HD}(x_0, x_1) = \mathtt{HW}(x_0 \oplus x_1) = \mathtt{HW}(x)$$

leakage through Hamming distance

$$r_3 \leftarrow x_0$$
$$r_4 \leftarrow x_1$$

Implementation Heuristic

$r_3 \leftarrow x_0$

Implementation Heuristic

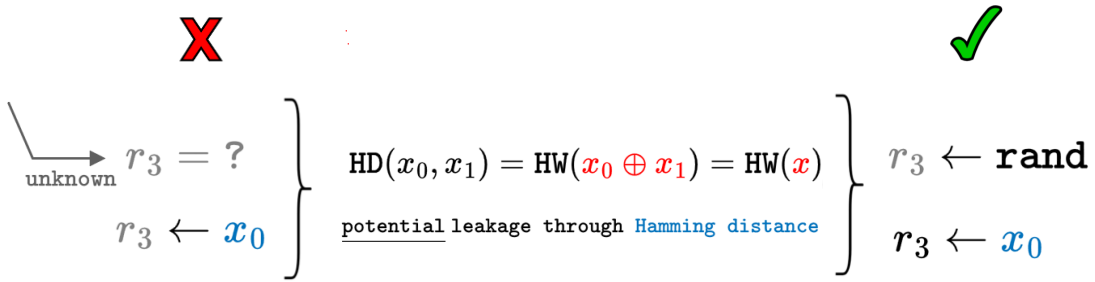
unknown $r_3 = ?$

$r_3 \leftarrow x_0$

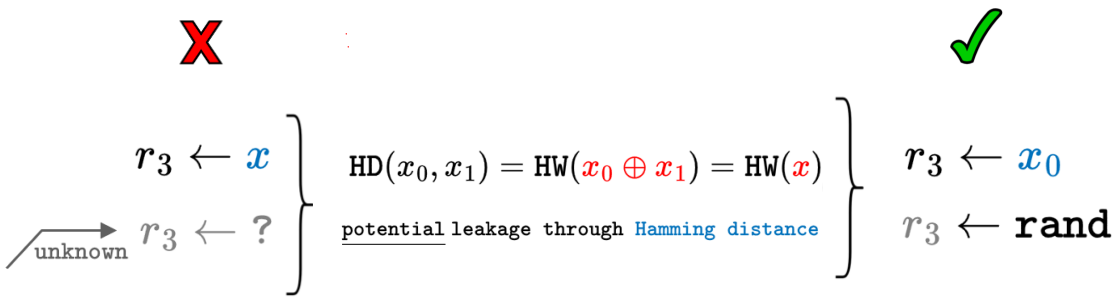
$$\mathtt{HD}(x_0, x_1) = \mathtt{HW}(x_0 \oplus x_1) = \mathtt{HW}(x)$$

potential leakage through Hamming distance

Implementation Heuristic



Implementation Heuristic

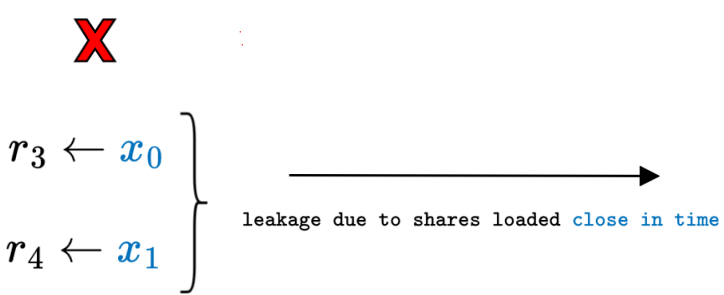


Implementation Heuristic

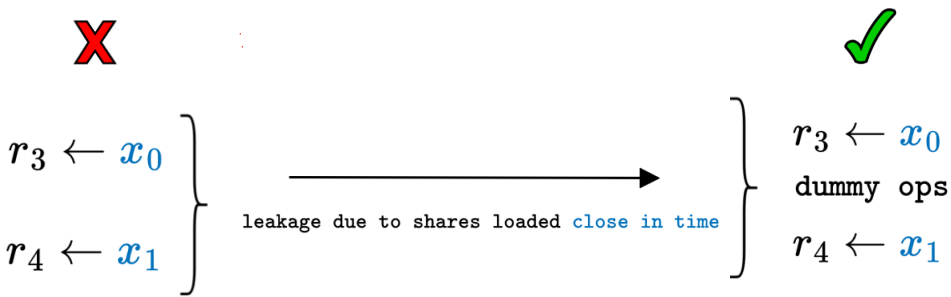
$$r_3 \leftarrow x_0$$

$$r_4 \leftarrow x_1$$

Implementation Heuristic



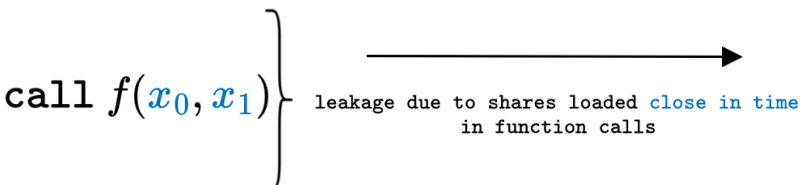
Implementation Heuristic



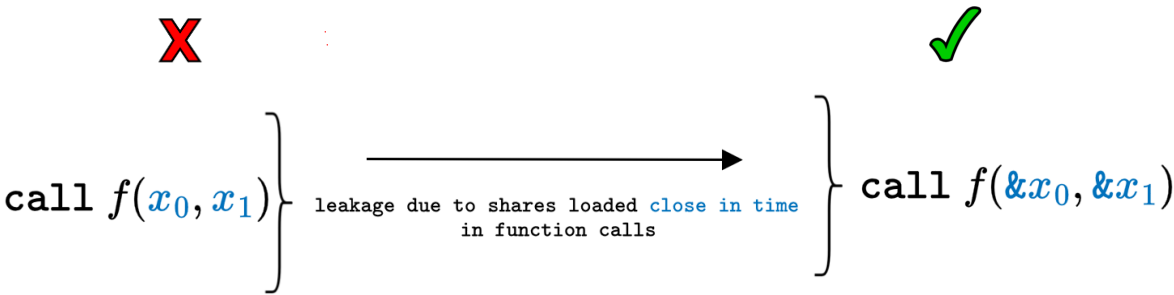
Implementation Heuristic

`call` $f(x_0, x_1)$

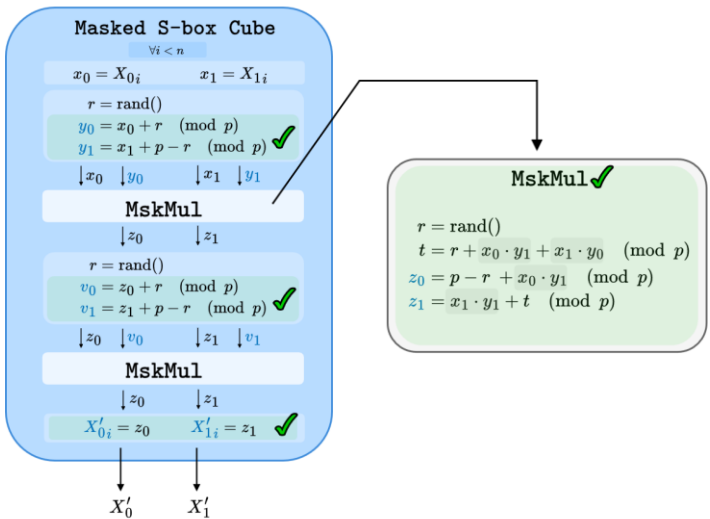
Implementation Heuristic



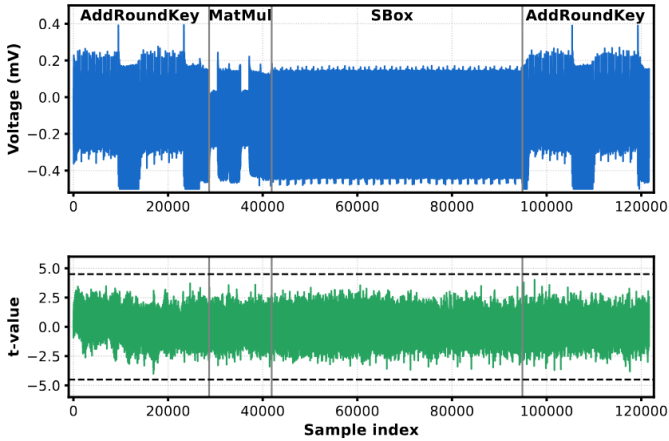
Implementation Heuristic



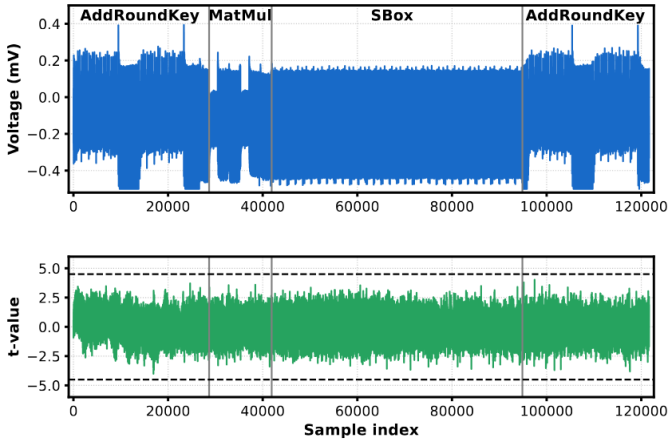
Application to HERA S-box



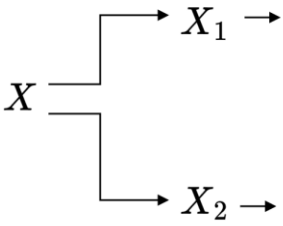
TVLA Results



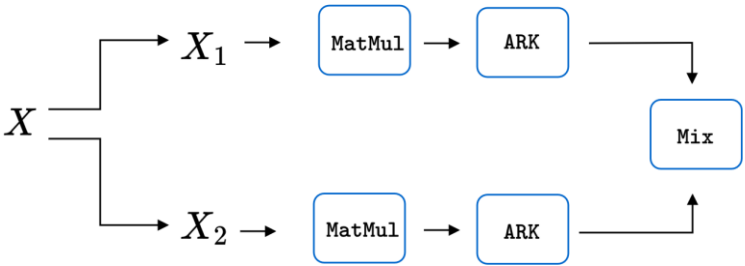
TVLA Results



PASTA Encryption

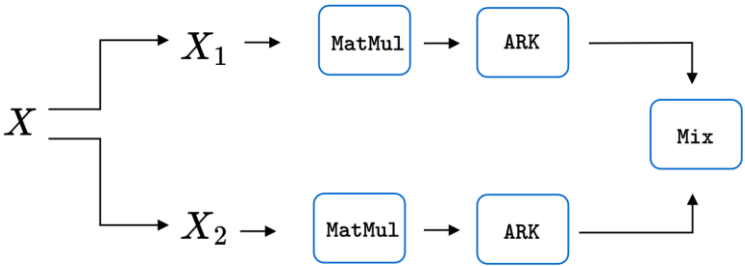


PASTA Encryption



Application to Similar Encryption Schemes

PASTA Encryption

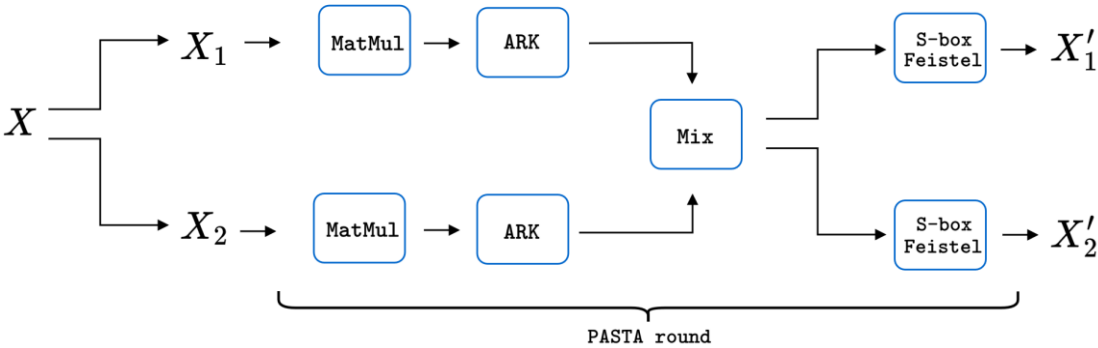


S-box Feistel

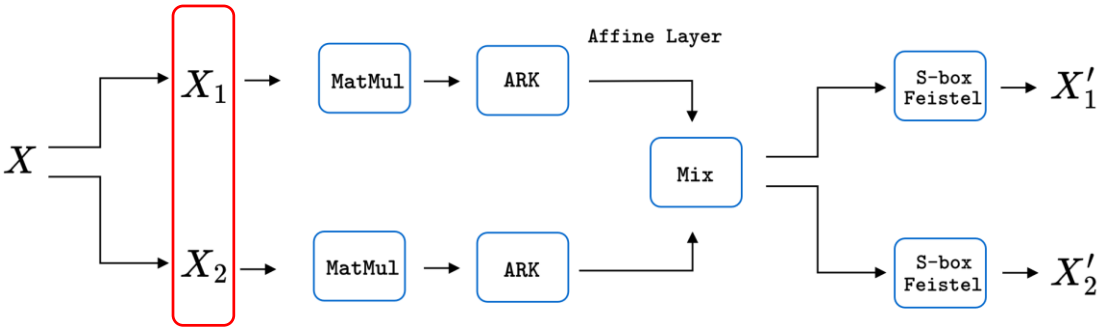
$$\forall i < n$$

$$= \begin{cases} x_i & \text{if } i = 0, \\ x_i + (x_{i-1})^2 & \text{otherwise.} \end{cases}$$

PASTA Encryption

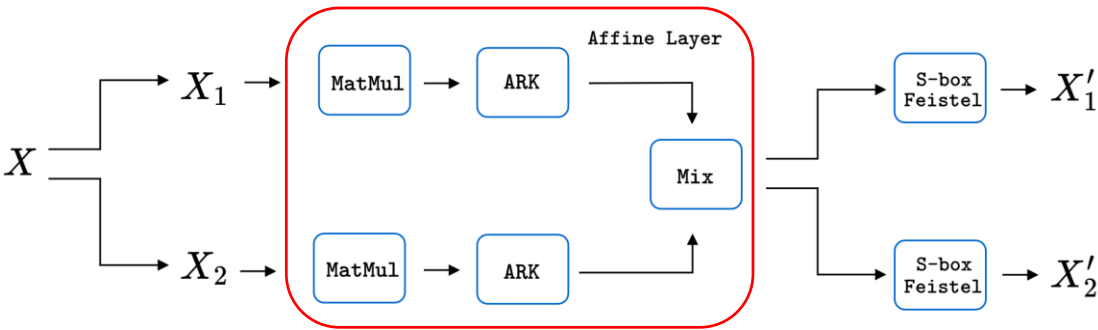


Masking PASTA Round



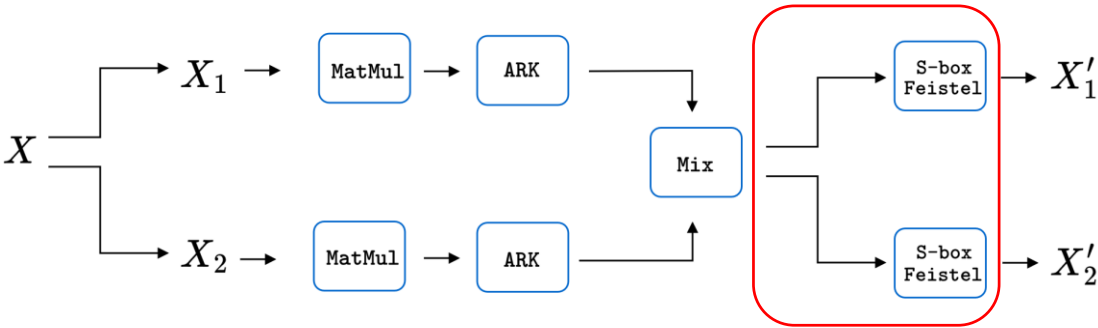
- ✓ mask both parts

Masking PASTA Round



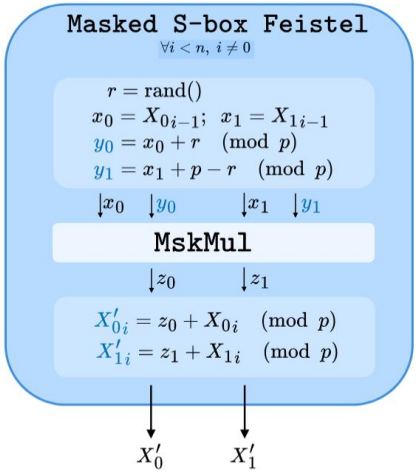
✓ apply linear layers on each share

Masking PASTA Round

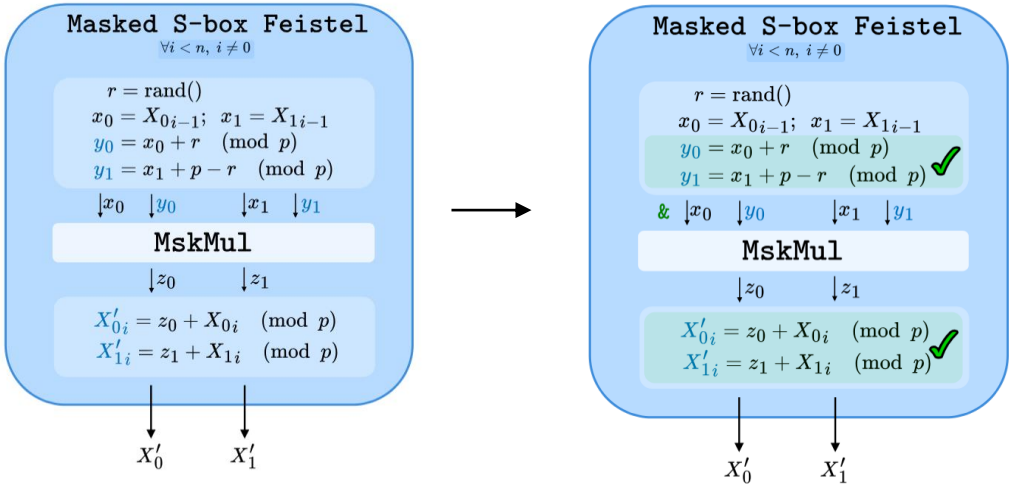


✓ Use secure multiplication

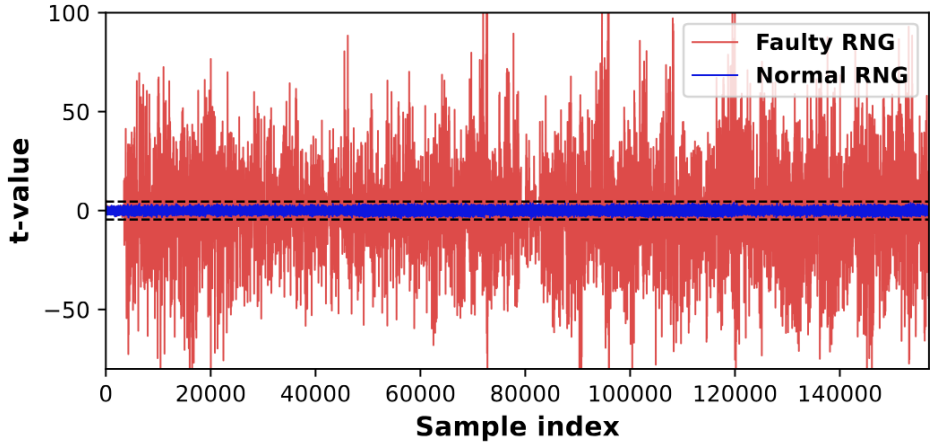
Masking PASTA Round



Masking PASTA Round



TVLA Results



Higher Order Masking

- First order masking protects only against single-probe attacks.
- Extend with **higher order masking** for more security, by expanding the MskMul accordingly.

Conclusion

- HERA vulnerable to side-channel attacks.
- Correct masking model can leak due to implementation specifities.
- The proposed approach application concretely verified on PASTA.
- First order masking protects only against single-probe attacks.