~~Boissier~~ Cryptanalysis of Elisabeth-4

Henri Gilbert[1,2], Rachelle Heim Boissier[2], Jérémy Jean[1], Jean-René Reinhard[1]

[1]ANSSI, [2]UVSQ

Journées C2 2023, Najac

Introduction

About `Elisabeth-4`

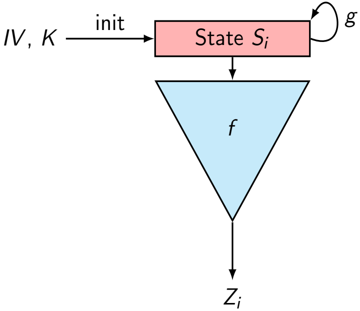
- Stream cipher published at ASIACRYPT 2022.
- Designed by Cosseron, Hoffman, Méaux, Standaert.
- Tailored for Fully Homomorphic Encryption (FHE) use cases.
- 128-bit security claim.

Our contribution (ASIACRYPT 2023)

- Full break of `Elisabeth-4`.
- Linearization attack that exploits:
  - Sparsity of the linear system;
  - Rank defects;
  - Filtering techniques.

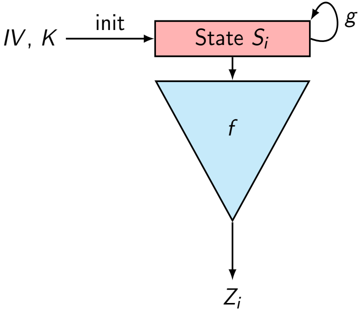
Plan

Stream ciphers



A **stream cipher** uses an IV and a secret key to produce a **keystream sequence** of arbitrary length.

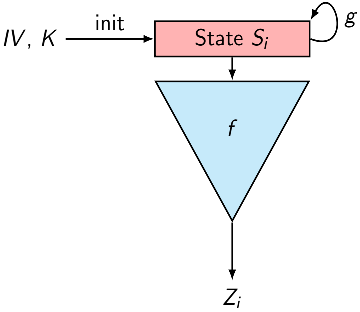
Stream ciphers



A **stream cipher** uses an IV and a secret key to produce a **keystream sequence** of arbitrary length.

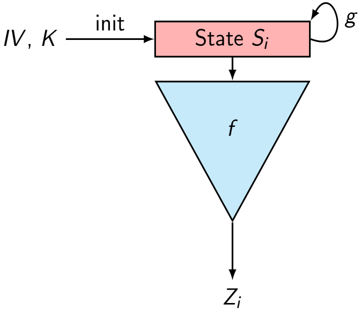
- The attacker has access to the keystream sequence;
- The *IV* is public.

Stream ciphers



A classical cryptanalysis technique: Linearisation

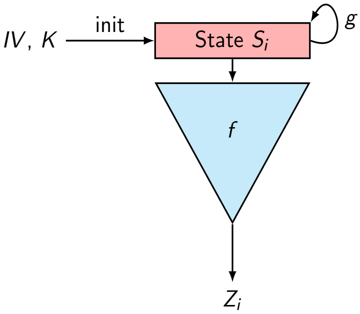
Stream ciphers



A classical cryptanalysis technique: Linearisation

- consider non-linear equations $Z_i = F_i(K_0, \cdots, K_{n-1})$.

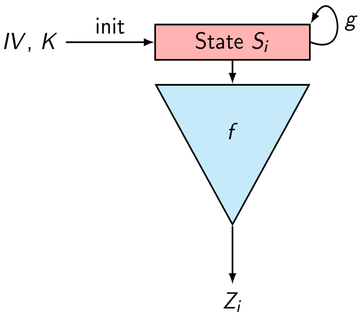
Stream ciphers



A classical cryptanalysis technique: Linearisation

- consider non-linear equations $Z_i = F_i(K_0, \cdots, K_{n-1})$.
- view them as linear equations: view each **monomial** in the key bits as an independant variable.

Stream ciphers



A classical cryptanalysis technique: Linearisation

- consider non-linear equations $Z_i = F_i(K_0, \cdots, K_{n-1})$.
- view them as linear equations: view each **monomial** in the key bits as an independant variable.
- solve the linear system.

Hybrid Homomorphic Encryption

Symmetric key K
Hom. key (SK, PK)
Data D

User | **Server**

- Encrypts D under K using symm. enc. algo
- Encrypts K under PK using hom. enc. algo

$(E^{hom}_{PK}(K), E_K(D)) \longrightarrow$

- *Transciphering*:
  Transforms $E_K(D)$ into $E^{hom}_{PK}(D)$
  using $E^{hom}_{PK}(K)$
- Performs computations homomorphically.
  Obtains R

$\longleftarrow R$

- Decrypts R using SK, and obtains the result of the computation

Symmetric cryptography for FHE

Encryption algorithms for FHE

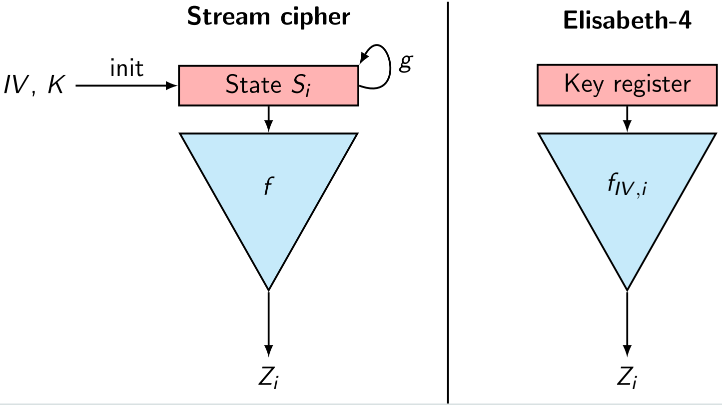
- Classical sym. enc. algorithms (e.g. AES): **not efficient** in FHE.
- This led to the design of **new algorithms**:
  Ex: `LowMC` [ARSTZ16], `Kreyvium` [CCFLNPS16], `FLIP` [CMJS16]
- `Elisabeth-4` is a recent example (ASIACRYPT2022).

`Elisabeth-4` is tailored for **practically relevant FHE applications** (e.g. machine learning algorithms).

`Elisabeth-4` FHE dedicated features

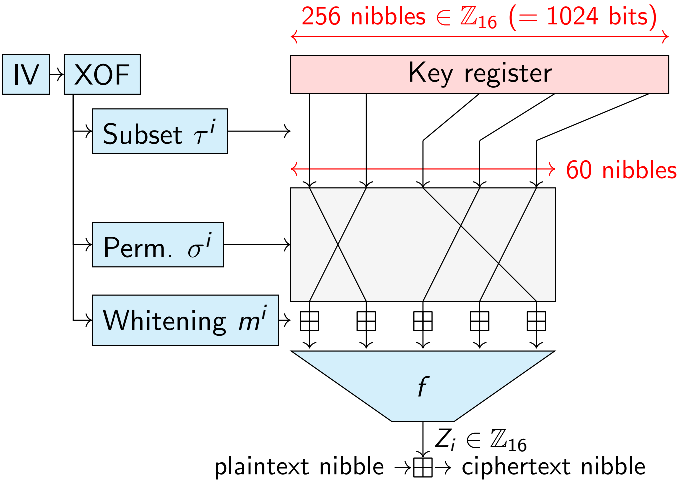
`Elisabeth-4` is...

- specified using operations over $\mathbb{Z}_q$ with $q = 2^4 = 16$;
- uses *negacyclic look-up tables*: $\forall X \in \mathbb{Z}_{16},\ S[X + 2^3] = S[-X]$;
- slightly different structure as compared to classical stream ciphers.

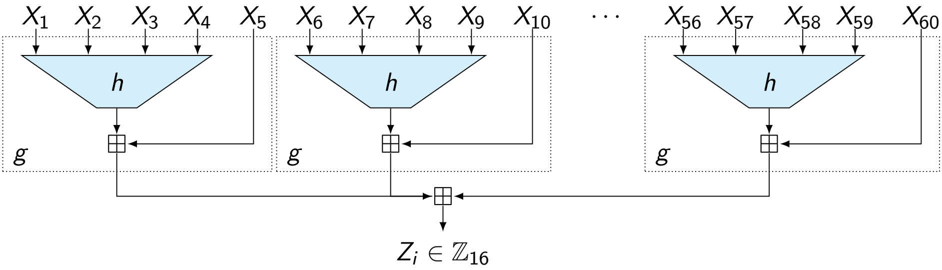


Plan

Elisabeth-4: overall structure



The filtering function f



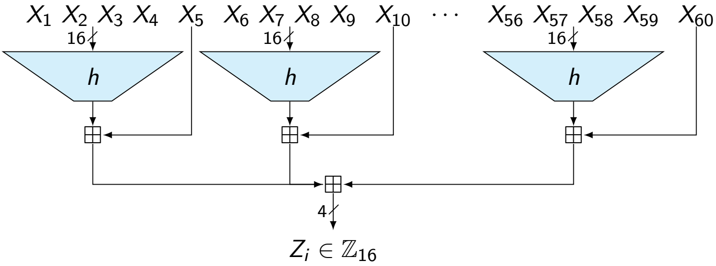
Structure of f

- 12 parallel calls to a 5-to-1 function g.
- $g(X_1, X_2, X_3, X_4, X_5) = h(X_1, X_2, X_3, X_4) + X_5$
- h is non-linear.
  - ingredients: $\boxplus$ and negacyclic look-up tables.

Plan

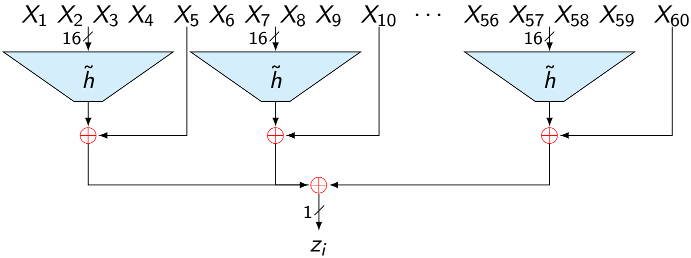
Basic linearisation in $\mathbb{F}_2$

The filtering function f



Basic linearisation in $\mathbb{F}_2$

The filtering function f

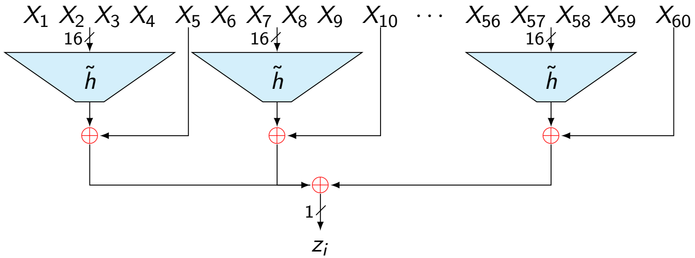


We focus on the **LSB of the output nibble**

$\rightarrow$ On the LSB, the addition in $\mathbb{Z}_{16}$ acts as a XOR

Basic linearisation in $\mathbb{F}_2$

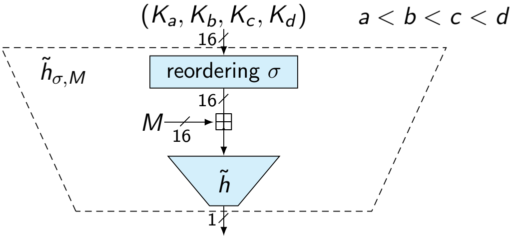
The filtering function f



How many monomials can appear in the ANF of the LSB*?

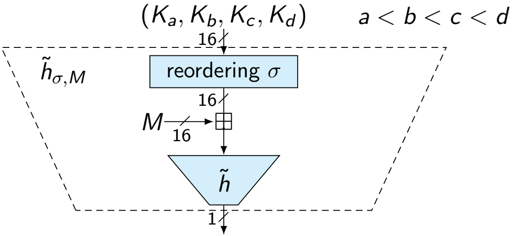
* *regardless of the choice of subset/permutation/whitening!*

Bounding the number of monomials



1. For any 4-tuple $a < b < c < d$ of key register positions, the number of monomials in **all** variations $\tilde{h}_{\sigma,M}$ of h is bounded by 2^{16}.

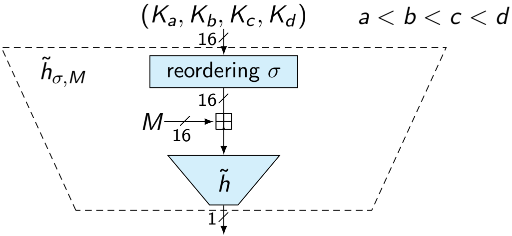
Bounding the number of monomials



1. For any 4-tuple $a < b < c < d$ of key register positions, the number of monomials in **all** variations $\tilde{h}_{\sigma,M}$ of h is bounded by 2^{16}.
2. How many possible choices of (K_a, K_b, K_c, K_d) in the 256-nibble key register?

$$\binom{256}{4}$$

Bounding the number of monomials



1. For any 4-tuple $a < b < c < d$ of key register positions, the number of monomials in **all** variations $\tilde{h}_{\sigma,M}$ of h is bounded by 2^{16}.
2. How many possible choices of (K_a, K_b, K_c, K_d) in the 256-nibble key register?

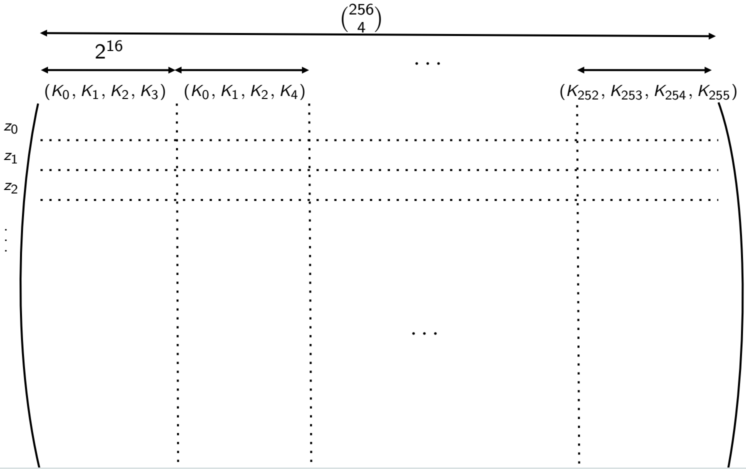
$$\binom{256}{4}$$

Total number of monomials $\leq \mu = \binom{256}{4} 2^{16}$.

Building a linearization matrix

Linearization matrix A

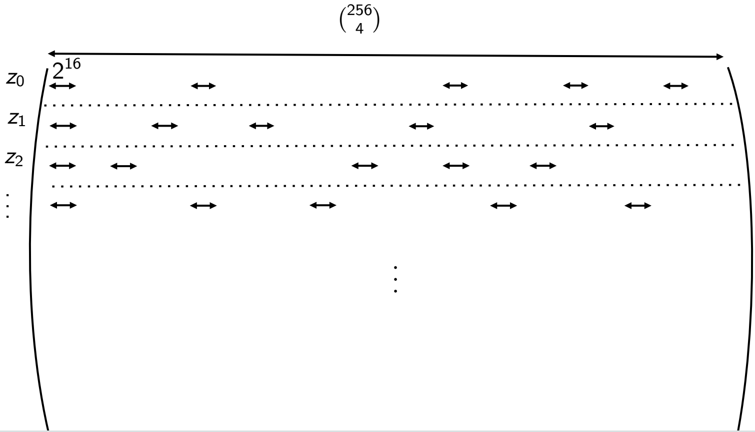
- $\binom{256}{4} 2^{16} \approx 2^{43.4}$ columns.
- Each set of 2^{16} columns corresponds to the monomials of a (K_a, K_b, K_c, K_d), $a < b < c < d$.



Building a linearization matrix

At each iteration of the stream cipher, the XOF outputs

- a **subset** and a **permutation** → selects 12 sets of 2^{16} columns;
- a **permutation** and a **whitening vector** → used to compute the ANF corresponding to this XOF output.



Resulting linearization attack

Basic linearization attack

- After at most $\mu = \binom{256}{4} \cdot 2^{16} \approx 2^{43.4}$ iterations, the linear system is solved in μ^ω operations.
  - Straightforward Gaussian elimination, $\omega = 3$, $T \approx 2^{131}$ operations.
- **Data complexity** is μ nibbles.

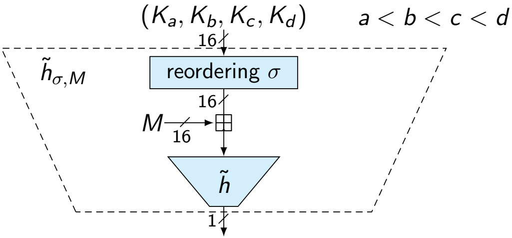
Crucial observation: A is sparse.

$\rightarrow$ at most $s = 12 \cdot 2^{16} \ll \mu$ active bits on each row.

- **Memory complexity**: $s \cdot \mu \approx 2^{63}$ bits.
- Sparse linear algebra: **Coppersmith's Block-Wiedemann algorithm**.
- **Improved time complexity:** $\mu^3 \rightarrow \frac{6}{64} \cdot s \cdot \mu^2$.
- $T \approx 2^{103}$ operations.

Plan

Identification of a rank defect



We pre-computed and stored the ANF of $2^{16} \cdot 4!$ *variations* $\tilde{h}_{\sigma,M}$ of h constructed by

- restricting the output to the LSB;
- considering the $4!$ **possible orderings of the variables**;
- adding the 2^{16} **possible masks**.

We computed the rank and obtained

$$\dim\left(< \tilde{h}_{IV,i} >\right) \leq \dim\left(< \tilde{h}_{M,\sigma} >\right) = \rho = 8705 \ll 2^{16}\,.$$

Explaining the rank defect (theoretically)

Our results

- We prove a theoretical bound $2^{14.01}$, with $\rho = 2^{13.08} < 2^{14.01} \ll 2^{16}$.
- The proof is *partially* elegant.

Explaining the rank defect (theoretically)

Our results

- We prove a theoretical bound $2^{14.01}$, with $\rho = 2^{13.08} < 2^{14.01} \ll 2^{16}$.
- The proof is *partially* elegant.

Some algebraic parts of the proof are not too elegant... but it's all **fun**, check out the paper :)

Explaining the rank defect (theoretically)

Our results

- We prove a theoretical bound $2^{14.01}$, with $\rho = 2^{13.08} < 2^{14.01} \ll 2^{16}$.
- The proof is *partially* elegant.

Some algebraic parts of the proof are not too elegant... but it's all **fun**, check out the paper :)

Our analysis

- The rank defect is caused by **HHE-dedicated features**.
- Interaction between
  - Negacyclic look-up tables;
  - Addition in $\mathbb{Z}_{16}$.

Explaining the rank defect (theoretically)

Our results

- We prove a theoretical bound $2^{14.01}$, with $\rho = 2^{13.08} < 2^{14.01} \ll 2^{16}$.
- The proof is *partially* elegant.

Some algebraic parts of the proof are not too elegant... but it's all **fun**, check out the paper :)

Our analysis

- The rank defect is caused by **HHE-dedicated features**.
- Interaction between
  - Negacyclic look-up tables;
  - Addition in $\mathbb{Z}_{16}$.

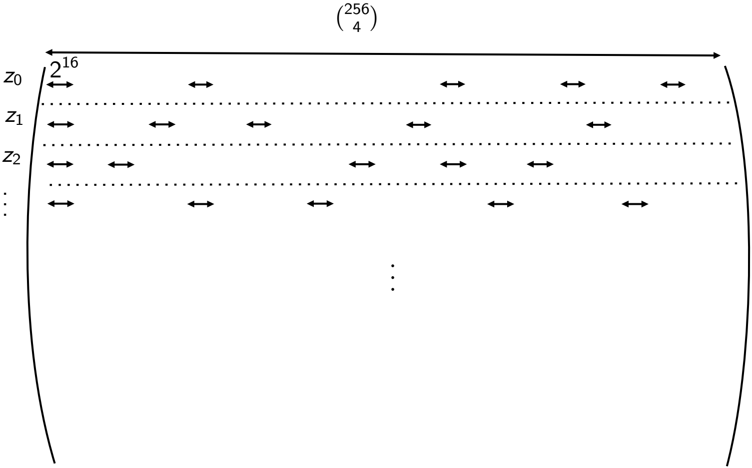
We also identify and *fully* prove a **degree** defect:

$$\text{For any } IV, i, \;\; \deg\left(\tilde{h}_{IV,i}\right) \leq 12 < 16\,.$$

.

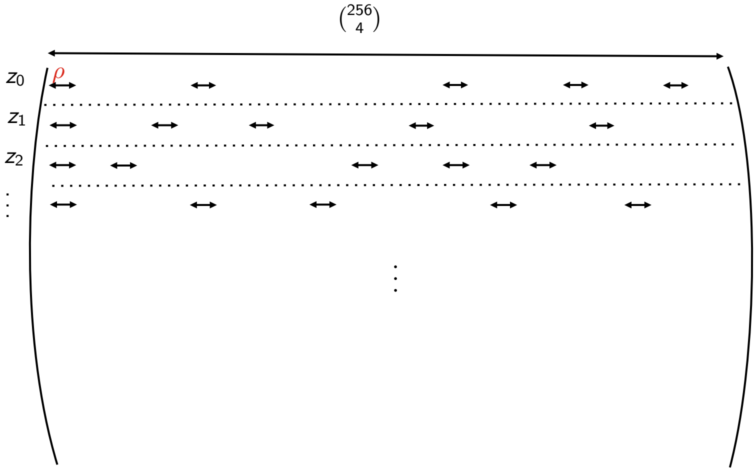
Exploiting the rank defect

Writing each ANF in the basis of size ρ...



Exploiting the rank defect

- $\mathbf{A}$ has now only $\mu' = \binom{256}{4}\rho$ columns
- Each row has at most $s' = 12 \cdot \rho$ active bits.



Improved attack

- **Time complexity:** $\frac{6}{64} \cdot s \cdot \mu^2$
- **Data complexity:** μ
- **Memory complexity:** $s \cdot \mu$

Improved attack

- **Time complexity:** $\frac{6}{64} \cdot s \cdot \mu^2 \rightarrow \frac{6}{64} \cdot s' \cdot (\mu')^2 \approx 2^{94}$ operations.
- **Data complexity:** μ
- **Memory complexity:** $s \cdot \mu$

Improved attack

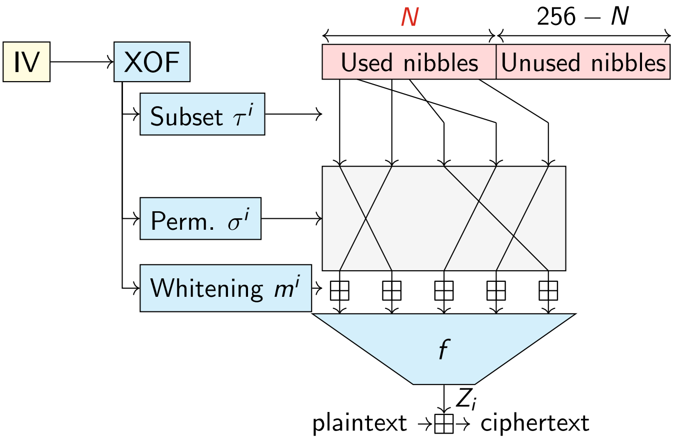
- **Time complexity:** $\frac{6}{64} \cdot s \cdot \mu^2 \rightarrow \frac{6}{64} \cdot s' \cdot (\mu')^2 \approx 2^{94}$ operations.
- **Data complexity:** $\mu \rightarrow \mu' = 2^{41}$ nibbles.
- **Memory complexity:** $s \cdot \mu$

Improved attack

- **Time complexity:** $\frac{6}{64} \cdot s \cdot \mu^2 \rightarrow \frac{6}{64} \cdot s' \cdot (\mu')^2 \approx 2^{94}$ operations.
- **Data complexity:** $\mu \rightarrow \mu' = 2^{41}$ nibbles.
- **Memory complexity:** $s \cdot \mu \rightarrow s' \cdot \mu' = 2^{57}$ bits.

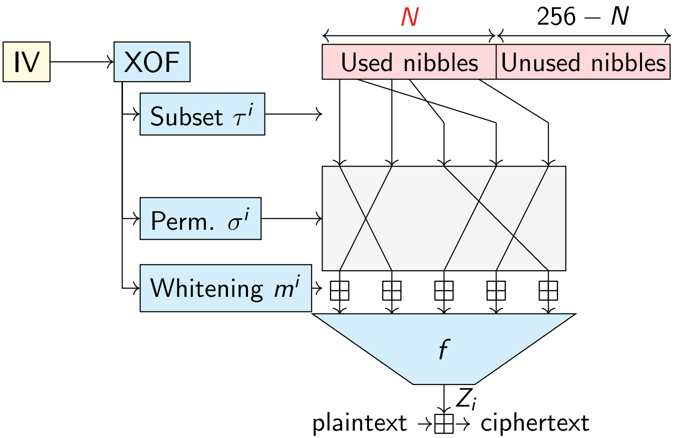
Plan

Considering only convenient XOF outputs



Total number of monomials: $\binom{\cancel{256}}{4}\rho$. .

Considering only convenient XOF outputs



- **Total number of monomials:** $\mu_N = \binom{N}{4}\rho$.

Improved attack

The nibbles are all selected in a subset of size N with probability $p_N \approx \binom{N}{48} / \binom{256}{48} \rightarrow$ **data complexity**: μ_N / p_N nibbles.

Trade-off: $N = 137$.

Known-IV attack

- **Data complexity**: $\mu' = 2^{41} \rightarrow \mu_N / p_N = 2^{87}$ nibbles.
- **Time complexity**: $\frac{6}{64} \cdot s' \cdot (\mu')^2 = 2^{94} \rightarrow \frac{12}{64} \cdot s' \cdot (\mu_N)^2 = 2^{88}$ operations.
- **Memory complexity**: $s' \cdot \mu' = 2^{57} \rightarrow s' \cdot \mu_N = 2^{54}$ bits.

Improved attack

The nibbles are all selected in a subset of size N with probability $p_N \approx \binom{N}{48} / \binom{256}{48} \rightarrow$ **data complexity**: μ_N / p_N nibbles.

Trade-off: $N = 137$.

Known-IV attack

- **Data complexity**: $\mu' = 2^{41} \rightarrow \mu_N / p_N = 2^{87}$ nibbles.
- **Time complexity**: $\frac{6}{64} \cdot s' \cdot (\mu')^2 = 2^{94} \rightarrow \frac{12}{64} \cdot s' \cdot (\mu_N)^2 = 2^{88}$ operations.
- **Memory complexity**: $s' \cdot \mu' = 2^{57} \rightarrow s' \cdot \mu_N = 2^{54}$ bits.

Improved attack

The nibbles are all selected in a subset of size N with probability $p_N \approx \binom{N}{48} / \binom{256}{48} \to$ **data complexity**: μ_N / p_N nibbles.

Trade-off: $N = 137$.

Known-IV attack

- **Data complexity**: $\mu' = 2^{41} \to \mu_N / p_N = 2^{87}$ nibbles.
- **Time complexity**: $\frac{6}{64} \cdot s' \cdot (\mu')^2 = 2^{94} \to \frac{12}{64} \cdot s' \cdot (\mu_N)^2 = 2^{88}$ operations.
- **Memory complexity**: $s' \cdot \mu' = 2^{57} \to s' \cdot \mu_N = 2^{54}$ bits.

Chosen-IV attack

- Pre-compute convenient IVs, then query these IVs only.
- **Improved data complexity**: 2^{37} nibbles.

Small-scale experiments

`https://github.com/jj-anssi/asiacrypt2023-cryptanalysis-elisabeth4`

Toy `Elisabeth-4`

- Operates on $\mathbb{Z}_8$ rather than $\mathbb{Z}_{16}$.
- **Subset** selects 10 key nibbles among 32.
- Still has a rank defect, with $\rho = 254 \ll 2^{12}$.

Implemented attack

- Two main things we checked:
  - Block-Wiedemann allows to solve an `Elisabeth-4` type linear system.
  - Solving the system allows to recover the key.
- BW implem. from $\mathrm{CADO\text{-}NFS}$ project for integer factorization.
- With these parameters, the attack required 44 hours.

Thank you for your attention :)

Questions?