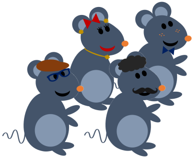
Algebraic Attacks against Some Arithmetization-Oriented Primitives

Clémence Bouvier [1,2]

joint work with Augustin Bariant[2], Gaëtan Leurent[2], Léo Perrin[2]

[1]Sorbonne Université, [2]Inria Paris



FSE, March, 2023

AOP

AOP: “Appellation d’origine protégée”

Camembert de Normandie

Motivation

A Cryptanalysis Challenge for ZK-friendly Hash Functions!
In November 2021, by the Ethereum Fundation.

Category	Parameters	Security Level	Bounty
~~Easy~~	~~$N = 4, m = 3$~~	~~25~~	~~\$2,000~~
Easy	$N = 6, m = 2$	25	\$4,000
Medium	$N = 7, m = 2$	29	\$6,000
Hard	$N = 5, m = 3$	30	\$12,000
Hard	$N = 8, m = 2$	33	\$26,000

(a) *Rescue–Prime*

Category	Parameters	Security Level	Bounty
~~Easy~~	~~$r = 6$~~	~~9~~	~~\$2,000~~
~~Easy~~	~~$r = 10$~~	~~15~~	~~\$4,000~~
~~Medium~~	~~$r = 14$~~	~~22~~	~~\$6,000~~
~~Hard~~	~~$r = 18$~~	~~28~~	\$12,000
~~Hard~~	~~$r = 22$~~	~~34~~	\$26,000

(b) *Feistel–MiMC*

Category	Parameters	Security Level	Bounty
~~Easy~~	~~RP = 3~~	~~8~~	~~\$2,000~~
~~Easy~~	~~RP = 8~~	~~16~~	~~\$4,000~~
~~Medium~~	~~RP = 13~~	~~24~~	~~\$6,000~~
Hard	RP = 19	32	\$12,000
Hard	RP = 24	40	\$26,000

(c) Poseidon

Category	Parameters	Security Level	Bounty
Easy	$p = 281474976710597$	24	\$4,000
Medium	$p = 72057594037926839$	28	\$6,000
Hard	$p = 18446744073709551557$	32	\$12,000

(d) `Reinforced Concrete`

Motivation

A Cryptanalysis Challenge for ZK-friendly Hash Functions!
In November 2021, by the Ethereum Fundation.

Category	Parameters	Security Level	Bounty
~~Easy~~	~~$N = 4, m = 3$~~	~~25~~	~~$2, 000~~
Easy	$N = 6, m = 2$	25	$4, 000
Medium	$N = 7, m = 2$	29	$6, 000
Hard	$N = 5, m = 3$	30	$12, 000
Hard	$N = 8, m = 2$	33	$26, 000

(a) *Rescue–Prime*

Category	Parameters	Security Level	Bounty
~~Easy~~	~~$r = 6$~~	~~9~~	~~$2, 000~~
~~Easy~~	~~$r = 10$~~	~~15~~	~~$4, 000~~
~~Medium~~	~~$r = 14$~~	~~22~~	~~$6, 000~~
~~Hard~~	~~$r = 18$~~	~~28~~	$12, 000
~~Hard~~	~~$r = 22$~~	~~34~~	$26, 000

(b) *Feistel–MiMC*

Category	Parameters	Security Level	Bounty
~~Easy~~	~~RP = 3~~	~~8~~	~~$2, 000~~
~~Easy~~	~~RP = 8~~	~~16~~	~~$4, 000~~
~~Medium~~	~~RP = 13~~	~~24~~	~~$6, 000~~
Hard	RP = 19	32	$12, 000
Hard	RP = 24	40	$26, 000

(c) POSEIDON

Category	Parameters	Security Level	Bounty
Easy	$p = 281474976710597$	24	$4, 000
Medium	$p = 72057594037926839$	28	$6, 000
Hard	$p = 18446744073709551557$	32	$12, 000

(d) Reinforced Concrete

Content

Algebraic Attacks against Some Arithmetization-Oriented Primitives.

1. Preliminaries
   - Arithmetization-Oriented Primitives
   - CICO Problem

2. Solving Systems
   - Univariate Systems
   - Multivariate Systems

3. Trick for SPN
   - Applied to POSEIDON
   - Applied to Rescue–Prime

4. CIMINION

Comparison with "usual" case

A new environment

"Usual" case

- Field size:
  $\mathbb{F}_{2^n}$, with $n \simeq 4, 8$ (AES: $n = 8$).
- Operations:
  logical gates/CPU instructions

Arithmetization-friendly

- Field size:
  $\mathbb{F}_q$, with $q \in \{2^n, p\}, p \simeq 2^n$, $n \geq 64$
- Operations:
  large finite-field arithmetic

Comparison with "usual" case

A new environment

"Usual" case

- Field size:
  $\mathbb{F}_{2^n}$, with $n \simeq 4, 8$ (AES: $n = 8$).
- Operations:
  logical gates/CPU instructions

Arithmetization-friendly

- Field size:
  $\mathbb{F}_q$, with $q \in \{2^n, p\}, p \simeq 2^n$, $n \geq 64$
- Operations:
  large finite-field arithmetic

$\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$, with p given by the order of some elliptic curves

Examples:

- Curve BLS12-381 $\qquad \log_2 p = 255$

  $p = 52435875175126190479447740508185965837690552500527637822603658699938581184513$

- Curve BLS12-377 $\qquad \log_2 p = 253$

  $p = 84444617494283704242488249387815465313758993351540638279352334559174092390 41$

Comparison with “usual” case

A new environment

“Usual” case

- Field size:
  $\mathbb{F}_{2^n}$, with $n \simeq 4, 8$ (AES: $n = 8$).
- Operations:
  logical gates/CPU instructions

Arithmetization-friendly

- Field size:
  $\mathbb{F}_q$, with $q \in \{2^n, p\}, p \simeq 2^n,\ n \geq 64$
- Operations:
  large finite-field arithmetic

New properties

“Usual” case

$$y \leftarrow E(x)$$

- Optimized for:
  implementation in software/hardware

Arithmetization-friendly

$$y \leftarrow E(x) \quad \text{and} \quad y == E(x)$$

- Optimized for:
  integration within advanced protocols

Comparison with "usual" case

A new environment

"Usual" case

- Field size:
  $\mathbb{F}_{2^n}$, with $n \simeq 4, 8$ (AES: $n = 8$).
- Operations:
  logical gates/CPU in

Arithmetization-friendly

- Field size:
  $\mathbb{F}_q$, with $q \in \{2^n, p\}, p \simeq 2^n$, $n \geq 64$.
- Operations:
  large finite-field arithme

Decades of Cryptanalysis

$\leq$ 5 years of Cryptanalysis

N

"U ual" case

$$y \leftarrow E(x)$$

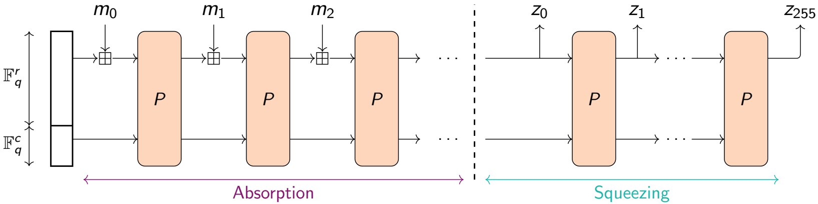
- Optimized for:
  implementation in software/hardware

Arit etization-friendly

$$y \leftarrow E(x) \quad \text{and} \quad y == E(x)$$

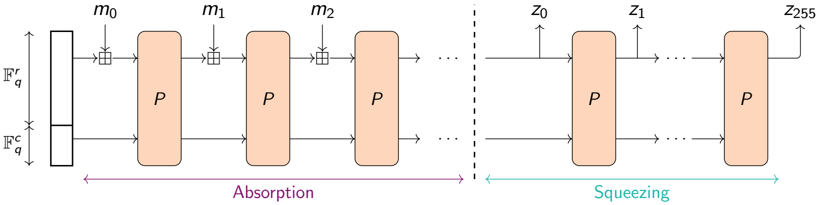
- Optimized for:
  integration within advanced protocols

CICO Problem



Sponge construction.

CICO Problem



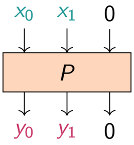
Sponge construction.

CICO: Constrained Input Constrained Output

Definition

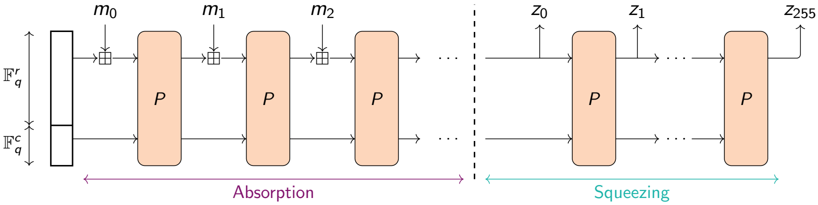
Let $F : \mathbb{F}_q^t \rightarrow \mathbb{F}_q^t$ and $u < t$. The **CICO** problem is:

Finding $X, Y \in \mathbb{F}_q^{t-u}$ s.t. $P(X, 0^u) = (Y, 0^u)$.



when $t = 3$, $u = 1$.

CICO Problem



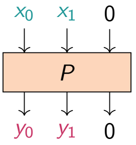
Sponge construction.

CICO: Constrained Input Constrained Output

Definition

Let $F : \mathbb{F}_q^t \to \mathbb{F}_q^t$ and $u < t$. The **CICO** problem is:

Finding $X, Y \in \mathbb{F}_q^{t-u}$ s.t. $P(X, 0^u) = (Y, 0^u)$.



when $t = 3$, $u = 1$.

Ethereum Challenges: solving CICO problem for AO primitives with $q \sim 2^{64}$ prime

Univariate Solving

Find the **roots** of a polynomial $P \in \mathbb{F}_q[X]$, with $\deg P = d$.

Steps:

1. Compute $Q = X^q - X \bmod P$.
   using a double-and-add algorithm.
2. Compute $R = gcd(P, Q)$.
   $roots(P) = roots(R)$ in $\mathbb{F}_q$
3. Factor R.
   $\deg(R) \simeq 1$ or 2 for random P

Cost (in theory):

$\mathcal{O}(d \log(q) \log(d) \log(\log(d)))$

$O(d \log^2(d) \log(\log(d)))$

negligible.

$$\boxed{\mathcal{O}\left(d \cdot \log(d) \cdot (\log(d) + \log(q)) \cdot \log(\log(d)) \right)}$$

Univariate Solving

Find the **roots** of a polynomial $P \in \mathbb{F}_q[X]$, with $\deg P = d$.

Steps:

1. Compute $Q = X^q - X \bmod P$.
   using a double-and-add algorithm.
2. Compute $R = gcd(P, Q)$.
   $roots(P) = roots(R)$ in $\mathbb{F}_q$
3. Factor R.
   $\deg(R) \simeq 1$ or 2 for random P

Cost (in practice):

Degree d	3^{11}	3^{15}	3^{18}
Step 1.	14s	1,433s	47,964s
Step 2.	7s	903s	38,693s

for random systems

$$\boxed{\mathcal{O}\left(d \cdot \log(d) \cdot (\log(d) + \log(q)) \cdot \log(\log(d)) \right)}$$

Multivariate Solving

Compute a **Gröbner Basis (GB)** from polynomial equations in $\mathbb{F}_q[X_1, \dots X_n]$:

$$\begin{cases} P_{j, j=1,\dots n}(X_1, \dots X_n) = 0, \end{cases} \qquad D_{\text{reg}} \leq 1 + \sum_{i=1}^{n} (d_i - 1), \qquad d \leq \prod_{i=1}^{n} d_i$$

Steps:

1. **F5** algorithm
   Compute a grevlex order GB.
2. **FGLM** algorithm
   Convert it into lex order GB.
3. Find the roots in $\mathbb{F}_q^n$ of the GB polynomials
   using univariate system resolution.

Cost (in theory):

$$\mathcal{O}\left(nD_{\text{reg}} \times \binom{n + D_{\text{reg}} - 1}{D_{\text{reg}}}^{\omega} \right), \text{ with } 2 \leq \omega \leq 3$$

for regular systems

$$\mathcal{O}(nd^3) \quad \text{or } \mathcal{O}(nd^{\omega})$$

$$\mathcal{O}(d \log^2(d))$$

Multivariate Solving

Compute a **Gröbner Basis (GB)** from polynomial equations in $\mathbb{F}_q[X_1, \ldots X_n]$:

$$\begin{cases} P_{j,\, j=1,\ldots n}(X_1, \ldots X_n) = 0, \end{cases} \qquad D_{\text{reg}} \leq 1 + \sum_{i=1}^{n} (d_i - 1), \qquad d \leq \prod_{i=1}^{n} d_i$$

Steps:

1. **F5** algorithm
   Compute a grevlex order GB.
2. **FGLM** algorithm
   Convert it into lex order GB.
3. Find the roots in $\mathbb{F}_q^n$ of the GB polynomials
   using univariate system resolution.

In practice:

Degree d	1024	4608	16384
F4	2.36s	92.9s	3,030s
FGLM	18.96s	1,011s	32,069s

for random systems
with 4 equations on 4 variables

Multivariate Solving

Compute a **Gröbner Basis (GB)** from polynomial equations in $\mathbb{F}_q[X_1, \dots X_n]$:

$$\begin{cases} P_{j,\, j=1,\dots n}(X_1, \dots X_n) = 0, \end{cases} \qquad D_{\text{reg}} \leq 1 + \sum_{i=1}^{n} (d_i - 1), \qquad d \leq \prod_{i=1}^{n} d_i$$

Steps:

1. **F5** algorithm
   Compute a grevlex order GB.
2. **FGLM** algorithm
   Convert it into lex order GB.
3. Find the roots in $\mathbb{F}_q^n$ of the GB polynomials
   using univariate system resolution.

In practice:

Degree d	1024	4608	16384
F4	2.36s	92.9s	3,030s
FGLM	18.96s	1,011s	32,069s

for random systems
with 4 equations on 4 variables

Take Away

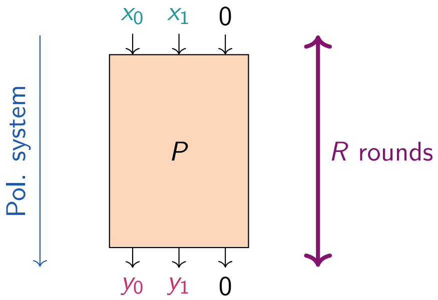
Build **univariate** $\widetilde{\mathcal{O}}(d)$ **instead of multivariate** $\widetilde{\mathcal{O}}(d^3)$ systems when possible!

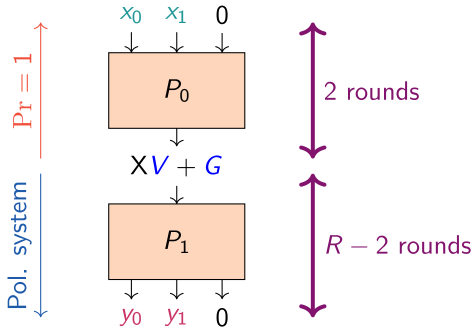
1. Preliminaries
   - Arithmetization-Oriented Primitives
   - CICO Problem
2. Solving Systems
   - Univariate Systems
   - Multivariate Systems
3. Trick for SPN
   - Applied to Poseidon
   - Applied to Rescue–Prime
4. Ciminion

Trick for SPN

Let $P = P_0 \circ P_1$ be a permutation of $\mathbb{F}_p^3$ and suppose

$$\exists\ V, G \in \mathbb{F}_p^3, \quad \text{s.t. } \forall\ \mathrm{X} \in \mathbb{F}_p, \quad P_0^{-1}(\mathrm{X}V + G) = (*, *, 0)\ .$$





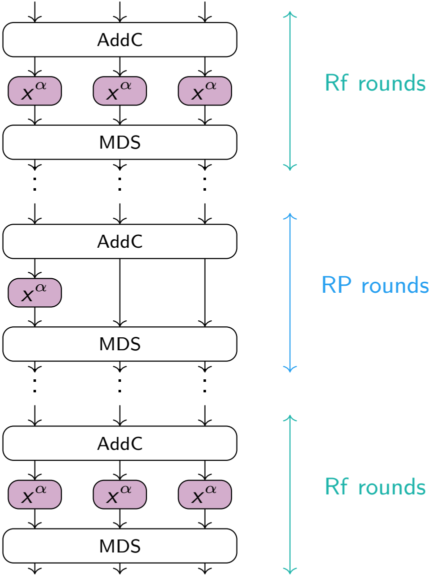
Approach used against POSEIDON and Rescue–Prime

Poseidon

L. Grassi, D. Khovratovich, C. Rechberger, A. Roy and M. Schofnegger, *USENIX 2021*

- SPN construction:
  - S-Box layer: $x \mapsto x^{\alpha}$, $(\alpha = 3)$
  - Linear layer: MDS
  - Round constants addition: AddC
- Number of rounds (for challenges):

$$\begin{aligned} R &= 2 \times \mathrm{Rf} + \mathrm{RP} \\ &= 8 + (\text{from } 3 \text{ to } 24)\ . \end{aligned}$$



POSEIDON

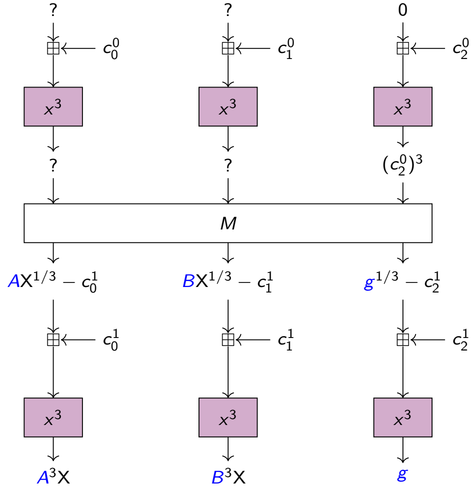
$$\begin{cases} V &= (A^3, B^3, 0) \ , \\ G &= (0, 0, g) \ , \end{cases}$$

with

$$\begin{cases} B &= -\frac{\alpha_{0,2}}{\alpha_{1,2}} A \\ g &= \left(\frac{1}{\alpha_{2,2}} \left(\alpha_{0,2} c_0^1 + \alpha_{1,2} c_1^1 \right) + c_2^1 + (c_2^0)^3 \right)^3 \ . \end{cases}$$

R	Designers claims	Ethereum estimations	d	complexity
$8+3$	2^{17}	2^{45}	3^9	2^{26}
$8+8$	2^{25}	2^{53}	3^{14}	2^{35}
$8+13$	2^{33}	2^{61}	3^{19}	2^{44}
$8+19$	2^{42}	2^{69}	3^{25}	2^{54}
$8+24$	2^{50}	2^{77}	3^{30}	2^{62}

Complexity of our attack against POSEIDON*.*



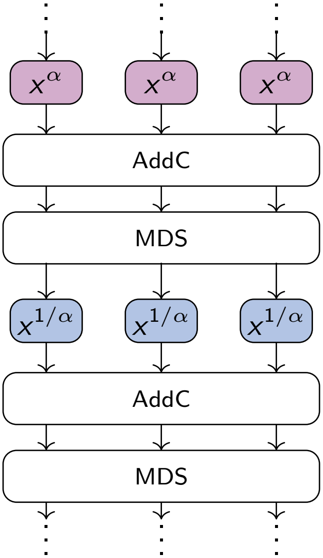
Rescue–Prime

A. Aly, T. Ashur, E. Ben-Sasson, S. Dhooghe and A. Szepieniec, *ToSC 2020*

- ⋆ SPN construction:
  - ⋆ S-Box layer: $x \mapsto x^\alpha$ and $x \mapsto x^{1/\alpha}$, $(\alpha = 3)$
  - ⋆ Linear layer: MDS
  - ⋆ Round constants addition: AddC
- ⋆ Number of rounds (for challenges):

$$R = \text{from 4 to 8}$$

(2 S-boxes per round).



Rescue–Prime

A. Aly, T. Ashur, E. Ben-Sasson, S. Dhooghe and A. Szepieniec, *ToSC 2020*

- SPN construction:
  - S-Box layer: $x \mapsto x^\alpha$ and $x \mapsto x^{1/\alpha}$, $(\alpha = 3)$
  - Linear layer: MDS
  - Round constants addition: AddC
- Number of rounds (for challenges):

$$R = \text{from 4 to 8}$$

(2 S-boxes per round).

Example of parameters

$$p = 18446744073709551557$$
$$\simeq 2^{64}$$

$$\alpha = 3$$
$$\alpha^{-1} = 12297829382473034371$$

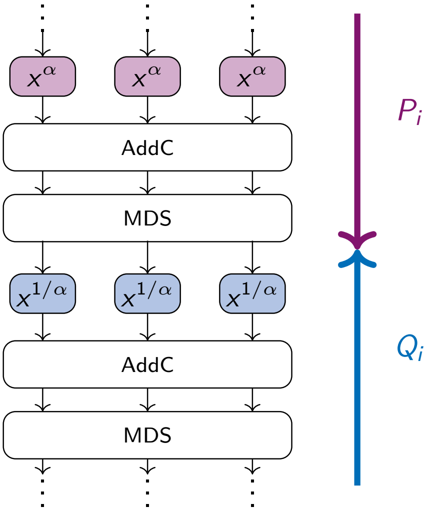
Rescue–Prime

A. Aly, T. Ashur, E. Ben-Sasson, S. Dhooghe and A. Szepieniec, *ToSC 2020*

- SPN construction:
  - S-Box layer: $x \mapsto x^\alpha$ and $x \mapsto x^{1/\alpha}$, $(\alpha = 3)$
  - Linear layer: MDS
  - Round constants addition: AddC
- Number of rounds (for challenges):

$$R = \text{from 4 to 8}$$

(2 S-boxes per round).



Rescue–Prime

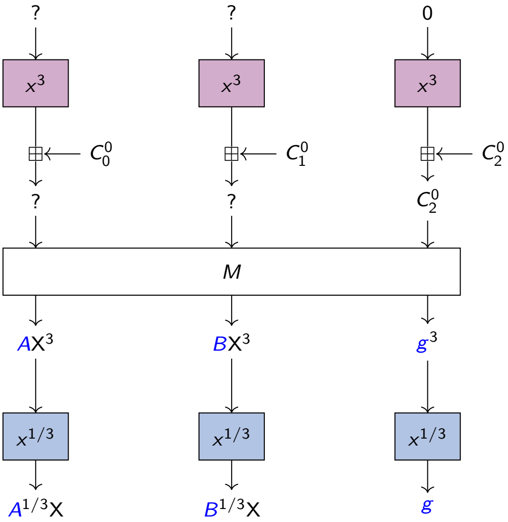
$$\begin{cases} V &= (A^3, B^3, 0) \ , \\ G &= (0, 0, g) \ , \end{cases}$$

with

$$\begin{cases} B &= -\frac{\alpha_{0,2}}{\alpha_{1,2}} A \\ g &= \left(\frac{1}{\alpha_{2,2}} \left(\alpha_{0,2} c_0^0 + \alpha_{1,2} c_1^0 \right) + c_2^0 \right)^{1/3} \ . \end{cases}$$

R	m	Designers claims	Ethereum estimations	d	complexity
4	3	2^{36}	$2^{37.5}$	3^9	2^{43}
6	2	2^{40}	$2^{37.5}$	3^{11}	2^{53}
7	2	2^{48}	$2^{43.5}$	3^{13}	2^{62}
5	3	2^{48}	2^{45}	3^{12}	2^{57}
8	2	2^{56}	$2^{49.5}$	3^{15}	2^{72}

Complexity of our attack against Rescue.

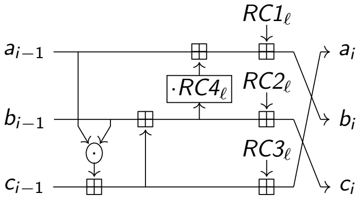


Ciminion

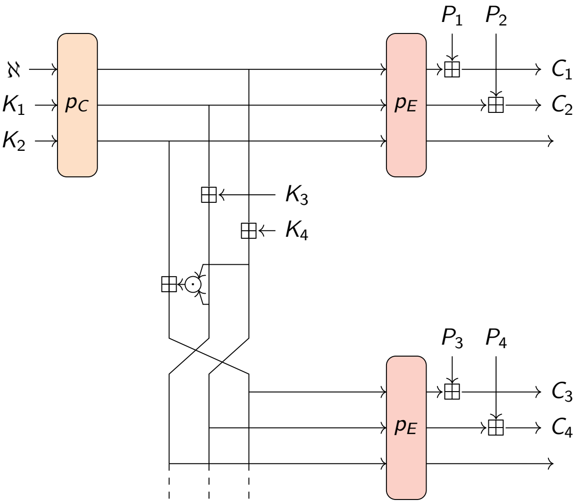
C. Dobraunig, L. Grassi, A. Guinet and D. Kuijster, *EUROCRYPT 2021*

Construction: **Toffoli gates**

$$(a, b, c) \mapsto (a, b, c + ab)$$



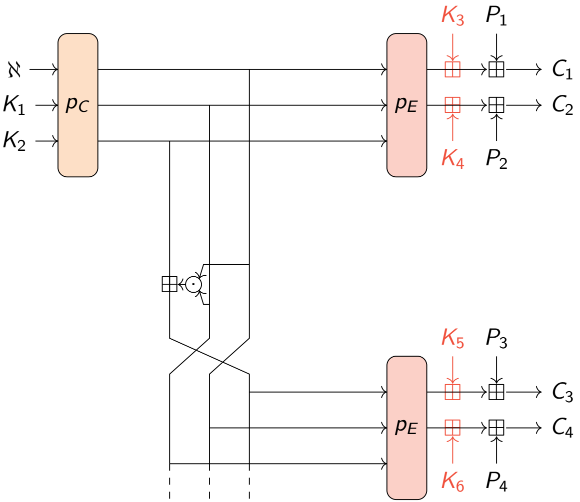
Round function.



Overview of Ciminion *in* $\mathbb{F}_p$.

Attack on Ciminion

- Designers' system:
  - 6 equations ...
  - over 6 variables ...
  - of degrees $\{2^{R-1}, 2^{R}, 2^{R}, 2^{R+1}, 2^{R+1}, 2^{R+2}\}$



Weaker Scheme.

Attack on Ciminion

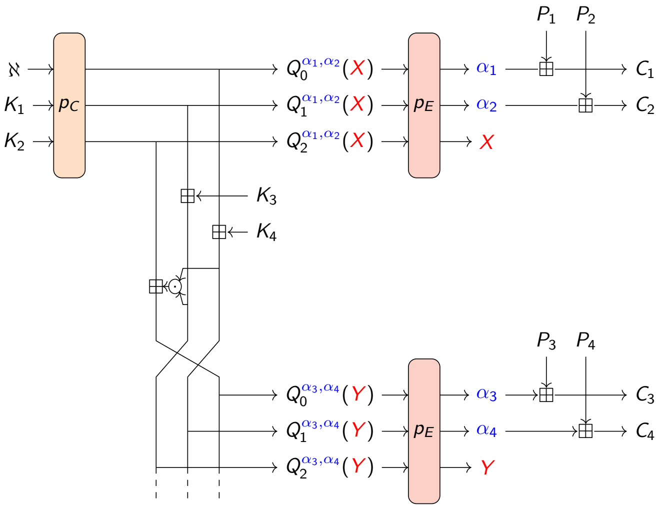
- Designers' system:
  - 6 equations ...
  - over 6 variables ...
  - of degrees $\{2^{R-1}, 2^R, 2^R, 2^{R+1}, 2^{R+1}, 2^{R+2}\}$

$$\Downarrow$$

- Our system
  - 4 equations ...
  - over 4 variables ...
  - of degrees $\{2^{R-1}, 2^R, 3 \cdot 2^{R-1}, 3 \cdot 2^{R-1}\}$

Attack in roughly

$$2^{112.4}$$



Original scheme.

Conclusions

Some suggestions for designers:

- ⋆ consider as many variants of encoding as possible
- ⋆ build univariate instead of multivariate systems when possible
- ⋆ start (and end) with a linear layer
- ⋆ 2 rounds can be skipped with the trick

Conclusions

Some suggestions for designers:

- ⋆ consider as many variants of encoding as possible
- ⋆ build univariate instead of multivariate systems when possible
- ⋆ start (and end) with a linear layer
- ⋆ 2 rounds can be skipped with the trick

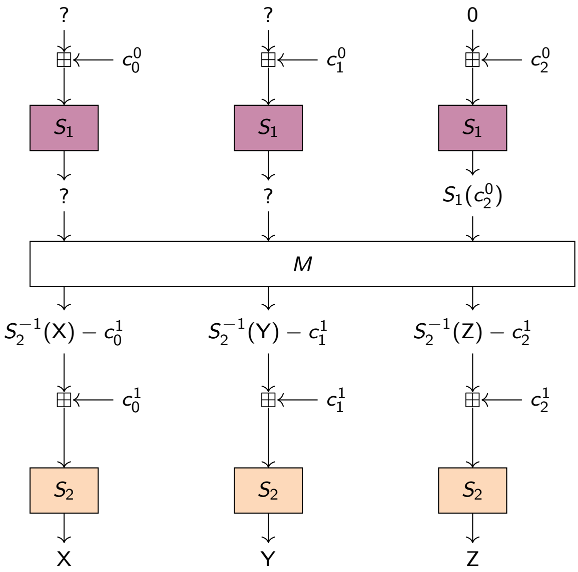
Conclusions

Some suggestions for designers:

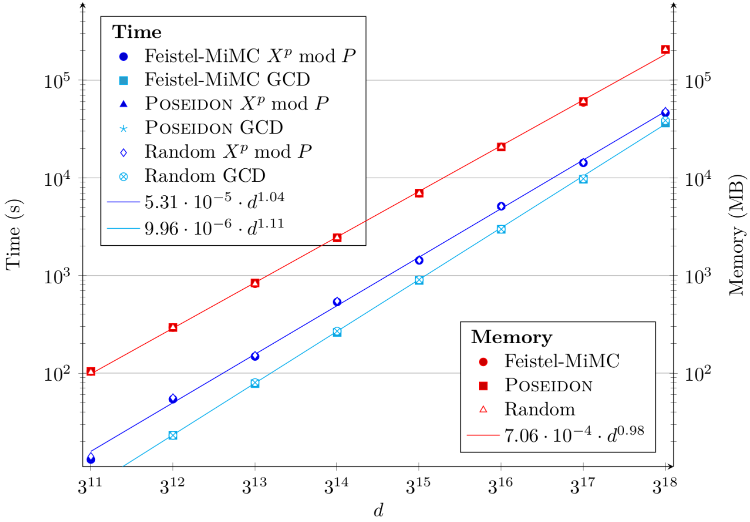
- ⋆ consider as many variants of encoding as possible
- ⋆ build univariate instead of multivariate systems when possible
- ⋆ start (and end) with a linear layer
- ⋆ 2 rounds can be skipped with the trick

Thanks for your attention

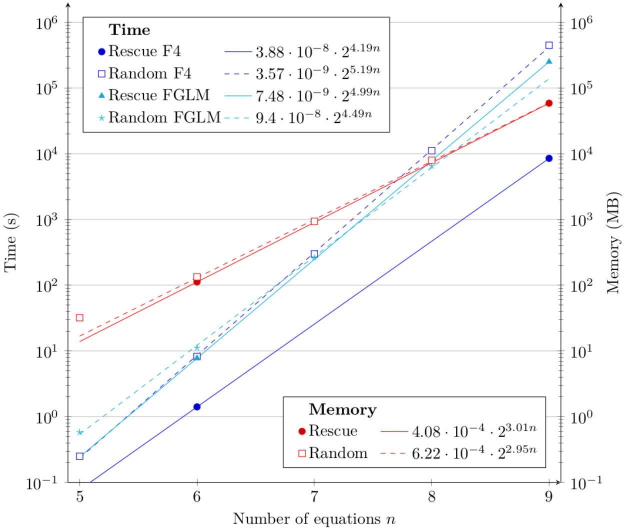
Trick for SPN



Univariate systems: Poseidon, Feistel–MiMC



Multivariate systems: Rescue–Prime



Multivariate systems: CIMINION