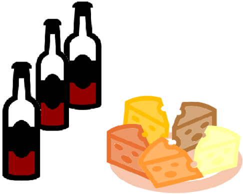
# Trendy Tastings: AOP (Arithmetization-Oriented Primitives)

*Savoring Symmetric Cryptography's Newest Arrivals*

**Clémence Bouvier**



Journées GDR, Rennes
June 10th, 2024

RUHR UNIVERSITÄT BOCHUM RUB

# Toy example of Zero-Knowledge Proof

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 2 | | 5 | | 1 | | 9 | |
| 8 | | | 2 | | 3 | | | 6 |
| | 3 | | | 6 | | | 7 | |
| | | 1 | | | | 6 | | |
| 5 | 4 | | | | | | 1 | 9 |
| | | 2 | | | | 7 | | |
| | 9 | | | 3 | | | 8 | |
| 2 | | | 8 | | 4 | | | 7 |
| | 1 | | 9 | | 7 | | 6 | |

Unsolved Sudoku

# Toy example of Zero-Knowledge Proof

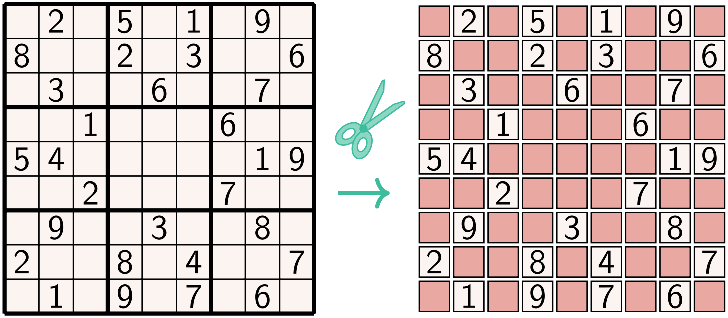
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 2 | | 5 | | 1 | | 9 | |
| 8 | | | 2 | | 3 | | | 6 |
| | 3 | | | 6 | | | 7 | |
| | | 1 | | | | 6 | | |
| 5 | 4 | | | | | | 1 | 9 |
| | | 2 | | | | 7 | | |
| | 9 | | | 3 | | | 8 | |
| 2 | | | 8 | | 4 | | | 7 |
| | 1 | | 9 | | 7 | | 6 | |

Unsolved Sudoku

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 4 | 2 | 6 | 5 | 7 | 1 | 3 | 9 | 8 |
| 8 | 5 | 7 | 2 | 9 | 3 | 1 | 4 | 6 |
| 1 | 3 | 9 | 4 | 6 | 8 | 2 | 7 | 5 |
| 9 | 7 | 1 | 3 | 8 | 5 | 6 | 2 | 4 |
| 5 | 4 | 3 | 7 | 2 | 6 | 8 | 1 | 9 |
| 6 | 8 | 2 | 1 | 4 | 9 | 7 | 5 | 3 |
| 7 | 9 | 4 | 6 | 3 | 2 | 5 | 8 | 1 |
| 2 | 6 | 5 | 8 | 1 | 4 | 9 | 3 | 7 |
| 3 | 1 | 8 | 9 | 5 | 7 | 4 | 6 | 2 |

Solved Sudoku

# Toy example of Zero-Knowledge Proof



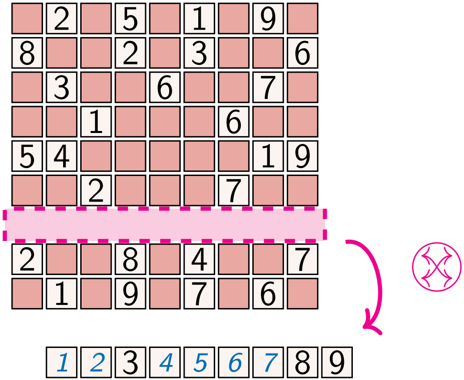

Unsolved Sudoku

Grid cutting

# Toy example of Zero-Knowledge Proof

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 2 | | 5 | | 1 | | 9 | |
| 8 | | | 2 | | 3 | | | 6 |
| | 3 | | | 6 | | | 7 | |
| | | 1 | | | | 6 | | |
| 5 | 4 | | | | | | 1 | 9 |
| | | 2 | | | | 7 | | |
| | 9 | | | 3 | | | 8 | |
| 2 | | | 8 | | 4 | | | 7 |
| | 1 | | 9 | | 7 | | 6 | |

Unsolved Sudoku

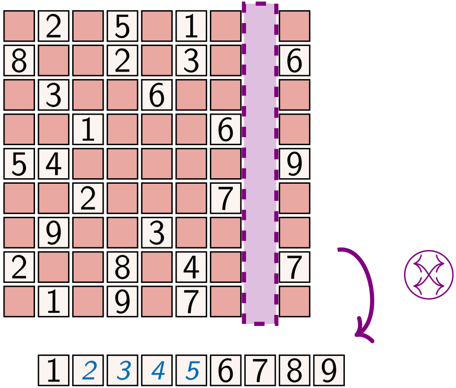



Rows checking

# Toy example of Zero-Knowledge Proof

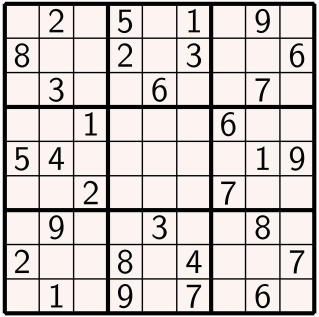
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 2 | | 5 | | 1 | | 9 | |
| 8 | | | 2 | | 3 | | | 6 |
| | 3 | | | 6 | | | 7 | |
| | | 1 | | | | 6 | | |
| 5 | 4 | | | | | | 1 | 9 |
| | | 2 | | | | 7 | | |
| | 9 | | | 3 | | | 8 | |
| 2 | | | 8 | | 4 | | | 7 |
| | 1 | | 9 | | 7 | | 6 | |

Unsolved Sudoku

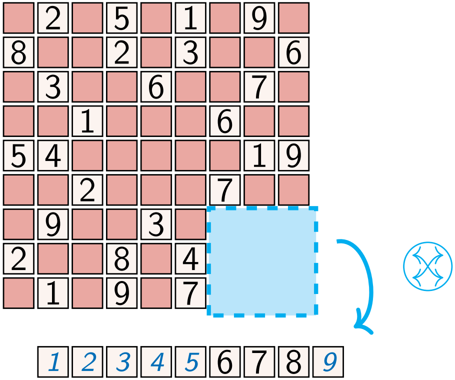



Columns checking

# Toy example of Zero-Knowledge Proof




Unsolved Sudoku

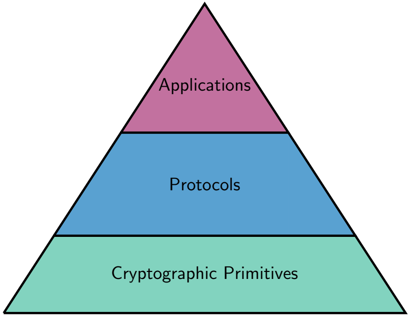



Squares checking

# A need for new primitives

Protocols requiring new primitives:

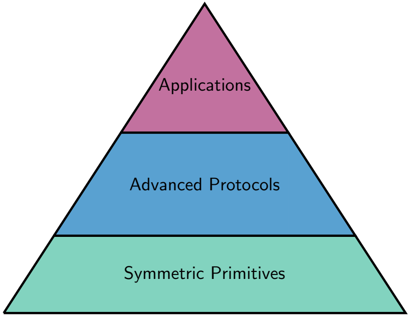
- ⋆ **MPC**: Multiparty Computation
- ⋆ **FHE**: Fully Homomorphic Encryption
- ⋆ **ZK**: Systems of Zero-Knowledge proofs
  Example: SNARKs, STARKs, Bulletproofs



# A need for new primitives

Protocols requiring new primitives:

- ⋆ **MPC**: Multiparty Computation
- ⋆ **FHE**: Fully Homomorphic Encryption
- ⋆ **ZK**: Systems of Zero-Knowledge proofs
  Example: SNARKs, STARKs, Bulletproofs



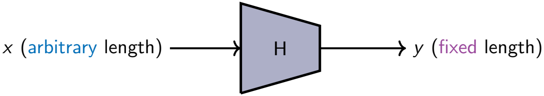

**Problem**: Designing new symmetric primitives

And analyse their security!

# Hash functions

**Definition**

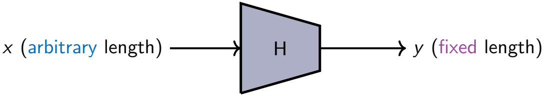
**Hash function:** $H : \mathbb{F}_q^{\ell} \to \mathbb{F}_q^{h}, x \mapsto y = H(x)$ where $\ell$ is arbitrary and $h$ is fixed.



# Hash functions

**Definition**

**Hash function:** $H : \mathbb{F}_q^{\ell} \rightarrow \mathbb{F}_q^{h}, x \mapsto y = H(x)$ where $\ell$ is arbitrary and $h$ is fixed.

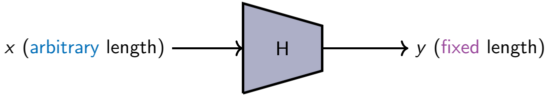



- ⋆ **Preimage resistance**: Given $y$ it must be *infeasible* to find $x$ s.t. $H(x) = y$ .
- ⋆ **Collision resistance**: It must be *infeasible* to find $x \neq x'$ s.t. $H(x) = H(x')$ .

# Hash functions

**Definition**

**Hash function:** $H : \mathbb{F}_q^\ell \rightarrow \mathbb{F}_q^h, x \mapsto y = H(x)$ where $\ell$ is arbitrary and $h$ is fixed.

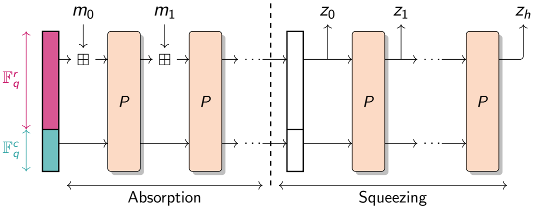



- ⋆ **Preimage resistance**: Given $y$ it must be *infeasible* to find $x$ s.t. $H(x) = y$ .
- ⋆ **Collision resistance**: It must be *infeasible* to find $x \neq x'$ s.t. $H(x) = H(x')$ .

## Sponge construction

Parameters:

- ⋆ rate $r > 0$
- ⋆ capacity $c > 0$
- ⋆ permutation of $\mathbb{F}_q^n$ $(n = r + c)$

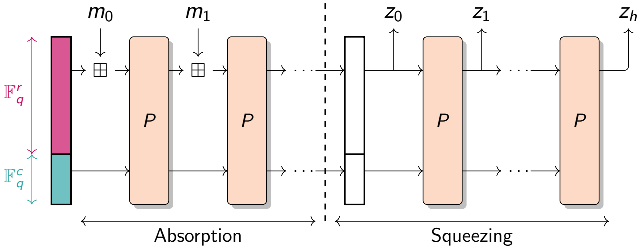


# Sponge construction

## Sponge construction

Parameters:

- ⋆ rate $r > 0$
- ⋆ capacity $c > 0$
- ⋆ permutation of $\mathbb{F}_q^n$ $(n = r + c)$

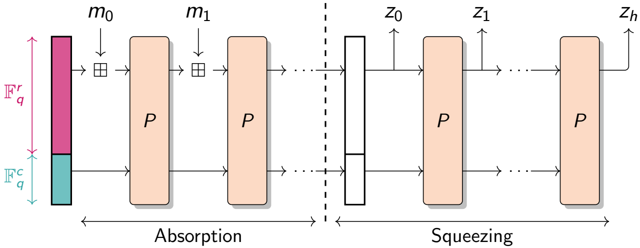


# Sponge construction

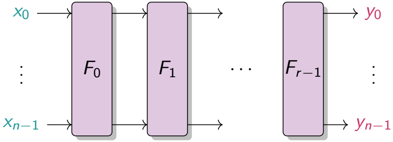
## Sponge construction

Parameters:

- ⋆ rate $r > 0$
- ⋆ capacity $c > 0$
- ⋆ permutation of $\mathbb{F}_q^n$ $(n = r + c)$




## Iterated construction

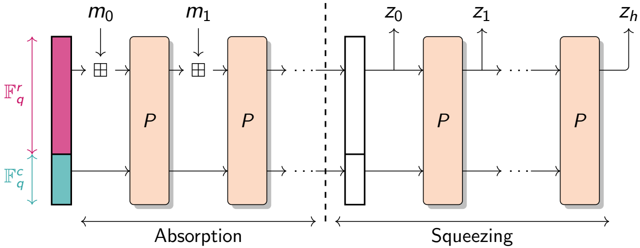


# Sponge construction

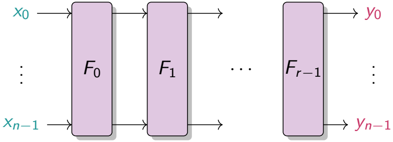
## Sponge construction

Parameters:

- ⋆ rate $r > 0$
- ⋆ capacity $c > 0$
- ⋆ permutation of $\mathbb{F}_q^n$ $(n = r + c)$




## Iterated construction

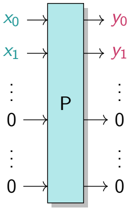



## CICO problem

**Definition**

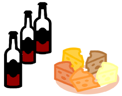
Finding $X, Y \in \mathbb{F}_q^r$ s.t.

$$P(X, 0^c) = (Y, 0^c)$$



# Content

⋆ Introduction of AOP



⋆ An example of AOP: `Anemoi`

⋆ Attacks against AOP

# Primitives to be integrated in advanced protocols

## Traditional case

⋆ Alphabet:
$\mathbb{F}_2^n$, with $n \simeq 4, 8$

Ex: Field of AES: $\mathbb{F}_2^n$ where $n = 8$

## Arithmetization-oriented (AO)

⋆ Alphabet:
$\mathbb{F}_q$, with $q \in \{2^n, p\}, p \simeq 2^n$, $n \geq 64$

Ex: Scalar Field of Curve BLS12-381: $\mathbb{F}_p$ where

$p =$ 0x73eda753299d7d483339d80809a1d805
53bda402fffe5bfeffffffff00000001

# Primitives to be integrated in advanced protocols

## Traditional case

- Alphabet:
  $\mathbb{F}_2^n$, with $n \simeq 4, 8$

  Ex: Field of AES: $\mathbb{F}_2^n$ where $n = 8$

- Operations:
  logical gates/CPU instructions

## Arithmetization-oriented (AO)

- Alphabet:
  $\mathbb{F}_q$, with $q \in \{2^n, p\}, p \simeq 2^n$, $n \geq 64$

  Ex: Scalar Field of Curve BLS12-381: $\mathbb{F}_p$ where

  $$p = \texttt{0x73eda753299d7d483339d80809a1d805} \\ \texttt{53bda402fffe5bfeffffffff00000001}$$

- Operations:
  large finite-field arithmetic

# Primitives to be integrated in advanced protocols

## Traditional case

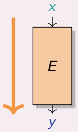
- Alphabet:
  $\mathbb{F}_2^n$, with $n \simeq 4, 8$

  Ex: Field of AES: $\mathbb{F}_2^n$ where $n = 8$

- Operations:
  logical gates/CPU instructions

- Metric:
  minimize time and memory

$$y \leftarrow E(x)$$




## Arithmetization-oriented (AO)

- Alphabet:
  $\mathbb{F}_q$, with $q \in \{2^n, p\}, p \simeq 2^n$, $n \geq 64$

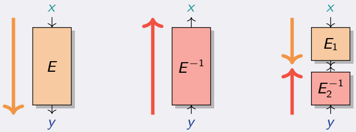
  Ex: Scalar Field of Curve BLS12-381: $\mathbb{F}_p$ where

  $p =$ 0x73eda753299d7d483339d80809a1d805
  53bda402fffe5bfeffffffff00000001

- Operations:
  large finite-field arithmetic

- Metric:
  minimize the number of multiplications

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



# Primitives to be integrated in advanced protocols

## Traditional case

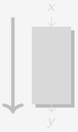
- Alphabet:
  $\mathbb{F}_2^n$, with $n \simeq 4, 8$

  Ex: Field of AES: $\mathbb{F}_2^n$ where $n = 8$

- Operations:
  logical o... ...tions

...imize time and memory

$$y \leftarrow E(x)$$




**Decades of Cryptanalysis**

## Arithmetization-oriented (AO)

- Alphabet:
  $\mathbb{F}_q$, with $q \in \{2^n, p\}, p \simeq 2^n$, $n \geq 64$

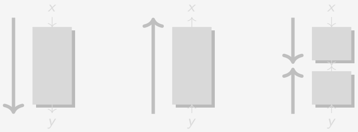
  Ex: Scalar Field of Curve BLS12-381: $\mathbb{F}_p$ where

  $p =$ 0x73eda753299d7d483339d80809a1d805
  53bda402fffe5bfeffffffff00000...

- Operations:
  large finit...

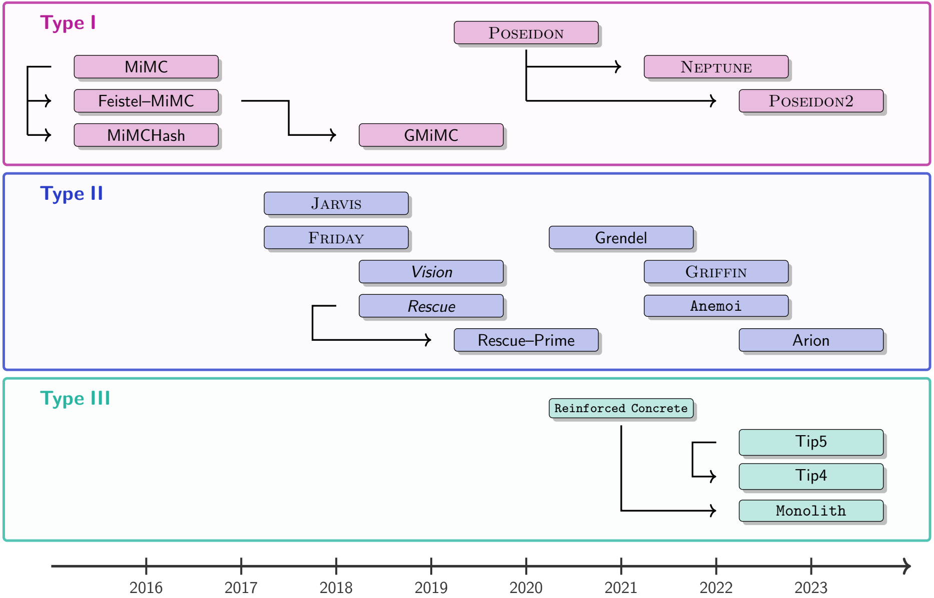
...ze the number of multiplications

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

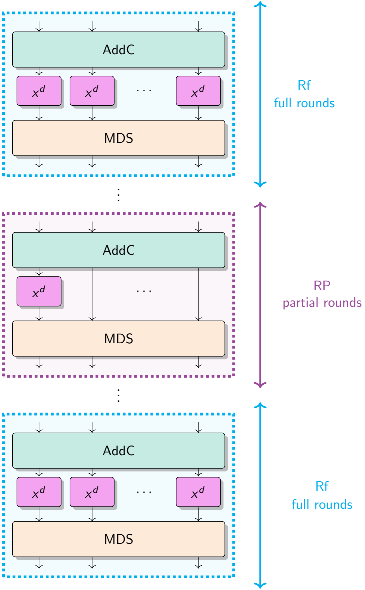



**$\leq$ 5 years of Cryptanalysis**

# Primitives overview



# Example of Type I: Poseidon




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

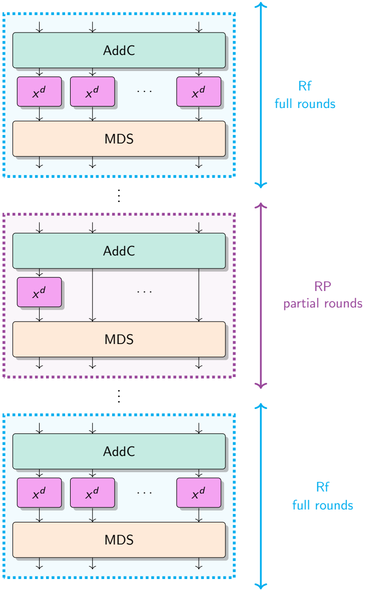
- S-box:

$$x \mapsto x^3$$

- Nb rounds:

$$\begin{aligned} R &= 2 \times Rf + RP \\ &= 8 + (\text{from } 56 \text{ to } 84) \end{aligned}$$

# Example of Type I: Poseidon




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

- S-box:

$$x \mapsto x^3$$

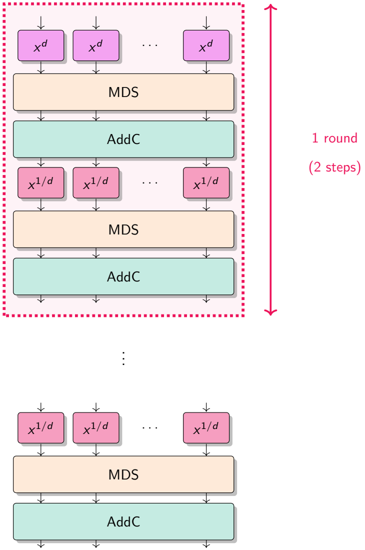
- Nb rounds:

$$R = 2 \times Rf + RP$$
$$= 8 + (\text{from } 56 \text{ to } 84)$$

**Type I (low-degree primitives)**

- fast in plain
- many rounds
- often more constraints

# Example of Type II: *Rescue*




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

- S-box:

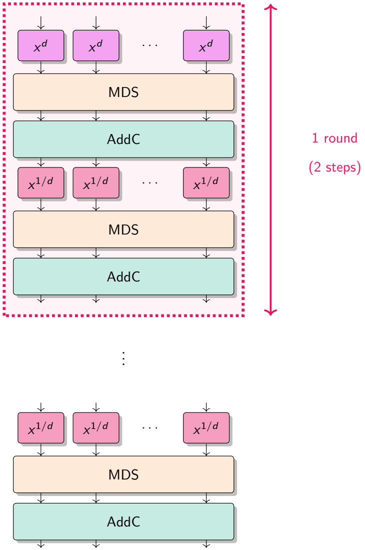
$$x \mapsto x^3 \quad \text{and} \quad x \mapsto x^{1/3}$$

- Nb rounds:

$$R = \text{from 8 to 26}$$

(2 S-boxes per round)

# Example of Type II: *Rescue*




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

- S-box:

$$x \mapsto x^3 \quad \text{and} \quad x \mapsto x^{1/3}$$

- Nb rounds:

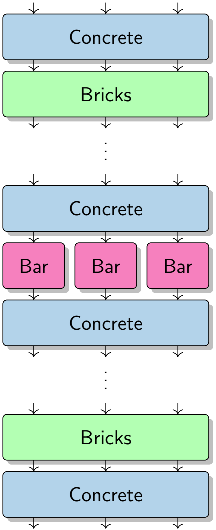
$$R = \text{from 8 to 26}$$

(2 S-boxes per round)

**Type II (equivalence relation)**

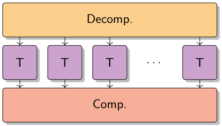
- slow in plain
- fewer rounds
- fewer constraints

# Example of Type III: Reinforced Concrete




L. Grassi, D. Khovratovich, R. Lüftenegger, C. Rechberger, M. Schofnegger and R. Walch, ACM CCS 2022

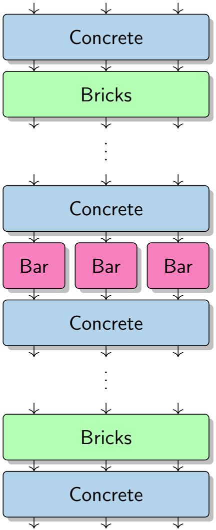
- ⋆ S-box:




- ⋆ Nb rounds:

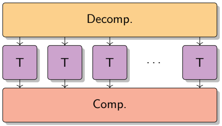
$$R = 7$$

# Example of Type III: Reinforced Concrete




L. Grassi, D. Khovratovich, R. Lüftenegger, C. Rechberger, M. Schofnegger and R. Walch, ACM CCS 2022

⋆ S-box:



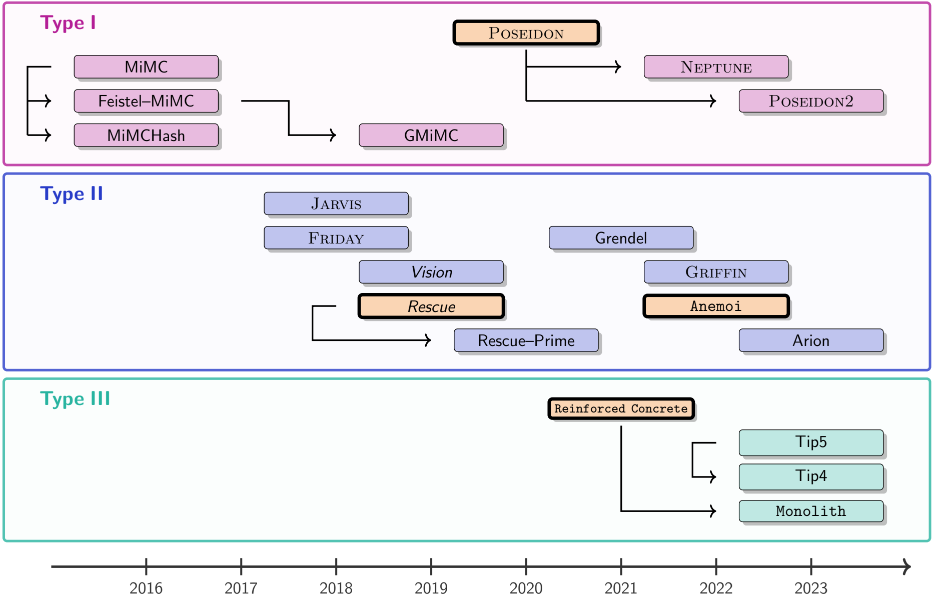

⋆ Nb rounds:

$$R = 7$$

**Type III (look-up tables)**

- ⋆ faster in plain
- ⋆ fewer rounds
- ⋆ constraints depending on proof systems

# Primitives overview



# Design of Anemoi

- ⋆ Link between CCZ-equivalence and Arithmetization-Orientation
- ⋆ A new S-Box: the Flystel
- ⋆ A new family of ZK-friendly hash functions: Anemoi

*joint work with P. Briaud, P. Chaidos, L. Perrin, R. Salen, V. Velichkov and D. Willems, published at CRYPTO 2023*

# Performance metric

What does "efficient" mean for Zero-Knowledge Proofs?

# Performance metric

What does "efficient" mean for Zero-Knowledge Proofs?

**"It depends"**

# Performance metric

What does "efficient" mean for Zero-Knowledge Proofs?

**"It depends"**

**Example**

**R1CS** (Rank-1 Constraint System): minimizing the number of multiplications

$$y = (ax + b)^3(cx + d) + ex$$

$t_0 = a \cdot x$

$t_1 = t_0 + b$

$t_2 = t_1 \times t_1$

$t_3 = t_2 \times t_1$

$t_4 = c \cdot x$

$t_5 = t_4 + d$

$t_6 = t_3 \times t_5$

$t_7 = e \cdot x$

$t_8 = t_6 + t_7$

# Performance metric

What does "efficient" mean for Zero-Knowledge Proofs?

**"It depends"**

**Example**

**R1CS** (Rank-1 Constraint System): minimizing the number of multiplications

$$y = (ax+b)^3(cx+d) + ex$$

$t_0 = a \cdot x$
$t_1 = t_0 + b$
$t_2 = t_1 \times t_1$
$t_3 = t_2 \times t_1$
$t_4 = c \cdot x$
$t_5 = t_4 + d$
$t_6 = t_3 \times t_5$
$t_7 = e \cdot x$
$t_8 = t_6 + t_7$

3 constraints

# Our approach

**Need:** verification using few multiplications.

High degree for security VS Low degree for performance

# Our approach

**Need:** verification using few multiplications.

High degree for security VS Low degree for performance

⋆ **First approach:** using inversion, e.g. *Rescue* [Aly et al., ToSC20]

$y \leftarrow E(x)$ $\rightsquigarrow$ $E$: high degree

$x == E^{-1}(y)$ $\rightsquigarrow$ $E^{-1}$: low degree

# Our approach

**Need:** verification using few multiplications.

High degree for security VS Low degree for performance

⋆ **First approach:** using inversion, e.g. *Rescue* [Aly et al., ToSC20]

$y \leftarrow E(x)$ $\leadsto$ $E$: high degree

$x == E^{-1}(y)$ $\leadsto$ $E^{-1}$: low degree

⋆ **Our approach:** using $(u, v) = \mathcal{L}(x, y)$, where $\mathcal{L}$ is linear

$y \leftarrow E(x)$ $\leadsto$ $E$: high degree

$v == F(u)$ $\leadsto$ $F$: low degree

# CCZ-equivalence

**Definition [Carlet, Charpin and Zinoviev, DCC98]**

$E : \mathbb{F}_q \to \mathbb{F}_q$ and $F : \mathbb{F}_q \to \mathbb{F}_q$ are **CCZ-equivalent** if

$$\Gamma_E = \mathcal{L}(\Gamma_F) + c , \quad \text{where } \mathcal{L} \text{ is linear.}$$

# CCZ-equivalence

**Definition [Carlet, Charpin and Zinoviev, DCC98]**

$E : \mathbb{F}_q \to \mathbb{F}_q$ and $F : \mathbb{F}_q \to \mathbb{F}_q$ are **CCZ-equivalent** if

$$\Gamma_E = \mathcal{L}(\Gamma_F) + c \ , \quad \text{where } \mathcal{L} \text{ is linear.}$$

**Inversion**

$$\Gamma_E = \{(x, E(x)), x \in \mathbb{F}_q\} \quad \text{and} \quad \Gamma_{E^{-1}} = \left\{\left(y, E^{-1}(y)\right), y \in \mathbb{F}_q\right\}$$

Noting that

$$\Gamma_E = \left\{\left(E^{-1}(y), y\right), y \in \mathbb{F}_q\right\} \ ,$$

then, we have:

$$\Gamma_E = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \Gamma_{E^{-1}} \ .$$

# Advantages of CCZ-equivalence

If $E : \mathbb{F}_q \to \mathbb{F}_q$ and $F : \mathbb{F}_q \to \mathbb{F}_q$ are **CCZ-equivalent**. Then

⋆ Differential properties are the same: $\delta_E = \delta_F$ .

**Differential uniformity**

$$\delta_E = \max_{a \neq 0, b} |\{x \in \mathbb{F}_q^m, E(x+a) - E(x) = b\}|$$

# Advantages of CCZ-equivalence

If $E : \mathbb{F}_q \rightarrow \mathbb{F}_q$ and $F : \mathbb{F}_q \rightarrow \mathbb{F}_q$ are **CCZ-equivalent**. Then

⋆ Differential properties are the same: $\delta_E = \delta_F$ .

**Differential uniformity**

$$\delta_E = \max_{a \neq 0, b} \left| \{x \in \mathbb{F}_q^m, E(x+a) - E(x) = b\} \right|$$

⋆ Linear properties are the same: $\mathcal{W}_E = \mathcal{W}_F$ .

**Linearity**

$$\mathcal{W}_E = \max_{a, b \neq 0} \left| \sum_{x \in \mathbb{F}_{2^n}^m} (-1)^{a \cdot x + b \cdot E(x)} \right|$$

# Advantages of CCZ-equivalence

If $E : \mathbb{F}_q \rightarrow \mathbb{F}_q$ and $F : \mathbb{F}_q \rightarrow \mathbb{F}_q$ are **CCZ-equivalent**. Then

- ⋆ Verification is the same: if $y \leftarrow E(x)$, $v \leftarrow F(u)$ and $(u, v) = \mathcal{L}(x, y)$

$$y == E(x)? \quad \Longleftrightarrow \quad v == F(u)?$$

# Advantages of CCZ-equivalence

If $E : \mathbb{F}_q \rightarrow \mathbb{F}_q$ and $F : \mathbb{F}_q \rightarrow \mathbb{F}_q$ are **CCZ-equivalent**. Then

- ⋆ Verification is the same: if $y \leftarrow E(x)$, $v \leftarrow F(u)$ and $(u, v) = \mathcal{L}(x, y)$

$$y == E(x)? \iff v == F(u)?$$

- ⋆ The degree is **not preserved**.

**Example**

in $\mathbb{F}_p$ where

$$p = \texttt{0x73eda753299d7d483339d80809a1d80553bda402fffe5bfeffffffff00000001}$$

if $F(x) = x^5$ then $F^{-1}(x) = x^{5^{-1}}$ where

$$5^{-1} = \texttt{0x2e5f0fbadd72321ce14a56699d73f002217f0e679998f19933333332cccccccd}$$

# Advantages of CCZ-equivalence

If $E : \mathbb{F}_q \to \mathbb{F}_q$ and $F : \mathbb{F}_q \to \mathbb{F}_q$ are **CCZ-equivalent**. Then

- ⋆ Verification is the same: if $y \leftarrow E(x)$, $v \leftarrow F(u)$ and $(u, v) = \mathcal{L}(x, y)$

$$y == E(x)? \quad \Longleftrightarrow \quad v == F(u)?$$

- ⋆ The degree is **not preserved**.

**Example**

in $\mathbb{F}_p$ where

$$p = \texttt{0x73eda753299d7d483339d80809a1d80553bda402fffe5bfeffffffff00000001}$$

if $F(x) = x^5$ then $F^{-1}(x) = x^{5^{-1}}$ where

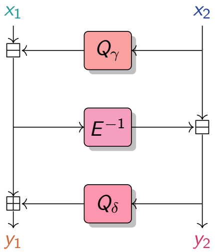
$$5^{-1} = \texttt{0x2e5f0fbadd72321ce14a56699d73f002217f0e679998f19933333332cccccccd}$$

# The Flystel

Butterfly + Feistel ⇒ Flystel

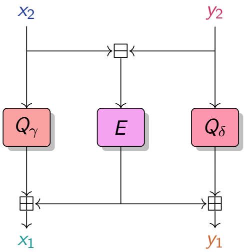
A 3-round Feistel-network with
$Q_\gamma : \mathbb{F}_q \rightarrow \mathbb{F}_q$ and $Q_\delta : \mathbb{F}_q \rightarrow \mathbb{F}_q$ two quadratic functions, and $E : \mathbb{F}_q \rightarrow \mathbb{F}_q$ a permutation

**High-Degree** permutation




*Open* Flystel $\mathcal{H}$.

**Low-Degree** function



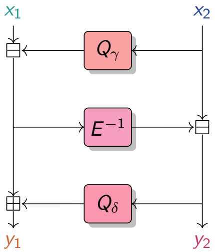

*Closed* Flystel $\mathcal{V}$.

# The Flystel

Butterfly + Feistel ⇒ Flystel

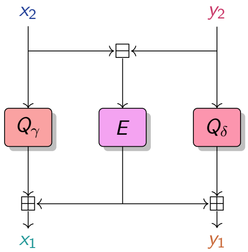
A 3-round Feistel-network with
$Q_\gamma : \mathbb{F}_q \to \mathbb{F}_q$ and $Q_\delta : \mathbb{F}_q \to \mathbb{F}_q$ two quadratic functions, and $E : \mathbb{F}_q \to \mathbb{F}_q$ a permutation

**High-Degree** permutation




*Open* Flystel $\mathcal{H}$.

**Low-Degree** function



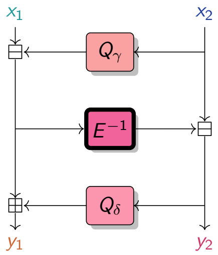

*Closed* Flystel $\mathcal{V}$.

$$\Gamma_{\mathcal{H}} = \mathcal{L}(\Gamma_{\mathcal{V}}) \quad \text{s.t.} \quad ((x_1, x_2), (y_1, y_2)) = \mathcal{L}\left( ((y_2, x_2), (x_1, y_1)) \right)$$

# Advantage of CCZ-equivalence

⋆ High-Degree Evaluation.

**High-Degree**
permutation




*Open* `Flystel` $\mathcal{H}$.

**Example**

if $E : x \mapsto x^5$ in $\mathbb{F}_p$ where

$$p = \texttt{0x73eda753299d7d483339d80809a1d805}$$
$$\texttt{53bda402fffe5bfeffffffff00000001}$$

then $E^{-1} : x \mapsto x^{5^{-1}}$ where

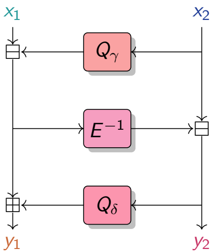
$$5^{-1} = \texttt{0x2e5f0fbadd72321ce14a56699d73f002}$$
$$\texttt{217f0e679998f19933333332cccccccd}$$

# Advantage of CCZ-equivalence

- ⋆ High-Degree Evaluation.
- ⋆ Low-Degree Verification.

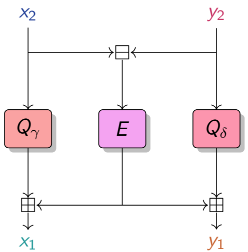
$$(y_1, y_2) == \mathcal{H}(x_1, x_2) \Leftrightarrow (x_1, y_1) == \mathcal{V}(x_2, y_2)$$

**High-Degree** permutation




*Open* `Flystel` $\mathcal{H}$.

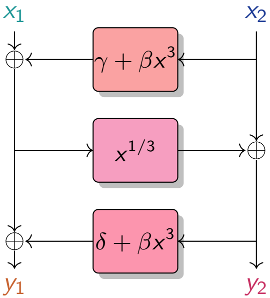
**Low-Degree** function



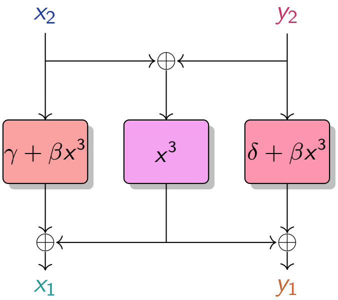

*Closed* `Flystel` $\mathcal{V}$.

# Flystel in $\mathbb{F}_{2^n}$, $n$ odd

$$Q_\gamma(x) = \gamma + \beta x^3 \;, \quad Q_\delta(x) = \delta + \beta x^3 \;, \quad \text{and} \quad E(x) = x^3$$

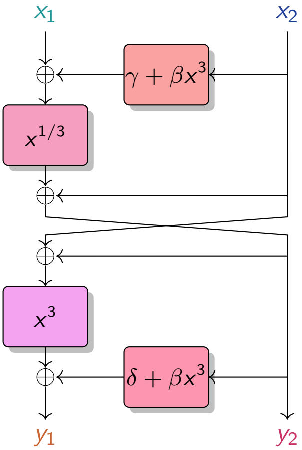



*Open* Flystel$_2$.




*Closed* Flystel$_2$.

# Properties of Flystel in $\mathbb{F}_{2^n}$, $n$ odd




*Degenerated Butterfly.*

Introduced by [Perrin et al. 2016].

Theorems in [Li et al. 2018] state that if $\beta \neq 0$:

- Differential properties

$$\delta_{\mathcal{H}} = \delta_{\mathcal{V}} = 4$$

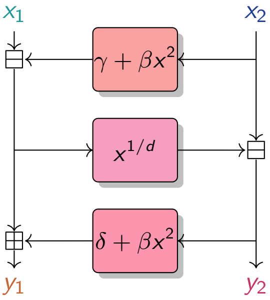
- Linear properties

$$\mathcal{W}_{\mathcal{H}} = \mathcal{W}_{\mathcal{V}} = 2^{n+1}$$

- Algebraic degree
  - Open Flystel$_2$: $\deg_{\mathcal{H}} = n$
  - Closed Flystel$_2$: $\deg_{\mathcal{V}} = 2$

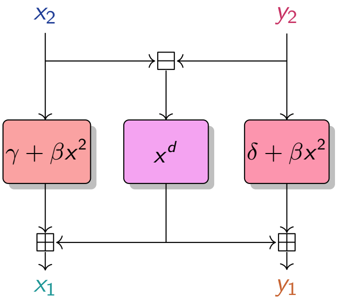
# Flystel in $\mathbb{F}_p$

$$Q_\gamma(x) = \gamma + \beta x^2 , \quad Q_\delta(x) = \delta + \beta x^2 , \quad \text{and} \quad E(x) = x^d$$




*Open Flystel$_p$.*

usually
$d = 3$ or 5.




*Closed Flystel$_p$.*

# Properties of Flystel in $\mathbb{F}_p$

⋆ Differential properties

$\texttt{Flystel}_\texttt{p}$ has a differential uniformity:

$$\delta_{\mathcal{H}} = \max_{a \neq 0, b} \left|\{x \in \mathbb{F}_p^2, \mathcal{H}(x + a) - \mathcal{H}(x) = b\}\right| \leq d - 1$$

# Properties of Flystel in $\mathbb{F}_p$

⋆ Differential properties

$\texttt{Flystel}_\texttt{p}$ has a differential uniformity:

$$\delta_{\mathcal{H}} = \max_{a \neq 0, b} |\{x \in \mathbb{F}_p^2, \mathcal{H}(x + a) - \mathcal{H}(x) = b\}| \leq d - 1$$

Solving the open problem of finding an APN (Almost-Perfect Non-linear) permutation over $\mathbb{F}_p^2$

# Properties of Flystel in $\mathbb{F}_p$

⋆ Differential properties

$\texttt{Flystel}_\texttt{p}$ has a differential uniformity:

$$\delta_{\mathcal{H}} = \max_{a \neq 0, b} |\{x \in \mathbb{F}_p^2, \mathcal{H}(x+a) - \mathcal{H}(x) = b\}| \leq d - 1$$

Solving the open problem of finding an APN (Almost-Perfect Non-linear) permutation over $\mathbb{F}_p^2$

⋆ Linear properties

Conjecture:

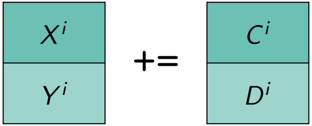
$$\mathcal{W}_{\mathcal{H}} = \max_{a, b \neq 0} \left| \sum_{x \in \mathbb{F}_p^2} exp\left(\frac{2\pi i(\langle a, x\rangle - \langle b, \mathcal{H}(x)\rangle)}{p}\right) \right| \leq p \log p \ ?$$

# The SPN Structure

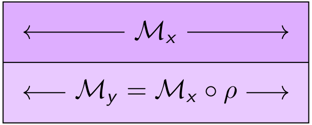
The internal state of `Anemoi` and its basic operations.

A Substitution-Permutation Network with:

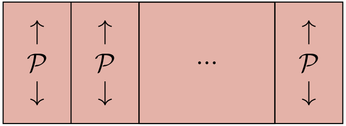

**(a)** *Internal state.*




**(b)** *The constant addition.*

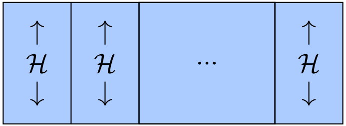



**(c)** *The diffusion layer.*



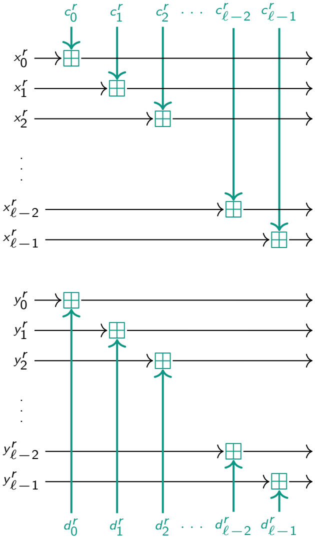

with $\mathcal{P} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$

**(d)** *The Pseudo-Hadamard Transform.*

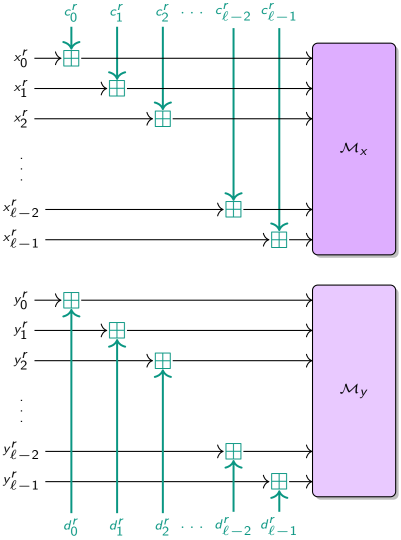



**(e)** *The S-box layer.*

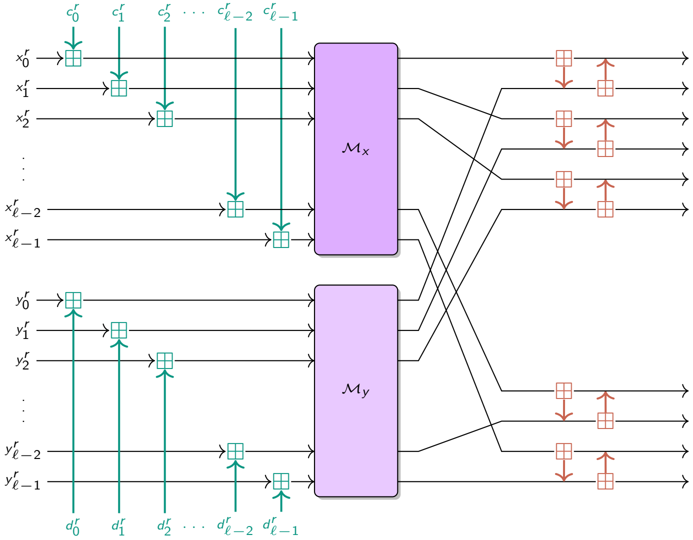
# The SPN Structure



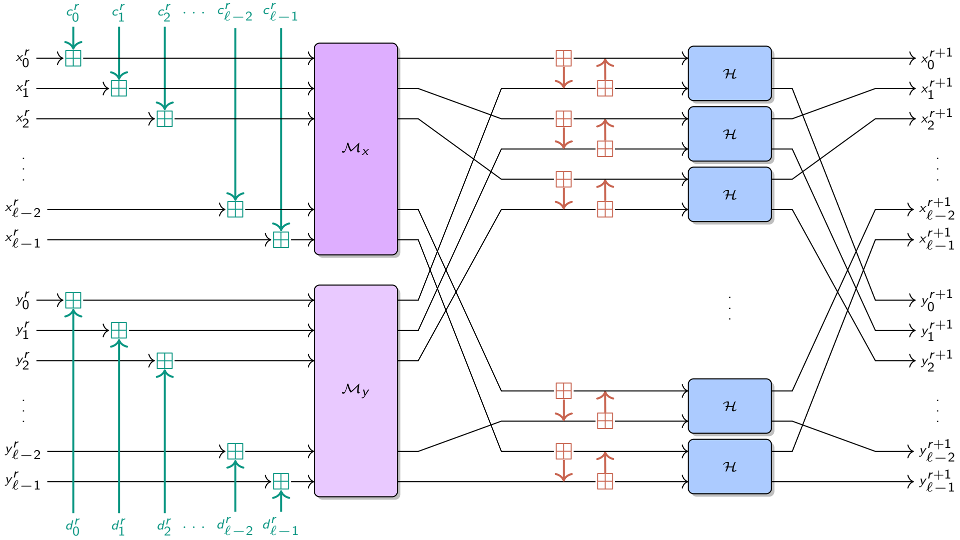
# The SPN Structure



# The SPN Structure



# The SPN Structure



# Performance metric

What does "efficient" mean for Zero-Knowledge Proofs?

**"It depends"**

**Example**

**R1CS** (Rank-1 Constraint System): minimizing the number of multiplications

$$y = (ax + b)^3 (cx + d) + ex$$

$t_0 = a \cdot x$ | $t_3 = t_2 \times t_1$ | $t_6 = t_3 \times t_5$

$t_1 = t_0 + b$ | $t_4 = c \cdot x$ | $t_7 = e \cdot x$

$t_2 = t_1 \times t_1$ | $t_5 = t_4 + d$ | $t_8 = t_6 + t_7$

3 constraints

# Some Benchmarks

| | $m$ (= $2\ell$) | RP[1] | POSEIDON[2] | GRIFFIN[3] | Anemoi |
|---|---|---|---|---|---|
| R1CS | 2 | 208 | 198 | - | **76** |
| | 4 | 224 | 232 | 112 | **96** |
| | 6 | 216 | 264 | - | **120** |
| | 8 | 256 | 296 | 176 | **160** |
| Plonk | 2 | 312 | 380 | - | **191** |
| | 4 | 560 | 832 | **260** | 316 |
| | 6 | 756 | 1344 | - | **460** |
| | 8 | 1152 | 1920 | **574** | 648 |
| AIR | 2 | 156 | 300 | - | **126** |
| | 4 | **168** | 348 | **168** | **168** |
| | 6 | **162** | 396 | - | 216 |
| | 8 | **192** | 456 | 264 | 288 |

**(a)** *when $d = 3$.*

| | $m$ (= $2\ell$) | RP | POSEIDON | GRIFFIN | Anemoi |
|---|---|---|---|---|---|
| R1CS | 2 | 240 | 216 | - | **95** |
| | 4 | 264 | 264 | **110** | 120 |
| | 6 | 288 | 315 | - | **150** |
| | 8 | 384 | 363 | **162** | 200 |
| Plonk | 2 | 320 | 344 | - | **212** |
| | 4 | 528 | 696 | **222** | 344 |
| | 6 | 768 | 1125 | - | **496** |
| | 8 | 1280 | 1609 | **492** | 696 |
| AIR | 2 | **200** | 360 | - | 210 |
| | 4 | **220** | 440 | **220** | 280 |
| | 6 | **240** | 540 | - | 360 |
| | 8 | **320** | 640 | 360 | 480 |

**(b)** *when $d = 5$.*

*Constraint comparison for standard arithmetization, without optimization ($s = 128$).*

[1] *Rescue* [Aly et al., ToSC20]   [2] POSEIDON [Grassi et al., USENIX21]   [3] GRIFFIN [Grassi et al., CRYPTO23]

# Some Benchmarks

*** Numbers to be updated! ***

| | $m\ (=2\ell)$ | RP[1] | Poseidon[2] | Griffin[3] | Anemoi |
|---|---|---|---|---|---|
| R1CS | 2 | 208 | 198 | - | **76** |
| | 4 | 224 | 232 | 112 | **96** |
| | 6 | 216 | 264 | - | **120** |
| | 8 | 256 | 296 | 176 | **160** |
| Plonk | 2 | 312 | 380 | - | **191** |
| | 4 | 560 | 832 | **260** | 316 |
| | 6 | 756 | 1344 | - | **460** |
| | 8 | 1152 | 1920 | **574** | 648 |
| AIR | 2 | 156 | 300 | - | **126** |
| | 4 | **168** | 348 | **168** | **168** |
| | 6 | **162** | 396 | - | 216 |
| | 8 | **192** | 456 | 264 | 288 |

**(a)** *when* $d = 3$.

| | $m\ (=2\ell)$ | RP | Poseidon | Griffin | Anemoi |
|---|---|---|---|---|---|
| R1CS | 2 | 240 | 216 | - | **95** |
| | 4 | 264 | 264 | **110** | 120 |
| | 6 | 288 | 315 | - | **150** |
| | 8 | 384 | 363 | **162** | 200 |
| Plonk | 2 | 320 | 344 | - | **212** |
| | 4 | 528 | 696 | **222** | 344 |
| | 6 | 768 | 1125 | - | **496** |
| | 8 | 1280 | 1609 | **492** | 696 |
| AIR | 2 | **200** | 360 | - | 210 |
| | 4 | **220** | 440 | **220** | 280 |
| | 6 | **240** | 540 | - | 360 |
| | 8 | **320** | 640 | 360 | 480 |

**(b)** *when* $d = 5$.

*Constraint comparison for standard arithmetization, without optimization* ($s = 128$).

[1] *Rescue* [Aly et al., ToSC20]  [2] Poseidon [Grassi et al., USENIX21]  [3] Griffin [Grassi et al., CRYPTO23]

# Take-Away

Anemoi: A new family of ZK-friendly hash functions

- ⋆ Identify a link between AO and CCZ-equivalence
- ⋆ Contributions of fundamental interest:
  - ⋆ New S-box: Flystel
  - ⋆ New mode: Jive

# Take-Away

Anemoi: A new family of ZK-friendly hash functions

- ⋆ Identify a link between AO and CCZ-equivalence
- ⋆ Contributions of fundamental interest:
  - ⋆ New S-box: Flystel
  - ⋆ New mode: Jive

Related works and cryptanalysis

- ⋆ $\texttt{AnemoiJive}_3$ with TurboPlonK [Liu et al., 2022]
- ⋆ Arion [Roy, Steiner and Trevisani, 2023]
- ⋆ APN permutations over prime fields [Budaghyan and Pal, 2023]
- ⋆ Algebraic attacks [Bariant et al., CRYPTO24], [Koschatko, Lüftenegger and Rechberger, 2024]

# Algebraic Attacks against AOP

- ⋆ Solving the CICO problem
- ⋆ Trick to bypass rounds of SPN construction
  - ⋆ Application to Poseidon and Rescue–Prime
  - ⋆ Solving Ethereum Challenges

*joint work with A. Bariant, G. Leurent and L. Perrin, published at ToSC 2022*

- ⋆ FreeLunch attack

# CICO Problem

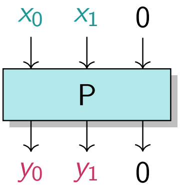
**CICO: Constrained Input Constrained Output**

**Definition**

Let $P : \mathbb{F}_q^t \to \mathbb{F}_q^t$ and $u < t$.

The **CICO** problem is:

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




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

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

- ⋆ Feistel–MiMC [Albrecht et al., AC16]
- ⋆ Poseidon [Grassi et al., USENIX21]
- ⋆ Rescue–Prime [Aly et al., ToSC20]
- ⋆ Reinforced Concrete [Grassi et al., CCS22]

# Solving polynomial systems

- **Univariate** solving: find the roots of $\mathcal{P}_j \in \mathbb{F}_q[X]$

$$\begin{cases} \mathcal{P}_0(X) & = 0 \\ & \vdots \\ \mathcal{P}_{m-1}(X) & = 0 \,. \end{cases}$$

- **Multivariate** solving: find the roots of $\mathcal{P}_j \in \mathbb{F}_q[X_0, \ldots, X_{n-1}]$

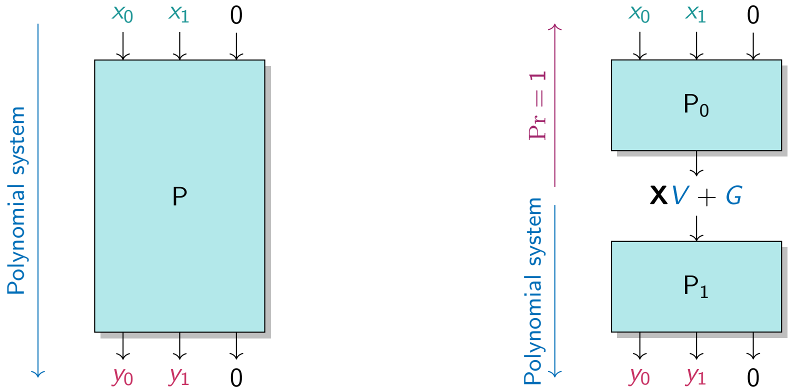
$$\begin{cases} \mathcal{P}_0(X_0, \ldots, X_{n-1}) & = 0 \\ & \vdots \\ \mathcal{P}_{m-1}(X_0, \ldots, X_{n-1}) & = 0 \,. \end{cases}$$

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

# Trick for SPN

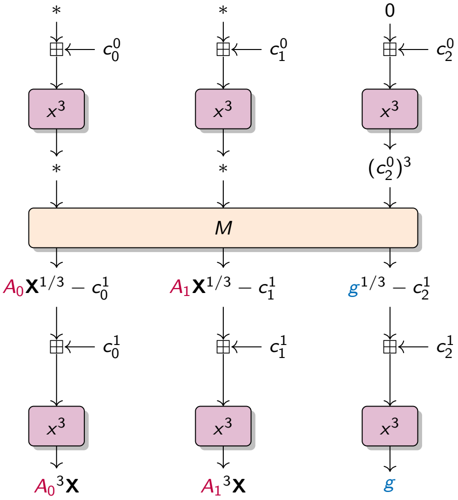
Let $P = P_0 \circ P_1$ be a permutation of $\mathbb{F}_p^3$ and suppose

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

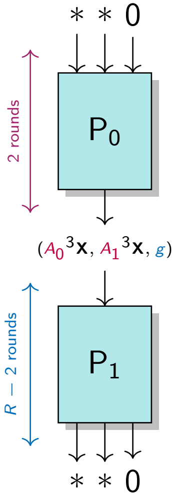



**(a)** *R-round system.* **(b)** *$(R-2)$-round system.*

# Trick for Poseidon

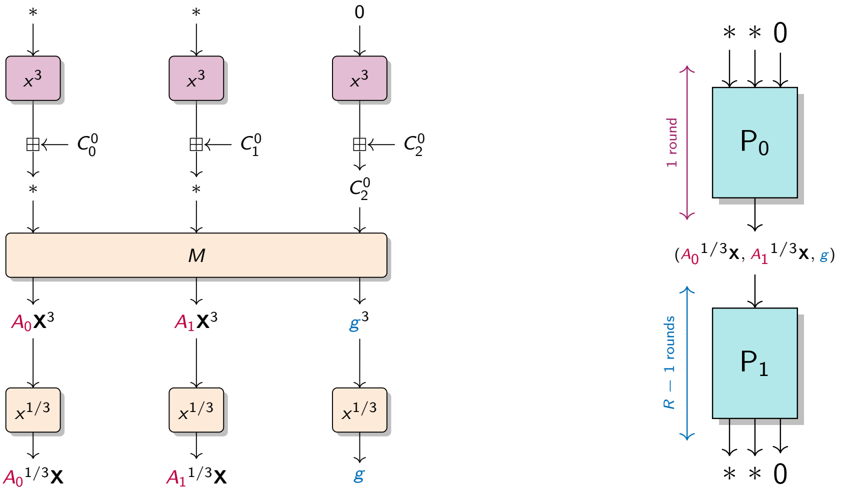



**(a)** *First two rounds.*




**(b)** *Overview.*

# Trick for Rescue–Prime




**(a)** *First round.* **(b)** *Overview.*

# Cryptanalysis Challenge

| 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`

# FreeLunch attack

A. Bariant, A. Boeuf, A. Lemoine, I. Manterola Ayala, M. Øygarden, L. Perrin, and H. Raddum, CRYPTO 2024

**Multivariate** solving:

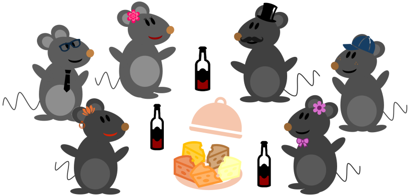
- ⋆ Define the system
- ⋆ Compute a grevlex order GB (**F5** algorithm)
- ⋆ Convert it into lex order GB (**FGLM** algorithm)
- ⋆ Find the roots in $\mathbb{F}_q^n$ of the GB polynomials using univariate system resolution.

# FreeLunch attack

A. Bariant, A. Boeuf, A. Lemoine, I. Manterola Ayala, M. Øygarden, L. Perrin, and H. Raddum, CRYPTO 2024

**Multivariate** solving:

- ⋆ Define the system
- ⋆ Compute a grevlex order GB (**F5** algorithm) $\rightsquigarrow$ **can be skipped**
- ⋆ Convert it into lex order GB (**FGLM** algorithm)
- ⋆ Find the roots in $\mathbb{F}_q^n$ of the GB polynomials using univariate system resolution.



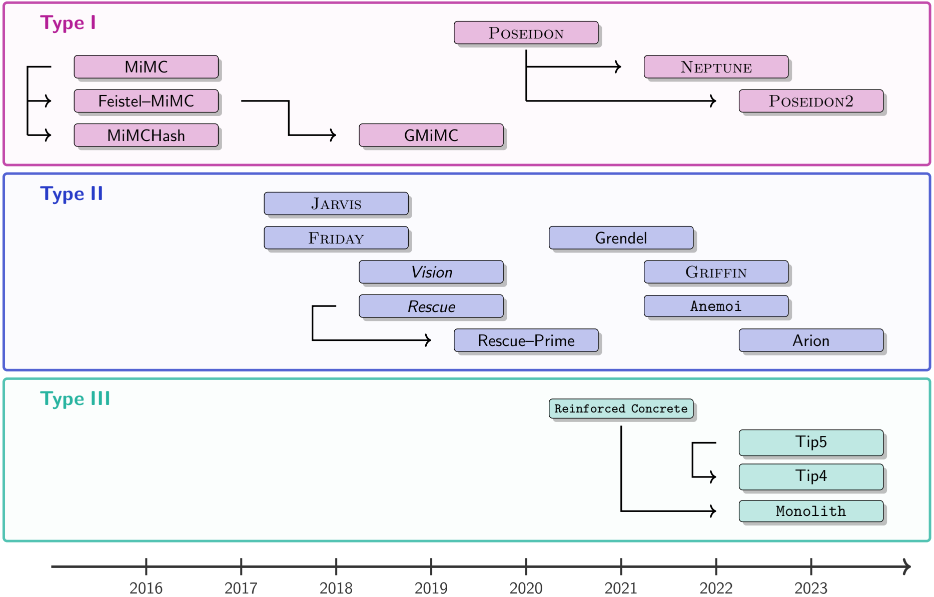
# Take-Away

# Take-Away

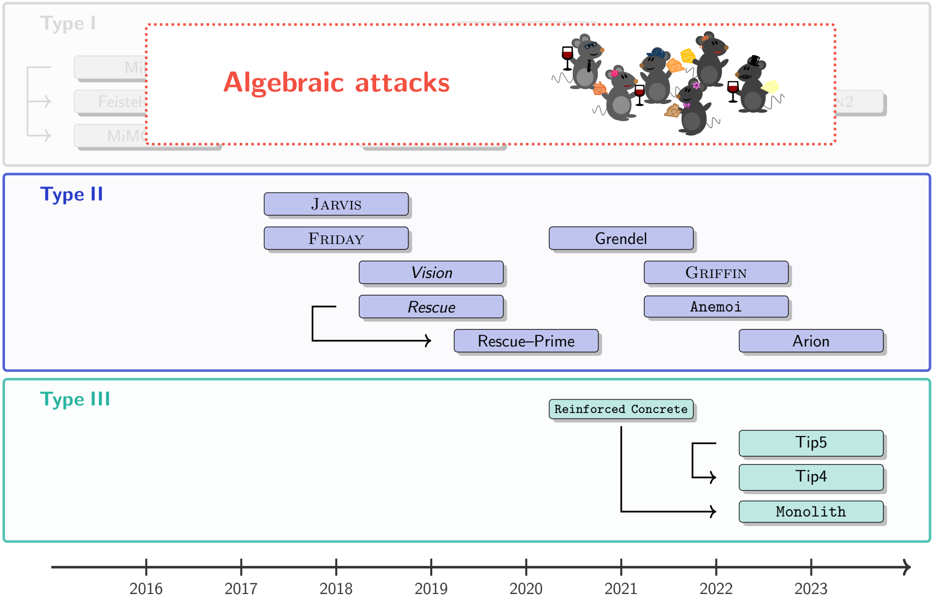
Recommendations for future designs

- ⋆ study possible tricks to bypass rounds
- ⋆ prefer univariate instead of multivariate systems
- ⋆ consider as many variants of modeling and ordering as possible

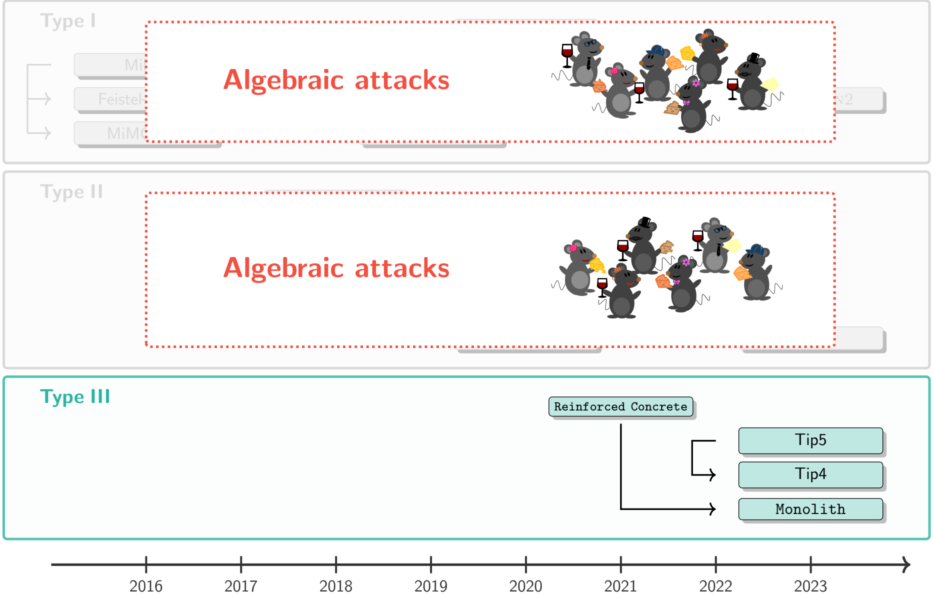
# Cryptanalysis overview



# Cryptanalysis overview



# Cryptanalysis overview



# Cryptanalysis overview

Type I

Mi
Feistel
MiMC

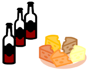
**Algebraic attacks**

Type II

**Algebraic attacks**

Type III

**No cryptanalysis**



2016 2017 2018 2019 2020 2021 2022 2023

# Conclusions and Perspectives

New designs and cryptanalysis techniques for AOP

- ⋆ Anemoi: new tools for designing primitives (Jive, Flystel)
- ⋆ A better insight into the behaviour of algebraic systems

# Conclusions and Perspectives

New designs and cryptanalysis techniques for AOP

- ⋆ `Anemoi`: new tools for designing primitives (`Jive`, `Flystel`)
- ⋆ A better insight into the behaviour of algebraic systems

**Cryptanalysis and designing of AOP remain to be explored!**

- ⋆ missing cryptanalysis for Type III
- ⋆ investigating new areas of application
- ⋆ . . .

# Conclusions and Perspectives

New designs and cryptanalysis techniques for AOP

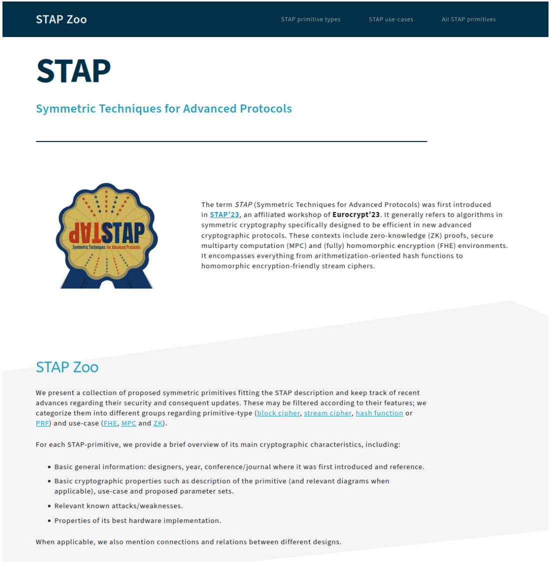
- ⋆ Anemoi: new tools for designing primitives (Jive, Flystel)
- ⋆ A better insight into the behaviour of algebraic systems

Cryptanalysis and designing of AOP remain to be explored!

- ⋆ missing cryptanalysis for Type III
- ⋆ investigating new areas of application
- ⋆ ...

Thank you

# Website


STAP Zoo    STAP primitive types    STAP use-cases    All STAP primitives

**STAP**

Symmetric Techniques for Advanced Protocols

The term *STAP* (Symmetric Techniques for Advanced Protocols) was first introduced in STAP'23, an affiliated workshop of **Eurocrypt'23**. It generally refers to algorithms in symmetric cryptography specifically designed to be efficient in new advanced cryptographic protocols. These contexts include zero-knowledge (ZK) proofs, secure multiparty computation (MPC) and (fully) homomorphic encryption (FHE) environments. It encompasses everything from arithmetization-oriented hash functions to homomorphic encryption-friendly stream ciphers.

STAP Zoo

We present a collection of proposed symmetric primitives fitting the STAP description and keep track of recent advances regarding their security and consequent updates. These may be filtered according to their features; we categorize them into different groups regarding primitive-type (block cipher, stream cipher, hash function or PRF) and use-case (FHE, MPC and ZK).

For each STAP-primitive, we provide a brief overview of its main cryptographic characteristics, including:

- Basic general information: designers, year, conference/journal where it was first introduced and reference.
- Basic cryptographic properties such as description of the primitive (and relevant diagrams when applicable), use-case and proposed parameter sets.
- Relevant known attacks/weaknesses.
- Properties of its best hardware implementation.

When applicable, we also mention connections and relations between different designs.

See more at

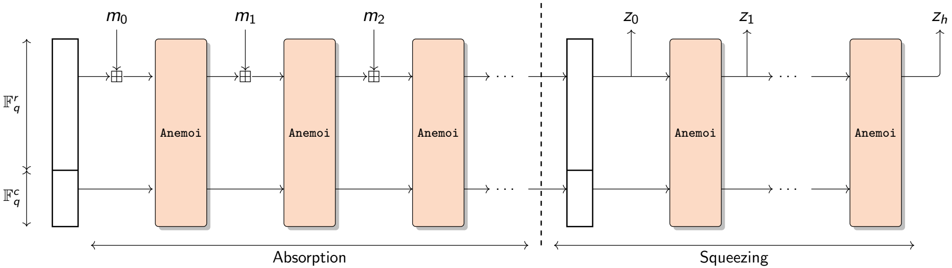
stap-zoo.com

# Anemoi

# More benchmarks and Cryptanalysis

# Sponge construction

- Hash function (random oracle):
  - input: arbitrary length
  - ouput: fixed length

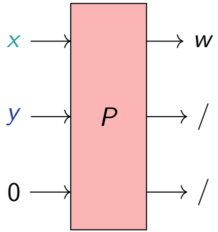


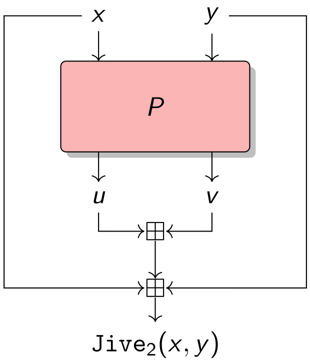
# New Mode: Jive

- ⋆ Compression function (Merkle-tree):
  - ⋆ input: fixed length
  - ⋆ output: (input length) /2

Dedicated mode: 2 words in 1

$$(x, y) \mapsto x + y + u + v \ .$$



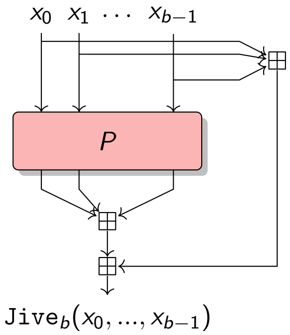



# New Mode: Jive

- Compression function (Merkle-tree):
  - input: fixed length
  - output: (input length) /b

Dedicated mode: b words in 1

$$\texttt{Jive}_b(P) : \begin{cases} (\mathbb{F}_q^m)^b & \to \mathbb{F}_q^m \\ (x_0, \ldots, x_{b-1}) & \mapsto \displaystyle\sum_{i=0}^{b-1} (x_i + P_i(x_0, \ldots, x_{b-1})) \end{cases} .$$



# Comparison for Plonk (with optimizations)

| | $m$ | Constraints |
|---|---|---|
| Poseidon | 3 | 110 |
| | 2 | 88 |
| Reinforced Concrete | 3 | 378 |
| | 2 | 236 |
| Rescue–Prime | 3 | 252 |
| Griffin | 3 | 125 |
| AnemoiJive | 2 | ~~86~~ 56 |

**(a)** *With 3 wires.*

| | $m$ | Constraints |
|---|---|---|
| Poseidon | 3 | 98 |
| | 2 | 82 |
| Reinforced Concrete | 3 | 267 |
| | 2 | 174 |
| Rescue–Prime | 3 | 168 |
| Griffin | 3 | 111 |
| AnemoiJive | 2 | **64** |

**(b)** *With 4 wires.*

*Constraints comparison with an additional custom gate for $x^\alpha$. ($s = 128$).*

**with an additional quadratic custom gate: 56 constraints**

# Native performance

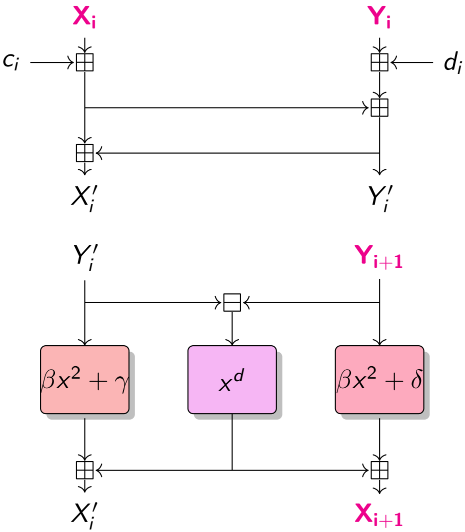
| *Rescue*-12 | *Rescue*-8 | Poseidon-12 | Poseidon-8 | Griffin-12 | Griffin-8 | Anemoi-8 |
|---|---|---|---|---|---|---|
| 15.67 $\mu$s | 9.13 $\mu$s | 5.87 $\mu$s | 2.69 $\mu$s | 2.87 $\mu$s | **2.59 $\mu$s** | 4.21 $\mu$s |

*2-to-1 compression functions for $\mathbb{F}_p$ with $p = 2^{64} - 2^{32} + 1$ ($s = 128$).*

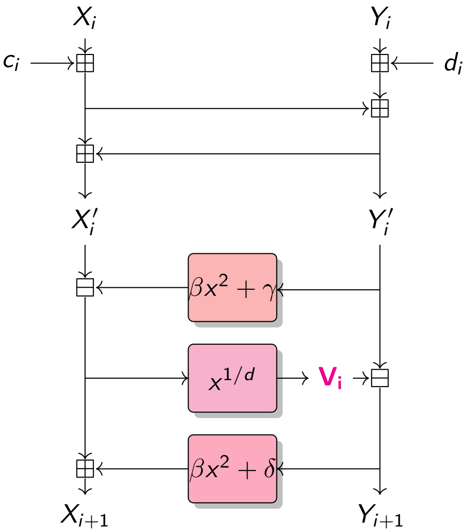
| *Rescue* | Poseidon | Griffin | Anemoi |
|---|---|---|---|
| 206 $\mu$s | **9.2 $\mu$s** | 74.18 $\mu$s | 128.29 $\mu$s |

*For $BLS12 - 381$, Rescue, Poseidon, Anemoi with state size of 2, Griffin of 3 ($s = 128$).*

# Algebraic attacks: 2 modelings




**(a)** *Model 1.*

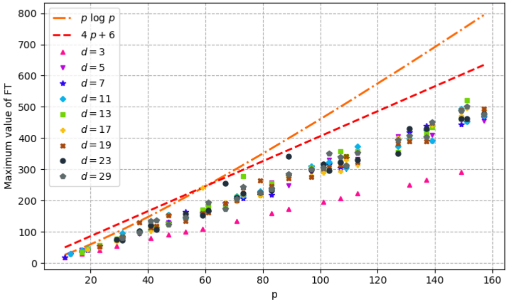



**(b)** *Model 2.*

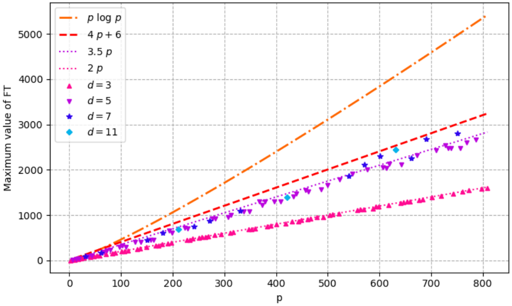
# Properties of `Flystel` in $\mathbb{F}_p$

⋆ Linear properties

$$\mathcal{W}_{\mathcal{H}} = \max_{a,b \neq 0} \left| \sum_{x \in \mathbb{F}_p^2} exp\left( \frac{2\pi i(\langle a, x\rangle - \langle b, \mathcal{H}(x)\rangle)}{p} \right) \right| \leq p \log p \ ?$$




**(a)** *For different d.*



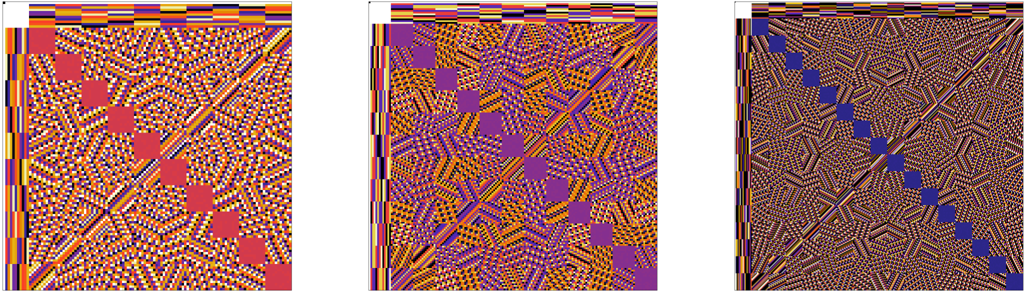

**(b)** *For the smallest d.*

*Conjecture for the linearity.*

# Properties of Flystel in $\mathbb{F}_p$

⋆ Linear properties

$$\mathcal{W}_{\mathcal{H}} = \max_{a,b \neq 0} \left| \sum_{x \in \mathbb{F}_p^2} exp\left( \frac{2\pi i(\langle a, x \rangle - \langle b, \mathcal{H}(x) \rangle)}{p} \right) \right| \leq p \log p \ ?$$



**(a)** *when* $p = 11$ *and* $d = 3$.

**(b)** *when* $p = 13$ *and* $d = 5$.

**(c)** *when* $p = 17$ *and* $d = 3$.

*LAT of* $\texttt{Flystel}_p$.