A New Approach for Arithmetization-Oriented Symmetric Primitives.

Clémence Bouvier [1,2]

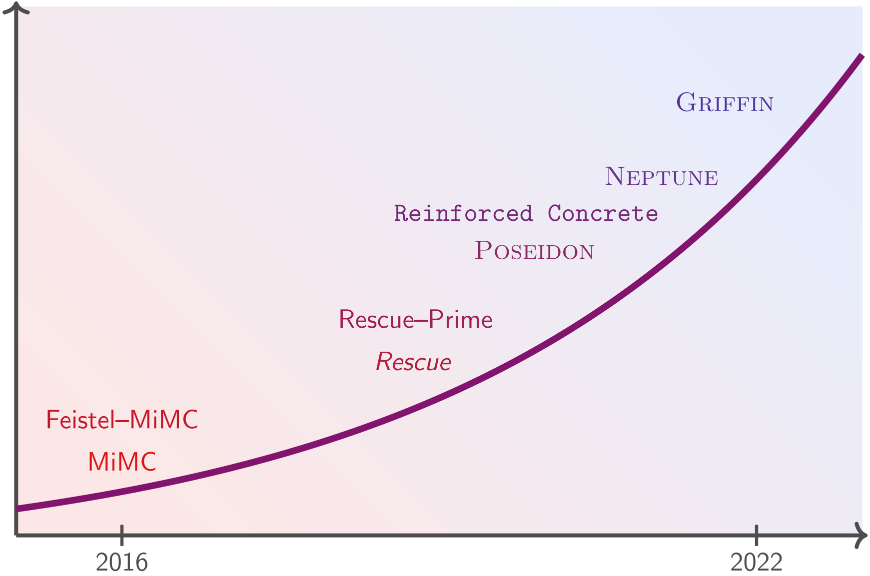
joint work with Pierre Briaud[1,2], Pyrros Chaidos[3], Léo Perrin[2],
Robin Salen[4], Vesselin Velichkov[5,6] and Danny Willems[7,8]

[1]Sorbonne Université, [2]Inria Paris,

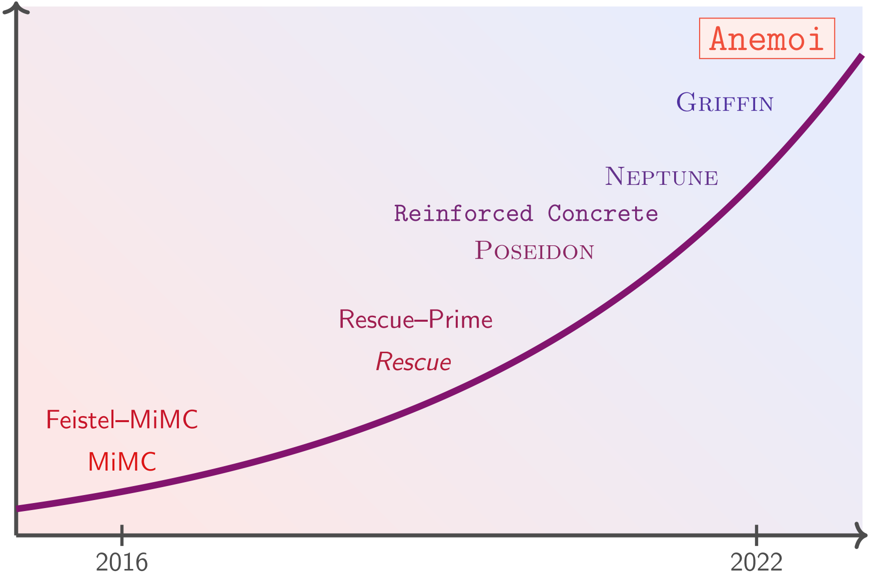
[3]National & Kapodistrian University of Athens, [4]Toposware Inc., Boston,
[5]University of Edinburgh, [6]Clearmatics, London, [7]Nomadic Labs, Paris, [8]Inria and LIX, CNRS

CrossFyre, October 7th, 2022

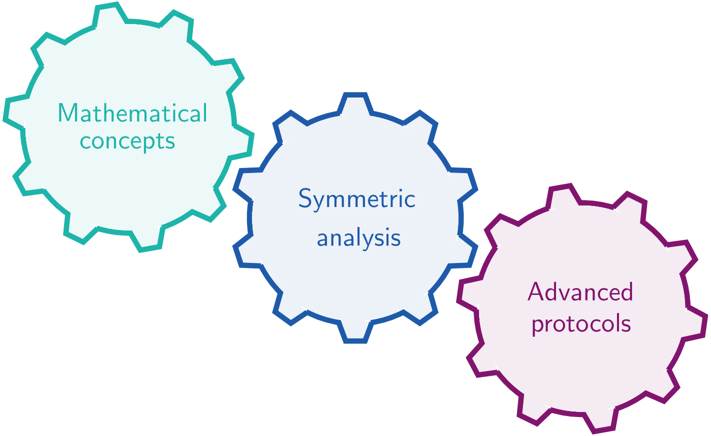
A fast moving domain



A fast moving domain



Designing Arithmetization-Oriented Primitives



Content

A New Approach for Arithmetization-Oriented Symmetric Primitives.

1. Preliminaries
   - Emerging uses in symmetric cryptography
   - CCZ-equivalence

2. Anemoi: a new family of hash-functions
   - New S-box: Flystel
   - New mode: Jive
   - Comparison to previous work

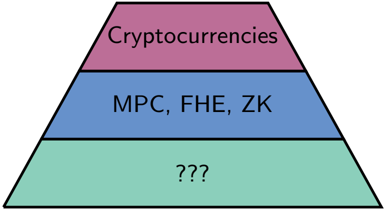
3. Conclusions

A need of new primitives

Problem: Designing new symmetric primitives

Protocols requiring new primitives:

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

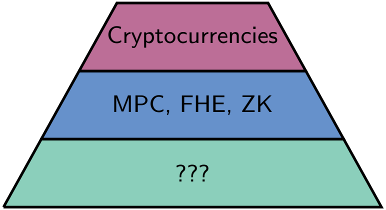


A need of new primitives

Problem: Designing new symmetric primitives

Protocols requiring new primitives:

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



Arithmetization-oriented primitives

$\Rightarrow$ What differs from the "usual" case?

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 Standardized Elliptic Curves.

Examples:

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

$$p = 4002409555221667393417789825735904156556882819939007885332\\058136124031650490837864442687629129015664037894272559787$$

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

$$p = 2586644260129690940106527336948935335363935127549146605398\\842626667204683483408227749688881395733601244403214581 77$$

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

- Operations:
  $$y \leftarrow E(x)$$
- Efficiency:
  implementation in software/hardware

Arithmetization-friendly

- Operations:
  $$y == E(x)$$
- Efficiency:
  integration within advanced protocols

Comparison with "usual" case

A new environment

"Usual" case

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

Arithmetization-friendly

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

New properties

"Usual" case

- Operations:
  $$\boxed{y \leftarrow E(x)}$$
- Efficiency:
  implementation in software/hardware

Arithmetization-friendly

- Operations:
  $$\boxed{y == E(x)}$$
- Efficiency:
  integration within advanced protocols

Our approach

Need: verification using few multiplications.

Our approach

Need: verification using few multiplications.

First approach: evaluation also using few multiplications.

Our approach

Need: verification using few multiplications.

First approach: evaluation also using few multiplications.

$\boxed{y \leftarrow E(x)}$ $\rightsquigarrow E$: low degree

$\boxed{y == E(x)}$ $\rightsquigarrow E$: low degree

Our approach

Need: verification using few multiplications.

First approach: evaluation also using few multiplications.

$\boxed{y \leftarrow E(x)}$ $\rightsquigarrow E$: low degree

$\boxed{y == E(x)}$ $\rightsquigarrow E$: low degree

$\Rightarrow$ vulnerability to some attacks...

Our approach

Need: verification using few multiplications.

First approach: evaluation also using few multiplications.

$\boxed{y \leftarrow E(x)}$ $\rightsquigarrow E$: low degree

$\boxed{y == E(x)}$ $\rightsquigarrow E$: low degree

$\Rightarrow$ vulnerability to some attacks...

New approach:

CCZ-equivalence

Our vision

A function is arithmetization-oriented if it is **CCZ-equivalent** to a function that can be verified efficiently.

Our approach

Need: verification using few multiplications.

First approach: evaluation also using few multiplications.

$\boxed{y \leftarrow E(x)}$ $\rightsquigarrow$ E: low degree

$\boxed{y == E(x)}$ $\rightsquigarrow$ E: low degree

$\Rightarrow$ vulnerability to some attacks...

New approach:

CCZ-equivalence

Our vision

A function is arithmetization-oriented if it is **CCZ-equivalent** to a function that can be verified efficiently.

$\boxed{y \leftarrow F(x)}$ $\rightsquigarrow$ F: high degree

$\boxed{v == G(u)}$ $\rightsquigarrow$ G: low degree

CCZ-equivalence

Definition [Carlet, Charpin, Zinoviev, DCC98]

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

$$\Gamma_F = \left\{ (x, F(x)) \mid x \in \mathbb{F}_q \right\} = \mathcal{A}(\Gamma_G) = \left\{ \mathcal{A}(x, G(x)) \mid x \in \mathbb{F}_q \right\} ,$$

where $\mathcal{A}$ is an affine permutation, $\mathcal{A}(x) = \mathcal{L}(x) + c$.

CCZ-equivalence

Definition [Carlet, Charpin, Zinoviev, DCC98]

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

$$\Gamma_F = \left\{ (x, F(x)) \mid x \in \mathbb{F}_q \right\} = \mathcal{A}(\Gamma_G) = \left\{ \mathcal{A}(x, G(x)) \mid x \in \mathbb{F}_q \right\},$$

where $\mathcal{A}$ is an affine permutation, $\mathcal{A}(x) = \mathcal{L}(x) + c$.

Important things to remember!

- Verification is the same: if $y \leftarrow F(x)$, $v \leftarrow G(u)$

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

CCZ-equivalence

Definition [Carlet, Charpin, Zinoviev, DCC98]

$F : \mathbb{F}_q \rightarrow \mathbb{F}_q$ and $G : \mathbb{F}_q \rightarrow \mathbb{F}_q$ are **CCZ-equivalent** if

$$\Gamma_F = \{ (x, F(x)) \mid x \in \mathbb{F}_q \} = \mathcal{A}(\Gamma_G) = \{ \mathcal{A}(x, G(x)) \mid x \in \mathbb{F}_q \} ,$$

where $\mathcal{A}$ is an affine permutation, $\mathcal{A}(x) = \mathcal{L}(x) + c$.

Important things to remember!

- Verification is the same: if $y \leftarrow F(x)$, $v \leftarrow G(u)$

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

- The degree is not preserved.

CCZ-equivalence

Definition [Carlet, Charpin, Zinoviev, DCC98]

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

$$\Gamma_F = \left\{ (x, F(x)) \mid x \in \mathbb{F}_q \right\} = \mathcal{A}(\Gamma_G) = \left\{ \mathcal{A}(x, G(x)) \mid x \in \mathbb{F}_q \right\},$$

where $\mathcal{A}$ is an affine permutation, $\mathcal{A}(x) = \mathcal{L}(x) + c$.

Important things to remember!

- Verification is the same: if $y \leftarrow F(x)$, $v \leftarrow G(u)$

$$\boxed{y == F(x)? \iff v == G(u)?}$$

- The degree is not preserved.

1. Preliminaries
   - Emerging uses in symmetric cryptography
   - CCZ-equivalence
2. `Anemoi`: a new family of hash-functions
   - New S-box: `Flystel`
   - New mode: `Jive`
   - Comparison to previous work
3. Conclusions

Why Anemoi?

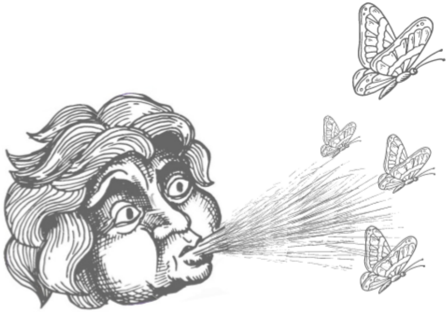
- **Anemoi**
  Family of ZK-friendly Hash functions

Why Anemoi?

- Anemoi
  Family of ZK-friendly Hash functions

  $\Downarrow$

- Anemoi
  Greek gods of winds

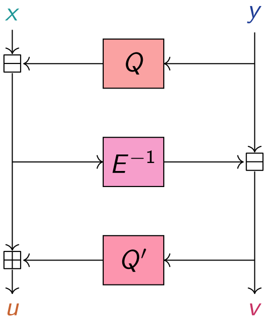


The Flystel

$\boxed{\text{Butterfly} + \text{Feistel} \Rightarrow \texttt{Flystel}}$

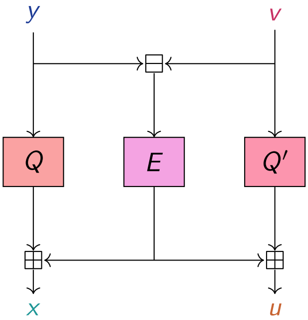
A 3-round Feistel-network with
$Q : \mathbb{F}_q \to \mathbb{F}_q$ and $Q' : \mathbb{F}_q \to \mathbb{F}_q$ two quadratic functions, and $E : \mathbb{F}_q \to \mathbb{F}_q$ a permutation

High-degree permutation



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

Low-degree function



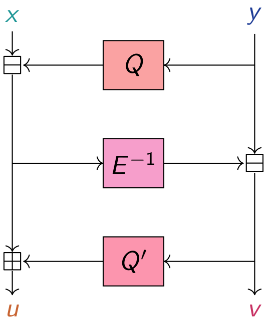
Closed `Flystel` $\mathcal{V}$.

The Flystel

$\mathcal{H}$ and $\mathcal{V}$
are CCZ-equivalent

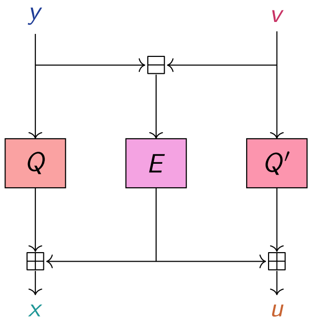
$$\Gamma_{\mathcal{H}} = \{(\ (x,y),\ \mathcal{H}((x,y))\ )\ |\ (x,y) \in \mathbb{F}_q^2\}$$
$$= \mathcal{A}\left(\{(\ (v,y),\ \mathcal{V}((v,y))\ )\ |\ (v,y) \in \mathbb{F}_q^2\}\right) = \mathcal{A}(\Gamma_{\mathcal{V}})$$

High-degree
permutation



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

Low-degree
function

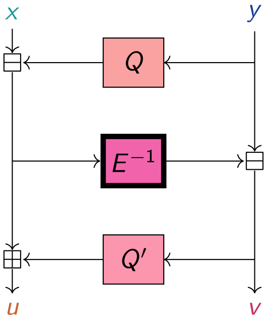


Closed `Flystel` $\mathcal{V}$.

Advantage of CCZ-equivalence

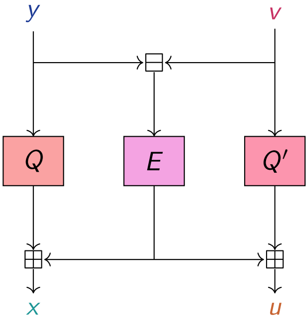
⋆ High Degree Evaluation.

High-degree permutation



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

Low-degree function



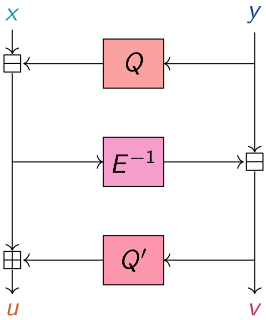
Closed `Flystel` $\mathcal{V}$.

Advantage of CCZ-equivalence

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

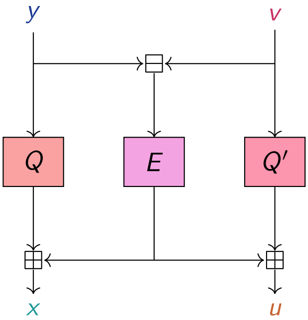
$$\boxed{(u, v) == \mathcal{H}(x, y) \Leftrightarrow (x, u) == \mathcal{V}(y, v)}$$

High-degree permutation



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

Low-degree function

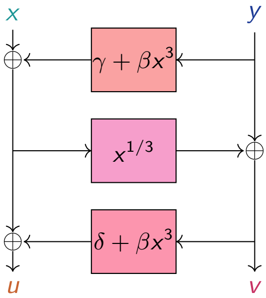


Closed `Flystel` $\mathcal{V}$.

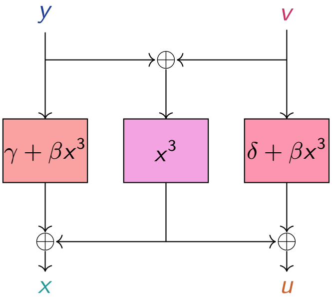
Flystel in $\mathbb{F}_{2^n}$

Well-studied butterfly. First introduced by [Perrin et al. 2016].

$$Q : \mathbb{F}_{2^n} \rightarrow \mathbb{F}_{2^n}, x \mapsto \gamma + \beta x^3 \qquad Q' : \mathbb{F}_{2^n} \rightarrow \mathbb{F}_{2^n}, x \mapsto \delta + \beta x^3 \qquad E : \mathbb{F}_{2^n} \rightarrow \mathbb{F}_{2^n}, x \mapsto x^3$$



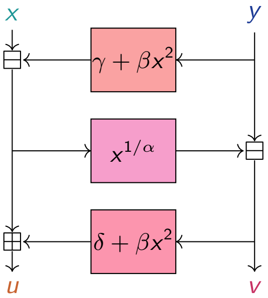
Open `Flystel`$_2$.



Closed `Flystel`$_2$.

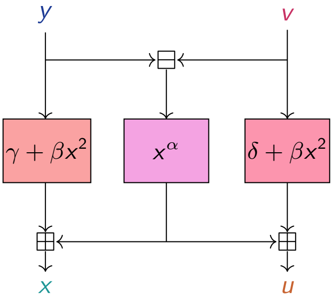
Flystel in $\mathbb{F}_p$

$$Q : \mathbb{F}_p \to \mathbb{F}_p, x \mapsto \gamma + \beta x^2 \qquad Q' : \mathbb{F}_p \to \mathbb{F}_p, x \mapsto \delta + \beta x^2 \qquad E : \mathbb{F}_p \to \mathbb{F}_p, x \mapsto x^\alpha$$



Open `Flystel`$_p$.

usually
$\alpha = 3$ or 5.



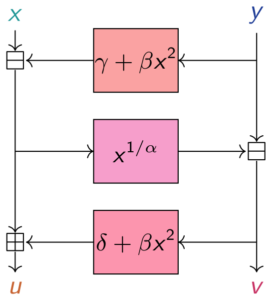
Closed `Flystel`$_p$.

Flystel in $\mathbb{F}_p$

$$Q : \mathbb{F}_p \to \mathbb{F}_p, x \mapsto \gamma + \beta x^2 \qquad Q' : \mathbb{F}_p \to \mathbb{F}_p, x \mapsto \delta + \beta x^2 \qquad E : \mathbb{F}_p \to \mathbb{F}_p, x \mapsto x^\alpha$$

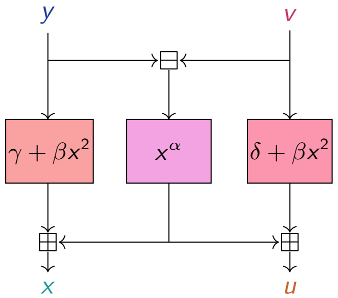
Example Curve BLS12-381: $\alpha = 5$

$\alpha^{-1} = 32019276441773339147342318605887233252455062559512063082656465088992253203926702915541501033032125312303154180478 29$



Open $\texttt{Flystel}_p$.

usually
$\alpha = 3$ or 5.



Closed $\texttt{Flystel}_p$.

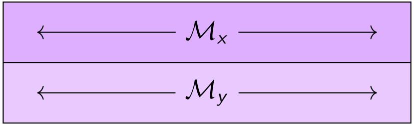
The SPN Structure

SPN: Substitution-Permutation Network

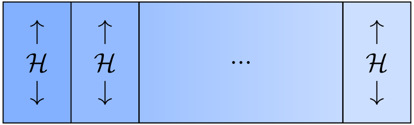
The internal state of `Anemoi` and its basic operations:

X	x_0	x_1	$\cdots$	$x_{\ell-1}$
Y	y_0	y_1	$\cdots$	$y_{\ell-1}$

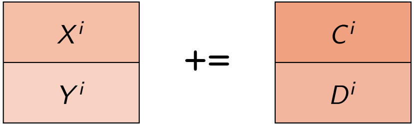
(a) *Internal state*



(b) *The diffusion layer (matrix multiplication).*

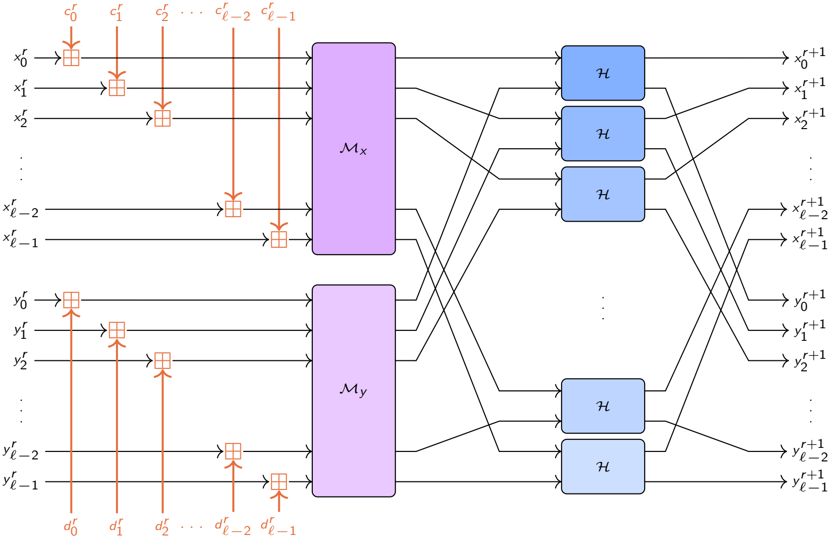


(c) *The confusion or S-box layer $\mathcal{H}$ (the `Flystel`).*



(d) *The constant addition.*

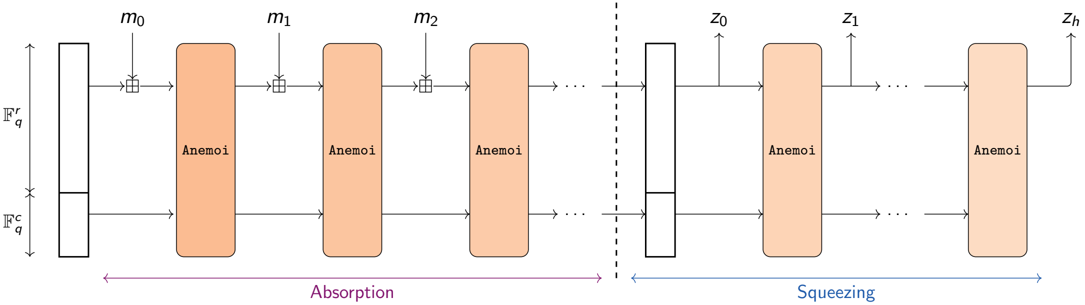
The SPN Structure



Overview of `Anemoi`.

New Mode

- Hash function:
  - input: arbitrary length
  - ouput: fixed length

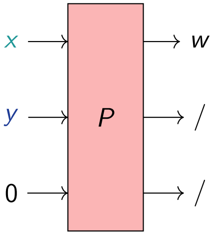


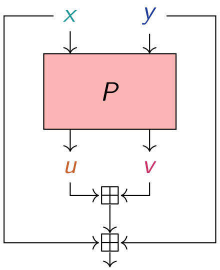
New Mode

- Hash function:
  - input: arbitrary length
  - ouput: fixed length
- Compression function:
  - input: fixed length
  - output: length 1

Dedicated mode $\Rightarrow$ 2 words in 1

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





Some Benchmarks

	m	*Rescue'*	Poseidon	Griffin	Anemoi
R1CS	2	208	198	-	**76**
	4	224	232	112	**96**
	6	216	264	-	**120**
	8	256	296	176	**160**
Plonk	2	312	380	-	**173**
	4	560	1336	291	**220**
	6	756	3024	-	**320**
	8	1152	5448	635	**456**
AIR	2	156	300	-	**114**
	4	168	348	168	**144**
	6	**162**	396	-	180
	8	**192**	480	264	240

(a) *when* $\alpha = 3$.

	m	*Rescue'*	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	-	**192**
	4	528	1032	253	**244**
	6	768	2265	-	**350**
	8	1280	4003	543	**496**
AIR	2	200	360	-	**190**
	4	**220**	440	**220**	240
	6	**240**	540	-	300
	8	**320**	640	360	400

(b) *when* $\alpha = 5$.

Constraint comparison for Rescue–Prime, Poseidon, Griffin *and* Anemoi *(we fix* $s = 128$*).*

Some Benchmarks

	m	*Rescue′*	Poseidon	Griffin	`Anemoi`
R1CS	2	208	198	-	**76**
	4	224	232	112	**96**
	6	216	264	-	**120**
	8	256	296	176	**160**
Plonk	2	312	380	-	**173**
	4	560	1336	291	**220**
	6	756	3024	-	**320**
	8	1152	5448	635	**456**
AIR	2	156	300	-	**114**
	4	168	348	168	**144**
	6	**162**	396	-	180
	8	**192**	480	264	240

(a) *when* $\alpha = 3$.

	m	*Rescue′*	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	-	**192**
	4	528	1032	253	**244**
	6	768	2265	-	**350**
	8	1280	4003	543	**496**
AIR	2	200	360	-	**190**
	4	**220**	440	**220**	240
	6	**240**	540	-	300
	8	**320**	640	360	400

(b) *when* $\alpha = 5$.

Constraint comparison for Rescue–Prime, Poseidon, Griffin *and* `Anemoi` *(we fix* $s = 128$*).*

Conclusions

- A new family of ZK-friendly hash functions:

  ⇒ Anemoi efficient accross proof system

- New observations of fundamental interest:
  - Standalone components:
    - New S-box: Flystel
    - New mode: Jive
  - Identify a link between AO and CCZ-equivalence

☞ More details on eprint.iacr.org/2022/840

Conclusions

- A new family of ZK-friendly hash functions:
  ⇒ Anemoi efficient accross proof system
- New observations of fundamental interest:
  - Standalone components:
    - New S-box: Flystel
    - New mode: Jive
  - Identify a link between AO and CCZ-equivalence

☞ More details on eprint.iacr.org/2022/840

Cryptanalysis and designing of arithmetization-oriented primitives remain to be explored!

Conclusions

- ⋆ A new family of ZK-friendly hash functions:
  $\Rightarrow$ Anemoi efficient accross proof system

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

☞ More details on eprint.iacr.org/2022/840

Cryptanalysis and designing of arithmetization-oriented primitives remain to be explored!

Thanks for your attention!