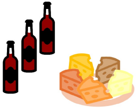
Attacks on AOP (Arithmetization-Oriented Primitives)

When cryptanalysis becomes lucrative!

Clémence Bouvier

Université de Lorraine, CNRS, Inria, LORIA



École d'hiver, Autrans, France
January 22nd, 2025

AOP

AOP: “Appellation d’origine protégée”

Bleu du Vercors-Sassenage

Sudoku

	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

Sudoku

	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	

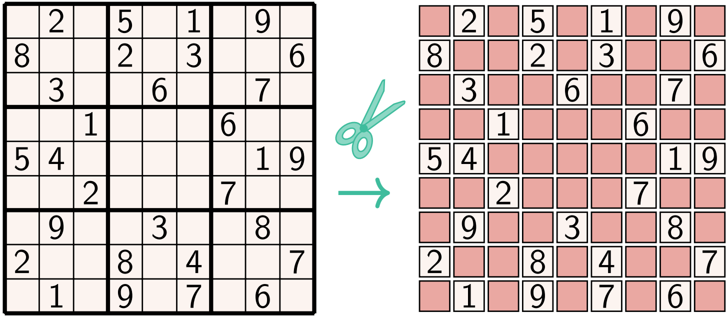
Unsolved Sudoku

→

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

Sudoku



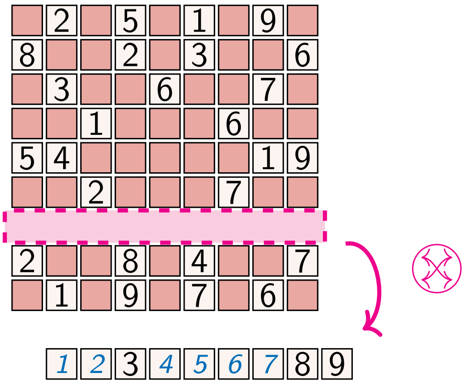
Unsolved Sudoku

Grid cutting

Sudoku

	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

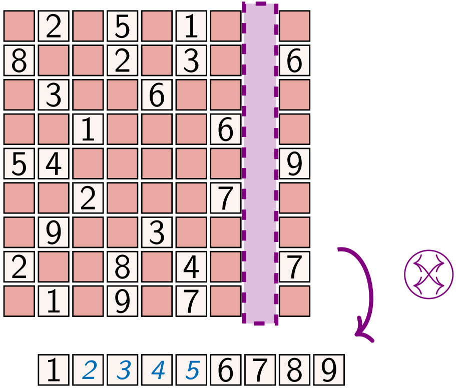


Rows checking

Sudoku

	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

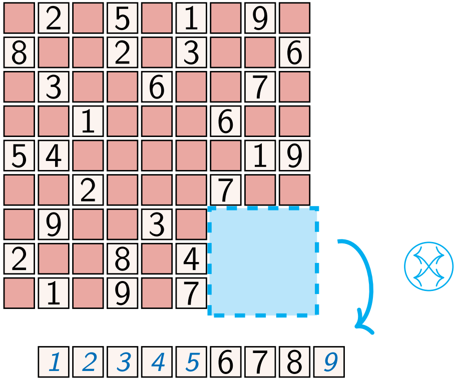


Columns checking

Sudoku

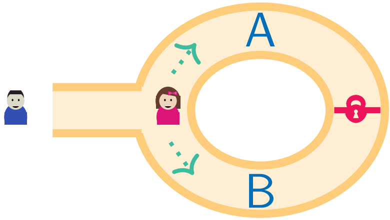
	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

Unsolved Sudoku

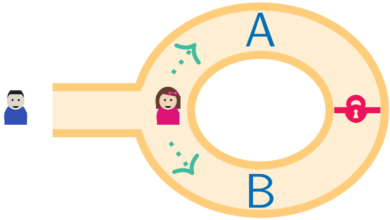


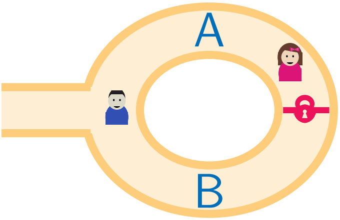
Squares checking

Ali-Baba cave

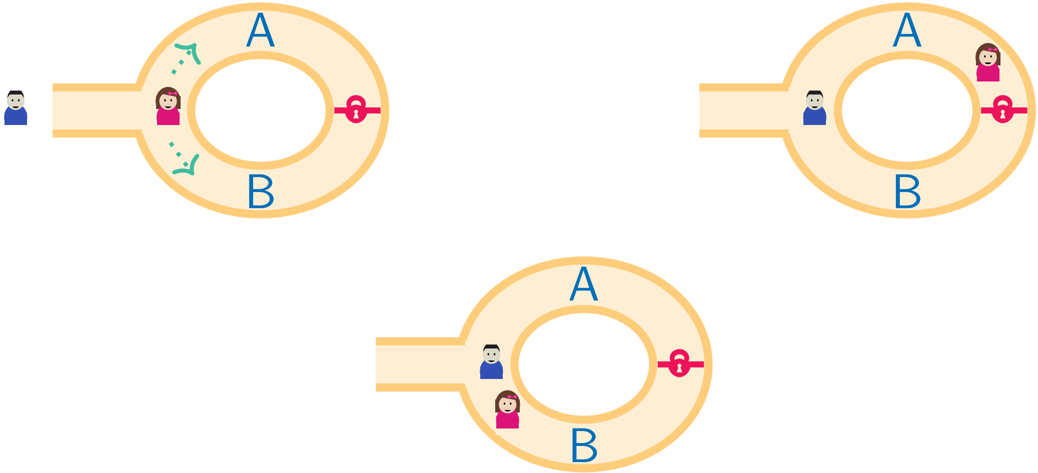


Ali-Baba cave

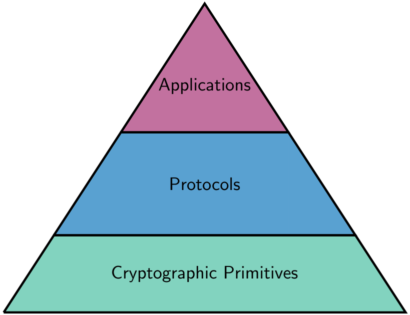




Ali-Baba cave



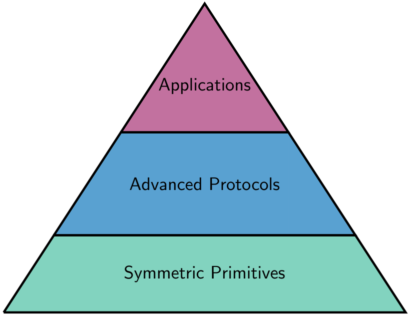
A need for new primitives



A need for new primitives

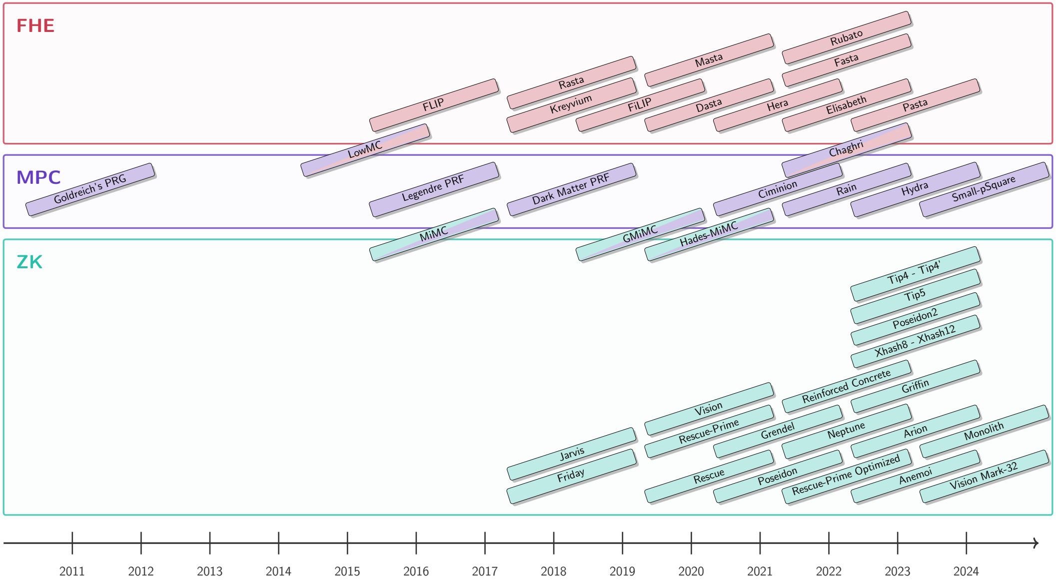
Protocols requiring new primitives:

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



Problem: Designing new symmetric primitives

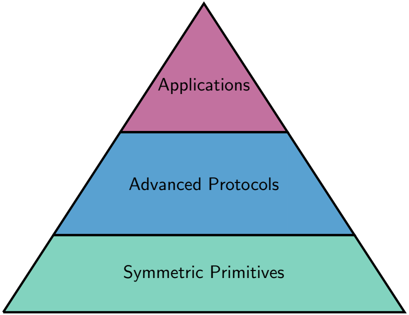
Primitives


FHE
MPC
ZK
Goldreich's PRG
LowMC
FLIP
Legendre PRF
MiMC
Rasta
Kreyvium
Dark Matter PRF
Jarvis
Friday
FiLIP
GMiMC
Masta
Dasta
Hades-MiMC
Vision
Rescue-Prime
Rescue
Hera
Ciminion
Grendel
Poseidon
Rubato
Fasta
Elisabeth
Chaghri
Rain
Reinforced Concrete
Neptune
Rescue-Prime Optimized
Pasta
Hydra
Tip4 - Tip4'
Tip5
Poseidon2
Xhash8 - Xhash12
Griffin
Arion
Anemoi
Small-pSquare
Monolith
Vision Mark-32
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024

A need for new primitives

Protocols requiring new primitives:

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



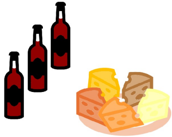
Problem: Designing new symmetric primitives

And analyse their security!

Content

⋆ Symmetric cryptography and cryptanalysis tools

⋆ Introduction of AOP



⋆ Attacks against AOP

Block ciphers

- input: n-bit block

$$x \in \mathbb{F}_2^n$$

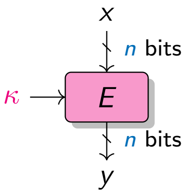
- parameter: k-bit key

$$\kappa \in \mathbb{F}_2^k$$

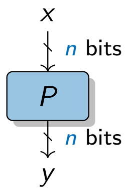
- output: n-bit block

$$y = E_\kappa(x) \in \mathbb{F}_2^n$$

- symmetry: E and E^{-1} use the same κ



(a) *Block cipher*



(b) *Random permutation*

Block ciphers

- input: n-bit block

$$x \in \mathbb{F}_2^n$$

- parameter: k-bit key

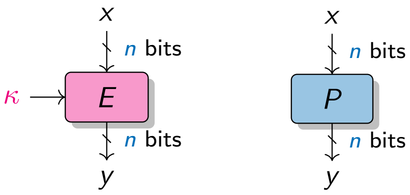
$$\kappa \in \mathbb{F}_2^k$$

- output: n-bit block

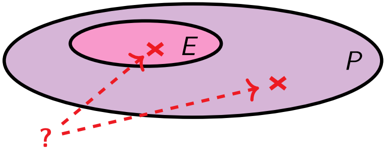
$$y = E_\kappa(x) \in \mathbb{F}_2^n$$

- symmetry: E and E^{-1} use the same κ

A block cipher is a family of 2^k permutations of $\mathbb{F}_2^n$.



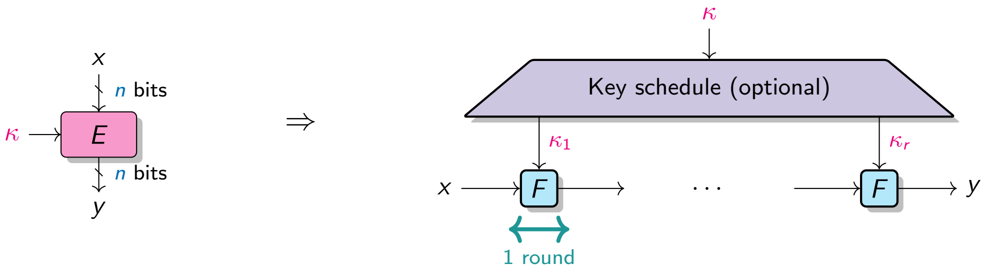
(a) *Block cipher* **(b)** *Random permutation*



Iterated constructions

How to build an efficient block cipher?

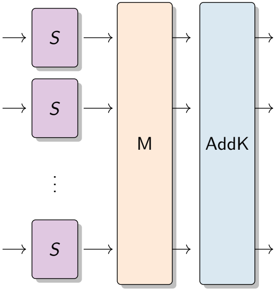
By iterating a round function.



Performance constraints! The primitive must be fast.

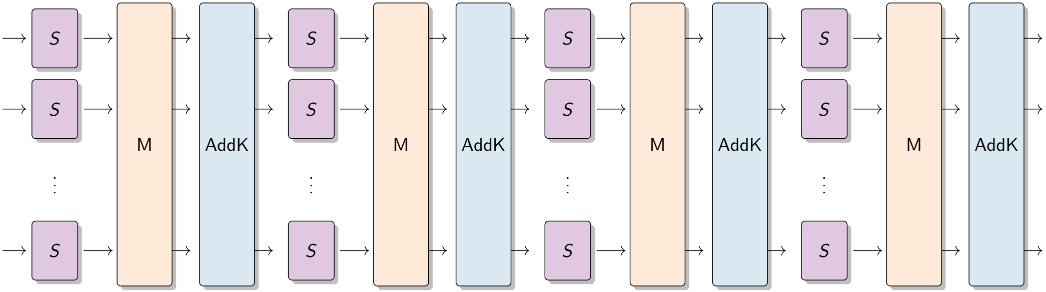
SPN construction

SPN = Substitution Permutation Networks



SPN construction

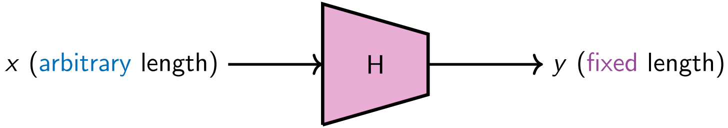
SPN = Substitution Permutation Networks



Hash functions

Definition

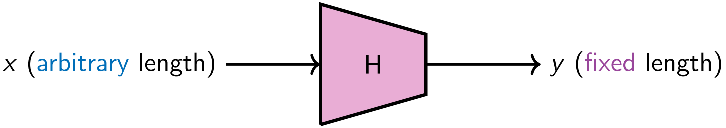
Hash function: $H : \mathbb{F}_q^{\ell} \to \mathbb{F}_q^{h}, x \mapsto y = H(x)$ where ℓ is arbitrary and h is fixed.



Hash functions

Definition

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



⋆ **Preimage resistance**: Given y it must be *infeasible* to find x s.t. $H(x) = y$.

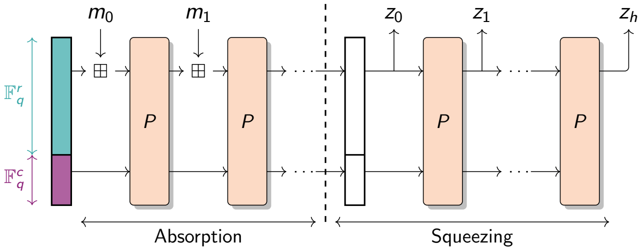
⋆ **Collision resistance**: It must be *infeasible* to find $x \neq x'$ s.t. $H(x) = H(x')$.

Sponge construction

Sponge construction

Parameters:

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

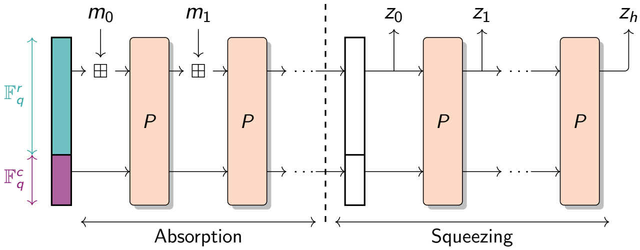


Sponge construction

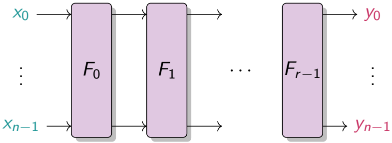
Sponge construction

Parameters:

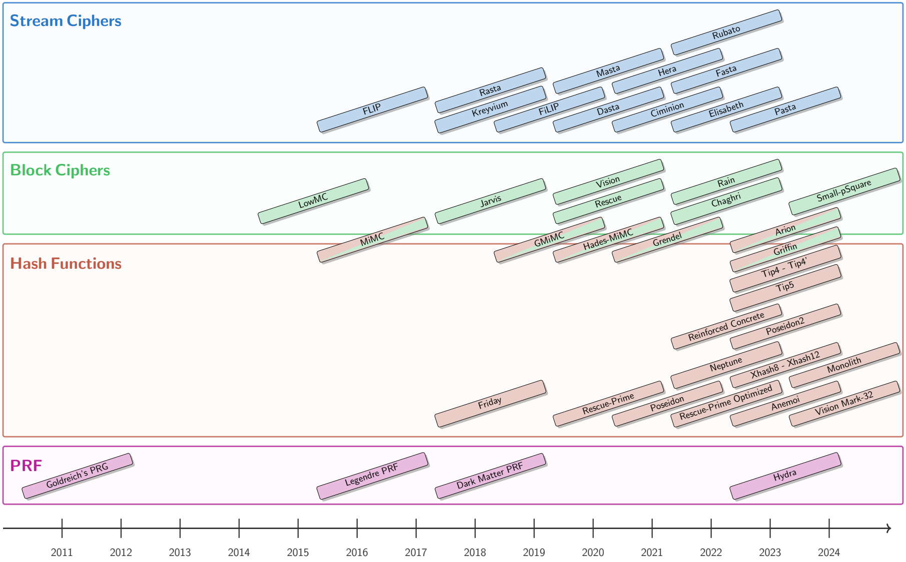
- ⋆ rate $r > 0$
- ⋆ capacity $c > 0$
- ⋆ permutation of $\mathbb{F}_q^n$ $(n = r + c)$



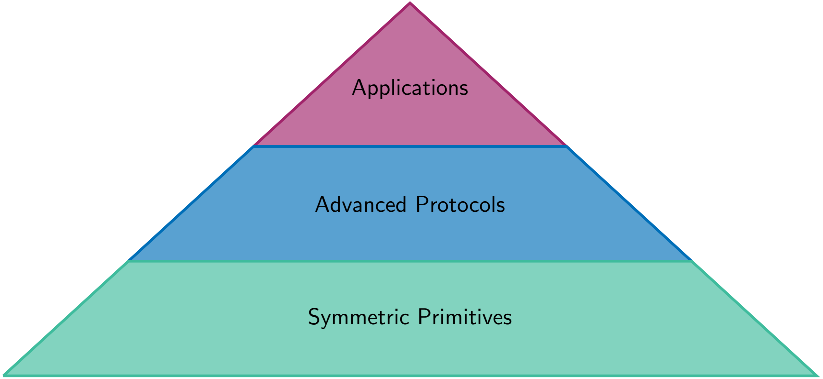
P is an iterated construction



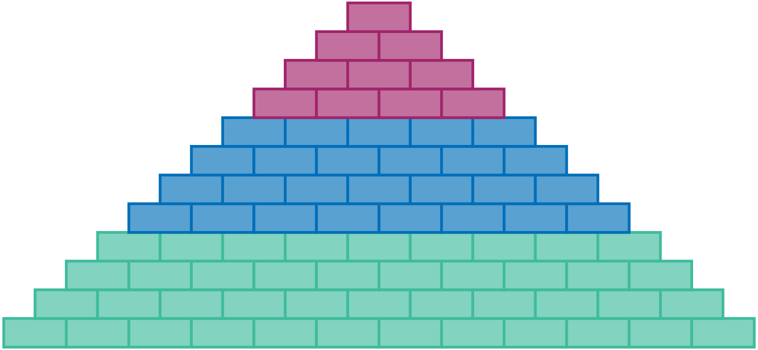
Primitives


Stream Ciphers
FLIP
Rasta
Kreyvium
FiLIP
Masta
Dasta
Hera
Ciminion
Rubato
Fasta
Elisabeth
Pasta
Block Ciphers
LowMC
MiMC
Jarvis
GMiMC
Vision
Rescue
Hades-MiMC
Grendel
Rain
Chaghri
Small-pSquare
Arion
Griffin
Hash Functions
Tip4 - Tip4'
Tip5
Reinforced Concrete
Poseidon2
Neptune
Xhash8 - Xhash12
Monolith
Friday
Rescue-Prime
Poseidon
Rescue-Prime Optimized
Anemoi
Vision Mark-32
PRF
Goldreich's PRG
Legendre PRF
Dark Matter PRF
Hydra
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024

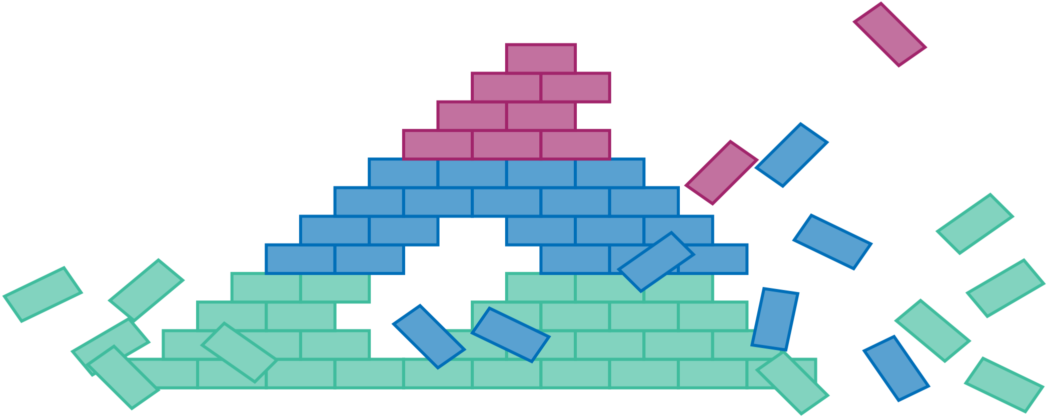
Building blocks of security



Building blocks of security



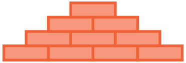
Cycle primitive



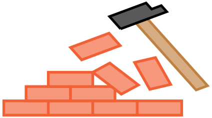
Primitive life cycle

Conception

- ⋆ Specification of the algorithm
- ⋆ Justification of design choices
- ⋆ First security analysis



Publication



Analysis

- ⋆ Trying to break algorithms

Standardization

Deployment

- ⋆ Implementation of algorithms

CICO problem

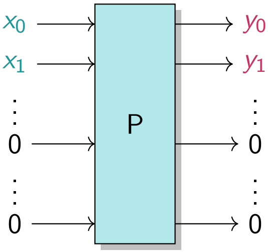
CICO: Constrained Input Constrained Output

Definition

Let $P : \mathbb{F}_q^{r+c} \rightarrow \mathbb{F}_q^{r+c}$. The **CICO** problem is:

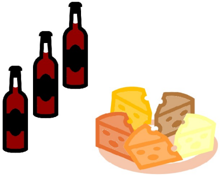
Finding $X, Y \in \mathbb{F}_q^r$ s.t.

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



Content

Introduction of AOP



A new environment

Traditional case

Operations based on logical gates or CPU instructions.

$$\mathbb{F}_2^n, \text{ with } n \simeq 4, 8$$

Example

Field of AES

$$\mathbb{F}_2^n, \text{ where } n = 8$$

$$(0,0,0,0,0,0,0,0),$$
$$(0,0,0,0,0,0,0,1),$$
$$\ldots$$
$$(1,1,1,1,1,1,1,1)$$

A new environment

Traditional case

Operations based on logical gates or CPU instructions.

$$\mathbb{F}_2^n, \text{ with } n \simeq 4, 8$$

Arithmetization-Oriented

Operations based on large finite-field arithmetic.

$$\mathbb{F}_q, \text{ with } q \in \{2^n, p\}, p \simeq 2^n, n \geq 32$$

Example

Field of AES

$$\mathbb{F}_2^n, \text{ where } n = 8$$

$$(0,0,0,0,0,0,0,0),$$
$$(0,0,0,0,0,0,0,1),$$
$$\dots$$
$$(1,1,1,1,1,1,1,1)$$

Example

Scalar Field of Curve BLS12-381

$$\mathbb{F}_p, \text{ where}$$

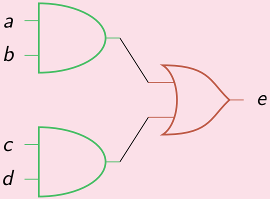
$$p = \texttt{0x73eda753299d7d483339d80809a1d805}$$
$$\texttt{53bda402fffe5bfeffffffff00000001}$$

$$0, 1, 2, \dots, p-1$$

New operations

Traditional case

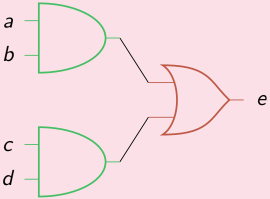
Use of logical gates and CPU instructions.



New operations

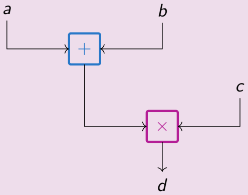
Traditional case

Use of logical gates and CPU instructions.



Arithmetization-Oriented

Use of Arithmetic circuit.

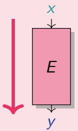


A new metric

Traditional case

Minimize time and memory.

$$y \leftarrow E(x)$$

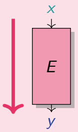


A new metric

Traditional case

Minimize time and memory.

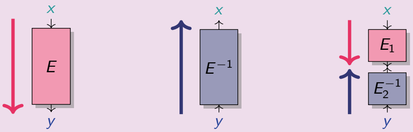
$$y \leftarrow E(x)$$



Arithmetization-Oriented

Minimize the number of multiplications.

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

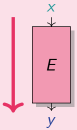


A new metric

Traditional case

Minimize time and memory.

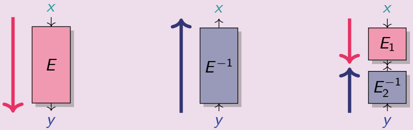
$$y \leftarrow E(x)$$



Arithmetization-Oriented

Minimize the number of multiplications.

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



Example

Let $E : \mathbb{F}_{11} \to \mathbb{F}_{11}, x \mapsto x^3$. We have $E^{-1} : \mathbb{F}_{11} \to \mathbb{F}_{11}, x \mapsto x^7$.

Evaluation: Given $x = 5$, compute $y = E(x)$.

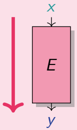
$$y = 4 \text{ (applying } E\text{)}$$

A new metric

Traditional case

Minimize time and memory.

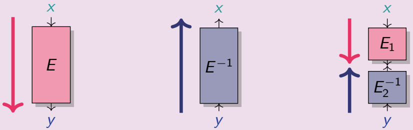
$$y \leftarrow E(x)$$



Arithmetization-Oriented

Minimize the number of multiplications.

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



Example

Let $E : \mathbb{F}_{11} \rightarrow \mathbb{F}_{11}, x \mapsto x^3$. We have $E^{-1} : \mathbb{F}_{11} \rightarrow \mathbb{F}_{11}, x \mapsto x^7$.

Verification: Given $x = 5$ and $y = 4$, check if $y = E(x)$.

$$5^3 = 4 \text{ (applying } E\text{)} \qquad \text{or} \qquad 4^7 = 5 \text{ (applying } E^{-1}\text{)}$$

Take-away

Traditional case

- Alphabet:

$$\mathbb{F}_2^n, \text{ with } n \simeq 4, 8$$

- Operations:
  Logical gates/CPU instructions
- Metric: minimize time and memory for the evaluation
- **Decades of Cryptanalysis**

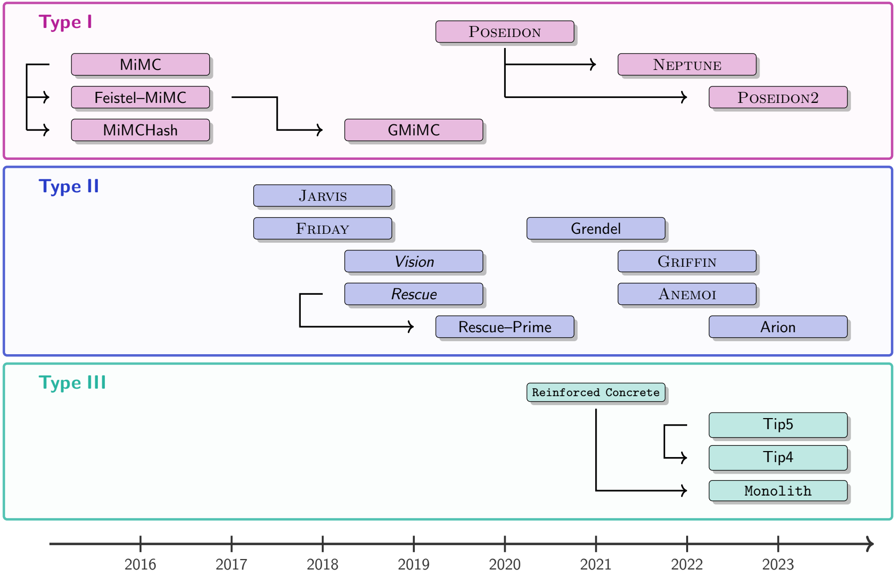
Arithmetization-Oriented

- Alphabet:

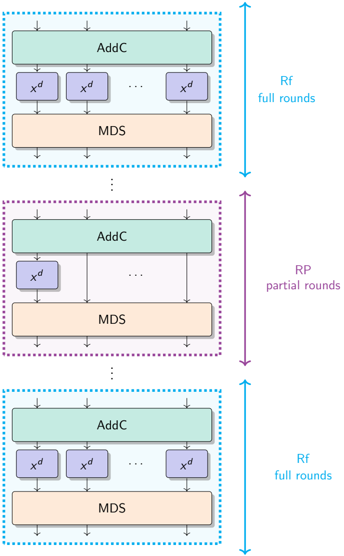
$$\mathbb{F}_q, \text{ with } q \in \{2^n, p\}, p \simeq 2^n, n \geq 32$$

- Operations:
  Large finite-field arithmetic
- Metric: minimize the number of multiplications for the verification
- ≤ 8 **years of Cryptanalysis**

Primitives overview



Example of Type I: POSEIDON



Low degree primitive

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

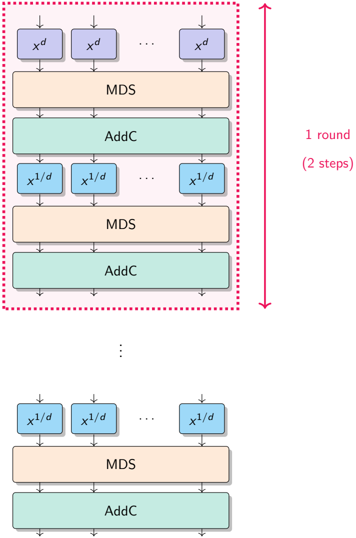
- ⋆ S-box:

$$x \mapsto x^3$$

- ⋆ Nb rounds:

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

Example of Type II: *Rescue*



Primitive based on equivalence

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

- ⋆ S-box:

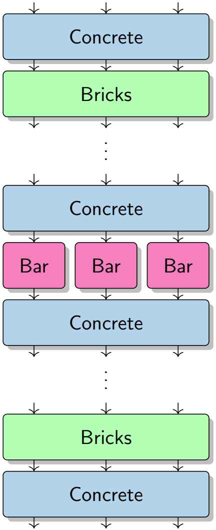
$$x \mapsto x^3 \quad \text{and} \quad x \mapsto x^{1/3}$$

- ⋆ Nb rounds:

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

(2 S-boxes per round)

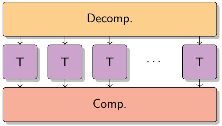
Example of Type III: Reinforced Concrete



Primitive using Look-up-Tables

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

⋆ S-box:



⋆ Nb rounds:

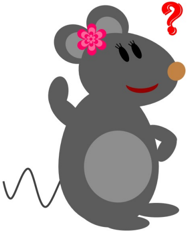
$$R = 7$$

Take-away

	Type I Low-degree primitives	**Type II** Equivalence relation	**Type III** Look-up tables
Alphabet	$\mathbb{F}_q^m$ for various q and m	$\mathbb{F}_q^m$ for various q and m	specific fields
Nb of rounds	many	few	fewer
Plain performance	fast	slow	faster
Nb of constraints	often more	fewer	it depends on the proof system

QUIZ !!

To which type of primitives (I, II, or III) belong AES?



QUIZ !!

Could we use AES for advanced protocols?

Content

Attacks against AOP

CICO Problem

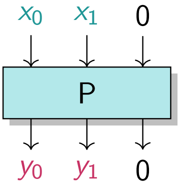
CICO: Constrained Input Constrained Output

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 $t = 3,\ u = 1.$

Need to solve polynomial systems

Solving polynomial systems

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

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

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 $\mathcal{P}_j \in \mathbb{F}_q[X_0, \ldots, X_{n-1}]$

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

Euclidean division

⋆ Integers

$$a = q \times b + r, \ 0 \leq r < b$$

Example: division of 2025 by 100

$$2025 = 20 \times 100 + 25$$

Euclidean division

⋆ Integers

$$a = q \times b + r,\ 0 \leq r < b$$

Example: division of 2025 by 100

$$2025 = 20 \times 100 + 25$$

⋆ Univariate polynomials

$$A = Q \times B + R,\ 0 \leq \deg(R) < \deg(B)$$

Example: division of $X^5 + 2X^3 + 3X$ by X^2

$$X^5 + 2X^3 + 3X = (X^3 + 2X) \times X^2 + 3X$$

Euclidean division

⋆ Integers

$$a = q \times b + r,\ 0 \leq r < b$$

Example: division of 2025 by 100

$$2025 = 20 \times 100 + 25$$

⋆ Univariate polynomials

$$A = Q \times B + R,\ 0 \leq \deg(R) < \deg(B)$$

Example: division of $X^5 + 2X^3 + 3X$ by X^2

$$X^5 + 2X^3 + 3X = (X^3 + 2X) \times X^2 + 3X$$

⋆ Multivariate polynomials

Euclidean division

⋆ Integers

$$a = q \times b + r,\ 0 \leq r < b$$

Example: division of 2025 by 100

$$2025 = 20 \times 100 + 25$$

⋆ Univariate polynomials

$$A = Q \times B + R,\ 0 \leq \deg(R) < \deg(B)$$

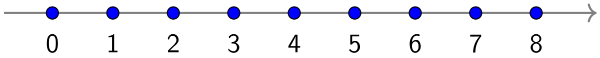
Example: division of $X^5 + 2X^3 + 3X$ by X^2

$$X^5 + 2X^3 + 3X = (X^3 + 2X) \times X^2 + 3X$$

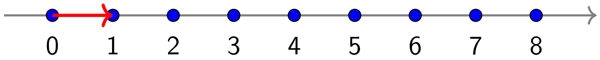
⋆ Multivariate polynomials

Need monomial ordering

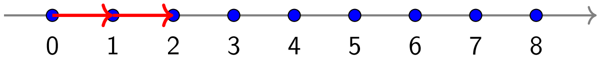
Monomial ordering



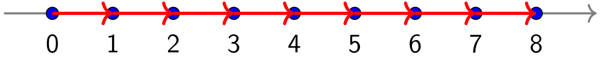
Monomial ordering



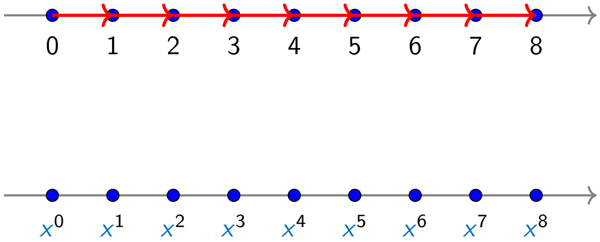
Monomial ordering



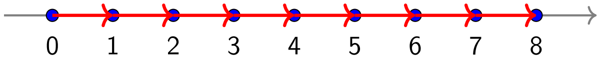
Monomial ordering

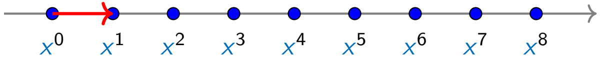


Monomial ordering

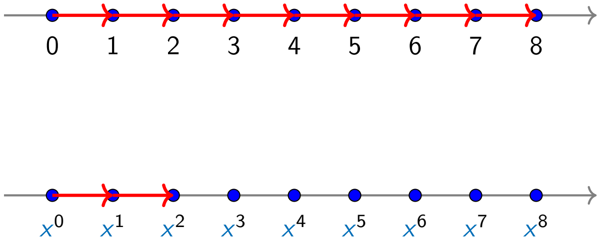


Monomial ordering

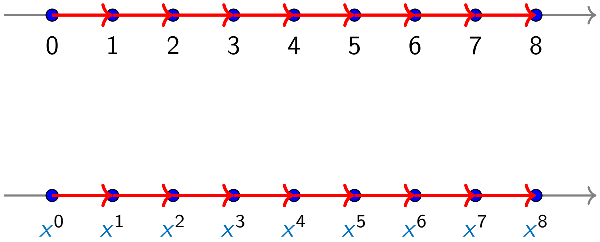




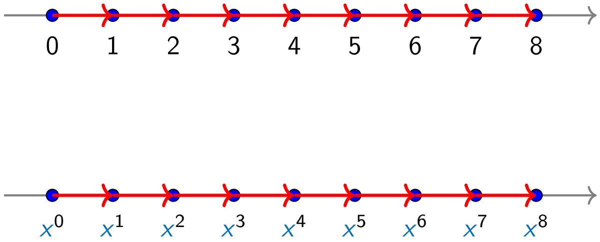
Monomial ordering



Monomial ordering

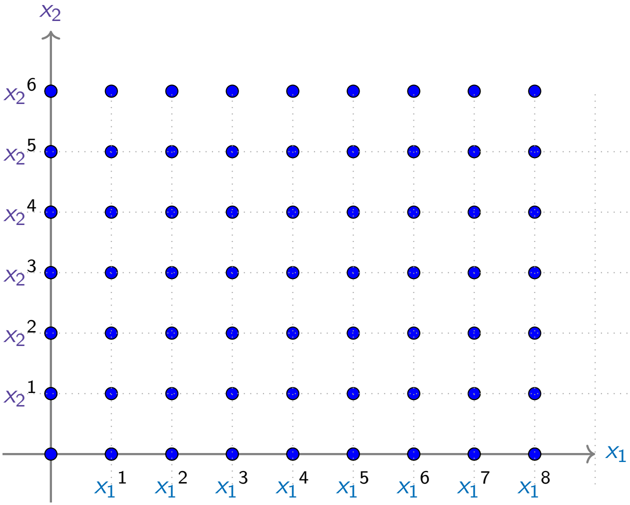


Monomial ordering

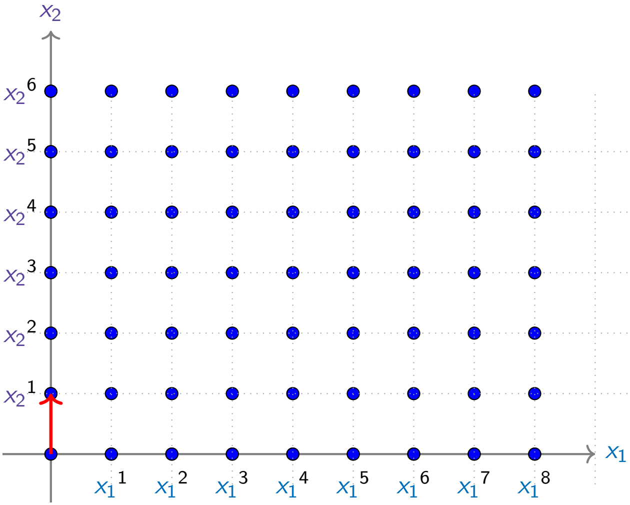


What about the multivariate case?

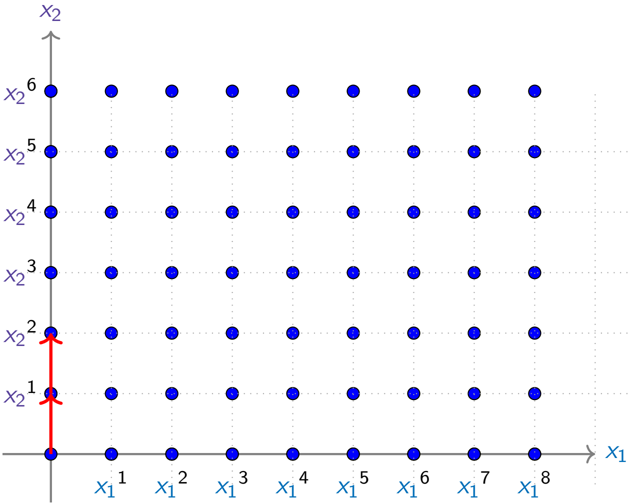
Lexicographical ordering



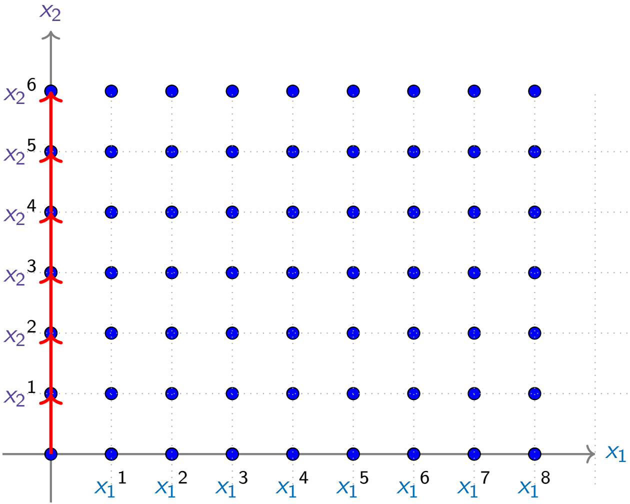
Lexicographical ordering



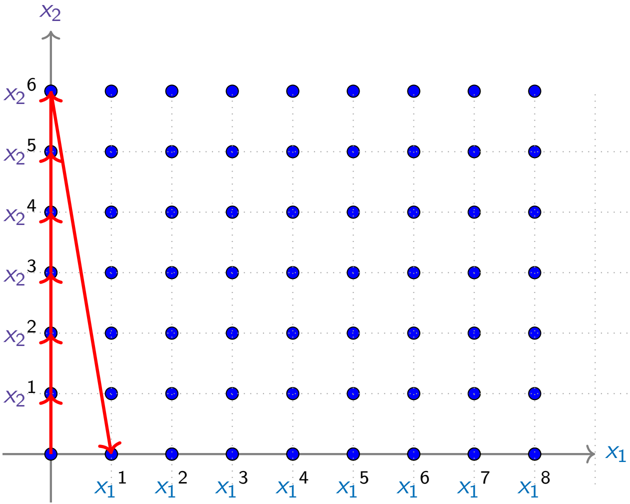
Lexicographical ordering



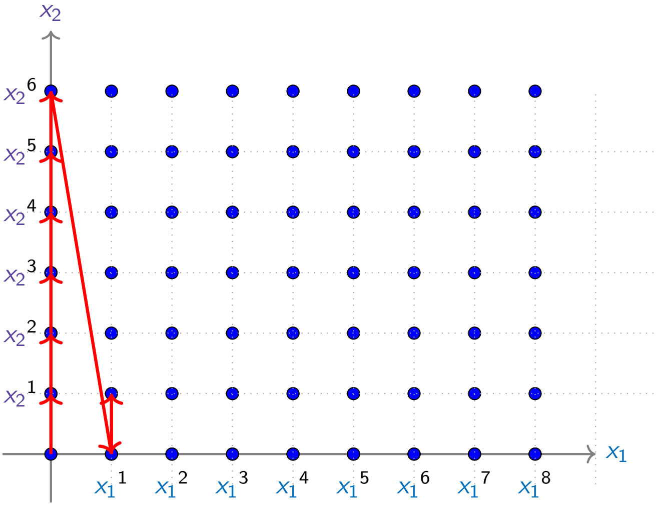
Lexicographical ordering



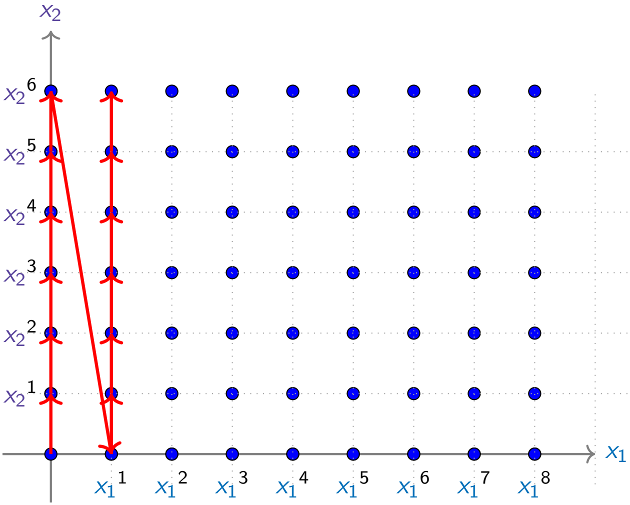
Lexicographical ordering



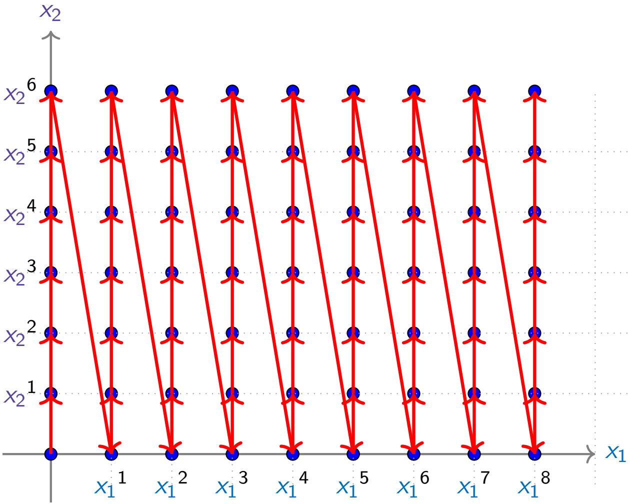
Lexicographical ordering



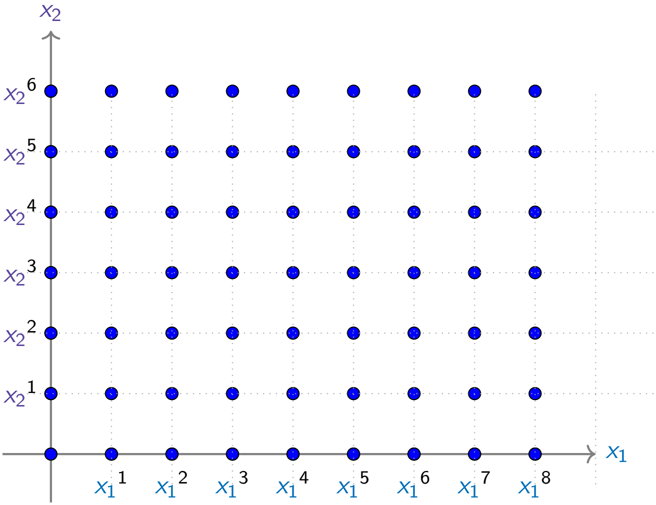
Lexicographical ordering



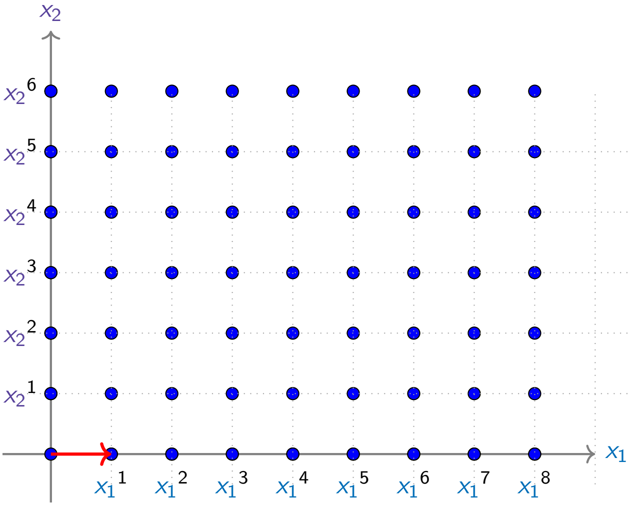
Lexicographical ordering



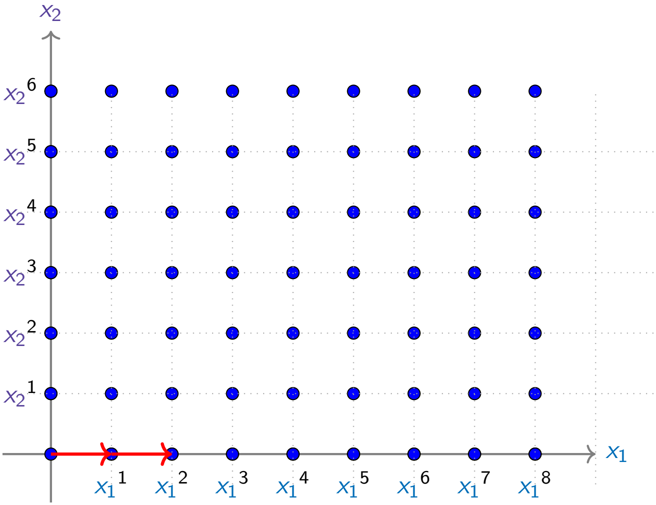
Reverse lex. ordering



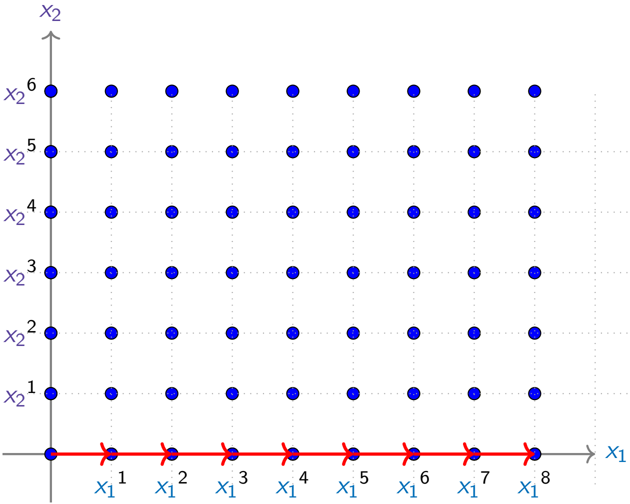
Reverse lex. ordering



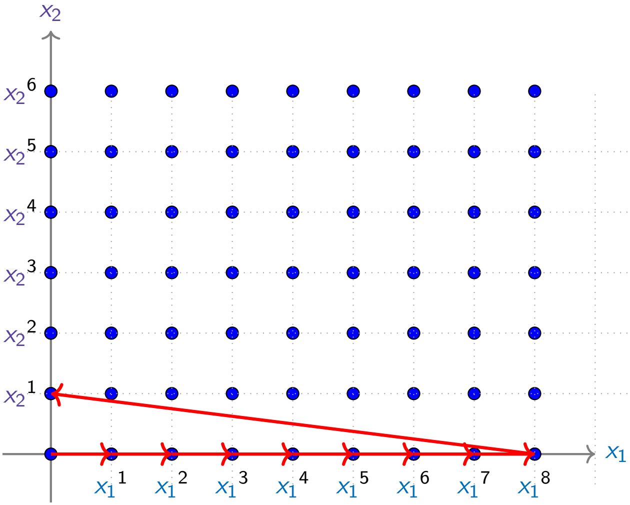
Reverse lex. ordering



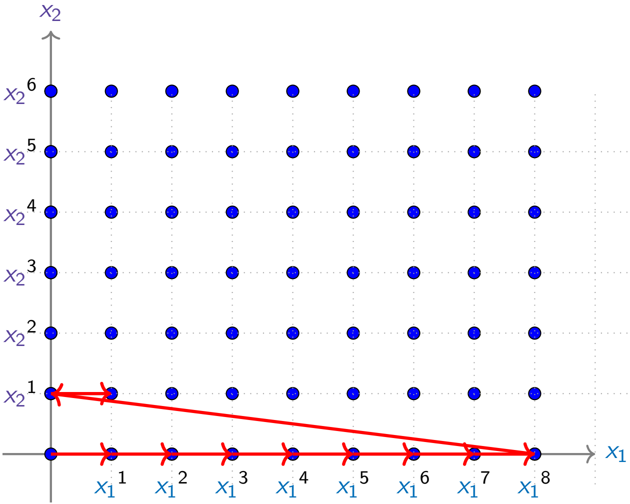
Reverse lex. ordering



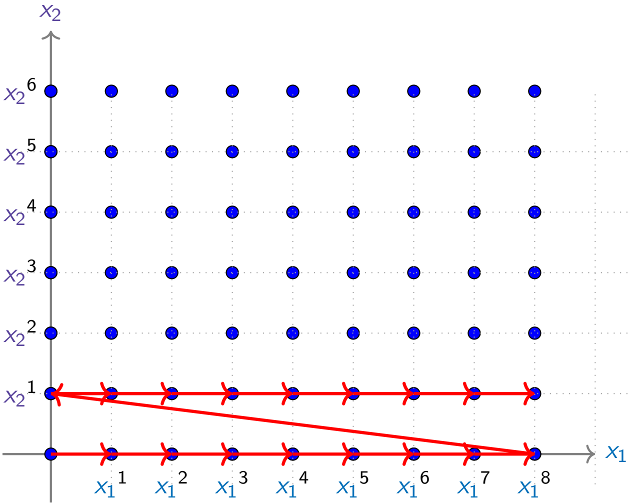
Reverse lex. ordering



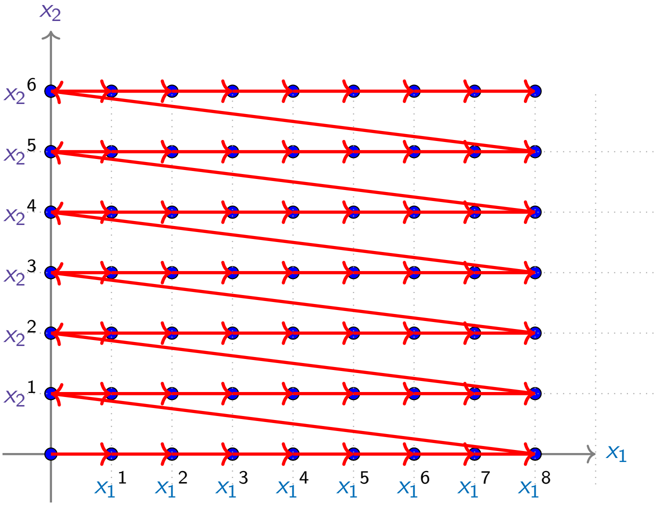
Reverse lex. ordering



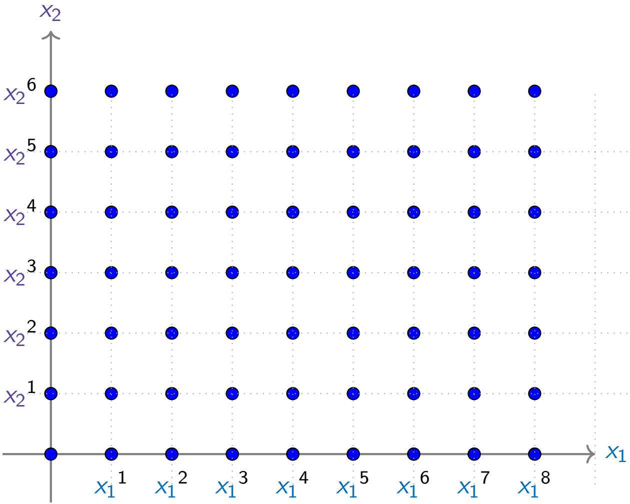
Reverse lex. ordering



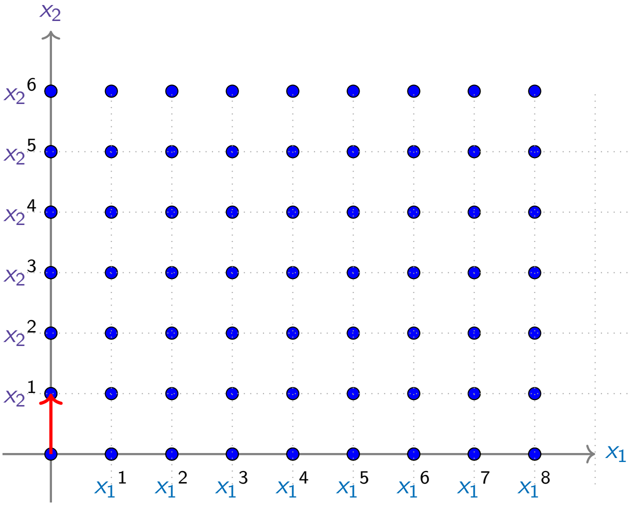
Reverse lex. ordering



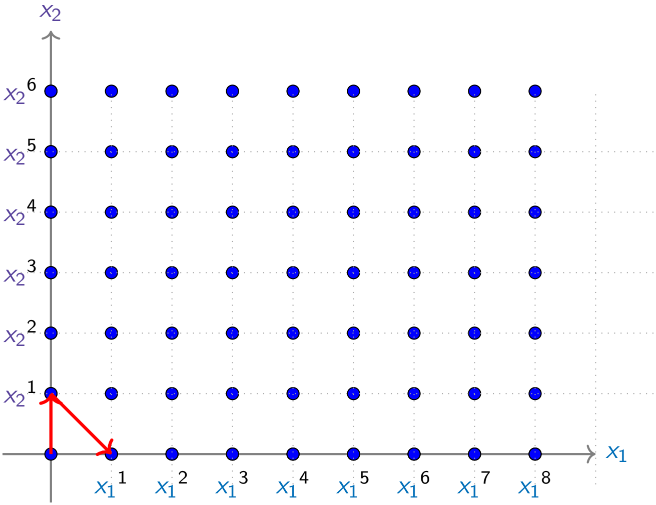
Graded lex. ordering



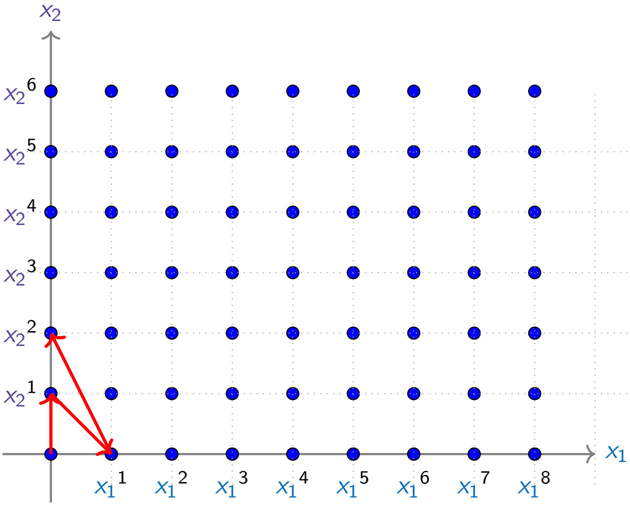
Graded lex. ordering



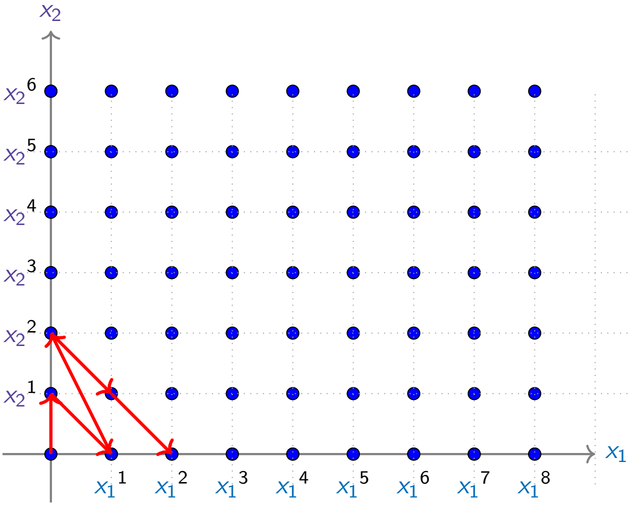
Graded lex. ordering



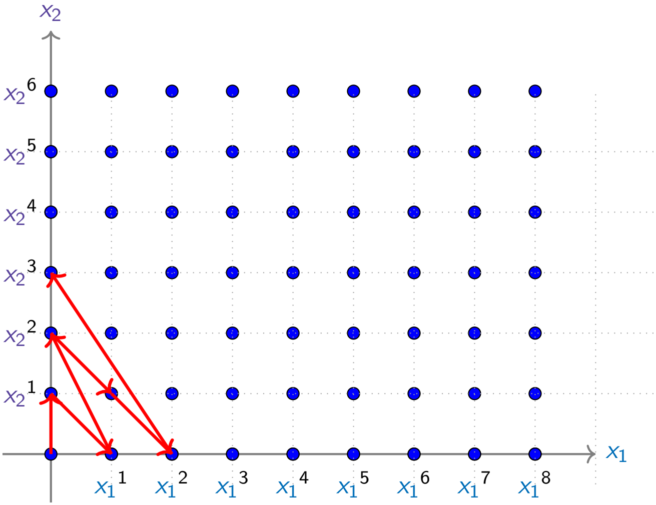
Graded lex. ordering



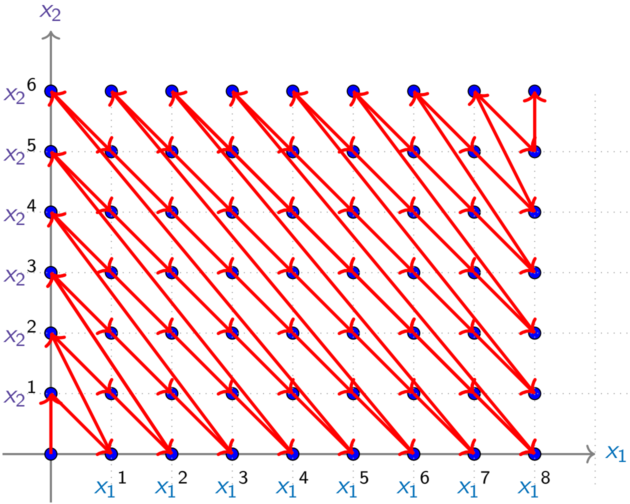
Graded lex. ordering



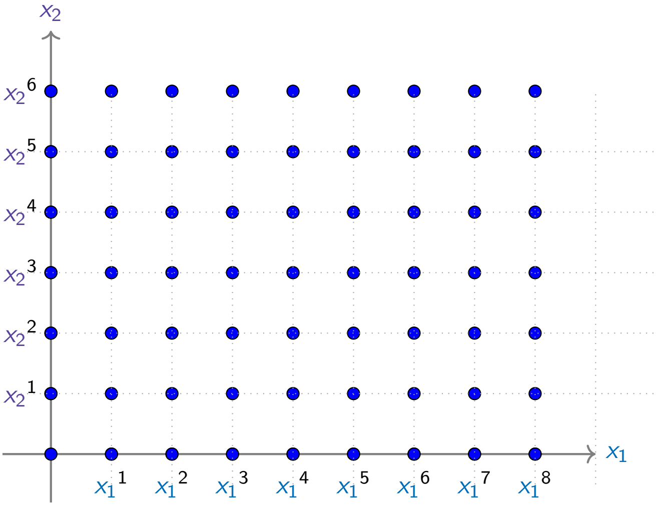
Graded lex. ordering



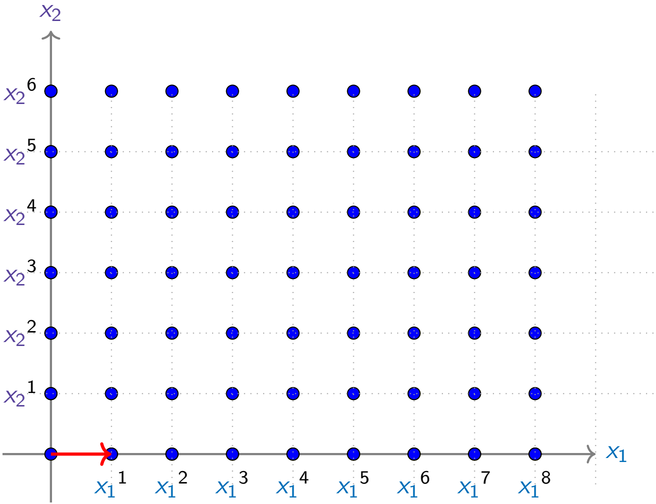
Graded lex. ordering



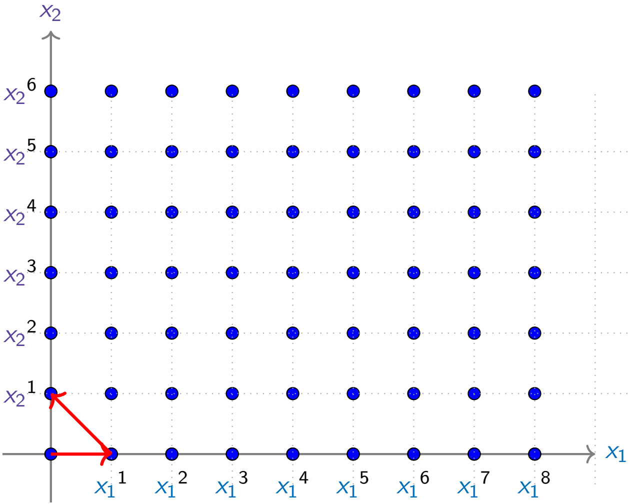
Graded reverse lex. ordering



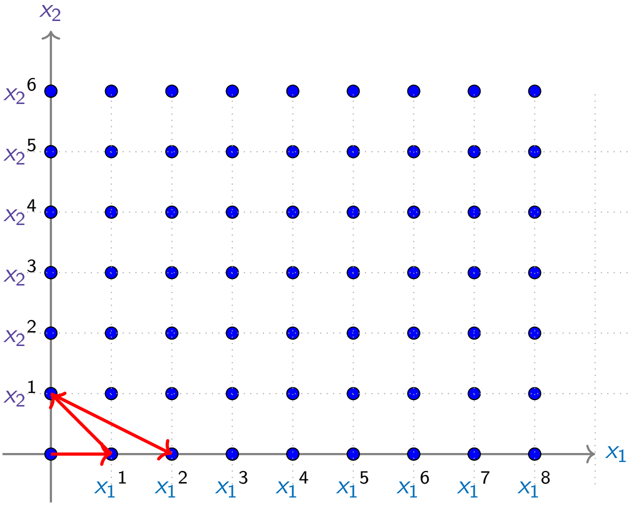
Graded reverse lex. ordering



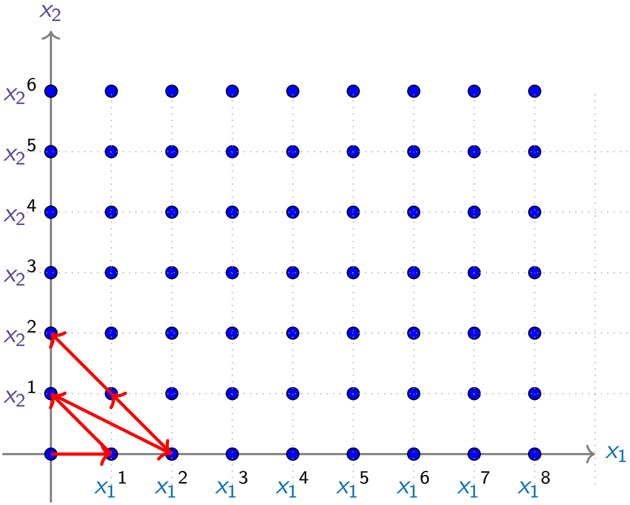
Graded reverse lex. ordering



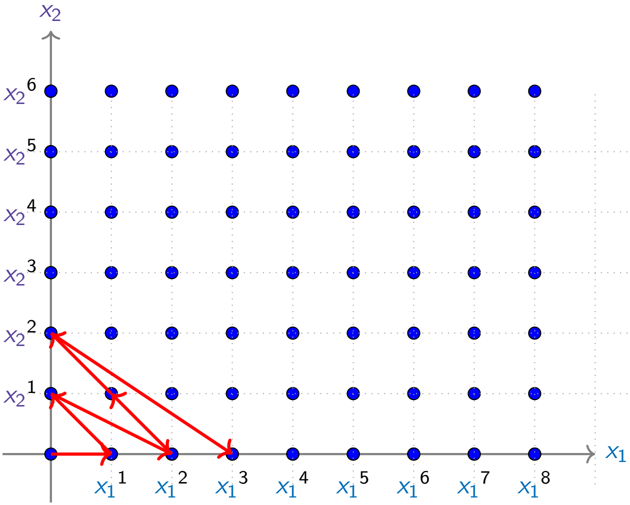
Graded reverse lex. ordering



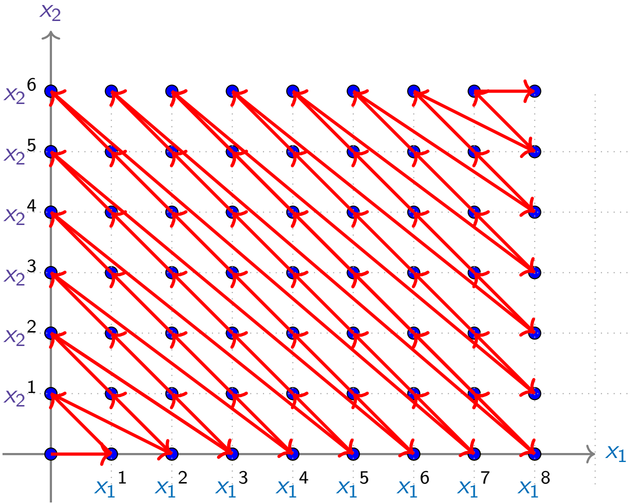
Graded reverse lex. ordering



Graded reverse lex. ordering



Graded reverse lex. ordering



Monomial ordering

Some orderings in $\mathbb{F}_q[x_1, x_2, x_3]$.

Lexicographical order (lex)

First, compare degrees of highest variable, then second variable, ...

$$x_1 > x_2 > x_3, \qquad x_1 > {x_2}^2,$$

$${x_1}^2 x_2 > {x_1}^2 x_3$$

Monomial ordering

Some orderings in $\mathbb{F}_q[x_1, x_2, x_3]$.

Lexicographical order (lex)

First, compare degrees of highest variable, then second variable, ...

$$x_1 > x_2 > x_3, \qquad x_1 > {x_2}^2,$$

$${x_1}^2 x_2 > {x_1}^2 x_3$$

Graded lex. order (grlex)

First, compare total degree, then lex. order if equality.

$$x_1 > x_2 > x_3, \qquad x_1 < {x_2}^2,$$

$${x_1}^2 x_2 > {x_1}^2 x_3$$

Monomial ordering

Some orderings in $\mathbb{F}_q[x_1, x_2, x_3]$.

Lexicographical order (lex)

First, compare degrees of highest variable, then second variable, ...

$$x_1 > x_2 > x_3, \qquad x_1 > {x_2}^2,$$

$${x_1}^2 x_2 > {x_1}^2 x_3$$

Graded lex. order (grlex)

First, compare total degree, then lex. order if equality.

$$x_1 > x_2 > x_3, \qquad x_1 < {x_2}^2,$$

$${x_1}^2 x_2 > {x_1}^2 x_3$$

Graded reverse lex. order (grevlex)

First, compare total degree, then inverse lex. order if equality.

$$x_1 < x_2 < x_3, \qquad x_1 < {x_2}^2,$$

$${x_1}^2 x_2 < {x_1}^2 x_3$$

Monomial ordering

Some orderings in $\mathbb{F}_q[x_1, x_2, x_3]$.

Lexicographical order (lex)

First, compare degrees of highest variable, then second variable, ...

$$x_1 > x_2 > x_3, \qquad x_1 > {x_2}^2,$$

$${x_1}^2 x_2 > {x_1}^2 x_3$$

Graded lex. order (grlex)

First, compare total degree, then lex. order if equality.

$$x_1 > x_2 > x_3, \qquad x_1 < {x_2}^2,$$

$${x_1}^2 x_2 > {x_1}^2 x_3$$

Graded reverse lex. order (grevlex)

First, compare total degree, then inverse lex. order if equality.

$$x_1 < x_2 < x_3, \qquad x_1 < {x_2}^2,$$

$${x_1}^2 x_2 < {x_1}^2 x_3$$

Weighted graded lex. order

First, compare weighted sum of degrees, then graded lex. order.

If $\mathrm{wt}(x_1) = 3$, $\mathrm{wt}(x_2) = 1$ and $\mathrm{wt}(x_3) = 4$, then

$$x_1 < {x_2}^2 x_3$$

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.

Strategies

How to efficiency solve polynomial systems to build algebraic attacks?

Strategies

How to efficiency solve polynomial systems to build algebraic attacks?

- ⋆ by bypassing some rounds of iterated constructions
- ⋆ by changing the modeling
- ⋆ by changing the ordering

Strategies

How to efficiency solve polynomial systems to build algebraic attacks?

- ⋆ by bypassing some rounds of iterated constructions
- ⋆ by changing the modeling
- ⋆ by changing the ordering
- ⋆ .... by doing nothing??

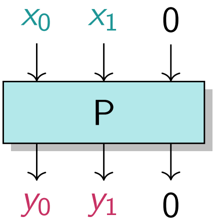
Ethereum Foundation Challenges

https://www.zkhashbounties.info/

(November 2021)

Solving CICO Problem

- ⋆ Feistel–MiMC [Albrecht et al., 2016]
- ⋆ Poseidon [Grassi et al., 2021]
- ⋆ Rescue–Prime [Aly et al., 2020]
- ⋆ `Reinforced Concrete` [Grassi et al., 2022]



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

A. Bariant, C. Bouvier, G. Leurent, L. Perrin, 2022

Cryptanalysis Challenge

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

(a) *Feistel–MiMC*

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

(b) *Rescue–Prime*

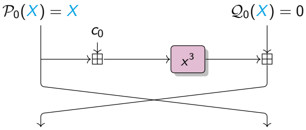
Category	Parameters	Security level	Bounty
Easy	$\mathrm{RP} = 3$	8	$2,000
Easy	$\mathrm{RP} = 8$	16	$4,000
Medium	$\mathrm{RP} = 13$	24	$6,000
Hard	$\mathrm{RP} = 19$	32	$12,000
Hard	$\mathrm{RP} = 24$	40	$26,000

(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

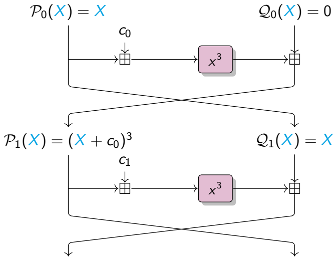
(d) `Reinforced Concrete`

Feistel-MiMC



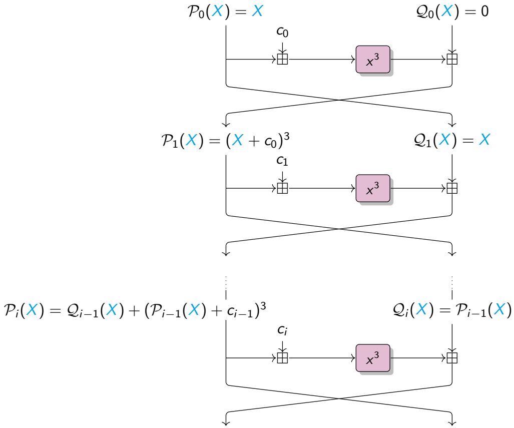
$$\begin{cases} \mathcal{P}_0(X) & = X \\ \mathcal{Q}_0(X) & = 0 \end{cases}$$

Feistel-MiMC



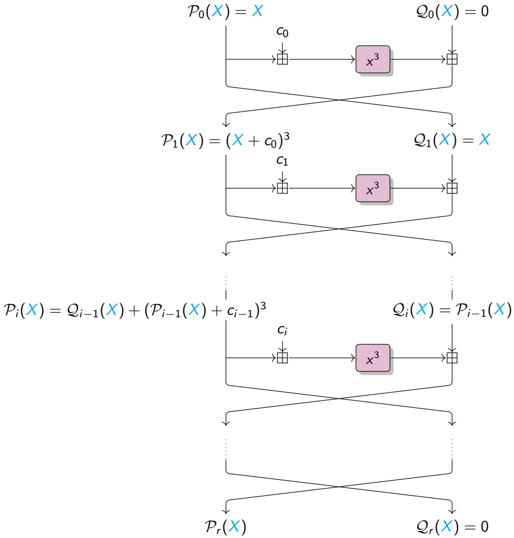
$$\begin{cases} \mathcal{P}_0(X) &= X \\ \mathcal{Q}_0(X) &= 0 \\ \mathcal{P}_1(X) &= (X + c_0)^3 \\ \mathcal{Q}_1(X) &= X \end{cases}$$

Feistel-MiMC



$$
\begin{cases}
\mathcal{P}_0(X) & = X \\
\mathcal{Q}_0(X) & = 0 \\
\mathcal{P}_1(X) & = (X + c_0)^3 \\
\mathcal{Q}_1(X) & = X \\
\dots & \\
\mathcal{P}_i(X) & = \mathcal{Q}_{i-1}(X) + (\mathcal{P}_{i-1}(X) + c_{i-1})^3 \\
\mathcal{Q}_i(X) & = \mathcal{P}_{i-1}(X)
\end{cases}
$$

Feistel-MiMC



$$
\begin{cases}
\mathcal{P}_0(X) & = X \\
\mathcal{Q}_0(X) & = 0 \\
\mathcal{P}_1(X) & = (X + c_0)^3 \\
\mathcal{Q}_1(X) & = X \\
\dots & \\
\mathcal{P}_i(X) & = \mathcal{Q}_{i-1}(X) + (\mathcal{P}_{i-1}(X) + c_{i-1})^3 \\
\mathcal{Q}_i(X) & = \mathcal{P}_{i-1}(X) \\
\dots & \\
\mathcal{Q}_r(X) & = 0
\end{cases}
$$

1 variable + $(2r + 1)$ equations

Cryptanalysis Challenge

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

(a) *Feistel–MiMC*

$12,000

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

(b) *Rescue–Prime*

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

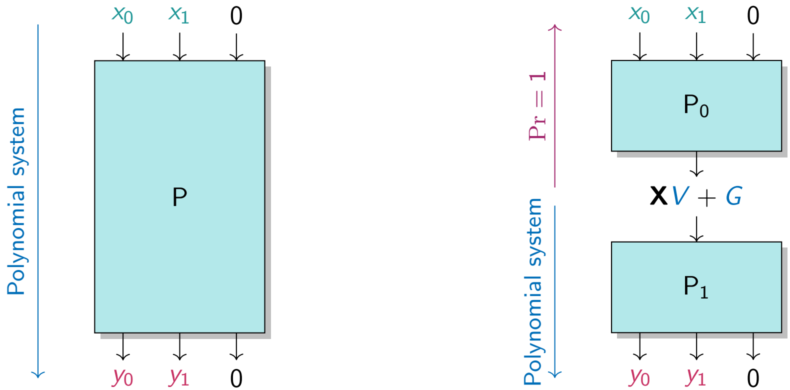
Category	Parameters	Security level	Bounty
Easy	$p = 281474976710597$	24	$4,000
Medium	$p = 72057594037926839$	28	$6,000
Hard	$p = 18446744073709551557$	32	$12,000

(d) `Reinforced Concrete`

Trick for SPN

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

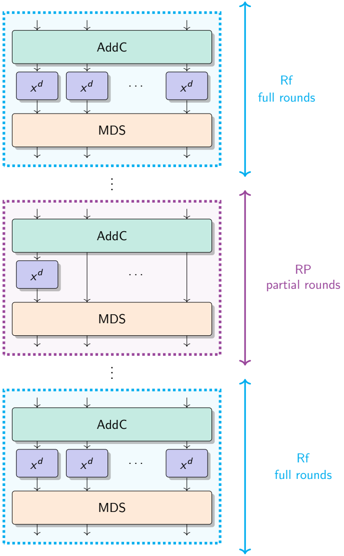
$$\exists\ V, G \in \mathbb{F}_p^3, \quad \text{s.t. } \forall\ \mathbf{X} \in \mathbb{F}_p, \quad P_0^{-1}(\mathbf{X}V + G) = (*, *, 0)\ .$$



(a) *R-round system.*

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

Poseidon

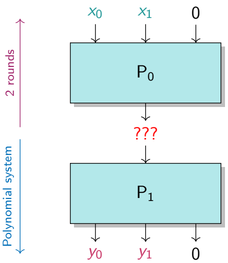


- ⋆ S-box:

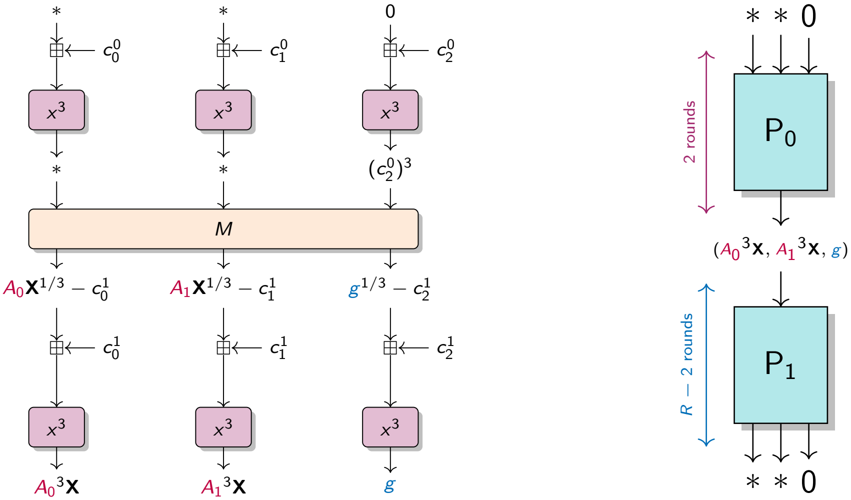
$$x \mapsto x^3$$

- ⋆ Nb rounds:

$$R = 2 \times Rf + RP$$
$$= 8 + (\text{from } 3 \text{ to } 24)$$

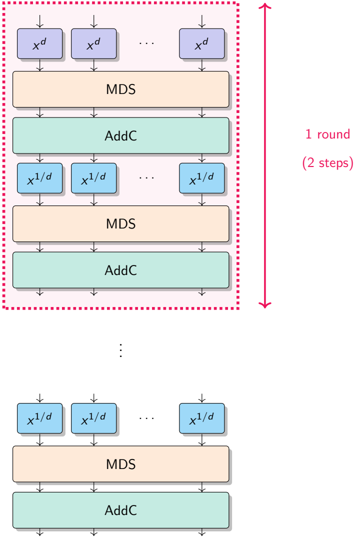


Trick for POSEIDON



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

Rescue–Prime



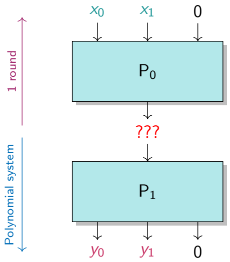
⋆ S-box:

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

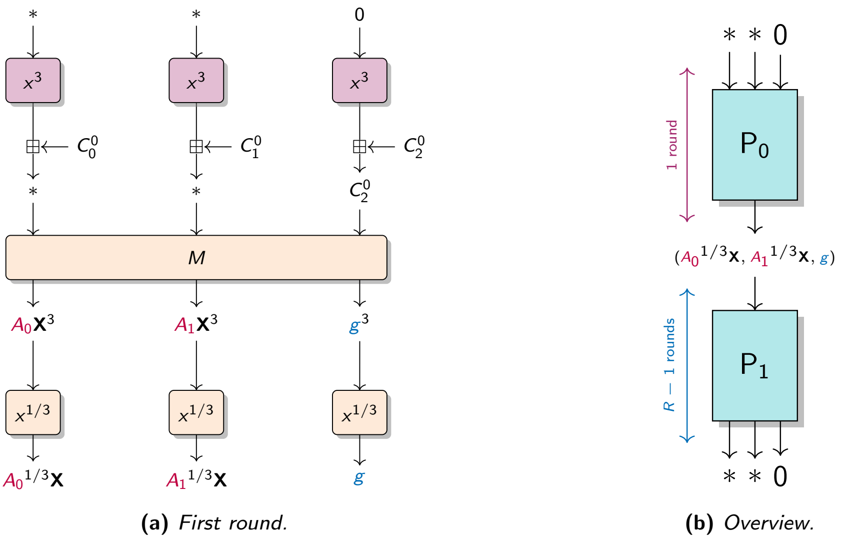
⋆ Nb rounds:

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

(2 S-boxes per round)



Trick for Rescue–Prime



Cryptanalysis Challenge

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

(a) *Feistel–MiMC*

$26, 000

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

(b) *Rescue–Prime*

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

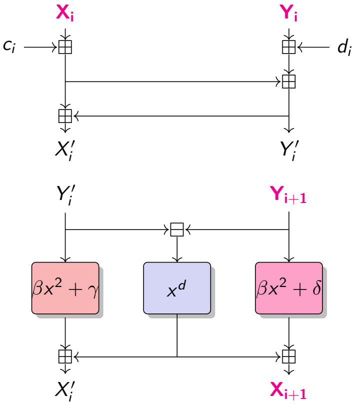
(c) POSEIDON

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

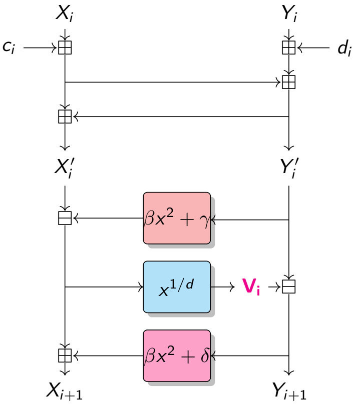
(d) `Reinforced Concrete`

Modeling of Anemoi

C. Bouvier, P. Briaud, P. Chaidos, L. Perrin, R. Salen, V. Velichkov and D. Willems, 2023

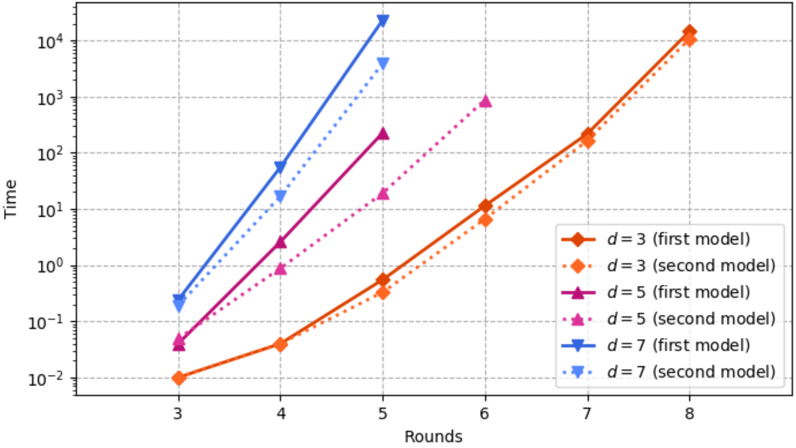


Model 1.



Model 2.

Importance of modeling



FreeLunch attack

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

Multivariate solving:

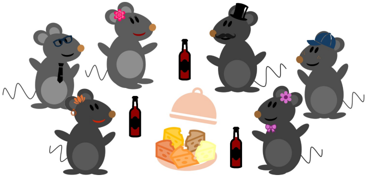
- ⋆ Define the system
- ⋆ Compute a grevlex order GB (**F5** algorithm)
- ⋆ Convert it into lex order GB (**FGLM** algorithm)
- ⋆ Find the roots in $\mathbb{F}_q^n$ of the GB polynomials using univariate system resolution.

FreeLunch attack

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

Multivariate solving:

- ⋆ Define the system
- ⋆ Compute a grevlex order GB (**F5** algorithm) ⇝ **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.



New Challenges

https://www.poseidon-initiative.info/

(November 2024)

New winners

More than $\$30,000$

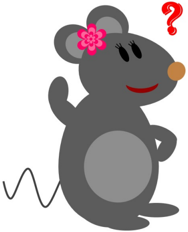
A. Bak,
A. Bariant,
A. Boeuf,
M. Hostettler,
G. Jazeron

and others...

- Poseidon-256:
- ~~24-bit estimated security: RF=6, RP=8.~~ $4000 **claimed 9 Dec 2024**
- ~~28-bit estimated security: RF=6, RP=9~~. $6000 **claimed 2 Jan 2025**
- 32-bit estimated security: RF=6, RP=11. $10000
- 40-bit estimated security: RF=6, RP=16. $15000
- Poseidon-64:
- 24-bit estimated security: RF=6, RP=7 $4000
- 28-bit estimated security: RF=6, RP=8. $6000
- 32-bit estimated security: RF=6, RP=10. $10000
- 40-bit estimated security: RF=6, RP=13. $15000
- Poseidon-31:
- 24-bit estimated security: ~~RF=4, RP=0 (M31)~~ **claimed 29 Nov 2025** ~~and RP=1 (KoalaBear). $4000~~ **claimed 30 Nov 2025**
- 28-bit estimated security: ~~RF=4, RP=1 (M31) and RP=3 (KoalaBear). $6000~~ **claimed 29 Nov 2025**
- 32-bit estimated security: ~~RF=6, RP=1 (M31)~~ **claimed 2 Dec 2025** ~~and RP=4 (KoalaBear). $10000~~ **claimed 5 Dec 2025**
- 40-bit estimated security: RF=6, RP=4 (M31 only). $15000

QUIZ !!

Could we use our trick for SPN on `Reinforced Concrete`?



QUIZ !!

Could we use the FreeLunch attack on Feistel–MiMC?

Conclusions and Perspectives

Conclusions

- ⋆ try as many modeling as possible
- ⋆ prefer univariate instead of multivariate system
- ⋆ be careful of tricks to bypass rounds

AOP: a new lucrative business?

Conclusions and Perspectives

Conclusions

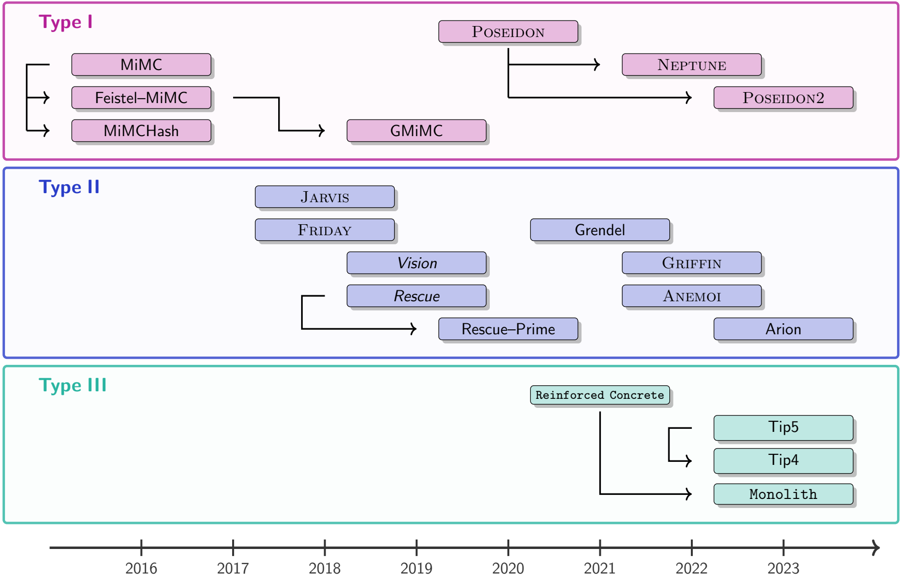
- try as many modeling as possible
- prefer univariate instead of multivariate system
- be careful of tricks to bypass rounds

AOP: a new lucrative business?

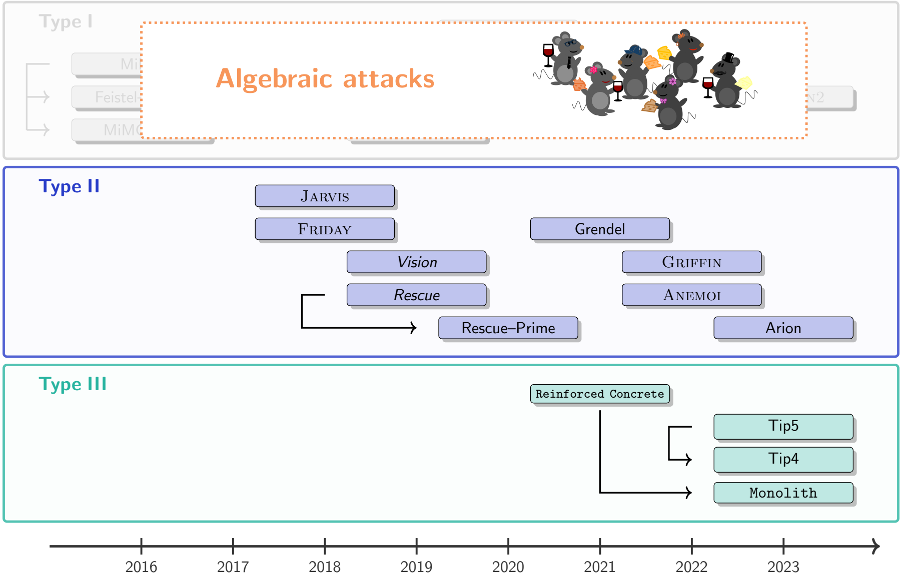
Perspectives

- study of other attacks
- study the security of Type III
- ...

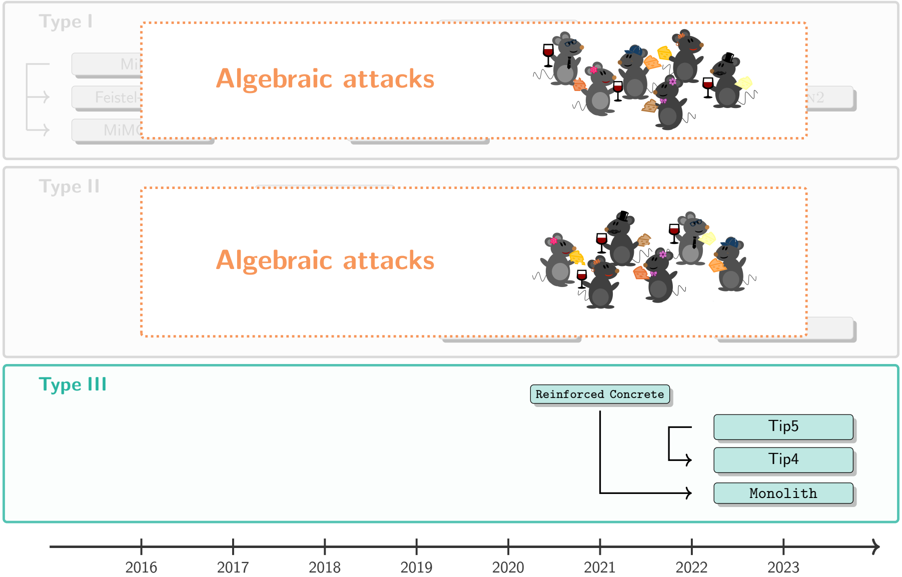
Primitives overview



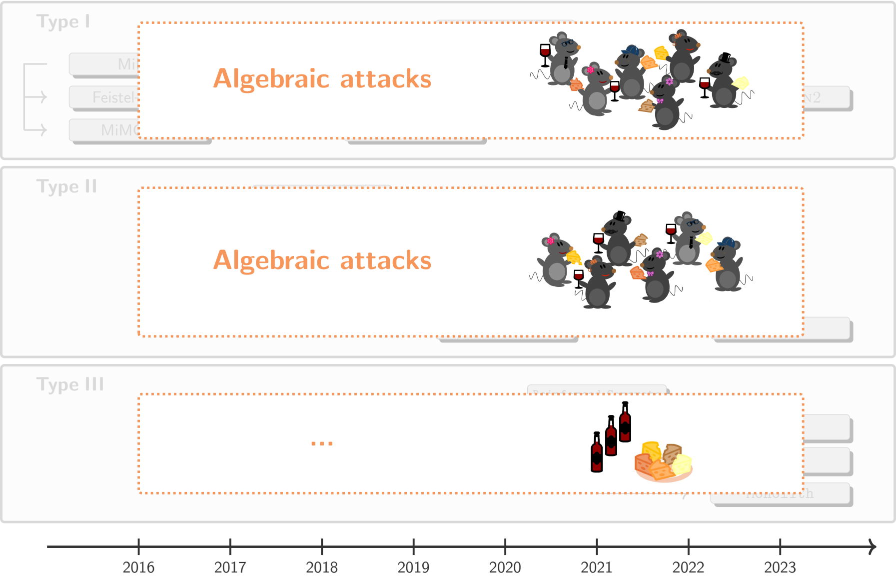
Primitives overview



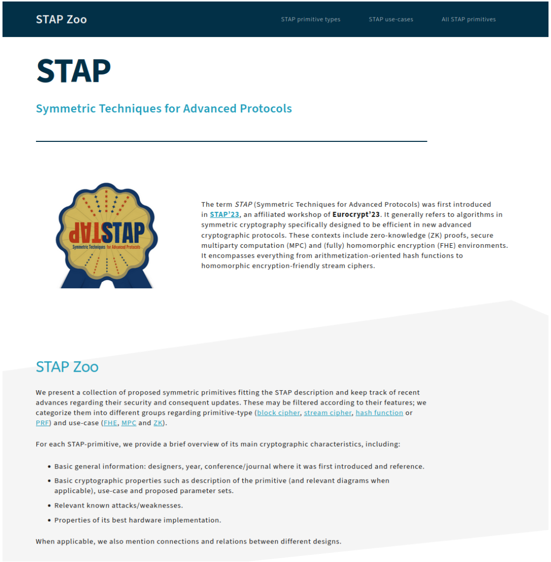
Primitives overview



Primitives overview



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

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

Thank you