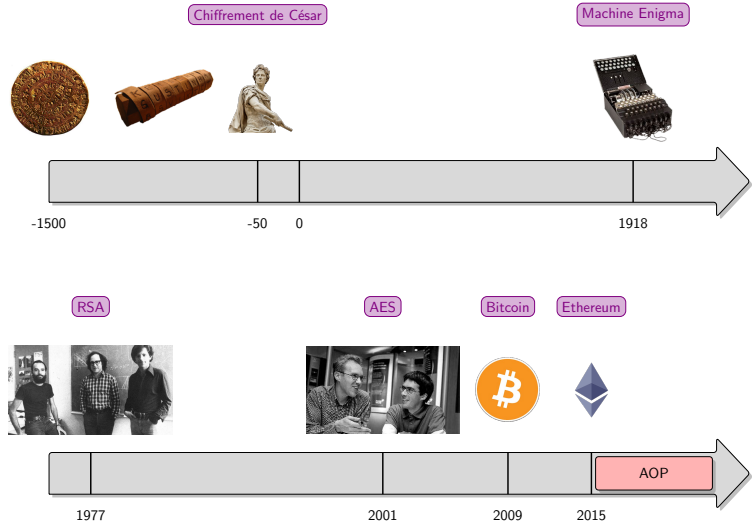
Comment prouver ce que l'on ne peut révéler ?

Clémence Bouvier

Université de Lorraine, CNRS, Inria, LORIA

AMUSEC, Marseille, France
13 Mars 2025

Un peu d'histoire



Sommaire

Introduction Générale

La cryptologie : premières définitions

Chiffrements par bloc

Fonctions de hachage

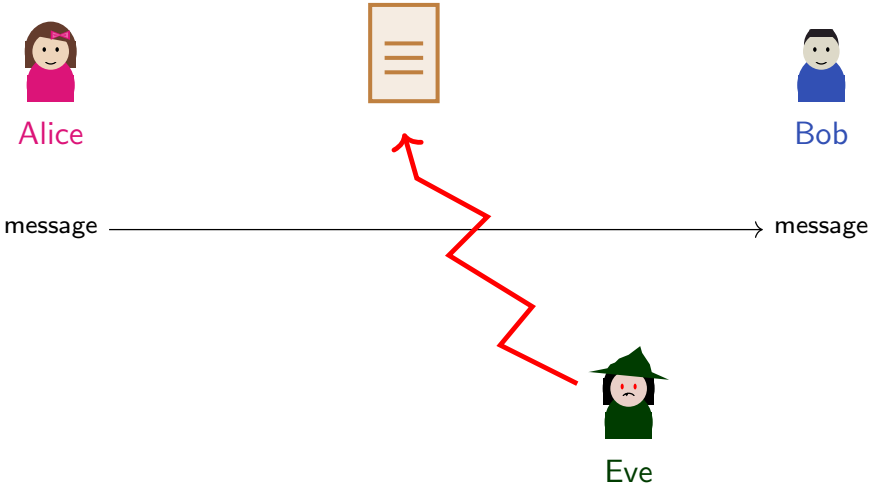
Pourquoi chiffrer ?

Alice

Bob

message ⟶ message

Pourquoi chiffrer ?



Quelques définitions

$$\text{Cryptologie} = \text{Cryptographie} + \text{Cryptanalyse}$$

Cryptographie

Ensemble des méthodes utilisées pour transformer un message clair en un message inintelligible.

Cryptanalyse

Ensemble des méthodes utilisées pour retrouver le message en clair à partir du message chiffré.

Quelques définitions

Cryptologie = Cryptographie + Cryptanalyse

Cryptographie

Ensemble des méthodes utilisées pour transformer un message clair en un message inintelligible.

Cryptanalyse

Ensemble des méthodes utilisées pour retrouver le message en clair à partir du message chiffré.

	Clair ⇒ Chiffré	Chiffré ⇒ Clair
En connaissant la clé	Chiffrer	Déchiffrer
Ne connaissant pas la clé		Décrypter

Quelques définitions

Cryptologie = Cryptographie + Cryptanalyse

Cryptographie

Ensemble des méthodes utilisées pour transformer un message clair en un message inintelligible.

Cryptanalyse

Ensemble des méthodes utilisées pour retrouver le message en clair à partir du message chiffré.

	Clair ⇒ Chiffré	Chiffré ⇒ Clair
En connaissant la clé	Chiffrer	Déchiffrer
Ne connaissant pas la clé	~~Crypter~~	Décrypter

Cryptographie symétrique

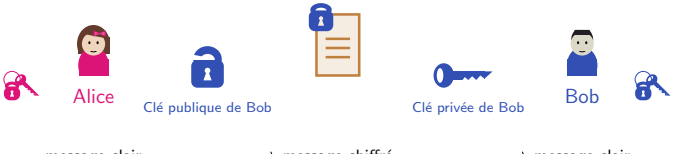
Exemple : AES

Cryptographie asymétrique

Exemple : RSA

message clair ⟶ message chiffré ⟶ message clair

Cryptographie hybride



Chiffrements par Bloc

- ⋆ entrée :

  bloc x de taille n

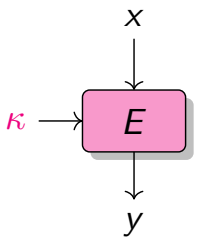
- ⋆ paramètre :

  clé κ de taille k

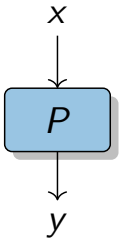
- ⋆ sortie :

  bloc y de taille n

- ⋆ symétrie : E et E^{-1} utilise la même clé κ



(a) *Chiffrement par bloc*



(b) *Permutation aléatoire*

Chiffrements par Bloc

- ⋆ entrée :

  bloc x de taille n

- ⋆ paramètre :

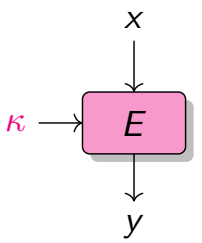
  clé κ de taille k

- ⋆ sortie :

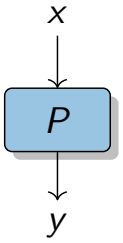
  bloc y de taille n

- ⋆ symétrie : E et E^{-1} utilise la même clé κ

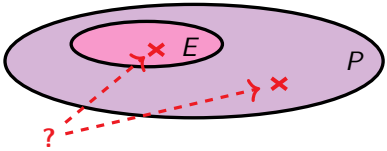
Un chiffrement par bloc est une famille de permutations de blocs de taille n.



(a) *Chiffrement par bloc*

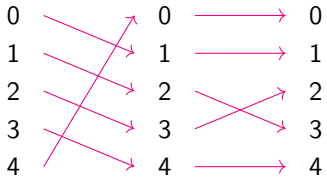


(b) *Permutation aléatoire*



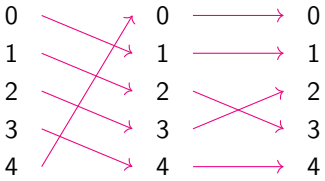
Indistinguabilité

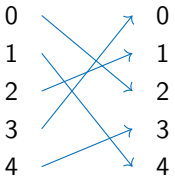
Exemple $E : x \mapsto (x+k)^3$ avec $x \in \{0,1,2,3,4\}$ et $k = 1$.



Indistinguabilité

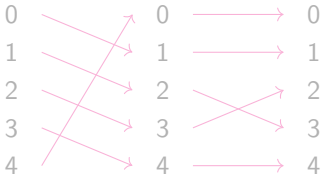
Exemple $E : x \mapsto (x + k)^3$ avec $x \in \{0, 1, 2, 3, 4\}$ et $k = 1$.

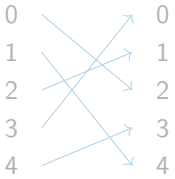


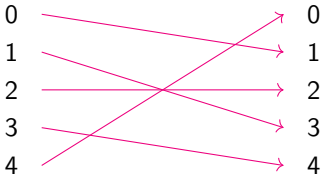


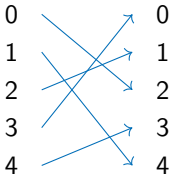
Indistinguabilité

Exemple $E : x \mapsto (x + k)^3$ avec $x \in \{0, 1, 2, 3, 4\}$ et $k = 1$.





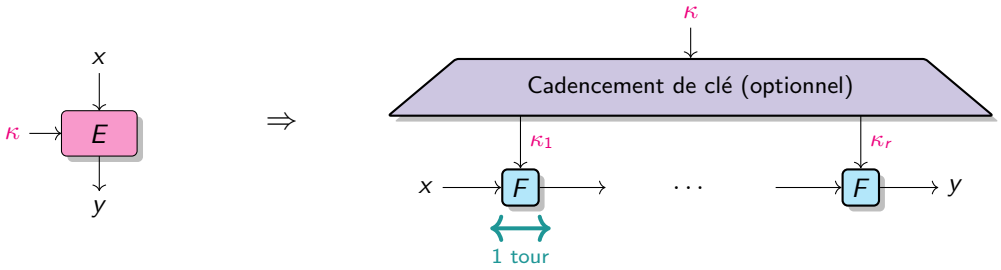




Constructions itérées

Comment construire un chiffrement par bloc efficace ?

En itérant une fonction de tour.

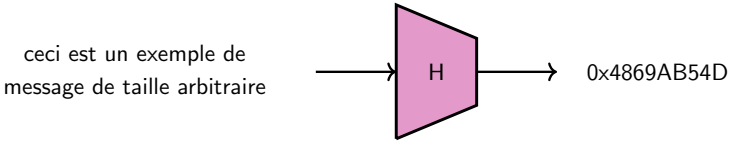


Contraintes de performance ! La primitive doit être rapide.

Fonctions de hachage

Définition

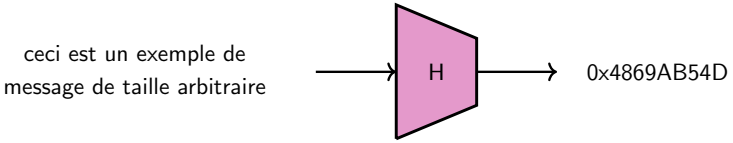
Pour n'importe quel message x de taille arbitraire, $H(x)$ est un message unique de taille fixe.



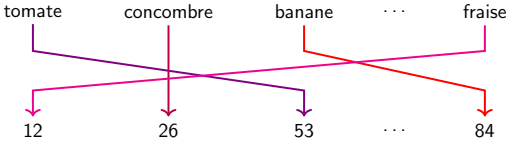
Fonctions de hachage

Définition

Pour n'importe quel message x de taille arbitraire, $H(x)$ est un message unique de taille fixe.



Exemple : Table de hachage des fruits et légumes



Un exemple

La fonction H renvoie les **8 premiers caractères** du message.

Message	Haché
Bonjour. Comment allez-vous ?	Bonjour.
Aujourd'hui nous sommes jeudi.	Aujourd'
Ils sont gentils.	Ils sont
Un zeste de citron	Un zeste
0123456789	01234567
abcdefghijklmnopqrstuvwxyz	abcdefgh
13 mars 2025	13 mars

Collisions et Pré-images

Pré-image

Ils sont
$\Downarrow$
Ils sont gentils.

Collision

Message	Haché
Bonjour.	Bonjour.
Bonjour. Comment allez-vous ?	Bonjour.

Collisions et Pré-images

Pré-image

Ils sont
$\Downarrow$
Ils sont gentils.

Collision

Message	Haché
Bonjour.	Bonjour.
Bonjour. Comment allez-vous ?	Bonjour.

Résistance à la pré-image

Il doit être *impossible* de trouver :

$$x \quad \text{tel que} \quad H(x) = y$$

Résistance à la collision

Il doit être *impossible* de trouver :

$$x \neq x' \quad \text{tel que} \quad H(x) = H(x')$$

Paradoxe des anniversaires

Dans un groupe de **23** personnes, il y a **51%** de chance que 2 personnes aient leur anniversaire le même jour.

Paradoxe des anniversaires

Dans un groupe de **23** personnes, il y a **51%** de chance que 2 personnes aient leur anniversaire le même jour.

Exemple : SHA-256 pour les fonctions de hachage

Taille du haché	N	256
Nombre de hachés possible	2^N	2^{256}
Nombre de hachés pour une collision	$2^{N/2}$	2^{128}

$$2^{128} = 340282366920938463463374607431768211456 \simeq 10^{68}$$

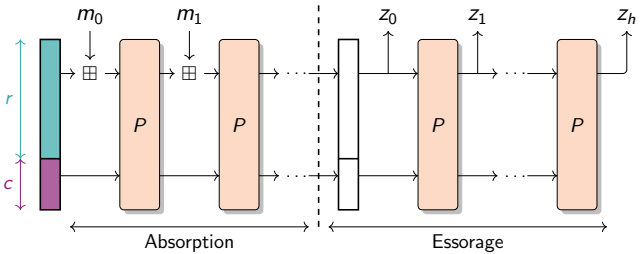
Il y a 10^{78} atomes dans l'univers !

Construction de type Éponge

Construction Éponge

Paramètres :

- ⋆ taux $r > 0$
- ⋆ capacité $c > 0$
- ⋆ permutation P

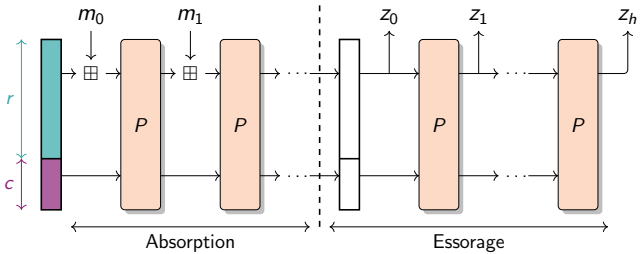


Construction de type Éponge

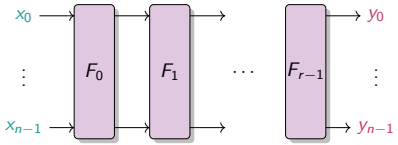
Construction Éponge

Paramètres :

- ⋆ taux $r > 0$
- ⋆ capacité $c > 0$
- ⋆ permutation P



P est une construction itérée



QUIZ ! !

- L'AES fait partie des chiffrements de type symétrique. Vrai ou Faux ?
- RSA est-il un chiffrement de type symétrique ou asymétrique ?
- Un chiffrement par bloc est de type asymétrique. Vrai ou Faux ?
- De quel type est une fonction de hachage ? Symétrique ? Asymétrique ?

A Retenir

Comment protéger les données ?

- ⋆ Cryptographie symétrique et/ou asymétrique
- ⋆ Chiffrements par bloc
- ⋆ Fonctions de hachage

« Crypter » n'existe pas !

Un nouveau contexte

Introduction des preuves à divulgation nulle de connaissance

Quelles conséquences ?

Sécuriser un calcul

Sécuriser l'échange de message

- **Confidentialité**

  Une personne extérieure à l'échange ne peut lire le message.

- **Intégrité**

  Une personne extérieure à l'échange ne peut le modifier.

- **Authentification**

  Le message a été écrit par la bonne personne.

Sécuriser un calcul

Sécuriser l'échange de message

- ⋆ **Confidentialité**

  Une personne extérieure à l'échange ne peut lire le message.

- ⋆ **Intégrité**

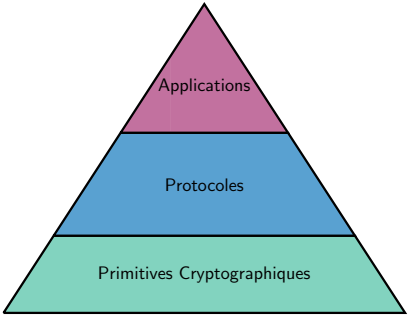
  Une personne extérieure à l'échange ne peut le modifier.

- ⋆ **Authentification**

  Le message a été écrit par la bonne personne.

Sécuriser un calcul

Un besoin de nouvelles primitives

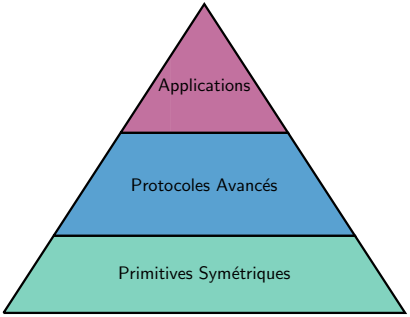


Un besoin de nouvelles primitives

Protocoles nécessitant de nouvelles primitives :

- ⋆ **FHE** : Chiffrement homomorphe
- ⋆ **MPC** : Calcul Multi-partite
- ⋆ **ZKP** : Preuves à divulgation nulle de connaissance

  Exemple : SNARKs, STARKs, Bulletproofs



Problème : Concevoir de nouvelles primitives symétriques

Chiffrement Homomorphe

Que vaut 37×15 ?

Chiffrement Homomorphe

Que vaut 37×15 ?

chiffre 37 et 15

envoie 27953 et 6144

Chiffrement Homomorphe

Que vaut 37×15 ?

chiffre 37 et 15

envoie 27953 et 6144

calcule 27953×6144

envoie 171743232

Chiffrement Homomorphe

Que vaut 37×15 ?

chiffre 37 et 15

envoie 27953 et 6144

calcule 27953×6144

envoie 171743232

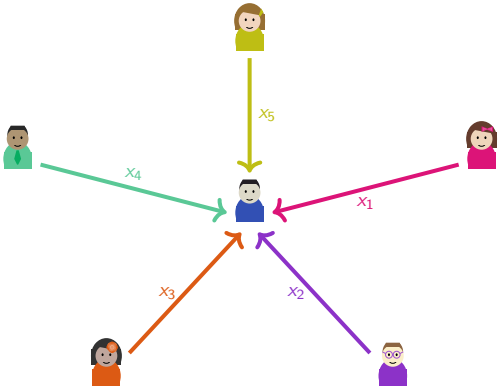
déchiffre 171743232
et obtient 555

Calcul Multi-partite

Comment calculer $x_1 + x_2 + x_3 + x_4 + x_5$?

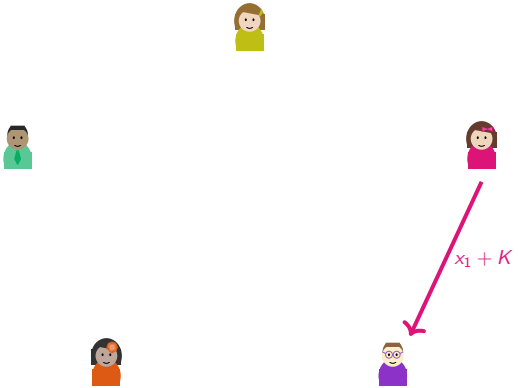
Calcul Multi-partite

Comment calculer $x_1 + x_2 + x_3 + x_4 + x_5$?



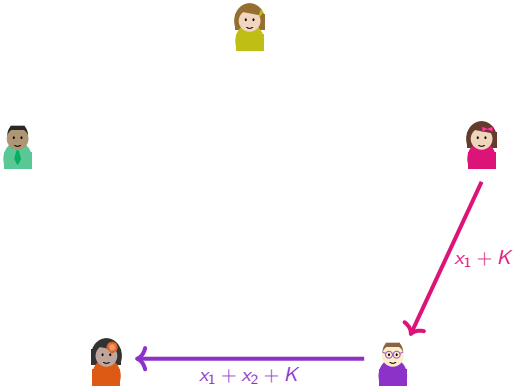
Calcul Multi-partite

Comment calculer $x_1 + x_2 + x_3 + x_4 + x_5$?



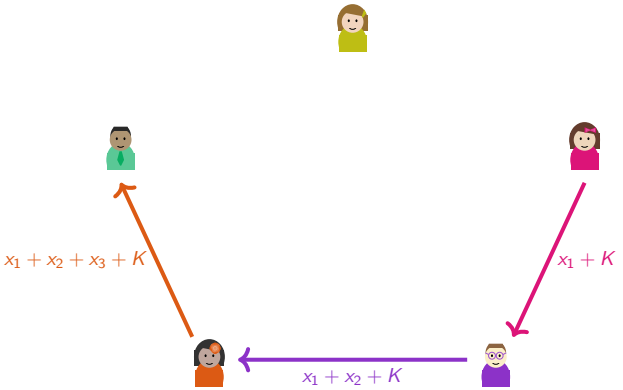
Calcul Multi-partite

Comment calculer $x_1 + x_2 + x_3 + x_4 + x_5$?



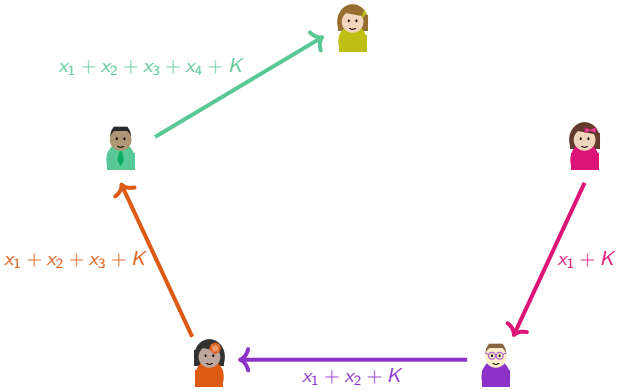
Calcul Multi-partite

Comment calculer $x_1 + x_2 + x_3 + x_4 + x_5$?



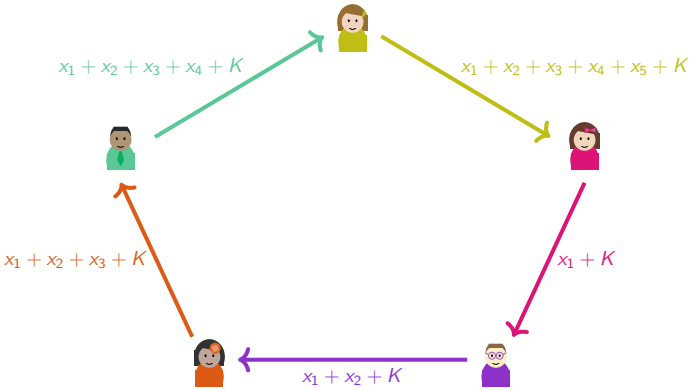
Calcul Multi-partite

Comment calculer $x_1 + x_2 + x_3 + x_4 + x_5$?



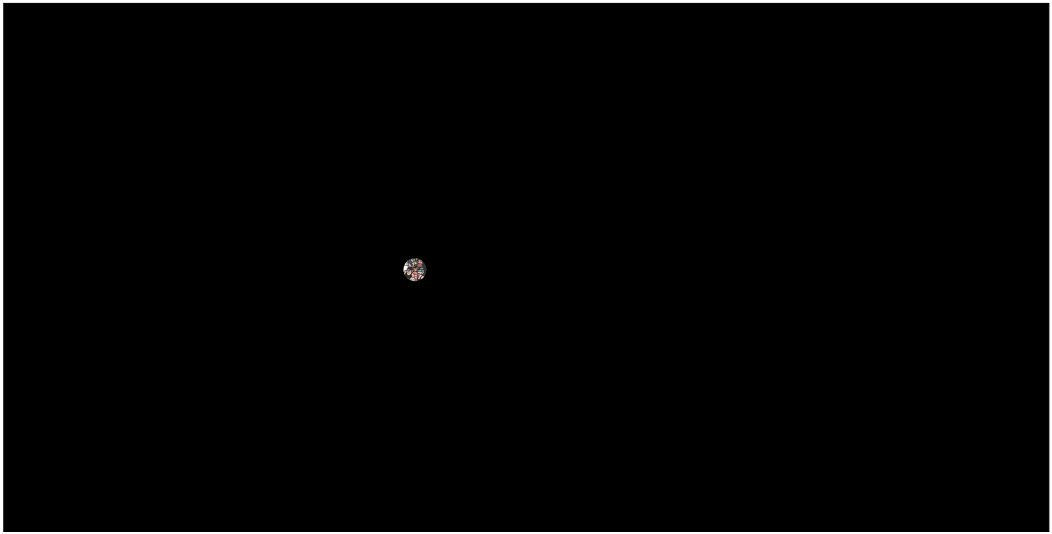
Calcul Multi-partite

Comment calculer $x_1 + x_2 + x_3 + x_4 + x_5$?



Où est Charlie ?

Où est Charlie ?



Où est Charlie ?

Où est Charlie ?

Où est Charlie ?

Où est Charlie ?

Le 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	

Grille non-résolue

Le Sudoku

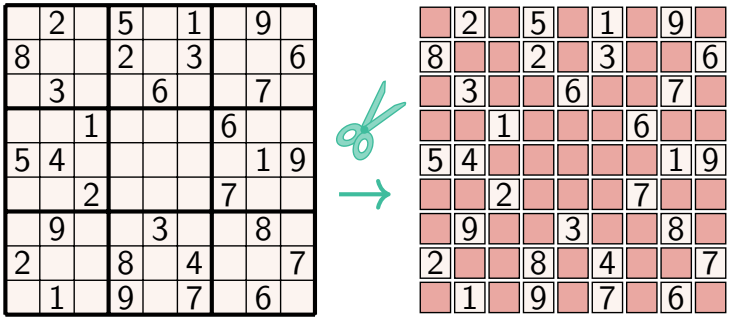
	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

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*

Grille non-résolue

Grille résolue

Le Sudoku



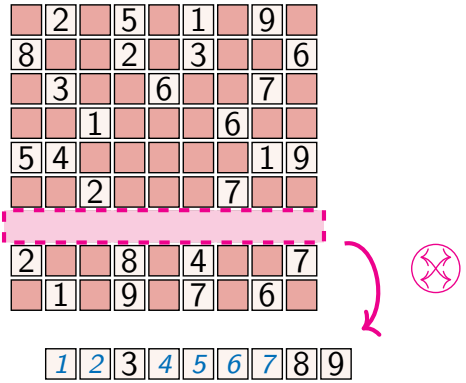
Grille non-résolue

Grille découpée

Le 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	

Grille non-résolue

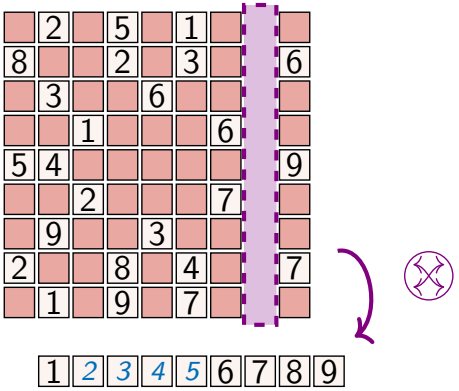


Vérification d'une ligne

Le 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	

Grille non-résolue

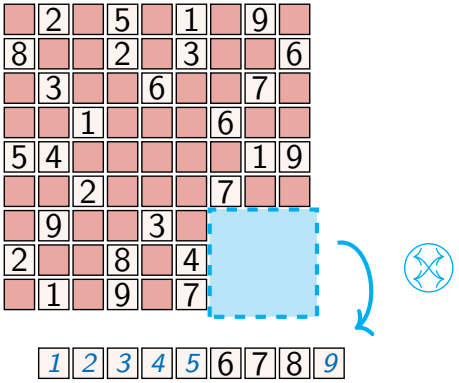


Vérification d'une colonne

Le Sudoku

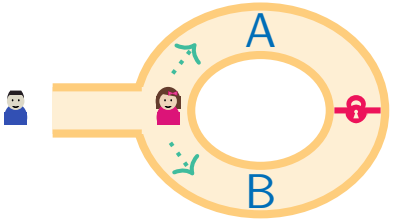
	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

Grille non-résolue

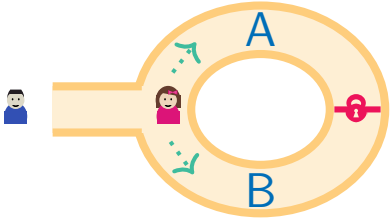


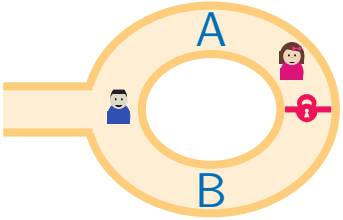
Vérification d'un carré

La cave d'Ali-Baba

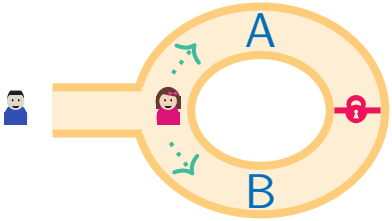


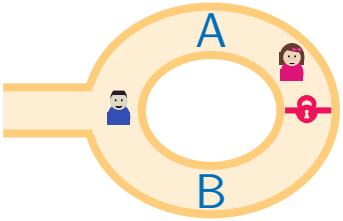
La cave d'Ali-Baba

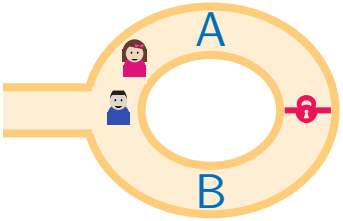




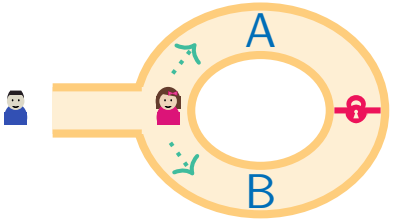
La cave d'Ali-Baba



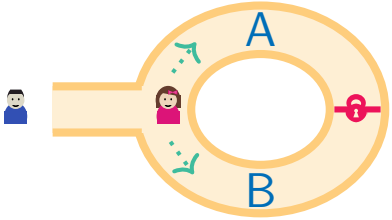


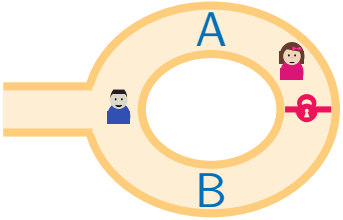


La cave d'Ali-Baba

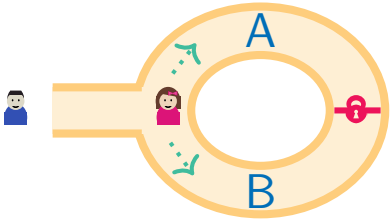


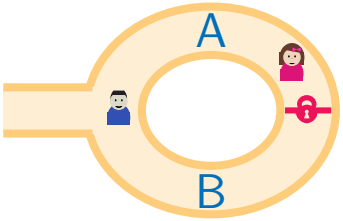
La cave d'Ali-Baba

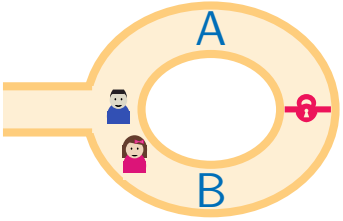




La cave d'Ali-Baba

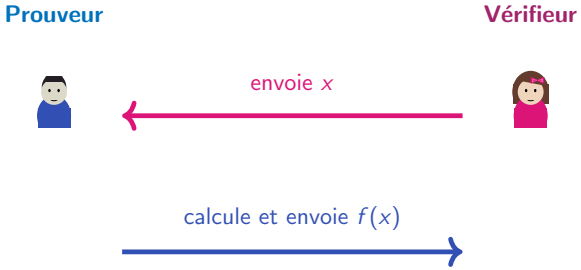




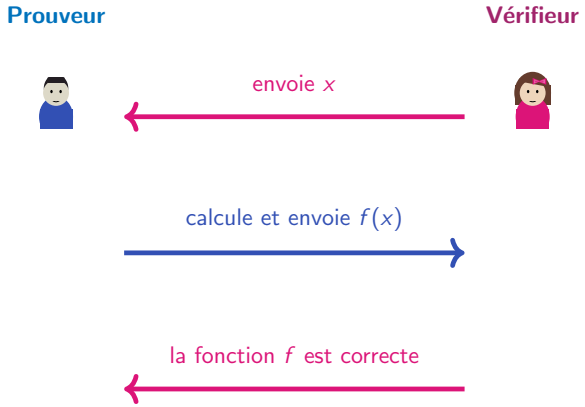


Preuves intéractives (IZKP)

Preuves intéractives (IZKP)



Preuves intéractives (IZKP)



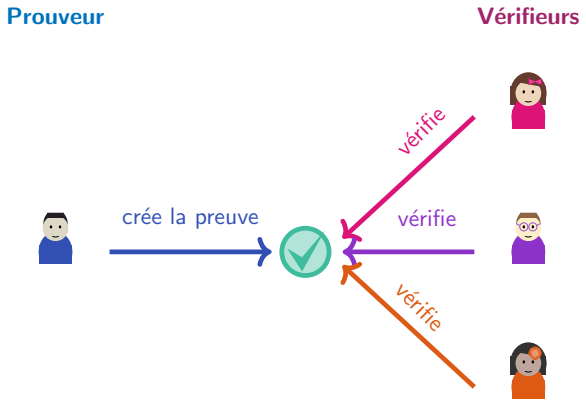
Preuves non-intéractives (NIZKP)

Prouveur

Vérifieurs

crée la preuve

Preuves non-intéractives (NIZKP)

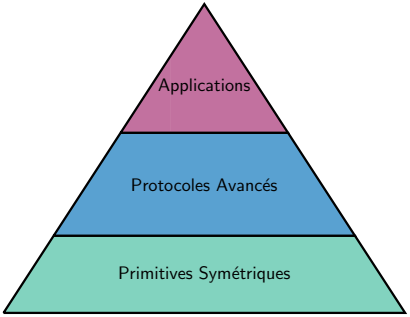


Un besoin de nouvelles primitives

Protocoles nécessitant de nouvelles primitives :

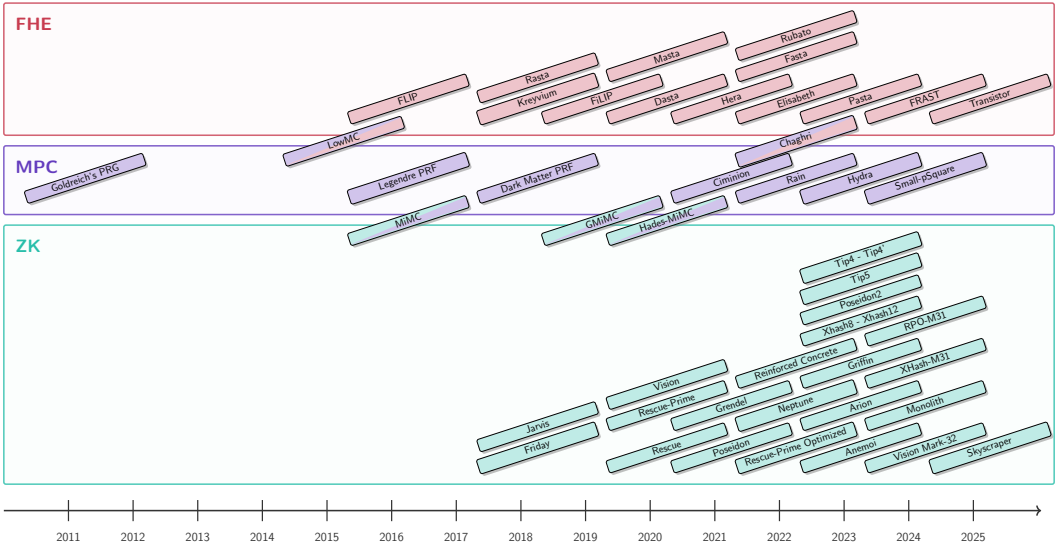
- ⋆ **FHE** : Chiffrement homomorphe
- ⋆ **MPC** : Calcul Multi-partite
- ⋆ **ZKP** : Preuves à divulgation nulle de connaissance

  Exemple : SNARKs, STARKs, Bulletproofs



Problème : Concevoir de nouvelles primitives symétriques

Primitives

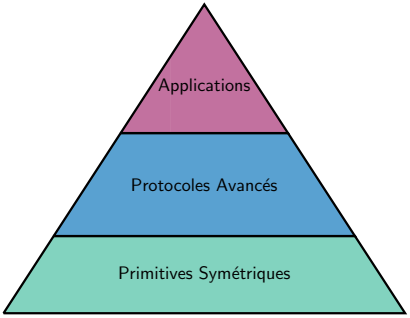

FHE
FLIP
LowMC
Rasta
Kreyvium
Masta
FiLIP
Dasta
Hera
Rubato
Fasta
Elisabeth
Pasta
FRAST
Transistor
MPC
Goldreich's PRG
Legendre PRF
MiMC
Dark Matter PRF
GMiMC
Hades-MiMC
Chaghri
Ciminion
Rain
Hydra
Small-pSquare
ZK
Jarvis
Friday
Vision
Rescue-Prime
Grendel
Rescue
Poseidon
Neptune
Reinforced Concrete
Rescue-Prime Optimized
Arion
Griffin
Anemoi
Tip4 - Tip4'
Tip5
Poseidon2
Xhash8 - Xhash12
RPO-M31
XHash-M31
Monolith
Vision Mark-32
Skyscraper
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025

Un besoin de nouvelles primitives

Protocoles nécessitant de nouvelles primitives :

- ⋆ **FHE** : Chiffrement homomorphe
- ⋆ **MPC** : Calcul Multi-partite
- ⋆ **ZKP** : Preuves à divulgation nulle de connaissance

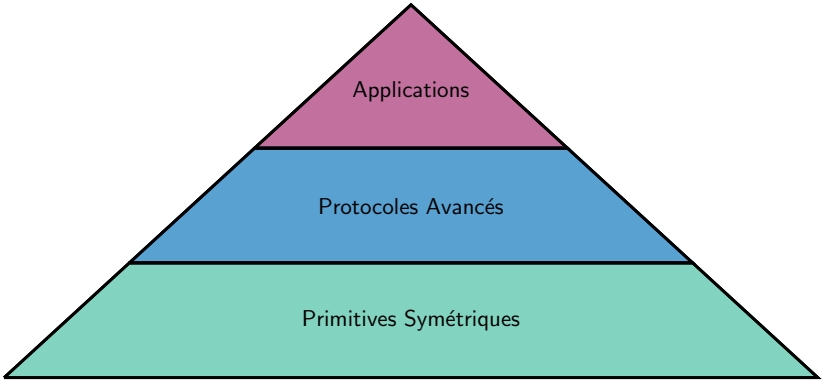
  Exemple : SNARKs, STARKs, Bulletproofs



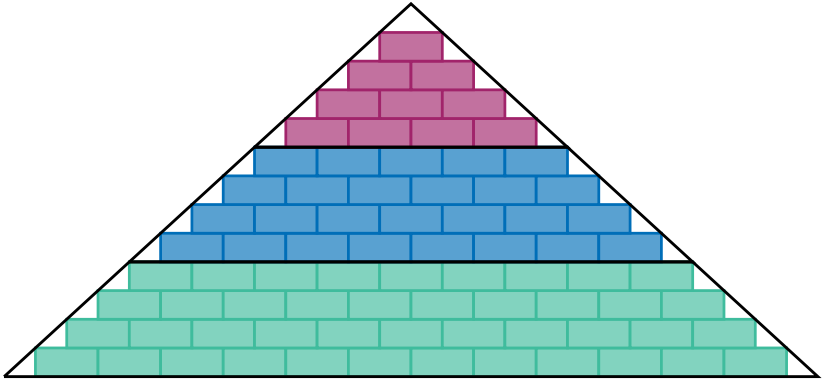
Problème : Concevoir de nouvelles primitives symétriques

Et analyser leur sécurité !

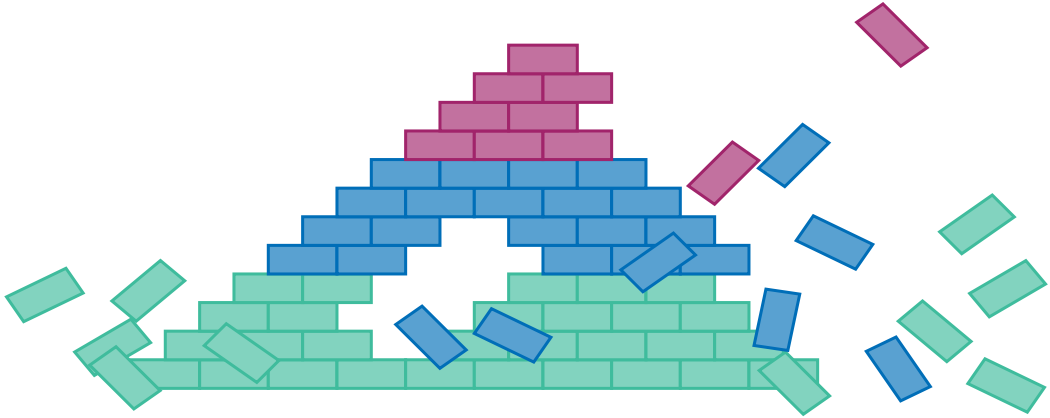
Les briques de base de la sécurité



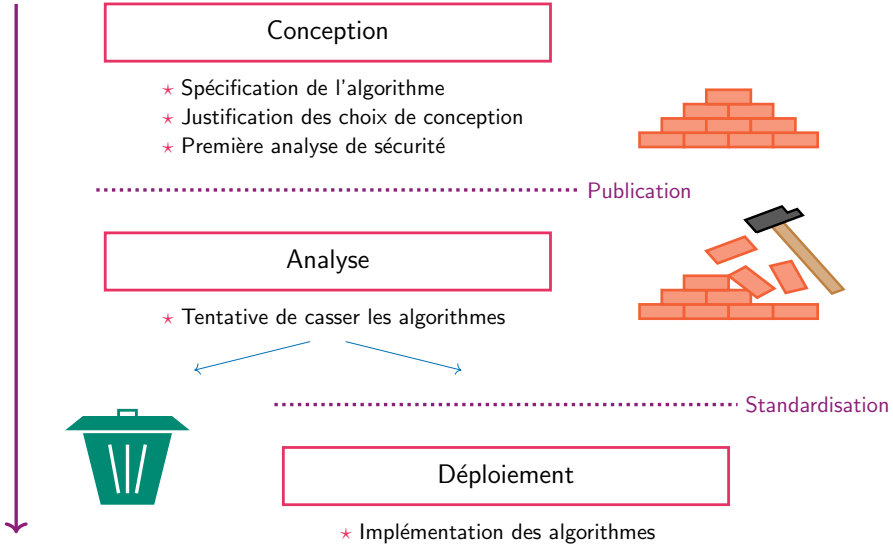
Les briques de base de la sécurité



Les briques de base de la sécurité



Le cycle de vie d'une primitive



QUIZ ! !

- Quelle est la signification de l'acronyme ZKP ?
- « Où est Charlie ? » est un exemple de IZKP. Vrai ou Faux ?
- Le sudoku est-il un exemple de IZKP ? de NIZKP ? des 2 ?
- La cave d'Ali-Baba est un exemple de NIZKP. Vrai ou Faux ?

A Retenir

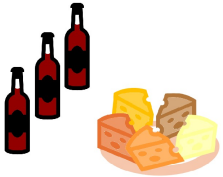
Comment prouver ce que l'on ne peut révéler ?

- ⋆ Grâce aux preuves à divulgation nulle de connaissance (ZKP)
- ⋆ De façon intéractive (IZKP) ou non intéractive (NIZKP)
- ⋆ Cela nécessite de nouvelles primitives symétriques...
- ⋆ ... et d'analyser leur sécurité !

Les nouvelles AOPs

Différences avec les primitives traditionnelles

Comment les classifier ?



AOP

« **Appellation d'origine protégée** »

Brousse du Rove

AOP

« **Arithmetization-Oriented Primitives** »

Brousse du Rove

Un nouvel environnement

Cas traditionnel

Opérations basées sur des portes logiques et instructions CPU.

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

Exemple

Espace vectoriel de l'AES

$$\mathbb{F}_2^n, \text{ où } n = 8$$

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

Un nouvel environnement

Cas traditionnel

Opérations basées sur des portes logiques et instructions CPU.

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

Exemple

Espace vectoriel de l'AES

$$\mathbb{F}_2^n, \text{ où } n = 8$$

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

Orienté Arithmétisation

Opérations basées sur l'arithmétique des grands corps finis.

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

Exemple

Corps scalaire de la courbe BLS12-381

$$\mathbb{F}_p, \text{ où}$$

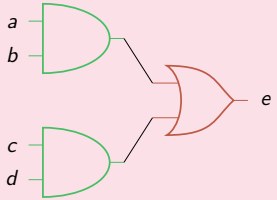
$$p = \texttt{0x73eda753299d7d483339d80809a1d805}$$
$$\texttt{53bda402fffe5bfeffffffff00000001}$$

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

De nouvelles opérations

Cas traditionnel

Portes logiques et instructions CPU.



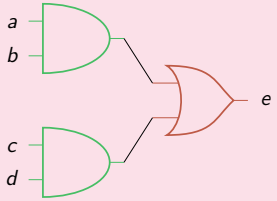
Exemple

$$(0,1,0)\&(1,1,0) = (0,1,0)$$

De nouvelles opérations

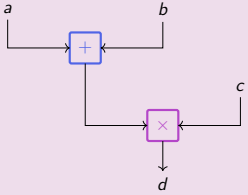
Cas traditionnel

Portes logiques et instructions CPU.



Orienté Arithmétisation

Utilisation de circuits arithmétiques.

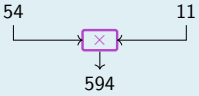


Exemple

$$(0,1,0)\&(1,1,0) = (0,1,0)$$

Exemple

$$54 \times 11 = 594$$

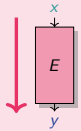


Une nouvelle métrique

Cas traditionnel

Minimiser le temps et la mémoire.

$$y \leftarrow E(x)$$

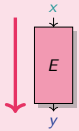


Une nouvelle métrique

Cas traditionnel

Minimiser le temps et la mémoire.

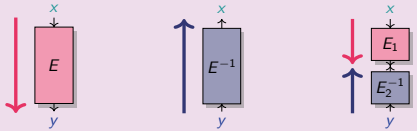
$$y \leftarrow E(x)$$



Orienté Arithmétisation

Minimiser le nombre de multiplications.

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

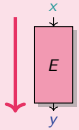


Une nouvelle métrique

Cas traditionnel

Minimiser le temps et la mémoire.

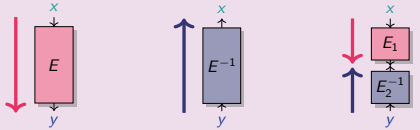
$$y \leftarrow E(x)$$



Orienté Arithmétisation

Minimiser le nombre de multiplications.

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



Exemple

Soit $E : \mathbb{F}_{11} \to \mathbb{F}_{11}, x \mapsto x^3$. On a $E^{-1} : \mathbb{F}_{11} \to \mathbb{F}_{11}, x \mapsto x^7$.

Évaluation : Étant donné $x = 5$, calculer $y = E(x)$.

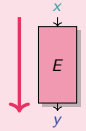
$$y = 5^3 = 4 \text{ (en appliquant } E)$$

Une nouvelle métrique

Cas traditionnel

Minimiser le temps et la mémoire.

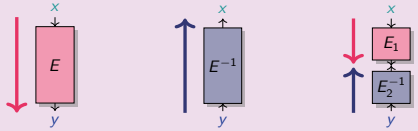
$$y \leftarrow E(x)$$



Orienté Arithmétisation

Minimiser le nombre de multiplications.

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



Exemple

Soit $E : \mathbb{F}_{11} \to \mathbb{F}_{11}, x \mapsto x^3$. On a $E^{-1} : \mathbb{F}_{11} \to \mathbb{F}_{11}, x \mapsto x^7$.

Vérification : Étant donné $x = 5$ et $y = 4$, vérifier si $y = E(x)$.

$$5^3 = 4 \text{ (en appliquant } E) \qquad \text{ou} \qquad 4^7 = 5 \text{ (en appliquant } E^{-1})$$

En résumé

Cas traditionnel

- Alphabet :

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

- Opérations :
  Portes logiques/instructions CPU
- Métrique :
  Minimiser le temps et la mémoire pour l'évaluation
- **Des décennies de Cryptanalyse**

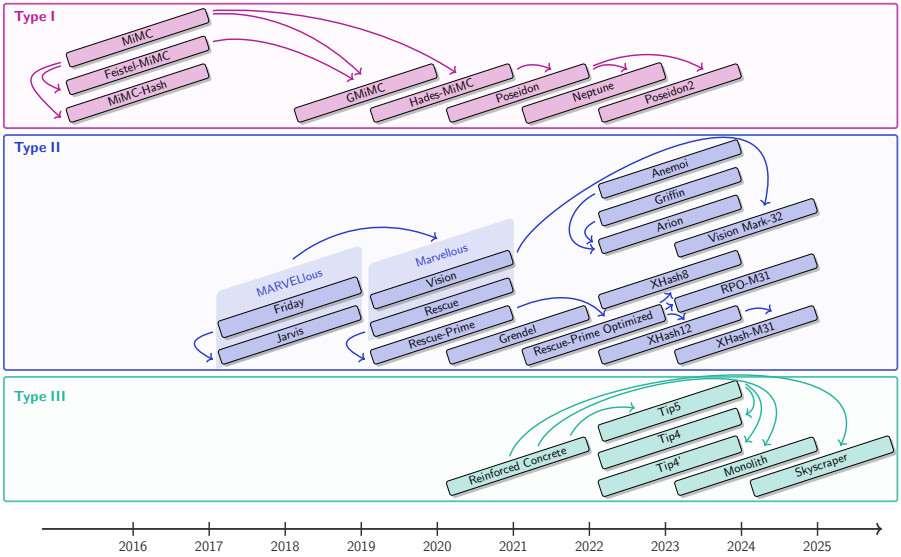
Orienté Arithmétisation

- Alphabet :

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

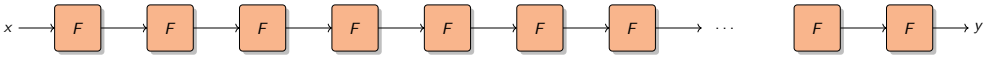
- Opérations :
  Arithmétique des grands corps finis
- Métrique :
  Minimiser le nombre de multiplications pour la vérification
- ≤ 8 **années de Cryptanalyse**

Primitives


Type I
MiMC
Feistel-MiMC
MiMC-Hash
GMiMC
Hades-MiMC
Poseidon
Neptune
Poseidon2
Type II
MARVELlous
Friday
Jarvis
Marvellous
Vision
Rescue
Rescue-Prime
Grendel
Anemoi
Griffin
Arion
Vision Mark-32
XHash8
RPO-M31
Rescue-Prime Optimized
XHash12
XHash-M31
Type III
Reinforced Concrete
Tip5
Tip4
Tip4'
Monolith
Skyscraper
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025

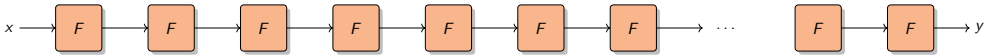
Type I

Primitives de bas degré



Type I

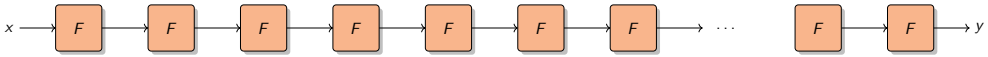
Primitives de bas degré



Degré	5	5^2	5^3	5^4	5^5	5^6	5^7	5^{79}	5^{80}

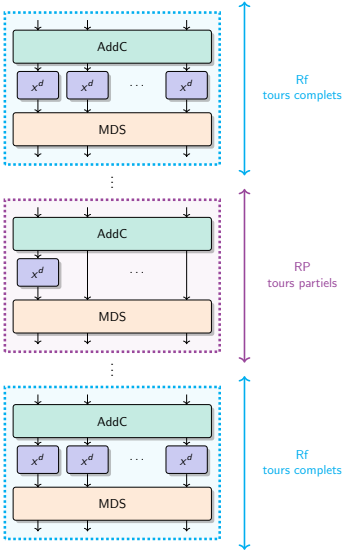
Type I

Primitives de bas degré



Degré	5	5^2	5^3	5^4	5^5	5^6	5^7		5^{79}	5^{80}
Contraintes	3	3×2	3×3	3×4	3×5	3×6	3×7		3×79	3×80

Exemple du Type I



Poseidon

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

- ⋆ Fonction non-linéaire :

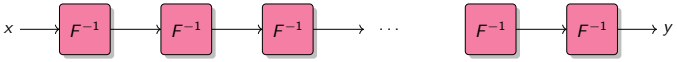
$$x \mapsto x^3$$

- ⋆ Nb de tours :

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

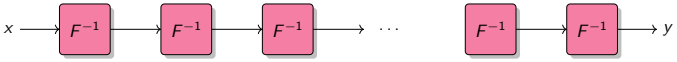
Type II

Relation d'équivalence



Type II

Relation d'équivalence



Degré 5^{-1} 5^{-2} 5^{-3} 5^{-19} 5^{-20}

Exemple

Dans $\mathbb{F}_p$ avec

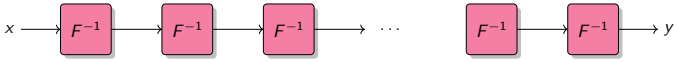
$$p = \texttt{0x73eda753299d7d483339d80809a1d80553bda402fffe5bfeffffffff00000001}$$

Si $F(x) = x^5$ alors $F^{-1}(x) = x^{5^{-1}}$ avec

$$5^{-1} = \texttt{0x2e5f0fbadd72321ce14a56699d73f002217f0e679998f19933333332ccccccccd}$$

Type II

Relation d'équivalence



Degré	5^{-1}	5^{-2}	5^{-3}	5^{-19}	5^{-20}
Contraintes	3×20	3×19	3×18	3×2	3

Exemple

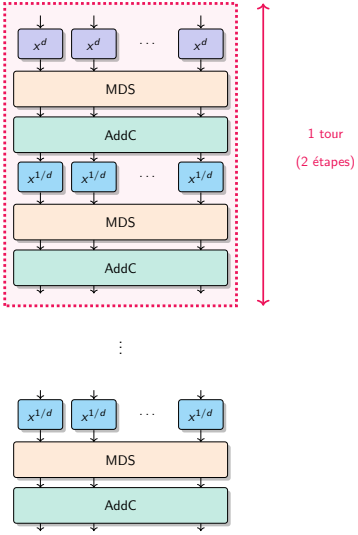
Dans $\mathbb{F}_p$ avec

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

Si $F(x) = x^5$ alors $F^{-1}(x) = x^{5^{-1}}$ avec

$$5^{-1} = \texttt{0x2e5f0fbadd72321ce14a56699d73f002217f0e679998f19933333332ccccccccd}$$

Exemple du Type II



Rescue

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

- ⋆ Fonctions non-linéaires :

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

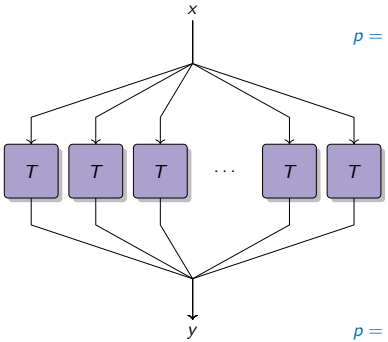
- ⋆ Nb de tours :

$$R = \text{de 8 à 26}$$

(2 étapes par tours)

Type III

Tables de valeurs

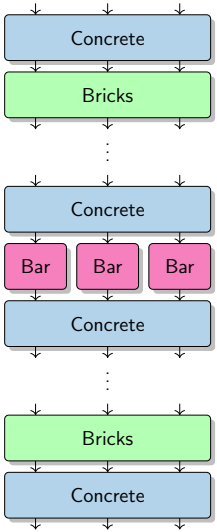


$\mathbb{F}_p$ avec
$p = 0x73eda753299d7d483339d80809a1d80553bda402fffe5bfeffffffff00000001$

$\mathbb{F}_2^8$
$(0,0,0,0,0,0,0,0) \ \ldots \ (1,1,1,1,1,1,1,1)$

$\mathbb{F}_p$ avec
$p = 0x73eda753299d7d483339d80809a1d80553bda402fffe5bfeffffffff00000001$

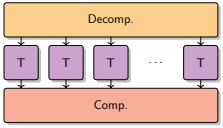
Exemple du Type III



Reinforced Concrete

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

- ⋆ Fonction non-linéaire :



- ⋆ Nb de tours :

$$R = 7$$

En résumé

	Type I Primitives de bas degré	**Type II** Relation d'équivalence	**Type III** Tables de valeurs
Alphabet	$\mathbb{F}_q^m$ pour différents q et m	$\mathbb{F}_q^m$ pour différents q et m	corps spécifiques
Nb de tours	beaucoup	peu	encore moins
Performance	rapide	lent	encore plus rapide
Nb de contraintes	souvent plus	moins	cela dépend du système de preuve

QUIZ ! !

- Les AOPs sont des primitives symétriques ou asymétriques ?
- A quel type de primitives (I, II, ou III) appartient l'AES ?
- Pourrait-on utiliser l'AES pour les protocoles avancés ?

A Retenir

Que sont ces AOPs ?

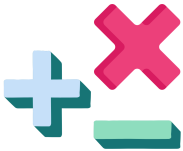
- ⋆ Une nouvelle sorte d'algorithmes symétriques
- ⋆ Dédiés à être combinés avec des protocoles avancés (FHE, MPC, ZKP)
- ⋆ On les classe en 3 catégories (I, II, III)

N'en faisons pas tout un fromage !

Calcul de contraintes

R1CS : un exemple facile

D'autres systèmes : Plonk, AIR, ...



Calcul de contraintes

Que signifie « efficace » pour les preuves à divulgation nulle de connaissance ?

Calcul de contraintes

Que signifie « efficace » pour les preuves à divulgation nulle de connaissance ?

« **Ça dépend** »

Calcul de contraintes

Que signifie « efficace » pour les preuves à divulgation nulle de connaissance ?

« **Ça dépend** »

Exemple

R1CS (Rank-1 Constraint System) : minimiser le nombre de multiplication

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

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

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

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

Calcul de contraintes

Que signifie « efficace » pour les preuves à divulgation nulle de connaissance ?

« **Ça dépend** »

Exemple

R1CS (Rank-1 Constraint System) : minimiser le nombre de multiplication

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

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

3 contraintes

Calcul de contraintes

Que signifie ≪ efficace ≫ pour les preuves à divulgation nulle de connaissance ?

≪ **Ça dépend** ≫

Exemple

R1CS (Rank-1 Constraint System) : minimiser le nombre de multiplication

$$y = x^7$$

Calcul de contraintes

Que signifie « efficace » pour les preuves à divulgation nulle de connaissance ?

« **Ça dépend** »

Exemple

R1CS (Rank-1 Constraint System) : minimiser le nombre de multiplication

$$y = x^7$$

$$t_0 = x \times x \qquad t_2 = t_1 \times t_0$$
$$t_1 = t_0 \times t_0 \qquad t_3 = t_2 \times x$$

Calcul de contraintes

Que signifie « efficace » pour les preuves à divulgation nulle de connaissance ?

« **Ça dépend** »

Exemple

R1CS (Rank-1 Constraint System) : minimiser le nombre de multiplication

$$y = x^7$$

$$t_0 = x \times x \qquad t_2 = t_1 \times t_0$$
$$t_1 = t_0 \times t_0 \qquad t_3 = t_2 \times x$$

4 contraintes

Développer ou factoriser ?

Expression factorisée

$$z = (x + y)^3 + 1$$

$t_0 = x + y$ $t_1 = t_0 \times t_0$ $t_2 = t_1 \times t_0$ $t_3 = t_2 + 1$

Développer ou factoriser ?

Expression factorisée

$$z = (x + y)^3 + 1$$

$t_0 = x + y$ $\quad t_1 = t_0 \times t_0$ $\quad t_2 = t_1 \times t_0$ $\quad t_3 = t_2 + 1$

2 contraintes

Développer ou factoriser ?

Expression factorisée

$$z = (x+y)^3 + 1$$

$t_0 = x + y$ $\quad t_1 = t_0 \times t_0$ $\quad t_2 = t_1 \times t_0$ $\quad t_3 = t_2 + 1$

2 contraintes

Expression développée

$$z = x^3 + 3x^2y + 3xy^2 + y^3$$

$t_0 = x \times x$ $\quad t_3 = 3 \cdot t_2$ $\quad t_6 = t_4 \times x$ $\quad t_9 = t_5 + t_7$

$t_1 = t_0 \times x$ $\quad t_4 = y \times y$ $\quad t_7 = 3 \cdot t_6$ $\quad t_{10} = t_8 + t_9$

$t_2 = t_0 \times y$ $\quad t_5 = t_4 \times y$ $\quad t_8 = t_1 + t_3$ $\quad t_{11} = t_{10} + 1$

Développer ou factoriser ?

Expression factorisée

$$z = (x+y)^3 + 1$$

$t_0 = x + y$ $\quad t_1 = t_0 \times t_0$ $\quad t_2 = t_1 \times t_0$ $\quad t_3 = t_2 + 1$

2 contraintes

Expression développée

$$z = x^3 + 3x^2y + 3xy^2 + y^3$$

$t_0 = x \times x$ $\quad t_3 = 3 \cdot t_2$ $\quad t_6 = t_4 \times x$ $\quad t_9 = t_5 + t_7$

$t_1 = t_0 \times x$ $\quad t_4 = y \times y$ $\quad t_7 = 3 \cdot t_6$ $\quad t_{10} = t_8 + t_9$

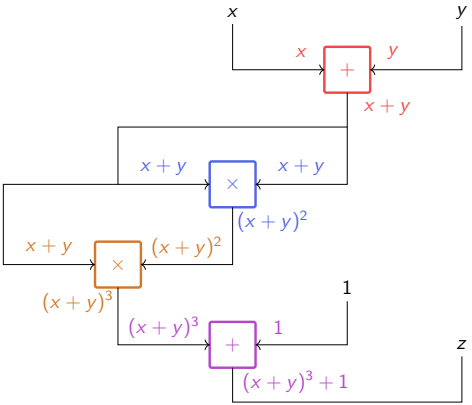
$t_2 = t_0 \times y$ $\quad t_5 = t_4 \times y$ $\quad t_8 = t_1 + t_3$ $\quad t_{11} = t_{10} + 1$

6 contraintes

Une représentation en circuit

Expression factorisée

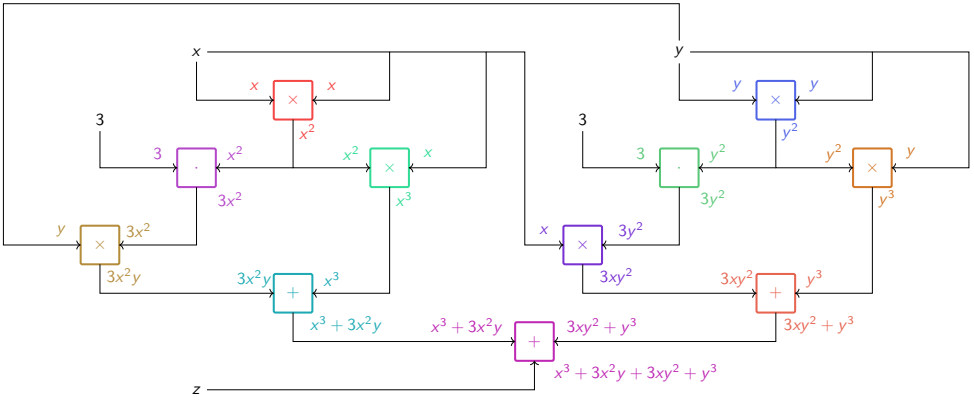
$$z = (x + y)^3 + 1$$



Une représentation en circuit

Expression développée

$$z = x^3 + 3x^2y + 3xy^2 + y^3$$



Développer ou factoriser ?

Expression factorisée

$$\begin{cases} w &= (2x+y)^3 + x^3 \\ z &= (x+2y)^3 + y^3 \end{cases}$$

$t_0 = 2 \cdot x$
$t_1 = t_0 + y$
$t_2 = t_1 \times t_1$

$t_3 = t_2 \times t_1$
$t_4 = 2 \cdot y$
$t_5 = t_4 + x$

$t_6 = t_5 \times t_5$
$t_7 = t_6 \times t_5$
$t_8 = x \times x$

$t_9 = t_8 \times x$
$t_{10} = y \times y$
$t_{11} = t_{10} \times y$

$t_{12} = t_3 + t_9$
$t_{13} = t_7 + t_{11}$

Développer ou factoriser ?

Expression factorisée

$$\begin{cases} w &= (2x+y)^3 + x^3 \\ z &= (x+2y)^3 + y^3 \end{cases}$$

$t_0 = 2 \cdot x$
$t_1 = t_0 + y$
$t_2 = t_1 \times t_1$

$t_3 = t_2 \times t_1$
$t_4 = 2 \cdot y$
$t_5 = t_4 + x$

$t_6 = t_5 \times t_5$
$t_7 = t_6 \times t_5$
$t_8 = x \times x$

$t_9 = t_8 \times x$
$t_{10} = y \times y$
$t_{11} = t_{10} \times y$

$t_{12} = t_3 + t_9$
$t_{13} = t_7 + t_{11}$

8 contraintes

Développer ou factoriser ?

Expression factorisée

$$\begin{cases} w &= (2x+y)^3 + x^3 \\ z &= (x+2y)^3 + y^3 \end{cases}$$

$t_0 = 2 \cdot x$
$t_1 = t_0 + y$
$t_2 = t_1 \times t_1$
$t_3 = t_2 \times t_1$
$t_4 = 2 \cdot y$
$t_5 = t_4 + x$
$t_6 = t_5 \times t_5$
$t_7 = t_6 \times t_5$
$t_8 = x \times x$
$t_9 = t_8 \times x$
$t_{10} = y \times y$
$t_{11} = t_{10} \times y$
$t_{12} = t_3 + t_9$
$t_{13} = t_7 + t_{11}$

8 contraintes

Expression développée

$$\begin{cases} w &= 9x^3 + 12x^2y + 6xy^2 + y^3 \\ z &= x^3 + 6x^2y + 12xy^2 + 9y^3 \end{cases}$$

$t_0 = x \times x$
$t_1 = t_0 \times x$
$t_2 = y \times y$
$t_3 = t_2 \times y$
$t_4 = t_0 \times y$
$t_5 = t_2 \times x$
$t_6 = 9 \cdot t_1$
$t_7 = 12 \cdot t_4$
$t_8 = 6 \cdot t_5$
$t_9 = 6 \cdot t_4$
$t_{10} = 12 \cdot t_5$
$t_{11} = 9 \cdot t_3$
$t_{12} = t_6 + t_7$
$t_{13} = t_{12} + t_8$
$t_{14} = t_{13} + t_3$
$t_{15} = t_1 + t_9$
$t_{16} = t_{15} + t_{10}$
$t_{17} = t_{16} + t_{11}$

Développer ou factoriser ?

Expression factorisée

$$\begin{cases} w &= (2x+y)^3 + x^3 \\ z &= (x+2y)^3 + y^3 \end{cases}$$

$t_0 = 2 \cdot x$
$t_1 = t_0 + y$
$t_2 = t_1 \times t_1$
$t_3 = t_2 \times t_1$
$t_4 = 2 \cdot y$
$t_5 = t_4 + x$
$t_6 = t_5 \times t_5$
$t_7 = t_6 \times t_5$
$t_8 = x \times x$
$t_9 = t_8 \times x$
$t_{10} = y \times y$
$t_{11} = t_{10} \times y$
$t_{12} = t_3 + t_9$
$t_{13} = t_7 + t_{11}$

8 contraintes

Expression développée

$$\begin{cases} w &= 9x^3 + 12x^2y + 6xy^2 + y^3 \\ z &= x^3 + 6x^2y + 12xy^2 + 9y^3 \end{cases}$$

$t_0 = x \times x$
$t_1 = t_0 \times x$
$t_2 = y \times y$
$t_3 = t_2 \times y$
$t_4 = t_0 \times y$
$t_5 = t_2 \times x$
$t_6 = 9 \cdot t_1$
$t_7 = 12 \cdot t_4$
$t_8 = 6 \cdot t_5$
$t_9 = 6 \cdot t_4$
$t_{10} = 12 \cdot t_5$
$t_{11} = 9 \cdot t_3$
$t_{12} = t_6 + t_7$
$t_{13} = t_{12} + t_8$
$t_{14} = t_{13} + t_3$
$t_{15} = t_1 + t_9$
$t_{16} = t_{15} + t_{10}$
$t_{17} = t_{16} + t_{11}$

6 contraintes

Contraintes Plonk

Les additions comptent aussi !

$$z = (x + y)^3 + 1$$

R1CS

$t_0 = x + y$ $\quad t_1 = t_0 \times t_0$ $\quad t_2 = t_1 \times t_0$ $\quad t_3 = t_2 + 1$

Contraintes Plonk

Les additions comptent aussi !

$$z = (x + y)^3 + 1$$

R1CS

$t_0 = x + y$ $\quad t_1 = t_0 \times t_0$ $\quad t_2 = t_1 \times t_0$ $\quad t_3 = t_2 + 1$

2 contraintes

Contraintes Plonk

Les additions comptent aussi !

$$z = (x + y)^3 + 1$$

R1CS

$t_0 = x + y$ $\quad t_1 = t_0 \times t_0$ $\quad t_2 = t_1 \times t_0$ $\quad t_3 = t_2 + 1$

2 contraintes

Plonk

$t_0 = x + y$ $\quad t_1 = t_0 \times t_0$ $\quad t_2 = t_1 \times t_0$ $\quad t_3 = t_2 + 1$

Contraintes Plonk

Les additions comptent aussi !

$$z = (x + y)^3 + 1$$

R1CS

$t_0 = x + y$ $\quad t_1 = t_0 \times t_0$ $\quad t_2 = t_1 \times t_0$ $\quad t_3 = t_2 + 1$

2 contraintes

Plonk

$t_0 = x + y$ $\quad t_1 = t_0 \times t_0$ $\quad t_2 = t_1 \times t_0$ $\quad t_3 = t_2 + 1$

3 contraintes

Contraintes Plonk

Les additions comptent aussi !

$$z = (x + y)^3 + 1$$

R1CS

$t_0 = x + y$ $\quad t_1 = t_0 \times t_0$ $\quad t_2 = t_1 \times t_0$ $\quad t_3 = t_2 + 1$

2 contraintes

Plonk

$t_0 = x + y$ $\quad t_1 = t_0 \times t_0$ $\quad t_2 = t_1 \times t_0$ $\quad t_3 = t_2 + 1$

3 contraintes

Mais c'est plus compliqué que ça... (customs gates)

Contraintes AIR

Exemple : Calcul des termes de la suite de Fibonacci

Calcul du n-ième terme
pour n donné

$a = 1$
$b = 0$
`Pour` i `de` 0 `à` $n-1$
    $a = a + b$
    $b = a$
`Retourner` a

Contraintes AIR

Exemple : Calcul des termes de la suite de Fibonacci

Calcul du n-ième terme pour n donné

$a = 1$
$b = 0$
Pour i de 0 à $n-1$
    $a = a + b$
    $b = a$
Retourner a

i	a_i	b_i
0	1	0
1	1	1
2	2	1
3	3	2
⋮	⋮	⋮
$n-1$	a_{n-1}	b_{n-1}

Trace de l'exécution

Contraintes AIR

Exemple : Calcul des termes de la suite de Fibonacci

Calcul du n-ième terme pour n donné

$a = 1$
$b = 0$
`Pour` i `de` 0 `à` $n-1$
    $a = a + b$
    $b = a$
`Retourner` a

i	a_i	b_i
0	1	0
1	1	1
2	2	1
3	3	2
⋮	⋮	⋮
$n-1$	a_{n-1}	b_{n-1}

Trace de l'exécution

Vérification intermédiaire des lignes 3 et 4 ($i = 2$ et $i = 3$)

$$a_3 = 3 = 2 + 1 = a_2 + b_2 \quad \text{et} \quad b_3 = 2 = a_2 \ .$$

Contraintes AIR

Exemple : Calcul des termes de la suite de Fibonacci

Calcul du n-ième terme pour n donné

$a = 1$
$b = 0$
Pour i de 0 à $n-1$
    $a = a + b$
    $b = a$
Retourner a

i	a_i	b_i
0	1	0
1	1	1
2	2	1
3	3	2
$\vdots$	$\vdots$	$\vdots$
$n-1$	a_{n-1}	b_{n-1}

Trace de l'exécution

Système de contraintes

$$\begin{cases} a_0 = 1 & \text{à la 1ère ligne,} \\ b_0 = 0 & \text{à la 1ère ligne,} \\ a_{i+1} = a_i + b_i & \text{pour } 0 \leq i \leq n-2\ , \\ b_{i+1} = a_i & \text{pour } 0 \leq i \leq n-2\ , \\ a_{n-1} = Fib(n-1) & \text{à la ligne } n-1\ . \end{cases}$$

Vérification intermédiaire des lignes 3 et 4 ($i = 2$ et $i = 3$)

$$a_3 = 3 = 2 + 1 = a_2 + b_2 \quad \text{et} \quad b_3 = 2 = a_2\ .$$

QUIZ ! !

- La multiplication scalaire est un type de contraintes R1CS. Vrai ou Faux ?
- Les additions importent lors du calcul des contraintes Plonk. Vrai ou Faux ?
- Les portes personnalisées réduisent les contraintes AIR. Vrai ou Faux ?
- Combien faut-il de contraintes R1CS pour vérifier $y = 3 \cdot x + 1$?
- Combien faut-il de contraintes R1CS pour vérifier $y = 3 \cdot x^2 + x$?
- Comment obtenir le moins de contraintes R1CS pour vérifier $y = (x + 1)^2$?

A Retenir

Comment minimiser le nombre de contraintes ?

- ⋆ Cela dépend du système de preuve considéré
- ⋆ Réduire le nombre de portes multiplicatives (souvent)
- ⋆ Factoriser ou développer ?
- ⋆ Des portes personnalisées

Attaques algébriques des AOPs

Quelques définitions

Astuces pour réduire leur complexité

Importance de la modélisation

Le problème CICO

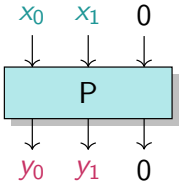
CICO : Constrained Input Constrained Output

Définition

Soit $P : \mathbb{F}_q^{r+c} \to \mathbb{F}_q^{r+c}$. Le problème **CICO** est :

Trouver $X, Y \in \mathbb{F}_q^r$ tel que

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



Nécessité de résoudre un système polynomial

Résolution de systèmes polynomiaux

⋆ Système **univarié** : trouver les racines de $\mathcal{P}_j \in \mathbb{F}_q[X]$

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

Résolution de systèmes polynomiaux

⋆ Système **univarié** : trouver les racines de $\mathcal{P}_j \in \mathbb{F}_q[X]$

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

⋆ Système **multivarié** : trouver les racines de $\mathcal{P}_j \in \mathbb{F}_q[X_0, \ldots, X_{n-1}]$

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

Division Euclidienne

⋆ Entiers

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

Exemple : division de 2025 par 100

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

Division Euclidienne

⋆ Entiers

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

Exemple : division de 2025 par 100

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

⋆ Polynomes Univariés

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

Exemple : division de $X^5 + 2X^3 + 3X$ par X^2

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

Division Euclidienne

⋆ Entiers

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

Exemple : division de 2025 par 100

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

⋆ Polynomes Univariés

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

Exemple : division de $X^5 + 2X^3 + 3X$ par X^2

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

⋆ Polynomes Multivariés

Division Euclidienne

★ Entiers

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

Exemple : division de 2025 par 100

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

★ Polynomes Univariés

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

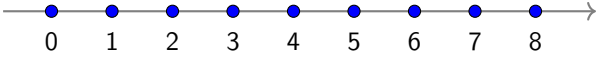
Exemple : division de $X^5 + 2X^3 + 3X$ par X^2

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

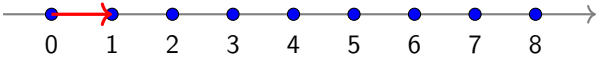
★ Polynomes Multivariés

Nécessité d'un ordre monomial

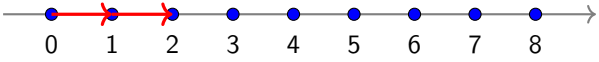
Ordre monomial



Ordre monomial

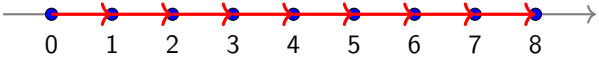


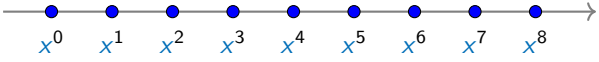
Ordre monomial



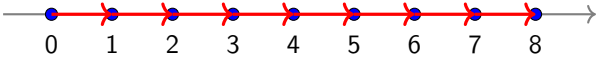
Ordre monomial

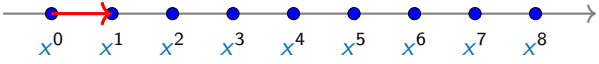
Ordre monomial



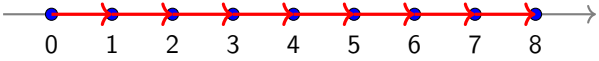


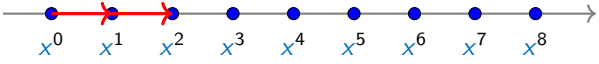
Ordre monomial



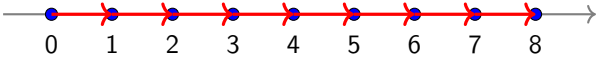


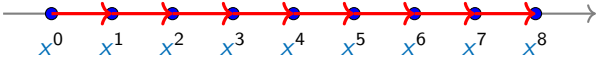
Ordre monomial



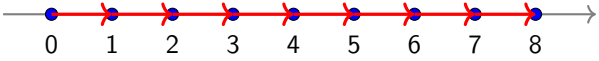


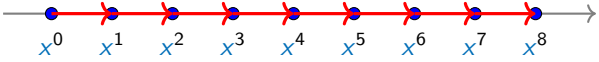
Ordre monomial





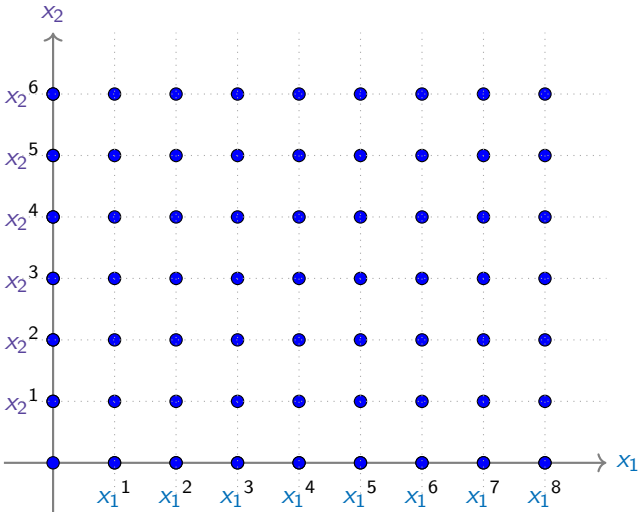
Ordre monomial





Qu'en est-il du cas multivarié ?

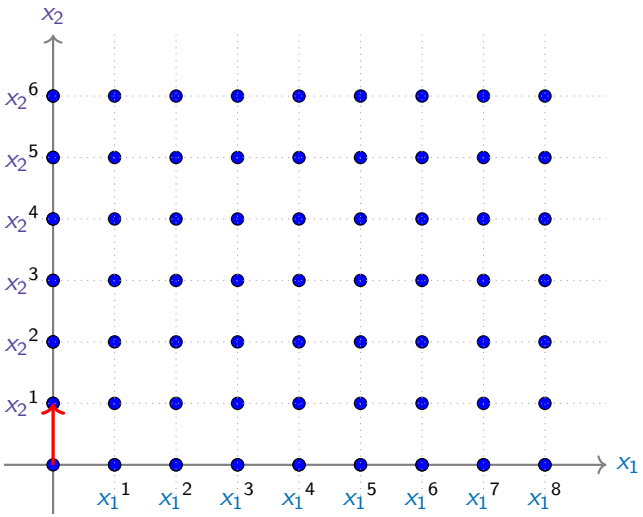
Ordre lexicographique



Ordre : x_1 est plus grand que n'importe quelle puissance de x_2.

$$x_1 > {x_2}^n$$

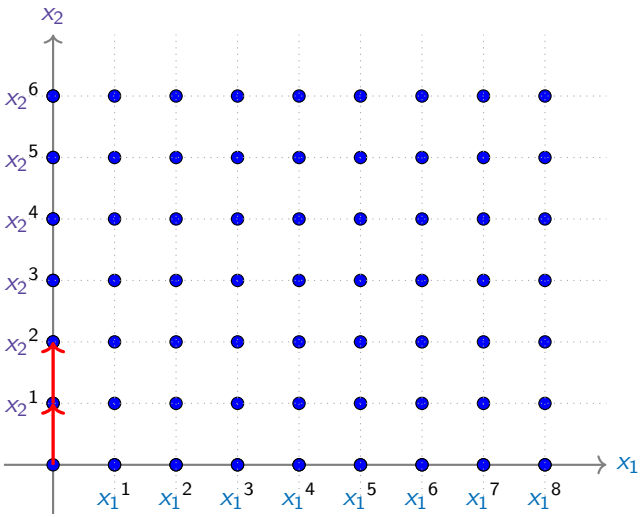
Ordre lexicographique



Ordre : x_1 est plus grand que n'importe quelle puissance de x_2.

$$x_1 > {x_2}^n$$

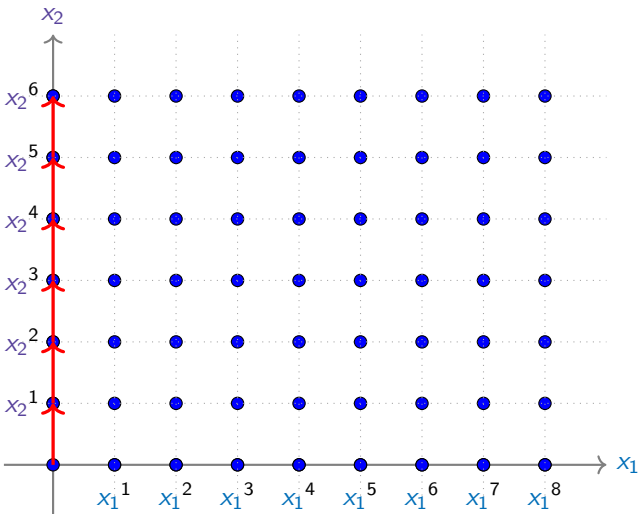
Ordre lexicographique



Ordre : x_1 est plus grand que n'importe quelle puissance de x_2.

$$x_1 > {x_2}^n$$

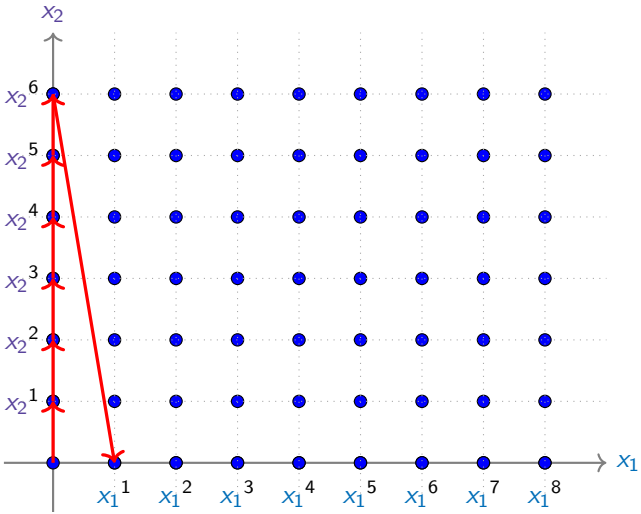
Ordre lexicographique



Ordre : x_1 est plus grand que n'importe quelle puissance de x_2.

$$x_1 > {x_2}^n$$

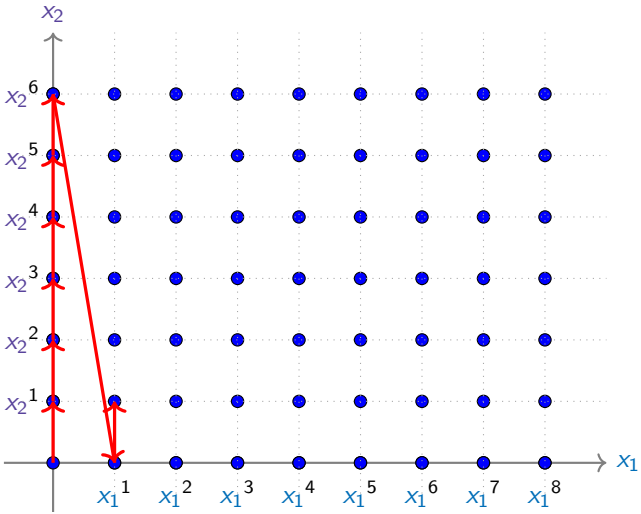
Ordre lexicographique



Ordre : x_1 est plus grand que n'importe quelle puissance de x_2.

$$x_1 > {x_2}^n$$

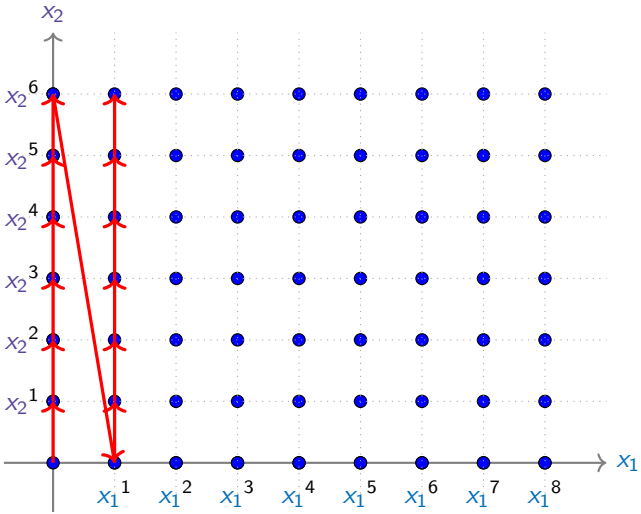
Ordre lexicographique



Ordre : x_1 est plus grand que n'importe quelle puissance de x_2.

$$x_1 > {x_2}^n$$

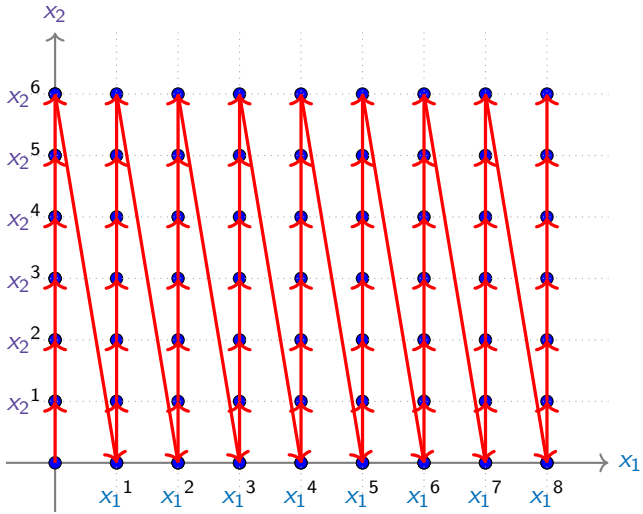
Ordre lexicographique



Ordre : x_1 est plus grand que n'importe quelle puissance de x_2.

$$x_1 > {x_2}^n$$

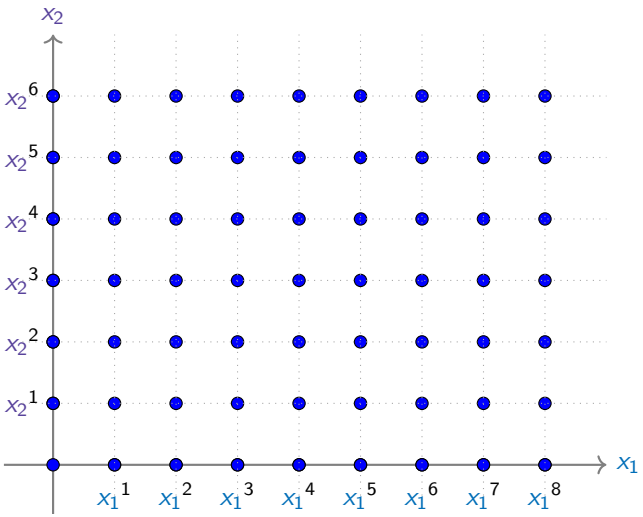
Ordre lexicographique



Ordre : x_1 est plus grand que n'importe quelle puissance de x_2.

$$x_1 > {x_2}^n$$

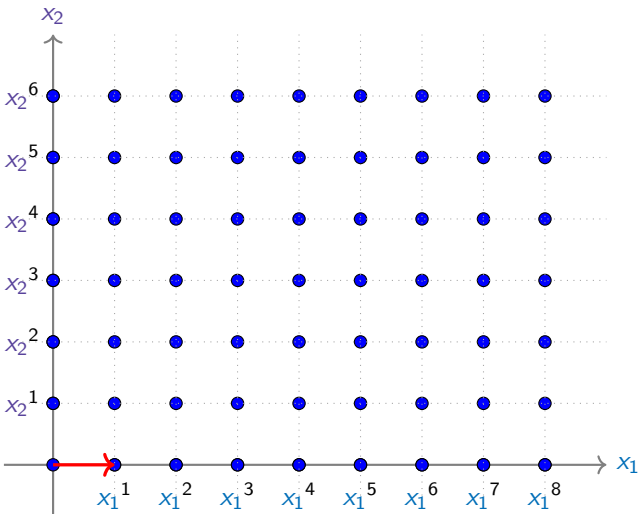
Ordre lexicographique inverse



Ordre : x_2 est plus grand que n'importe quelle puissance de x_1.

$$x_2 > {x_1}^n$$

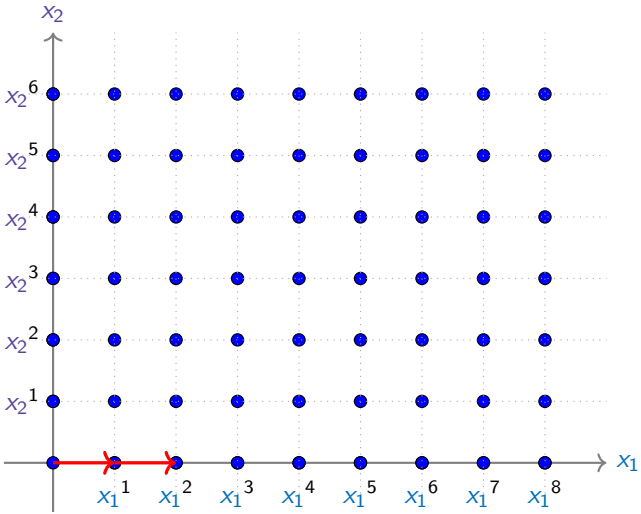
Ordre lexicographique inverse



Ordre : x_2 est plus grand que n'importe quelle puissance de x_1.

$$x_2 > {x_1}^n$$

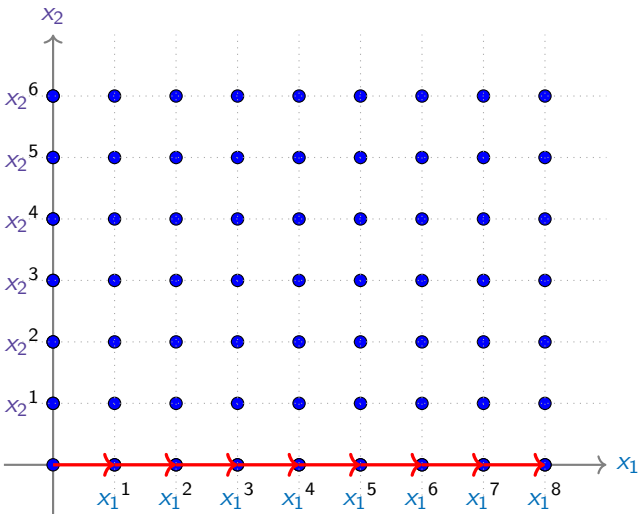
Ordre lexicographique inverse



Ordre : x_2 est plus grand que n'importe quelle puissance de x_1.

$$x_2 > {x_1}^n$$

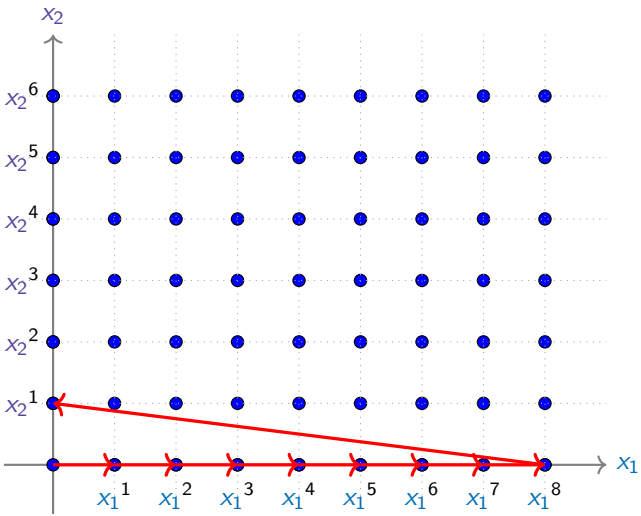
Ordre lexicographique inverse



Ordre : x_2 est plus grand que n'importe quelle puissance de x_1.

$$x_2 > {x_1}^n$$

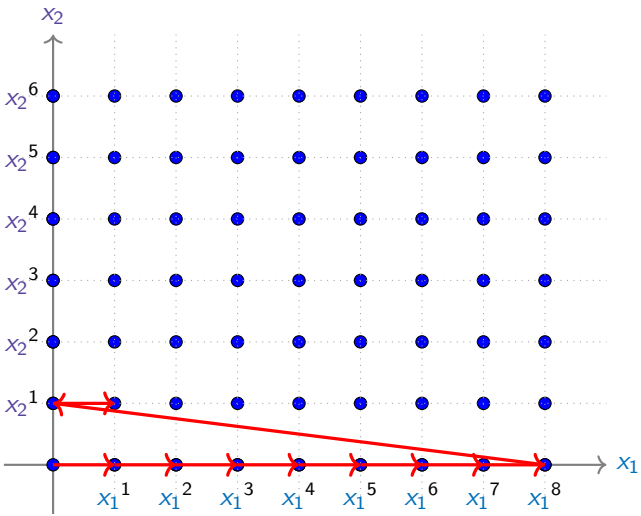
Ordre lexicographique inverse



Ordre : x_2 est plus grand que n'importe quelle puissance de x_1.

$$x_2 > {x_1}^n$$

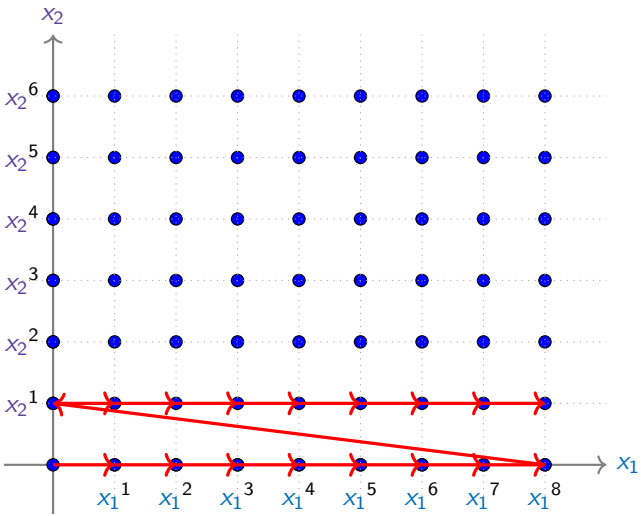
Ordre lexicographique inverse



Ordre : x_2 est plus grand que n'importe quelle puissance de x_1.

$$x_2 > {x_1}^n$$

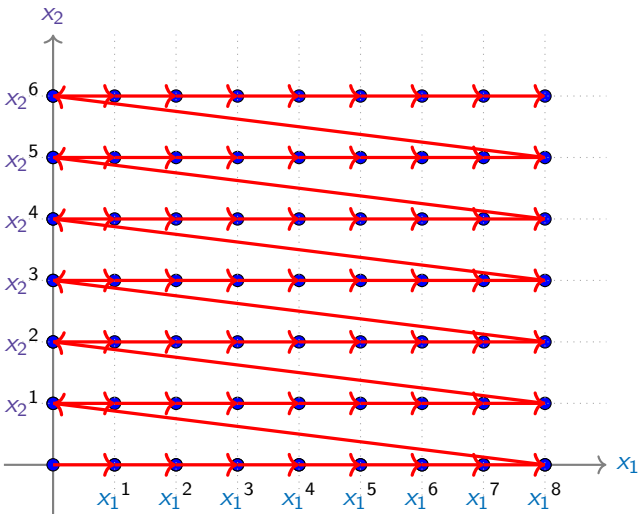
Ordre lexicographique inverse



Ordre : x_2 est plus grand que n'importe quelle puissance de x_1.

$$x_2 > {x_1}^n$$

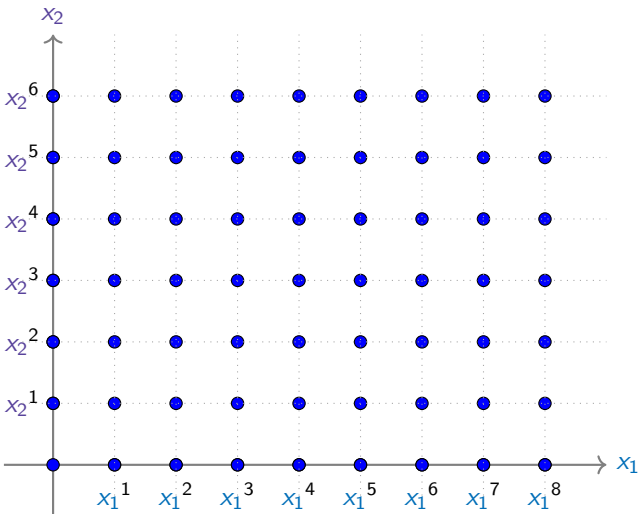
Ordre lexicographique inverse



Ordre : x_2 est plus grand que n'importe quelle puissance de x_1.

$$x_2 > {x_1}^n$$

Ordre lexicographique gradué



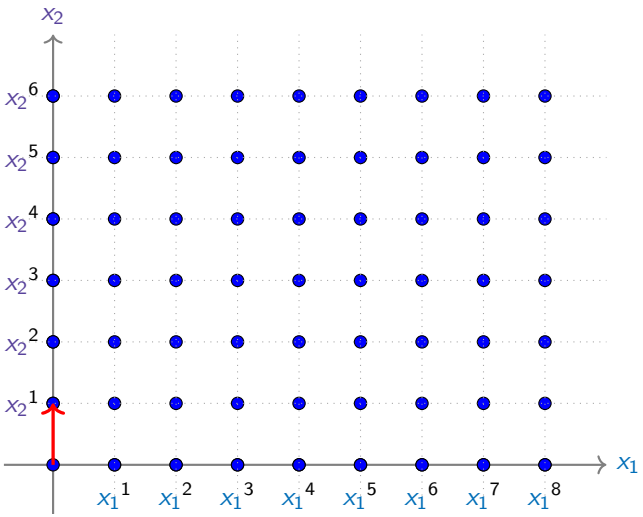
Ordre : L'élément de plus haut degré est le plus grand.

$$\begin{cases} {x_1}^{n_1}{x_2}^{n_2} & > {x_1}^{m_1}{x_2}^{m_2} \\ n_1 + n_2 & > m_1 + m_2 \end{cases}$$

S'il y a égalité, x_1 est plus grand x_2.

$$x_1 > x_2$$

Ordre lexicographique gradué



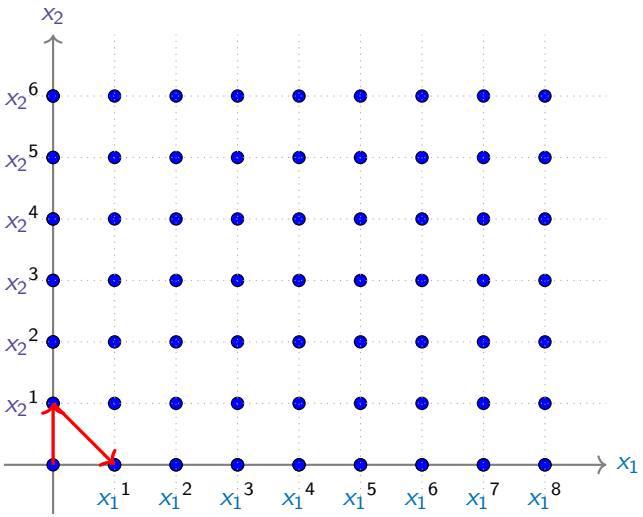
Ordre : L'élément de plus haut degré est le plus grand.

$$\begin{cases} {x_1}^{n_1}{x_2}^{n_2} & > {x_1}^{m_1}{x_2}^{m_2} \\ n_1 + n_2 & > m_1 + m_2 \end{cases}$$

S'il y a égalité, x_1 est plus grand x_2.

$$x_1 > x_2$$

Ordre lexicographique gradué



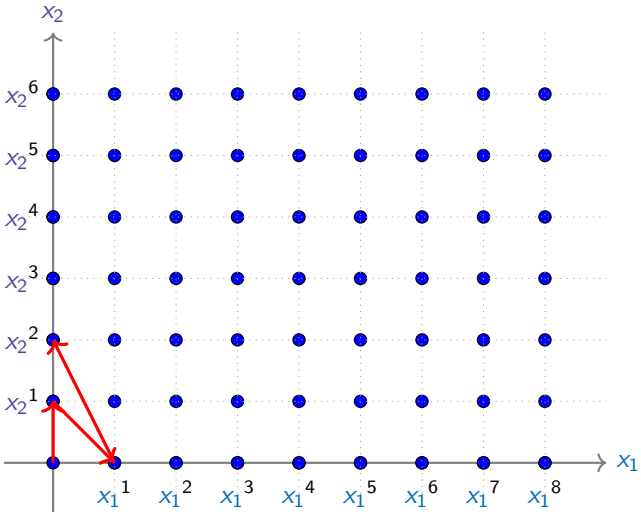
Ordre : L'élément de plus haut degré est le plus grand.

$$\begin{cases} {x_1}^{n_1}{x_2}^{n_2} & > {x_1}^{m_1}{x_2}^{m_2} \\ n_1 + n_2 & > m_1 + m_2 \end{cases}$$

S'il y a égalité, x_1 est plus grand x_2.

$$x_1 > x_2$$

Ordre lexicographique gradué



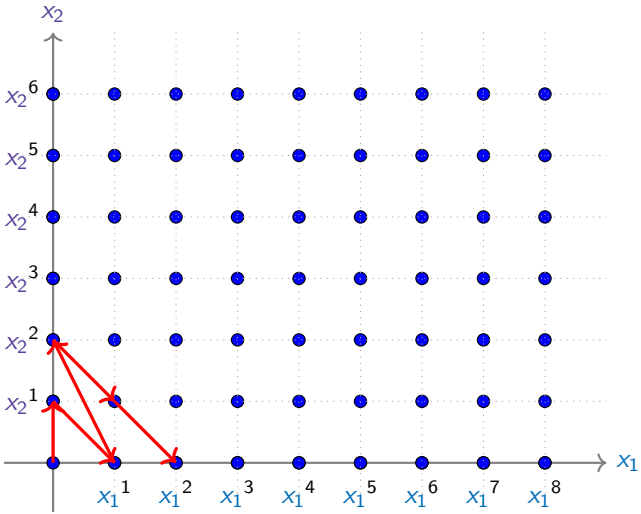
Ordre : L'élément de plus haut degré est le plus grand.

$$\begin{cases} {x_1}^{n_1}{x_2}^{n_2} & > {x_1}^{m_1}{x_2}^{m_2} \\ n_1 + n_2 & > m_1 + m_2 \end{cases}$$

S'il y a égalité, x_1 est plus grand x_2.

$$x_1 > x_2$$

Ordre lexicographique gradué



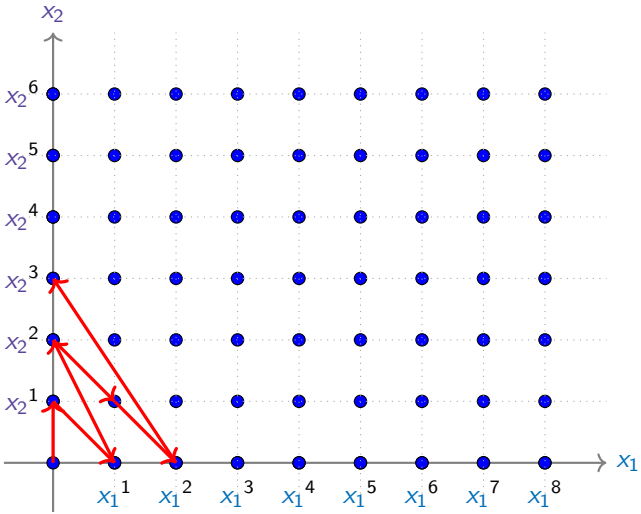
Ordre : L'élément de plus haut degré est le plus grand.

$$\begin{cases} {x_1}^{n_1}{x_2}^{n_2} & > {x_1}^{m_1}{x_2}^{m_2} \\ n_1 + n_2 & > m_1 + m_2 \end{cases}$$

S'il y a égalité, x_1 est plus grand x_2.

$$x_1 > x_2$$

Ordre lexicographique gradué



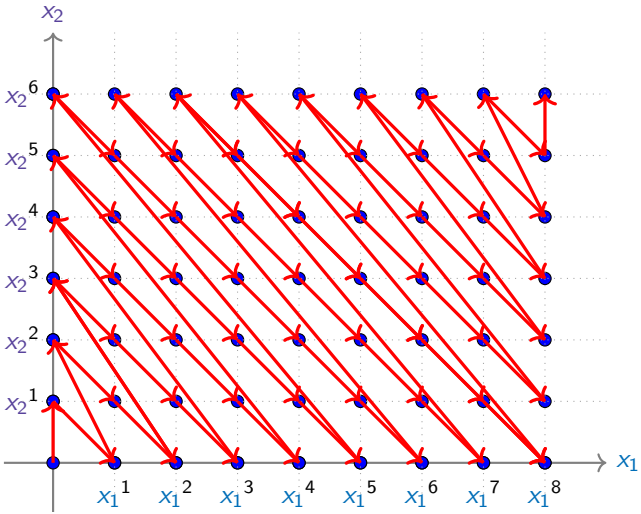
Ordre : L'élément de plus haut degré est le plus grand.

$$\begin{cases} {x_1}^{n_1}{x_2}^{n_2} & > {x_1}^{m_1}{x_2}^{m_2} \\ n_1 + n_2 & > m_1 + m_2 \end{cases}$$

S'il y a égalité, x_1 est plus grand x_2.

$$x_1 > x_2$$

Ordre lexicographique gradué



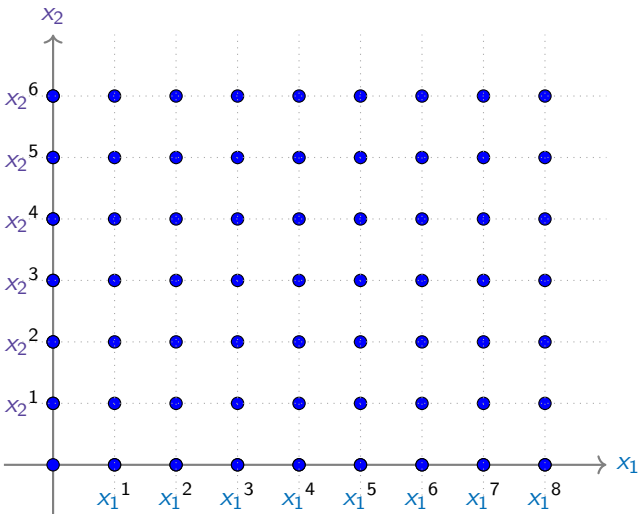
Ordre : L'élément de plus haut degré est le plus grand.

$$\begin{cases} {x_1}^{n_1}{x_2}^{n_2} & > {x_1}^{m_1}{x_2}^{m_2} \\ n_1 + n_2 & > m_1 + m_2 \end{cases}$$

S'il y a égalité, x_1 est plus grand x_2.

$$x_1 > x_2$$

Ordre lexicographique inverse gradué



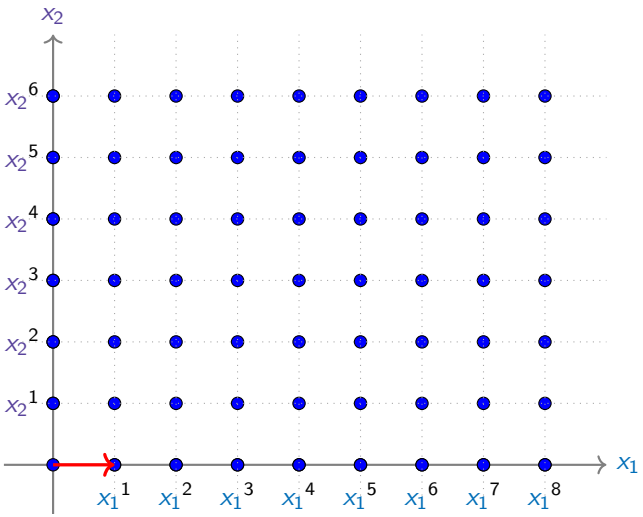
Ordre : L'élément de plus haut degré est le plus grand.

$$\begin{cases} {x_1}^{n_1}{x_2}^{n_2} & > {x_1}^{m_1}{x_2}^{m_2} \\ n_1 + n_2 & > m_1 + m_2 \end{cases}$$

S'il y a égalité, x_2 est plus grand x_1.

$$x_2 > x_1$$

Ordre lexicographique inverse gradué



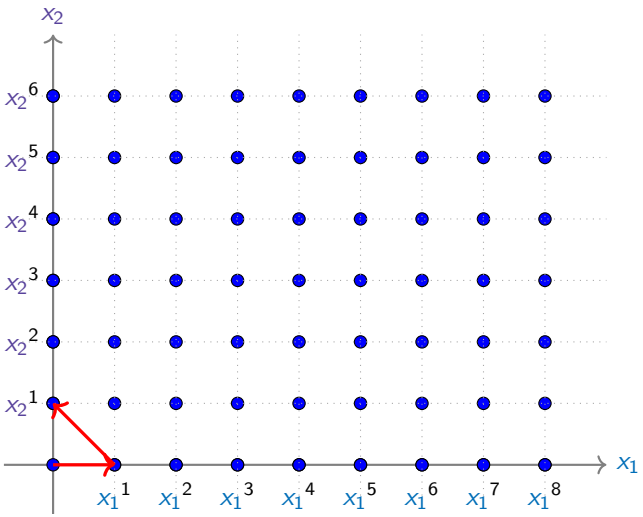
Ordre : L'élément de plus haut degré est le plus grand.

$$\begin{cases} {x_1}^{n_1}{x_2}^{n_2} & > {x_1}^{m_1}{x_2}^{m_2} \\ n_1 + n_2 & > m_1 + m_2 \end{cases}$$

S'il y a égalité, x_2 est plus grand x_1.

$$x_2 > x_1$$

Ordre lexicographique inverse gradué



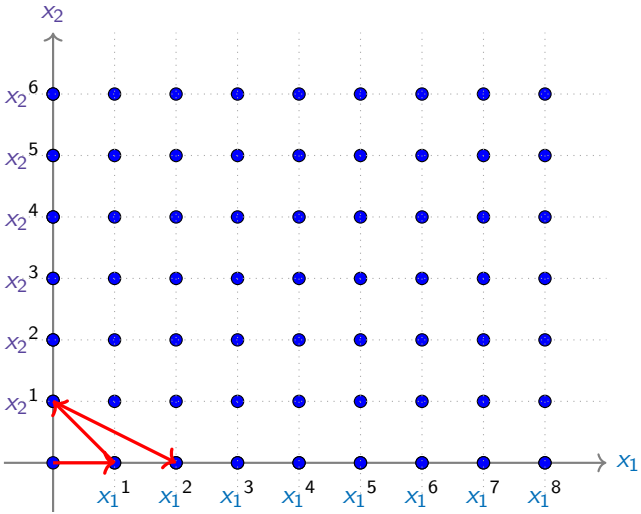
Ordre : L'élément de plus haut degré est le plus grand.

$$\begin{cases} {x_1}^{n_1}{x_2}^{n_2} & > {x_1}^{m_1}{x_2}^{m_2} \\ n_1 + n_2 & > m_1 + m_2 \end{cases}$$

S'il y a égalité, x_2 est plus grand x_1.

$$x_2 > x_1$$

Ordre lexicographique inverse gradué



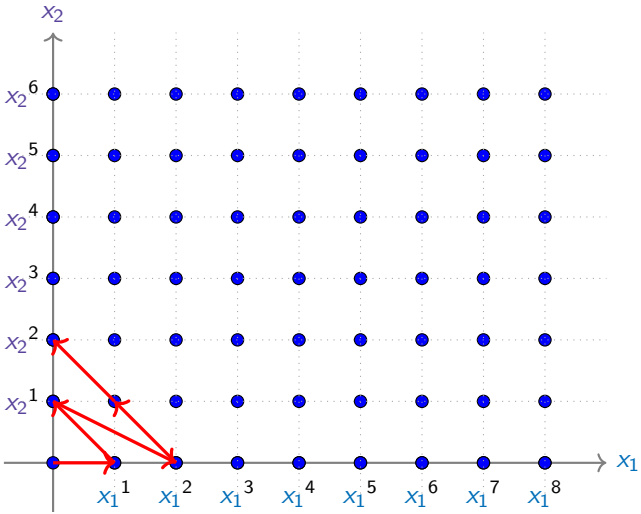
Ordre : L'élément de plus haut degré est le plus grand.

$$\begin{cases} {x_1}^{n_1}{x_2}^{n_2} & > {x_1}^{m_1}{x_2}^{m_2} \\ n_1 + n_2 & > m_1 + m_2 \end{cases}$$

S'il y a égalité, x_2 est plus grand x_1.

$$x_2 > x_1$$

Ordre lexicographique inverse gradué



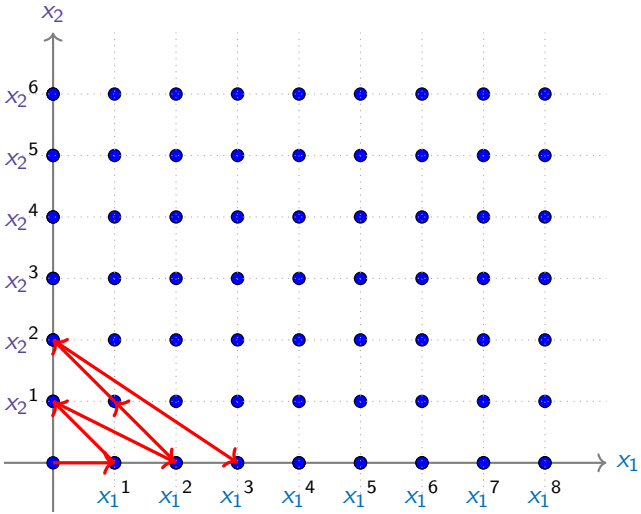
Ordre : L'élément de plus haut degré est le plus grand.

$$\begin{cases} {x_1}^{n_1}{x_2}^{n_2} & > {x_1}^{m_1}{x_2}^{m_2} \\ n_1 + n_2 & > m_1 + m_2 \end{cases}$$

S'il y a égalité, x_2 est plus grand x_1.

$$x_2 > x_1$$

Ordre lexicographique inverse gradué



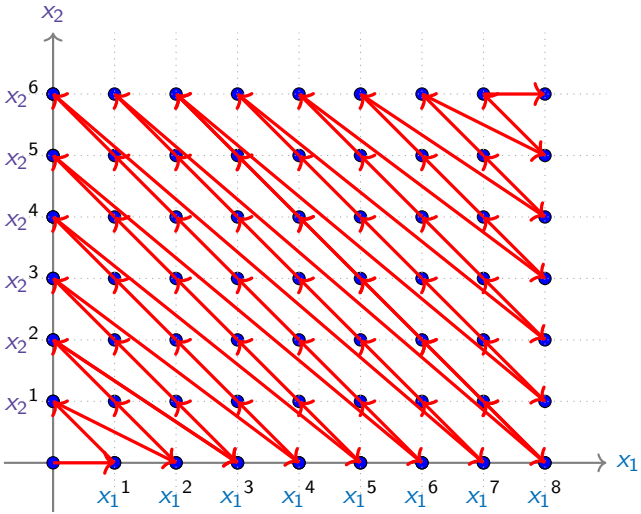
Ordre : L'élément de plus haut degré est le plus grand.

$$\begin{cases} {x_1}^{n_1}{x_2}^{n_2} & > {x_1}^{m_1}{x_2}^{m_2} \\ n_1 + n_2 & > m_1 + m_2 \end{cases}$$

S'il y a égalité, x_2 est plus grand x_1.

$$x_2 > x_1$$

Ordre lexicographique inverse gradué



Ordre : L'élément de plus haut degré est le plus grand.

$$\begin{cases} {x_1}^{n_1}{x_2}^{n_2} & > {x_1}^{m_1}{x_2}^{m_2} \\ n_1 + n_2 & > m_1 + m_2 \end{cases}$$

S'il y a égalité, x_2 est plus grand x_1.

$$x_2 > x_1$$

Ordre monomial

Quelques ordres dans $\mathbb{F}_q[x_1, x_2, \ldots, x_n]$.

Ordre lexicographique (lex)

Comparer les degrés en la plus grande variable, puis la seconde plus grande, ...

$$x_1 > x_2 > \ldots > x_n, \qquad x_1 > {x_2}^2,$$

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

Ordre monomial

Quelques ordres dans $\mathbb{F}_q[x_1, x_2, \ldots, x_n]$.

Ordre lexicographique (lex)

Comparer les degrés en la plus grande variable, puis la seconde plus grande, ...

$$x_1 > x_2 > \ldots > x_n, \qquad x_1 > {x_2}^2,$$

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

Ordre lex. gradué (grlex)

Comparer le degré total, puis appliquer l'ordre lex. si égalité.

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

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

Ordre monomial

Quelques ordres dans $\mathbb{F}_q[x_1, x_2, \ldots, x_n]$.

Ordre lexicographique (lex)

Comparer les degrés en la plus grande variable, puis la seconde plus grande, ...

$$x_1 > x_2 > \ldots > x_n, \qquad x_1 > {x_2}^2,$$

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

Ordre lex. gradué (grlex)

Comparer le degré total, puis appliquer l'ordre lex. si égalité.

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

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

Ordre lex. inverse gradué (grevlex)

Comparer le degré total, puis appliquer l'ordre lex. inverse si égalité.

$$x_1 < x_2 < \ldots < x_n, \qquad x_1 < {x_2}^2,$$

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

Ordre monomial

Quelques ordres dans $\mathbb{F}_q[x_1, x_2, \ldots, x_n]$.

Ordre lexicographique (lex)

Comparer les degrés en la plus grande variable, puis la seconde plus grande, ...

$$x_1 > x_2 > \ldots > x_n, \qquad x_1 > {x_2}^2,$$

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

Ordre lex. gradué (grlex)

Comparer le degré total, puis appliquer l'ordre lex. si égalité.

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

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

Ordre lex. inverse gradué (grevlex)

Comparer le degré total, puis appliquer l'ordre lex. inverse si égalité.

$$x_1 < x_2 < \ldots < x_n, \qquad x_1 < {x_2}^2,$$

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

Ordre lex. pondéré

Comparer la somme pondérée des degrés, puis appliquer ordre lex. gradué.

Si $\mathrm{wt}(x_1) = 3$, $\mathrm{wt}(x_2) = 1$ et $\mathrm{wt}(x_n) = 4$, alors

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

Résolution de systèmes polynomiaux

- Système **univarié** : trouver les racines de $\mathcal{P}_j \in \mathbb{F}_q[X]$

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

- Système **multivarié** : trouver les racines de $\mathcal{P}_j \in \mathbb{F}_q[X_0, \ldots, X_{n-1}]$

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

  - Calculer une BG avec l'ordre grevlex (algorithme **F5**)
  - Convertir en une BG avec l'ordre lex (algorithme **FGLM**)
  - Trouver les racines dans $\mathbb{F}_q^n$ des polynomes de la BG en résolvant un système univarié.

Stratégies

Comment résoudre efficacement un système polynomial pour construire des attaques algébriques ?

Stratégies

Comment résoudre efficacement un système polynomial pour construire des attaques algébriques ?

- ⋆ en contournant certains tours de constructions itérées
- ⋆ en changeant le modèle
- ⋆ en changeant l'ordre

Stratégies

Comment résoudre efficacement un système polynomial pour construire des attaques algébriques ?

- ⋆ en contournant certains tours de constructions itérées
- ⋆ en changeant le modèle
- ⋆ en changeant l'ordre
- ⋆ .... en ne faisant rien ? ?

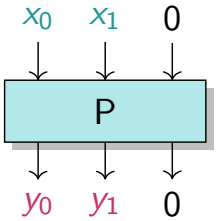
Challenges de la Fondation Ethereum

https://www.zkhashbounties.info/

(Novembre 2021)

Résolution du problème CICO

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



Challenges Ethereum : résoudre le problème CICO pour des AOP avec $q \sim 2^{64}$ premier

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

Challenges de Cryptanalyse

Catégorie	Paramètres	Niveau de Sécurité	Gain
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*

Catégorie	Paramètres	Niveau de Sécurité	Gain
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*

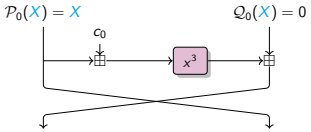
Catégorie	Paramètres	Niveau de Sécurité	Gain
Easy	RP $= 3$	8	\$2,000
Easy	RP $= 8$	16	\$4,000
Medium	RP $= 13$	24	\$6,000
Hard	RP $= 19$	32	\$12,000
Hard	RP $= 24$	40	\$26,000

(c) *Poseidon*

Catégorie	Paramètres	Niveau de Sécurité	Gain
Easy	$p = 281474976710597$	24	\$4,000
Medium	$p = 72057594037926839$	28	\$6,000
Hard	$p = 18446744073709551557$	32	\$12,000

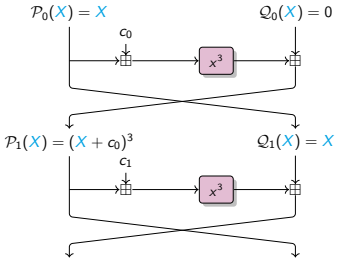
(d) *Reinforced Concrete*

Feistel-MiMC



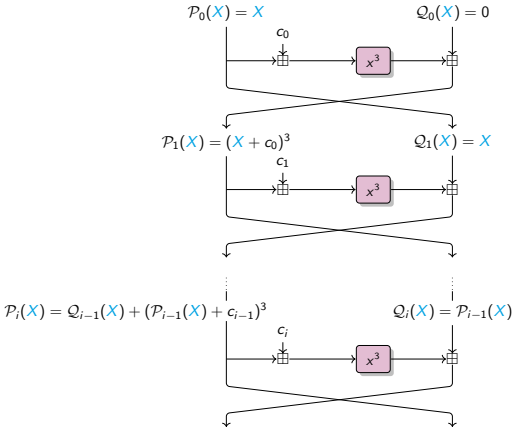
$$\begin{cases} \mathcal{P}_0(X) & = X \\ \mathcal{Q}_0(X) & = 0 \end{cases}$$

Feistel-MiMC



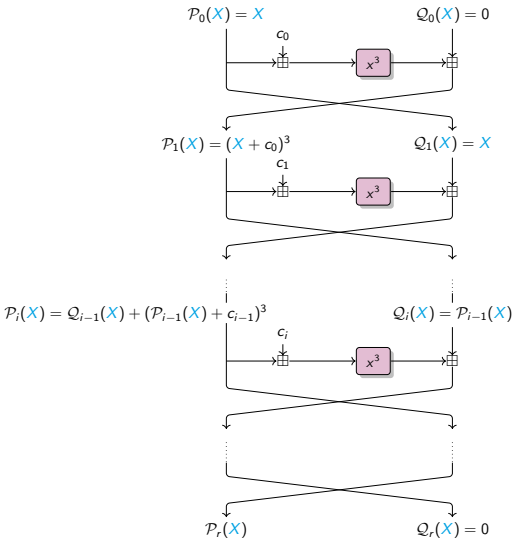
$$\begin{cases} \mathcal{P}_0(X) & = X \\ \mathcal{Q}_0(X) & = 0 \\ \mathcal{P}_1(X) & = (X + c_0)^3 \\ \mathcal{Q}_1(X) & = X \end{cases}$$

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 + $(2r + 1)$ équations

Challenges de Cryptanalyse

Catégorie	Paramètres	Niveau de Sécurité	Gain
~~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*

Catégorie	Paramètres	Niveau de Sécurité	Gain
Easy	$N = 4, m = 3$	25	$2,000
Easy	$N = 6, m = 2$	25	$4,000
Medium	$N = 7, m = 2$	29	$6,000
	$N = 5, m = 3$	30	$12,000
	$N = 8, m = 2$	33	$26,000

(b) *Rescue–Prime*

$12,000

Catégorie	Paramètres	Niveau de Sécurité	Gain
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*

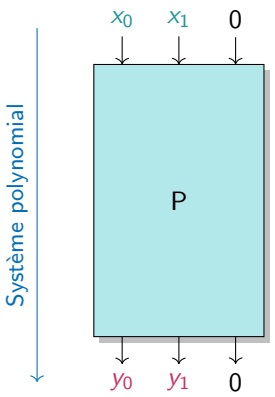
Catégorie	Paramètres	Niveau de Sécurité	Gain
Easy	$p = 281474976710597$	24	$4,000
Medium	$p = 72057594037926839$	28	$6,000
Hard	$p = 18446744073709551557$	32	$12,000

(d) *Reinforced Concrete*

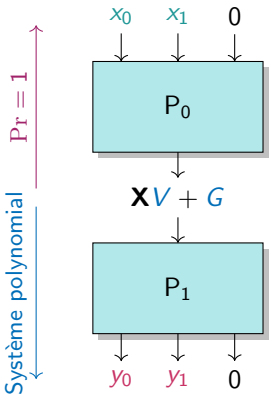
Astuce pour les SPN

Soit $P = P_0 \circ P_1$ une permutation de $\mathbb{F}_p^3$ et supposons

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

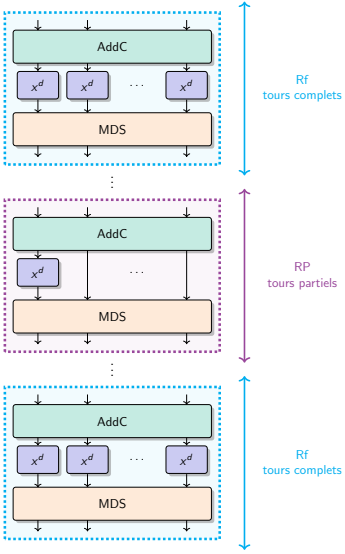


(a) *Système sur R tours.*



(b) *Système sur $R - 2$ tours.*

Poseidon

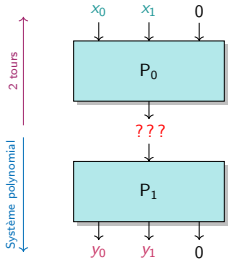


- ⋆ Fonction :

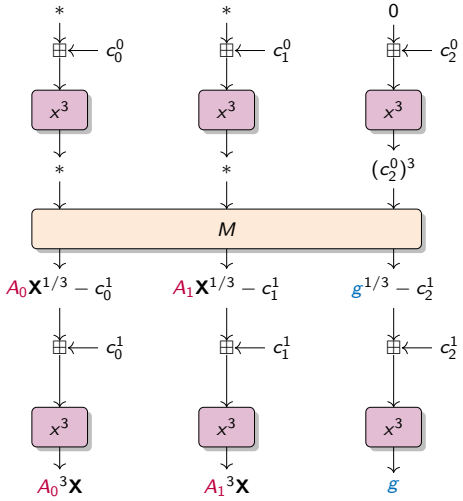
$$x \mapsto x^3$$

- ⋆ Nb tours :

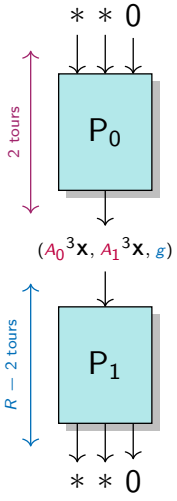
$$R = 2 \times Rf + RP$$
$$= 8 + (\text{de } 3 \text{ à } 24)$$



Astuce pour Poseidon

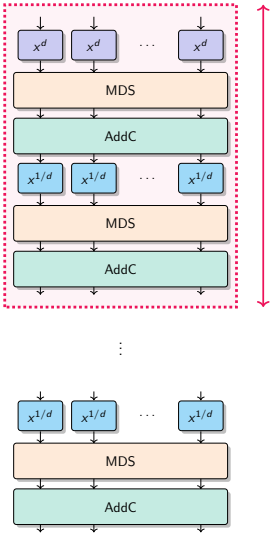


(a) *Deux 1ers tours.*



(b) *Résumé.*

Rescue–Prime



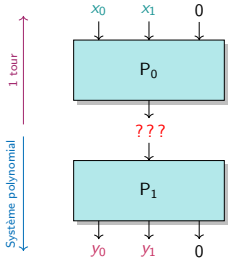
⋆ Fonctions :

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

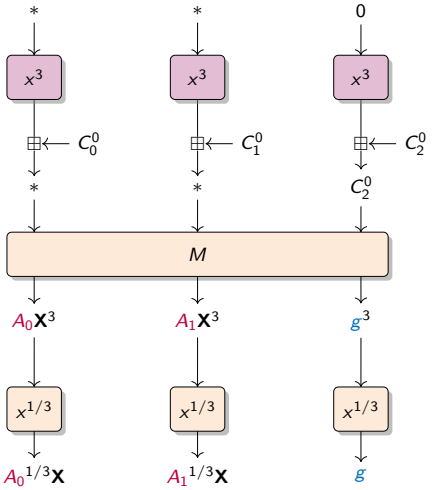
⋆ Nb tours :

$$R = \text{de 4 à 8}$$

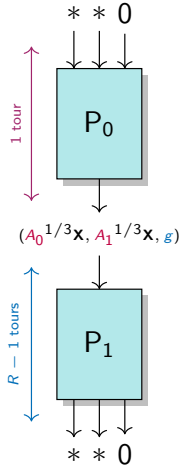
(2 fonctions par tours)



Astuce pour Rescue–Prime



(a) *1er tour.*



(b) *Résumé.*

Challenges de Cryptanalyse

Catégorie	Paramètres	Niveau de Sécurité	Gain
~~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*

Catégorie	Paramètres	Niveau de Sécurité	Gain
~~Easy~~	~~$N = 4, m = 3$~~	~~25~~	~~$2,000~~
Easy	$N = 6, m = 2$	25	$4,000
Medium	$N = 7, m = 2$	29	$6,000
	$N = 5, m = 3$	30	$12,000
	$N = 8, m = 2$	33	$26,000

(b) *Rescue–Prime*

$26,000

Catégorie	Paramètres	Niveau de Sécurité	Gain
~~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

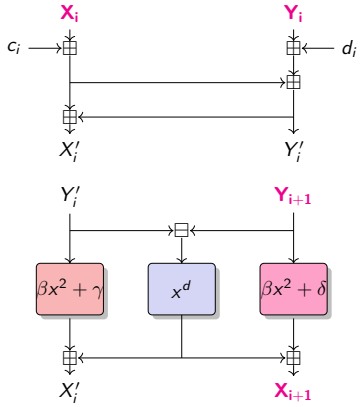
(c) *Poseidon*

Catégorie	Paramètres	Niveau de Sécurité	Gain
Easy	$p = 281474976710597$	24	$4,000
Medium	$p = 72057594037926839$	28	$6,000
Hard	$p = 18446744073709551557$	32	$12,000

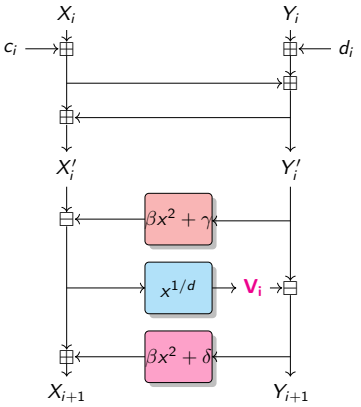
(d) *Reinforced Concrete*

Modélisation d'Anemoi

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

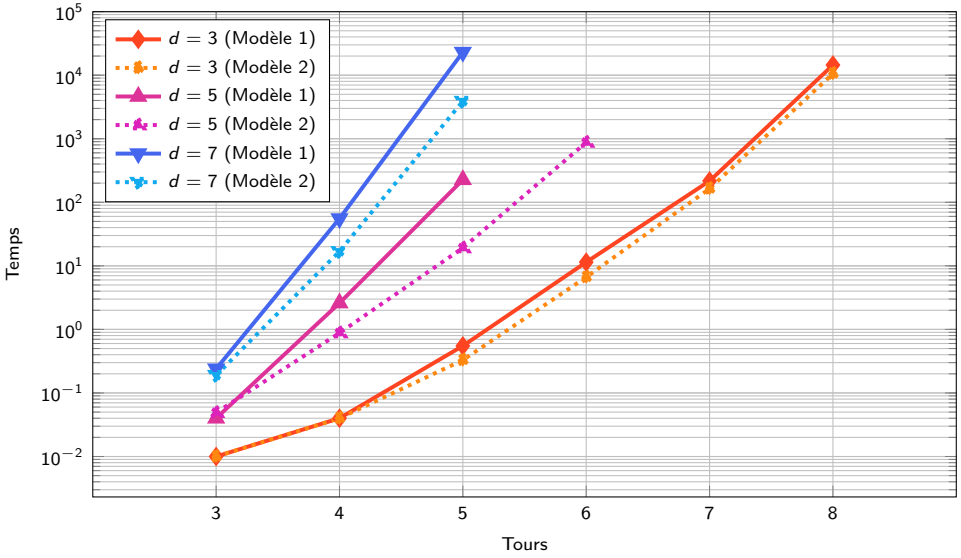


Modèle 1.



Modèle 2.

Importance du modèle



L'attaque FreeLunch

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

Résolution d'un système **multivarié** :

- ⋆ Définir le système
- ⋆ Calculer une BG avec l'ordre grevlex (algorithme **F5**)
- ⋆ Convertir en une BG avec l'ordre lex (algorithme **FGLM**)
- ⋆ Trouver les racines dans $\mathbb{F}_q^n$ des polynomes de la BG en résolvant un système univarié.

L'attaque FreeLunch

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

Résolution d'un système **multivarié** :

- ⋆ Définir le système
- ⋆ Calculer une BG avec l'ordre grevlex (algorithme **F5**) $\rightsquigarrow$ **peut être contournée**
- ⋆ Convertir en une BG avec l'ordre lex (algorithme **FGLM**)
- ⋆ Trouver les racines dans $\mathbb{F}_q^n$ des polynomes de la BG en résolvant un système univarié.

De nouveaux Challenges !

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

(Novembre 2024)

De nouveaux gagnants !

- 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 ! !

- Selon l'ordre lexicographique, $x_1x_2 < x_2x_3$? $x_3 > x_1^3$? $x_1 > x_2^3$?
- Selon l'ordre lexicographique inverse gradué, $x_1x_2x_3 > x_4x_5$?
- Pourrait-on utiliser l'astuce des SPN sur Reinforced Concrete ?
- Est-il pertinent d'utiliser l'attaque FreeLunch pour Feistel-MiMC ?

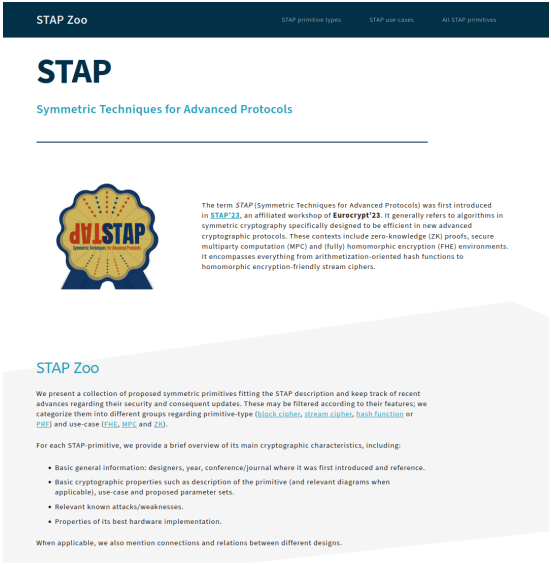
A Retenir

Comment prévenir les attaques algébriques ?

- ⋆ Essayer le plus de modèles possible
- ⋆ Préférer les systèmes univariés que les systèmes multivariés
- ⋆ Être vigilant face aux astuces permettant de contourner les tours

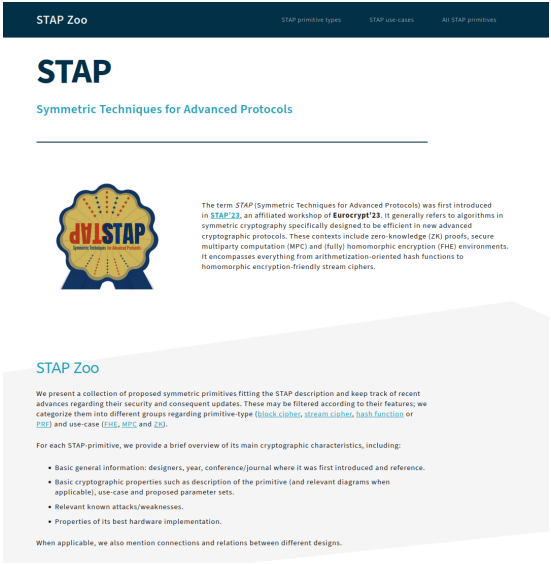
Les AOPs : un nouveau marché lucratif ?

STAP Zoo



stap-zoo.com

STAP Zoo



stap-zoo.com

Merci !