



Simula
UiB

Algebraic Attacks on ZK-friendly Hash Functions and the Impact of CICO-Variants

Based on joint work with A. Bariant, A. Boeuf, A. Lemoine, I. Manterola Ayala, L. Perrin and H. Raddum. Presented at Crypto 2024.

Morten Øy garden

Simula UiB, Bergen

End-of-Escada, Nijmegen, August 2024

The Setting



Symmetric Techniques for Advanced Primitives (STAP)

<https://stap-zoo.com/>

Recent years have seen many new ciphers and hash functions that aim to be efficient in MPC/FHE/ZK-settings.

They typically work over large finite fields and have "simple" algebraic structure.

⇒ Potentially vulnerable to algebraic attacks.

A large family is that of ZK-friendly hash functions, instantiated by the sponge construction.

The ideas in this talk will be particularly relevant for the ZK-friendly hash functions: **Griffin**, Arion, Anemoi and XHash.



The Sponge Function

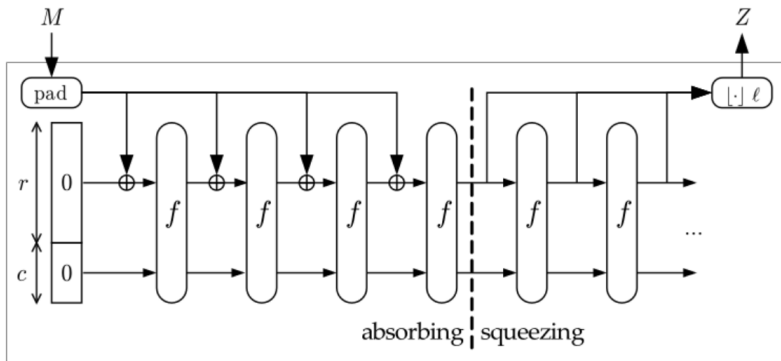


Figure: Source: Keccak.team



The CICO Problem

The CICO Problem of size c for permutation P :

$$P(x_0, \dots, x_r, \underbrace{0, \dots, 0}_{c \text{ elements}}) = (*', \dots, *', \underbrace{0, \dots, 0}_{c \text{ elements}})$$

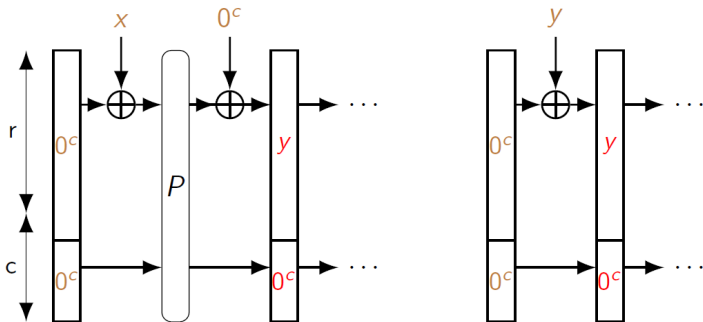
For this talk we focus on $c = 1$:

$$P(x_0, *, \dots, *, 0) = (*', \dots, *', 0).$$



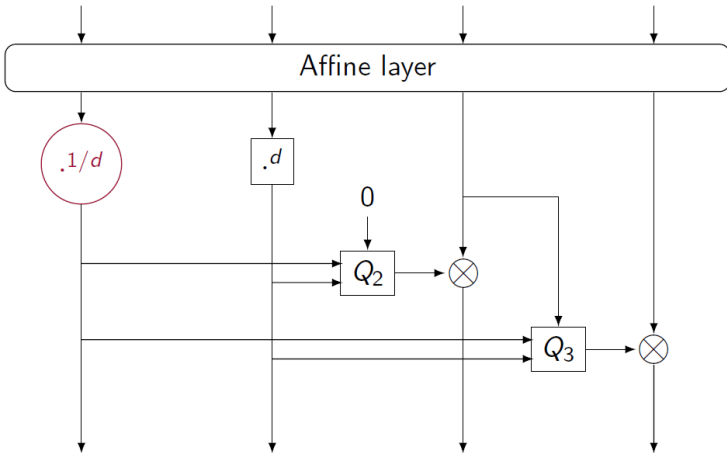
Example: Collision From the CICO-Problem

Suppose we know x such that $P(x \parallel 0^c) = (y \parallel 0^c)$.



Griffin- π - Round Function (4 branches)

d small (typically $d = 3$).



Polynomial System Solving (In This Setting)



Polynomial System Solving

Consider a set of m polynomials $f_1, \dots, f_m \in \mathbb{F}[x_1, \dots, x_n]$. We want to solve the following system of equations:

$$f_1(x_1, \dots, x_n) = 0,$$

$$\vdots$$

$$f_m(x_1, \dots, x_n) = 0.$$



Polynomial System Solving

Consider a set of m polynomials $f_1, \dots, f_m \in \mathbb{F}[x_1, \dots, x_n]$. We want to solve the following system of equations:

$$\begin{aligned} f_1(x_1, \dots, x_n) &= 0, \\ &\vdots \\ f_m(x_1, \dots, x_n) &= 0. \end{aligned}$$

NP-complete when $n = m$.

Hard to find good complexity estimates for specific systems that arise in cryptography.



Solving Strategies

(At a very high level)

Easy to solve: linear systems; **univariate polynomials**; sparse + small field.



Solving Strategies

(At a very high level)

Easy to solve: linear systems; **univariate polynomials**; sparse + small field.

The ideal of generated by the set of polynomials

$f_1, \dots, f_m \in R = \mathbb{F}[x_1, \dots, x_n]$ is

$$I := \langle f_1, \dots, f_m \rangle := \left\{ \sum_{i=1}^m f_i g_i \mid g_i \in R \right\}.$$



Solving Strategies

(At a very high level)

Easy to solve: linear systems; **univariate polynomials**; sparse + small field.

The ideal of generated by the set of polynomials

$f_1, \dots, f_m \in R = \mathbb{F}[x_1, \dots, x_n]$ is

$$I := \langle f_1, \dots, f_m \rangle := \left\{ \sum_{i=1}^m f_i g_i \mid g_i \in R \right\}.$$

Many solving strategies involves searching for polynomials in I that are easy to solve.



Gröbner Basis (GB)

A Gröbner basis is a particular generating set of I with nice computational properties.

Facilitates computations in I and R/I . Is the "go-to" method for finding solutions to the polynomials generating I .



Gröbner Basis (GB)

A Gröbner basis is a particular generating set of I with nice computational properties.

Facilitates computations in I and R/I . Is the "go-to" method for finding solutions to the polynomials generating I .

Fix a total ordering $\prec$ of the monomials (monomial order). $\text{LT}(f)$ is the largest term in f w.r.t. $\prec$.



Gröbner Basis (GB)

A Gröbner basis is a particular generating set of I with nice computational properties.

Facilitates computations in I and R/I . Is the "go-to" method for finding solutions to the polynomials generating I .

Fix a total ordering $\prec$ of the monomials (monomial order). $\text{LT}(f)$ is the largest term in f w.r.t. $\prec$.

Definition (Gröbner Basis)

A set of polynomials $G = \{g_1, \dots, g_r\}$ is a GB of I (w.r.t. $\prec$) if

- i) $\langle G \rangle = I$; and
- ii) for all $f \in I$, there is some j such that $\text{LT}(g_j) \mid \text{LT}(f)$.



What We Will Use for This Talk

Proposition

A set of polynomials $G = \{g_1, \dots, g_\ell\}$ is a Gröbner basis for $I = \langle G \rangle$ if

$$\text{LM}_{<}(g_1), \dots, \text{LM}_{<}(g_\ell)$$

are pairwise **coprime**.

(E.g. x^2 and y are coprime; x^2 and xy are not.)



The Remaining Solving Process

(In our setting)

Once a Gröbner basis has been found there are two remaining steps:

- 1) Use the GB to search through I for a univariate polynomial (monomial order change).
- 2) Find the roots to this univariate polynomial.



Gröbner Bases From Griffin



Monomial Order and Weight

All monomial orders can be thought of through weight vectors $(\text{wt}(x_0), \dots, \text{wt}(x_{n-1}))$, where monomials are compared by the values

$$x_0^{a_0} \cdots x_{n-1}^{a_{n-1}} \longrightarrow a_0 \text{wt}(x_0) + \dots + a_{n-1} \text{wt}(x_{n-1}).$$



Monomial Order and Weight

All monomial orders can be thought of through weight vectors $(\text{wt}(x_0), \dots, \text{wt}(x_{n-1}))$, where monomials are compared by the values

$$x_0^{a_0} \cdots x_{n-1}^{a_{n-1}} \longrightarrow a_0 \text{wt}(x_0) + \dots + a_{n-1} \text{wt}(x_{n-1}).$$

Example: Grading (comparison by degree) has weight vector

$$\text{wt}(x_0) = \dots = \text{wt}(x_{n-1}) = 1.$$



Gröbner Basis - Example

Proposition

If $\text{LM}_{<}(g_1), \dots, \text{LM}_{<}(g_\ell)$ are pairwise **coprime**, then G is a Gröbner basis.

$$f_0 = y^3 + x, \quad f_1 = y^3 + x^2.$$



Gröbner Basis - Example

Proposition

If $\text{LM}_{<}(g_1), \dots, \text{LM}_{<}(g_\ell)$ are pairwise **coprime**, then G is a Gröbner basis.

$$f_0 = y^3 + x, \quad f_1 = y^3 + x^2.$$

f_0, f_1 is not a Gröbner basis for the **lex** or **deglex** order.



Gröbner Basis - Example

Proposition

If $\text{LM}_{<}(g_1), \dots, \text{LM}_{<}(g_\ell)$ are pairwise **coprime**, then G is a Gröbner basis.

$$f_0 = y^3 + x, \quad f_1 = y^3 + x^2.$$

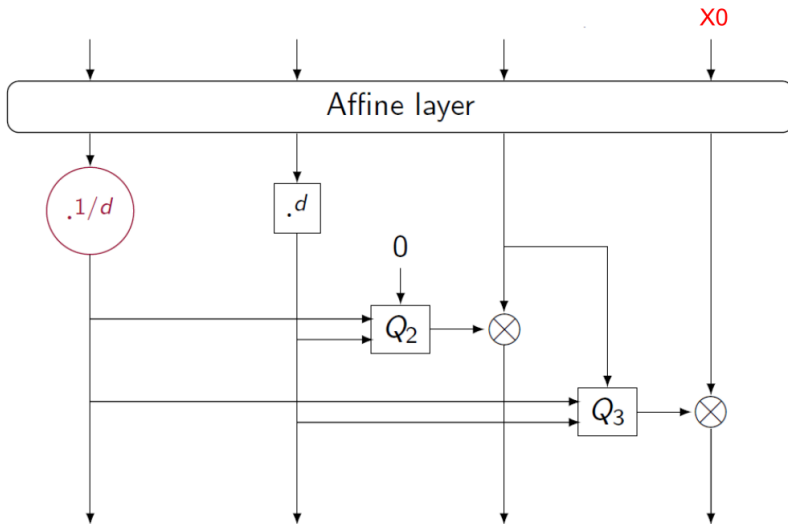
f_0, f_1 is not a Gröbner basis for the **lex** or **deglex** order.

Choose $\text{wt}(x) = 2$ and $\text{wt}(y) = 1$. Then $\text{LM}(f_0) = y^3$ and $\text{LM}(f_1) = x^2$ are **coprime**.

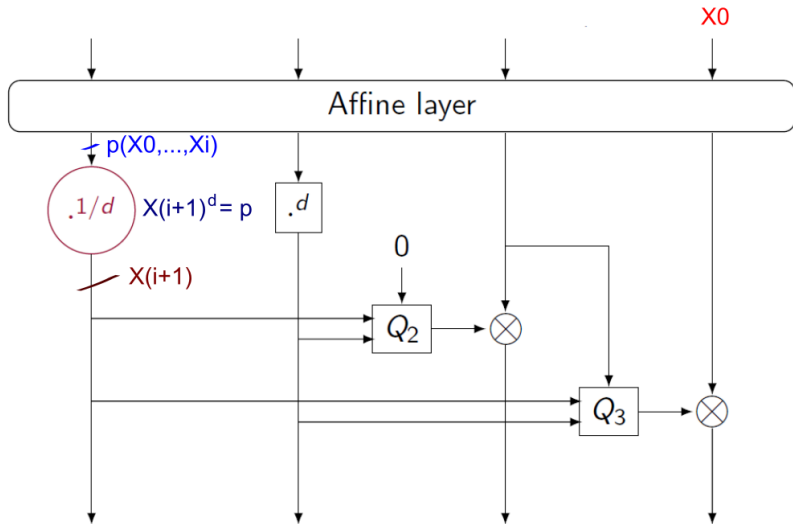
$\implies f_0, f_1$ is a Gröbner basis under this monomial order.



Griffin- π - Round Function (4 branches)



Griffin- π - Round Function (4 branches)



Griffin- π - Model

$d = 3$, two rounds

$$x_1^3 = ax_0 + b$$

$$x_2^3 = cx_0^7 + \dots$$

$$x_0^{49} + dx_0^{46} + ex_0^{45} + \dots = 0$$



Griffin- π - Model

$d = 3$, two rounds

$$x_1^3 = ax_0 + b$$

$$x_2^3 = cx_0^7 + \dots$$

$$x_0^{49} + dx_0^{46} + ex_0^{45} + \dots = 0$$

In **grevlex** (degree-first), the leading monomials are x_1^3 , x_0^7 and x_0^{49} .
The Proposition does not apply.



Griffin- π - Model

$d = 3$, two rounds

$$x_1^3 = ax_0 + b$$

$$x_2^3 = cx_0^7 + \dots$$

$$x_0^{49} + dx_0^{46} + ex_0^{45} + \dots = 0$$

In **grevlex** (degree-first), the leading monomials are x_1^3 , x_0^7 and x_0^{49} .
The Proposition does not apply.

In an order with $\text{wt}(x_0) = \text{wt}(x_1) = 1$ and $\text{wt}(x_2) = 3$, the leading monomials are x_1^3 , x_2^3 and x_0^{49} .

$\implies$ It's a Gröbner basis.

This generalizes for more rounds.



Implications

We can compute Gröbner bases for Griffin "for free"¹.

¹Generating the polynomials gets difficult in practice for large r . Still $\ll 2^{128}$ operations

²Likely a very loose upper bound - work in progress.



Implications

We can compute Gröbner bases for Griffin "for free"¹.

We can easily compute the **ideal degree**

$$D_I = \dim_{\mathbb{F}} (R/I) .$$

¹Generating the polynomials gets difficult in practice for large r . Still $\ll 2^{128}$ operations

²Likely a very loose upper bound - work in progress.



Implications

We can compute Gröbner bases for Griffin "for free"¹.

We can easily compute the **ideal degree**

$$D_I = \dim_{\mathbb{F}} (R/I) .$$

This upper bounds² the step of finding a univariate polynomial by $\mathcal{O}(D_I^3)$.

The univariate polynomial we find has degree at most D_I .

¹Generating the polynomials gets difficult in practice for large r . Still $\ll 2^{128}$ operations

²Likely a very loose upper bound - work in progress.



Results

The Gröbner basis trick we have seen can be applied to the ZK-friendly hash-functions Griffin, Arion, Anemoi and XHash8.

Along with some extra tricks we can break some variants³ of Griffin and Arion.

The exact impact on Anemoi is unclear.

XHash is looking pretty robust.

³Likely more with the aforementioned work in progress.



Open Problems



Modes of Operation as a Countermeasure

Recall that we have only looked at CICO-problems with a **single** output constraint. Different set-ups with the sponge framework yield different CICO-problems:



Modes of Operation as a Countermeasure

Recall that we have only looked at CICO-problems with a **single** output constraint. Different set-ups with the sponge framework yield different CICO-problems:

- How do we generalize for more output constraints? (E.g., the idea in this talk already fails for x_0, x_1 in; and $f_0 = f_1 = 0$ out).



Modes of Operation as a Countermeasure

Recall that we have only looked at CICO-problems with a **single** output constraint. Different set-ups with the sponge framework yield different CICO-problems:

- How do we generalize for more output constraints? (E.g., the idea in this talk already fails for x_0, x_1 in; and $f_0 = f_1 = 0$ out).
- What if we have more constraints than variables in? (Also relevant if we can guess some inputs).



Modes of Operation as a Countermeasure

Recall that we have only looked at CICO-problems with a **single** output constraint. Different set-ups with the sponge framework yield different CICO-problems:

- How do we generalize for more output constraints? (E.g., the idea in this talk already fails for x_0, x_1 in; and $f_0 = f_1 = 0$ out).
- What if we have more constraints than variables in? (Also relevant if we can guess some inputs).
→ (In this case the Gröbner basis is likely the same as the solution, so in this case it **should** be hard to find.)



Other Questions

- What if we have a **full** layer of inversions? (Relevant for Rescue, XHash12).



Other Questions

- What if we have a **full** layer of inversions? (Relevant for Rescue, XHash12).
- Can we generalize this to SPNs with other operations than inversions? (Could be relevant for a lot of MPC/ZK/FHE-friendly constructions).



Other Questions

- What if we have a **full** layer of inversions? (Relevant for Rescue, XHash12).
- Can we generalize this to SPNs with other operations than inversions? (Could be relevant for a lot of MPC/ZK/FHE-friendly constructions).
- All algebraic attacks we have talked about are **preimage** attacks w.r.t. the permutation. Can we do **algebraic collision** attacks in a meaningful way?



Some Thoughts in These Directions

Probably no more "low-hanging fruit" with "free" Gröbner bases.



Some Thoughts in These Directions

Probably no more "low-hanging fruit" with "free" Gröbner bases.

Maybe possible to split into subsystems that can be understood with these techniques?



Some Thoughts in These Directions

Probably no more "low-hanging fruit" with "free" Gröbner bases.

Maybe possible to split into subsystems that can be understood with these techniques?

Opens the door for more "holistic" approaches where:

- Modeling of the polynomial system.
- Choice of monomial order.
- Gröbner basis computation.

should all be taken into account from the beginning.



ArcticCrypt 2025

<https://simula-uib.com/arcticcrypt2025/>

Longyearbyen, Svalbard, July 6th-11th 2025.
Submission Deadline: September 13th 2024.

