The Algebraic Freelunch: Efficient Gröbner Basis Attacks Against Arithmetization-Oriented Primitives

Augustin Bariant[1,2], Aurélien Boeuf[2], Axel Lemoine[2,4], Irati Manterola Ayala[3], Morten Øygarden[3], Léo Perrin[2], and Håvard Raddum[3]

[1]ANSSI, Paris, France [2]INRIA, Paris, France
[3]Simula UiB, Bergen, Norway [4]DGA, France

Crypto 2024, Santa Barbara

Anemoi
Crypto23

Griffin
Crypto23

ArionHash
arXiv

Anemoi
Crypto23

~~**Griffin**~~
~~Crypto23~~

Full-round break
of some instances

ArionHash
arXiv

Anemoi
Crypto23

Griffin
Crypto23

Full-round break
of some instances

ArionHash
arXiv

Full-round break
of some instances

Anemoi
Crypto23

Maybe full-round break?

Griffin
Crypto23

Full-round break
of some instances

ArionHash
arXiv

Full-round break
of some instances

Anemoi
Crypto23

Maybe full-round break?

Griffin
Crypto23

Full-round break
of some instances

ArionHash
arXiv

Full-round break
of some instances

Three main improvements on previous algebraic cryptanalysis:

1. Free Gröbner basis for some monomial orders.
2. Better approach to solving the system than generic FGLM variants.
3. Bypassing the first few rounds of Griffin and Arion with symmetric-like techniques.

Arithmetization-Oriented Primitives

Freelunch Systems for Free Gröbner Bases

Solving the System given a Gröbner Basis

Arithmetization-Oriented Primitives

AOPs: dedicated primitives for advanced protocols (ZK proofs, MPC, FHE...)

Classic	**Arithmetization-Oriented**
Binary operations	Arithmetic operations
Algebraically complex (for cheap)	Algebraically simple
Small field ($\mathbb{F}_{2^8}$)	Large field ($\mathbb{F}_q$, $q > 2^{64}$)
e.g. AES, SHA-3	e.g. Griffin, Anemoi

Quick Overview of Griffin, Arion, Anemoi

Our targets:

Anemoi	**Griffin**	**ArionHash**
Crypto23	Crypto23	arXiv

- Griffin, ArionHash and AnemoiSponge = Arithmetization-Oriented families of hash functions.
- Instantiated with the Griffin-π, Arion-π and Anemoi families of permutations.

Quick Overview of Griffin, Arion, Anemoi

Our targets:

Anemoi	**Griffin**	**ArionHash**
Crypto23	Crypto23	arXiv

- Griffin, ArionHash and AnemoiSponge = Arithmetization-Oriented families of hash functions.
- Instantiated with the Griffin-π, Arion-π and Anemoi families of permutations.
- **Many** instances are defined: variable $\mathbb{F}_p$, number of branches, exponents for monomial permutations...

$\implies$ We attack some instances better than others.

CICO Problem

CICO Problem of size c (capacity of the sponge) for permutation P:

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

CICO Problem

CICO Problem of size c (capacity of the sponge) for permutation P:

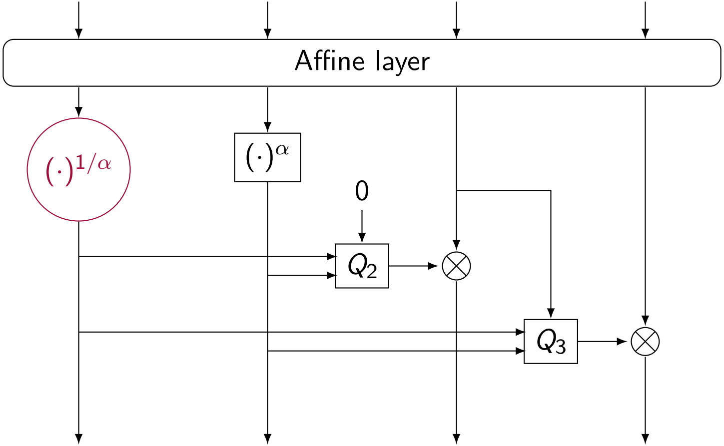
$$P(*, \dots, *, \underbrace{0, \dots, 0}_{c \text{ elements}}) = (*', \dots, *', \underbrace{0, \dots, 0}_{c \text{ elements}})$$

Solving CICO of size c gives collisions to the hash function.

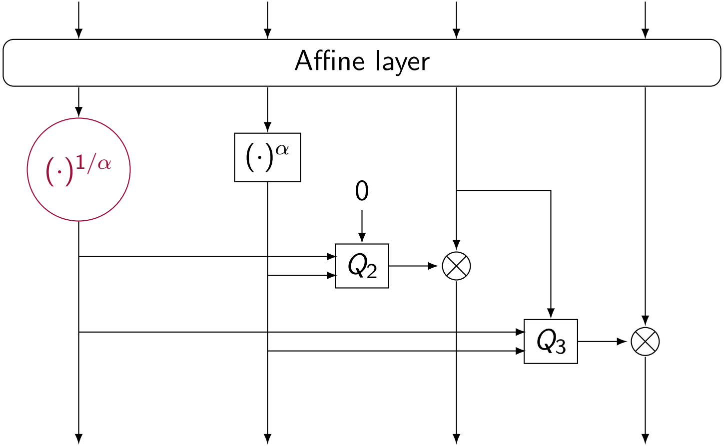
⇒ Multivariate attack: solve CICO faster than brute-force attacks using a model of P.
⇒ We focus on $c = 1$.

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

Griffin-π - Round Function (4 branches)



Griffin-π - Round Function (4 branches)



$(\cdot)^{1/\alpha}$ is the only high-degree operation $\implies$ add one variable per $(\cdot)^{1/\alpha}$.

Griffin-π - Model

- **CICO problem**: $\mathcal{G}_\pi(\cdots||0) = (\cdots||0)$.
  $\Longrightarrow$ One variable x_0 in the input. One equation for the output (last branch at 0).
- N_{rounds} equations of the form $x_i^\alpha = P_i(x_0, x_1, \ldots x_{i-1})$ $\quad \left((\cdot)^{1/\alpha} \text{ S-boxes}\right)$.

Griffin-π - Model

- **CICO problem**: $\mathcal{G}_\pi(\cdots||0) = (\cdots||0)$.
  $\implies$ One variable x_0 in the input. One equation for the output (last branch at 0).
- N_{rounds} equations of the form $x_i^\alpha = P_i(x_0, x_1, \ldots x_{i-1})$ $\left((\cdot)^{1/\alpha} \text{ S-boxes}\right)$.

Example ($\alpha = 3$, one round)

$$x_1^3 = a x_0 + b$$
$$x_0^7 + c x_0^4 x_1 + d x_0 x_1^2 + \cdots = 0$$

Generic System Solving

$$\begin{cases} p_1(x_1, \ldots, x_N) = 0 \\ \vdots \\ p_{k-1}(x_1, \ldots, x_N) = 0 \\ p_k(x_1, \ldots, x_N) = 0 \end{cases}$$

1. Define system

Generic System Solving

$$\begin{cases} p_1(x_1, \ldots, x_N) = 0 \\ \vdots \\ p_{k-1}(x_1, \ldots, x_N) = 0 \\ p_k(x_1, \ldots, x_N) = 0 \end{cases} \qquad \begin{cases} g_1(x_1, \ldots, x_N) = 0 \\ \vdots \\ g_{\kappa-1}(x_1, \ldots, x_N) = 0 \\ g_\kappa(x_1, \ldots, x_N) = 0 \end{cases}$$

1. Define system
2. Find a **Gröbner Basis**

Generic System Solving

$$\begin{cases} p_1(x_1, \ldots, x_N) = 0 \\ \vdots \\ p_{k-1}(x_1, \ldots, x_N) = 0 \\ p_k(x_1, \ldots, x_N) = 0 \end{cases} \quad \begin{cases} g_1(x_1, \ldots, x_N) = 0 \\ \vdots \\ g_{\kappa-1}(x_1, \ldots, x_N) = 0 \\ g_\kappa(x_1, \ldots, x_N) = 0 \end{cases} \quad \begin{cases} g_1^*(x_1, \ldots, x_N) = 0 \\ \vdots \\ g_{N-1}^*(x_{N-1}, x_N) = 0 \\ g_N^*(x_N) = 0 \end{cases}$$

1. Define system
2. Find a **Gröbner Basis**
3. Change order to **lex** (FGLM)

Generic System Solving

$$\begin{cases} p_1(x_1, \ldots, x_N) = 0 \\ \vdots \\ p_{k-1}(x_1, \ldots, x_N) = 0 \\ p_k(x_1, \ldots, x_N) = 0 \end{cases} \quad \begin{cases} g_1(x_1, \ldots, x_N) = 0 \\ \vdots \\ g_{\kappa-1}(x_1, \ldots, x_N) = 0 \\ g_\kappa(x_1, \ldots, x_N) = 0 \end{cases} \quad \begin{cases} g_1^*(x_1, \ldots, x_N) = 0 \\ \vdots \\ g_{N-1}^*(x_{N-1}, x_N) = 0 \\ g_N^*(x_N) = 0 \end{cases}$$

1. Define system
2. Find a **Gröbner Basis**
3. Change order to **lex** (FGLM)
4. Find the roots in $\mathbb{F}_q$ of g_N^* with univariate methods, etc.

Step 2 and **Step 3** are the most costly. Designers of Anemoi and Griffin base their security on the hardness of **Step 2**.

Generic System Solving

$$\begin{cases} p_1(x_1,\ldots,x_N) = 0 \\ \vdots \\ p_{k-1}(x_1,\ldots,x_N) = 0 \\ p_k(x_1,\ldots,x_N) = 0 \end{cases} \quad \begin{cases} g_1(x_1,\ldots,x_N) = 0 \\ \vdots \\ g_{\kappa-1}(x_1,\ldots,x_N) = 0 \\ g_\kappa(x_1,\ldots,x_N) = 0 \end{cases} \quad \begin{cases} g_1^*(x_1,\ldots,x_N) = 0 \\ \vdots \\ g_{N-1}^*(x_{N-1},x_N) = 0 \\ g_N^*(x_N) = 0 \end{cases}$$

1. Define system
2. ~~Find a **Gröbner Basis**~~
3. Change order to **lex** (FGLM)
4. Find the roots in $\mathbb{F}_q$ of g_N^* with univariate methods, etc.

Step 2 and **Step 3** are the most costly. Designers of Anemoi and Griffin base their security on the hardness of **Step 2**.

But we can skip it!

Arithmetization-Oriented Primitives

Freelunch Systems for Free Gröbner Bases

Solving the System given a Gröbner Basis

Gröbner Basis - Examples

Useful Proposition

*If $LM_<(p_1), \ldots, LM_<(p_k)$ are pairwise **coprime** (e.g. x^2 and y), then G is a Gröbner basis.*

Gröbner Basis - Examples

Useful Proposition

*If $LM_<(p_1), \ldots, LM_<(p_k)$ are pairwise **coprime** (e.g. x^2 and y), then G is a Gröbner basis.*

In $\mathbb{F}[x, y]$:

- $\{x^2 - 1, y^2 - x\}$ is a Gröbner basis for the **grevlex** order (leading monomials are x^2 and y^2).

Gröbner Basis - Examples

Useful Proposition

*If $LM_<(p_1), \ldots, LM_<(p_k)$ are pairwise **coprime** (e.g. x^2 and y), then G is a Gröbner basis.*

In $\mathbb{F}[x, y]$:

- $\{x^2 - 1, y^2 - x\}$ is a Gröbner basis for the **grevlex** order (leading monomials are x^2 and y^2).
- $\{x^2 - 1, y^2 - x\}$ is not a Gröbner basis for the **lex** order with $x > y$ (leading monomials are x^2 and x).

Gröbner Basis - Examples

Useful Proposition

If $LM_<(p_1), \ldots, LM_<(p_k)$ are pairwise ***coprime*** *(e.g. x^2 and y), then G is a Gröbner basis.*

In $\mathbb{F}[x, y]$:

- $\{x^2 - 1, y^2 - x\}$ is a Gröbner basis for the **grevlex** order (leading monomials are x^2 and y^2).
- $\{x^2 - 1, y^2 - x\}$ is not a Gröbner basis for the **lex** order with $x > y$ (leading monomials are x^2 and x).
- $\{y^3 + x, y^3 + x^2\}$ is not a Gröbner basis for any **lex** or **deglex** order.

Gröbner Basis - Examples

Useful Proposition

*If $LM_<(p_1), \ldots, LM_<(p_k)$ are pairwise **coprime** (e.g. x^2 and y), then G is a Gröbner basis.*

In $\mathbb{F}[x, y]$:

- $\{x^2 - 1, y^2 - x\}$ is a Gröbner basis for the **grevlex** order (leading monomials are x^2 and y^2).
- $\{x^2 - 1, y^2 - x\}$ is not a Gröbner basis for the **lex** order with $x > y$ (leading monomials are x^2 and x).
- $\{y^3 + x, y^3 + x^2\}$ is not a Gröbner basis for any **lex** or **deglex** order.
- $\{y^3 + x, y^3 + x^2\}$ is a Gröbner basis for **weighted degree** orders with $\mathbf{wt}(x) = 2$ and $\mathbf{wt}(y) = 1$, as then $\mathrm{LM}(y^3 + x) = y^3$ and $\mathrm{LM}(y^3 + x^2) = x^2$ are **coprime**.

Griffin-π - Model

Example ($\alpha = 3$, two rounds)

$$x_1^3 = a x_0 + b$$
$$x_2^3 = c x_0^7 + \cdots$$
$$x_0^{49} + d x_0^{46} + e x_0^{45} + \cdots = 0$$

Griffin-π - Model

Example ($\alpha = 3$, two rounds)

$$x_1^3 = a x_0 + b$$
$$x_2^3 = c x_0^7 + \cdots$$
$$x_0^{49} + d x_0^{46} + e x_0^{45} + \cdots = 0$$

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

Griffin-π - Model

Example ($\alpha = 3$, two rounds)

$$x_1^3 = a x_0 + b$$
$$x_2^3 = c x_0^7 + \cdots$$
$$x_0^{49} + d x_0^{46} + e x_0^{45} + \cdots = 0$$

$\Longrightarrow$ *In **grevlex** (degree-first), the leading monomials are x_1^3, x_0^7 and x_0^{49}. Proposition does not apply.*
$\Longrightarrow$ *In **weighted grevlex**, with $\mathbf{wt}(x_0) = \mathbf{wt}(x_1) = 1$ and $\mathbf{wt}(x_2) = 3$, the leading monomials are x_1^3, x_2^3 and x_0^{49}.*

Griffin-π - Model

Example ($\alpha = 3$, two rounds)

$$x_1^3 = a x_0 + b$$
$$x_2^3 = c x_0^7 + \cdots$$
$$x_0^{49} + d x_0^{46} + e x_0^{45} + \cdots = 0$$

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

$\Longrightarrow$ *In **weighted grevlex**, with $\mathbf{wt}(x_0) = \mathbf{wt}(x_1) = 1$ and $\mathbf{wt}(x_2) = 3$, the leading monomials are x_1^3, x_2^3 and x_0^{49}.*

$\Longrightarrow$ ***It's a Gröbner basis**! (coprime leading monomials)*

$\Longrightarrow$ *This generalizes for more rounds.*

Arithmetization-Oriented Primitives

Freelunch Systems for Free Gröbner Bases

Solving the System given a Gröbner Basis

FGLM in a Nutshell

- Given a Gröbner basis G_1 for some ordering $<_1$, and an ordering $<_2$, FGLM computes a Gröbner basis G_2 for $<_2$ in $O(n_{var} D_I^3)$.
- D_I the number of **solutions of the system** in the algebraic closure (in our case the product of the degrees of the leading monomials of the GB).

FGLM in a Nutshell

- Given a Gröbner basis G_1 for some ordering $<_1$, and an ordering $<_2$, FGLM computes a Gröbner basis G_2 for $<_2$ in $O(n_{var} D_I^3)$.
- D_I the number of **solutions of the system** in the algebraic closure (in our case the product of the degrees of the leading monomials of the GB).
- **Order change is interesting** because a GB in **lex** order **must have** a univariate polynomial in the smallest variable, which we can solve. (This corresponds to eliminating the other variables.)

FGLM in a Nutshell

- Given a Gröbner basis G_1 for some ordering $<_1$, and an ordering $<_2$, FGLM computes a Gröbner basis G_2 for $<_2$ in $O(n_{var} D_I^3)$.
- D_I the number of **solutions of the system** in the algebraic closure (in our case the product of the degrees of the leading monomials of the GB).
- **Order change is interesting** because a GB in **lex** order **must have** a univariate polynomial in the smallest variable, which we can solve. (This corresponds to eliminating the other variables.)
- Free Gröbner basis, FGLM and symmetric techniques to bypass the first rounds is already enough to break some instances of Griffin and Arion.

Faster Change of Order Strategy

- Idea from a 2022 paper by Jérémy Berthomieu, Vincent Neiger, Mohab Safey El Din.
- Strategy: for the smallest variable x, compute the characteristic polynomial χ of the linear operation $P \mapsto \mathrm{Red}_{<}(x \cdot P, G)$.
- $\chi(x) = 0$.
- **Issue:** our systems **do not** verify an important property of the original paper.

Computing the Characteristic Polynomial

Step 1: Compute the matrix T of the linear operation in $\mathbb{F}[x_0, x_1, \ldots, x_N]$ that maps P to $\mathrm{Red}_<(x_0 \cdot P, G)$. **We only have very loose complexity bounds for this step**.

Step 2: Compute $\det(XI - T)$.
$\implies$ T is sparse. With block matrix reasoning, this reduces to computing the determinant of a polynomial matrix of size $D_1 = d_1 \cdots d_N$.
$\implies$ In Griffin and Arion, d_0 is by far the highest degree, so this reduces complexity by a lot.
$\implies$ This can be computed with fast linear algebra, in $\mathcal{O}(d_0 \log(d_0)^2 d_1^\omega \cdots d_N^\omega)$.

Our Full Algorithm

1. `sysGen`: Compute the Freelunch system and the order for a free Gröbner basis.
2. `matGen`: Compute the multiplication matrix T of multiplication by x_0.
   $\implies$ **Complexity hard to evaluate**.
3. `polyDet`: Compute the characteristic polynomial χ of T ($\chi(x_0) = 0$).
   $\implies$ **Longest step aside from** `matGen`.
4. `uniSol`: Find roots of χ with Berlekamp-Rabin in $\tilde{\mathcal{O}}(D_I)$.

Conclusion

- These Arithmetization-Oriented hash functions (and similar) should not base their security on the complexity of finding a Gröbner basis (F4/F5).
- Instead, designers can focus on the growth of D_I with the number of rounds (impacts the complexity of solving algorithms).
- Anemoi, Griffin and Arion need to recompute their numbers of rounds in order to be secure.

Conclusion

- These Arithmetization-Oriented hash functions (and similar) should not base their security on the complexity of finding a Gröbner basis (F4/F5).
- Instead, designers can focus on the growth of D_I with the number of rounds (impacts the complexity of solving algorithms).
- Anemoi, Griffin and Arion need to recompute their numbers of rounds in order to be secure.

Ongoing work:

- Better approach for `matGen` (stay tuned).
- Other contexts where we can get free or “cheap” Gröbner bases?
- CICO on more than one branch?

Conclusion

- These Arithmetization-Oriented hash functions (and similar) should not base their security on the complexity of finding a Gröbner basis (F4/F5).
- Instead, designers can focus on the growth of D_I with the number of rounds (impacts the complexity of solving algorithms).
- Anemoi, Griffin and Arion need to recompute their numbers of rounds in order to be secure.

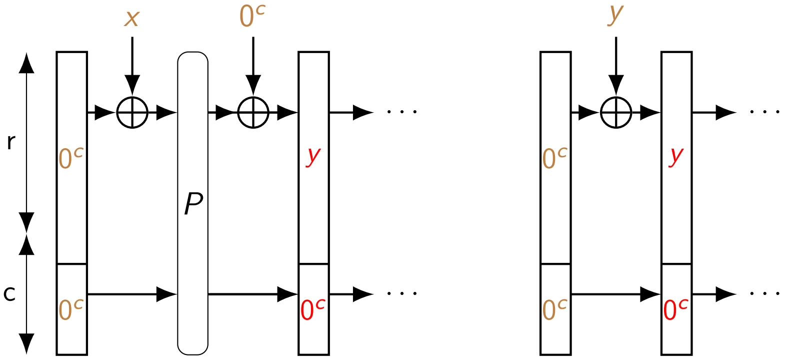
Ongoing work:

- Better approach for `matGen` (stay tuned).
- Other contexts where we can get free or "cheap" Gröbner bases?
- CICO on more than one branch?

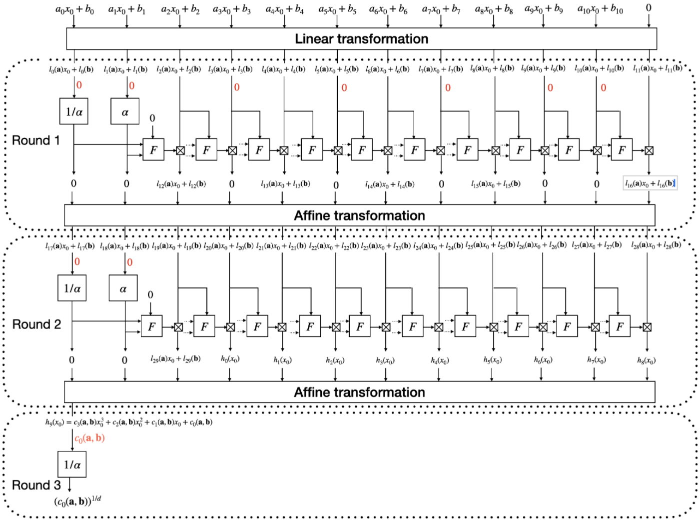
Thank you for your attention!

Collision from the CICO Problem

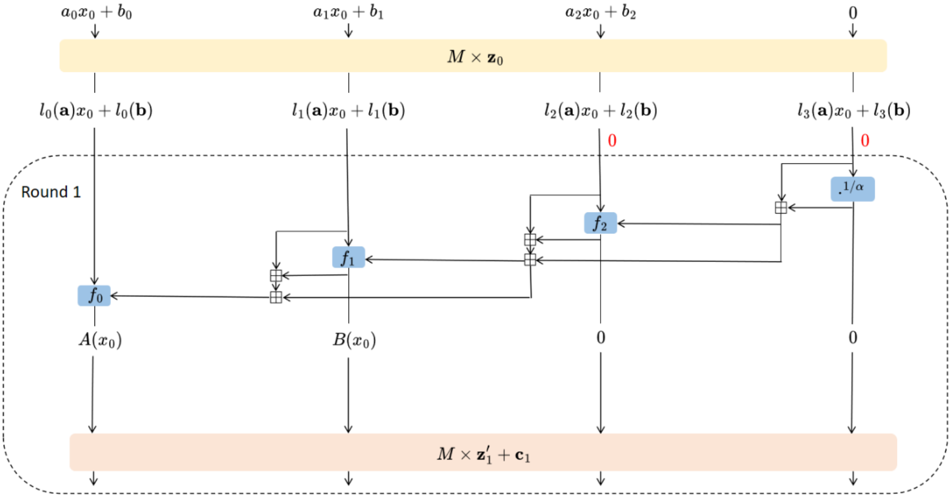
- Suppose you know x such that $P(x \mid\mid 0^c) = (y \mid\mid 0^c)$.



Griffin Trick



Arion Trick



$a_0x_0 + b_0$
$a_1x_0 + b_1$
$a_2x_0 + b_2$
0
$M \times \mathbf{z}_0$
$l_0(\mathbf{a})x_0 + l_0(\mathbf{b})$
$l_1(\mathbf{a})x_0 + l_1(\mathbf{b})$
$l_2(\mathbf{a})x_0 + l_2(\mathbf{b})$
$l_3(\mathbf{a})x_0 + l_3(\mathbf{b})$
0
0
Round 1
$\cdot^{1/\alpha}$
f_2
f_1
f_0
$A(x_0)$
$B(x_0)$
0
0
$M \times \mathbf{z}_1' + \mathbf{c}_1$

What do we want?

Consider a multivariate polynomial ring $\mathbb{F}[x_1, x_2, \ldots, x_N]$.
We want to solve:

$$\begin{cases} p_1(x_1, \ldots, x_N) = 0 \\ p_2(x_1, \ldots, x_N) = 0 \\ \vdots \\ p_k(x_1, \ldots, x_N) = 0 \end{cases}$$

What do we want?

$$\begin{cases} m_{1,1}x_1 + \cdots + m_{1,N}x_N + a_1 = 0 \\ m_{2,1}x_1 + \cdots + m_{2,N}x_N + a_2 = 0 \\ \vdots \\ m_{k,1}x_1 + \cdots + m_{k,N}x_N + a_k = 0 \end{cases}$$

Polynomials of **degree 1**: Linear system $\Rightarrow$ **Linear algebra**.

What do we want?

$$\begin{cases} p_1(x_1) = 0 \\ p_2(x_1) = 0 \\ \vdots \\ p_k(x_1) = 0 \end{cases}$$

One variable: Univariate root finding $\Rightarrow$ **Euclidian division** (for Berlekamp-Rabin algorithm).

What do we want?

$$\begin{cases} p_1(x_1, \dots, x_N) = 0 \\ p_2(x_1, \dots, x_N) = 0 \\ \vdots \\ p_k(x_1, \dots, x_N) = 0 \end{cases}$$

Several variables, high degree: **Linear algebra** + **Euclidian division** (F4/F5, FGLM, Fast-FGLM...).

The Problem with Multivariate

- Euclidian division on **integers**:

$$a = bq + r \ , \ \ 0 \leq r < b.$$

Division of 13 by 3:

$$13 = 4 \times 3 + 1.$$

The Problem with Multivariate

- Euclidian division on **integers**:

$$a = bq + r\ ,\ \ 0 \leq r < b.$$

Division of 13 by 3:

$$13 = 4 \times 3 + 1.$$

- Euclidian division on **univariate polynomials** ($\mathbb{F}[X]$):

$$A = BQ + R\ ,\ \ \deg(R) < \deg(B).$$

Division of $X^3 + X + 1$ by X:

$$X^3 + X + 1 = (X^2 + 1)X + 1.$$

The Problem with Multivariate

- Euclidian division on **multivariate polynomials**:

$$A = BQ + R\ldots \text{ condition on } R?$$

The Problem with Multivariate

- Euclidian division on **multivariate polynomials**:

$$A = BQ + R\ldots \text{ condition on } R?$$

Division of x by $x + y$ in $\mathbb{F}[x, y]$:

$$x = 0 \cdot (x+y) + x$$

or

$$x = 1 \cdot (x+y) - y \text{ ?}$$

The Problem with Multivariate

- Euclidian division on **multivariate polynomials**:

$$A = BQ + R \dots \text{ condition on } R?$$

Division of x by $x+y$ in $\mathbb{F}[x, y]$:

$$x = 0 \cdot (x+y) + x \qquad \Leftarrow \quad x < y$$

or

$$x = 1 \cdot (x+y) - y \qquad \Leftarrow \quad y < x$$

Need to define a **monomial ordering**.
$\implies$ Division steps determined by **leading monomials (LM)**.

Monomial orderings

In $\mathbb{F}[x, y, z]$:

- **LEXicographical:** Compare degree of highest variable, then second-highest, etc.

$$x <_{\text{lex}} y <_{\text{lex}} z \,, \quad x^{1000} \;?\; y$$

Monomial orderings

In $\mathbb{F}[x, y, z]$:

- **LEXicographical:** Compare degree of highest variable, then second-highest, etc.

$$x <_{\text{lex}} y <_{\text{lex}} z \,, \quad x^{1000} <_{\text{lex}} y \,, \quad x^6 yz \;\; ? \;\; y^2 z$$

Monomial orderings

In $\mathbb{F}[x, y, z]$:

- **LEXicographical:** Compare degree of highest variable, then second-highest, etc.

$$x <_{\text{lex}} y <_{\text{lex}} z \;, \quad x^{1000} <_{\text{lex}} y \;, \quad x^6 y z <_{\text{lex}} y^2 z \;.$$

Monomial orderings

In $\mathbb{F}[x, y, z]$:

- **LEXicographical:** Compare degree of highest variable, then second-highest, etc.

$$x <_{\text{lex}} y <_{\text{lex}} z \ , \quad x^{1000} <_{\text{lex}} y \ , \quad x^6 yz <_{\text{lex}} y^2 z \ .$$

- **Graded LEX:** Compare **total degree** first, then switch to lex if equality.

$$x <_{\text{lex}} y <_{\text{lex}} z \ , \quad y \quad ? \quad x^2$$

Monomial orderings

In $\mathbb{F}[x, y, z]$:

- **LEXicographical:** Compare degree of highest variable, then second-highest, etc.

$$x <_{\text{lex}} y <_{\text{lex}} z \,, \quad x^{1000} <_{\text{lex}} y \,, \quad x^6yz <_{\text{lex}} y^2z \,.$$

- **Graded LEX:** Compare **total degree** first, then switch to lex if equality.

$$x <_{\text{lex}} y <_{\text{lex}} z \,, \quad y <_{\text{glex}} x^2 \,, \quad z^2 \quad ? \quad xyz$$

Monomial orderings

In $\mathbb{F}[x, y, z]$:

- **LEXicographical:** Compare degree of highest variable, then second-highest, etc.

$$x <_{\text{lex}} y <_{\text{lex}} z \,, \quad x^{1000} <_{\text{lex}} y \,, \quad x^6 yz <_{\text{lex}} y^2 z \,.$$

- **Graded LEX:** Compare **total degree** first, then switch to lex if equality.

$$x <_{\text{lex}} y <_{\text{lex}} z \,, \quad y <_{\text{glex}} x^2 \,, \quad z^2 <_{\text{glex}} xyz \,, \quad xy <_{\text{glex}} xz <_{\text{glex}} yz \,.$$

Monomial orderings

In $\mathbb{F}[x, y, z]$:

- **LEXicographical:** Compare degree of highest variable, then second-highest, etc.

$$x <_{\text{lex}} y <_{\text{lex}} z \ , \quad x^{1000} <_{\text{lex}} y \ , \quad x^6 yz <_{\text{lex}} y^2 z \ .$$

- **Graded LEX:** Compare **total degree** first, then switch to lex if equality.

$$x <_{\text{lex}} y <_{\text{lex}} z \ , \quad y <_{\text{glex}} x^2 \ , \quad z^2 <_{\text{glex}} xyz \ , \quad xy <_{\text{glex}} xz <_{\text{glex}} yz \ .$$

- **Weighted Graded LEX:** Compare the **weighted sum** of degrees, then lex if equality. Examples for $x <_{\text{lex}} y <_{\text{lex}} z$ and $\mathbf{wt}(x) = 6, \mathbf{wt}(y) = 1, \mathbf{wt}(z) = 2$:

$$x \quad ? \quad yz^2$$

Monomial orderings

In $\mathbb{F}[x, y, z]$:

- **LEXicographical:** Compare degree of highest variable, then second-highest, etc.

$$x <_{\text{lex}} y <_{\text{lex}} z \ , \quad x^{1000} <_{\text{lex}} y \ , \quad x^6 yz <_{\text{lex}} y^2 z \ .$$

- **Graded LEX:** Compare **total degree** first, then switch to lex if equality.

$$x <_{\text{lex}} y <_{\text{lex}} z \ , \quad y <_{\text{glex}} x^2 \ , \quad z^2 <_{\text{glex}} xyz \ , \quad xy <_{\text{glex}} xz <_{\text{glex}} yz \ .$$

- **Weighted Graded LEX:** Compare the **weighted sum** of degrees, then lex if equality. Examples for $x <_{\text{lex}} y <_{\text{lex}} z$ and $\mathbf{wt}(x) = 6, \mathbf{wt}(y) = 1, \mathbf{wt}(z) = 2$:

$$x \ >_{\text{wglex}} yz^2 \text{ because } \mathbf{wt}(x) = 6 \text{ and } \mathbf{wt}(yz) = \mathbf{wt}(y) + 2\mathbf{wt}(z) = 5 \ .$$

$$x^2 \quad ? \quad z^6$$

Monomial orderings

In $\mathbb{F}[x, y, z]$:

- **LEXicographical:** Compare degree of highest variable, then second-highest, etc.

$$x <_{\text{lex}} y <_{\text{lex}} z \,, \quad x^{1000} <_{\text{lex}} y \,, \quad x^6 yz <_{\text{lex}} y^2 z \,.$$

- **Graded LEX:** Compare **total degree** first, then switch to lex if equality.

$$x <_{\text{lex}} y <_{\text{lex}} z \,, \quad y <_{\text{glex}} x^2 \,, \quad z^2 <_{\text{glex}} xyz \,, \quad xy <_{\text{glex}} xz <_{\text{glex}} yz \,.$$

- **Weighted Graded LEX:** Compare the **weighted sum** of degrees, then lex if equality. Examples for $x <_{\text{lex}} y <_{\text{lex}} z$ and $\mathbf{wt}(x) = 6, \mathbf{wt}(y) = 1, \mathbf{wt}(z) = 2$:

$$x \;\; >_{\text{wglex}} yz^2 \text{ because } \mathbf{wt}(x) = 6 \text{ and } \mathbf{wt}(yz) = \mathbf{wt}(y) + 2\mathbf{wt}(z) = 5 \,.$$

$$x^2 \;\; <_{\text{wglex}} z^6 \text{ because } \mathbf{wt}(x^2) = \mathbf{wt}(z^6) = 12 \text{ and } x^2 <_{\text{lex}} z^6 \,.$$

The Problem... Still.

Consider a system $\{p_1, \ldots, p_k\}$.
$\implies$ Division of a polynomial p by $\{p_1, \ldots, p_k\}$ for some ordering: **final remainder can depend on the choice of divisors!**

The Problem... Still.

Consider a system $\{p_1, \ldots, p_k\}$.
$\implies$ Division of a polynomial p by $\{p_1, \ldots, p_k\}$ for some ordering: **final remainder can depend on the choice of divisors!**
Example: in $\mathbb{F}[x, y]$ with **lex** ordering ($x <_{lex} y$), divide y^2 by $\{y^2 - 1, y - x\}$.

y^2	y^2
$\Downarrow$ red. by $y^2 - 1$	$\Downarrow$ red. by $y - x$
1	xy
$\Downarrow$ no further red.	$\Downarrow$ red. by $y - x$
1	x^2

The Problem... Still.

Consider a system $\{p_1, \ldots, p_k\}$.

$\Longrightarrow$ Division of a polynomial p by $\{p_1, \ldots, p_k\}$ for some ordering: **final remainder can depend on the choice of divisors!**

Example: in $\mathbb{F}[x, y]$ with **lex** ordering ($x <_{lex} y$), divide y^2 by $\{y^2 - 1, y - x\}$.

$y^2 \overset{\text{red. by } y^2-1}{\Longrightarrow} 1 \overset{\text{no further red.}}{\Longrightarrow} 1$	$y^2 \overset{\text{red. by } y-x}{\Longrightarrow} xy \overset{\text{red. by } y-x}{\Longrightarrow} x^2$

The solution: Gröbner Bases.

What is a Gröbner Basis?

Let $G = \{p_1, \ldots, p_k\}$ and $<$ a monomial ordering.

Definition

G is a Gröbner basis iff reduction defined by $<$ of any polynomial P does not depend on the order chosen for the reductors.

What is a Gröbner Basis?

Let $G = \{p_1, \ldots, p_k\}$ and $<$ a monomial ordering.

Definition

G is a Gröbner basis iff reduction defined by $<$ of any polynomial P does not depend on the order chosen for the reductors.

Useful Proposition

If $LM_<(p_1), \ldots, LM_<(p_k)$ are pairwise ***coprime*** *(e.g. x^2 and y), then G is a Gröbner basis.*

Gröbner Basis - Examples

In $\mathbb{F}[x, y]$:

- $\{y^2 - 1, y - x\}$ is not a Gröbner basis for **lex** order with $x < y$ (previous example).

Gröbner Basis - Examples

In $\mathbb{F}[x, y]$:

- $\{y^2 - 1, y - x\}$ is not a Gröbner basis for **lex** order with $x < y$ (previous example).
- However, it is a Gröbner basis for **lex** order with $x > y$. Proof: $\mathrm{LM}(y^2 - 1) = y^2$ and $\mathrm{LM}(y - x) = x$ are **coprime**.

Gröbner Basis - Examples

In $\mathbb{F}[x, y]$:

- $\{y^2 - 1, y - x\}$ is not a Gröbner basis for **lex** order with $x < y$ (previous example).
- However, it is a Gröbner basis for **lex** order with $x > y$. Proof: $\text{LM}(y^2 - 1) = y^2$ and $\text{LM}(y - x) = x$ are **coprime**.
- $\{y^3 + x, y^3 + x^2\}$ is not a Gröbner basis for any **lex** or **deglex** order.

Gröbner Basis - Examples

In $\mathbb{F}[x, y]$:

- $\{y^2 - 1, y - x\}$ is not a Gröbner basis for **lex** order with $x < y$ (previous example).
- However, it is a Gröbner basis for **lex** order with $x > y$. Proof: $\mathrm{LM}(y^2 - 1) = y^2$ and $\mathrm{LM}(y - x) = x$ are **coprime**.
- $\{y^3 + x, y^3 + x^2\}$ is not a Gröbner basis for any **lex** or **deglex** order.
- However, it is a Gröbner basis for **weighted degree** orders with $\mathbf{wt}(x) = 2$ and $\mathbf{wt}(y) = 1$, as then $\mathrm{LM}(y^3 + x) = y^3$ and $\mathrm{LM}(y^3 + x^2) = x^2$ are **coprime**.

Anemoi - Model

Example ($\alpha = 3$, one round)

$$x_1^3 = a x_0^2 + b x_0 + c$$

$$x_0 x_1 + d x_1^2 + e x_0 + f x_1 + g = 0$$

Anemoi - Model

Example ($\alpha = 3$, one round)

$$x_1^3 = a x_0^2 + b x_0 + c$$

$$x_0 x_1 + d x_1^2 + e x_0 + f x_1 + g = 0$$

x_0^2 **cancels out: this isn't a Gröbner basis for any order!**

Example ($\alpha = 3$, one round)

$$x_1^3 = a x_0^2 + b x_0 + c$$
$$x_0 x_1 + d x_1^2 + e x_0 + f x_1 + g = 0$$

x_0^2 **cancels out: this isn't a Gröbner basis for any order!**
Solution: multiply last equation by x_1^2 and reduce it by the first equation. We get:

$$p^*(x_0, x_1) = a x_0^3 + bd x_0^2 x_1 + \cdots$$

Anemoi - Model

Example ($\alpha = 3$, one round)

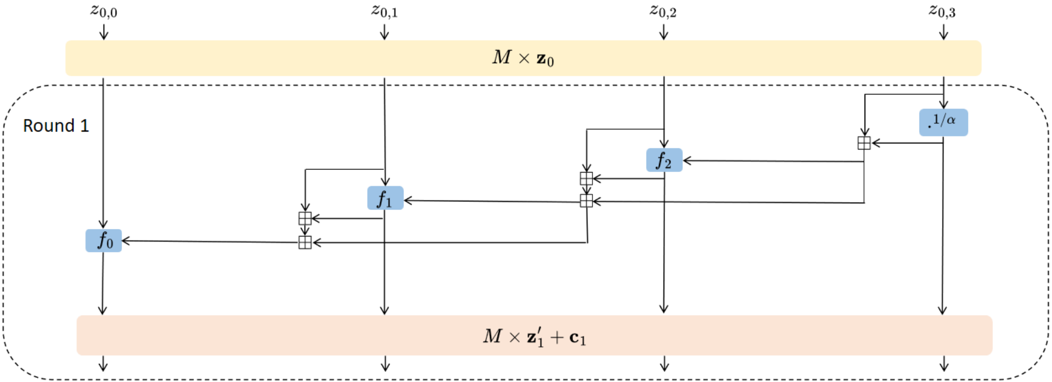
$$x_1^3 = ax_0^2 + bx_0 + c$$
$$x_0x_1 + dx_1^2 + ex_0 + fx_1 + g = 0$$

x_0^2 **cancels out: this isn't a Gröbner basis for any order!**
Solution: multiply last equation by x_1^2 and reduce it by the first equation. We get:

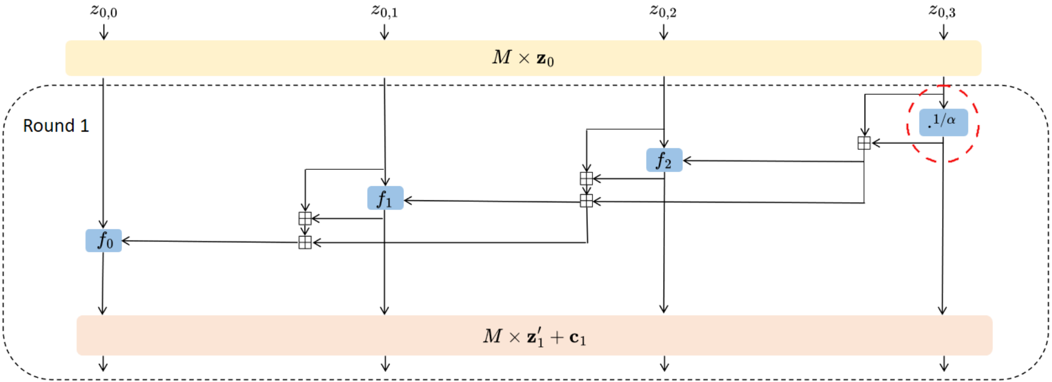
$$p^*(x_0, x_1) = ax_0^3 + bdx_0^2x_1 + \cdots$$

$\Longrightarrow$ The first equation and p^* are a Gröbner basis for some weighted order.
$\Longrightarrow$ This adds a few parasitic solutions (corresponding to $x_1 = 0$), but not many.
$\Longrightarrow$ This generalizes for more rounds (multiply the last polynomial by some of the x_i and reduce it). **Freelunch is saved**!

Arion-π - Round Function (4 branches)

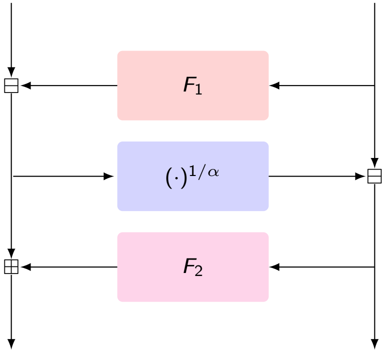


Arion-π - Round Function (4 branches)

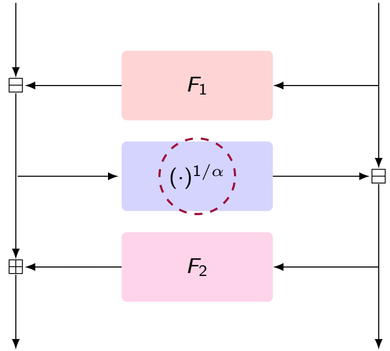


$(\cdot)^{1/\alpha}$ is the only high-degree operation $\implies$ add one variable per $(\cdot)^{1/\alpha}$.

Anemoi - Nonlinear layer (2 branches)

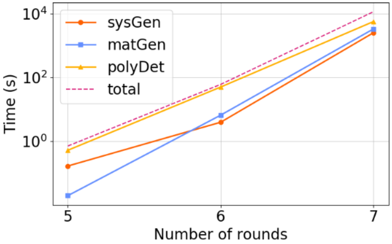


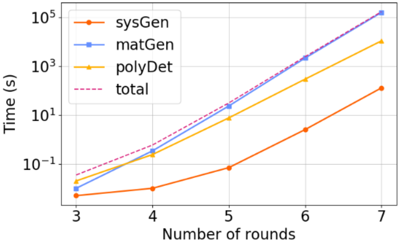
Anemoi - Nonlinear layer (2 branches)



$(\cdot)^{1/\alpha}$ is the only high-degree operation $\Longrightarrow$ add one variable per $(\cdot)^{1/\alpha}$.

Experimental Results

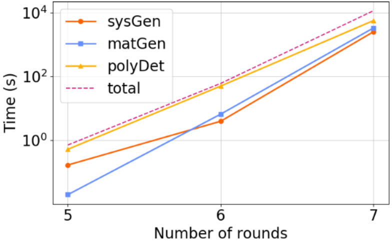


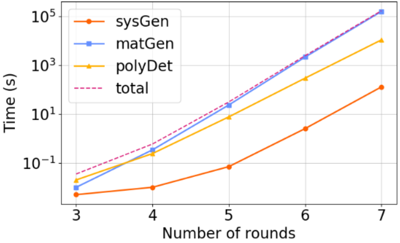


Complexity of Griffin (broke up to 7 out of 10 rounds, α=3)

Complexity of Anemoi (broke up to 7 out of 21 rounds, $\alpha = 3$)

Experimental Results





Complexity of Griffin (broke up to 7 out of 10 rounds, α=3)

Complexity of Anemoi (broke up to 7 out of 21 rounds, $\alpha = 3$)

$\Longrightarrow$ For Griffin, `polyDet` upper-bounds the others up to 7 rounds.

$\Longrightarrow$ For Anemoi, `matGen` is the bottleneck.