Simula UiB

On FreeLunches and Resultants: The Current Status of Algebraic Attacks Against AO-Hash Functions

Morten Øygarden

Simula UiB, Bergen

ALPSY, Obergurgl

January 2025

Part I - Some Thoughts on Solving Multivariate Polynomial System

Multivariate Polynomial System Solving

Consider a set of m polynomials $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$, and $(a_1, \ldots, a_m) \in \mathbb{F}^m$. How hard is it to solve the following system of equations?

$$
\begin{aligned}
f_1(x_1, \ldots, x_n) &= a_1, \\
&\vdots \\
f_m(x_1, \ldots, x_n) &= a_m.
\end{aligned}
$$

Solving Strategies

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

Solving Strategies

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

The ideal of generated by the set of polynomials $f_1, \ldots, f_m \in R = \mathbb{F}[x_1, \ldots, x_n]$ is

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

Solving Strategies

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

The ideal of generated by the set of polynomials $f_1, \ldots, f_m \in R = \mathbb{F}[x_1, \ldots, x_n]$ is

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

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

Gröbner Basis (GB)

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

Gröbner Basis (GB)

A GB 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). LT(f) is the largest term in f w.r.t. $\prec$.

Gröbner Basis (GB)

A GB 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). LT(f) is the largest term in f w.r.t. $\prec$.

Definition (Gröbner Basis)

A set of polynomials $G = \{g_1, ..., g_r\}$ 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 LT$(g_j)|$Lt(f).

Quick Overview

Gröbner Basis Attack in the Setting of This Talk, with $m = n$.

1. Compute a Gröbner Basis G.
2. Use G to compute[1] a univariate polynomial $F(X) \in I$.
3. Find the roots of $F(x)$.
4. Recover solutions for the other variables.

Rule of thumb: Step 1 or 2 typically the bottleneck. Steps 3 and 4 tend to be negligible in comparison.

[1] We will treat this step as a black box in this talk. (There are, however, lots of interesting things to discuss here!)

Part II - Polynomial Modeling
(On Blackboard)

Part III - The "FreeLunch" Method

Based on joint work with A. Bariant, A. Boeuf, A. Lemoine, I. Manterola Ayala, L. Perrin and H. Raddum. Presented at Crypto2024.

Monomial Order and Weight

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

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

Monomial Order and Weight

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

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

Example: Grading (comparison by degree) has weight vector

$$\mathsf{wt}(x_0) = \ldots = \mathsf{wt}(x_{n-1}) = 1.$$

An Easy Gröbner Basis Condition

Proposition

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

$$\mathsf{LM}_<(g_1), \ldots, \mathsf{LM}_<(g_\ell)$$

are pairwise **coprime**.

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

Choosing Monomial Orders

Let $wt(P_i) = max\{wt(m) \mid m$ is a monomial in $P_i\}$.

If x^{d^r} is a monomial in g and $\alpha > 1$, then we can choose $wt(z_i) = wt(P_i) - \delta$, for some small $\delta > 0$ s.t.

$$\alpha \cdot wt(z_i) > wt(P_i) > wt(z_i).$$

Example: Griffin-π - Model

$\alpha = 3$, $d = 7$, two rounds.

$$\begin{aligned} z_1^3 - ax + b &= 0, \\ z_2^3 - cx^7 + \cdots &= 0, \\ x^{49} + dx^{46} + ex^{45} + \cdots &= 0 \end{aligned}$$

Example: Griffin-π - Model

$\alpha = 3, d = 7$, two rounds.

$$
\begin{aligned}
z_1^3 - ax + b &= 0, \\
z_2^3 - cx^7 + \cdots &= 0, \\
x^{49} + dx^{46} + ex^{45} + \cdots &= 0
\end{aligned}
$$

In **grevlex** (degree-first), the leading monomials are z_1^3, x^7 and x^{49}. The Proposition does not apply.

Example: Griffin-π - Model

$\alpha = 3$, $d = 7$, two rounds.

$$
\begin{aligned}
z_1^3 - ax + b &= 0, \\
z_2^3 - cx^7 + \cdots &= 0, \\
x^{49} + dx^{46} + ex^{45} + \cdots &= 0
\end{aligned}
$$

In **grevlex** (degree-first), the leading monomials are z_1^3, x^7 and x^{49}. The Proposition does not apply.

In an order with $\mathbf{wt}(x) = \mathbf{wt}(z_1) = 1$ and $\mathbf{wt}(z_2) = 3$, the leading monomials are z_1^3, z_2^3 and x^{49}.
$\implies$ It's a Gröbner basis.

This generalizes for more rounds.

When does this *not* work?

We need some assumption on the pure x-term of highest degree in g.

There is only a single initial input x (and output).

We need at least one of the branches to not be inverted.

Part IV - Resultants

Based on the work of H. Yang, Q.-X. Zheng, J. Yang, Q. Liu and D. Tang, presented at AsiaCrypt2024.

Elimination Theory

For an ideal $I \subset \mathbb{F}[x, z_1, \ldots, z_r]$, we have the i-th *elimination ideal*

$$I_i = I \cap \mathbb{F}[x, z_1, \ldots, z_{r-i}].$$

Elimination Theory

For an ideal $I \subset \mathbb{F}[x, z_1, \ldots, z_r]$, we have the i-th *elimination ideal*

$$I_i = I \cap \mathbb{F}[x, z_1, \ldots, z_{r-i}].$$

The Elimination Theorem

Let G be a Gröbner basis of I w.r.t. the lexicographic order $x < z_1 < \ldots < z_r$. Then $G_i = G \cap \mathbb{F}[x, z_1, \ldots, z_{r-i}]$ is a Gröbner basis of I_i.

Elimination with Resultants

(Of two Polynomials)

Consider $f, p \in R[x]$ for some commutative ring R, where

$$f = \sum_{i=0}^{\gamma} a_i x^i,\ a_i \in R, \qquad g = \sum_{i=0}^{\delta} b_i x^i,\ b_i \in R.$$

Elimination with Resultants

(Of two Polynomials)

Consider $f, p \in R[x]$ for some commutative ring R, where

$$f = \sum_{i=0}^{\gamma} a_i x^i,\ a_i \in R, \qquad\qquad g = \sum_{i=0}^{\delta} b_i x^i,\ b_i \in R.$$

The *Sylvester matrix* of f and g in $R^{(\gamma+\delta)\times(\gamma+\delta)}$ is defined as:

$$\mathrm{Syl}_x(f,g) = \underbrace{\begin{bmatrix} a_\gamma & \cdots & a_1 & a_0 & & 0 \\ & \ddots & & \ddots & \ddots & \\ 0 & & a_\gamma & \cdots & a_1 & a_0 \\ b_\delta & b_{\delta-1} & \cdots & b_0 & & 0 \\ & \ddots & \ddots & & \ddots & \\ 0 & & b_\delta & b_{\delta-1} & \cdots & b_0 \end{bmatrix}}_{\gamma+\delta} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ \ddots \\ a \end{matrix}}\right\}\delta \\ \left.\vphantom{\begin{matrix} b \\ \ddots \\ b \end{matrix}}\right\}\gamma \end{matrix}.$$

Elimination with Resultants

(Of two Polynomials)

The resultant of f and g with respect to x is defined as:

$$\mathrm{Res}_x(f, g) = |Syl_x(f, g)| \in R.$$

$\mathrm{Res}_x(f, g)$ is a polynomial in the coefficients of f and g that does not depend on x.

A Succession of (two-polynomial) Resultants

Yang et. al., observes that for "our" polynomial systems, we can compute generators for

$$I_1 \supseteq I_2 \supseteq \ldots \supseteq I_r = I \cap \mathbb{F}[x],$$

by successively computing the resultants of two polynomials.

A Succession of (two-polynomial) Resultants

Yang et. al., observes that for "our" polynomial systems, we can compute generators for

$$I_1 \supseteq I_2 \supseteq \ldots \supseteq I_r = I \cap \mathbb{F}[x],$$

by successively computing the resultants of two polynomials. Write $f_i = z_i^\alpha - p_i$. Then

$$I_1 = \langle f_1, f_2, \ldots, f_{r-1}, \mathsf{Res}_{z_r}(g, f_r) \rangle,$$

A Succession of (two-polynomial) Resultants

Yang et. al., observes that for "our" polynomial systems, we can compute generators for

$$I_1 \supseteq I_2 \supseteq \ldots \supseteq I_r = I \cap \mathbb{F}[x],$$

by successively computing the resultants of two polynomials. Write $f_i = z_i^\alpha - p_i$. Then

$$I_1 = \langle f_1, f_2, \ldots, f_{r-1}, \mathsf{Res}_{z_r}(g, f_r) \rangle,$$

$$I_2 = \langle f_1, f_2, \ldots, f_{r-2}, \mathsf{Res}_{z_{r-1}}\left(\mathsf{Res}_{z_r}(g, f_r), f_{r-1}\right) \rangle,$$

and so on.

When Does This Not Work?

We need to have a single output constraint and input x. Otherwise all variables will show up in at least three polynomials.

Large $\alpha \Rightarrow$ large Sylvester matrix. I'm currently not aware of good algorithms for computing determinants over multivariate polynomial rings.

Part V - Open Problems

Open Problems 1

Can we do better than generic commutative algebra algorithms when taking extra structure from the cryptographic problem into account?

Open Problems 1

Can we do better than generic commutative algebra algorithms when taking extra structure from the cryptographic problem into account?

E.g.,

- Constructing multiplication matrices w.r.t. a GB G.
- Computing determinants over multivariate polynomial rings.
- General resultants involving more than two polynomials.

Open Problems 2

Everything in this talk requires a single output constraint. How will the techniques generalize for ≥ 2 outputs?