Gröbner Bases and Polynomial Systems from Cryptography

Shuhong Gao

Clemson University

SaTC Workshop on Privacy and Security
June 15–17, 2016, University of Wisconsin

Motivation

Solving polynomial systems is a ubiquitous problem in mathematics, sciences and engineerings:

- Any SAT problem can be easily converted into a system of quadratic polynomials over $\mathbb{F}_2$.
- An arithmetic circuit over $\mathbb{F}_q$ corresponds to a system of quadratic polynomials over $\mathbb{F}_q$.
- Breaking cryptosystems is equivalent to solving systems of quadratic polynomials.
- .....

Gröbner bases is a powerful tool for solving general polynomial systems.

What is Gröbner bases?

Let $\mathbb{F}$ be a field, for example, $\mathbb{F} = \mathbb{F}_q, \mathbb{Q}, \mathbb{R}, \mathbb{C}, \mathbb{F}_q(t_1, t_2)$ (the field of rational functions), ...

Example (Gcd and Euclidean algorithm)

Let $f(x), g(x) \in \mathbb{F}[x]$ be univariate polynomials and $\mathbf{I} = \langle f(x), g(x) \rangle$. Then $G = \{h(x)\}$ is a Gröbner basis for $\mathbf{I}$ iff $h(x) = \gcd(f(x), g(x))$.

What is Gröbner bases?

Let $\mathbb{F}$ be a field, for example, $\mathbb{F} = \mathbb{F}_q, \mathbb{Q}, \mathbb{R}, \mathbb{C}, \mathbb{F}_q(t_1, t_2)$ (the field of rational functions), ...

Example (Gcd and Euclidean algorithm)

Let $f(x), g(x) \in \mathbb{F}[x]$ be univariate polynomials and $\mathbf{I} = \langle f(x), g(x) \rangle$. Then $G = \{h(x)\}$ is a Gröbner basis for $\mathbf{I}$ iff $h(x) = \gcd(f(x), g(x))$.

Example (Gauss elimination)

$$f_i = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n + b_i \in \mathbb{F}[x_1, \ldots, x_n], \quad 1 \leq i \leq m.$$

Then $f_1, f_2, \ldots, f_m$ form a Gröbner basis iff they are in triangular form under some order of the variables.

What is Gröbner bases?

Example (Unique solution)

A system of polynomials in $\mathbb{F}[x_1, x_2, \ldots, x_n]$ has a unique solution (a solution with multiplicity one, over the algebraic closure of $\mathbb{F}$) iff the system has a Gröbner basis containing

$$x_1 - a_1, x_2 - a_2, \ldots, x_n - a_n$$

for some $a_i \in \mathbb{F}$, and $(a_1, a_2, \ldots, a_n)$ is the unique solution.

What is Gröbner bases?

Example (Hilbert's Theorem)

For any field $\mathbb{F}$, a system of polynomials $f_1, f_2, \ldots, f_m \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ has no solution over the algebraic closure of $\mathbb{F}$ if and only if there exist polynomials $u_1, u_2, \ldots, u_m \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ so that

$$u_1 f_1 + u_2 f_2 + \cdots + u_m f_m = 1.$$

This is equivalent to that the system has a Gröbner basis containing 1.

What is Gröbner Bases

For any $f_1, \ldots, f_m \in R = \mathbb{F}[x_1, \cdots, x_n]$, they define an **ideal** in R:

$$\mathbf{I} = \langle f_1, \ldots, f_m \rangle = \{u_1 f_1 + \cdots + u_m f_m : u_1, \ldots, u_m \in R\}.$$

The ideal $\mathbf{I}$ has the same solutions as the original polynomials.

Definition

For any **monomial order**, a subset $G = \{g_1, \ldots, g_m\} \subseteq \mathbf{I}$ is called a **Gröbner basis** (GB) for $\mathbf{I}$ if every $f \in \mathbf{I}$ is **reducible** by G, that is, there exists some $g \in G$ such that $\mathrm{lm}(g)$ divides $\mathrm{lm}(f)$.

A Gröbner basis will tells us if a polynomial system has a common solution, the number of solutions, and more...

Buchberger's Criterion

Theorem (Buchberger 1965)

Suppose $G = \{g_1, \ldots, g_m\}$ generate an ideal $\mathbf{I} \subseteq R$. Then G is a Gröbner basis for $\mathbf{I}$ iff, for every pair $1 \leq i < j \leq m$, $S(g_i, g_j)$ reduces to zero by G.

Note that $S(f, g)$ is called the S-polynomial of f and g, defined as

$$S(f, g) = t_1 f - c t_2 g,$$

where $c \in \mathbb{F}$ and t_1 and t_2 are monomials (minimum) so that the leading terms cancel. For example, let

$$f = 4x^3y^4 + \cdots, \qquad g = 5x^4yz^2 + \cdots.$$

$$S(f, g) = (5xz^2)f - (4y^3)g = xz^2(4x^3y^4 + \cdots) - \frac{4}{5}y^3(5x^4yz^2 + \cdots).$$

Buchberger's Criterion

Corollary: For any computable field $\mathbb{F}$, there is an algorithm to decide if any system of polynomials in $\mathbb{F}[x_1, \ldots, x_n$ has a solution over the algebraic closure of $\mathbb{F}$.

Warning: **Matiyasevich's Theorem (1970)**. If $\mathbb{F} = \mathbb{Z}$ (computable), deciding if a system of polynomials over $\mathbb{F}$ has a solution over $\mathbb{F}$ is **undecidable**, i.e. there is no algorithm even if you allow exponential time. (Using a recursively enumerable set and invoking Gödel's Incompleteness Theorem)

Open If $\mathbb{F} = \mathbb{Q}$ (computable), it is still an open question whether there exists an algorithm for deciding whether a system of polynomials over $\mathbb{F}$ has a solution over $\mathbb{F}$.

Buchberger's Algorithm

- The criterion tells us exactly what must be done.
- Suppose $G = \{g_1, \ldots, g_m\}$ is any given list of polynomials.

Algorithm

(1) *For each pair g_i and g_j from G,*
  (1a) *Reduce $S(g_i, g_j)$ by G until not reducible by G,*
  (1b) *If the remainder is nonzero, add it to G.*

(2) *Repeat Step 1 until all S-polynomials of G reduce to* 0.

Buchberger's Algorithm

- The criterion tells us exactly what must be done.
- Suppose $G = \{g_1, \ldots, g_m\}$ is any given list of polynomials.

Algorithm

(1) *For each pair g_i and g_j from G,*
  (1a) *Reduce $S(g_i, g_j)$ by G until not reducible by G,*
  (1b) *If the remainder is nonzero, add it to G.*

(2) *Repeat Step 1 until all S-polynomials of G reduce to 0.*

- **Step (1a) is very expensive, and many S-polynomials reduce to 0!**
- **How to detect such S-polynomials without performing reductions?**

Detecting useless S-polynomials

- Buchberger (1979): If $\gcd(\mathrm{lm}(g_i), \mathrm{lm}(g_j)) = 1$ then $S(g_i, g_j)$ can be top-reduced to 0 by G.
- Lazard (1983), Möller, Mora and Traverso (1992):

$$\text{syzygies} \longleftrightarrow \text{"reduction to 0"}.$$

For $(g_1, \ldots, g_m) \in R^m$, its syzygy module is defined as

$$H = \{\mathbf{u} = (u_1, \ldots, u_m) \in R^m : u_1 g_1 + \cdots + u_m g_m = 0\}.$$

- Faugère (F5, 2002): Introduces **signatures** and uses principal syzygies to detect useless S-polynomials.

Recent papers

- Bardet (PhD Thesis, 2006), Stegers (2006), Gash (PhD thesis, 2008), Eder and Perry (2009), Sun and Wang (2009),
- Hashemi and Ars (2010), Sun and Wang (2010), G., Guan and Volny (2010), Zobnin (2010),
- G., Volny and Wang (2010/2011), Volny (PhD Thesis, 2011),
- Huang (2010), Eder and Perry (2010),
- Arri and Perry (2011), Eder and Perry (2011), Eder, Gash, Perry (2011), Sun and Wang (2011), Bigatti, Caboara and Robbiano (2011),
- Roune and Stillman (2012), Galkin (2012), Sun and Wang (2012),
- Eder (2013), Eder and Roune (2013), Gerdt and Hashime (2013), Pan, Hu and Wang (2013), Sun and Wang (2013),
- Simões (PhD thesis, 2013), Sun (2013).
- .....

Our Contribution

Let $g_1, \ldots, g_m \in R = \mathbb{F}[x_1, \cdots, x_n]$. Define

$$H = \{(u_1, \ldots u_m) \in R^m : u_1 g_1 + \cdots + u_m g_m = 0\},$$

called **the syzygy module** of $\mathbf{g} = (g_1, \ldots, g_m)$.

Problem

Given $g_1, \ldots, g_m \in R$, we wish to compute a Gröbner basis for the ideal $I = \langle g_1, \ldots, g_m \rangle$ and a Gröbner basis for the syzygy module H.

Our Contribution

For any $g_1, g_2, \ldots, g_m \in R$, define the following R-submodule of $R^m \times R$:

$$M = \left\{(\mathbf{u}, v) \in R^m \times R : \mathbf{u}\mathbf{g}^t = u_1 g_1 + u_2 g_2 + \cdots + u_m g_m = v\right\}.$$

Then M is generated by

$$(\mathbf{E}_1, g_1), (\mathbf{E}_2, g_2), \ldots, (\mathbf{E}_m, g_m).$$

Definition

A subset G of M is called a **Strong Gröbner basis for** M if every pair in M is top-reducible by some pair in G.

A strong Gröbner basis contains Gröbner bases for the ideal $\mathbf{I}$ and the syzygy module H.

Our Contribution

Theorem (G, Volny and Wang 2011)

Suppose G is a subset of M containing $(\mathbf{E}_1, g_1), \ldots, (\mathbf{E}_m, g_m)$. For any term order on R and any compatible term order on R^m, the following are equivalent:

(a) G is a strong Gröbner basis for M,

(b) every J-pair of G is **covered** *by G.*

Shuhong Gao, Frank Volny and Mingsheng Wang, "A new framework for computing Gröbner bases", Mathematics of Computation, 85 (2016), no. 297, 449–465.

Remarks on implementation

- Checking if a pair $(\mathbf{u}, v) \in R^m \times R$ is covered G needs no reduction!!
- Any j-pair covered by G should be discarded. This gives all the rewritten rules used in $\mathbb{F}_5$ algorithms.
- Store only the signature $\mathrm{lm}(\mathbf{u})$, not the whole vector $\mathbf{u}$. This gives Gröbner basis for $\mathbf{I}$ and the minimal leading terms of the syzygy module.
- Use **trivial syzygies**. Any two pairs $p_1 = (\mathbf{u}_1, v_1)$ and $p_2 = (\mathbf{u}_2, v_2)$ give a trivial syzygy:

$$v_2 p_1 - v_1 p_2 = (\mathbf{u}, 0).$$

Computing discrete logarithms

Let $\mathbb{F}_q$ be a finite field of q elements. The discrete logarithm problem (DLP) over $\mathbb{F}_q$ is, given $\alpha, \beta \in \mathbb{F}_q$, find (if exists) an integer x, $0 \leq x < q-1$, so that

$$\beta^x = \alpha.$$

There are subexponential algorithms for solving DLP (via function field sieves or number field sieves).

Computing discrete logarithms

There is a nice way to convert DLP into a polynomial system over $\mathbb{F}_q$. Write x in binary form

$$x = x_0 + x_1 \cdot 2 + x_2 \cdot 2^2 + \cdots + x_{n-1} \cdot 2^{n-1},$$

where $x_i \in \{0, 1\}$. Then

$$\begin{aligned} \beta^x &= (\beta)^{x_0} \left(\beta^2\right)^{x_1} \left(\beta^{2^2}\right)^{x_2} \cdots \left(\beta^{2^{n-1}}\right)^{x_{n-1}} \\ &= (\beta_0)^{x_0} (\beta_1)^{x_1} (\beta_2)^{x_2} \cdots (\beta_{n-1})^{x_n - 1}, \end{aligned}$$

where $\beta_i = \beta^{2^i}$ for $0 \leq i \leq n-1$. Note that, in $\mathbb{F}_q$,

$$(\beta_i)^{x_i} = x_i(\beta_i - 1) + 1 = (x_i + \gamma_i)(\beta_i - 1)$$

as $x_i \in \{0, 1\}$ and $\gamma_i = 1/(\beta_i - 1) \in \mathbb{F}_q$.

Computing discrete logarithms

The DLP is reduced to solving the following system of polynomials over $\mathbb{F}_q$:

$$
\begin{aligned}
& x_i^2 - x_i, \quad i = 0, 1, 2, \ldots, n-1, \\
& y_1 - (x_0 + \gamma_0)(x_1 + \gamma_1), \\
& y_2 - y_1(x_2 + \gamma_2), \\
& y_3 - y_2(x_3 + \gamma_3), \\
& \vdots \\
& y_{n-2} - y_{n-3}(x_{n-2} + \gamma_{n-2}), \\
& \alpha_1 - y_{n-2}(x_{n-1} + \gamma_{n-1}).
\end{aligned}
$$

$2n$ equations in $2n-2$ variables: $x_0, x_1, \ldots, x_{n-1}, y_1, y_2, \ldots, y_{n-2}$.
How hard to solve such a structured system of polynomials?

Factoring integers

Factoring N is equivalent to deciding if the following polynomial system has a solution over $\mathbb{Z}_N$:

$$
\begin{aligned}
& x_i^2 - x_i, \quad i = 0, 1, 2, \ldots, n-1, \\
& y_1 - (x_0 + \gamma_0)(x_1 + \gamma_1), \\
& y_2 - y_1(x_2 + \gamma_2), \\
& y_3 - y_2(x_3 + \gamma_3), \\
& \vdots \\
& y_{n-2} - y_{n-3}(x_{n-2} + \gamma_{n-2}), \\
& 1 - y_{n-2}(x_{n-1} + \gamma_{n-1}),
\end{aligned}
$$

where γ_i are the same as in DLP with $\beta \in \mathbb{Z}_N$ random.

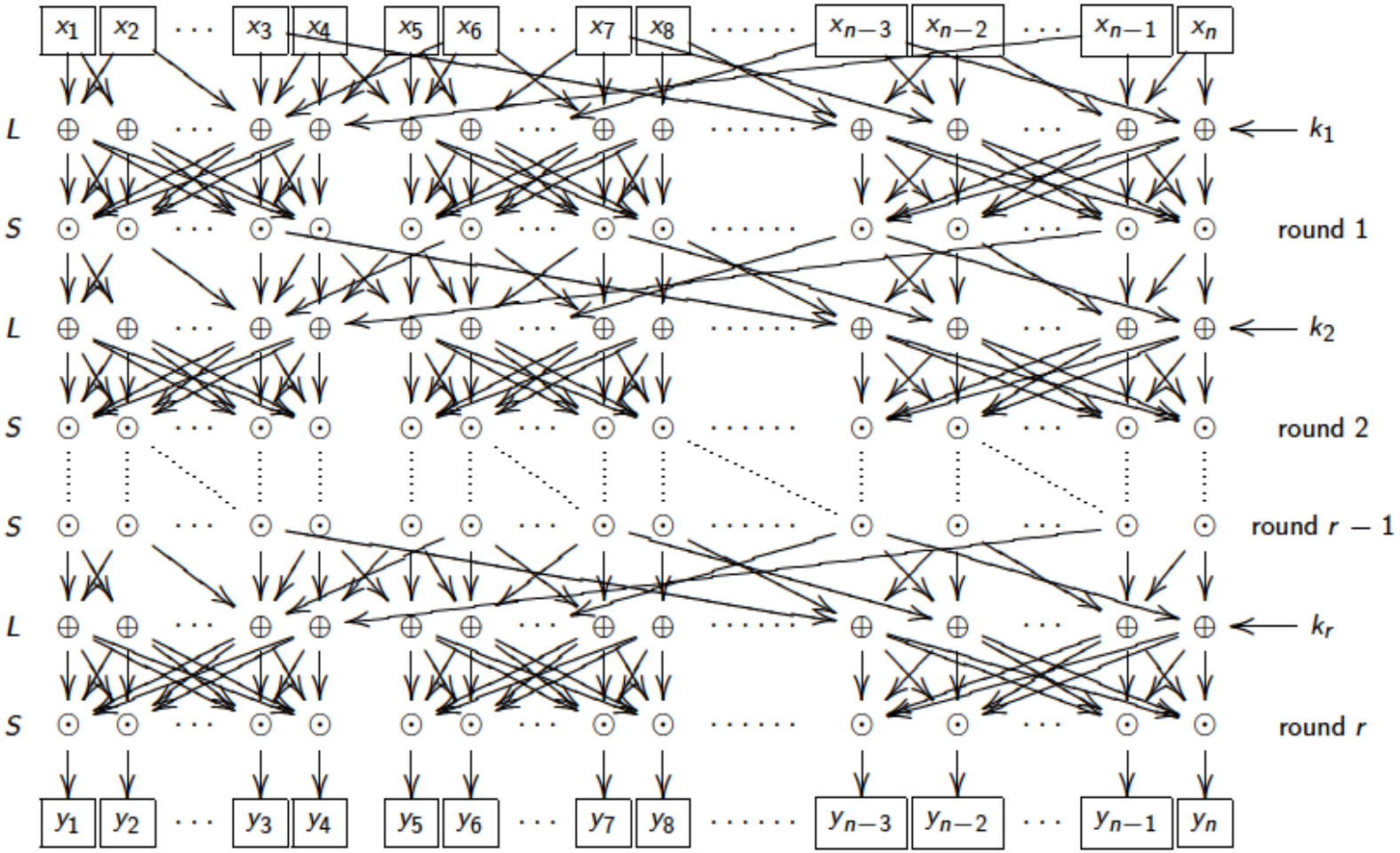
Advanced Encryption Standard: AES-128

- AES encryption has $r = 10$ rounds, each round consists of a linear map L (for diffusion) and an S-box S (which is nonlinear).
- The S-box has a nice algebraic description: for $x \in \mathbb{F}_{2^8}$,

$$y = S(x) = \begin{cases} 0, & \text{if } x = 0, \\ x^{-1}, & \text{if } x \neq 0. \end{cases}$$

- Each k has 128 bits, and is expanded into r subkeys $k_1, k_2, \ldots, k_r$. The key expansion algorithm also uses the same S-box, the only nonlinear operation.

Block cipher: AES-128 ($n = 128$ and $r = 10$)



AES-128: S-box

Polynomial systems over $\mathbb{F}_2$ for the S-box:

$$x_i^2 - x_i, 0 \le i \le 7, \quad y_i^2 - y_i, 0 \le i \le 7$$

$x_1 + y_6x_5 + y_5x_5 + y_3x_6 + y_4x_6 + y_6x_6 + y_1x_7 + y_2x_7 + y_4x_7 + y_7x_7 + y_2x_8 + y_5x_8 + y_7x_8 + y_8x_8 + y_5x_3 + y_1x_5 + y_8x_4$
$+y_1x_1 + y_8x_5 + y_7x_4 + y_7x_2 + y_3x_4,$
$y_1 + y_8x_2 + y_7x_1 + y_4x_3 + y_1x_1 + y_7x_2 + y_5x_1 + y_6x_3 + y_6x_4 + y_7x_4 + y_3x_5 + y_5x_5 + y_8x_5 + y_5x_6 + y_6x_6 + y_2x_7 + y_7x_7$
$+y_4x_7 + y_4x_8 + y_8x_7 + y_8x_8 + y_5x_8,$
$x_8 + y_5x_5 + y_3x_6 + y_1x_7 + y_7x_8 + y_7x_4 + y_5x_6 + y_7x_5 + y_3x_7 + y_8x_7 + y_1x_8 + y_6x_8 + y_6x_7 + y_4x_8 + y_8x_6 + y_2x_6 + y_8x_3$
$+y_4x_5 + y_6x_4 + y_8x_1 + y_2x_4 + y_4x_3 + y_6x_2,$
$y_8 + y_7x_1 + y_6x_3 + y_5x_5 + y_4x_7 + y_8x_7 + y_7x_3 + y_8x_1 + y_6x_5 + y_8x_6 + y_7x_8 + y_5x_7 + y_8x_4 + y_6x_8 + y_7x_6 + y_6x_2 + y_3x_8$
$+y_4x_6 + y_5x_4 + y_2x_6 + y_3x_4 + y_4x_2 + y_1x_8,$
$x_7 + y_6x_5 + y_4x_6 + y_2x_7 + y_7x_7 + y_5x_8 + y_8x_4 + y_7x_6 + y_8x_7 + y_5x_7 + y_3x_8 + y_6x_8 + y_1x_6 + y_7x_3 + y_3x_5 + y_2x_6 + y_8x_3$
$+y_5x_4 + y_4x_5 + y_6x_4 + y_5x_2 + y_1x_4 + y_3x_3 + y_7x_1,$
$y_7 + y_7x_2 + y_6x_4 + y_8x_5 + y_5x_6 + y_7x_7 + y_4x_8 + y_8x_3 + y_7x_5 + y_8x_6 + y_7x_8 + y_6x_7 + y_5x_3 + y_3x_7 + y_4x_5 + y_6x_1 + y_6x_2$
$+y_3x_8 + y_4x_6 + y_5x_4 + y_2x_5 + y_3x_3 + y_4x_1 + y_1x_7,$
$y_6 + y_4x_7 + y_7x_4 + y_5x_2 + y_4x_4 + y_2x_4 + y_6x_1 + y_1x_6 + y_7x_6 + y_3x_7 + y_4x_5 + y_8x_2 + y_8x_4 + y_6x_8 + y_3x_6 + y_8x_5 + y_7x_1$
$+y_6x_6 + y_5x_5 + y_5x_3 + y_7x_7 + y_5x_8 + y_6x_3 + y_3x_2 + y_2x_8 + y_8x_8,$
$y_4x_7 + x_6 + y_7x_3 + y_7x_4 + y_8x_6 + y_5x_4 + y_4x_4 + y_6x_1 + y_1x_6 + y_8x_2 + y_3x_6 + y_8x_5 + y_4x_2 + y_6x_6 + y_1x_7 + y_5x_5 + y_7x_7$
$+y_5x_8 + y_6x_7 + y_4x_8 + y_6x_3 + y_2x_3 + y_2x_8 + y_8x_8 + y_2x_5 + y_3x_5,$
$y_1x_5 + x_5 + y_8x_6 + y_5x_7 + y_4x_4 + y_6x_4 + y_7x_6 + y_3x_7 + y_2x_6 + y_4x_5 + y_8x_2 + y_7x_5 + y_8x_3 + y_7x_2 + y_3x_4 + y_3x_8 + y_1x_8$
$+y_5x_3 + y_5x_1 + y_1x_3 + y_6x_7 + y_4x_8 + y_6x_3 + y_5x_6 + y_3x_2 + y_8x_8 + y_2x_5 + y_7x_8,$

AES-128: S-box

Polynomial systems over $\mathbb{F}_2$ for the S-box (continued):

$y_1x_5 + y_5 + y_7x_3 + y_3x_1 + y_2x_7 + y_5x_2 + y_6x_2 + y_5x_4 + y_6x_5 + y_5x_7 + y_4x_4 + y_8x_1 + y_4x_3 + y_7x_6 + y_7x_5 + y_8x_4 + y_8x_3 + y_8x_7 + y_6x_8 + y_3x_6 + y_3x_8 + y_5x_1 + y_6x_7 + y_2x_3 + y_2x_8 + y_8x_8 + y_3x_5 + y_4x_6,$

$y_3 + y_7x_4 + y_5x_2 + y_8x_6 + y_2x_1 + y_4x_4 + y_6x_4 + y_6x_1 + y_7x_6 + y_3x_7 + y_4x_5 + y_8x_2 + y_7x_5 + y_8x_4 + y_8x_3 + y_8x_7 + y_6x_8 + y_3x_6 + y_6x_6 + y_7x_2 + y_5x_3 + y_1x_3 + y_5x_8 + y_6x_7 + y_4x_8 + y_5x_6 + y_2x_8 + y_8x_8 + y_7x_8,$

$y_4x_7 + x_3 + y_7x_3 + y_3x_1 + y_2x_7 + y_8x_6 + y_5x_4 + y_6x_5 + y_5x_7 + y_4x_4 + y_1x_6 + y_1x_2 + y_7x_6 + y_8x_2 + y_8x_4 + y_8x_7 + y_6x_8 + y_8x_5 + y_6x_6 + y_3x_8 + y_6x_7 + y_4x_8 + y_6x_3 + y_2x_8 + y_8x_8 + y_2x_5 + y_3x_5 + y_7x_8 + y_4x_6,$

$y_4x_7 + y_2 + y_7x_3 + y_2x_7 + y_7x_4 + y_5x_2 + y_8x_6 + y_6x_5 + y_5x_7 + y_4x_4 + y_8x_1 + y_1x_2 + y_4x_3 + y_8x_2 + y_7x_5 + y_8x_3 + y_8x_7 + y_3x_6 + y_8x_5 + y_7x_1 + y_6x_6 + y_5x_5 + y_5x_1 + y_7x_7 + y_5x_8 + y_6x_7 + y_6x_3 + y_2x_8 + y_3x_5 + y_7x_8,$

$y_4x_7 + y_1x_5 + x_2 + y_7x_4 + y_2x_1 + y_5x_7 + y_4x_4 + y_7x_6 + y_3x_7 + y_8x_2 + y_7x_5 + y_8x_7 + y_6x_8 + y_3x_6 + y_8x_5 + y_6x_6 + y_7x_2 + y_3x_4 + y_1x_7 + y_3x_8 + y_1x_8 + y_5x_5 + y_5x_3 + y_7x_7 + y_5x_8 + y_6x_3 + y_5x_6 + y_2x_8 + y_2x_5 + y_7x_8,$

$y_4x_7 + y_4 + y_7x_3 + y_2x_7 + y_7x_4 + y_6x_2 + y_5x_4 + y_6x_5 + y_5x_7 + y_8x_1 + y_2x_2 + y_6x_4 + y_6x_1 + y_4x_3 + y_7x_6 + y_3x_7 + y_4x_5 + y_8x_2 + y_7x_5 + y_8x_4 + y_8x_3 + y_6x_8 + y_7x_1 + y_6x_6 + y_7x_2 + y_3x_8 + y_5x_5 + y_5x_3 + y_5x_1 + y_5x_8 + y_6x_7 + y_4x_8 + y_6x_3 + y_5x_6 + y_3x_5 + y_1x_4 + y_4x_6,$

$y_4x_7 + y_1x_5 + x_4 + y_7x_3 + y_2x_7 + y_7x_4 + y_8x_6 + y_5x_4 + y_6x_5 + y_5x_7 + y_2x_2 + y_6x_4 + y_1x_6 + y_7x_6 + y_3x_7 + y_2x_6 + y_4x_5 + y_7x_5 + y_8x_4 + y_8x_3 + y_4x_1 + y_3x_6 + y_8x_5 + y_6x_6 + y_7x_2 + y_3x_4 + y_1x_7 + y_3x_8 + y_1x_8 + y_5x_5 + y_5x_3 + y_6x_7 + y_4x_8 + y_5x_6 + y_2x_8 + y_3x_5 + y_4x_6.$

A Gröbner basis for this system has 39 linearly independent quadratic polynomials. .....omitted.....

AES-128

AES-128 is equivalent to a system of quadratic polynomials over $\mathbb{F}_2$:

- Number of variables: 1728.
- Number of quadratic equations: 7688.

They have some layered structure corresponding to the rounds in encryption.

Open Problems

1. What size of random (or general) polynomial systems over $\mathbb{F}_2$ can be solved by the recent algorithms? The number of variables and number of equations both bigger than 200?
2. For bock ciphers and stream ciphers, is it possible to explore the structure? Say compute Gröbner bases for many local or subsystems (of one or two rounds), mix them together, then compute Gröbner bases for many random small systems?

Open Problems

3. There are efficient SAT solvers (for satisfiability problem), which are good for certain systems, but bad for systems from cryptography. Is it possible to combine the approach of SAT solvers with Gröbner bases?
4. Improve on bounds for the degree D so that

$$\{u_1 g_1 + \cdots + u_m g_m : u_i \in \mathbb{F}[x_1, \ldots x_n], \deg(u_i) \leq D\}$$

contains a Gröbner basis. This is closely related to the Castelnuovo-Mumford regularity (assuming g_i's are homogeneous).

Open Problems

5. Apply Gröbner basis approach to LWE? (There is a recent paper on this by Ge and Arora)
6. Exploring algebraic techniques to show that there is no polynomial time algorithm for solving systems of quadratic polynomials? NP $\neq$ P?

Thank you!