Algebraic Cyrptanalysis of Flurry and Curry using correlated messages.

Jean-Charles Faugère Ludovic Perret
Salsa Project
INRIA, Université Paris 6

with partial support of Celar/DGA

Journées C2 2008 - Carcans

Goal: how Gröbner bases can be used to break (block) ciphers ?

1. Use the same benchmark during the talk: non-trivial iterated block ciphers from **"Block Ciphers Sensitive to Gröbner Basis Attacks"**, [BPW] *J. Buchmann, A. Pyshkin and R.-P. Weinmann, CT-RSA 2006*

Goal: how Gröbner bases can be used to break (block) ciphers ?

1. Use the same benchmark during the talk: non-trivial iterated block ciphers from **"Block Ciphers Sensitive to Gröbner Basis Attacks"**, [BPW] *J. Buchmann, A. Pyshkin and R.-P. Weinmann, CT-RSA 2006*
2. Quick state of the art: zero-dimensional solving using Gröbner Bases

Goal: how Gröbner bases can be used to break (block) ciphers ?

1. Use the same benchmark during the talk: non-trivial iterated block ciphers from **"Block Ciphers Sensitive to Gröbner Basis Attacks"**, [BPW] *J. Buchmann, A. Pyshkin and R.-P. Weinmann, CT-RSA 2006*
2. Quick state of the art: zero-dimensional solving using Gröbner Bases
3. Test different algorithms and strategies: Direct, Substitution of some variables, several plaintexts/ciphertexts, several correlated plaintexts/ciphertexts.
   Results: Algebraic Crytptanalysis ⟷ essentially experimental results in this talk

Properties of Gröbner bases I

$\mathbb{K}$ a (finite) field (and ω is a primitive element when $\mathbb{K} = \mathbb{F}_{2^k}$), $\mathbb{K}[x_1, \ldots, x_n]$ polynomials in n variables.

Definition (Buchberger)

$<$ admissible ordering (lexicographical, total degree, DRL)

$G \subset \mathbb{K}[x_1, \ldots, x_n]$ is a Gröbner basis of an ideal I if

$$\forall f \in I, \text{ exists } g \in G \text{ such that } \mathrm{LT}_<(g) \mid \mathrm{LT}_<(f)$$

Solving algebraic systems:
Computing the algebraic variety: $\mathbb{K} \subset \mathbb{L}$ (for instance $\mathbb{L} = \overline{\mathbb{K}}$ the algebraic closure)

$$V_{\mathbb{L}} = \{(z_1, \ldots, z_n) \in \mathbb{L}^n \mid f_i(z_1, \ldots, z_n) = 0, \quad i = 1, \ldots, m\}$$

Properties of Gröbner bases II

Solutions in finite fields:

We compute the Gröbner basis of $G_{\mathbb{F}_2}$ of $[f_1, \ldots, f_m, x_1^2 - x_1, \ldots, x_n^2 - x_n]$, in $\mathbb{F}_2[x_1, \ldots, x_n]$. It is a description of all the solutions of $V_{\mathbb{F}_2}$.

Properties of Gröbner bases III

Theorem

- $V_{\mathbb{F}_2} = \emptyset$ *(no solution) iff* $G_{\mathbb{F}_2} = [1]$.
- $V_{\mathbb{F}_2}$ *has exactly one solution iff* $G_{\mathbb{F}_2} = [x_1 - a_1, \ldots, x_n - a_n]$ *where* $(a_1, \ldots, a_n) \in \mathbb{F}_2^n$.

Shape position:

If $m \geq n$ and the number of solutions is finite ($\# V_K < \infty$), then *in general* the shape of a lexicographical Gröbner basis: $x_1 > \cdots > x_n$:

$$\text{Shape Position} \left\{ \begin{array}{c} h_n(x_n)(=0) \\ x_{n-1} - h_{n-1}(x_n)(=0) \\ \vdots \\ x_1 - h_1(x_n)(=0) \end{array} \right.$$

Feistel cipher: FLURRY I

Flurry(k, t, r, f, D) the parameters used are:

- k *size* of the finite field $\mathbb{K}$: $k = \log(\mathbb{K})$.
- t is the *size* of the message/secret key and $m = \frac{t}{2}$ the half size.
- r the *number of rounds*.
- f a non-linear mapping giving the *S-Box* of the round function.

In practice: $f(x) = f_p(x) = x^p$ or $f(x) = f_{\text{inv}}(x) = x^{k-2}$.

- D a $m \times m$ matrix describing the *linear diffusion mapping* of the round function (coefficients in $\mathbb{K}$).

Feistel cipher: FLURRY II

We set $L = [l_1, \ldots, l_m] \in \mathbb{K}^m$ and $R = [r_1, \ldots, r_m]$ the *left/right* side of the current state. and $K = [k_1, \ldots, k_m]$ the *secret key*.

We define the *round function* $\rho : \mathbb{K}^m \times \mathbb{K}^m \times \mathbb{K}^m \to \mathbb{K}^m \times \mathbb{K}^m$ as

$$\rho(L, R, K) = (R, D.^{T} [f(r_1 + k_1), \ldots, f(r_m + k_m)] + L)$$

The key schedule. from an initial secret key $[K_0, K_1]$ (size $t = 2\,m$) we compute subsequent round keys for $2 \leq i \leq r + 1$ as follows:

$$K_i = D.^{T} K_{i-1} + K_{i-2} + v_i, \quad i = 2, 3, \ldots, (r + 1)$$

where v_i are round constants.

Feistel cipher: FLURRY III

A plaintext $[L_0, R_0]$ (size t) is *encrypted* into a ciphertext (L_r, R_r) by iterating the round function ρ over r rounds:

$$\begin{aligned} (L_i, R_i) &= \rho(L_{i-1}, R_{i-1}, K_{i-1}) \quad \text{for} \quad i = 1, 2, \ldots, (r-1) \\ (L_r, R_r) &= \rho(L_{r-1}, R_{r-1}, K_{r-1}) + (0, K_{r+1}) \end{aligned}$$

and $L_i = R_{i-1}$.

SPN: Curry I

We describe now the **Curry**$(k = \log(\mathbb{K}), m, r, f, D)$ family of SPN ciphers. Here :

- $m \in \mathbb{N}$ is the dimension of the plaintext space, the ciphertext and secret key spaces are $\mathbb{K}^{m \times m}$
- $r \in \mathbb{N}$ is the number of rounds
- $f : \mathbb{K} \to \mathbb{K}$ is a bijective non-linear mapping giving the S-Box of the round function
- $D \in \mathcal{M}_{m \times m}(\mathbb{K})$ is a matrix describing the linear diffusion mapping of the round function

The round function $\rho : \mathbb{K}^{m \times m} \times \mathbb{K}^{m \times m} \to \mathbb{K}^{m \times m}$ of Curry is given by :

$$\rho(S, K) = G(S, K) \cdot D,$$

with

$$G : X = \{x_{i,j}\} \in \mathbb{K}^{m \times m} \to G(X) = \{f(x_{i,j})\} \in \mathbb{K}^{m \times m}$$

SPN: Curry II

being the parallel application of the S-Box f to the components of X.

A plaintext $m = S_0 \in \mathbb{K}^{m \times m}$ is encrypted into a ciphertext $c = \in \mathbb{K}^{m \times m}$ by iterating the round function ρ exactly r times followed by an additional key-addition after the last round, namely :

$$\begin{array}{rcl} S_\ell & = & \rho(S_{\ell-1}, K_{\ell-1}), \text{ for all } \ell, \quad 1 \leq \ell \leq r-1, \\ c = S_r & = & \rho(S_{r-1}, K_{r-1}) + K_r. \end{array}$$

The key used for the first round key is the secret-key $K_0 \in K^{m \times m}$; subsequent round keys $K_i, i \geq 1$ are recursively computed via the formula :

$$K_i = K_{i-1} \cdot D + M_i,$$

$M_i = \{\omega^{i+(j-1)m+k}\}_{j,k,1 \leq j,k \leq m} \in \mathbb{K}^{m \times m}$ being a round constant.

Feistel/Curry ciphers: algebraic attack. I

Algebraic attack: The encryption process can be described by very simple polynomial equations: introduce variables for each round $L_j = [x_{1,j}, \ldots, x_{m,j}]$, $R_j = [x_{m+1,j}, \ldots, x_{t,j}]$ and $K_j = [k_{1,j}, \ldots, k_{m,j}] \longrightarrow F$ *algebraic set of equations.*

$$\text{for} \quad \begin{array}{rl} \text{plaintext:} & \vec{p} = L_0 \cup R_0 \\ \text{ciphertext:} & \vec{c} = L_{r+1} \cup R_{r+1} \\ \text{secret key:} & \vec{k} = K_0 \cup K_1 \end{array} \quad \text{of size } t \text{ equations:}$$

$\boxed{\mathcal{S}_{\vec{k}}(\vec{p}, \vec{c})}$ is the corresponding algebraic system

In the following: if $\vec{p}$ is explicitly known then we note $\vec{p}^*$; hence we obtain $\mathcal{S}_{\vec{k}}(\vec{p}^*, \vec{c}^*)$

Feistel/Curry ciphers: algebraic attack. II

Theorem

[Buchmann, Pyshkin, Weinmann]. If $f(x) = x^p$*, for an appropriate variable order* $x_{i,j}, k_{i,j}$ *then* $\mathcal{S}_{\vec{k}}(\vec{p}^*, \vec{c}^*)$ *is already a Gröbner basis for a total degree ordering.*

Main problem: we are computing $V_{\overline{\mathbb{K}}}$ and not $V_{\mathbb{K}}$!

and *many* solutions: $\boxed{p^{m\,r}}$

Algorithms I

Algorithms: for *computing* Gröbner bases.

- Buchberger (1965,1979,1985)
- F_4 using linear algebra (1999) (strategies)
- F_5 no reduction to zero (2002)

F_4 (1999) linear algebra
Small subset of rows: F_5 (2002) **full rank matrix** $F_5/2$ (2002) **full rank matrix** GF(2) (includes Frobenius $h^2 = h$)

$$A_d = \begin{array}{r c} & \textit{momoms degree } \mathbf{d} \textit{ in } x_1,\ldots,x_n \\ \begin{array}{r} \textit{monom} \times f_{i_1} \\ \textit{monom} \times f_{i_2} \\ \textit{monom} \times f_{i_3} \end{array} & \begin{pmatrix} \qquad\qquad & \ldots & \qquad\qquad \\ & \ldots & \\ & \ldots & \end{pmatrix} \end{array}$$

Complexity: driven by the maximal degree $D_{\max}$ occurring in the computation:

Algorithms II

Theorem

If $D_{\max}$ is the maximal degree occurring in the computation of a Gröbner basis of an ideal I, then the complexity of the computation is:

$$O\left(M(n, D_{\max})^3\right)$$

where $M(n,d)$ is the number of monomials in $\mathbb{K}[x_1,\ldots,x_n]$ of degree $\leq d$.

Sometimes we have a theoretical bound for $D_{\max}$: for a (semi)-regular sequence of m quadratic equations $(f_1,\ldots,f_m)$ we have

m	$D_{\max}$
$m \leq n$	$m+1$
$n+1$	$\frac{n+1}{2}$
$n+l$	$\frac{n+l}{2} - \mathrm{Herm}_{l,1}\sqrt{\frac{n+l}{2}} + o(1)$
$2\,n$	$\frac{n}{11.66} + O\left(n^{\frac{1}{3}}\right)$

FLURRY: first step

		Magma 2.11	Magma 2.13	FGb
Flurry(m,r,f)	deg(I)	F_4	F_4	F_5
(1,4, x^5)	625	0s	0s	0s
(m,r,x^p)	$p^{r\ m}$	0s	0s	0s
(2,4,x^3)	6521	0s	0s	0s
(1,10,x^{-1})	221	22.1 s	10.7 s	0.8 s
(1,12,x^{-1})	596	×	209.8 s	9.1 s
(2,5,x^{-1})	274	26.0 s	14.3 s	1.2 s
(2,6,x^{-1})	1126	×	902 s	46.9 s
(3,4,x^{-1})	583	×	83 s	12.2s

CPU Time: Gröbner DRL

Solving zero-dimensional system

When $\dim(I) = 0$ (finite number of solutions); in general:

- It is easier to compute a Gröbner Basis of I for a total degree ($<_{\mathrm{DRL}}$) ordering
- Triangular structure of Gb valid only for a lex. ordering:

$$\text{Shape Position} \begin{cases} h_n(x_n) = 0 \\ x_{n-1} = h_{n-1}(x_n) \\ \vdots \\ x_1 = h_1(x_n) \end{cases}$$

Dedicated Algorithm: efficiently change the ordering

FGLM, Gröbner Walk, LLL, . . .

FGLM

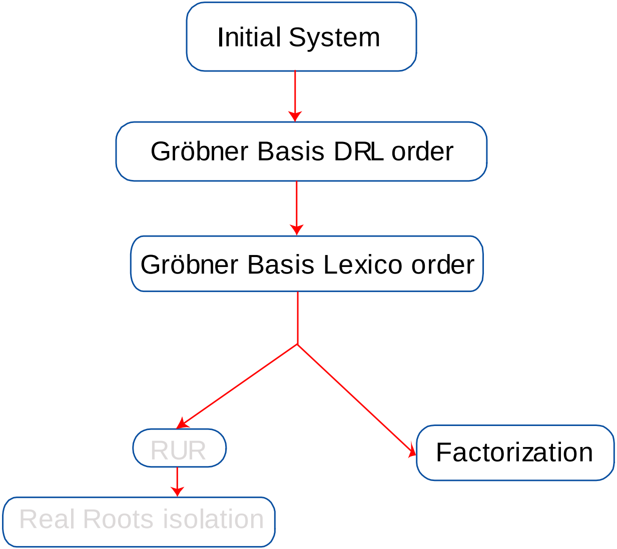
$\boxed{\text{Dedicated Algorithm: efficiently change the ordering}}$

FGLM = use only linear algebra.

Theorem (FGLM)

If $\dim(I) = 0$ and $D=\deg(I)$. Assume that G a Gröbner basis of I is already computed, then G_{new} a Gröbner basis for the same ideal I and a new ordering $<_{new}$can be computed in $O(n\,D^3)$.

Zero dim solve



Initial System

Gröbner Basis DRL order

Gröbner Basis Lexico order

RUR

Real Roots isolation

Factorization

Solving FLURRY I

		[BPW]	
Flurry(k, m, r, f, D)	#Sols	Magma	FGb
Flurry$(64, 1, 4, f_3, I_1)$	3^4	$<$ 0.1 s.	$<$ 0.1 s.
Flurry$(64, 1, 4, f_5, I_1)$	5^4	2.3 s.	$<$ 0.1 s.
Flurry$(64, 1, 4, f_7, I_1)$	7^4	82.62 s.	19.4 s.
Flurry$(64, 1, 6, f_{-1}, I_1)$	26	0.1 s.	0.7 s.
Flurry$(64, 1, 6, f_3, I_1)$	3^6	145.08 s.	2.1 s.
Flurry$(64, 1, 8, f_{-1}, I_1)$	79	1.46 s.	0.9 s.
Flurry$(64, 1, 10, f_{-1}, I_1)$	221	60.61 s.	1.9 s.
Flurry$(64, 1, 12, f_{-1}, I_1)$	596	2064 s.	16.5 s.
Flurry$(32, 2, 4, f_{-1}, D_2)$	59	65.78 s.	0.9 s.

Solving FLURRY II

Interpretation of the Results.

- non-negligible practical gain when using a sparse version of FGLM.
- this approach becomes quickly impractical due to huge dimension.
  $\longrightarrow$ mainly due to the fact that the field equations are not included in the input system. Therefore, the variety associated to these systems will mostly contain spurious solutions.

$$\boxed{\text{Intractable systems for large } t, r}$$

For $x \longmapsto x^p$ the complexity is $O\left(p^{\frac{3}{2}\, m\, r}\, \log\left(\#\mathbb{K}\right)\right)$.

Substitution of 1 variable

Compute a Gröbner basis of $I + \langle x_n - \alpha \rangle$ for some $\alpha \in \mathbb{K}$ (finite field).
Now we have an overdetermined algebraic system and only 1 or 0 solution !

$$\boxed{\text{DRL} + \text{FGLM}} \overset{?}{\longleftrightarrow} \boxed{(\#\mathbb{K})\,(\text{CPU overdetermined})}$$

Substitution of 1 variable

		Magma 2.13	FGb
Flurry(m,r,f)	deg(I)	F_4	F_5
(2,4,x^3)	6521	1.5 s	0.21 s
(2,6,x^{-1})	1126	6.0 s	0.39 s
(3,4,x^{-1})	583	0.22 s	0.10 s

CPU Time: Gröbner overdetermined

to be compared with:

		Magma 2.13	FGb
Flurry(m,r,f)	deg(I)	FGLM	FGLM
(2,4,x^3)	6521	Out of memory	991s
(2,6,x^{-1})	1126	20 m 35	1 m 21
(3,4,x^{-1})	583	441.2s	26.8s

CPU Time: Gröbner DRL + FGLM

Substitution of 1 variable

		FGb	FGb
Flurry(m,r,f)	deg(I)	FGLM	F_5
(2,4,x^3)	6521	991s	0.21 s
(2,6,x^{-1})	1126	1 m 21	0.39 s
(3,4,x^{-1})	583	26.8s	0.10 s

CPU Time: Gröbner overdetermined

Hence the second method is more efficient when

$$\#\mathbb{K} \leq \frac{60+21}{0.39} \approx \boxed{136} \text{ for FGb}$$

$$\#\mathbb{K} \leq \frac{20*60+35}{6.0} \approx \boxed{206} \text{ for Magma 2.13-10}$$

the complexity is $O\left((\#\mathbb{K}) \log\left(\#\mathbb{K}\right)\right)$

Several plaintexts I

We choose randomly several plaintexts: $\vec{p}_1^*, \ldots, \vec{p}_N^*$ and we assume that we known the corresponding ciphertext: $\vec{c}_i^*$
We obtain an algebraic system:

$$\mathcal{S}_N = \bigcup_{i=1}^{N} \mathcal{S}_{\vec{k}}(\vec{p}_i^*, \vec{c}_i^*)$$

Surprisingly, in this form this approach will not lead to any improvement !

N Nb of plain/cipher text	1	2	3
CPU	0.43 s	25.8s	16m42s
Nb of solutions	184	1	1

Flurry($\mathbb{K} = GF(2^7)$, $m = 4$, x^{-1})

Same behavior if we fix $k_{1\,0}$ (1 component of the secret key):

Several plaintexts II

N Nb of plain/cipher text	1	2	5	10
CPU	0.01s	0.09s	2.3s	99.5s
Nb of solutions	1	1	1	1

Flurry($\mathbb{K} = GF(2^7)$, $m = 4$, x^{-1}), substitution of 1 variable

This is likely due to the fact that we have increased the number of equations/variables.

Chosen plaintexts I

In order to take advantage of the use of several pairs, we have considered set of messages highly correlated.
The intuition behind this approach is the following: we expect that the quadratic terms of equations corresponding to different messages will be canceled.
For instance, consider a quadratic polynomial $p \in \mathbb{K}[x_1, \ldots, x_n]$, then its differential in $\mathbf{a} \in \mathbb{K}^n, p(\mathbf{x} + \mathbf{a}) - p(\mathbf{x})$ is linear.
Thus, we expect to generate new linear equations allowing the ease Gröbner basis computation. In particular, we hope to keep the maximal degree reached during the Gröbner basis computation as small as possible.
In our experiments, we have generated the set of messages by fixing :

$$\begin{aligned} \vec{p}_0^* &= (0, \ldots, 0), \text{ and} \\ \vec{p}_1^* &= (1, \ldots, 0) \end{aligned}$$

Chosen plaintexts II

We then constructed messages m_i, for all $i, 2 \leq i \leq N$, using the relation :

$$\begin{array}{rcl} \vec{p}_i^{\,*} & = & \vec{p}_j^{\,*} + \vec{e}_k \;\text{ with }\; j < i,\; 1 \leq k \leq t \\ \text{or}\;\; \vec{p}_i^{\,*} & = & \omega\; \vec{p}_{i-1}^{\,*} \end{array}$$

where $\vec{e}_i = [\ldots, 0, 1, 0, \ldots]$ canonical basis of $\mathbb{K}^t$.
We obtain an algebraic system:

$$\mathcal{S}_N = \bigcup_{i=1}^{N} \mathcal{S}_{\vec{k}}(\vec{p}_i^{\,*}, \vec{c}_i^{\,*})$$

Experimental results I

In the table below, we have quoted the results we have obtained on Flurry and Curry with different values of N. Note that we selected the parameters for having a secret-key of 128-bit. In the table :

- T_{BPW} is the estimated complexity of the attack of BPW
- T is the total time of our attack
- $D_{\max}$ is the maximal reached during the Gröbner basis computation

Experimental results II

		N	$D_{\max}$	T
Flurry(k, m, r, f, D)	T_{BPW}			
Flurry$(32, 2, 4, f_3, D_2)$	$\approx 2^{41}$	2	3	0.1 s.
Flurry$(32, 2, 4, f_5, D_2)$	$\approx 2^{58}$	3	5	$<$ 0.1 s.
Flurry$(32, 2, 4, f_7, D_2)$	$\approx 2^{70}$	3	7	0.13 s.
Flurry$(32, 2, 4, f_7, D_2)$	$\approx 2^{70}$	**4**	7	3.8 s.
Flurry$(32, 2, 5, f_3, D_2)$	$\approx 2^{47}$	3	5	58.8 s.
Flurry$(32, 2, 5, f_3, D_2)$	$\approx 2^{47}$	4	4	2.2 s.
Flurry$(32, 2, 5, f_3, D_2)$	$\approx 2^{47}$	**5**	3	0.1 s.
Flurry$(32, 2, 6, f_3, D_2)$	$\approx 2^{60.6}$	6	4	1438.4 s.
Flurry$(32, 2, 6, f_3, D_2)$	$\approx 2^{60.6}$	**14**	3	8.14 s.
Flurry$(32, 2, 7, f_3, D_2)$	$\approx 2^{60.6}$	30	3	317.51 s.
Flurry$(16, 4, 4, f_3, D_4)$	$\approx 2^{76}$	3	3	$<$ 0.1 s.
Flurry$(16, 4, 4, f_5, D_4)$	$\approx 2^{126}$	3	5	3.99 s.
Flurry$(16, 4, 4, f_7, D_4)$	$\approx 2^{177}$	3	7	105.4 s.

Experimental results III

		N	$D_{\max}$	T
Curry(k, m, r, f, D)	T_{BPW}			
Curry$(32, 2, 3, f_3, D_2)$	$\approx 2^{57}$	2	10	1.3 s.
Curry$(32, 2, 3, f_3, D_2)$	$\approx 2^{57}$	17	6	0.5 s.
Curry$(32, 2, 3, f_3, D_2)$	$\approx 2^{57}$	**20**	3	< 0.1 s.

Interpretation of the results

We can observe that our approach is significantly faster than the Gröbner conversion attack of BPW.

The most important is to observe that – in all cases – we have been able to find a number of pairs $N^* > 1$ such that the maximal degree reached during the Gröbner basis computation is equal to the degree d of the S-box function. In practice, we have found this N^* by performing the following test incrementally on $N \geq 2$: we compute a DRL Gröbner basis of $\mathcal{P}^N$. If the maximal degree reached during this computation is greater than d then we stop the computation and set $N \leftarrow N+1$, otherwise $N \leftarrow N^*$.

On this basis, we can extrapolate the (experimental) complexity of our attack. Let b_{F} (resp. b_{C}) be the number of variables of the system $\mathcal{P}^{N^*}_{\mathrm{Flurry}}$ (resp. $\mathcal{P}^{N^*}_{\mathrm{Curry}}$), we evaluate the complexity of our attack to :

$$\mathcal{O}\left(b_{\mathrm{F}}^{3\deg(f)}\right), \text{ for } \textbf{Flurry}(n, m, r, f, D)$$

$$\mathcal{O}\left(b_{\mathrm{C}}^{3\deg(f)}\right), \text{ for } \textbf{Curry}(n, m, r, f, D).$$

Conclusion

- One test example: $\mathbf{Flurry}(\log(\mathbb{K}), m, r, f, D)$ *Buchmann, Pyshkin, Weinmann*
- Several efficient algorithms for computing Gröbner Bases: F_4, F_5, *FGLM*
- Several implementations: Magma, FGb, Singular, . . .
- Different strategies: Direct, Substitution of some variables, chosen plaintexts
  - *Direct computation*: Gb + FGLM $O\left(p^{\frac{3}{2}\,m\,r}, \#\mathbb{K}\right)$
  - *Chosen plaintexts*:
    - Flurry broken (?) when $f = x^3$ and chosen plaintexts, complexity $O\left((t\,r)^\beta \log(\#\mathbb{K})\right)$ and $\beta \leq 9$.
    - The attack does not work for $f = \frac{1}{x}$ (or too big)
- Need some theory to predict/explain the behavior of Gröbner bases with correlated messages.