Comparison between XL and Gröbner Basis Algorithms

Gwénolé Ars[1], Jean-Charles Faugère[2], Hideki Imai[3], Mitsuru Kawazoe[4], and Makoto Sugita[5]

[1] IRMAR, University of Rennes 1
Campus de Beaulieu 35042 Rennes, France
gwenole.ars@univ-rennes1.fr
[2] LIP6/CNRS/INRIA, University of Paris VI
8 rue du Capitaine Scott Paris 75015 Paris, France
Jean-Charles.Faugere@lip6.fr
[3] Institute of Industrial Science, University of Tokyo
4-6-1 Komaba, Meguro-ku Tokyo, 153-8505, Japan
imai@iis.u-tokyo.ac.jp
[4] Department of Mathematics and Information Sciences, Osaka Prefecture University
1-1 Gakuen-cho Sakai Osaka 599-8531 Japan
kawazoe@mi.cias.osakafu-u.ac.jp
[5] IT Security Center, Information-technology Promotion Agency, Japan
2-28-8 Honkomagome, Bunkyo-ku Tokyo, 113-6591, Japan
m-sugita@ipa.go.jp

Abstract. This paper compares the XL algorithm with known Gröbner basis algorithms. We show that to solve a system of algebraic equations via the XL algorithm is equivalent to calculate the reduced Gröbner basis of the ideal associated with the system. Moreover we show that the XL algorithm is also a Gröbner basis algorithm which can be represented as a redundant variant of a Gröbner basis algorithm F_4. Then we compare these algorithms on semi-regular sequences, which correspond, in conjecture, to almost all polynomial systems in two cases: over the fields $\mathbb{F}_2$ and $\mathbb{F}_q$ with $q \gg n$. We show that the size of the matrix constructed by XL is large compared to the ones of the F_5 algorithm. Finally, we give an experimental study between XL and the Buchberger algorithm on the cryptosystem HFE and find that the Buchberger algorithm has a better behavior.

Keywords : Multivariate polynomial equations, Algebraic attacks, Solving Systems, Gröbner basis, XL algorithm, Semi-regular Sequences.

1 Introduction

Algebraic attacks are among the most efficient attacks for public key cryptosystems, block ciphers and stream ciphers. They try to recover a secret key by solving a system of algebraic equations. Algebraic attacks were first applied to

Matsumoto-Imai Public Key Scheme in [19] by Jacques Patarin and a similar attack was also applied in [15]. Algebraic attacks were also applied to block ciphers in [6], where the complexity for attacking AES and Serpent was evaluated. Moreover, algebraic attacks were applied to stream cipher in [7], [8], [9] and improved in [1].

As a general method to solve a system of algebraic equations, we know Gröbner basis algorithms. The fastest of such algorithms previously known are the F_4 and F_5 algorithms introduced in [11] and [12], respectively.

The XL algorithm was proposed as an efficient algorithm for algebraic attacks. It was first introduced in [20] and applied to an attack for HFE which is an improved version of Matsumoto-Imai Public Key Scheme. It was improved in [5]. As stated in [20], in cryptographic scheme, a system of algebraic equations we are interested in has a unique solution over its defining field. The XL algorithm was proposed as a powerful technique to solve such special systems. In [20], it was stated that the XL algorithm does not try to calculate a whole Gröbner basis and therefore it should be more efficient.

Recently, by using the algorithms F_4 and F_5, 80-bit HFE were first cryptanalyzed in [14], whereas the XL algorithm was not applicable to 80-bit HFE. Time results with an implementation under Magma are presented on A. Steel's web page (http://magma.maths.usyd.edu.au/users/allan/gb/). As we stated above, the F_4 and F_5 algorithms are Gröbner basis algorithms. Why did algebraic cryptanalysis based on these Gröbner basis algorithms exceed XL? We give an answer for this question in this article.

In this paper we clarify a relation between the XL algorithm and Gröbner basis algorithms. Moreover, we study the XL algorithm on semi-regular sequences, which correspond, according to a conjecture in a report [3], to almost all overdefined polynomial systems, and on the cryptosystem HFE.

More precisely, we show the following:

1. The XL algorithm does not introduce explicitly a monomial ordering. But we have proved that if the XL algorithm terminates, it will also terminate with a lexicographic ordering.
2. To solve a system of algebraic equations whose solution in a given finite field is unique amounts to nothing but to calculate the reduced Gröbner basis for the ideal associated with that system.
3. By 2, the XL algorithm is actually a Gröbner basis algorithm. Moreover it is *essentially* the same as the one treated in [17] and can be viewed as a redundant variant of a Gröbner basis algorithm F_4.
4. We study the XL algorithm on semi-regular sequences.
   On $\mathbb{F}_2$, that the degree D of the parameter needed for the XL algorithm is almost the same as the degree of the polynomials in the matrix constructed by the F_5 algorithm. But the complexity of these two algorithms is specified by the size of the matrix: for example, for a quadratic multivariate polynomials with $n = 128$ and $m = 130$, both algorithms reached the same degree 17 and the matrices generated by the XL algorithm will have about 170×10^{20} rows and 6×10^{20} columns compared to squared matrices with only 6×10^{20} rows and columns for the F_5 algorithm.

On the field $\mathbb{F}_q$, with q very large compared to n, we show the XL algorithm terminates for a degree higher than Gröbner basis algorithms with a DRL order. Then it is obvious that XL matrices are huge compared to F_5 matrices.

5. We complete this study on generic systems with a comparison of the XL algorithm and the Buchberger algorithm for a cryptosystem HFE. For this cryptosystem, a Gröbner basis algorithm finds a structure in the multivariate systems and never exceeds a low degree, whereas, for the XL algorithm, the degree seems to still increase with the number of variables n.

The XL algorithm was proposed to be a more efficient algorithm to solve a system of equations under a special condition without trying to calculate a whole Gröbner basis. But our results imply that the XL algorithm is not so efficient as it was expected to be.

In Section 2, we recall the description of the XL algorithm. In Section 3, we give an overview of the theory of Gröbner bases. In Section 4, we clarify a relation between the XL algorithm and the F_4 algorithm. In Section 5, we study the behavior of the XL algorithm on semi-regular sequences. In Section 6, we give experimental results on HFE systems and in Section 7, we conclude this report.

2 The basic principle of XL

The XL algorithm is given as an algorithm which solves systems of quadratic equations having a solution in k^n for a finite field $k = \mathbb{F}_q$. Let $\mathcal{A}$ be a system of multivariate equations $f_j = 0$, $(1 \le j \le m)$ for $f_j \in k[\mathbf{x}] := k[x_1, \ldots, x_n]$. We denote the ideal generated by all f_j in $\mathcal{A}$ by $\mathcal{I}_{\mathcal{A}}$. Then, XL is described as follows [20].

Algorithm 1 (The XL algorithm) *For a positive integer D, execute the following steps:*

1. **Multiply:** *Generate all the products $\prod_{j=1}^{r} x_{\ell_j} * f_i \in \mathcal{I}_{\mathcal{A}}$ with $r \le D-2$ and total degree $\le D$.*
2. **Linearize:** *Consider each monomial in the x_i of degree $\le D$ as a new variable and perform the Gaussian elimination on the equations obtained in Step 1. The ordering on the monomials must be such that all the terms containing one variable (say x_1) are eliminated last.*
3. **Solve:** *Assume that step 2 yields at least one univariate equation in the powers of x_1. Solve this equation over the finite fields (e.g., with Berlekamp's algorithm).*
4. **Repeat:** *Simplify the equations and repeat the process to find the values of the other variables.*

In the original definition of the XL algorithm in [20], only quadratic equations are treated. If we change the condition "with $r \le D-2$ and total degree $\le D$" in Step 1 to "with $r \le D - \deg(f_i)$", we can apply XL to a system of equations including a non-quadratic equation. Note that this change does not contradict the original XL setting when a system of equations consists of quadratic equations. So hereafter, we use this generalized version in order to work in general case.

Remark 1. We can replace Step 1 of the XL algorithm by considering f_i^* the *homogenization* of f_i: $f_i^* = Z^d f(\frac{x_1}{Z}, \ldots, \frac{x_n}{Z}) \in k[\mathbf{x}, Z]$ and products mf_i^* with m a monomial with degree $D - \deg(f_i^*)$. All the computation is exactly the same. So the behavior of XL is the same on the homogenization of the system $\mathcal{A}$ as on $\mathcal{A}$. We will use this remark on section 5, and for more properties of homogenization, we refer to [4].

3 Gröbner basis and some algorithms

3.1 Basic notation and definitions

Let $k[\mathbf{x}] = k[x_1, \ldots, x_n]$ be a polynomial ring with variables $x_1, \ldots, x_n$ over a field k. For a monomial $\mathbf{x}^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$, $|\alpha| := \sum_{i=1}^n \alpha_i$ is called the *total degree* of this monomial. In the following, the set of all monomials in variables $x_1, \ldots, x_n$ is denoted by $M(x_1, \cdots, x_n)$, or simply by M. In the theory of Gröbner bases, we need to consider a *monomial ordering* (cf. [10]). One of such ordering is the *degree reverse lexicographical order* (DRL) defined as follows:

Definition 1 (cf. [14]). *For $\alpha = (\alpha_1, \ldots, \alpha_n), \beta = (\beta_1, \ldots, \beta_n) \in \mathbb{Z}_{\geq 0}^n$, We say $\mathbf{x}^\alpha >_{DRL} \mathbf{x}^\beta$ if $|\alpha| = \sum_{i=1}^n \alpha_i > |\beta| = \sum_{i=1}^n \beta_i$, or $|\alpha| = |\beta|$ and the right-most nonzero entry of the vector $\alpha - \beta \in \mathbb{Z}^n$ is negative.*

There are many monomial orderings. We choose one of such orderings on T and write it as $<$.

A nonzero polynomial f in $k[\mathbf{x}]$ is written as $f = \sum_\alpha c_\alpha \mathbf{x}^\alpha$, $c_\alpha \neq 0$. We use the following notations:

$T(f) = \{c_{(\alpha_1, \cdots, \alpha_n)} x_1^{\alpha_1} \cdots x_n^{\alpha_n} \mid c_{(\alpha_1, \cdots, \alpha_n)} \neq 0\}$: the set of *terms* of f
$M(f) = \{x_1^{\alpha_1} \cdots x_n^{\alpha_n} \mid c_{(\alpha_1, \cdots, \alpha_n)} \neq 0\}$: the set of *monomials* of f

We denote the *total degree*, the *leading term*, the *leading coefficient* and the *leading term* with respect to $<$, by $\deg(f)$, $\mathrm{LM}(f)$, $\mathrm{LC}(f)$ and $\mathrm{LT}(f)$ respectively. (For each definition, see [10].)

The ideal in $k[\mathbf{x}]$ generated by a subset F is denoted by $\langle F \rangle$. We also denote by $\langle I_1, \ldots, I_n \rangle$ the minimal ideal containing ideals $I_1, \ldots, I_n$.

Under the above notation, a *Gröbner basis* is defined as follows.

Definition 2. *Let M be the set of all monomial of $k[\mathbf{x}]$ with a fixed ordering. A finite subset $G = \{g_1, \ldots, g_m\}$ of an ideal $\mathcal{I}$ is called a Gröbner basis if*

$$\langle \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_m) \rangle = \langle \mathrm{LT}(\mathcal{I}) \rangle.$$

For a given ideal $\mathcal{I}$, its Gröbner basis is not unique. But the *reduced Gröbner basis*, which is defined as follows, is uniquely determined.

Definition 3. *A Gröbner basis $G = \{f_1, \ldots f_m\}$ of an ideal $\mathcal{I}$ is called reduced Gröbner basis if for all i, $\mathrm{LC}(f_i) = 1$ and any monomial of f_i is not divisible by any element of $\mathrm{LM}(G \backslash \{f_i\})$.*

3.2 The Buchberger algorithm

An algorithm which calculates a Gröbner basis is called a *Gröbner basis algorithm*. The Buchberger algorithm is one of them.

Definition 4. *Let $f, g \in k[\mathbf{x}]$ be nonzero polynomials. The S-polynomial of f and g is the combination*

$$S(f,g) := \mathrm{LC}(g)\frac{\mathrm{lcm}(\mathrm{LM}(f),\mathrm{LM}(g))}{\mathrm{L}M(f)}f - \mathrm{LC}(f)\frac{\mathrm{lcm}(\mathrm{LM}(f),\mathrm{LM}(g))}{\mathrm{L}M(g)}g.$$

For a finite set G of polynomials in $k[\mathbf{x}]$ and a polynomial $f \in k[\mathbf{x}]$, we denote by $\bar{f}^G$, a remainder of f on division by G. (For the definition of division by a finite set of polynomials, see [10] for example.)

Theorem 1. *A basis $G = \{g_1, \dots g_m\}$ of an ideal $\mathcal{I}$ in $k[\mathbf{x}]$ is a Gröbner basis if and only if for all pairs $i \neq j$, $\overline{S(g_i, g_j)}^G = 0$.*

As a result of Theorem 1, we have the *The Buchberger algorithm*:

Algorithm 2 (The Buchberger algorithm)

Input: an ordered set $F = (f_1, \dots, f_m)$ in $k[\mathbf{x}]$
Output: a Gröbner basis $G = \{g_1, \dots, g_s\}$ for $I = \langle f_1, \dots, f_m\rangle$ with $F \subset G$
$G := F$
Repeat
    $H := G$
    For each pair (p, q), $p \neq q$ in H,
        If $S := \overline{S(p, q)}^H \neq 0$, Then $G := G \cup \{S\}$
Until H=G

We remark that the reduced Gröbner basis is calculated in a finite number of steps from a Gröbner basis.

3.3 Some other algorithms

D. Lazard in the articles [17] describes a relationship between the method of the computation of Gröbner bases and the one based on Gaussian Eliminations on matrix for the system $\mathcal{A}$. Moreover there are some other Gröbner basis algorithms based on Gaussian elimination: F_4 [11], $FGLM$ [13] and F_5 [12]. We explain now the relationship between polynomials and matrices.

For a system $\mathcal{A}$ of equations $f_j = 0$ $(j = 1, 2, \dots, m)$, let us consider a finite list $G = (g_1, \dots, g_m)$ of elements of the ideal generated by f_j, the ordered set $M_G = [t_1, \dots, t_l]$ of monomials of all g_i with respect to a fixed order $<$. A matrix A whose (i, j)-entry is given as the coefficient of t_j in g_i is called the *coefficient matrix* of G. Note that ${}^tG = A\,{}^tM_G$ where ${}^tG$ and ${}^tM_G$ mean the transpose of each. Let $\tilde{A}$ be the row echelon form of A obtained by using elementary row operations in a standard linear algebra[6]. Then we call $\tilde{G}$ given by ${}^t\tilde{G} := \tilde{A}\,{}^tM_G$ the *row echelon basis* of G. When we take the reduced row echelon form of G, we say $\tilde{G}$ *the reduced row echelon basis* of G (In [11], this is called the row echelon basis). Calculation of the reduced row echelon basis is an essential part of F_4.

[6] This procedure is so-called the Gaussian elimination.

4 Relation between XL and Gröbner basis algorithms

4.1 The choice of a monomial ordering

To compare the XL algorithm with Gröbner basis algorithms, we need to give an explicit monomial ordering for XL. As the XL algorithm does not give an explicit monomial ordering, we need to introduce the following lemma :

Lemma 1. *Let $\mathcal{A}$ be a system of m multivariate equations with n variables.*

$$\text{XL terminates for a degree } D \Longleftrightarrow \begin{array}{l}\text{XL terminates for a degree } D\\ \text{with the Lexicographic ordering}\end{array}$$

Proof. Let be M (respect. M') the coefficient matrix of the list $\{(\prod_{j=1}^{k} x_{i_j}) * f_i\}$ with $k \leq D - \deg(f_i)$ for XL (respect. with the Lexicographic ordering). So we can write $M = \left(A \middle| B\right)$ and $M' = \left(A' \middle| B'\right)$ such that B (respect. B') corresponds to the columns for the univariate monomials. Moreover M', A' and B' are only column permutations of M, A and B.

If XL terminates for a degree D, it means that $\text{rank}(M) > \text{rank}(A)$. Then $\text{rank}(M') > \text{rank}(A')$ and then XL will find an univariate polynomial with the lexicographic ordering. $\square$

4.2 Pre-assumption of the XL algorithm

Let $k = \mathbb{F}_q$ be a finite field with q elements and let $\mathcal{A}$ be a system of multivariate equations $f_j = 0$ $(1 \leq j \leq m)$ where $f_j \in k[x_1, \ldots, x_n]$. As stated *implicitly* in the introduction of [20], XL was proposed to be an efficient algorithm to solve a system of multivariate equations satisfying the following condition.

Condition 1 *The system $\mathcal{A}$ has only one solution $(x_1, \ldots, x_n) = (a_1, \ldots, a_n)$ in k^n. (i.e. $\mathcal{A}$ has a solution $(a_1, \ldots, a_n)$ in k^n and no other solution in k^n.)*

Note that the system $\mathcal{A}$ under Condition 1 can have another solution in K^n for some extension field $K (\neq k)$ of k. To determine the solution in k^n, we need extra equations $x_i^q - x_i = 0$ $(i = 1, \ldots, n)$. Thus the ideal we have to consider is generated by f_j $(j = 1, \ldots, m)$ and $x_i^q - x_i$ $(i = 1, \ldots, n)$. We denote this ideal by $\widetilde{\mathcal{I}}_{\mathcal{A}}$. Then we have the following important theorem.

Theorem 2. *Let $\mathcal{A}$ be a system of multivariate equations $f_j = 0$, $j = 1, 2, \ldots, m$ in $k[x_1, \ldots, x_n]$ with $k = \mathbb{F}_q$. Let $\widetilde{\mathcal{I}}_{\mathcal{A}}$ be the ideal $\langle f_1, \ldots, f_m, x_1^q - x_1, \ldots, x_n^q - x_n\rangle$. Then a solution $(x_1, \ldots, x_n) = (a_1, \ldots, a_n) \in k^n$ of $\mathcal{A}$ is unique in k^n if and only if $\widetilde{\mathcal{I}}_{\mathcal{A}} = \langle x_1 - a_1, \ldots, x_n - a_n\rangle$.*

Proof. If $(x_1, \ldots, x_n) = (a_1, \ldots, a_n)$ is a unique solution in k^n of $\mathcal{A}$, $\widetilde{\mathcal{I}}_{\mathcal{A}} \subset \langle x_1 - a_1, \ldots, x_n - a_n\rangle$ and $(a_1, \ldots, a_n)$ is a unique solution in $\bar{k}^n$ of a system which consists of $f_j = 0$ $(j = 1, \ldots, m)$ and $x_i^q - x_i = 0$ $(i = 1, \ldots, n)$ for an algebraic closure $\bar{k}$ of k because $x_i^q - x_i = 0$ has solutions only in k. Then from Hilbert's Nullstellensatz (cf. [10]), for each $i = 1, \ldots, n$, there exists a positive integer ℓ_i such that $(x_i - a_i)^{\ell_i} \in \widetilde{\mathcal{I}}_{\mathcal{A}}$. Since $x_i - a_i = \gcd(x_i^q - x_i, (x_i - a_i)^{\ell_i}) \in \widetilde{\mathcal{I}}_{\mathcal{A}}$, we have $\widetilde{\mathcal{I}}_{\mathcal{A}} = \langle x_1 - a_1, \ldots, x_n - a_n\rangle$. For the converse, it is obvious. $\square$

By this theorem, Condition 1 is equivalent to the following condition.

Condition 2 *The reduced Gröbner basis with respect to DRL of the ideal* $\widetilde{\mathcal{I}}_{\mathcal{A}} = \langle f_1, \ldots, f_m, x_1^q - x_1, \ldots, x_n^q - x_n \rangle$ *is* $\{x_1 - a_1, \ldots, x_n - a_n\}$.

Thus the problem to solve $\mathcal{A}$ defined over $k = \mathbb{F}_q$ under the Condition 1 coincides with the problem to calculate the reduced Gröbner basis of the ideal generated by equations in $\mathcal{A}$ and field equations $x_i^q - x_i = 0$ under the Condition 2, which is not a new problem. In particular, if the XL algorithm can solve a system $\mathcal{A}$ of algebraic equations over $\mathbb{F}_q$ under the Condition 1, it actually computes the reduced Gröbner basis of the ideal $\widetilde{\mathcal{I}}_{\mathcal{A}}$.

4.3 Relation between XL and the F_4 algorithm

We use the same notation as in (3.1). Here we show the XL algorithm gives a Gröbner basis algorithm which can be viewed as a redundant variant of the F_4 algorithm. (For the description of the original F_4, see [11].) To give such a description, we need the following definition.

Definition 5. *(1) A critical pair of two polynomials* (f_i, f_j) *is an element of* $M^2 \times k[\mathbf{x}] \times M \times k[\mathbf{x}]$, $\mathit{Pair}(f_i, f_j) := (\mathit{lcm}_{ij}, t_i, f_i, t_j, f_j)$ *such that*

$$\mathit{lcm}(\mathit{Pair}(f_i, f_j)) = \mathit{lcm}_{ij} = \mathrm{LM}(t_i f_i) = \mathrm{LM}(t_j f_j) = \mathit{lcm}(\mathrm{LM}(f_i), \mathrm{LM}(f_j)).$$

(2) For a critical pair $p_{ij} = \mathit{Pair}(f_i, f_j)$, $\deg(\mathit{lcm}_{ij})$ *is called the degree of* p_{ij} *and denoted by* $\deg(p_{ij})$*. Let* P *be a list of critical pairs. For* $p = \mathit{Pair}(f, g) \in P$ *and* $d \in \mathbb{N}$*, we define two functions* $\mathit{XLLeft}(p, d) = \{(t, f) | t \in M, \deg(t * f) \leq d\}$*, and* $\mathit{XLRight}(p, d) = \{(t, g) | t \in M, \deg(t * g) \leq d\}$*. We write* $\mathit{XLLeft}(P, d) = \bigcup_{p \in P} \mathit{XLLeft}(p, d)$ *and* $\mathit{XLRight}(P, d) = \bigcup_{p \in P} \mathit{XLRight}(p, d)$*.*

For a list of critical pairs P and a positive integer $d \in \mathbb{N}$, we set

$$\mathit{Sel}(P, d) := \{p \in P \mid \deg(\mathit{lcm}(p)) \leq d\}.$$

Now we give an F_4-like description of the XL algorithm.

Algorithm 3 (The XL Algorithm)

Input: $\begin{cases} F : \text{a finite subset of } k[\mathbf{x}] \\ \mathit{Sel} : \text{fixed as above.} \end{cases}$

Output: a finite subset of $k[\mathbf{x}]$.

$G := F$, $\tilde{F}_0^+ := F$ *and* $d := 0$

$P := \{\mathit{Pair}(f, g) | f, g \in G \text{ with } f \neq g\}$

While $P \neq \phi$ *Do*

  $d := d + 1$

  $L_d := \mathrm{XLLeft}(P, d) \cup \mathrm{XLRight}(P, d)$

  $P_d := \mathit{Sel}(P, d)$

  $P := P \setminus P_d$

  $\tilde{F}_d^+ := \mathrm{Reduction}(L_d)$

For $h \in \tilde{F}_d^+$ *Do*
$P := P \cup \{Pair(h,g) | g \in G\}$
$G := G \cup \{h\}$
Return G

Reduction
Input: a finite subset L *of* $M \times k[\mathbf{x}]$
Output: a finite subset of $k[\mathbf{x}]$ *(possibly an empty set).*
$F :=$ Symbolic Preprocessing(L)
$\tilde{F} :=$ *Reduction to Row Echelon Basis of* F *w.r.t.* $<$
$\tilde{F}^+ := \{f \in \tilde{F} | \mathrm{LM}(f) \notin \mathrm{LM}(F)\}$
Return $\tilde{F}^+$

Symbolic Preprocessing
Input: a finite subset L *of* $M \times k[\mathbf{x}]$
Output: a finite subset of $k[\mathbf{x}]$
$F := \{t * f \,|\, (t,f) \in L\}$
Return F

Remark 2. In the original description of XL, it seems that the bound D is taken globally at once. However, to implement XL, there seems to be the following four ways to realize the process determining the optimal value of D. Let $\mathcal{A}$ be a system of equations you want to solve. Then each way is described as follows.

1. Begin with $D = 1$. Do XL described as in Definition 1 for $\mathcal{A}$. If you cannot obtain the solution, set $D := D + 1$ and do XL again for $\mathcal{A}$ with the new D.
2. Begin with $D = 1$. Iterate 'Multiply' and 'Linearize' described as in Definition 1 for $\mathcal{A}$ by adding new equations obtained by 'Linearize' to $\mathcal{A}$. If you cannot solve the resulting system, then return to the original $\mathcal{A}$, set $D := D + 1$ and iterate the same procedure as for $D = 1$. Repeat until you obtain the solution.
3. Begin with $D = 1$. Do XL described as in Definition 1 for $\mathcal{A}$. If you cannot obtain the solution, then set $D := D + 1$, replace $\mathcal{A}$ by the resulting system obtained by 'Linearize' in the previous XL and do XL again for the new $\mathcal{A}$ and D. Repeat until you obtain the solution.
4. Begin with $D = 1$. Iterate 'Multiply' and 'Linearize' described as in Definition 1 for $\mathcal{A}$ by adding new equations obtained by 'Linearize' to $\mathcal{A}$. If you cannot solve the resulting system $\mathcal{A}'$, then replace $\mathcal{A}$ by $\mathcal{A}'$, set $D := D + 1$ and iterate the same procedure as for $D = 1$. Repeat until you obtain the solution.

The first two processes are slightly different from the others. The degree reached for the third and the fourth ones can be lower than the degree of the others. The Gaussian elimination of polynomials with degree D can give polynomials with lower or equal to $D - 1$. For example, let us consider the system $x_2^2 + x_3 = 0$, $x_1x_2 - x_2 = 0$, $x_3^3 + x_1 = 0$. For $D = 3$, the polynomial $x_3x_1 - x_3 = (x_1 - 1)(x_2^2 + x_3) - x_2(x_1x_2 - x_2)$ appear in resulting system obtained by 'Linearize', and then for $D = 4$, the third and fourth methods find the univariate polynomial $x_1^2 - x_1 = (x_1 - 1)(x_3^3 + x_1) - x_3^2(x_3x_1 - x_3)$. Whereas, the two first methods need a degree $D = 5$ to find this polynomial because $x_1^2 - x_1 = (x_1 - 1)(x_3^3 + x_1) - (x_3^2x_1 - x_3^2)(x_2^2 + x_3) + x_3^2x_2(x_1x_2 - x_2)$.

In the above description of XL, we take the third one. You may take one of the other three realizations but the rest of our result holds for all of them. We should remark that XL taking D as in the first one is *essentially* the same as the Gröbner basis algorithm treated in [17].

In the above description of the XL algorithm, we keep some redundancy in the description to show the similarity to the F_4 algorithm. Note that in algebraic attacks using XL, the input F should be a set of polynomials which comes from all equations in a given system of equations $\mathcal{A}$ whose solution in k^n is unique and all field equations $x_i^q - x_i = 0$ for all variables x_i. 'Multiply' in XL corresponds to the calculation of L_d and "Symbolic Preprocessing". And 'linearize' corresponds to "Reduction". Note that, XL in the above description can be viewed as a redundant variant of F_4. This is because XLLeft and XLRight collect more polynomials and therefore the set of polynomials constructed in "Symbolic Preprocessing" is much larger than the one in F_4. In fact, XL collects all the products $\prod_{j=1}^{r} x_{l_j} * f_i$ with $r \le D - \deg(f_i)$, whereas F_4 collects only polynomials needed in the Gaussian elimination.

The above description enables us to prove the following theorem.

Theorem 3. *Let F be a finite set of polynomials in $k[\mathbf{x}]$. Then Algorithm 3 computes a Gröbner basis G for the ideal $\langle F \rangle$ in $k[\mathbf{x}]$ such that $F \subseteq G$.*

Proof. Let d be a positive integer and G_d the set G obtained for that d in the while-loop. If $\tilde{F}_d^+ \neq \phi$, then $\deg h \le d$ for any $h \in \tilde{F}_d^+$ and hence $h \in L_{d+1}$ in the next loop. Then it is obvious that $h \notin \tilde{F}_{d+1}^+$. Since any $g \in G_{d-1}$ of $\deg g \le d$ is contained in L_d, $h \notin G_{d-1}$ for any $h \in \tilde{F}_d^+$ and hence we have $G_{d-1} \subsetneq G_d$ when $\tilde{F}_d^+ \neq \phi$.

First, we show that Algorithm 3 terminates in a finite number of steps. Suppose that Algorithm 3 does not terminate. Then there is an infinite sequence (d_i) of positive integers such that $d_i < d_{i+1}$ and $\tilde{F}_{d_i}^+ \neq \phi$ for all i. From the above observation, we have an infinite ascending chain $G_{d_i} \subsetneq G_{d_{i+1}} \subsetneq \cdots$. But it contradicts to the fact that the ring $k[\mathbf{x}]$ is noetherian.

Now we show the output G of Algorithm 3 is actually a Gröbner basis of $\langle F \rangle$. Since $G = \bigcup_{d \ge 0} \tilde{F}_d^+$ and $\tilde{F}_d^+ \subset \langle F \rangle$, we have $F \subset G \subset \langle F \rangle$. The remaining task is to show $\overline{S(f,g)}^G = 0$ for all $f \neq g$ in G. Put $\tilde{d} := \deg(Pair(f,g))$. Then the S-polynomial $S(f,g)$ is contained in $L_{\tilde{d}}$ and hence $\overline{S(f,g)}^{G_{\tilde{d}}} = 0$. In particular, we obtain $\overline{S(f,g)}^G = 0$. Thus, by Theorem 1, the output G is actually a Gröbner basis of $\langle F \rangle$. □

5 Semi-regular sequences

In this section, we try to give a bound on the matrix size of the XL algorithm compared to the matrix size of the F_5 algorithm for most polynomial systems.

5.1 Presentation of Semi-regular sequences

In the report [3], the notion of semi-regular sequences was presented for overdefined systems over the finite field $\mathbb{F}_2$ and for affine systems. We have to distinguish two important cases for finite fields, $\mathbb{F}_2$ and $\mathbb{F}_q$. In the field $\mathbb{F}_2$, we have a criterion deduced from the Frobenius application. If we are interested in a system $\mathcal{A}$ on a field $\mathbb{F}_q$, with $q \gg n$, i.e. q is very high compared to n, then the trivial relation issued from the Frobenius application will not be reached during computation and all the computation done is similar to computation on $\mathbb{Q}$.

Definition 6.

Homogeneous semi-regular sequence : *Let $f_1, \ldots, f_m$ be a sequence of m homogeneous polynomials (i.e. for all monomial t of f_i, $\deg(t) = \deg(f_i)$ in $\mathcal{R}_n^h := \mathbb{F}_2[x_1, \ldots, x_n]/\langle x_1^2, \ldots, x_n^2\rangle$ or $\mathbb{Q}[x_1, \ldots, x_n]$), and $\mathcal{I} = \langle f_1, \ldots, f_m\rangle$ an ideal of $\mathcal{R}_n^h$ or $\mathbb{Q}[x_1, \ldots, x_n]$.*

- *The* degree of regularity *of $\mathcal{I}$ is the minimal degree d such that $\{LT(f) \mid f \in \mathcal{I}, \deg(f) = d\}$ is exactly the set of monomials of degree d in $\mathcal{R}_n^h$, denoted by $D_{reg}(\mathcal{I})$.*
- *$f_1, \ldots, f_m$ is a* homogeneous semi regular sequence on $\mathbb{F}_2$ *if $\mathcal{I} \neq \mathcal{R}_n^h$ and for $i \in \{1, \ldots, m\}$, if $g_i f_i = 0$ in $\mathcal{R}_n^h/\langle f_1, \ldots, f_{i-1}\rangle$ and $\deg(g_i f_i) < D_{reg}(\mathcal{I})$ then $g_i = 0$ in $\mathcal{R}_n^h/\langle f_1, \ldots, f_{i-1}, f_i\rangle$.*
- *$f_1, \ldots, f_m$ is a* homogeneous semi regular sequence on $\mathbb{Q}$ *if $\mathcal{I} \neq \mathbb{Q}[x_1, \ldots, x_n]$ and for $i \in \{1, \ldots, m\}$, if $g_i f_i = 0$ in $\mathbb{Q}[x_1, \ldots, x_n]/\langle f_1, \ldots, f_{i-1}\rangle$ and $\deg(g_i f_i) < D_{reg}(\mathcal{I})$ then $g_i = 0$ in $\mathbb{Q}[x_1, \ldots, x_n]/\langle f_1, \ldots, f_{i-1}\rangle$.*

Affine semi-regular sequence : *Let $f_1, \ldots, f_m$ be a sequence of m polynomials, and $\mathcal{I} = \langle f_1, \ldots, f_m\rangle$ an ideal of $\mathbb{F}_2[x_1, \ldots, x_n]/\langle x_1^2 - x_1, \ldots, x_n^2 - x_n\rangle$ or $\mathbb{Q}[x_1, \ldots, x_n]$. Let f_i^h the homogeneous part of the largest degree of f_i.*

- *$f_1, \ldots, f_m$ is a* semi regular sequence *if $f_1^h, \ldots, f_m^h$ is a homogeneous semi-regular sequence.*
- *the* degree of regularity *of $\mathcal{I}$ is the degree of regularity of $\langle f_1^h, \ldots, f_m^h\rangle$, denoted by D_{reg}.*

With this sequence of polynomials, the matrix generated by the F_5 algorithm has a full rank for the degree $d < D_{reg}$. Moreover, all polynomials computed by F_5 have a degree lower or equal to D_{reg}.

This means that, for semi-regular sequences, the number of rows $H_{m,n}(d)$ of the matrix in the homogeneous case, for $d < D_{reg}$, is known, and is given by a recurrence formula $H_{m,n}(d) = H_{m-1,n}(d) + \#\{m_\ell \text{ monomial of degree } d - d_m\} - H_{m,n}(d - d_m)$ with initial conditions $H_{m,n}(d) = 0$ if $m \leq 0$ or $d < min(\deg(f_k) \mid k \leq m\}$. Then the number of rows of a matrix for the affine case is $\sum_{d'=1}^{d} H_{m,n}(d')$.

The degree D_{reg} corresponds to the degree d when we will have more rows than columns for the homogeneous part of the largest degree. It is the minimal degree such that $H_{m,n}(d) > \#\{m_\ell \text{ monomial of degree } d\}$. If we consider the series $f(y) = \sum_{d \geq 0}(H_{m,n}(d) - \#\{m_\ell \text{ monomial of degree } d\})y^d$, the degree D_{reg} is given when the coefficient of this series is negative. the expression of f for quadratic equations is :

$$\frac{(1+y)^n}{(1+y^2)^m} \text{ for } \mathbb{F}_2 \qquad\qquad \frac{(1-y^2)^m}{(1-y)^n} \text{ for } \mathbb{F}_q, \text{ with } q \gg n.$$

Moreover, in the article [3], the authors have made a conjecture verified on many computer experiments:

Conjecture 1. almost all polynomial systems are semi-regular sequences.

As the XL algorithm computes for an homogeneous system, we work on semi-regular sequences such that the homogenization of the sequences is still semi-regular. With these hypotheses, the conjecture is still true.

If we want to find an univariate polynomial for the original description of XL, we need to have a number of rows higher than the number of monomials with degree D minus the number of univariate monomials in X_1 (i.e., X_1 and 1 for $\mathbb{F}_2$ and $1, \ldots, X_1^D$, for $\mathbb{F}_q$).

This means that the degree D of the XL algorithm is given when the coefficient of this series is negative. the expression of f for quadratic equations is :

$$\frac{(1+y)^n}{(1-y)(1+y^2)^m} - \frac{1+y}{1-y} \text{ for } \mathbb{F}_2 \qquad\qquad \frac{(1-y^2)^m}{(1-y)^{n+1}} - \frac{1}{(1-y)^2} \text{ for } \mathbb{F}_q, \text{ with } q \gg n.$$

5.2 On the field $\mathbb{F}_2$

Figure 1(a) presents a comparison of the degree reached between the XL algorithm and Gröbner basis computation for a variation of the number of variables n and Figure 1(b) for a variation of the number of equations m.

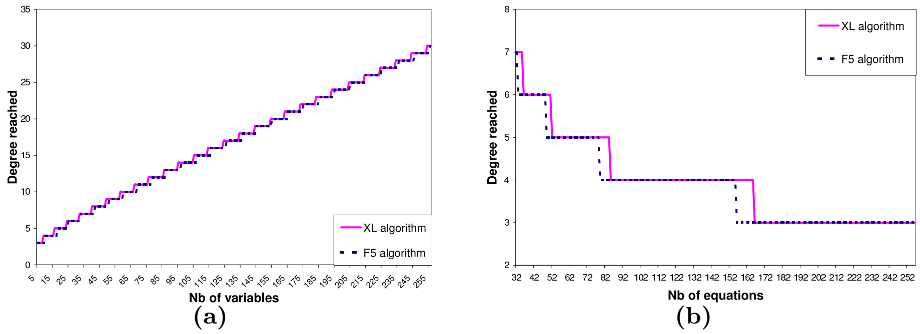


Fig. 1. Behavior of the XL algorithm and the F_5 algorithm on $\mathbb{F}_2$

With these figures, we do not have a noticeable difference between the degree reached by the two algorithms. So we can say that for random systems, the methods of XL and Gröbner basis are almost the same.

For the complexity point of view, if N_D is the size of the matrix constructed, then the whole complexity is the cost of linear algebra on this matrix, which is

N_D^w where $w \leq 3$ is the coefficient of linear algebra. The XL algorithm creates matrices with $\sum_{i=1}^{m} \sum_{k=0}^{D-\deg(f_i)} \binom{n}{k}$ rows and $\sum_{k=0}^{D} \binom{n}{k}$ columns, whereas F_5 creates square matrices with $\sum_{k=0}^{D} \binom{n}{k}$ columns.

So the number of columns for F_5 algorithm matrices is lower or equal to the one for XL algorithm matrices whereas the number of rows of the matrices constructed is very different, Figure 2 presents the number of rows of each matrices with a logarithm scale. As we can see, the difference between the two curves gives us a multiplicative constant.

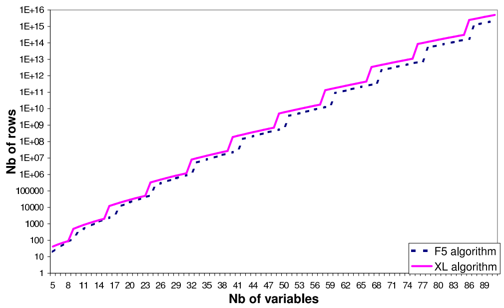


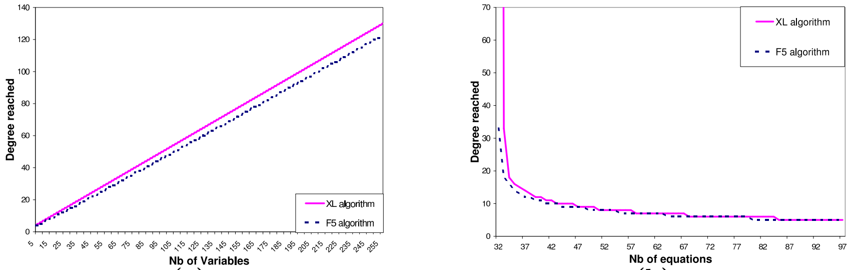
Fig. 2. Matrices of the XL algorithm and F_5 algorithm on $\mathbb{F}_2$

5.3 On the field $\mathbb{F}_q$, with q large

Figure 3(a) presents a comparison of the degree reached between the XL algorithm and Gröbner basis computation for a variation of the number of variables n with $m = n + 2$ and Figure 3(b) for a variation of the number of equations m. First we can see that for random polynomials we have always computed a Gröbner basis before finding the univariate polynomial for a degree D. Moreover, we can see the behavior of the degree of the XL algorithm does not seem to follow the formula $\frac{n}{\sqrt{m}}$ as it was said in [6].

As the complexity is N_D^w, where N_D is the size of the matrix constructed and w the coefficient of linear algebra and the XL algorithm has a higher degree D than the F_5 algorithm, the difference of the size of constructed matrices is very important. For example, for quadratic multivariate polynomials with $n = 128$ and $m = 130$, the XL algorithm reached a degree 66 whereas the F_5 algorithm reached a degree 61. So the matrices generated by the XL algorithm will have about 94317×10^{49} rows and 6332×10^{49} columns compared to squared matrices with only 8.4×10^{49} rows and columns for the F_5 algorithm.

For the case $m = n$, the number of solutions with multiplicity of a random system with quadratic equations is $\prod_{i=1}^{m} \deg(f_i) = 2^n$, which is the Bezout bound. So the univariate polynomial has this degree and XL will terminate for this degree. Whereas, the computation of the Gröbner basis will not exceed



(a) **(b)**

Fig. 3. Behavior of the XL algorithm and the F_5 algorithm on $\mathbb{F}_q$

$1+\sum_{i=1}^{n}(\deg(f_i)-1) = n+1$ for any ordering. This computation is done with a DRL ordering and then we use the FGLM algorithm [13, 10] to find the wanted ordering.

All this study is still true if $D < q$ and not only for $q \gg n$.

6 Example on HFE systems

In cryptography, the systems studied seem to be random but have a structure behind them. So we need to make experimental tests on cryptosystems to have an idea of the efficiency of both algorithms.

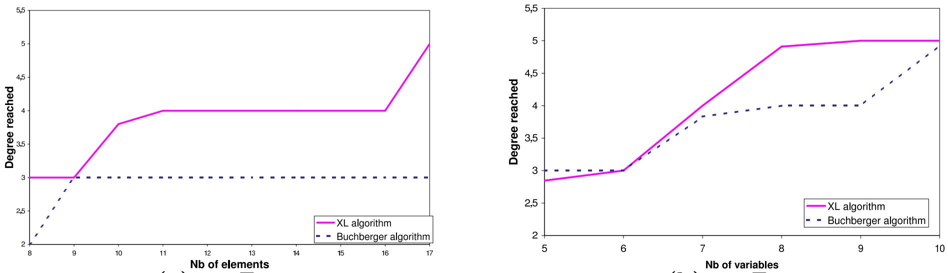
Hidden Field Equations (HFE) is an asymmetric cryptosystem. It does not use the number theory but it is based on multivariate polynomials over a finite field (cf [18]). The idea of HFE is to take a secret univariate polynomial (the private key) on an extension of the finite field, then to express this polynomial on the finite field. We thus obtain an algebraic system (the public key). This system is composed with polynomials of degree 2.

We have implemented the XL algorithm in Magma to test on the examples. Moreover as the XL algorithm has a better behavior for $m > n$, we have fixed some variables to be in the case $m = n + 2$. We studied on both cases presented in section 5, for the field $\mathbb{F}_2$, we use secret polynomials with degree 17 and with degree 24 for the field $\mathbb{F}_{16}$.

With Figure 4(a), we see that the XL algorithm's maximal degree increases whereas for Gröbner basis computation, the degree of resolution does not change and does not exceed 3. In fact, the XL algorithm seems to follow Figure 1(a). So XL does not seem to find a difference between a random system and the HFE cryptosystem contrary to Gröbner basis computation.

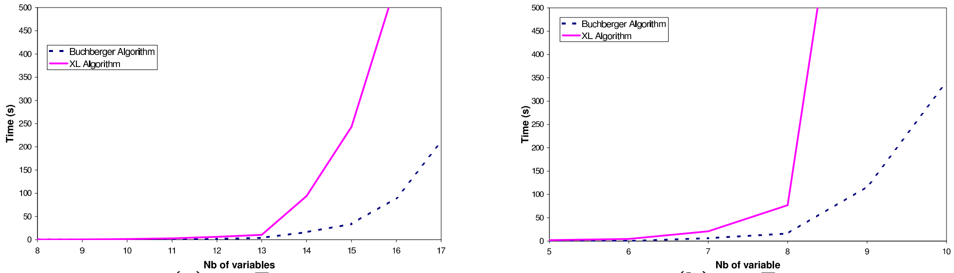
Figure 4(b) confirms that the Buchberger algorithm is still better than the XL algorithm on a bigger field for a number of elements higher than 6.

We present then time computation on figure 5. For the XL algorithm, the main part of computation is done in the Gaussian elimination and not in the other part of the algorithm. As we can see, the Buchberger algorithm has a better behavior than the XL algorithm.



(a) on $\mathbb{F}_2$ **(b)** on $\mathbb{F}_{16}$

Fig. 4. Comparison between XL and Gröbner algorithms on HFE



(a) on $\mathbb{F}_2$ **(b)** on $\mathbb{F}_{16}$

Fig. 5. Time comparison between XL and Gröbner algorithms on HFE

7 Conclusion

In this paper, we compared the XL algorithm with Gröbner basis algorithms. First, we showed that to solve a system of algebraic equations treated in XL is equivalent to calculate the reduced Gröbner basis of the ideal associated with the system. Moreover we showed that the XL algorithm is also a Gröbner basis algorithm which can be represented as a redundant variant of a Gröbner basis algorithm F_4. Then we compared these algorithms on semi-regular sequences in two cases: in the fields $\mathbb{F}_2$ and $\mathbb{F}_q$ with $q \gg n$. We showed that the size of the matrix constructed by XL is huge compared to the ones of F_5 algorithm. We gave an experimental study between XL and Buchberger algorithms on the cryptosystem HFE and found that the Buchberger algorithm had a better behavior. Our results imply that the XL algorithm is not so efficient as it was expected.

References

1. F. Armknecht, M. Krause, "Algebraic Attacks on Combiners with Memory", Crypto 2003, LNCS 2729, pp. 162-176, Springer.
2. G. Ars and J.-C. Faugère. "Comparison of XL and Gröbner Basis Algorithms over Finite Fields." , Technical report, INRIA Rocquencourt, 2004.

3. M. Bardet, J.-C. Faugère, and B. Salvy. "Complexity of Gröbner basis computation for semi-regular sequences over $\mathbb{F}_2$ with solutions in $\mathbb{F}_2$." , Technical report, INRIA Rocquencourt, 2003.
4. T. Becker and V. Weispfenning. "Gröbner Basis : A Computational Approach to Commutative Algebra", Springer-Verlag, New York, 1993.
5. N. Courtois, "The security of Hidden Field Equations (HFE)", Cryptographers' Track RSA Conference 2001, San Francisco 8-12 April 2001, LNCS 2020, Springer, pp. 266-281.
6. N. Courtois and J. Pieprzyk, "Cryptanalysis of Block Ciphers with Overdefined Systems of Equations", Asiacrypt 2002, LNCS 2501, Springer.
7. N. Courtois, "Higher Order Correlation Attacks, XL algorithm and Cryptanalysis of Toyocrypt", ICISC 2002, LNCS 2587, Springer.
8. N. Courtois and W. Meier, "Algebraic Attacks on Stream Ciphers with Linear Feedback", Eurocrypt 2003, Warsaw, Poland, LNCS 2656, pp. 345-359, Springer.
9. N. Courtois, "Fast Algebraic Attacks on Stream Ciphers with Linear Feedback", Crypto 2003, LNCS 2729, Springer.
10. D. Cox, J. Little, and D. O'Shea, "Using Algebraic Geometry", Springer-Verlag, New York, 1998.
11. J.-C. Faugère, "A new efficient algorithm for computing Gröbner bases (F_4)", Journal of Pure and Applied Algebra 139 (1999) pp. 61-88.
12. J.-C. Faugère, "A new efficient algorithm for computing Gröbner basis without reduction to zero (F_5)", In T. Mora, editor, Proceeding of ISSAC, pages 75-83, ACM Press, July 2002.
13. J.-C. Faugère, P. Gianni, D. Lazard, and T. Mora. "Efficient computation of zero-dimensional Gröbner bases by change of ordering". Journal of Symbolic Computation, 16(4):329–344, 1993.
14. J.-C. Faugère and A. Joux, "Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner bases", Crypto 2003, LNCS 2729, pp. 44-60, Springer.
15. A. Kipnis, J. Patarin, and L. Goubin, "Unbalanced Oil and Vinegar Signature Schemes", Eurocrypt 1999, Springer-Verlag, pp. 216-222.
16. A. Kipnis and A. Shamir, "Cryptanalysis of the HFE Public Key Cryptosystem", Proceedings of Crypto'99, Springer-Verlag.
17. D. Lazard, "Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations", Computer algebra (London, 1983), LNCS 162, pp. 146–156, Springer.
18. J. Patarin, "Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms", Lecture Notes in Computer Science, 1070:33–48, 1996.
19. J. Patarin, "Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88", Crypto'95, Springer, LNCS 963, pp. 248-261, 1995.
20. A. Shamir, J. Patarin, N. Courtois, and A. Klimov, "Efficient Algorithms for solving Overdefined Systems of Multivariate Polynomial Equations", Eurocrypt'2000, LNCS 1807, Springer, pp. 392-407.
21. M. Sugita and H. Imai, "Relations between Algebraic Attacks and Gröbner Base Algorithms", In The 2004 Symposium on Cryptography and Information Security, Japan – SCIS 2004, Jan.27–Feb.30, 2004.
22. M. Sugita, M. Kawazoe and H. Imai, "Relation between XL Algorithm and Gröbner Bases Algorithms", Cryptology ePrint Archive, Report 2004/112, 2004, http://eprint.iacr.org/.