A Variant of the XL Algorithm Using the Arithmetic over Polynomial Matrices *

Hiroki Furue
Graduate School of Information Science and Technology,
The University of Tokyo

Momonari Kudo
Graduate School of Information Science and Technology,
The University of Tokyo

Abstract

Solving a system of multivariate polynomials is a classical but very important problem in many areas of mathematics and its applications, and in particular quadratic systems over finite fields play a major role in the multivariate public key cryptography. The XL algorithm is known to be one of the main approaches for solving a multivariate system, as well as Groebner basis approaches, and so far many variants of XL have been proposed. In this talk, we present a new variant of XL, which we name "Polynomial XL", by using Macaulay matrices over polynomial rings.

1 Introduction

Solving a system of multivariate polynomials over a finite field is one of the most major problems in the field of computer science. Especially, the problem of solving a quadratic system (MQ problem) is very important, since it is used to construct various cryptographic systems (multivariate public key cryptosystems (MPKCs)). MPKCs are expected to be candidates for post-quantum cryptography because of the NP-completeness of the MQ problem [6]. Throughout the rest of this paper, we deal with only the case where the number of variables n is smaller than or equal to the number of equations m, because, in the case $n > m$, after $n - m$ variables are randomly specified, algorithms solving the MQ problem with $n \leq m$ can be applied.

Among various methods for solving algebraic systems, Gröbnor basis methods are main approaches solving the MQ problem. Faugère's F4 [4] and F5 [5] are today two major algorithms for computing Gröbnor bases, and they are proposed as efficient variants of Buchberger's algorithm [2]. On the other hand, another strategy to solve the MQ problem is linearization. In 2000, Courtois et al. [3] proposed a linearization based algorithm, which is called the XL algorithm, as an extension of the Relinearization algorithm proposed by Kipnis and Shamir [7]. The idea of XL is very simple; linearizing a given system by regarding each monomial as one variable. To solve the problem by such an idea, the number of independent equations needs to be close to the total number of monomials. To realize this, we generate a system composed of polynomials obtained by multiplying every polynomial of a given system by every monomial with degree smaller than or equal to a certain degree. For this system, we then generate its coefficient matrix (a *Macaulay matrix*). If the sufficient number of equations are prepared, then a univariate equation is obtained with Gaussian elimination on the Macaulay matrix. Finally, a solution is

[1] The title of this paper has been changed from the title of talk "Polynomial XL: A Variant of the XL Algorithm Using Macaulay Matrices over Polynomial Rings" at "Computer Algebra – Foundations and Applications".

found by solving the obtained univariate equation and repeating such processes to solve the remaining variables.

Subsequently, we introduce the hybrid approach [1, 8] (first proposed as FXL), which mixes exhaustive search and an MQ solver such as F4, F5, or XL. For a positive integer k with $k \leq n$ where n denotes the number of variables of a given system, the hybrid approach fixes the values of k variables and solves the remaining system in $n-k$ variables by an MQ solver. These processes are iterated until a solution is found. The hybrid approach may be effective in the case where the gain obtained by working on systems with less variables may overcome the loss due to the exhaustive search on the fixed variables. In this paper, we call the hybrid approach with XL *h-XL*.

In this paper, we propose a new variant of h-XL, which we call the polynomial XL (PXL). For the MQ system $\mathcal{F}$ of m equations in n variables $(x_1, \ldots, x_n)$ over $\mathbb{F}_q$, the proposed algorithm first sets the number k of guessed variables as in the hybrid approach. The main idea of PXL is to partly perform Gaussian elimination on a Macaulay matrix over a polynomial ring before fixing the values of k variables. By doing so, we can reduce the amount of manipulations for each guessed value compared with h-XL. Throughout this paper, let $F = (f_1, \ldots, f_m) \in \mathbb{F}_q[x_1, \ldots, x_n]^m$ be an MQ system of m polynomials in n variables $x_1, \ldots, x_n$ over $\mathbb{F}_q$, where q is a power of a prime.

2 Preliminaries

This subsection fixes the notations that are used in the rest this paper. In particular, we construct a Macaulay matrix *over the polynomial ring* $\mathbb{F}_q[x_1, \ldots, x_k]$ with respect to $x_{k+1}, \ldots, x_n$ for $1 \leq k \leq n$, where each entry belongs to $\mathbb{F}_q[x_1, \ldots, x_k]$. Such a Macaulay matrix, together with our construction, plays a key role in the main algorithm in Subsection 3.1 below.

In the following, an integer $1 \leq k \leq n$ is fixed, unless otherwise noted. Similarly to the hybrid approach [1, 8], our main algorithm divides $x_1, \ldots, x_n$ into k variables $x_1, \ldots, x_k$ and the remaining $n-k$ variables, and then regards $f_1, \ldots, f_m$ as elements of $(\mathbb{F}_q[x_1, \ldots, x_k])[x_{k+1}, \ldots, x_n]$. Let $K := (\mathbb{F}_q[x_1, \ldots, x_k])[x_{k+1}, \ldots, x_n]$ and $\mathrm{Mon}(K)$ denote the set of all monomials in $x_{k+1}, \ldots, x_n$, say

$$\mathrm{Mon}(K) := \left\{ x_{k+1}^{\alpha_{k+1}} \cdots x_n^{\alpha_n} : (\alpha_{k+1}, \ldots, \alpha_n) \in \mathbb{Z}^{n-k} \right\}.$$

For a positive integer $a \geq 0$, we here define two subsets T_a and $T_{\leq a}$ of $\mathrm{Mon}(K)$ as follows:

$$\begin{aligned} T_a &:= \{ t \in \mathrm{Mon}(K) : \deg(t) = a \}, \\ T_{\leq a} &:= T_0 \cup T_1 \cup \cdots \cup T_a. \end{aligned}$$

Furthermore, for a positive integer $b \geq 2$, we define two subsets I_b and $I_{\leq b}$ of K as follows:

$$\begin{aligned} I_b &:= \bigcup_{i=1}^{m} \{ t \cdot f_i : t \in T_{b-2} \}, \\ I_{\leq b} &:= I_2 \cup I_3 \cup \cdots \cup I_b. \end{aligned}$$

We here construct a Macaulay matrix of $I_{\leq D}$ with respect to $T_{\leq D}$ for an integer $D \geq 2$. For this, we use a *graded* monomial order (e.g., graded lexicographic order), which is a monomial order first comparing the total degree of two monomials. Furthermore, as for the order of elements in $I_{\leq D}$, we also use an order which first compares the degree of two polynomials. We then define the Macaulay matrix of $I_{\leq D}$ with respect to $T_{\leq D}$ by an $(|I_{\leq D}| \times |T_{\leq D}|)$-matrix over $\mathbb{F}_q[x_1, \ldots, x_k]$ whose (i, j)-entry is the coefficient of the j-th element of $T_{\leq D}$ in the i-th element of $I_{\leq D}$.

For simplicity of notation, we denote by $\mathcal{PM}$ the Macaulay matrix constructed as above, and call it a *Macaulay matrix of* F *at degree* D *over* $\mathbb{F}_q[x_1, \ldots, x_k]$. For two integers d_1 and d_2 ($2 \leq d_1 \leq D$, $0 \leq d_2 \leq D$), we also denote by $\mathcal{PM}[I_{d_1}, T_{d_2}]$ the submatrix of $\mathcal{PM}$ whose rows (resp. columns) correspond to polynomials of I_{d_1} (resp. monomials of T_{d_2}). Then, $\mathcal{PM}$ is clearly divided by submatrices $\mathcal{PM}[I_{d_1}, T_{d_2}]$ $(2 \leq d_1 \leq D, 0 \leq d_2 \leq D)$.

Thanks to our choice of a monomial order together with the quadraticity of F, the following lemma holds clearly:

Lemma 1
For an MQ system F and positive integers $k \leq n$ and $D \geq 2$, let $\mathcal{PM}$ be a Macaulay matrix of F at degree D over $\mathbb{F}_q[x_1, \ldots, x_k]$. Then, for $2 \leq d \leq D$, every $\mathcal{PM}[I_d, T_{d'}]$ with $d' \notin \{d, d-1, d-2\}$ is a zero matrix, and all elements of $\mathcal{PM}[I_d, T_d]$ belong to $\mathbb{F}_q$.

3 Main Algorithm

In this section, we propose a new variant of the XL algorithm.We describe the outline of our proposed algorithm "polynomial XL (PXL)" in Subsection 3.1 and the details of the most technical step in Subsection 3.2.

3.1 Outline of our algorithm PXL

This subsection describes the proposed algorithm polynomial XL (PXL). As in the h-XL, PXL first sets the first k variables $x_1, \ldots, x_k$ as guessed variables, whereas the main difference between our PXL and h-XL is the following: While h-XL performs row reduction after substituting actual k values to $x_1, \ldots, x_k$, PXL *partly* performs Gaussian elimination *before* fixing k variables. These manipulations are possible due to our construction of Macaulay matrices over $\mathbb{F}_q[x_1, \ldots, x_k]$ described in Lemma 1.

Here, we give the outline of PXL. The notations are same as those in Section 2.

Algorithm 1 (Polynomial XL)

Input: An MQ system $F = (f_1, \ldots, f_m) \in \mathbb{F}_q[x_1, \ldots, x_n]^m$, the number k of guessed variables, and a degree bound D.

Output: A solution over $\mathbb{F}_q$ to $f_i(x_1, \ldots, x_n) = 0$ for $1 \leq i \leq m$.

1. ***Multiply**: Compute the set $I_{\leq D}$ of all the products $t \cdot f_i$ with $t \in T_{\leq D-2}$.*
2. ***Linearize(1)**: Generate $\mathcal{PM}$, which is the Macaulay matrix of F at degree D over $\mathbb{F}_q[x_1, \ldots, x_k]$, and partly perform Gaussian elimination on it. (The details will be described in Subsection 3.2.)*
3. ***Fix**: Fix the values for the k variables $x_1, \ldots, x_k$ in the resulting matrix of step 2.*
4. ***Linearize(2)**: Compute the row echelon form of the resulting matrix of step 3.*
5. ***Solve**: If step 4 yields a univariate polynomial in $\mathbb{F}_q[x_n]$, compute its root.*
6. ***Repeat**: Substituting a root into x_n, simplify the equations, and then repeat the process to find the values of the other variables.*

Note that the definition of the resulting matrix of step 2 is given in the following paragraph, and that the last four steps from **Fix** to **Repeat** are iterated until a solution is found.

Let us here roughly describe the proposed algorithm. The **Multiply** step generates $I_{\leq D}$ of F by $T_{\leq D-2}$, defined in Section 2, by regarding each polynomial as that in $(\mathbb{F}_q[x_1, \ldots, x_k])[x_{k+1}, \ldots, x_n]$. By utilizing the property stated by Lemma 1, the **Linearize(1)** step described in Subsection 3.2 partly perform Gaussian elimination on $\mathcal{PM}$. After the **Linearize(1)** step, the resulting Macaulay matrix is supposed to be the following form $\begin{pmatrix} I & * \\ 0 & A \end{pmatrix}$, by interchanging rows (and columns). Here I is an identity matrix, and A is a matrix over $\mathbb{F}_q[x_1, \ldots, x_k]$. Then, the last four steps deal with only the submatrix A, and find a solution by the same way as h-XL. This submatrix A is called the *resulting matrix of* **Linearize(1)**.

3.2 Details of Linearize(1) step

In this subsection, we describe the details of the **Linearize(1)** step in the proposed algorithm, and show that it works well as row operations on $\mathcal{PM}$. We use the same notations as in Subsection 2. In the following, we also denote by $\mathcal{PM}[I_a, T_b]$ the same part even after $\mathcal{PM}$ is transformed.

The **Linearize(1)** step is mainly performed on each submatrices $\mathcal{PM}[I_d, T_d]$, $\mathcal{PM}[I_d, T_{(d-1)}]$, and $\mathcal{PM}[I_d, T_{(d-2)}]$, starting from $d = D$ down to 2. Each iteration d consists of the following three substeps:

(d)-1. Perform Gaussian elimination on $\mathcal{PM}[I_d, T_d]$.

(d)-2. Perform the same row operations as those of (d)-1 on $\mathcal{PM}[I_d, T_{(d-1)}]$ and $\mathcal{PM}[I_d, T_{(d-2)}]$.

(d)-3. Using the *leading coefficients* of the resulting $\mathcal{PM}[I_d, T_d]$, eliminate the corresponding columns of $\mathcal{PM}$. Here, a leading coefficient is the leftmost nonzero entry in each row of the row echelon form of a matrix.

Here, we show that the **Linearize(1)** step described above works well as row operations on $\mathcal{PM}$. Note that for any $3 \leq d \leq D$, the (d)-3 step does not affect the submatrix $\mathcal{PM}[I_{\leq(d-1)}, T_d]$, since $\mathcal{PM}[I_{\leq(d-1)}, T_d]$ is always a zero matrix by Lemma 1. This indicates that $\mathcal{PM}[I_d, T_{\leq D}]$ does not change from the original structure at the beginning of the (d)-1 step. Therefore, from Lemma 1, the manipulations in the (d)-1 and (d)-2 steps can be performed correctly and seen as row operations on $\mathcal{PM}$. Furthermore, the (d)-3 step can be also performed correctly, since the leading coefficients of the resulting $\mathcal{PM}[I_d, T_d]$ belong to $\mathbb{F}_q$. As a result, we have that all the manipulations are practicable and regarded as row operations on $\mathcal{PM}$.

After the **Linearize(1)** step, all manipulations are performed on the resulting matrix of **Linearize(1)** composed of rows and columns including no leading coefficient of the row echelon form $\mathcal{PM}[I_d, T_d]$ with $2 \leq d \leq D$.

Acknowledgements

This work was supported by JST CREST Grant Number JPMJCR2113, JSPS KAKENHI Grant Number JP21J20391, and JSPS Grant-in-Aid for Young Scientists 20K14301.

References

[1] Bettale, L., Faugère, J.-C., Perret, L.: Hybrid approach for solving multivariate systems over finite fields. Journal of Mathematical Cryptology **3**, pp. 177–197 (2009)

[2] Buchberger, B.: Ein algorithmus zum auffinden der basiselemente des restklassenringes nach einem nulldimensionalen polynomideal. PhD thesis, Universität Innsbruck (1965)

[3] Courtois, N., Klimov, A., Patarin, J., Shamir, A.: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: EUROCRYPT 2000, LNCS, vol. 1807, pp. 392–407. Springer (2000)

[4] Faugère, J.-C.: A new efficient algorithm for computing Gröbner bases (F4). Journal of Pure and Applied Algebra **139**(1-3), pp. 61–88 (1999)

[5] Faugère, J.-C.: A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In: ISSAC 2002, pp. 75–83. ACM (2002)

[6] Garey, M.-R., Johnson, D.-S.: Computers and intractability: a guide to the theory of NP-completeness. W. H. Freeman (1979)

[7] Kipnis, A., Shamir, A.: Cryptanalysis of the HFE public key cryptosystem by relinearization. In: CRYPTO 1999, LNCS, vol. 1666, pp. 19–30. Springer (1999)

[8] Yang, B.-Y., Chen, J.-M., Courtois, N.: On asymptotic security estimates in XL and Gröbner bases-related algebraic cryptanalysis. In: ICICS 2004, LNCS, vol. 3269, pp. 401–413. Springer (2004)

Graduate School of Information Science and Technology, The University of Tokyo
The University of Tokyo
Tokyo 113-8656
JAPAN
E-mail address: `furue-hiroki261@g.ecc.u-tokyo.ac.jp`
東京大学・情報理工学系研究科　古江弘樹

Graduate School of Information Science and Technology, The University of Tokyo
The University of Tokyo
Tokyo 113-8656
JAPAN
E-mail address: `kudo@mist.i.u-tokyo.ac.jp`
東京大学・情報理工学系研究科　工藤桃成