Computer Security Symposium 2022
24 - 27 October 2022

Gröbner基底計算のためのMXL4アルゴリズムの提案

五太子 政史[1,a)]　辻井 重男[1]

概要：多変数公開鍵暗号など，暗号解読を行うための Gröbner 基底計算では汎用的な F4 等が主流だが，ペアからの S-多項式計算を行わない XL 法も改良次第で効率的な計算が期待できる．本稿では，以前提案された MXL3 の欠点を改良した MXL4 を考案し，その性能評価を行ったところ，$m = 2n$ 程度の overdetermined の連立方程式では F4 を凌ぐ計算速度が得られることが確認された．

キーワード：耐量子計算機暗号, 多変数公開鍵暗号, Gröbner 基底, XL 法

Proposal of Gröbener Basis Computing Algorithm MXL4

Masahito Gotaishi[1,a)]　Shigeo Tsujii[1]

Abstract: While Gröbner Bases are computed by the generic F4, which computes them by generating S-polynomials, XL algorithms, which do not compute S-polynomials also have advantage. Here a new Gröbner Bases computing algorithm, MXL4, is proposed. This algorithm was found to outperform F4 for overdetermined polynomial systems such as $m = 2n$.

Keywords: Post-Quantum Cryptography, Multivariate Public Key Cryptography, Grb̈ner Bases, XL algorithm

1. はじめに

1.1 非線形多変数連立方程式を用いた暗号方式とその解析

非線形多変数連立方程式 (次数 2 以上の連立代数方程式) の求解は計算困難 (NP 完全) であることが証明されており [1]，かつ量子コンピュータでも効率的に求解するアルゴリズムは未だ発見されていない．このことを利用した多変数公開鍵暗号 (Multi-variate Public-Key Cryptography, MPKC) は耐量子計算機暗号の候補の一つとして研究が進められており，そのうちで Unbalanced Oil & Venegar 方式 (UOV) に基づいた Rainbow[2] は NIST の Post-Quantum Cryptography 標準化の候補に上がったことがあり，また，同じく UOV に基づいた署名方式も 2021 年に提案されている [3].

多変数公開鍵暗号は Matsumoto-Imai 暗号 [4] に始まり，その後多数の秘匿用暗号方式が提案されたが，このほぼ全てが，変数の数 n と多項式数 m がほぼ等しい determined であった．しかしその後 2010 年代以降に発表された方式は，Simple Matrix[5] を初め，

[1] 中央大学研究開発機構
Research and Development Initiative, Chuo University
[a)] gotaishi@tamacc.chuo-u.ac.jp

overdetermined のものが多数を占めており，多変数連立方程式求解ののチャレンジである Fukuoka MQ Challenge[6] の課題は $m = 2n$ の overdetermined になっている．

この多変数非線形連立方程式は，暗号などに使われる場合は殆どが二次方程式 (MQ) であり，この方程式を解く方法としてはそれら多項式集合を生成元とする ideal の Gröbner 基底を求めることが一般的である．MPKC の評価で最初に行うことは，公開鍵の MQ 多項式集合について Gröbner 基底計算を行うときの計算量が落とし戸構造を持たない多項式集合の場合と同じか (S-多項式の最高次数が同じか) を検証することであり，出来上がる暗号方式についてパラメータを決めるにも Gröbner 基底計算の計算量 (time/space complexity) を知ることが必要であり，そのために Gröbner 基底計算アルゴリズムの効率追求が活発に行われている．

2. 背景

2.1 用語及び基本的概念

本稿で用いる用語，及び概念を以下に列挙して説明する：

(1) 項順序
多項式を扱う場合，項の並び順を決める項順序が重要である．これには主に以下のようなものがある：
- 全次数辞書式： 次数が高い順に並べるが，同じ次数では順序の高い変数のある項が先になる．1 番目の変数をまず比較し，それが同じであれば 2 番目,,, という順に比較してゆく．例えば，$x_1x_6^2 \succ x_2x_3x_4$ 等.
- 全次数逆辞書式：全次数辞書式とは逆に，順序の低い変数のある項が後になる．先ほどの項の順序例では，は全次数逆辞書式では逆に $x_1x_6^2 \prec x_2x_3x_4$ となる．

(2) 頭項 (leading term): 多項式 f の項の中で最も項順序の高いものを言う．$LT(f)$ などと表す.

(3) 頭変数 (leading variable)：項の中で最も順序の高い変数を言う．項 $x_3x_4^2x_5$ の leading variable は x_3 である．本稿では「多項式の leading variable」という用語も用いる．ある多項式 f について，$t := LT(f)$ の頭変数が v であるとき，v を「多項式 f の leading variable」と呼ぶ.

(4) semi-regular sequence: ideal 要素同士に trivial なもの以外は線形関係が存在しないもの．則ち，互いに線形独立な任意の ideal 要素 f_1, f_2 に対して線形独立な多項式 g_1, g_2 を掛けた場合，$f_1 * g_1$ と $f_2 * g_2$ が線形独立 (trivial な線形関係を除く) である，ということの成立つものが semi-regular sequence である．trivial な線形関係とは，$f_if_j - f_jf_i = 0$，及び $f_i^2 - f_i = 0$ というようなものを言う．
MPKC の落とし戸構造を持つ多項式集合では trivial 以外の線形関係が成り立つ場合が多い.

(5) degree of regularity: 高い次数の ideal 要素を生成してゆくと，最終的にはその次数の項よりも多項式の方が多くなる．すると，ある次数 D の要素の線形結合が次数 D 未満になるという現象が現れる．そのようなことの起こる最低次数 D が degree of regularity (D_{reg}) である.

(6) Mutant: 上記の degree of regularity に達したら，D 次の ideal 要素の線形結合を取ったら D より低い次数の多項式が現れることになる．このような ideal 要素を Mutant と呼ぶ.

2.2 Gröbner 基底計算法-Buchberger アルゴリズムから $F4/F5$ まで

Gröbner 基底計算法として最初に考案されたのは Buchberger アルゴリズムと呼ばれるもので，ある多項式のペア f_i, f_j について以下のような S-多項式を計算するものである．下式で，f_i, f_j の頭項を t_i, t_j, $f_i - t_i := f_i'$, $f_j - t_j := f_j'$, $(LCM(t_i, t_j) = u_i * t_i = u_j * t_j)$ として，

$$\begin{aligned} S(f_i, f_j) &:= u_i * f_i - u_j - f_j \\ &= u_i * t_i - u_j * t_j + (u_i * f_i' - u_j * f_j') \\ &= u_i * f_i' - u_j * f_j' \end{aligned} \quad (1)$$

で求められる $S(f_i, f_j)$ が f_i と f_j の S-多項式である．このようにして求めた S-多項式は t_i や t_j では割り切れない頭項を持つことになるが，それを割り切る頭項を持つ他の要素を探し，その多項式で割った剰余を求めてゆく (単項簡約)．このようなことを，他の要素で簡約できない多項式が得られるまで繰り返すのが Buchberger アルゴリズムで，この単項簡約を行列のガウス消去によってまとめて行う方式が $F4$ である．この S-多項式を用いる方法では全次数逆辞書式の項順序を使うのが最も効率的に計算できるとされる．

これらの方式で問題となるのは，殆どの S-多項式は単項簡約の結果 0 にしかならないことである．“90% of the (CPU) times is spent computing zero”[7] と Faugère が述べており，彼らは 0 に簡約されない S-多項式のみを生成しながら Gröbner 基底計算を行う $F5$[8] を発表した．

2.3 XL 法

eXtended Linearization (XL) 法 [9] とは，非線形多項式の各項を変数と見なして線形方程式として解く方法である．生成元に変数を掛けてゆくことによって多数の多項式を得られるが，次数が D_{reg} に達すると，得られる多項式の方が項の数よりも大きくなる．その次数まで，生成元に変数を掛けていって ideal 要素を生成し，得られた多項式集合を係数行列と項ベクトルの積の形で表し，その係数行列をガウス消去して基底を求めてゆく．生成元集合 $F = \{f_1, f_2, \ldots, f_m\}$ の要素が全て 2 次式であれば，それら全てに全ての変数を掛けてゆくと 3 次式集合 $\{x_i f_1, x_2 f_1, \ldots, x_n f_1, x_1 f_2, \ldots, x_n f_m\}$ が得られる．このようにして，4 次，5 次,，，と次数を上げて row echelon form を求めてゆくものである．

XL 法の優位点は以下の 2 点である：

(1) S-多項式を求める際のペアの生成と評価を行わずに済むこと
次数 $(D+1)$ の S-多項式を計算するには，次数 D の全 ideal 要素 k 個についてペアを作成し，そのそれぞれについて頭項の最小公倍数を求めなければならない．この計算量は $\mathcal{O}(\log D)$ で，それをペアの数だけ行うので $\mathcal{O}(k^2 \log D)$ である．ペア選定基準はあるが，その場合でもペアの評価は行わなければならない．XL 法では $\mathcal{O}(kn)$ で良い．

(2) 低次の多項式に変数を掛けて高次の多項式を得て Gauss 消去を行う際に (変数行列×項ベクトル) の形のままで処理できるためメモリ効率が良いこと

XL 法でガウス消去の結果得られる D 次多項式集合は，D 次以下の全ての ideal 要素の張る線形空間の基底であり，D-Gröbner 基底と呼ばれる [10].

2.4 Mutant Strategy による XL 法

Mutant を得ることが方程式求解の鍵であり，Mutant を得るのに最少限の ideal 要素のみを生成してゆくように考えられた XL 法が Mutant XL(MXL3) である [11]．この方式では，項ベクトルの並び順を全次数辞書式とする．XL 法で得られた D 次多項式集合の echelon form では，x_1, x_2 など順序の高い変数ではそれを Leading Variable とする全ての項についてそれを頭項に持つ要素が存在する．d 次の ideal 要素についてそのような条件を満足する変数の集合を $FLV(F,d)$(Full Leading Variable), $\mathrm{Min}(FLV(F),d)$ を $LFLV(F,d)$, 多項式集合 F で次数 d の要素のうち leading variable が v 以降であるものの集合を $LVT(F,d,v)$ と定義する．次数 d でこのような結果が得られたならば，$LFLV(F,d)$ 以上の変数を頭項に持つ多項式は全て簡約可能であり，それより低い変数を頭項に持つ多項式を生成するようにする戦略を取るべきであろう．この方針を Mohamed らは「Partial Enlargement」と呼んでいる．

d 次で row echelon form の D-ideal 要素全ての中から $v := LFLV(F,d)$ に対して多項式集合 $LVT(F,d,v)$ を取り，それらに v 以下の変数のみを掛けて $(D+1)$ 次多項式集合を得る (Partial Enlargement)．このようにして生成される多項式は，$(D+1)$ 次部分に v 以降の変数のみを含むものに限られ，次数が上がるたびに多項式数が増えてゆくのを抑止できる．

MXL3 を発表した論文 [11] では，26 変数 determined の random 多項式の場合について Magma の $F4$ 関数 (GroebnerBasis) で係数行列の最大が 298592×148804 で計算に 3325 秒かかるところ，MXL3 では 88513×102246 で計算が 1429 秒，31 変数 determined では $F4$ で行列 868614×489702 で計算時間 162118 秒のところ MXL3 では 415654×436598, 計算時間 94191 と計算速度・メモリ消費共に向上したことを報告している．

この方式に対しては Albrecht[12] らが，

(1) 生成される多項式数が大きいならば変数行列を分けてガウス消去すれば済むことで大きな優位点にはならない．

(2) $LVT(F,d,v)$ の多項式に v 以降の変数を掛けて $LVT(F,d+1,v)$ の要素が全て生成されるという保証はなく，MXL3 は不完全である

という批判を述べている．(2) についてはその通りで，実は方程式が overdetermined の場合は ideal 要素の中の一部しか生成されない場合がある (第 3.2 節

参照).

しかし MXL で「生成される多項式が少なくて済む」とは即ち「0 に簡約される多項式が少ない」ということであり，これはかなり大きな優位点である．Mohamed らが性能を評価した多項式集合は determined であり，この場合は検証した限りでは全ての ideal 要素が生成されており，非常に効率的な方式と言える．

3. 提案方式

今までに提案された方式は，F4 も MXL も必要最少限の ideal 要素のみを生成しながら Mutant を得ることを目指すものだが，F4/Buchberger では S-多項式を生成しうる全てのペアを評価しなければならないという点が非効率である．この点を改善すべきである．

3.1 D-Gröbner 基底の数

目指すべきは，D 次の要素について，$(D+1)$-Gröbner 基底が全て得られるまで各要素に変数を掛けて多項式を生成し，生成し終わったら直ちに止めて $(D+1)$ 次の要素へ移ることであるが，この終了点 (全て生成し終わったこと) の判定は可能である．生成元が semi-regular sequence である限り，D-Gröbner 基底の数は m, n, D の関数であり，実際に多項式を生成しなくとも求めることができる．これらの張る線形空間をを D-ideal と定義し，以下のようにして求める．

n 変数 m 連の semi-regular sequence F の δ-Gröbner 基底 ($\delta \leq D$．D は偶数とする) のうち次数 δ のものの集合 $A(D, F, \delta)$ を以下の部分集合に分ける：

$$B(D, F, \delta, h) := \{g \mid g = \eta_{2(h-i)} \prod_{i=1}^{h} f_{\nu(i)}\} \quad (2)$$
$$(h \leq D/2)$$

上記で，η_d は D-ideal(F) に含まれない任意の d 次式である．F が semi-regular sequence の場合，$\#B(deg, F, \delta) = \#B(D, f, \delta)$ $(\forall deg \leq D)$ が成り立つ．ここで，D-Gröbner 基底 $B(D, F, \delta, h)$ のうち次数 δ の多項式集合について，明らかに $\bigcup_{h=1}^{D/2} B(D, F, \delta, h) = A(D, F, \delta)$ であり，$B(D, F, \delta, h) \cup B(D, F, \delta, k) = \phi$ $(\forall h \neq k)$ である．ある次数 δ $(\delta \leq D)$ で D-ideal に含まれない多項式 η_δ の数は以下のようにして求められる．

$$\#\{\eta \mid TotalDegree(\eta) = \delta, \eta \notin B(D, F, \delta)\}$$
$$= \begin{pmatrix} n \\ \delta \end{pmatrix} - \#B(D, F, \delta) \quad (3)$$

以上のような手順で次数 δ の D-Gröbner 基底の数 (D-ideal の次元) を求めることが可能である：

$$\#A(D, F, D) = \sum_{h=0}^{D/2} \#B(D, F, D, h) \quad (4)$$

即ち，XL 計算を行っていくときには (1) 線形独立な多項式が上式で求めた数だけ生成される (2) Mutant が得られる のどちらかしか起こり得ない．

従って，次数 4 の D-Gröbner 基底の数は，例えば 30 変数 45 連であれば以下のように求められる：

$$\begin{pmatrix} 45 \\ 2 \end{pmatrix} + 45 \times \left\{ \begin{pmatrix} 30 \\ 2 \end{pmatrix} - 45 \right\}$$
$$= 990 + 17550 = 18540 \quad (5)$$

であり，次数 3 の D-Gröbner 基底に関しては，trivial な線形関係も存在しないので $30 \times 45 = 1350$ である．このとき，$x_1, \ldots, x_3$ を Leading Variable とする全ての 3 次項は頭項となり得るので，$x_1, \ldots, x_3$ を頭項に持つ多項式は全て簡約可能である．4 次で x_4 以降を Leading Variable に持つ要素は 8685 個あり，それらを得るために XL 法の計算を行う．まず，Leading Variable が x_3 以降である多項式に x_3 以降の変数を掛け row echelon form を求めて 4 次式を得る．これが最も変数行列を小さくできる多項式集合の取り方である．

こうして x_4 以降を Leading Variable に持つ多項式が 8685 個得られたら終了して 5 次式の生成に移る．得られなかったら x_2 を Leading Variable に持つ多項式も含め，元の多項式集合にも x_2 を掛けることによって更に多項式を生成する．

以上の操作を D_{reg} に達するまで続けてゆく．なお上の例では D_{reg} は 5 である．

3.2 MXL4 アルゴリズム

上記が本方式の概要だが，第 2.4 節で述べた通り，例えば $m = 2n$ の overdetermined では 4 次の ideal 要素を生成するとき，明らかに $MXL3$ の Partial Enlargement では生成する ideal 要素数が不足する (表 1).

表 1: Overdetermined の場合の Partial Enlargement

n=28, m=56			
Degree	$\#F_d$	LFLV	LV が LFLV である要素の数
3	1568	5	316
4	20475		9723
n=30, m=60			
3	1800	5	340
4	24270		11815

表 1 にある通り，$n = 28, m = 56$ の場合は 3 次の LFLV である x_5 以降を頭項に持つ多項式は 316 であり，これらに $x_5, \ldots, x_{28}$ の全ての変数を掛けて得られる多項式が全て線形独立だったとしても $316 \times 24 = 7584$ で，実際に生成してみるまでもなく，必要な数の多項式は得られないことが判る．$n = 30$ の場合も同様である．

そこで，Partial Enlargement を開始するに当たっては，まず，必要数の多項式を生成し得る最低限の多項式集合を取ることにする．$v \succ LFLV(F, d)$ であり，$\#FVT(F, d, v) \times (n - v + 1)$ が必要多項式数以上となる v を求め，$FVT(F, d, v)$ に対して partial enlargement を行う．$d + 1$ が D_{reg} の場合は，$\#FVT(F, d, v) \times (n - v + 1)$ が $\begin{pmatrix} n - v + 1 \\ d + 1 \end{pmatrix}$ 以上となる v を求める．

Algorithm 1 に MXL4 のアルゴリズムを示す．このアルゴリズムでは生成元 F の要素は全て 2 次式とし，3 次の要素については全て線形独立であることが判っているため 3 次式生成のステップでは全ての要素に全ての変数を掛けてガウス消去する．本稿では，ideal(F) の D_{reg} が 4 以上であると仮定している．

Algorithm 1 MXL4 アルゴリズム

Input: m 個の n 変数多項式から成る MQ 生成元 F
Output: F の生成するイデアルの Gröbner 基底 G

begin
$Mu \leftarrow \phi; G \leftarrow F; v \leftarrow x_1; d \leftarrow 2$
$\tilde{G} \leftarrow G \cup \{f * v | f \in F, v \in x\}$
while $Mu = \phi$ **do**
  Increment d
  $v \leftarrow FLV(G_{=d}, d)$
  while $\#LVT(G_{=d}, d, v) \times (n - v + 1) < \#LVT(G_{=d+1}, d + 1, v)$ **do**
    Increment v;
  end while
  Compute $\#LVT(G_{=d+1}, d + 1, v)$
  while $\#\tilde{G}_+ < \#LVT(G_{=d+1}, d + 1, v)$ **do**
    $G_+ \leftarrow \{f * u | f \in LVT(G_{=d}, d, v), u \preceq v\}$
    $G_+ \leftarrow \tilde{G}_+$
    if Exists f such that TotalDegree(f) $< d + 1$) and $f \notin G$ **then**
      $Mu \leftarrow \{f \mid$ TotalDegree(f) $< d + 1$) and $f \notin G\}$
      Break; // Mutants were found
    end if
    Increment v
  end while
  $G \leftarrow G \cup G_+$
end while
Reduce G with Mu until Reduced Gröbner bases are obtained.
end

4. 実験結果

4.1 *D*-Gröbner 基底の数

実際に XL 法によって D-Gröbner 基底を生成し，理論値との一致を確認した．

表 2: D-Gröbner Base の理論値

次数	多項式数	次数	多項式数
n=24, m=24		n=25, m=27	
4	6324	4	7722
5	41376	5	52650

4.2 MXL4 による連立方程式求解

上記のアルゴリズムを Magma[13] のスクリプトに実装し，Magma の乱数で各係数を生成することによって発生させた連立方程式について求解を行った．計算を行った条件は，Lenovo L480 SSD 480GB, RAM 20GB, Windows 11, Windows 版 Magma(32 bit) V2.11-13 である．

試したところ，やはり Partial Enlargement を行う際の多項式集合及び変数は初期値のままでは終了しない．

乱数で係数を生成させた多項式の集合を 27 変数 50 連，及び 28 変数 56 連について Gröbner 基底計算を行い，途中で生成される多項式数と実行時間を比較した結果を表 3.2 に示す．残念ながら Magma の F4 計算関数 (GroebnerBasis) ではガウス消去実行後の多項式数が出力されないため，F4 での計算結果では，各次数で生成される多項式数のみを記録している．

表 3: MXL4 及び F4 の Gröbner 基底計算の比較

n=27, m=50			
次数	Partial Enlargement		計算時間
	初期値	最終値	(秒)
4	$x_4 - x_{27}$	$x_3 - x_{11}$	45.937
多項式数	12176 → 10748	13690 → 12775	
5	$x_{12} - x_{23}$		73.734
多項式数	15520 → 15520		
計算終了			77.359
F4 計算			
4	18270		5.646
5	131722		77.828
計算終了			78.093
n=28, m=56			
次数	Partial Enlargement		計算時間
	初期値	最終値	(秒)
4	$x_4 - x_{28}$	$x_3 - x_{11}$	64.093
多項式数	16424 → 13368	16669 → 12775	
5	$x_{14} - x_{24}$		86.312
多項式数	16658 → 16658		
計算終了			90.203
F4 計算			
4	21715		8.640
5	163970		102.406
計算終了			102.765

上記の表に見る通り，次数が D_{reg} である 5 に達したところで，MXL 法で生成される多項式は F4 と比べて圧倒的に少ない数で終わっている．このため Magma の内部関数である F4 計算よりもスクリプト言語で書いた MXL4 のプログラムの方が速いという驚くべき結果となった．この差が D_{reg} における計算終了 (今回の実装では次数 D_{reg} の全ての項について，その項を頭項とする多項式が生成されたことを以て終了としている) の判断が速いためか，次数が高くなるほど生成される多項式が少なくて済むためかは，今回試験した規模の多項式集合では定かでない．

5. 今後の展望

筆者らの環境では現在，計算機能力の問題で，D_{reg} が 6 程度までしか試すことができない．このため，本方式については，40 変数以上の大きな連立方程式について検証ができていない．今後は環境を整備した上で更に動作検証を行うと共に，有料のプラットフォームを必要としない Python/SAGE 等に実装を行って Fukuoka MQ Challenge 等への求解なども行ないたいと考えている．

本方式を開発するに当たって検討を進めて感じたことであるが，まだまだ非線形多変数連立方程式の求解方法は「濡れ雑巾」であり，効率向上の余地は十分に残っていると感じる．例えば $F4$ などに対する本方式の優位点は主に，全次数辞書式の優位点と言える．従って S-多項式を使う方法についても本方式と同じような改良を加えることによって更に効率の高い計算法の得られる可能性があり，検討すべき点はまだ多数残されている．

6. 結論

XL 法の非効率性は既に指摘されている [10][12] が，元々 $F4$ や Buchberger アルゴリズムは，暗号解読で出てくる overdetermined の多変数連立方程式を解くことを目的としたものでは無く汎用的に使えることを意図したものであるため，その点に由来する欠点もある．通常の暗号解読で使われる多項式集合を想定して最適化を行うべきであろう．

謝辞 本稿に報告した内容は令和 4 年度より (一財) テレコム先端技術研究支援センター (SCAT) から助成を受けている研究課題「マイナンバー・STR(DNA) を秘密鍵に内蔵する 3 層型公開鍵暗号の提案」の成果の一部です．ご支援下さった同法人の皆様，有難

うございました.

参考文献

[1] Buss, J. F., Frandsen, G. S. and Shallit, J. O.: The Computational Complexity of Some Problems of Linear Algebra, *J. Comput. Syst. Sci.*, Vol. 58, No. 3, pp. 572–596 (online), DOI: 10.1006/jcss.1998.1608 (1999).

[2] Ding, J. and Schmidt, D.: Rainbow, a New Multivariable Polynomial Signature Scheme, *Applied Cryptography and Network Security* (Ioannidis, J., Keromytis, A. and Yung, M., eds.), Lecture Notes in Computer Science, Vol. 3531, Springer Berlin / Heidelberg, pp. 164–175 (2005).

[3] Furue, H., Ikematsu, Y., Kiyomura, Y. and Takagi, T.: A New Variant of Unbalanced Oil and Vinegar Using Quotient Ring: QR-UOV, *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, pp. 187–217 (2021).

[4] Matsumoto, T. and Imai, H.: A class of asymmetric crypto-systems based on polynomials over finite rings, *IEEE International Symposium on Information Theory*, St. Jovite, Quebec, Canada, IEEE (1983).

[5] Tao, C., Xiang, H., Petzoldt, A. and Ding, J.: Simple Matrix-A Multivariate Public Key Cryptosystem (MPKC) for Encryption, *Finite Fields and Their Applications*, Vol. 35, No. C, pp. 352–368 (2015).

[6] Yasuda, T., Dahan, X., Huang, Y., Takagi, T. and Sakurai, K.: MQ Challenge: Hardness Evaluation of Solving Multivariate Quadratic Problems, *IACR Cryptol. ePrint Arch.*, p. 275 (online), available from ⟨http://eprint.iacr.org/2015/275⟩ (2015).

[7] Faugère, J.-C.: A new efficient algorithm for computing Gröbner bases (F4), *Journal of Pure and Applied Algebra*, Vol. 139, No. 1–3, pp. 61–88 (online), DOI: DOI: 10.1016/S0022-4049(99)00005-5 (1999).

[8] Faugère, J. C.: A new Efficient Algorithm for computing Gröbner bases without reduction to zero (F5), *ISSAC '02: Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, New York, NY, USA, ACM, pp. 75–83 (online), DOI: http://doi.acm.org/10.1145/780506.780516 (2002).

[9] Courtois, N. T., Klimov, A., Patarin, J. and Shamir, A.: Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding* (Preneel, B., ed.), Lecture Notes in Computer Science, Vol. 1807, Springer, pp. 392–407 (2000).

[10] Sugita, M., Kawazoe, M. and Imai, H.: Relation between the XL algorithm and Grobner basis algorithms, *IEICE transactions on fundamentals of electronics, communications and computer sciences*, Vol. 89, No. 1, pp. 11–18 (2006).

[11] Mohamed, M. S. E., Cabarcas, D., Ding, J., Buchmann, J. and Bulygin, S.: MXL3: An Efficient Algorithm for Computing Gröbner Bases of Zero-Dimensional Ideals, *Information, Security and Cryptology – ICISC 2009* (Lee, D. and Hong, S., eds.), Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 87–100 (2010).

[12] Albrecht, M. R., Cid, C., Faugere, J.-C. and Perret, L.: On the relation between the MXL family of algorithms and Gröbner basis algorithms, *Journal of Symbolic Computation*, Vol. 47, No. 8, pp. 926–941 (2012).

[13] Bosma, W., Cannon, J. and Playoust, C.: The Magma Algebra System I: The User Language, *Journal of Symbolic Computation*, Vol. 24, No. 3-4, pp. 235–265 (1997).