Magma のグレブナ基底計算を利用した暗号解析
～多変数公開鍵暗号に対する代数攻撃～

藤田　亮
中央大学研究開発機構

2010 年 11 月 6 日

謝辞

研究集会「Magma で広がる数学の世界」における講演の機会を与えてくださった，本研究集会のオーガナイザーを務められた木田雅成教授（電気通信大学），ならびに，松尾和人教授（情報セキュリティ大学院大学）に感謝いたします．

本研究における成果の多くは，辻井重男教授，只木孝太郎准教授，五太子政史専任研究員（中央大学研究開発機構）との共同研究によるものです．また，本研究における計算機実験環境においては，松尾教授，只木准教授，五太子研究員によるご協力と，中央大学 21 世紀 COE プログラム「電子社会の信頼性向上と情報セキュリティ」ならびに，平成 20 年度総務省戦略的情報通信研究開発推進制度（SCOPE）ICT イノベーション創出型研究開発 ICT 安心・安全技術（量子コンピュータの出現に対抗し得る公開鍵暗号の研究）の援助を受けました．この場をお借りして，御礼申し上げます．

1　はじめに

グレブナ基底は，1960 年代，多項式環のイデアルに関する問題を解くために導入された概念である．グレブナ基底の理論および応用は，非常に多岐にわたっており，特に，暗号の分野において，多変数非線形連立方程式の解を求めるために，しばしば，グレブナ基底計算が用いられる．とりわけ，多変数公開鍵暗号と呼ばれる公開鍵暗号に対しては，暗号解析や安全性解析において，グレブナ基底計算を行うために，Magma が利用されることが多い．

本文では，多変数公開鍵暗号に対する暗号解析のための一手法として広く知られている，代数攻撃について取り上げ，そのために提案されている，さまざまな方法について紹介する．また，代数攻撃を利用した，多変数公開鍵暗号の安全性解析において，これまでに行われている研究成果について述べる．

本文の構成は以下の通りである．

まず 2 節では，有限体上の多変数非線形連立方程式の求解について，具体的に Magma を利用して解く例を提示し，グレブナ基底計算が実際にどのように行われるかについて述べる．

続く 3 節では，多変数非線形連立方程式の求解困難性に，その安全性を求める，多変数公開鍵暗号について説明する．多変数公開鍵暗号は，1980 年代に，日本が発祥の公開鍵暗号であり，多くの文献 [Kob98, Kob99, DGS06b, TK08, BBD09] に取り上げられているように，近年，注目を集めている，量子コンピュータの出現に対抗し得る公開鍵暗号の候補の一つとして考えられている暗号方式である[*1]．

4 節では，多変数公開鍵暗号に対する，代表的な攻撃手法として知られている，代数攻撃について述べる．Magma のグレブナ基底計算は，多変数公開鍵暗号に対する代数攻撃のための強力なツールとして利用可能である．一方で，Magma を利用する以外に，代数攻撃のために数多くの手法が，これまでに提案されている．本文では，これらの手法について簡単に紹介する．

5 節では，代数攻撃の手法を利用した，多変数公開鍵暗号の安全性解析において，これまでに得られている結果について述べる．多変数公開鍵暗号の安全性を強化するための一手法として提案されている持駒方式について，代数攻撃に対する安全性を評価する際に，

[*1] その他の候補としては，線形符号の復号問題に基づく暗号方式，格子に関する問題に基づく暗号方式，ハッシュ関数の安全性に基づく暗号方式などが考えられている [BBD09].

Magma のグレブナ基底計算を用いている．

6 節では，まとめと今後の研究課題について述べる．

1.1 記法について

本文では，ベクトルについて，たとえば $\boldsymbol{p}$, $\boldsymbol{X}$ といったようにボールド体を用いる．

- $\circ$: 合成写像.
- $\mathbf{F}_q$: GF(q)，すなわち元の数が q の有限体.
- $(\mathbf{F}_q)^n = \left\{(\alpha_1, \ldots, \alpha_n)^T \mid \alpha_i \in \mathbf{F}_q, i = 1, \ldots, n\right\}$: n 次元 $\mathbf{F}_q$ ベクトル空間.
- ^{T} : ベクトルの転置.
- $\mathbf{F}_q[x_1, \ldots, x_n]$: 係数を $\mathbf{F}_q$ に持つ，$x_1, \ldots, x_n$ を変数とする多項式全体の集合（多項式環）.
- ϕ: $\mathbf{F}_{q^l} \to (\mathbf{F}_q)^l$: $\mathbf{F}_{q^l}$ の多項式基底 $\omega_1, \ldots, \omega_l$ を用いた以下の写像：

$$\begin{aligned} \phi(x_1\omega_1 + x_2\omega_2 + \cdots + x_l\omega_l) = \ \phi(X) \ &= \quad (x_1, x_2, \ldots, x_l)^T \quad = \boldsymbol{x}, \\ \phi^{-1}((x_1, x_2, \ldots, x_l)^T) \quad &= \phi^{-1}(\boldsymbol{x}) = x_1\omega_1 + x_2\omega_2 + \cdots + x_l\omega_l = X. \end{aligned}$$

  例えば，$q = 2$, $l = 3$ の場合に，多項式基底を $\omega_1 = 1$, $\omega_2 = \omega$, $\omega_3 = \omega^2$ として，$\omega^3 + \omega + 1$ などのような，ω に関する l 次の $\mathbf{F}_q$ 上既約多項式 $f(\omega)$ に対し，$\mathbf{F}_{q^l} = \mathbf{F}_q[\omega]/f(\omega)$ を考えると，ベクトル空間 $(\mathbf{F}_q)^l$ と拡大体 $\mathbf{F}_{q^l}$ は ϕ を通じて同一と見ることができる.
- $x \in_U A$: 要素 x は，集合 A から一様な確率でランダムに選んだものである.

1.2 計算機実験環境

本文における計算機実験において，下記の環境 A，または，環境 B を使用した．

計算機実験環境 A

- コンピュータ：PROSIDE edAEW416R2 workstation
- プロセッサ：2.80GHz AMD Opteron Model 854 プロセッサ
- メモリ：64GB RAM
- 数式処理ソフトウェア：Magma V2.12-21
- グレブナ基底計算アルゴリズム：F_4
- 項順序（単項式順序）：全次数逆辞書式順序（DRL; grevlex）

計算機実験環境 B

- コンピュータ : 日本コンピューティングシステム（JCS）VC98220WSA-4U/T workstation
- プロセッサ：2.80GHz AMD Opteron 8220 クアッドコアプロセッサ
- メモリ：128GB RAM
- 数式処理ソフトウェア：Magma V2.15-5
- グレブナ基底計算アルゴリズム：F_4
- 項順序（単項式順序）: 全次数逆辞書式順序（DRL; grevlex）

2　有限体上の多変数非線形連立方程式の求解

例えば，係数体を $\mathbf{F}_7$ とする，下記の方程式 (2.1) の解を求めることを考える．

$$\begin{cases} 3x^2+xy+2xz+6y^2+4yz+5z^2+6x+4y+5z=3 \\ 6x^2+6xy+xz+3y^2+yz+3z^2+2x+y+4z=5 \\ 5x^2+2xy+3xz+3y^2+4yz+5z^2+6x+5y+z=3 \end{cases} \tag{2.1}$$

方程式 (2.1) は，一般に，$\mathbf{F}_q[x_1,\ldots,x_n]$ の要素である m 個の多項式 $e_1(x_1,\ldots,x_n),\ldots,e_m(x_1,\ldots,x_n)$ と，m 次元ベクトル $\boldsymbol{c}=(c_1,c_2,\ldots,c_m)^T\in(\mathbf{F}_q)^m$ を用いて，以下の式 (2.2) のように表すことができる．

$$\begin{cases} e_1(x_1,\ldots,x_n)=c_1 \\ e_2(x_1,\ldots,x_n)=c_2 \\ \qquad\vdots \\ e_m(x_1,\ldots,x_n)=c_m \end{cases} \tag{2.2}$$

このような方程式を解くための準備として，まず，方程式 (2.2) を以下のような形に変形する．

$$\begin{cases} f_1=e_1(x_1,\ldots,x_n)-c_1=0 \\ f_2=e_2(x_1,\ldots,x_n)-c_2=0 \\ \qquad\vdots \\ f_m=e_m(x_1,\ldots,x_n)-c_m=0 \end{cases} \tag{2.3}$$

方程式 (2.1) について，ここまでを，Magma を用いて計算する[*2] ．

[*2] 方程式 (2.1) に関する一連の計算に使用した Magma のバージョンは V2.12-17 である．

```
> q := 7; // 有限体の位数
> n := 3; // 変数の個数
> m := 3; // 式の数
> G := GF(q); // 有限体 G を定義する
> R<x,y,z> := PolynomialRing(G,n); // G 上の n 変数多項式環 R を定義する
> E := [R |
> 3*x^2 +   x*y + 2*x*z + 6*y^2 + 4*y*z + 5*z^2 + 6*x + 4*y + 5*z,
> 6*x^2 + 6*x*y +   x*z + 3*y^2 +   y*z + 3*z^2 + 2*x +   y + 4*z,
> 5*x^2 + 2*x*y + 3*x*z + 3*y^2 + 4*y*z + 5*z^2 + 6*x + 5*y +   z
> ];
> c := [G | 3, 5, 3];
> F := [E[i] - c[i] : i in [1..m]];
> F;
[
    3*x^2 + x*y + 2*x*z + 6*x + 6*y^2 + 4*y*z + 4*y + 5*z^2 + 5*z + 4,
    6*x^2 + 6*x*y + x*z + 2*x + 3*y^2 + y*z + y + 3*z^2 + 4*z + 2,
    5*x^2 + 2*x*y + 3*x*z + 6*x + 3*y^2 + 4*y*z + 5*y + 5*z^2 + z + 4
]
```

なお，方程式に用いられる変数 x, y, z を，例えば `u[1]`, `u[2]`, `u[3]` といったように表示するには，下記のように，多項式環を定義すればよい．

```
> Ru<[u]> := PolynomialRing(G,n);
> Ru;
Polynomial ring of rank 3 over GF(7)
Order: Lexicographical
Variables: u[1], u[2], u[3]
> Ru ! F[1]; // F[1] を Ru の元として表す
3*u[1]^2 + u[1]*u[2] + 2*u[1]*u[3] + 6*u[1] + 6*u[2]^2 + 4*u[2]*u[3] + 4*u[2]
         + 5*u[3]^2 + 5*u[3] + 4
```

多項式環における項順序については後述するが，Magma において，項順序を何も指定しない場合，辞書式順序（lexicographic order）として定義される．

2.1 XL アルゴリズム

方程式 (2.3) を解く方法として，いくつか考えられるが，ここでは，まず，XL（eXtended Linearization）と呼ばれる，以下に示すアルゴリズム [CKPS00] を用いて解くことを考える．

1. ある次数 d 以下のすべての単項式を各多項式 $f_1, \ldots, f_m$ に乗じ，その結果からなる多項式集合（ここでは LargeF と表す）を生成する．
2. 1. において生成された LargeF の各要素である，多項式のなかに現れる単項式を，それぞれ別個の変数とみなし，各多項式における係数からなる行列（ここでは CoefMat と表す）を掃き出す．ただし，項順序は，ある変数（ここでは x_i とする）が，最後に消去されるような順序とする．
3. 2. において，x_i に関する一変数多項式が得られたものとする．この多項式の零点を求めることにより，その変数 x_i の値を得る．
4. 3. において求めた変数を消去することにより，方程式を簡単化し，他のすべての変数の値が得られるまで 1. 2. 3. を繰り返す．

Magma を利用し，方程式 (2.1) を XL アルゴリズムによって解いてみる．

```
> // **********************
> // *****  Step 1.  *****
> // **********************
> d := 5;
> LinPoly := &+[R.i : i in [1..Rank(R)]] + 1;
> dPoly := (LinPoly)^d;
> MonosSeq := Monomials(dPoly); // d 次以下のすべての単項式からなる Sequence
> LargeF := [];
> for i in [1..m] do
for>    for j in [1..#MonosSeq] do
for|for>                Include(~LargeF, F[i] * MonosSeq[j]);
for|for>        end for;
for> end for;
> // **********************
> // *****  Step 2.  *****
> // **********************
> PickUpMonos := [];
> for i in [1..#LargeF] do
for> // MonosLargeF は LargeF[i] に含まれる単項式からなる Sequence
for>    MonosLargeF := Monomials(LargeF[i]);
for>    for j in [1..#MonosLargeF] do
for|for>                Include(~PickUpMonos, MonosLargeF[j]);
for|for>        end for;
for> end for;
> MonosSet := SequenceToSet(PickUpMonos); // Set にして，重複する要素（単項式）を消す
> MonosSeq0 := SetToSequence(MonosSet); // Set から Sequence に戻す
> // MonosSeq0 の各要素の順序を並べ替える
> Apoly := &+[MonosSeq0[i] : i in [1..#MonosSeq0]];
> MonosSeq := Monomials(Apoly); // 項順序に従って，Sequence の要素の順番を並べ替える
> CoefSeq := [[BaseRing(Parent(LargeF[1])) | 0 : i in [1..#MonosSeq]] : i in [1..#LargeF]];
```

```
> for i in [1..#LargeF] do
for>    for j in [1..#MonosSeq] do
for|for>                CoefSeq[i][j] := MonomialCoefficient(LargeF[i],MonosSeq[j]);
for|for>        end for;
for> end for;
> CoefMat := Matrix(CoefSeq); // LargeF[i] の各多項式における，単項式の係数からなる行列
> Sweep := EchelonForm(CoefMat); // 行列 CoefMat を掃き出す
> TailPoly := &+[MonosSeq[i] * Sweep[Rank(Sweep)][i] : i in [1..#MonosSeq]];
> TailPoly;
z^7 + 3*z^6 + 5*z^5 + 6*z^3 + 3
```

以上，方程式 (2.1) に対し，XL アルゴリズムにおける 1. 2. を行うことにより，z に関する一変数多項式 (2.4) を得ることができた．

$$z^7 + 3z^6 + 5z^5 + 6z^3 + 3 \tag{2.4}$$

Magma を用いて得られた式 (2.4) は，多変数多項式環の元となっており，例えば，`Roots` コマンドなどを使って，直接，零点を求めようとするとエラーとなる．そこで，`UnivariatePolynomial` コマンドを用いて一変数多項式環上の多項式に移し変えてから解いてみる．

```
> // *********************
> // *****  Step 3.  *****
> // *********************
> UniTailPoly := UnivariatePolynomial(TailPoly);
> Factorization(UniTailPoly);
[
    <$.1 + 4, 1>,
    <$.1^2 + 2, 1>,
    <$.1^4 + 6*$.1^3 + $.1 + 3, 1>
]
> Roots(UniTailPoly); // TailPoly の零点を求める.
[ <3, 1> ]
```

以上により，変数 z の値が 3 と求めることができた．以下，方程式から z を消去し，残りの変数 x, y を求めてゆくが，ここでは省略する．

なお，XL アルゴリズムに関する理論的な解析や計算量については，以下に述べるグレブナ基底計算と関連して，文献 [CP03, YC04a, YCC04, YC04b] などに述べられている．また，本文において取り上げる，多変数公開鍵暗号に対する代数攻撃以外に，共通鍵暗号に対する攻撃手法としての XL アルゴリズムについては，文献 [CP02, Cou02] などに記述されている．

2.2 グレブナ基底計算

XL アルゴリズムにより，方程式 (2.3) における多項式 $f_1, \ldots, f_m$ から，一変数多項式を得ようとする操作は，実は，多項式環 $\mathbf{F}_q[x_1, \ldots, x_n]$ における，$f_1, \ldots, f_m$ を基底とするイデアル

$$I = \langle f_1, \ldots, f_m \rangle = \{g_1 f_1 + \cdots + g_m f_m \mid g_1, \ldots, g_m \in \mathbf{F}_q[x_1, \ldots, x_n]\} \tag{2.5}$$

を考え，この I における要素を，くまなく探していることになる．つまり，各 g_i として，まず，d 次以下の単項式を考え，これらを各 f_i に乗じたもの同士の和を，順次，取ってゆくことにより，I の要素を探す操作を行っている．

式 (2.5) のようなイデアル I の基底は一意でないが，それらのうち，性質のよいものの一つとして，グレブナ基底と呼ばれる基底が知られている．なお，ここでは，グレブナ基底の概念や理論などについて詳述しない．これらについては，これまでに数多くの良書が出版されているので，それらを参照されたい（例えば [CLO00, KR00, NY03, KR05, Hib06, CLO07] など）．

さて，グレブナ基底を計算するアルゴリズムとして，いくつかのアルゴリズムが，これまでに提案されているが，Magma には，Buchberger によるアルゴリズム [Buc65]，Faugère による F_4 アルゴリズム [Fau99] が実装されている．特に F_4 アルゴリズムは，XL アルゴリズムよりも，グレブナ基底の計算において，効率的となり得ることが示されている [AFIKS04]．

方程式 (2.1) の解空間をなす，式 (2.5) のようなイデアル I について，Magma を利用して，グレブナ基底を計算する．

```
> I := ideal<R | [F[i] : i in [1..m]]>; // 各 F[i] を基底とするイデアル
> GroebnerBasis(I); // I のグレブナ基底を求める
[
    x + 3*y + 2*z^6 + 2*z^5 + 5*z^4 + 3*z^3 + 5*z^2 + 4*z,
    y^2 + 5*z^5 + 3*z^3 + z^2 + 2*z + 3,
    y*z + 4*y + 5*z^6 + 2*z^5 + 3*z^4 + 4*z^3 + 2*z^2 + 4*z + 3,
    z^7 + 3*z^6 + 5*z^5 + 6*z^3 + 3
]
```

方程式 (2.1) に関する，ここまでの計算では，項順序を特に設定していなかったため，辞書式順序としてグレブナ基底の計算を行っている．辞書式順序のグレブナ基底は三角形式，すなわち，一変数ずつ解いてゆくことができるような形となることが知られている．

グレブナ基底の計算過程を考えず，Magma を利用して，単に，イデアルの零点を求めるだけであれば，以下のように計算することも可能となっている．

```
> I := ideal<R | [F[i] : i in [1..m]]>;
> Variety(I); // I の零点を求める
[ <0, 4, 3>, <3, 3, 3> ]
>
> VarietySequence(I); // Sequence として解を出力する
[
    [ 0, 4, 3 ],
    [ 3, 3, 3 ]
]
```

2.2.1 項順序

グレブナ基底を計算する際，計算効率に大きく影響を及ぼすのが，項順序である．

Magma では，代表的な項順序として，辞書式順序（lexicographic order），全次数辞書式順序（degree（graded）lexicographic order），全次数逆辞書式順序（Degree（Graded）Reverse Lexicographic order: DRL（grevlex））を利用することができる[*3]．一般に，全次数逆辞書式順序によるグレブナ基底計算が高速であり，辞書式順序による計算は，さほど速くないと考えられている．

```
> R1<x,y,z> := PolynomialRing(GF(7),3,"lex"); // 辞書式順序
> R1;
Polynomial ring of rank 3 over GF(7)
Lexicographical Order
Variables: x, y, z
> R2<x,y,z> := PolynomialRing(GF(7),3,"glex"); // 全次数辞書式順序
> R2;
Polynomial ring of rank 3 over GF(7)
Graded Lexicographical Order
Variables: x, y, z
> R3<x,y,z> := PolynomialRing(GF(7),3,"grevlex"); // 全次数「逆」辞書式順序
> R3;
Polynomial ring of rank 3 over GF(7)
Graded Reverse Lexicographical Order
Variables: x, y, z
```

[*3] これらの項順序以外では，消去順序，ブロック順序，重み付き順序などの項順序が利用できる．

例えば，$f(x,y,z) = (x+y+z+1)^3$ は，これらの項順序に基づいて，下記のように表される．

```
> f := (x+y+z+1)^3;
> R1 ! f; // 辞書式順序
x^3 + 3*x^2*y + 3*x^2*z + 3*x^2 + 3*x*y^2 + 6*x*y*z + 6*x*y + 3*x*z^2
    + 6*x*z + 3*x + y^3 + 3*y^2*z + 3*y^2 + 3*y*z^2 + 6*y*z + 3*y
    + z^3 + 3*z^2 + 3*z + 1
> R2 ! f; // 全次数辞書式順序
x^3 + 3*x^2*y + 3*x^2*z + 3*x*y^2 + 6*x*y*z + 3*x*z^2 + y^3 + 3*y^2*z
    + 3*y*z^2 + z^3 + 3*x^2 + 6*x*y + 6*x*z + 3*y^2 + 6*y*z + 3*z^2
    + 3*x + 3*y + 3*z + 1
> R3 ! f; // 全次数「逆」辞書式順序
x^3 + 3*x^2*y + 3*x*y^2 + y^3 + 3*x^2*z + 6*x*y*z + 3*y^2*z + 3*x*z^2
    + 3*y*z^2 + z^3 + 3*x^2 + 6*x*y + 3*y^2 + 6*x*z + 6*y*z + 3*z^2
    + 3*x + 3*y + 3*z + 1
```

2.2.2 表示オプション

Magma のコマンド `SetVerbose` によって値を設定することにより，グレブナ基底の計算過程を表示させることができる．設定する数値が大きくなるにつれて，計算過程のより詳細を表示させることができるようになっている．

```
> // グレブナ基底の計算過程を表示する（0: 表示しない; 1, 2, 3, 4: 表示する）
> SetVerbose("Groebner",4);
> GetVerbose("Groebner"); // 設定されている値を出力する
4
```

その他，グレブナ基底計算に関しては `Buchberger`, `Faugere` と，グレブナ基底の項順序を変換するアルゴリズムである `FGLM` [FGLM93], `GroebnerWalk` [CKM97] にも表示オプションがあり，それぞれのアルゴリズムにおいて，計算過程の表示について設定することができる．

2.2.3 計算アルゴリズム

Magma のデフォルトでは，実装されている F_4 アルゴリズム [Fau99] を利用して，グレブナ基底計算を行っている．

例えば，方程式 (2.1) を解くために行ったグレブナ基底計算について，Magma の F_4 アルゴリズムによる計算過程を表示すると，下記のようになる．

```
> I := ideal<R | [F[i] : i in [1..m]]>;
> SetVerbose("Groebner",1); // グレブナ基底の計算過程を表示する
> GroebnerBasis(I);
Homogeneous weights search
Number of variables: 3, nullity: 0
********************
FAUGERE F4 ALGORITHM
********************
Coefficient ring: GF(7)
Rank: 3
Order: Graded Reverse Lexicographical
NEW hash table
Matrix kind: Modular FP
Datum size: 4
Initial length: 3
Inhomogeneous

*******
STEP 1
Basis length: 3, queue length: 2, step degree: 2, num pairs: 2
Basis total mons: 30, average length: 10.000
Number of pair polynomials: 2, at 10 column(s), 0.000
Average length for reductees: 10.00 [2], reductors: 10.00 [1]
Symbolic reduction time: 0.000, column sort time: 0.000
2 + 1 = 3 rows / 10 columns, 100% / 100% (10/r)
Before ech memory: 3.5MB
Row sort time: 0.000
0.000 + 0.000 = 0.000 [2]
After ech memory: 3.5MB
Queue insertion time: 0.000
Step 1 time: 0.000, [0.010], mat/total: 0.000/0.000 [0.010], mem: 3.5MB

*******
STEP 2
Basis length: 5, queue length: 2, step degree: 3, num pairs: 2
Basis total mons: 46, average length: 9.200
Number of pair polynomials: 2, at 15 column(s), 0.000
Average length for reductees: 8.00 [2], reductors: 8.67 [9]
Symbolic reduction time: 0.000, column sort time: 0.000
2 + 9 = 11 rows / 19 columns, 44.976% / 62.959% (8.5455/r)
Before ech memory: 3.5MB
Row sort time: 0.000
0.000 + 0.000 = 0.000 [2]
After ech memory: 3.5MB
Queue insertion time: 0.000
Step 2 time: 0.000, [0.000], mat/total: 0.000/0.000 [0.010], mem: 3.5MB
```

```
*******
STEP 3
Basis length: 7, queue length: 4, step degree: 4, num pairs: 4
Basis total mons: 64, average length: 9.143
Number of pair polynomials: 4, at 20 column(s), 0.000
Average length for reductees: 9.00 [4], reductors: 8.77 [13]
Symbolic reduction time: 0.000, column sort time: 0.000
4 + 13 = 17 rows / 22 columns, 40.107% / 58.743% (8.8235/r)
Before ech memory: 3.5MB
Row sort time: 0.000
0.000 + 0.000 = 0.000 [1]
After ech memory: 3.5MB
Queue insertion time: 0.000
Step 3 time: 0.000, [0.000], mat/total: 0.000/0.000 [0.020], mem: 3.5MB

*******
STEP 4
Basis length: 8, queue length: 2, step degree: 5, num pairs: 2
Basis total mons: 71, average length: 8.875
Number of pair polynomials: 2, at 18 column(s), 0.000
Average length for reductees: 7.00 [2], reductors: 8.57 [14]
Symbolic reduction time: 0.000, column sort time: 0.000
2 + 14 = 16 rows / 22 columns, 38.068% / 58.035% (8.375/r)
Before ech memory: 3.5MB
Row sort time: 0.000
0.000 + 0.000 = 0.000 [0]
After ech memory: 3.5MB
Queue insertion time: 0.000
Step 4 time: 0.000, [0.010], mat/total: 0.000/0.000 [0.030], mem: 3.5MB

Reduce 8 final polynomial(s) by 8
0 redundant polynomial(s) removed; time: 0.000
Interreduce 6 (out of 8) polynomial(s)
Symbolic reduction time: 0.000
6 + 0 = 6 rows / 14 columns, 60.714% / 80.52% (8.5/r)
Row sort time: 0.000
0.000 + 0.000 = 0.000 [6]
Interreduction time: 0.000
Final number of polynomials: 8

Number of pairs: 10
Total pair setup time: 0.000
Max num entries matrix: 17 by 22
Max num rows matrix: 17 by 22
Total symbolic reduction time: 0.000
Total column sort time: 0.000
Total row sort time: 0.000
```

```
Total matrix time: 0.000
Total new polys time: 0.000
Total queue update time: 0.000
Total Faugere F4 time: 0.000, real time: 0.040
*****************
FGLM ORDER CHANGE
*****************

Coefficient ring: GF(7)
Rank: 3
Initial order: Graded Reverse Lexicographical
Final order: Lexicographical
Basis length: 6
Dimension: 8
New polynomial 0, leading monomial: z^7
New polynomial 1, leading monomial: y*z
New polynomial 2, leading monomial: y^2
New polynomial 3, leading monomial: x
Total FGLM time: 0.000
[
    x + 3*y + 2*z^6 + 2*z^5 + 5*z^4 + 3*z^3 + 5*z^2 + 4*z,
    y^2 + 5*z^5 + 3*z^3 + z^2 + 2*z + 3,
    y*z + 4*y + 5*z^6 + 2*z^5 + 3*z^4 + 4*z^3 + 2*z^2 + 4*z + 3,
    z^7 + 3*z^6 + 5*z^5 + 6*z^3 + 3
]
```

F_4 アルゴリズムによるグレブナ基底計算は，V2.11 以降の Magma に導入され，開発者のホームページにおいて，その性能について記載されている [Ste04]．当時，開発されていた他のソフトウェアとの性能比較などが行われており，非常に高速にグレブナ基底計算を行うことが報告されている．その後，Magma におけるグレブナ基底計算アルゴリズムの改良が行われている一方で，最近の結果では，F_4 アルゴリズムの改良版である F_5 アルゴリズム [Fau02] と，行列演算部分を改良したパッケージを利用して，グレブナ基底計算のベンチマークとして，しばしば用いられる Katsura-n 方程式 [KFSM83, KFIFG87] の求解において，$\mathbf{F}_{65521}$ 上の Katsura-16 方程式を 1 時間半程度かけて解いたという報告がなされている [FL10] [*4]．このアルゴリズムが，本文執筆時（2010 年 11 月）における，世界最速のグレブナ基底計算アルゴリズムと見られる．

F_4 アルゴリズムではなく，従来のアルゴリズムである，Buchberger アルゴリズムを用いて計算する場合には，以下のように入力すればよい．

[*4] 複数の CPU を使用した並列計算を，アルゴリズムに組み込むことによって，高速化を図っているものと思われるが，Magma V2.16-1 を利用した場合と比較して，数十倍から数百倍，高速であるという結果が示されている．

```
> I := ideal<R | [F[i] : i in [1..m]]>;
> SetVerbose("Groebner",0);
> SetVerbose("Buchberger",4); // Buchberger アルゴリズムだけを詳しく表示させる
> GroebnerBasis(I: Faugere := false); // F4 アルゴリズムを使わずに計算する
********************
BUCHBERGER ALGORITHM
********************

Coefficient ring: GF(7)
Rank: 3
Order: Graded Reverse Lexicographical
Initial length: 3
Using monomial division list
Initial Reduce: true
Remove Redundant: true
New Reduce: false
Final Reduce: true
Homogenization: false
Initial basis:
[
    3*x^2 + x*y + 6*y^2 + 2*x*z + 4*y*z + 5*z^2 + 6*x + 4*y + 5*z + 4,
    6*x^2 + 6*x*y + 3*y^2 + x*z + y*z + 3*z^2 + 2*x + y + 4*z + 2,
    5*x^2 + 2*x*y + 3*y^2 + 3*x*z + 4*y*z + 5*z^2 + 6*x + 5*y + z + 4
]
Reduce initial basis

Insert 0:
    x*y + 6*x*z + 6*y*z + 4*z^2 + 2*x + 2*y + 6*z + 6
    Total degree: 2

Insert 1:
    y^2 + 3*x*z + 5*y*z + z^2 + 2*x + 4*y + z + 1
    Total degree: 2

Insert 2:
    x^2 + 2*x*z + y*z + 3*z^2 + 2*x + 2*y + 2
    Total degree: 2

Reduce pairs

1 [0, 1] (0 done): Degree: 3, lcm = x*y^2, 0.000

Insert 3:
    x*z^2 + 3*y*z^2 + 3*z^3 + 4*x*z + 2*y*z + 5*x + 3*y + 2*z
    Total degree: 3

2 [0, 2] (1 done): Degree: 3, lcm = x^2*y, 0.000
```

```
Insert 4:
    y*z^2 + 4*z^3 + 5*x*z + 5*y*z + 5*z^2 + 6*x + 4*y + 6*z + 4
    Total degree: 3

3 [1, 4] (2 done): Degree: 4, lcm = y^2*z^2, 0.000

Insert 5:
    z^4 + 6*z^3 + x*z + 5*y*z + 4*x + 6*y + 3*z
    Total degree: 4

4 [0, 3] (3 done): Degree: 4, lcm = x*y*z^2, 0.000
3 [0, 4] (4 done): Degree: 4, lcm = x*y*z^2, 0.000
2 [2, 3] (5 done): Degree: 4, lcm = x^2*z^2, 0.000
1 [4, 5] (6 done): Degree: 5, lcm = y*z^4, 0.000
0 [3, 5] (7 done): Degree: 5, lcm = x*z^4, 0.000
Reduce final polynomials
Reduce 0
Reduce 1
Reduce 2
Reduce 3
Reduce 4
Reduce 5
Reduction time: 0.000
Number of pairs reduced: 8
Total Buchberger time: 0.000
[
    x + 3*y + 2*z^6 + 2*z^5 + 5*z^4 + 3*z^3 + 5*z^2 + 4*z,
    y^2 + 5*z^5 + 3*z^3 + z^2 + 2*z + 3,
    y*z + 4*y + 5*z^6 + 2*z^5 + 3*z^4 + 4*z^3 + 2*z^2 + 4*z + 3,
    z^7 + 3*z^6 + 5*z^5 + 6*z^3 + 3
]
```

グレブナ基底の計算過程に関する出力から，Magma では，グレブナ基底計算において，項順序として，全次数逆辞書式順序のみを用いて計算しているものと見られる．一般に，この順序による計算が高速と考えられているが，文献 [Saw02] などにおいて報告されているように，全次数逆辞書式順序による計算が，必ずしも高速であると限らない．解く問題によっては，非常に非効率な項順序となってしまうことも考えられ得る．また，F_4 アルゴリズムの提案者によって，その後，提示された F_5 アルゴリズム [Fau02] などのように，グレブナ基底の計算過程における無駄な計算をなくすことにより，Magma の F_4 アルゴリズムを利用するよりも，効率的に計算を行うことができたという報告がなされている [GT08b, MCDBB09].

Magma はオープンソースでないため，グレブナ基底計算について，どのような実装が

なされているか，窺い知ることができない．しかしながら，ここにいくつか挙げた改善点のように，Magma のグレブナ基底計算が効率化される可能性は，まだ残されていると考えられる．

2.3 問題の困難性

方程式 (2.2) を解くような，有限体上の多変数非線形連立方程式の求解は，NP 困難であることが知られている [GJ79, p.251]．実際に，計算機実験環境 A（1.2 節）のもとで，方程式 (2.2) について，$q = 2, n = m$ とし，方程式の次数を 2 とした場合に，グレブナ基底を計算するために要した時間を表 1 に示す．

表 1　連立 2 次方程式を解くためのグレブナ基底計算時間（係数体 $\mathbf{F}_2$, 変数の数 = 式の数）

n：変数の数	17	18	19	20	21	22	23	24	25	26	27	28	29
計算時間（秒）	1	2	4	8	16	54	85	577	1185	2516	5274	11521	22236
メモリ使用量（MB）	26	40	62	93	136	337	472	3065	5341	8733	13591	20423	30026

表 1 からわかるように，変数の数が増えるにつれて，計算時間，および，計算に必要なメモリは，指数的に増大する．ただし，この結果は，変数の数 n と，式の数 m が等しい場合であり，そうでない場合について，グレブナ基底計算以外の方法もまた，考えられ得る．上に述べた XL アルゴリズムは，本来，特に，式の数 m の方が，変数の数 n よりも非常に多い，すなわち $n \ll m$ である場合に，有効となるものと考えられたアルゴリズムである．また，これとは逆に $m \ll n$ である場合，有効となるアルゴリズムが，いくつか提案されている [CGMT02, Has09]．

さて，こうした NP 困難な問題は，現用のコンピュータのみならず，量子状態を利用して，高度な並列計算を行うことができる量子コンピュータを用いたとしても，解くことが困難であると強く信じられている．一方で，素因数分解や離散対数問題などといった，現用の公開鍵暗号である，RSA 暗号や楕円曲線暗号の安全性の根拠となっている問題は，量子コンピュータを用いて，容易に解かれてしまう [Sho94, BL95]．このような背景から，量子コンピュータの出現に対抗し得る公開鍵暗号の研究が盛んに行われており，その候補の一つとして，多変数非線形連立方程式の求解困難性に，その安全性を求める，多変数公開鍵暗号がある．

表 2　主要な多変数 2 次公開鍵暗号

	暗号系	Modifier を 適用した暗号系	Modifier
MI 型	MI [MI88a]	C^{*--} [PGC98] SFLASH [CGP03]	–
		PMI [Din04]	i
		PMI+ [DG06]	i, +
	HFE [Pat96a]	HFE$^-$ [Pat96a]	–
		QUARTZ [PCG01]	v, –
		IPHFE [DS05a]	i
順序解法型	順序解法 [TKIFM86]	Birational Permutation Scheme [Sha93]	–
	R(S)SE [KS04, KS05a]	RSSE$^-$ [KS05b]	–
UOV 型	UOV [KPG99]		

–: Minus method, +: Plus method, v: Vinegar 変数, i: Internal Perturbation

3　多変数公開鍵暗号

多変数公開鍵暗号（Multivariate Public Key Cryptosystem: MPKC）とは，MI 暗号 [MI88a] や，順序解法に基づく公開鍵暗号 [TKIFM86] などのように，公開鍵が，平文変数，および，暗号文変数を変数とする多項式のタプルとして表される公開鍵暗号である．

多変数公開鍵暗号のうち，公開鍵多項式 $e_i \in \mathbf{F}_q[x_1, ..., x_n]$ の次数が 2，すなわち，平文変数 $x_1, \ldots, x_n$ を変数とする 2 次多項式タプルによって表され，暗号文変数 y_i について $y_i = e_i(x_1, ..., x_n)$ といった形をなすものを，ここでは，多変数 2 次公開鍵暗号と呼ぶ．

これまでに提案されている，主要な多変数 2 次公開鍵暗号について，表 2 に要約する．

3.1　多変数 2 次公開鍵暗号

多変数 2 次公開鍵暗号とは，n 変数からなるベクトルを平文変数ベクトル $\boldsymbol{x} = (x_1, \ldots, x_n)^T$，$m$ 変数からなるベクトルを暗号文変数ベクトル $\boldsymbol{y} = (y_1, \ldots, y_m)^T$ とし，線形変換 L_1, L_2 と非線形変換 G を秘密鍵，

$$\boldsymbol{y} = E(\boldsymbol{x}) = (L_1 \circ G \circ L_2)(\boldsymbol{x})$$

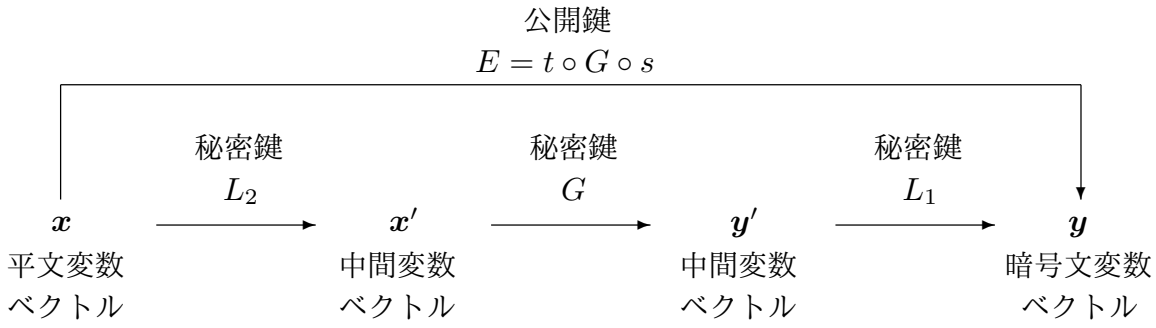

図 1　多変数 2 次公開鍵暗号

なる暗号化変換 E を公開鍵とする公開鍵暗号である（図 1）．$\boldsymbol{y}$ から $\boldsymbol{x}$ への復号変換は

$$\boldsymbol{x} = (L_2^{-1} \circ G^{-1} \circ L_1^{-1})(\boldsymbol{y})$$

となる．秘密鍵 G の違いにより，これまでにさまざまな多変数 2 次公開鍵暗号が提案されている．

多変数 2 次公開鍵暗号において，各公開鍵多項式は以下のような形をなす：

$$y_i = \sum_{1 \le j,l \le n} \alpha_{i,j,l} x_j x_l + \sum_{1 \le j \le n} \beta_{i,j} x_j + \gamma_i.$$

これらの方程式の各係数について，$\alpha_{i,j,l}$, $\beta_{i,j}$, $\gamma_i \in \mathbf{F}_q$ である．

一方，以下のように，公開鍵多項式の全次数が 3 となる，Dragon 型多変数公開鍵暗号が提案されている [Pat96b].

$$\sum_{\substack{1 \le j_1, j_2 \le n \\ 1 \le l \le m}} \alpha_{i,j_1,j_2,l}\ x_{j_1} x_{j_2} y_l + \sum_{1 \le j,l \le n} \beta_{i,j,l} x_j x_l$$
$$+ \sum_{\substack{1 \le j \le n \\ 1 \le l \le m}} \gamma_{i,j,l}\ x_j y_l + \sum_{1 \le j \le n} \delta_{i,j} x_j + \sum_{1 \le l \le m} \varepsilon_{i,l} y_l + \xi_i = 0.$$

これらの方程式の各係数について，$\alpha_{j_1,j_2,l}$, $\beta_{i,j,l}$, $\gamma_{i,j,l}$, $\delta_{i,j}$, $\varepsilon_{i,l}$, $\xi_i \in \mathbf{F}_q$ である．

3.1.1　暗号系

- パラメータ：
  - q : 暗号系を構成する有限体の位数.
  - n : 平文（ベクトル）の次元，すなわち，平文変数 x_i の個数.
  - m : 暗号文（ベクトル）の次元，すなわち，公開鍵多項式 $e_i(\boldsymbol{x})$ の数.

- 平文：$\boldsymbol{p} = (p_1, p_2, \ldots, p_n)^T \in (\mathbf{F}_q)^n$.
- 暗号文：$\boldsymbol{c} = (c_1, c_2, \ldots, c_m)^T \in (\mathbf{F}_q)^m$.
- 秘密鍵：
  1. $L_1 : (\mathbf{F}_q)^m \to (\mathbf{F}_q)^m$：正則な線形変換.
  2. $L_2 : (\mathbf{F}_q)^n \to (\mathbf{F}_q)^n$：正則な線形変換.
  3. $G : (\mathbf{F}_q)^n \to (\mathbf{F}_q)^m$：2 次多項式タプルによって表される，
     逆変換が容易な非線形変換.

  ここでの線形変換の代わりとして，一般にはアフィン変換を用いる.
- 公開鍵：$\boldsymbol{y} = E(\boldsymbol{x}) = (e_1(\boldsymbol{x}), \ldots, e_m(\boldsymbol{x}))^T$
  $\Leftrightarrow \boldsymbol{y} = (L_1 \circ G \circ L_2)(\boldsymbol{x})$

  ここに，$e_i(\boldsymbol{x}) \in \mathbf{F}_q[x_1, \ldots, x_n]$ $(i = 1, \ldots, m)$.
- 暗号化：公開鍵の $\boldsymbol{x}$ に平文 $\boldsymbol{p}$ を代入し，$\boldsymbol{y}$ の値となる暗号文 $\boldsymbol{c}$ を得る.

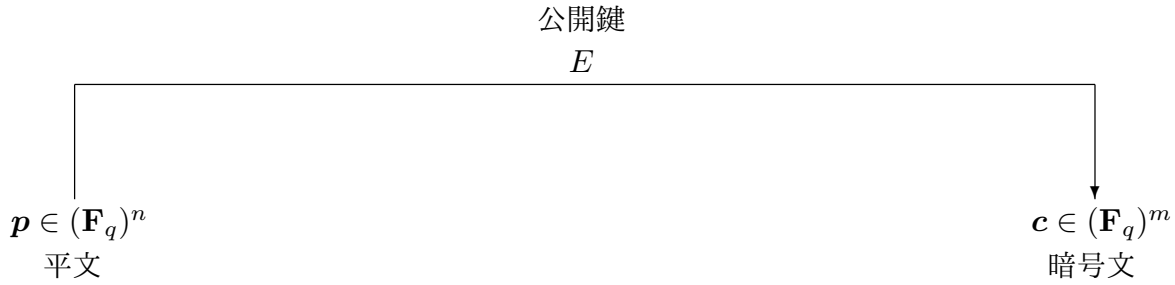

図 2　多変数 2 次公開鍵暗号系における暗号化

- 復号：
  1. $\boldsymbol{w} = L_1^{-1}(\boldsymbol{c})$.
  2. $\boldsymbol{v} = G^{-1}(\boldsymbol{w})$.
  3. $\boldsymbol{p} = L_2^{-1}(\boldsymbol{v})$.

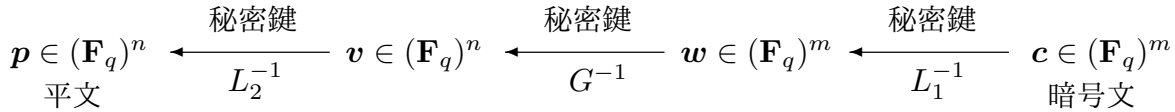

図 3　多変数 2 次公開鍵暗号系における復号

多変数 2 次公開鍵暗号系を用いた署名については，上記における平文を署名，暗号文を文書（ハッシュ値）とし，秘密鍵を署名鍵，公開鍵を検証鍵とする．また，秘密鍵を用いた復号が，署名者による署名鍵を用いた署名生成となり，署名検証は，文書 $\boldsymbol{c}$ と署名 $\boldsymbol{p}$ に

対し，検証鍵 E を用いて $\boldsymbol{c} = E(\boldsymbol{p})$ かどうか検査することとなる．

なお，図 1 に示した暗号系は秘密鍵 G について 1 段構成であるが，このような構成を拡張したものとして，非線形変換 G_1, G_2, 線形変換 L_1, L_2, L_3 について，公開鍵 $E = L_1 \circ G_1 \circ L_2 \circ G_2 \circ L_3$ といった 2 段構成をなし，公開鍵多項式の次数が 4 以上となる暗号系がいくつか提案されている [TFH89, PG97a, PG97b]．以下では，図 1 に示されるような，1 段構成の暗号系のみを取り上げる．

3.2 多変数 2 次公開鍵暗号の分類

多変数 2 次公開鍵暗号系の秘密鍵 G について，本文では，MI 型，順序解法型，UOV 型の 3 つに分類する．この分類法については，これまでにいくつか提案されているものの [WP05, DGS06b]，新たに提案されている暗号系について，こうした分類に含まれないものがある．そこで，本文では，上記のような，新しい分類を提案する．

以下では，表 2 に分類された，多変数 2 次公開鍵暗号の方式について説明する．これらの暗号方式のうち，いくつかの方式に関する Magma のプログラムが，多変数公開鍵暗号の解説書 [DGS06b] の著者らによる下記ホームページにて公開されている．

`http://math.uc.edu/~aac/MPKC/software.html`

3.2.1 MI 型

MI 型とは，中間変数ベクトル $\boldsymbol{v}$, $\boldsymbol{w}$ を拡大体の元 V, W として表現し，この拡大体上の単項式写像，あるいは多項式写像を非線形変換 G に用い，かつ，G^{-1} において，順序解法型に用いられるような計算を行わないものである（図 4）．

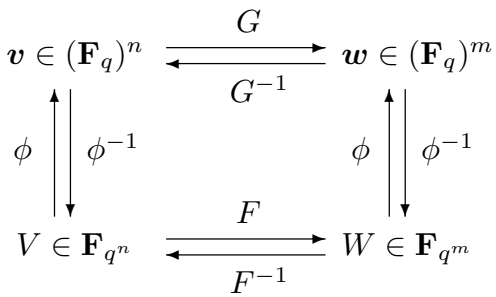

図 4　MI 型の非線形変換 G

■MI (Matsumoto-Imai; MIA, C^*) [MIHM83, IM85, MI88a, MI88b]　中間変数ベクトル $\boldsymbol{v}$, $\boldsymbol{w}$ のそれぞれの次元 n, m について $n = m$ とする．$V = \phi^{-1}(\boldsymbol{v}) \in \mathbf{F}_{q^n}$ に関する単項式写像 F を $F(V) = V^{q^\theta+1} = V^\iota$ とする．ただし $\mathrm{g.c.d.}(q^\theta + 1, q^n - 1) = 1$ とする．この条件は，F が全単射であることと同値である．

MI における G は以下のように表される：

$$G(\boldsymbol{v}) = (\phi \circ F \circ \phi^{-1})(\boldsymbol{v}).$$

法 $q^n - 1$ における ι の乗法の逆元を ι' とすると，$W = \phi^{-1}(\boldsymbol{w}) \in \mathbf{F}_{q^n}$ に対する F の逆変換 F^{-1} は以下のように表される：

$$F^{-1}(W) = W^{\iota'}.$$

F^{-1} を用いて，G の逆変換 G^{-1} は以下のように表される：

$$G^{-1}(\boldsymbol{w}) = (\phi \circ F^{-1} \circ \phi^{-1})(\boldsymbol{w}).$$

■HFE (Hidden Field Equations) [Pat96a]　中間変数ベクトル $\boldsymbol{v}$, $\boldsymbol{w}$ のそれぞれの次元 n, m について $n = m$ とする．$V = \phi^{-1}(\boldsymbol{v}) \in \mathbf{F}_{q^n}$ に関する d 次多項式写像 F が以下のような形をなすものとする：

$$F(V) = \sum_{\substack{0 \le i,j \le d \\ q^i + q^j \le d}} \beta_{i,j} V^{q^i+q^j} + \sum_{\substack{0 \le l \le d \\ q^l \le d}} \alpha_l V^{q^l} + \mu_0.$$

ここで，$\beta_{i,j}, \alpha_l, \mu_0 \in_U \mathbf{F}_{q^n}$ である．

HFE における G は以下のように表される：

$$G(\boldsymbol{v}) = (\phi \circ F \circ \phi^{-1})(\boldsymbol{v}).$$

HFE における G^{-1} は下記のように計算される：

1. $W = \phi^{-1}(\boldsymbol{w})$.
2. 1 において得られた W と未知変数 V に関する方程式 $F(V) = W$ を V について解く．
3. 2 において得られた V それぞれについて $\boldsymbol{v} = \phi(V)$ を得る．

3.2.2　順序解法型

順序解法型とは，G^{-1} を計算する際に，中間変数 $v_1, \ldots, v_n$ のうち，1 変数，あるいはいくつかの変数について順序的に解いてゆくものである．

■順序解法 [Tsu85, TKIFM86, Sha93]　順序解法における $G = (g_1, \ldots, g_n)$ は以下のように表される：

$$\begin{aligned}
w_1 &= g_1(v_1, \ldots, v_n) = v_1 \\
w_2 &= g_2(v_1, \ldots, v_n) = v_2 \cdot l_2(v_1) + q_2(v_1) \\
w_3 &= g_3(v_1, \ldots, v_n) = v_3 \cdot l_3(v_1, v_2) + q_3(v_1, v_2) \\
&\vdots \\
w_n &= g_n(v_1, \ldots, v_n) = v_n \cdot l_n(v_1, \ldots, v_{n-1}) + q_n(v_1, \ldots, v_{n-1})
\end{aligned}$$

ここに l_i は $v_1, \ldots, v_{i-1}$ に関する線形変換，q_i は $v_1, \ldots, v_{i-1}$ に関する非線形変換である．

順序解法 における G の逆変換 G^{-1} は以下のように計算される：

1. $v_1 = w_1$.
2. $v_2 = \dfrac{w_2 - q_2(v_1)}{l_2(v_1)}, v_3 = \dfrac{w_3 - q_3(v_1, v_2)}{l_3(v_1, v_2)}, \ldots, v_n = \dfrac{w_n - q_n(v_1, \ldots, v_{n-1})}{l_n(v_1, \ldots, v_{n-1})}$.

■R(S)SE (Random (Singular) Simultaneous Equations) [KS04, KS05a]　中間変数ベクトル $\boldsymbol{v}$, $\boldsymbol{w}$ のそれぞれの次元 n, m について $n = m$ とする．また，整数 $t \geq 2$, $N \geq 2$ に対し，$n = Nt$ をみたすものとする．

$\boldsymbol{v}$, $\boldsymbol{w}$ をそれぞれ N 個の t 次元ベクトルに分割したものを，それぞれ $\boldsymbol{v}_i$, $\boldsymbol{w}_i$ と表す．すなわち $\boldsymbol{v}_i = (v_{(i-1)t+1}, \ldots, v_{it})^T$, $\boldsymbol{w}_i = (w_{(i-1)t+1}, \ldots, w_{it})^T$ である．また $1 \leq i < j \leq N$ に対し $\boldsymbol{v}_{i,\ldots,j} = (v_{(i-1)t+1}, \ldots, v_{jt})^T$, $\boldsymbol{w}_{i,\ldots,j} = (w_{(i-1)t+1}, \ldots, w_{jt})^T$ とする．

F_i を逆変換が一意な（非特異な）非線形変換とし，Ψ_i を（必ずしも逆変換が一意とは限らない）非線形変換とする．

RSE における $G = (G_1, \ldots, G_N) : \boldsymbol{v} \mapsto \boldsymbol{w}$ は以下のように表される：

$$\begin{aligned}
\boldsymbol{w}_1 &= G_1(v_1, \ldots, v_n) = F_1(\boldsymbol{v}_1) \\
\boldsymbol{w}_2 &= G_2(v_1, \ldots, v_n) = F_2(\boldsymbol{v}_2) + \Psi_2(\boldsymbol{v}_1) \\
\boldsymbol{w}_3 &= G_3(v_1, \ldots, v_n) = F_3(\boldsymbol{v}_3) + \Psi_3(\boldsymbol{v}_{1,2}) \\
&\vdots \\
\boldsymbol{w}_N &= G_N(v_1, \ldots, v_n) = F_N(\boldsymbol{v}_N) + \Psi_N(\boldsymbol{v}_{1,\ldots,N-1})
\end{aligned}$$

RSSE における $G = (G_1, \ldots, G_N)$ は以下のように表される：

$$\begin{aligned}
\boldsymbol{w}_1 &= G_1(v_1, \ldots, v_n) = \Psi_1(\boldsymbol{v}_1) \\
\boldsymbol{w}_2 &= G_2(v_1, \ldots, v_n) = \Psi_2(\boldsymbol{v}_2) \\
\boldsymbol{w}_3 &= G_3(v_1, \ldots, v_n) = \Psi_3(\boldsymbol{v}_3) \\
&\vdots \\
\boldsymbol{w}_N &= G_N(v_1, \ldots, v_n) = \Psi_N(\boldsymbol{v}_N)
\end{aligned}$$

RSE における G^{-1} は以下のように計算される：

1. $\boldsymbol{v}_1 = F_1^{-1}(\boldsymbol{w}_1)$.
2. $\boldsymbol{v}_2 = F_2^{-1}(\boldsymbol{w}_2 - \Psi_2(\boldsymbol{v}_1)), \ldots, \boldsymbol{v}_N = F_N^{-1}(\boldsymbol{w}_N - \Psi_N(\boldsymbol{v}_{1,\ldots,N-1}))$.

RSSE における G^{-1} の計算は，各 i について $\boldsymbol{w}_i = \Psi_i(\boldsymbol{v}_i)$ となるような $\boldsymbol{v}_i$ の候補をそれぞれ求めることにより行う．

3.2.3 UOV 型

UOV 型とは，oil 変数と vinegar 変数による双線形形式を非線形変換 G として用いるものである．

■UOV (Unbalanced Oil and Vinegar) [KPG99] 中間変数ベクトル $\boldsymbol{v}$, $\boldsymbol{w}$ のそれぞれの次元 n, m について，任意の整数 $k \geq 1$ に対し $n = m + k$ であるものとする．

UOV における $G = (g_1, \ldots, g_m) : \boldsymbol{v} \mapsto \boldsymbol{w}$ は以下のように表される：

$$g_i = \sum_{\substack{1 \leq j \leq m \\ m+1 \leq l \leq m+k}} \gamma_{i,j,l} v_j v_l + \sum_{m+1 \leq j,l \leq m+k} \lambda_{i,j,l} v_j v_l + \sum_{1 \leq j \leq m} \xi_{i,j} v_j + \sum_{m+1 \leq j \leq m+k} \eta_{i,j} v_j + \delta_i.$$

これらの方程式の係数について，$\gamma_{i,j,l}$, $\lambda_{i,j,l}$, $\xi_{i,j}$, $\eta_{i,j}$, δ_i $\in_U$ $\mathbf{F}_q$ である．ここで $v_1, \ldots, v_m$ を oil 変数，$v_{m+1}, \ldots, v_{m+k}$ を vinegar 変数と呼ぶ．g_i には oil 変数同士の積，すなわち $1 \leq j, l \leq m$ に対する $v_j v_l$ の項が含まれていない．

UOV における G の逆変換 G^{-1} を計算するためには，g_i における vinegar 変数 $v_{m+1}, \ldots, v_{m+k}$ に値を与えたもの $\overline{g}_i(v_1, \ldots, v_m)$ について以下の線形連立方程式を $v_1, \ldots, v_m$ について解く：

$$\begin{cases}
\overline{g}_1(v_1, \ldots, v_m) &= w_1 \\
&\vdots \\
\overline{g}_m(v_1, \ldots, v_m) &= w_m
\end{cases}$$

UOV において，非線形変換 G はランダムに生成される．このため，秘密鍵 L_1 は非線形変換 G に吸収される．それゆえ，UOV において L_1 を考えないのが一般的である．

UOV のパラメータとしては，以下が提案されている：

- $q = 2$, $n = 192$, $m = 64$, $k = 128$.
- $q = 16$, $n = 48$, $m = 16$, $k = 32$.

3.3 多変数 2 次公開鍵暗号に対する Modifier

多変数 2 次公開鍵暗号に用いられる非線形変換 G は，その構造を攻撃に利用されやすい．そこで，こうした構造を破壊するように，多変数 2 次公開鍵暗号に対する Modifier が提案されている．

以下では，多変数 2 次公開鍵暗号に対する Modifier として提案されている，主要な手法について説明する．Modifier を適用した，非線形変換 G や公開鍵多項式による変換 E などを，チルダを付けて，それぞれ $\widetilde{G}$, $\widetilde{E}$ といったように表記する．

■Minus method [Sha93, Pat96a, PGC98, PCG01, CGP03]　Minus method とは，m 個の公開鍵多項式のうち, r 個 $(1 \leq r < m)$ を公開せず，残りの $(m - r)$ 個のみを公開する Modifier である（図 5）.

$$\left.\begin{array}{rcl} \widetilde{e}_1(x_1, \ldots, x_n) & = & e_1(x_1, \ldots, x_n) \\ \vdots & & \\ \widetilde{e}_{m-r}(x_1, \ldots, x_n) & = & e_{m-r}(x_1, \ldots, x_n) \end{array}\right\} \text{公開}$$

$$\left.\begin{array}{c} e_{m-r+1}(x_1, \ldots, x_n) \\ \vdots \\ e_m(x_1, \ldots, x_n) \end{array}\right\} \text{秘密}$$

図 5　Minus method

この r について，秘匿目的には小さな r しか使えない．なぜなら，暗号文 $\boldsymbol{c} = (c_1, \ldots, c_{m-r})^T$ から平文 $\boldsymbol{p} = (p_1, \ldots, p_n)^T$ を復号するためには, 欠落した r 個の要素 $c_{m-r+1}, \ldots, c_m$ について, q^r 通りの計算を行わなければならないためである.

一方で, 署名目的なら, この r について十分大きな値でも差し支えない. 特に, 攻撃者が文書 $\boldsymbol{c}$ から署名 $\boldsymbol{p}$ を偽造するのを手こずらせるためには, $r = 1, 2$ あるいは $r = m/2$ 程度が効果的とされている.

Minus method を適用した多変数 2 次公開鍵暗号としては，MI を原方式とする SFLASH などが提案されている．これらの方式のパラメータは以下の通りである．

- C^{*--} (MI minus): $q = 128 = 2^7$, $n = 67$, $r = 11$, $\theta = 33$ (SFLASH^{v3})
- HFE$^-$: $q = 16$, $n = 36$, $d = 4352$, $r = 4$ (HFE Challenge 2)

■Plus method [Pat96a, PGC98, DG06]　Plus method とは，m 個の公開鍵多項式と h 個のランダムな多項式について，これらの $(m + h)$ 個の多項式を線形変換したものを新たに公開鍵多項式とする Modifier である（図 6）．

$$\widetilde{E}(\boldsymbol{x}) = \widetilde{L}_1(e_1, \ldots, e_m, \varepsilon_1, \ldots, \varepsilon_h) \longleftarrow \left\{ \begin{array}{l} e_1(x_1, \ldots, x_n) \\ \quad \vdots \\ e_m(x_1, \ldots, x_n) \\ \left.\begin{array}{l} \varepsilon_1(x_1, \ldots, x_n) \\ \quad \vdots \\ \varepsilon_h(x_1, \ldots, x_n) \end{array}\right\} \text{ランダム多項式} \end{array} \right.$$

図 6　Plus method

この h について，署名目的には小さな h しか使えない．なぜなら，文書 $\boldsymbol{c} = (c_1, \ldots, c_{m+h})^T$ から生成された署名 $\boldsymbol{p} = (p_1, \ldots, p_n)^T$ に対して，$\boldsymbol{c} = \widetilde{E}(\boldsymbol{p})$ をみたす確率は $1/q^h$ であるためである．

■Vinegar 変数 [Pat96a, KPG99, PCG01]　Vinegar 変数とは，非線形変換 $G : \boldsymbol{v} \mapsto \boldsymbol{w}$ に導入する，中間変数 $\boldsymbol{v}$, $\boldsymbol{w}$ と独立な変数である．

MI 型の MPKC における非線形変換 $F : V \mapsto W$ が以下のような形をなすものとする：

$$F(V) = \sum_{0 \le i,j \le n} \beta_{i,j} V^{q^i + q^j} + \sum_{0 \le l \le n} \alpha_l V^{q^l} + \mu_0.$$

F に k 個の Vinegar 変数 $\boldsymbol{z} = (z_1, \ldots, z_k)^T$ を導入した非線形変換 $\widetilde{F}(V)$ は以下のよう

に表される：

$$\begin{aligned}
\widetilde{F}(V) &= \sum_{0\le i,j\le n} \beta_{i,j} V^{q^i+q^j} + \sum_{0\le l\le n} \eta_l(\boldsymbol{z}) V^{q^l} + \tau_0(\boldsymbol{z}),\\
\eta_l(\boldsymbol{z}) &= \sum_{1\le i\le k} z_i \lambda_{i,l} + \alpha_l,\\
\tau_0(\boldsymbol{z}) &= \sum_{1\le i\le j\le k} z_i z_j \varphi_{i,j} + \sum_{1\le i\le k} z_i \sigma_i + \mu_0.
\end{aligned}$$

ここで $\lambda_{i,l}, \varphi_{i,j}, \sigma_i \in_U \mathbf{F}_{q^n}$ である.

一方，MPKC における非線形変換 $G = (g_1, \ldots, g_m) : \boldsymbol{v} \mapsto \boldsymbol{w}$ が以下のような形をなすものとする：

$$\begin{aligned}
g_1(\boldsymbol{v}) &= \sum_{1\le j\le l\le n} \alpha_{1,j,l} v_j v_l + \sum_{1\le j\le n} \beta_{1,j} v_j + \gamma_1,\\
&\vdots\\
g_i(\boldsymbol{v}) &= \sum_{1\le j\le l\le n} \alpha_{i,j,l} v_j v_l + \sum_{1\le j\le n} \beta_{i,j} v_j + \gamma_i,\\
&\vdots\\
g_m(\boldsymbol{v}) &= \sum_{1\le j\le l\le n} \alpha_{m,j,l} v_j v_l + \sum_{1\le j\le n} \beta_{m,j} v_j + \gamma_m.
\end{aligned}$$

ここに $\alpha_{i,j,l}, \beta_{i,j}, \gamma_i \in_U \mathbf{F}_q$ である.

G に k 個の Vinegar 変数 $\boldsymbol{z} = (z_1, \ldots, z_k)^T$ を導入した非線形変換 $\widetilde{G}(\boldsymbol{v}) = (\widetilde{g}_1, \ldots, \widetilde{g}_m)$ は以下のように表される：

$$\begin{aligned}
\widetilde{g}_1(\boldsymbol{v}) &= \sum_{1\le j\le l\le n} \alpha_{1,j,l} v_j v_l + \sum_{1\le j\le n} \eta_{1,j}(\boldsymbol{z}) v_j + \tau_1(\boldsymbol{z}),\\
&\vdots\\
\widetilde{g}_i(\boldsymbol{v}) &= \sum_{1\le j\le l\le n} \alpha_{i,j,l} v_j v_l + \sum_{1\le j\le n} \eta_{i,j}(\boldsymbol{z}) v_j + \tau_i(\boldsymbol{z}),\\
&\vdots\\
\widetilde{g}_m(\boldsymbol{v}) &= \sum_{1\le j\le l\le n} \alpha_{m,j,l} v_j v_l + \sum_{1\le j\le n} \eta_{m,j}(\boldsymbol{z}) v_j + \tau_m(\boldsymbol{z}),\\
\eta_{i,j}(\boldsymbol{z}) &= \sum_{1\le l\le k} \lambda_{i,j,l} z_l + \beta_{i,j}, \quad \tau_i(\boldsymbol{z}) = \sum_{1\le j\le l\le k} \varphi_{i,j,l} z_j z_l + \sum_{1\le j\le k} \sigma_{i,j} z_j + \gamma_i.
\end{aligned}$$

ここで $\lambda_{i,l}, \varphi_{i,j}, \sigma_i \in_U \mathbf{F}_q$ である．

$\widetilde{G}$ の逆変換は，$\boldsymbol{z}$ に値を与えたものに対し，G^{-1} を計算することにより行う．

Vinegar 変数を導入した多変数 2 次公開鍵暗号としては，HFE を原方式とする QUARTZ が提案されている [PCG01]．QUARTZ には Vinegar 変数に加えて，Minus method が Modifier として適用されている．QUARTZ のパラメータは以下の通りである．

- $q = 2$, $n = 103$, $k = 4$, $r = 3$, $d = 129$.

■Internal Perturbation [Din04, DS05a, DG06]　Internal Perturbation とは，非線形変換 G に中間変数 $\boldsymbol{v}$ に対するランダムな変換（摂動多項式）を加える Modifier である（図 7）．

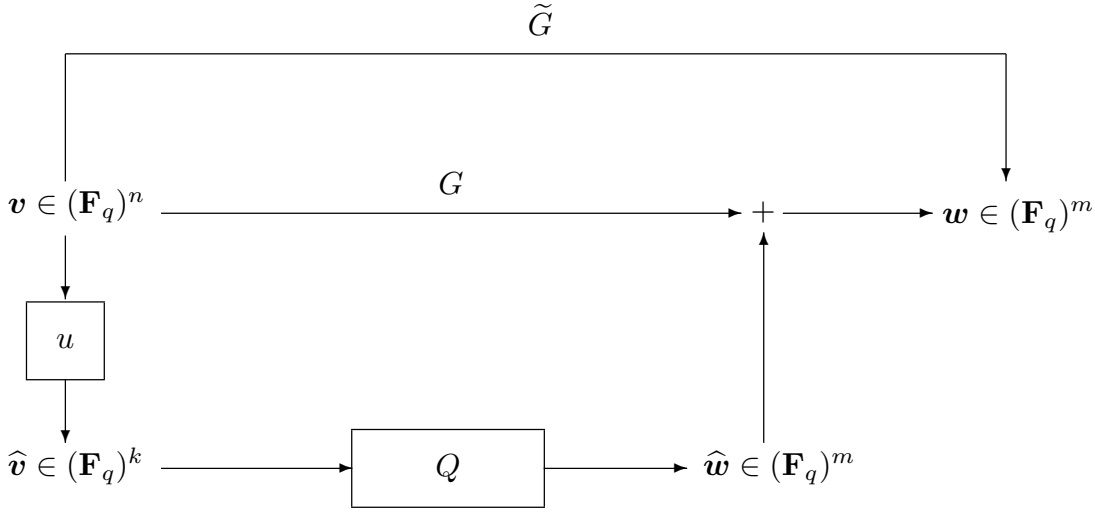

図 7　Internal Perturbation

線形変換 $u : (\mathbf{F}_q)^n \to (\mathbf{F}_q)^k$，非線形変換 $Q = (q_1, \ldots, q_m) : (\mathbf{F}_q)^k \to (\mathbf{F}_q)^m$ に対し，非線形変換 G に Internal Perturbation を適用した変換 $\widetilde{G}$ は $\widetilde{G}(\boldsymbol{v}) = (G + (Q \circ u))(\boldsymbol{v})$ と表される．

$\widetilde{G}^{-1}$ の計算は，すべての $\widehat{\boldsymbol{v}} = u(\boldsymbol{v})$ に対し，$G^{-1}(\boldsymbol{w} - Q(\widehat{\boldsymbol{v}})) = \boldsymbol{v}$ であるかどうか検査するため G^{-1} を q^k 回繰り返す必要がある．

Internal Perturbation を適用した多変数 2 次公開鍵暗号について，以下のパラメータが提案されている．なお，PMI+ とは，MI に Internal Perturbation と Plus method を適用したものである．

- PMI: $q = 2$, $n = 96$, $k = 5$, $h = 0$.
- PMI: $q = 2$, $n = 136$, $\theta = 40$, $k = 6$, $h = 0$.
- PMI+: $q = 2$, $n = 84$, $\theta = 4$, $k = 6$, $h = 14$.
- PMI+: $q = 2$, $n = 136$, $\theta = 8$, $k = 6$, $h = 18$.
- IPHFE: $q = 2$, $n = 89$, $d = 9$, $k = 2$.

4 代数攻撃

多変数 2 次公開鍵暗号の公開鍵

$$E(\boldsymbol{x}) = (e_1(\boldsymbol{x}), \ldots, e_m(\boldsymbol{x}))^T \in (\mathbf{F}_q[x_1, \ldots, x_n])^m$$

と暗号文 $\boldsymbol{c} = (c_1, c_2, \ldots, c_m)^T \in (\mathbf{F}_q)^m$ を用いて，以下の非線形連立方程式

$$\begin{cases} e_1(x_1, \ldots, x_n) = c_1 \\ e_2(x_1, \ldots, x_n) = c_2 \\ \vdots \\ e_m(x_1, \ldots, x_n) = c_m \end{cases} \tag{4.1}$$

を $x_1, \ldots, x_n$ について解くことにより，攻撃者が暗号文に対応する平文を得る攻撃を，多変数 2 次公開鍵暗号に対する代数攻撃と呼ぶ.

4.1 代入攻撃

$n > m$ の場合，一般に，方程式 (4.1) の解の個数は q^{n-m} であり，特に $n \gg m$ の場合，これらの解をすべて求めるのは困難である.

方程式 (4.1) について，少なくとも 1 つの解を求めればよい場合，e_i における n 個の変数のうち $(n - m + u)$ 個の変数に値を代入したもの $\overline{e}_i$ について，以下の非線形連立方程式

$$\begin{cases} \overline{e}_1(x_1, \ldots, x_{m-u}) = c_1 \\ \overline{e}_2(x_1, \ldots, x_{m-u}) = c_2 \\ \vdots \\ \overline{e}_m(x_1, \ldots, x_{m-u}) = c_m \end{cases} \tag{4.2}$$

を解く方がより効率的である．このような計算によって行う代数攻撃を代入攻撃と呼ぶ.

$u = 0$ の代入攻撃の場合，方程式 (4.2) の解を必ず得られると限らないものの，解を得られる確率は $(1 - 1/e)$（e は自然対数）であり，平均して 1.6 回に 1 回の割合で解を得

ることができる．一方，$u > 0$ の代入攻撃の場合，$u = 0$ の場合よりも余分に代入した q^u 個の値に相当する回数分，方程式 (4.2) を解く必要がある．

4.2 グレブナ基底攻撃

方程式 (4.1) をグレブナ基底計算を用いて解く代数攻撃は，グレブナ基底攻撃と呼ばれている．

方程式 (4.1) に対するグレブナ基底計算のための計算量は，計算の際の中間基底の最大次数 d_{reg} に対し，$\mathcal{O}\left(\left(m\binom{n+d_{\mathrm{reg}}-1}{d_{\mathrm{reg}}}\right)^w\right)$ である [BFS04, BFSY05]．ここに $2 \leq w \leq 3$ は線形代数の計算コストである．$m = 16, 20$ に対する d_{reg} の理論値および実測値を以下に示す [FP08, BFP08]．

m	n	d_{reg}（理論値）	d_{reg}（実測値）
16	15	9	9
16	14	7	7
16	13	6	6

m	n	d_{reg}（理論値）	d_{reg}（実測値）
20	19	11	
20	18	9	9
20	17	8	8
20	16	7	7
20	15	6	6

4.2.1 HFE に対するグレブナ基底攻撃 [Pat96a, CDF03, FJ03, GJS06]

HFE の提案時，グレブナ基底攻撃のための計算量は理論的に $\mathcal{O}(D^{3n})$ あるいは，実験的に $\mathcal{O}(D^{2.7n})$ と考えられていた．ここに D は公開鍵多項式の次数であり，HFE の場合 $D = 2$ である．それゆえ，この攻撃計算量が 2^{64} 以上となるためには，$n \geq 23$ が必要と考えられていた．

その後の研究により，$d \leq 512$ の HFE に対し，グレブナ基底攻撃によって，以下の計算量により暗号解読可能とされている．

- $4 \leq d \leq 16$ の場合 $\mathcal{O}(n^6)$
- $17 \leq d \leq 128$ の場合 $\mathcal{O}(n^8)$
- $129 \leq d \leq 512$ の場合 $\mathcal{O}(n^{10})$

実際に，$n = 80, d = 96$ の HFE について，約 2 日かけてグレブナ基底攻撃に成功したという結果が示されている[*5] ．

[*5] 文献 [FJ03] では，1GHz の CPU を搭載した Sunfire v880 と，F_5 アルゴリズムを用いて，およそ 52.2 時間かけて解読に成功したことが報告されている．一方 [Ste04] では，750 MHz の CPU を搭載したコンピュータを用いて，Magma V2.11-9 に実装された F_4 アルゴリズムと，HFE 解読のためのオプションを利用することにより，およそ 22.1 時間かけて解いたという結果が示されている．

この計算量は，ヒューリスティックな議論により $\mathcal{O}(n^{\mathcal{O}(\log d)})$ と見積もられている．特に，d がある定数 α に対して $d = \mathcal{O}(n^{\alpha})$ である場合，この計算量は $2^{\mathcal{O}(\log n)^2}$ となり，n に対する準指数時間計算量となる．しかしながら，この計算量は，いわゆる準指数時間計算量よりもずっと小さいため，準多項式時間 (quasipolynomial time) 計算量と言われている．

一方，Modifier（Vinegar 変数，Minus method）を適用した，$q = 2$ の HFE については，平文変数の数が n であり，k 個の Vinegar 変数を導入し，r 個の公開鍵多項式を公開しない Minus method を適用した場合，$u = 0$ の代入グレブナ基底攻撃のための計算量は $q^{k+r} \cdot (n-r)^{10}/4$ と推測されている．たとえば $n = 103, k = 4, r = 3$ の場合，この計算量は 2^{71} 程度と見積もられていた．しかしながら，その後の研究により，この計算量は 2^{62} と推測されている．

4.3 代数攻撃のためのアルゴリズム

XL アルゴリズムや，Magma のグレブナ基底計算アルゴリズム以外にも，代数攻撃のために，さまざまなアルゴリズムが提案されている．

■GeometricXL [MP08] XL アルゴリズムにおける計算過程を，代数幾何学の観点から解釈した，GeometricXL と呼ばれるアルゴリズムが提案されている．

■HXL（“Heuristic and Hybrid” XL）アルゴリズム [GT08a, GT08b] XL アルゴリズムにおいて生成される多項式集合について，互いに線形従属な多項式が生成されにくくなるように，アルゴリズムの改善を行っている．Magma のスクリプト言語を用いて実装した HXL アルゴリズムと，Magma V2.13 の F_4 アルゴリズムを用いて計算した場合との比較が行われており，HXL の方が，計算に使用するメモリ容量を，より少なくすることができるという結果が報告されている．

■MXL（Mutant XL）アルゴリズム [DBMMW08, MMDB08, MCDBB09] Mutant と呼ばれる，グレブナ基底の計算過程において，突然変異的に得られる多項式に着目して，XL アルゴリズムを改善したものである．MXL アルゴリズムの新しいバージョン（MXL_3）においては，Magma の F_4 アルゴリズムよりも，効率的に計算されうるという結果が示されている．

■Zhuang-Zi（荘子）アルゴリズム MI 型の多変数 2 次公開鍵暗号の構成に用いられているように，有限体上の多変数多項式は，一変数多項式として見ることができる．この性

質を利用し，主に，一変数多項式の因数分解を行うことにより，求解を行ってゆくアルゴリズムである [DGS06a]．いくつかの例においては，Magma の F_4 アルゴリズムを用いて解くのが困難であっても，Zhuang-Zi アルゴリズムを用いて解けることが確認されている．最近，Mutant の概念を導入した Zhuang-Zi アルゴリズムが提案されている [DS10]．

■CS method　特に，幾何の定理の自動証明などの分野において，多変数非線形連立方程式を解くために，古くから用いられている手法として，CS（Characteristic Set）method（特性集合法）と呼ばれる方法がある [Rit50, Wu78] [*6]．この方法を，共通鍵ストリーム暗号の解読に利用するといった研究が行われており，グレブナ基底計算よりも効率的に計算され得ることが示されている [GH09]．

■F_5 アルゴリズム [Fau02]　F_4 アルゴリズムにおいては，必ずしも，掃き出す行列がフルランクにならないという問題点がある．F_5 アルゴリズムでは，この行列が，必ず，フルランクとなるように，グレブナ基底の計算過程に関する履歴を利用して，冗長な部分を取り除いている．

■PET SNAKE [GMS09]　共通鍵ブロック暗号の解読を目的として，開発が進められている，代数攻撃用のハードウェアである．PET SNAKE は，非線形連立方程式を解くために，新たに提案された，MRHS（Multiple Right Hand Sides）と呼ばれる手法 [RS08] をベースとしている．ちなみに PET SNAKE とは，Parallel Elimination Technique Supporting Nice Algebraic Key Elimination を略した名称である．

■PolyBoRi [BD07]　特に，係数体，および，解の値が，いずれも $\mathbf{F}_2$ 上にあるような，非線形連立方程式の求解は，暗号分野のみならず，論理回路の設計など，さまざまな応用が考えられる．このような，ブール多項式環上の多項式（Polynomials over Boolean Rings）に関するグレブナ基底計算のためのパッケージを提供するプロジェクトが進められている．

5　多変数公開鍵暗号の安全性解析

多変数公開鍵暗号に対する代数攻撃の手法は，単に，暗号方式を攻撃するだけでなく，その安全性を評価するために用いられている．以下では，代数攻撃を利用して，多変数公開鍵暗号の安全性を解析した，いくつかの結果について述べる．

[*6] Wu の方法，Ritt-Wu の分解アルゴリズムなどと呼ばれることがある．

5.1 UOV に対する安全性解析 [BWP05, FP08]

Magma の F_4 アルゴリズムを用いて，UOV（3.2.3 節）に対する代入攻撃を行った結果から，$n = 3m, 4m$ の UOV に対し，$u = 0$ の代入グレブナ基底攻撃のための計算量が 2^{64} 以上となるためには，$q = 2$ の場合 $m \geq 38$，$q = 3$ の場合 $m \geq 24$ である必要があると推測されている [BWP05].

方程式 (4.1) において，$m = 16, n = 14$ の場合，$d_{\mathrm{reg}} = 7$ であり，グレブナ基底計算の計算量は高々 $2^{52.7}$ となる．一方で，$q = 16, m = 16, n = 32, 48$ の UOV に対し，F_5 アルゴリズムを用いた $u = 2$ の代入グレブナ基底攻撃のための計算量は，この値よりもずっと小さく，$2^{32.3}$ となった実験結果が報告されている [FP08].

5.2 PMI に対する安全性解析

Magma の F_4 アルゴリズムを用いて，PMI（3.3 節）に対するグレブナ基底攻撃を行った結果から，$q = 2$ の PMI について，提案者によって，十分安全と考えられていたパラメータである，$k = 5$, $n = 96$ の場合であっても，グレブナ基底攻撃のための計算量が 2^{80} を大きく下回ると推測されている．一方，$k = 6$ の場合，$n \geq 83$ であれば，グレブナ基底攻撃の計算量が 2^{80} を上回ると推測されている [DGSWY05].

$k = 6$ の場合，PMI に対するグレブナ基底攻撃の計算量が指数時間となる可能性が示唆されている．しかしながら，計算機実験環境 A（1.2 節）を使用して，実験を行ってみたところ，$k \leq 10$, $24 \leq n \leq 26$ の場合，PMI に対するグレブナ基底攻撃と，グレブナ基底計算によるランダムな連立 2 次方程式の求解のための計算量が同等とならないという結果が得られている（表 3）.

表 3　PMI に対するグレブナ基底攻撃のための計算時間（秒）

		$n = 22$	$n = 23$	$n = 24$	$n = 25$	$n = 26$
PMI	$k = 9$	56	102	160	303	501
	$k = 10$	57	90	156	403	515
	$k = 11$	57	89	596	1225	2597
	$k = 12$	57	89	604	1231	2592
random		54	85	577	1185	2516

5.3 持駒方式の安全性解析

任意の多変数公開鍵暗号系の安全性を強化する概念として，持駒概念 [Tsu03] が提案されている．この概念を具現化したものである，持駒方式と呼ばれる安全性強化手法が，これまでに，いくつか提案されている．持駒方式の具体的な構成方法については，文献 [TTF07a, TTF08, FTT08c] などを参照されたい．

以下では，これらの持駒方式の安全性について，グレブナ基底計算を用いた代数攻撃に対する安全性に関する結果について述べる．下記の計算機実験においては，計算機実験環境 A（1.2 節）を使用した．なお，HFE の解読に使われた HFE オプションなどのような，グレブナ基底計算に関する Magma のオプションについては一切使用していない．

5.3.1 線形持駒行列方式 [TTF07a]

HFE（3.2.1 節）に乱数変数を付加した線形持駒行列方式を適用することにより，グレブナ基底攻撃に対する安全性が強化されることが，計算機実験により示されている（表 4）．特に平文変数の数を一定（10）とした場合，原方式である HFE と比較して，持駒方式において，乱数変数を導入することにより，変数の総数 $z = 39$ とした方が，計算時間が約 10^6 倍になることが計算機実験から明らかとなった．また，復号時間が一定となるように，n を一定（20）とした場合，原方式である HFE と比較して，持駒方式において変数の総数 $z = 39$ の方が，計算時間が 740 倍以上になることが計算機実験から明らかとなった．

表 4　グレブナ基底攻撃のための計算時間の比較（線形持駒行列方式）

方式	パラメータ				計算時間
	p	n	z	g	(sec.)
HFE（$q = 2$, $128 < d < 513$）		10			$< 10^{-3}$
		20			8
		25			184
		28			959

方式	パラメータ				計算時間
	p	n	z	g	(sec.)
線形持駒行列方式（原方式：HFE（$q = 2$, $128 < d < 513$)）	10	20	35	25	1000
	10	20	37	25	2424
	10	20	38	25	5288
	10	20	32	28	665
	10	20	36	28	2290
	10	20	38	28	4460
	10	20	39	28	5963

p : 持駒方式における平文変数の数，n : 原方式における平文変数の数

z : 持駒方式における変数（平文変数，乱数変数）の総数，g : 持駒方式における公開鍵多項式の数

5.3.2　非線形持駒行列方式 [TTF08]

MI（3.2.1 節），RSE（3.2.2 節）に，非線形持駒行列方式を適用することにより，線形持駒行列方式よりもグレブナ基底攻撃に対する安全性が強化されることが，計算機実験により示されている（表 5）．表 5 から，非線形持駒行列方式の方が，線形持駒行列方式と比較して，計算時間が約 10 倍から 100 倍大きくなっていることがわかる．また，平文変数の数を一定（25）とした場合，原方式である MI，RSE と比較して持駒方式において変数の総数 $z = 52$ の方が，計算時間が約 10^4 倍になることが計算機実験から明らかとなった．

表 5　グレブナ基底攻撃のための計算時間の比較（持駒行列方式）

方式	パラメータ				計算時間 (sec.)
	p	n	z	g	
MI ($q = 2$)		15			$< 10^{-2}$
		20			0.01
		25			0.03
		30			0.07
		35			0.2
		40			0.4
		45			0.7
		50			1
		55			2
		60			4

方式	パラメータ				計算時間 (sec.)	
	p	n	z	g	線形	非線形
持駒行列方式（原方式：MI（$q = 2$））	25	35	50	47	3	52
	25	35	51	47	6	260
	25	35	52	47	22	1307
	25	35	54	47	58	n/a
	25	35	56	47	829	n/a
	30	40	54	50	3	59
	30	40	55	50	5	263
	30	40	56	50	7	1281
	30	40	58	50	47	n/a
	30	40	60	50	1016	n/a

方式	パラメータ				計算時間 (sec.)
	p	n	z	g	
RSE ($q = 2$)		15			0.01
		20			0.03
		25			0.08
		30			0.2
		35			0.5
		40			1
		45			2
		50			5
		55			9
		60			16

方式	パラメータ				計算時間 (sec.)	
	p	n	z	g	線形	非線形
持駒行列方式（原方式：RSE（$q = 2$））	25	35	50	47	6	50
	25	35	51	47	13	250
	25	35	52	47	19	1309
	25	35	54	47	131	n/a
	25	35	56	47	1622	n/a
	30	40	54	50	8	58
	30	40	55	50	11	264
	30	40	56	50	28	1285
	30	40	58	50	158	n/a
	30	40	60	50	1770	n/a

p : 持駒方式における平文変数の数，n : 原方式における平文変数の数
z : 持駒方式における変数（平文変数，乱数変数）の総数，g : 持駒方式における公開鍵多項式の数
n/a は計算不可を示す．

5.3.3　非線形持駒摂動ベクトル方式 [FTT08c]

MI に非線形持駒摂動ベクトル方式を適用することにより，グレブナ基底攻撃に対する安全性が Internal Perturbation（3.3 節）と同等に強化されることが計算機実験により示されている（表 6，表 7）.

乱数変数を付加しない非線形持駒摂動ベクトル方式について，特に，平文変数の数を一定 ($n = 30$) とした場合，原方式である MI と比較して，持駒方式は PMI+ と同様に計算時間が約 10^4 倍になることが計算機実験から明らかとなった.

一方，乱数変数を付加した非線形持駒摂動ベクトル方式について，平文変数の数を一定（15）とした場合，原方式である MI，RSE と比較して，持駒方式において変数の総数 z と公開鍵多項式の数 g がそれぞれ $z = 47, g = 35$（原方式：MI）$z = 44, g = 35$（原方式：RSE）の方が，計算時間が約 10^5 倍になることが計算機実験から明らかとなった（表 8）.

表 6　PMI+（$q = 2$）に対するグレブナ基底攻撃のための計算時間

パラメータ			計算時間
n	k	h	(sec.)
28	6	0	845
28	6	5	733
28	6	10	563
28	6	15	436
29	6	15	747
30	6	15	1305

n: 平文変数の数
k: perturbation dimension
h: Plus 多項式の数

表 7　非線形持駒摂動ベクトル方式を適用した MI（$q = 2$）に対するグレブナ基底攻撃のための計算時間

パラメータ			計算時間
n	l	h	(sec.)
28	17	3	290
28	17	4	289
28	17	5	263
29	17	3	537
29	17	8	402
29	17	10	349
30	17	3	936
30	17	8	701
30	17	13	513

n: 平文変数の数
l: 補助方式における変数の数
h: ランダム項ベクトルの次元

なお，非線形持駒摂動ベクトル方式の計算機実験において，補助方式として，HFE の公開鍵多項式による非線形変換を用いたが，補助方式として，どのような方式が最適であるかについては未解決問題である.

表 8　グレブナ基底攻撃のための計算時間の比較（非線形持駒摂動ベクトル方式）

方式	パラメータ p	n	z	g	計算時間 (sec.)
MI（$q=2$）		15			$< 10^{-2}$
		20			0.01
		25			0.03
		30			0.07
		35			0.2
		40			0.4
		45			0.7
		50			1
		55			2
		60			4

方式	パラメータ p	n	z	g	計算時間 (sec.)
非線形持駒摂動ベクトル方式（原方式：MI（$q=2$））	15	20	40	35	75
	15	20	43	35	129
	15	20	45	35	260
	15	20	46	35	320
	15	20	47	35	1029
	15	20	40	40	97
	15	20	43	40	161
	15	20	47	40	284
	15	20	48	40	495
	15	20	49	40	1077

方式	パラメータ p	n	z	g	計算時間 (sec.)
RSE（$q=2$）		15			0.01
		20			0.03
		25			0.08
		30			0.2
		35			0.5
		40			1
		45			2
		50			5
		55			9
		60			16

方式	パラメータ p	n	z	g	計算時間 (sec.)
非線形持駒摂動ベクトル方式（原方式：RSE（$q=2$））	15	20	40	35	40
	15	20	41	35	71
	15	20	42	35	179
	15	20	43	35	713
	15	20	44	35	2791
	15	20	40	40	51
	15	20	42	40	82
	15	20	44	40	231
	15	20	45	40	877
	15	20	46	40	2327

p : 持駒方式における平文変数の数，n : 原方式における平文変数の数

z : 持駒方式における変数（平文変数，乱数変数）の総数，g : 持駒方式における公開鍵多項式の数

5.4　rSTS 型多変数公開鍵暗号の安全性解析

5.4.1　rSTS 型多変数公開鍵暗号 [WBP04, WBP06]

多変数 2 次公開鍵暗号の秘密鍵 $G(\boldsymbol{v}) = (g_1(\boldsymbol{v}), \ldots, g_n(\boldsymbol{v}))$ が図 8 のような形（gSTS: general STS あるいは単に STS）をなすものを，STS 型多変数公開鍵暗号と呼ぶ（以下では，STS 型 MPKC と略記する）．ここに，$n = r_1 + \cdots + r_L, m = m_1 + \cdots + m_L$ である．

STS 型多変数公開鍵暗号のうち，特に $r_1 = r_2 = \cdots = r_L = m_1 = m_2 = \cdots = m_L (= r)$ であるものを rSTS 型多変数公開鍵暗号と呼ぶ（以下では，rSTS 型 MPKC と略記

$$
\text{Step } 1 \left\{ \begin{array}{c} g_1(v_1, \ldots, v_{r_1}) \\ \vdots \\ g_{m_1}(v_1, \ldots, v_{r_1}) \end{array} \right.
$$

$$
\vdots
$$

$$
\text{Step } l \left\{ \begin{array}{l} g_{m_1+\cdots+m_{l-1}+1}(v_1, \ldots, v_{r_1}, \ldots, v_{r_1+\cdots+r_{l-1}+1}, \ldots, v_{r_1+\cdots+r_l}) \\ \vdots \\ g_{m_1+\cdots+m_l}(v_1, \ldots, v_{r_1}, \ldots, v_{r_1+\cdots+r_{l-1}+1}, \ldots, v_{r_1+\cdots+r_l}) \end{array} \right.
$$

$$
\vdots
$$

$$
\text{Step } L \left\{ \begin{array}{c} g_{m_1+\cdots+m_{L-1}+1}(v_1, \ldots, v_n) \\ \vdots \\ g_m(v_1, \ldots, v_n) \end{array} \right.
$$

図 8　STS 型多変数公開鍵暗号における秘密鍵 G

する）.

rSTS 型 MPKC としては，これまでに，$r = 1$ の場合である，順序解法を用いた方式，r が任意の場合である，R(S)SE（いずれの方式も 3.2.2 節）が提案されている．

5.4.2　rSTS 型多変数公開鍵暗号の安全性

$r = 1, 4, 10$ として，$q = 2$, $n = m = 40$ である rSTS 型多変数公開鍵暗号に対し，計算機実験環境 B（1.2 節）を使用して，グレブナ基底攻撃の計算機実験を行った．

実験方法としては，まず，各 r ごとに，10 個の公開鍵を生成した．次に，これらの公開鍵を用いて，暗号文をそれぞれ 100 対ずつ生成し，これらの暗号文に対応する平文をグレブナ基底攻撃によって求め，計算時間を計測した．求めたグレブナ基底における線形式の数と，計算時間の関係を図 9 に示す．図 9 の横軸である，グレブナ基底における線形式の数は，そのイデアルの零点における非線形性，すなわち，方程式 (4.1) の解空間の複雑さに影響する．一般に，この複雑さが，グレブナ基底計算の困難性に影響を及ぼすといわれている．また，図 9 においては，計算時間を表す縦軸を対数目盛としている．これは，グレブナ基底計算途中に生成される中間多項式の次数の増加に対し，計算時間が指数的に増加するためである．

図 9 より，rSTS 型 MPKC に対するグレブナ基底攻撃において，グレブナ基底における線形式の数と，計算時間との間に，強い相関がみられない．このため，この線形式の数が，計算時間の長短に及ぼす影響は大きくないと考えられる．また，rSTS 型 MPKC のパラメータを一定としても，公開鍵や暗号文によって，計算時間に大きなばらつきがあ

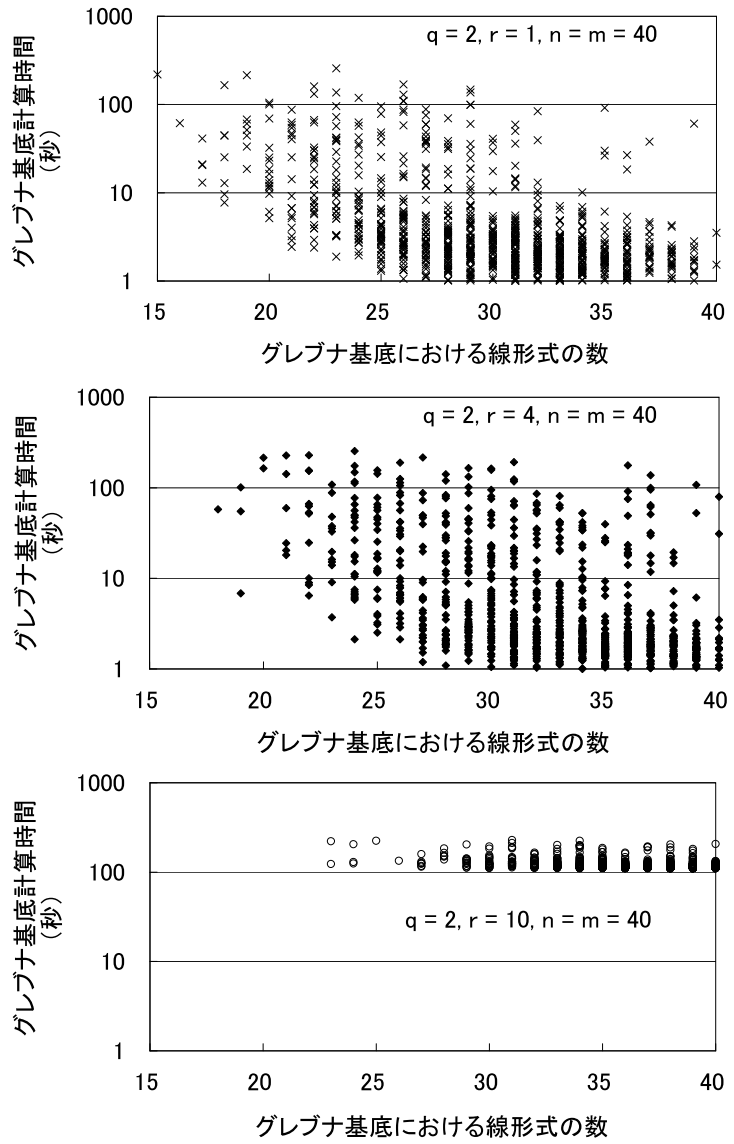

図 9　rSTS 型 MPKC に対するグレブナ基底攻撃の計算時間（鍵数 10 × 平文数 100）

り，特に，$r = 1, 4$ の場合，その差が数百倍程度に及んでいる．一方，$r = 10$ の場合，$r = 1, 4$ の場合と比較して，グレブナ基底攻撃の計算時間が全体的に増大するとともに，そのばらつきが小さくなることが，図 9 から明らかとなった．

次に，グレブナ基底攻撃における計算途中の中間多項式の次数の遷移を図 10 に示す．

図 10 より，r が小さくなるにつれて，グレブナ基底計算のステップ数が大きくなり，途中の中間多項式の次数の変動回数が多くなるという結果が得られた．こうした次数の変動が，グレブナ基底攻撃の計算時間のばらつきなどに影響を及ぼすと見られるが，この次

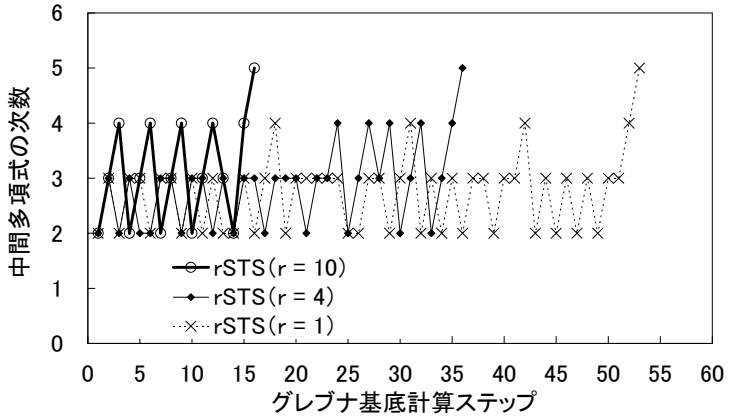

図 10　グレブナ基底計算途中の中間多項式の次数の遷移（rSTS 型 MPKC : $q = 2$, $n = m = 40$）

数の遷移に関する理論的な解析は，今後の研究課題である．

5.4.3　R(S)SE の安全性

$q = 2, n = m = 40$ とし，t あるいは r を $4, 10$ とした場合の RSE, RSSE（3.2.2 節）と，rSTS 型 MPKC に対するグレブナ基底攻撃の計算時間の比較を表 9 に示す．

表 9 では，各方式およびパラメータにおいて生成した 9 個の鍵に対し，それぞれ 100 通りの暗号文に対応する平文を，グレブナ基底攻撃によって求める計算時間の平均値について，それぞれの 9 個の鍵の間での最大値，最小値，中央値を示している．また，標準偏差についても，各鍵ごとに統計量を算出し，それらのうちの中央値を表 9 に示す．

表 9　グレブナ基底攻撃の計算時間の比較

$q = 2$ $n = m = 40$		計算時間（秒）			
	方式	最小	中央値	最大	標準偏差（中央値）
$t = 4$	RSE	0.049	0.426	0.541	0.031
	RSSE	0.306	0.336	0.459	0.039
$r = 4$	rSTS	6	14	27	25.508
$t = 10$	RSE	0.586	0.652	0.738	0.006
	RSSE	81	82	91	0.477
$r = 10$	rSTS	119	121	124	15.588

表 9 から，RSE, RSSE は，いずれも，rSTS 型 MPKC に分類されるものの，グレブナ基底攻撃の計算時間が同等になると限らないことが明らかとなった．

次に，RSE, RSSE に対するグレブナ基底攻撃における，計算途中の中間多項式の次数の遷移を図 11 に示す．

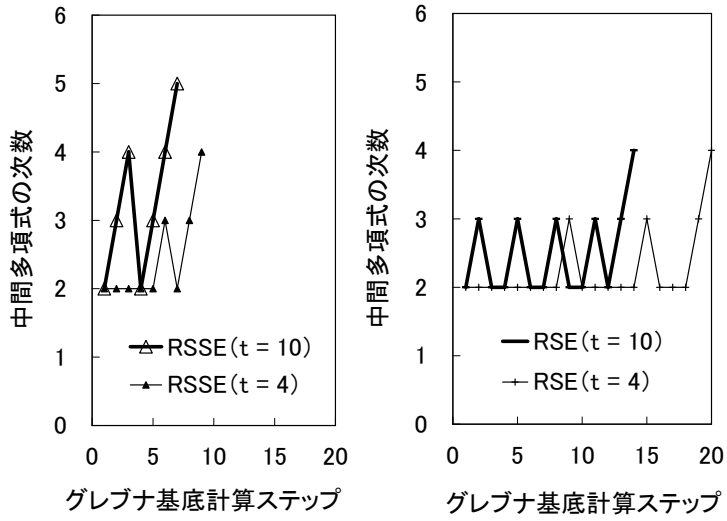

図 11　グレブナ基底計算途中の中間多項式の次数の遷移（RSE, RSSE : $q = 2$, $n = m = 40$）

図 10，図 11 より，表 9 における計算時間が小さいものは，いずれも，中間多項式の最大次数が小さいという結果が得られた．

グレブナ基底攻撃の際，計算途中の中間多項式について，その最大次数の理論的な見積もりについては，5.4 節に述べた，次数の遷移と同様に，今後の研究課題として残されている．

6　おわりに

多変数公開鍵暗号は，その数学的構造や構成方法，安全性に関して，まだ解明されていない部分が非常に多い．このため，既存の攻撃手法に対して，十分に安全であると考えられて，多変数公開鍵暗号が提案されても，その後まもなく，実用的な攻撃法が提案されるケースが多々ある．このような現状を特徴付けるように，[DFSS07] の結言において，Dubois らは以下のように述べている．

> Multivariate cryptographic schemes are very efficient but have a lot of exploitable mathematical structure. Their security is not fully understood, and

new attacks against them are found on a regular basis. It would thus be prudent not to use them in any security-critical applications.

多変数公開鍵暗号が耐量子コンピュータ公開鍵暗号として実用に供されるためには，今後多くの研究を積み重ねてゆく必要があると考える．

一方，グレブナ基底計算において，例えば，計算途中の中間多項式の次数の遷移など，アルゴリズムの動作について，いまだ十分に解明されていない部分が多い．また，計算アルゴリズムについても，どのような問題に対して，どのようなアルゴリズムを用いれば，より効率的に解くことができるのかといったことなど，いまだ明らかとなっていない点が多い．

以上，多変数公開鍵暗号と，その周辺分野における数学について，解くべき問題が山積している．これらの分野における，理論および応用面における研究の発展が，新しい暗号を生み出す礎となり，来るべき未来に備えるべく，情報セキュリティの基盤技術としての暗号の分野を切り開き，多大なる貢献をもたらすものと考えている．

参考文献

[AFIKS04] G. Ars, J. C. Faugère, H. Imai, M. Kawazoe, and M. Sugita, "Comparison between XL and Gröbner basis algorithms," *Proc. ASIACRYPT 2004*, Lecture Notes in Computer Science, vol.3329, pp.338–353, Springer, 2004.

[BFS04] M. Bardet, J. C. Faugère, and B. Salvy, "On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations," Proceedings of International Conference on Polynomial System Solving (ICPSS 2004), pp.71–75, Nov. 2004.

[BFSY05] M. Bardet, J. C. Faugère, B. Salvy, and B. Y. Yang, "Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems," Proceedings of MEGA 2005, May 2005.

[BBD09] D. J. Bernstein, J. Buchmann, and E. Dahmen (editors), Post-Quantum Cryptography, Springer, 2009.

[BFP08] L. Bettale, J. C. Faugère, and L. Perret, "Cryptanalysis of the TRMS signature scheme of PKC'05," *Proc. AFRICACRYPT 2008*, Lecture Notes in Computer Science, vol.5023, pp.143–155, Springer, 2008.

[BL95] D. Boneh and R. J. Lipton, "Quantum cryptanalysis of hidden linear func-

tions," *Proc. CRYPTO '95*, Lecture Notes in Computer Science, vol.963, pp.424–437, Springer, 1995.

[BWP05] A. Braeken, C. Wolf, and B. Preneel, "A study of the security of unbalanced oil and vinegar signature schemes," *Proc. CT-RSA 2005*, Lecture Notes in Computer Science, vol.3376, pp.29–43, Springer, 2005.

[BD07] M. Brickenstein and A. Dreyer, "POLYBORI: A Gröbner basis framework for Boolean polynomials," Reports of Fraunhofer ITWM, no.122, 2007.

[Buc65] B. Buchberger, "Ein Algorithmus zum auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal," PhD thesis, Innsbruck, 1965.

[CKM97] S. Collart, M. Kalkbrener, and D. Mall, "Converting bases with the Gröbner walk," Journal of Symbolic Computation, vol.24, no.3, pp.465–469, Sept. 1997.

[CKPS00] N. Courtois, A. Klimov, J. Patarin, and A. Shamir, "Efficient algorithms for solving overdefined systems of multivariate polynomial equations," *Proc. EUROCRYPT 2000*, Lecture Notes in Computer Science, vol.1807, pp.392–407, Springer, 2000.

[Cou01] N. Courtois, "The security of Hidden Field Equations (HFE)," *Proc. CT-RSA 2001*, Lecture Notes in Computer Science, vol.2020, pp.266–281, Springer, 2001.

[CGMT02] N. Courtois, L. Goubin, W. Meier, and J. D. Tacier, "Solving underdefined systems of multivariate quadratic equations," *Proc. PKC 2002*, Lecture Notes in Computer Science, vol.2274, pp.211–227, Springer, 2002.

[CP02] N. Courtois and J. Pieprzyk, "Cryptanalysis of block ciphers with overdefined systems of equations," *Proc. ASIACRYPT 2002*, Lecture Notes in Computer Science, vol.2501, pp.267–287, Springer, 2002.

[Cou02] N. Courtois, "Higher order correlation attacks, XL algorithm and cryptanalysis of Toyocrypt," *Proc. ICISC 2002*, Lecture Notes in Computer Science, vol.2587, pp.182–199, Springer, 2003.

[CDF03] N. Courtois, M. Daum, and P. Felke, "On the security of HFE, HFEv- and Quartz," *Proc. PKC 2003*, Lecture Notes in Computer Science, vol.2567, pp.337–350, Springer, 2003.

[CP03] N. Courtois and J. Patarin, "About the XL algorithm over $GF(2)$," *Proc. CT-RSA 2003*, Lecture Notes in Computer Science, vol.2612, pp.141–157, Springer,

2003.

[CGP03] N. Courtois, L. Goubin, and J. Patarin, "SFLASHv3, a fast asymmetric signature scheme," Cryptology ePrint Archive, Report 2003/211, 2003. http://eprint.iacr.org/

[CLO00] D. コックス，J. リトル，D. オシー著，落合啓之，示野信一，西山享，室政和，山本敦子訳，グレブナ基底と代数多様体入門（上・下），シュプリンガー・フェアラーク東京, 2000.

[CLO07] D. Cox, J. Little, and D. O'Shea, Ideals, Varieties, and Algorithms, third edition, Springer, 2007.

[Din04] J. Ding, "A new variant of the Matsumoto-Imai cryptosystem through perturbation," *Proc. PKC 2004*, Lecture Notes in Computer Science, vol.2947, pp.305–318, Springer, 2004.

[DS05a] J. Ding and D. Schmidt, "Cryptanalysis of HFEv and internal perturbation of HFE," *Proc. PKC 2005*, Lecture Notes in Computer Science, vol.3386, pp.288–301, Springer, 2005.

[DGSWY05] J. Ding, J. E. Gower, D. Schmidt, C. Wolf, and Z. Yin, "Complexity estimates for the F_4 attack on the perturbed Matsumoto-Imai cryptosystem," *Proc. IMA Int. Conf. 2005*, Lecture Notes in Computer Science, vol.3796, pp.262–277, Springer, 2005.

[DG06] J. Ding and J. E. Gower, "Inoculating multivariate schemes against differential attacks," *Proc. PKC 2006*, Lecture Notes in Computer Science, vol.3958, pp.290–301, Springer, 2006.

[DGS06a] J. Ding, J. E. Gower, and D. Schmidt, "Zhuang-Zi: a new algorithm for solving multivariate polynomial equations over a finite field," Workshop Record of the International Workshop on Post-Quantum Cryptography (PQCrypto 2006), pp.227–240, May 2006.

[DGS06b] J. Ding, J. E. Gower, and D. Schmidt, Multivariate Public Key Cryptosystems, Springer, 2006.

[DBMMW08] J. Ding, J. Buchmann, M. S. E. Mohamed, W. S. A. E. Mohamed, and R. P. Weinmann, "MutantXL," Proceedings of the First International Conference on Symbolic Computation and Cryptography (SCC 2008), pp.16–22, Apr. 2008.

[DS10] J. Ding and D. Schmidt, "Mutant Zhuang-Zi algorithm," *Proc. PQCrypto 2010*, Lecture Notes in Computer Science, vol.6061, pp.28–40, Springer, 2010.

[DFSS07] V. Dubois, P. A. Fouque, A. Shamir, and J. Stern, "Practical cryptanalysis of SFLASH," *Proc. CRYPTO 2007*, Lecture Notes in Computer Science, vol.4622, pp.1–12, Springer, 2007.

[FGLM93] 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, vol.16, no.4, pp.329–344, 1993.

[Fau99] J. C. Faugère, "A new efficient algorithm for computing Gröbner bases (F_4)," Journal of Pure and Applied Algebra, vol.139, issues 1-3, pp.61–88, June 1999.

[Fau02] J. C. Faugère, "A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5)," *Proc. ISSAC 2002*, pp.75–83, ACM Press, 2002.

[FJ03] J. C. Faugère and A. Joux, "Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases," *Proc. CRYPTO 2003*, Lecture Notes in Computer Science, vol.2729, pp.44–60, Springer, 2003.

[FP08] J. C. Faugère and L. Perret, "On the security of UOV," Proceedings of the First International Conference on Symbolic Computation and Cryptography (SCC 2008), pp.103–109, Apr. 2008.

[FL10] J. C. Faugère and S. Lachartre, "Parallel Gaussian elimination for Gröbner bases computations in finite fields," Proceedings of the 4th International Workshop on Parallel and Symbolic Computation (PASCO 2010), pp.89–97, July 2010.

[FTT08a] 藤田亮，只木孝太郎，辻井重男，"多様な多変数公開鍵暗号を汎用的に強化する非線形持駒摂動ベクトル方式，" *Proc. SCIS2008*, 1F1-1, Jan. 2008.

[FTT08b] R. Fujita, K. Tadaki, and S. Tsujii, "Nonlinear piece in hand perturbation vector method for enhancing security of multivariate public key cryptosystems," Cryptology ePrint Archive, Report 2008/298, July 2008. `http://eprint.iacr.org/`

[FTT08c] R. Fujita, K. Tadaki, and S. Tsujii, "Nonlinear piece in hand perturbation vector method for enhancing security of multivariate public key cryptosystems," *Proc. PQCrypto 2008*, Lecture Notes in Computer Science, vol.5299, pp.148–164, Springer, 2008.

[Fuj10a] 藤田亮，"rSTS 型多変数公開鍵暗号のグレブナ基底計算を用いた代数攻撃に対する安全性解析，" *Proc. SCIS2010*, 3A3-3, Jan. 2010.

[Fuj10b] R. Fujita, "Security analysis of rSTS type multivariate public key cryptosystems against algebraic attack using Gröbner bases," Recent Results Session at

the third international workshop on Post-Quantum Cryptography (PQCrypto 2010), May 25-28, 2010, Darmstadt, Germany.

[GH09] X. S. Gao and Z. Huang, "Efficient characteristic set algorithms for equation solving in finite fields and application in analysis of stream ciphers," Cryptology ePrint Archive, Report 2009/637, 2009. http://eprint.iacr.org/

[GJ79] M. Garey and D. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, Freeman, 1979.

[GMS09] W. Geiselmann, K. Matheis, and R. Steinwandt, "PET SNAKE: A special purpose architecture to implement an algebraic attack in hardware," Cryptology ePrint Archive, Report 2009/222, 2009. http://eprint.iacr.org/

[GT08a] 五太子政史，辻井重男，"有限体上の多変数連立二次方程式に関する新しい求解法の提案，" *Proc. SCIS2008*, 3B1-3, Jan. 2008.

[GT08b] M. Gotaishi and S. Tsujii, "HXL — a variant of XL algorithm computing Gröbner bases," Proceedings of Inscrypt 2008 Special Track on Symbolic Computation and Cryptology, pp.2–21, December 2008.

[GJS06] L. Granboulan, A. Joux, and J. Stern, "Inverting HFE is quasipolynomial," *Proc. CRYPTO 2006*, Lecture Notes in Computer Science, vol.4117, pp.345–356, Springer, 2006.

[Has09] Y. Hashimoto "Algorithms to solve massively under-defined systems of multivariate quadratic equations," Cryptology ePrint Archive, Report 2009/154. http://eprint.iacr.org/

[Hib06] 日比孝之編，グレブナー基底の現在，数学書房，2006.

[IM85] H. Imai and T. Matsumoto, "Algebraic methods for constructing asymmetric cryptosystems," *Proc. AAECC-3*, Lecture Notes in Computer Science, vol.229, pp.108–119, Springer, 1985.

[KS04] M. Kasahara and R. Sakai, "A construction of public key cryptosystem for realizing ciphertext of size 100 bit and digital signature scheme," *IEICE Transactions on Fundamentals*, vol.E87-A, no.1, pp.102–109, Jan. 2004.

[KS05a] M. Kasahara and R. Sakai, "A construction of public-key cryptosystem based on singular simultaneous equations," *IEICE Transactions on Fundamentals*, vol.E88-A, no.1, pp.74–80, Jan. 2005.

[KS05b] M. Kasahara and R. Sakai, "A construction of public-key cryptosystem based on singular simultaneous equations and its variants," IEICE Technical Report,

ISEC2005-7 (2005-05), May 2005.

[KFSM83] 桂重俊，藤木澄義，末永敏幸，松野明，“ランダムスピン系の統計力学における積分方程式，” 京都大学数理解析研究所講究録，no.486, pp.166–175, Apr. 1983.

[KFIFG87] S. Katsura, W. Fukuda, S. Inawashiro, N. M. Fujiki, and R. Gebauer, “Distribution of effective field in the ising spin glass of the $\pm J$ model at $T = 0$,” Cell Biochemistry and Biophysics, vol.11, no.1, pp.309–319, 1987.

[KPG99] A. Kipnis, J. Patarin, and L. Goubin, “Unbalanced oil and vinegar signature schemes,” *Proc. EUROCRYPT '99*, Lecture Notes in Computer Science, vol.1592, pp.206–222, Springer, 1999.

[KS99] A. Kipnis and A. Shamir, “Cryptanalysis of the HFE public key cryptosystem by relinearization,” *Proc. CRYPTO '99*, Lecture Notes in Computer Science, vol.1666, pp.19–30, Springer, 1999.

[Kob98] N. Koblitz, Algebraic Aspects of Cryptography, Springer, 1998.

[Kob99] N. コブリッツ著，林 彬訳，暗号の代数理論，シュプリンガー・フェアラーク東京，1999.

[KR00] M. Kreuzer and L. Robbiano, Computational Commutative Algebra 1, Springer, 2000.

[KR05] M. Kreuzer and L. Robbiano, Computational Commutative Algebra 2, Springer, 2005.

[MIHM83] 松本勉，今井秀樹，原島博，宮川洋，“暗号化変換の自明でない表現を用いる非対称暗号系，” 昭和 58 年度電子通信学会情報・システム部門全国大会講演論文集，S8-5, Sept. 1983.

[MI88a] T. Matsumoto and H. Imai, “Public quadratic polynomial-tuples for efficient signature-verification and message-encryption,” *Proc. EUROCRYPT '88*, Lecture Notes in Computer Science, vol.330, pp.419–453, Springer, 1988.

[MI88b] 松本勉，今井秀樹，“署名機能と機密保持機能を効率よく実現する多変数多項式タプル非対称暗号系の構成，” 電子情報通信学会論文誌（A）, vol.J71-A，no.7，pp.1441–1452，July 1988.

[MMDB08] M. S. E. Mohamed, W. S. A. E. Mohamed, J. Ding, and J. Buchmann, “$MXL2$: Solving polynomial equations over GF(2) using an improved mutant strategy,” *Proc. PQCrypto 2008*, Lecture Notes in Computer Science, vol.5299, pp.203–215, Springer, 2008.

[MCDBB09] M. S. E. Mohamed, D. Cabarcas, J. Ding, J. Buchmann, and S. Bulygin,

"MXL$_3$: An efficient algorithm for computing Gröbner bases of zero-dimensional ideals," *Proc. ICISC 2009*, Lecture Notes in Computer Science, vol.5984, pp.87–100, Springer, 2009.

[MP08] S. Murphy and M. B. Paterson, "A geometric view of cryptographic equation solving," Journal of Mathematical Cryptology, vol.2, no.1, pp.63–107, Apr. 2008.

[NY03] 野呂正行，横山和弘，グレブナー基底の計算　基礎篇　計算代数入門，東京大学出版会，2003.

[Pat96a] J. Patarin, "Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms," *Proc. EUROCRYPT '96*, Lecture Notes in Computer Science, vol.1070, pp.33–48, Springer, 1996.

[Pat96b] J. Patarin, "Asymmetric cryptography with a hidden monomial," *Proc. CRYPTO '96*, Lecture Notes in Computer Science, vol.1109, pp.45–60, Springer, 1996.

[PG97a] J. Patarin and L. Goubin, "Trapdoor one-way permutations and multivariate polynomials," *Proc. ICICS '97*, Lecture Notes in Computer Science, vol.1334, pp.356–368, Springer, 1997.

[PG97b] J. Patarin and L. Goubin, "Asymmetric cryptography with S-boxes," *Proc. ICICS '97*, Lecture Notes in Computer Science, vol.1334, pp.369–380, Springer, 1997.

[PGC98] J. Patarin, L. Goubin, and N. Courtois, "C^*_{-+} and HM: variations around two schemes of T. Matsumoto and H. Imai," *Proc. ASIACRYPT '98*, Lecture Notes in Computer Science, vol.1514, pp.35–49, Springer, 1998.

[PCG01] J. Patarin, N. Courtois, and L. Goubin, "QUARTZ, 128-bit long digital signatures," *Proc. CT-RSA 2001*, Lecture Notes in Computer Science, vol.2020, pp.282–297, Springer, 2001

[RS08] H. Raddum and I. Semaev, "Solving multiple right hand sides linear equations," Designs, Codes and Cryptography, vol.49, no.1-3, pp.147–160, Dec. 2008.

[Rit50] J. F. Ritt, Differential Algebra, American Mathematical Society, Colloquium Publications, vol.33, 1950.

[Saw02] 沢田浩之，"グレブナ基底計算を効率的に行うための項順序自動設定法，" 数式処理，vol.9, no.2, pp.56–77, 2002.

[Sha93] A. Shamir, "Efficient signature schemes based on birational permutations," *Proc. CRYPTO '93*, Lecture Notes in Computer Science, vol.773, pp.1–12,

Springer, 1993.

[Sho94] P. W. Shor, “Algorithms for quantum computation: Discrete logarithms and factoring,” *Proc. FOCS '94*, pp.124–134, Nov. 1994.

[Ste04] A. Steel, Allan Steel's Gröbner basis timings page, `http://magma.maths.usyd.edu.au/users/allan/gb/` (last updated 2004/10/22)

[Tsu85] 辻井重男，“非線形連立方程式の順序解法を利用する公開鍵暗号方式，” 情報理論とその応用研究会，第 8 回シンポジウム資料，pp.156–157，Dec. 1985.

[TKIFM86] 辻井重男，黒澤馨，伊東利哉，藤岡淳，松本勉，“非線形連立方程式の順序解法による公開鍵暗号方式，” 電子通信学会論文誌（D），vol.J69-D，no.12，pp.1963–1970，Dec. 1986.

[TFH89] 辻井重男，藤岡淳，平山裕介，“順序解法の一般化による公開鍵暗号系，” 電子情報通信学会論文誌（A），vol.J72-A，no.2，pp.390–397，Feb. 1989.

[Tsu03] S. Tsujii, A new structure of primitive public key cryptosystem based on soldiers in hand matrix. Technical Report TRISE 02-03, Chuo University, July 2003.

[TFT04] S. Tsujii, R. Fujita, and K. Tadaki, “Proposal of MOCHIGOMA (piece in hand) concept for multivariate type public key cryptosystem,” Technical Report of IEICE, ISEC2004-74 (2004-09), Sept. 2004.

[TTF04] S. Tsujii, K. Tadaki, and R. Fujita, “Piece in hand concept for enhancing the security of multivariate type public key cryptosystems: public key without containing all the information of secret key,” Cryptology ePrint Archive, Report 2004/366, Dec. 2004. `http://eprint.iacr.org/`

[TTF05] S. Tsujii, K. Tadaki, and R. Fujita, “Piece in hand concept for enhancing the security of multivariate type public key cryptosystems: public key without containing all the information of secret key,” *Proc. SCIS2005*, 2E1-3, pp.487–492, Jan. 2005.

[TTF06a] 辻井重男，只木孝太郎，藤田亮，“持駒行列の提案 その 2 —多変数多項式型公開鍵暗号の安全性強化のための汎用的手法—，” *Proc. SCIS2006*, 2A4-1, Jan. 2006.

[TTF06b] S. Tsujii, K. Tadaki, and R. Fujita, “Proposal for piece in hand matrix ver.2: general concept for enhancing security of multivariate public key cryptosystems,” Cryptology ePrint Archive, Report 2006/051, Feb. 2006. `http://eprint.iacr.org/`

[TTF06c] S. Tsujii, K. Tadaki, and R. Fujita, "Proposal for piece in hand matrix ver.2: general concept for enhancing security of multivariate public key cryptosystems," Workshop Record of the International Workshop on Post-Quantum Cryptography (PQCrypto 2006), pp.103–117, May 2006.

[TTF07a] S. Tsujii, K. Tadaki, and R. Fujita, "Proposal for piece in hand matrix: general concept for enhancing security of multivariate public key cryptosystems," *IEICE Transactions on Fundamentals*, vol.E90-A, no.5, pp.992–999, May 2007.

[TTF07b] 辻井重男，只木孝太郎，藤田亮，"多様な多変数公開鍵暗号を汎用的に強化する非線形持駒行列の構成法，" 電子情報通信学会技術研究報告, ISEC2007-56 (2007-07), July 2007.

[TTF08] S. Tsujii, K. Tadaki, and R. Fujita, "Nonlinear piece in hand matrix method for enhancing security of multivariate public key cryptosystems," Proceedings of the First International Conference on Symbolic Computation and Cryptography (SCC 2008), pp.124–144, 2008.

[TK08] 辻井重男，笠原正雄編著，"暗号理論と楕円曲線，" 森北出版，2008.

[WBP04] C. Wolf, A. Braeken, and B. Preneel, "Efficient cryptanalysis of RSE(2)PKC and RSSE(2)PKC," *Proc. SCN 2004*, Lecture Notes in Computer Science, vol.3352, pp.294–309, Springer, 2004.

[WP05] C. Wolf and B. Preneel, "Taxonomy of public key schemes based on the problem of Multivariate Quadratic equations," Cryptology ePrint Archive, Report 2005/077, 2005. `http://eprint.iacr.org/`

[Wol05] C. Wolf, "Multivariate Quadratic polynomials in public key cryptography," Ph.D. thesis, Katholieke Universiteit Leuven, Cryptology ePrint Archive, Report 2005/393, 2005. `http://eprint.iacr.org/`

[WBP06] C. Wolf, A. Braeken, and B. Preneel, "On the security of stepwise triangular systems," Designs, Codes and Cryptography, vol.40, no.3, pp.285–302, Sept. 2006.

[Wu78] W. T. Wu, "On the decision problem and the mechanization of theorem-proving in elementary geometry," Science in China Series A: Mathematics, vol.21, no.2, pp.159–172, 1978.

[YC04a] B. Y. Yang and J. M. Chen, "Theoretical analysis of XL over small fields," *Proc. ACISP 2004*, Lecture Notes in Computer Science, vol.3108, pp.277–288, Springer, 2004.

[YC04b] B. Y. Yang and J. M. Chen, "All in the XL family: theory and practice," *Proc. ICISC 2004*, Lecture Notes in Computer Science, vol.3506, pp.67–86, Springer, 2004.

[YCC04] B. Y. Yang, J. M. Chen, and N. Courtois, "On asymptotic security estimates in XL and Gröbner bases-related algebraic cryptanalysis," *Proc. ICICS 2004*, Lecture Notes in Computer Science, vol.3269, pp.401–413, Springer, 2004.

Ryo Fujita
Research and Development Initiative, Chuo University
1-13-27 Kasuga, Bunkyo-ku, Tokyo, 112-8551 Japan
E-mail Address: rfujita@tamacc.chuo-u.ac.jp