多変数暗号に関する故障利用攻撃について

橋本康史（ISIT）*，高木剛（九州大学），櫻井幸一（ISIT）*

* Partially supported by JST Strategic Japanese-Indian Cooperative Programme on multidisciplinary Research Field, which combines Information and Communications Technology with Other Fields, entitled "Analysis of cryptographic algorithms and evaluation on enhancing network security Based on Mathematical Science"

多変数暗号（Multivariate Pulic Key Cryptosystem）

公開鍵が多変数連立二次式から構成される公開鍵暗号方式．

$$f_1(x_1, \cdots, x_n) = \sum_{i,j} a_{ij}^{(1)} x_i x_j + \sum_i b_i^{(1)} x_i + c^{(1)},$$

$$\vdots$$

$$f_m(x_1, \cdots, x_n) = \sum_{i,j} a_{ij}^{(m)} x_i x_j + \sum_i b_i^{(m)} x_i + c^{(m)}.$$

平文を求めるためには，多変数連立二次方程式

$$f_1(x) = \cdots = f_n(x) = 0$$

を解く必要がある．

（ランダムに選ばれた）多変数連立二次方程式の求解は NP **困難**

⇓

多変数暗号は量子計算機による攻撃に耐えうる暗号
（Post Quantum Cryptology**）の候補として期待される．**

(**その他の候補：格子暗号，符号暗号** etc.)

RSA **や** ECC **と比べて高速である．**

⇓

スマートカードなどへの利用も期待できる．

Chen et al, CHES 2009.

Scheme	PubKey	SecKey	Encryp	Decryp
RSA(1024)	128B	1024B	22.4μs	813.5μs
ECDSA(160)	40B	60B	409.2μs	357.8μs
3HFE-p(31,9)	7KB	5KB	2.3μs	60.5μs
Rainbow(31,24,20,20)	57KB	150KB	17.7μs	70.6μs
TTS(31,24,20,20)	57KB	16KB	18.4μs	14.2μs

多変数暗号に対する攻撃法.

・グレブナ基底攻撃,
・ランク攻撃,
・差分攻撃,

etc.

ほとんどが理論的・実験的（理論的な計算量評価はないが計算機で解読可能）な攻撃.

物理的な攻撃に対する検討は, Okeya et al (2005) **による** Sflash **に対するサイドチャンネル攻撃を除いてほとんどない.**

本研究：多変数暗号に対する故障利用攻撃

一般的な仕組み

q：素数べき
k：位数 q の有限体
$n, m \geq 1$：自然数（n: 変数の個数，m: 二次式の個数）．

秘密鍵：$S : k^n \to k^n$：線型写像
$G : k^n \to k^m$：二次写像（G^{-1} の計算が容易な写像）
$T : k^m \to k^m$：線型写像

公開鍵：$F := T \circ G \circ S$

$$F : k^n \xrightarrow{S} k^n \xrightarrow{G} k^m \xrightarrow{T} k^m$$

暗号化： x (平文) $\mapsto F(x) = y$ (暗号文).
復号化： $y \mapsto S^{-1}(G^{-1}(T^{-1}(y))) = x$.

署名生成： y (メッセージ) $\mapsto S^{-1}(G^{-1}(T^{-1}(y), r)) = x$ (署名)
署名認証： $y = F(x)$.

r（ランダムパラメータ）

$$F : k^n \xrightarrow{S} k^n \xrightarrow{G} k^m \xrightarrow{T} k^m$$

G^{-1} は容易に計算可能だが，ランダムな線型写像 S, T をとることで，F は解読困難な一方向関数となる．

$$\Downarrow$$

攻撃の目標：S, T（の部分情報）を求める.

S, T の解読法は方式（G の構成法）によって異なる．

G の（大雑把な）分類

1. 拡大体（Big Field）型．

有限体 k の拡大体 K 上の多項式を k 上の二次式にみせかけたもの．
（松本 - 今井，HFE，Sflash，IIC **など）**

2. 順序解法（STS）型．
連立二次方程式が「少しずつ順々に」解けるように構成したもの．
（辻井の順序解法方式，UOV，Rainbow **など）**

I. Permanent Fault attack.
G を表現している多項式の係数の値を変える．

$$F : k^n \xrightarrow{S} k^n \xrightarrow{G} k^m \xrightarrow{T} k^m$$

II. Transient Fault attack.
署名方式に対する攻撃．ランダムパラメータ r の（一部）を 0 にする．

$$y \overset{T^{-1}}{\longmapsto} T^{-1}(y) \overset{G^{-1},r}{\longmapsto} G^{-1}(T^{-1}(y), r)$$
$$\overset{S^{-1}}{\longmapsto} S^{-1}(G^{-1}(T^{-1}(y), r)) =: x$$

結果

Table: 1. Our permanent fault attacks on MPKC

	Big Field	STS
#PF	1	$n-1$
$\#(x,\delta)$	$\frac{1}{2}(n+1)(n+2)$	1
Recovering	parts of S, T	a part of T

Table: 2. Our transient fault attacks on MPKC

	Big Field	STS
#TF	$\frac{1}{2}(n+1)(n+2)$	$n-u_1+1$
Recovering	hidden $g_{m-u_1+1}, \cdots, g_m$	a part of S

拡大体（Big Field）型

K：k の有限次拡大体（$N := [K : k]$）.

$$G : k^n \overset{1-1}{\to} K^{n/N} \overset{\mathcal{G}}{\to} K^{m/N} \overset{1-1}{\to} k^m$$

$\mathcal{G}$：K 上の多項式写像.

松本ー今井暗号（1984）

$$\mathcal{G}(X) = X^{q^i+1} \quad (i \geq 0).$$

$\{1, w, \cdots, w^{n-1}\}$：$K$ の k 上の基底.
$x_1, \cdots, x_n \in k$.

$$\begin{aligned} X &= x_1 + x_2 w + \cdots + x_n w^{n-1}, \\ X^q &= (x_1, \cdots, x_n\text{-linear}) + \cdots + (x_1, \cdots, x_n\text{-linear}) w^{n-1}. \\ X^{q^i+1} &= (x_1, \cdots, x_n\text{-quadratic}) + \cdots + (x_1, \cdots, x_n\text{-quadratic}) w^{n-1} \end{aligned}$$

Patarin により、G の一方向性に対する攻撃が提案された（1995）.

HFE（Patarin, 1996）

$r \geq 1$.

$$\mathcal{G}(X) = \sum_{0 \leq i,j \leq r} \alpha_{ij} X^{q^i + q^j} + \sum_{0 \leq i \leq r} \beta_i X^{q^i} + \gamma, \quad (\alpha_{ij}, \beta_i, \gamma \in K).$$

復号：K 上の方程式 $\mathcal{G}(X) = Y$ を解く．
計算量は $O(q^{2r} \times (\text{polyn.}))$.

攻撃：
・Kipnis-Shamir **の攻撃** (1999)**：秘密鍵** S, T**（の部分情報）を求める．**
・グレブナ基底攻撃（F_4**）：平文を求める．**
ともに，r **が小さい場合に解読が容易になる．**

順序解法（STS）型

$G(x) = (g_1(x), \cdots, g_m(x))$.
$1 \leq n_1 < \cdots < n_l = n$
$1 \leq m_1 < \cdots < m_l = m$

$$
\begin{aligned}
g_1(x), \cdots, g_{m_1}(x) &= (x_1, \cdots, x_{n_1} \text{ の二次式}) \\
g_{m_1+1}(x), \cdots, g_{m_2}(x) &= (x_1, \cdots, x_{n_1}, \cdots, x_{n_2} \text{ の二次式}) \\
&\vdots \\
g_{m_{l-1}+1}(x), \cdots, g_m(x) &= (x_1, \cdots, x_{n_1}, \cdots, x_{n_2}, \cdots, x_n \text{ の二次式})
\end{aligned}
$$

辻井の順序解法方式（1986）

$$g_1(x) = (x_1 \text{ の一次式}) \tag{1}$$
$$g_2(x) = (x_1 \text{ の二次式}) + x_2(x_1 \text{ の一次式}) \tag{2}$$
$$\vdots$$
$$g_n(x) = (x_1, \cdots, x_{n-1} \text{ の二次式}) + x_n(x_1, \cdots, x_{n-1} \text{ の一次式}) \tag{n}$$

復号：まず，(1) を使って，x_1 を求める．x_1 を他に代入．
次に (2) を使って，x_2 を求める．x_2 を他に代入．

$$\vdots$$

長谷川 - 金子より、G の一方向性に対する攻撃が提案された（1987）．

UOV(Patarin, 1997)

$$g_l(x) = \sum_{1 \le i \le m} x_i(x_{m+1}, \cdots, x_n \text{ の一次式}) + (x_{m+1}, \cdots, x_n \text{ の二次式})$$

$$= x^t \begin{pmatrix} 0_m & * \\ * & * \end{pmatrix} x + (\text{linear}) \quad (1 \le l \le m).$$

署名生成：

1. $x_{m+1}, \cdots, x_n$ **にランダムな値を代入する．**
2. **線型方程式を解いて，**$x_1, \cdots, x_m$ **を求める．**

攻撃：

- Kipnis-Shamir **の攻撃（**1999**）：**$O(q^{n-2m} \times (\text{polyn.}))$

$$\Downarrow$$

変数の個数が方程式の個数の 2 **倍以上必要．**

多変数暗号に対する主な攻撃法

1. **直接攻撃：グレブナ基底アルゴリズム（F_4）や XL アルゴリズムなどを用いて，直接方程式を解き，平文を求める．**

2. **ランク攻撃：二次式の係数行列の階数（ランク）が特殊な条件をみたすときに，秘密鍵 T の部分情報を求める．**

3. **差分攻撃：差分 $F(x+t)-F(x)-F(t)$ を使った，拡大体型の方式を「マイナス」や「Vinegar」などに変形したもの（MI−, HFEv, Sflash など）に対する攻撃．**

4. **UOV に対する攻撃：G が UOV と類似のときに秘密鍵 S の部分情報を求める．**

etc.

Fault attack

I. Permanent Fault attack.
G を構成する二次式の係数をひとつ変える．

$$F : k^n \xrightarrow{S} k^n \xrightarrow{G} k^m \xrightarrow{T} k^m$$

II. Transient Fault attack.
署名方式に対する攻撃．ランダムパラメータ r の（一部）を 0 にする．

$$y \overset{T^{-1}}{\longmapsto} T^{-1}(y) \overset{G^{-1},r}{\longmapsto} G^{-1}(T^{-1}(y), r)$$
$$\overset{S^{-1}}{\longmapsto} S^{-1}(G^{-1}(T^{-1}(y), r)) =: x$$

Permanent Fault attack

$$F : k^n \xrightarrow{S} k^n \xrightarrow{G} k^m \xrightarrow{T} k^m$$

拡大体型 (HFE の場合)：

$$\mathcal{G}(X) = \sum_{0 \le i,j \le r} \alpha_{ij} X^{q^i + q^j} + \sum_{0 \le i \le r} \beta_i X^{q^i} + \gamma, \quad (\text{over } K).$$

順序解法型：

$$g_1(x_1, \cdots, x_n) = \sum_{i,j} a_{ij}^{(1)} x_i x_j + \sum_i b_i^{(1)} x_i + c^{(1)},$$

$$\vdots$$

$$g_m(x_1, \cdots, x_n) = \sum_{i,j} a_{ij}^{(m)} x_i x_j + \sum_i b_i^{(m)} x_i + c^{(m)}, \quad (\text{over } k).$$

Step 1. G を故障させ，係数（α_{ij} or $a_{ij}^{(l)}$）をひとつ変える．

$$F' : k^n \xrightarrow{S} k^n \xrightarrow{G'} k^m \xrightarrow{T} k^m$$

Step 2. ランダムに選ばれた $y_1, \cdots, y_l \in k^m$ を，G' を使って復号する．

$$x_i := S^{-1}(G'^{-1}(T^{-1}(y_i))) = F'^{-1}(y_i).$$

Step 3. F を使って x_i を暗号化する．

$$z_i := F(x_i).$$

Step 4. $\delta_i := y_i - z_i$ を使って秘密鍵 S, T を求める．

$$\delta_i = y_i - z_i = (F - F')(x_i) = T \circ (G - G') \circ S(x_i)$$

$(G - G')(x)$ **は非常に "疎な" 二次式なので，秘密鍵を容易に解読できる．**

拡大体型の場合

$(\mathcal{G} - \mathcal{G}')(X) = cX^{q^i+q^j}$ **（松本 - 今井暗号とほぼ同じ形）．**

$\Downarrow$

Kipnis-Shamir **の攻撃で** S, T **を解読できる．**

順序解法型の場合

$(G - G')(x) = (0, \cdots, 0, cx_ix_j, 0, \cdots, 0)^t.$

$\Downarrow$

$\{\delta_i\}$ **の比で** T **の一部が解読できる．**

Table: 1. Our permanent fault attacks on MPKC

	Big Field	STS
#PF	1	$n-1$
$\#(x, \delta)$	$\frac{1}{2}(n+1)(n+2)$	1
Recovering	parts of S, T	a part of T

Transient Fault attack

署名生成

$$y \overset{T^{-1}}{\longmapsto} T^{-1}(y) \overset{G^{-1},r}{\longmapsto} G^{-1}(T^{-1}(y), r)$$
$$\overset{S^{-1}}{\longmapsto} S^{-1}(G^{-1}(T^{-1}(y), r)) =: x$$

Step 1. ランダムパラメータ r（の一部）を 0 にして，対応する署名を求める．

Step 2. Step 1 **を何回か繰り返す．**

Step 3. 得られた署名を使って，秘密情報（S, T）を得る．

拡大体型の「マイナス」の場合

「マイナス」によって隠された二次式を求めることが可能
(e.g. HFE$-$ $\rightarrow$ HFE)**.**

$\Downarrow$

「マイナス」をとる前の方式に対する攻撃が有効になる.

順序解法型の場合

秘密鍵 S **の一部を解読可能.**

$\Downarrow$

ランク攻撃や UOV **に対する攻撃への耐性が弱まる.**

Table: 2. Our transient fault attacks on MPKC

	Big Field	STS
#TF	$\frac{1}{2}(n+1)(n+2)$	$n-u_1+1$
Recovering	hidden $g_{m-u_1+1}, \cdots, g_m$	a part of S

（素朴な）対策

Permanent Fault Attack：

G^{-1} の計算のときに，G が壊れていないかをチェックし，壊れていると判定すれば，復号を行わないような仕組みを組み込む．（例．G の係数の総和をあらかじめ，計算しておき，復号のたびにチェックする）

Transient Fault Attack：

短時間にランダムパラメータに多くの 0 が出現したら，署名生成を行わないような仕組みを組み込む．

まとめ

・**多変数暗号に対する** Permanent Fault Attack **と** Transient Fault Attack **を提案した．これによって，拡大体型，順序解法型ともに，秘密鍵（の一部）の解読が可能になった．**

・**そのほかの物理攻撃（**e.g. **サイドチャンネル攻撃）はどうか？**

(TFA **はそのままサイドチャンネル攻撃に応用できそうだが…**)