Algebraic Cryptanalysis of the PKC'2009 Algebraic Surface Cryptosystem

Jean–Charles Faugère Pierre–Jean Spaenlehauer

UPMC – CNRS – INRIA Paris - Rocquencourt
LIP6 – SALSA team

PKC'2010 – École Normale Supérieure – Paris
2010/05/26

Post-quantum Cryptography

- Lattice-based crypto.
- Code-based crypto.
- Knapsack-based crypto.
- **Multivariate crypto.**

Post-quantum Cryptography

- Lattice-based crypto.
- Code-based crypto.
- Knapsack-based crypto.
- **Multivariate crypto.**

Multivariate crypto $\longrightarrow$ often based on the difficulty of Polynomial System Solving (HFE, UOV, ...).

Post-quantum Cryptography

- Lattice-based crypto.
- Code-based crypto.
- Knapsack-based crypto.
- **Multivariate crypto.**

Multivariate crypto $\longrightarrow$ often based on the difficulty of Polynomial System Solving (HFE, UOV, ...).

Algebraic cryptanalysis

Evaluation of the **security** of various crypto primitives by means of **algebraic tools**.

Another difficult **algebraic problem**:

Section Finding Problem

Given $S(x, y, t) \in \mathbb{F}_p[x, y, t]$, find $\mathbf{u_x}(t), \mathbf{u_y}(t) \in \mathbb{F}_p[t]$ such that

$$S(\mathbf{u_x}(t), \mathbf{u_y}(t), t) = 0.$$

Principle of **ASC**: use S as **public key** and (u_x, u_y) as **secret key**.

Another difficult **algebraic problem**:

Section Finding Problem

Given $S(x, y, t) \in \mathbb{F}_p[x, y, t]$, find $\mathbf{u_x}(t), \mathbf{u_y}(t) \in \mathbb{F}_p[t]$ such that

$$S(\mathbf{u_x}(t), \mathbf{u_y}(t), t) = 0.$$

Principle of **ASC**: use S as **public key** and (u_x, u_y) as **secret key**.

High degree polynomials, few variables
$\longrightarrow$ **short keys** ($\mathcal{O}(n)$ for a security of 2^n) !!

Another difficult **algebraic problem**:

Section Finding Problem

Given $S(x, y, t) \in \mathbb{F}_p[x, y, t]$, find $\mathbf{u_x}(t), \mathbf{u_y}(t) \in \mathbb{F}_p[t]$ such that

$$S(\mathbf{u_x}(t), \mathbf{u_y}(t), t) = 0.$$

Principle of **ASC**: use S as **public key** and (u_x, u_y) as **secret key**.

High degree polynomials, few variables
$\longrightarrow$ **short keys** ($\mathcal{O}(n)$ for a security of 2^n) !!

- **ASC**: *Akiyama/Goto/Miyake PKC'09.*
  Resistant to all known attacks.

Another difficult **algebraic problem**:

Section Finding Problem

Given $S(x, y, t) \in \mathbb{F}_p[x, y, t]$, find $\mathbf{u_x}(t), \mathbf{u_y}(t) \in \mathbb{F}_p[t]$ such that

$$S(\mathbf{u_x}(t), \mathbf{u_y}(t), t) = 0.$$

Principle of **ASC**: use S as **public key** and (u_x, u_y) as **secret key**.

High degree polynomials, few variables
$\longrightarrow$ **short keys** ($\mathcal{O}(n)$ for a security of 2^n) !!

- **ASC**: *Akiyama/Goto/Miyake PKC'09.*
  Resistant to all known attacks.
- *Akiyama/Goto 04, PQCrypto'06, SCIS'07.*
  3 **SFP-based** cryptosystems.

$\longrightarrow$ Security analysis: *Uchiyama/Tokunaga 07.*
Attacks: *Voloch 07, Iwami ASCM'08.*

Security parameters:

p: **cardinality** of the ground field $\mathbb{F}_p$.
d: **degree** of the secret section $(u_x(t), u_y(t))$.
w: **degree** in x, y of the public surface: $w = \deg_{xy}(S(x, y, t))$.

Security parameters:

p: **cardinality** of the ground field $\mathbb{F}_p$.
d: **degree** of the secret section $(u_x(t), u_y(t))$.
w: **degree** in x, y of the public surface: $w = \deg_{xy}(S(x, y, t))$.

Cryptanalysis of PKC'09 ASC

- New **algebraic attack** on the **PKC'09 version of ASC**...
- ... which relies on **Gröbner bases computations** and on **decomposition of ideals**.
- **Message recovery** attack.
- Often **faster** than the decryption algorithm !
- Breaks **recommended parameters** in **0.05 seconds** !
- **Complexity**: **quasi-linear** in the size of the secret key...
- ... and **polynomial** in all other **security parameters**: $\widetilde{\mathcal{O}}(w^7 d \log(p))$.

Outline

1. **Description** of ASC.
2. **Level 1 Attack**: deterministic.
3. **Level 2 Attack**: deterministic.
4. **Level 3 Attack**: probabilistic.
5. **Complexity analysis** of the Level 3 Attack.
6. **Experimental results**.

Notation: $g \in \mathrm{Pol}(\Gamma) \rightarrow$ the support of the polynomial g is a subset of Γ.

Security **parameters**: p, d, w.

Other public **parameters**: $\Gamma_f, \Gamma_m, \Gamma_S$.

$$m \in \mathrm{Pol}(\Gamma_m).$$

Notation: $g \in \mathrm{Pol}(\Gamma) \rightarrow$ the support of the polynomial g is a subset of Γ.

Security **parameters**: p, d, w.

Other public **parameters**: $\Gamma_f, \Gamma_m, \Gamma_S$.

$$m \in \mathrm{Pol}(\Gamma_m).$$

Encryption

$f \in_R \mathrm{Pol}(\Gamma_f)$.

$r_0, r_1 \in_R \mathrm{Pol}(\Gamma_f)$.

$s_0, s_1 \in_R \mathrm{Pol}(\Gamma_S)$.

$F_i = m + r_i S + s_i f, \quad i \in \{0, 1\}$

return $(F_0(x, y, t), F_1(x, y, t))$.

Notation: $g \in \mathrm{Pol}(\Gamma) \rightarrow$ the support of the polynomial g is a subset of Γ.

Security **parameters**: p, d, w.

Other public **parameters**: $\Gamma_f, \Gamma_m, \Gamma_S$.

$$m \in \mathrm{Pol}(\Gamma_m).$$

Encryption

$f \in_R \mathrm{Pol}(\Gamma_f)$.

$r_0, r_1 \in_R \mathrm{Pol}(\Gamma_f)$.

$s_0, s_1 \in_R \mathrm{Pol}(\Gamma_S)$.

$F_i = m + r_i S + s_i f, \quad i \in \{0, 1\}$

return $(F_0(x, y, t), F_1(x, y, t))$.

Decryption

$$\begin{aligned} h(t) &= (F_0 - F_1)(u_x, u_y, t) \\ &= (f \times (s_0 - s_1))(u_x, u_y, t). \end{aligned}$$

Factor $h(t)$ and recover a factor $\tilde{f}$ of degree $\deg(f(u_x(t), u_y(t), t))$.

$m(u_x, u_y, t) = F_0(u_x, u_y, t) \mod \tilde{f}$.

Recover m by solving a **linear system**.

Verify with a **MAC**.

Level 1 Attack (I)

substitution (need the secret key), **factorization**, **linear system**.
Can we get rid of the substitution step ?

Level 1 Attack (I)

substitution (need the secret key), **factorization**, **linear system**.
Can we get rid of the substitution step ?
Decomposition of ideals: generalization of factorization.

substitution (need the secret key), **factorization**, **linear system**.
Can we get rid of the substitution step ?
Decomposition of ideals: generalization of factorization.

$$F_i = m + r_i S + s_i f, \quad i \in \{0, 1\}.$$

Lemma (decomposition of ideals)

$$\begin{aligned} \langle F_0 - F_1, S \rangle &= \langle (s_0 - s_1) f, S \rangle \\ &= \langle s_0 - s_1, S \rangle \cap \langle f, S \rangle \end{aligned}$$

How to compute $\langle f, S \rangle$:

Level 1 Attack (I)

substitution (need the secret key), **factorization**, **linear system**.
Can we get rid of the substitution step ?
Decomposition of ideals: generalization of factorization.

$$F_i = m + r_i S + s_i f, \quad i \in \{0, 1\}.$$

Lemma (decomposition of ideals)

$$\begin{aligned} \langle F_0 - F_1, S \rangle &= \langle (s_0 - s_1) f, S \rangle \\ &= \langle s_0 - s_1, S \rangle \cap \langle f, S \rangle \end{aligned}$$

How to compute $\langle f, S \rangle$:

- **Eliminate** the variable x (Gröbner basis, resultant,...):

$$\langle F_0 - F_1, S \rangle \cap \mathbb{F}_p[y, t] = \langle Q(y, t) \rangle.$$

Level 1 Attack (I)

substitution (need the secret key), **factorization**, **linear system**.
Can we get rid of the substitution step ?
Decomposition of ideals: generalization of factorization.

$$F_i = m + r_i S + s_i f, \quad i \in \{0, 1\}.$$

Lemma (decomposition of ideals)

$$\begin{aligned} \langle F_0 - F_1, S \rangle &= \langle (s_0 - s_1) f, S \rangle \\ &= \langle s_0 - s_1, S \rangle \cap \langle f, S \rangle \end{aligned}$$

How to compute $\langle f, S \rangle$:

- **Eliminate** the variable x (Gröbner basis, resultant,...):
$$\langle F_0 - F_1, S \rangle \cap \mathbb{F}_p[y, t] = \langle Q(y, t) \rangle.$$
- **Factor** $Q(y, t) = Q_0(y, t) Q_1(y, t)$ where $\deg_y(Q_0) \geq \deg_y(Q_1)$.

substitution (need the secret key), **factorization**, **linear system**.
Can we get rid of the substitution step ?
Decomposition of ideals: generalization of factorization.

$$F_i = m + r_i S + s_i f, \quad i \in \{0, 1\}.$$

Lemma (decomposition of ideals)

$$\begin{aligned} \langle F_0 - F_1, S \rangle &= \langle (s_0 - s_1) f, S \rangle \\ &= \langle s_0 - s_1, S \rangle \cap \langle f, S \rangle \end{aligned}$$

How to compute $\langle f, S \rangle$:

- **Eliminate** the variable x (Gröbner basis, resultant,...):
$$\langle F_0 - F_1, S \rangle \cap \mathbb{F}_p[y, t] = \langle Q(y, t) \rangle.$$
- **Factor** $Q(y, t) = Q_0(y, t) Q_1(y, t)$ where $\deg_y(Q_0) \geq \deg_y(Q_1)$.
- $\langle s_0 - s_1, S \rangle = \langle F_0 - F_1, S, Q_1 \rangle$
$\langle f, S \rangle = \langle F_0 - F_1, S, Q_0 \rangle$.

Level 1 Attack (II)

$$F_i = m + r_i S + s_i f, \quad i \in \{0, 1\}.$$

Lemma

$$J = \langle f, S \rangle + \langle F_0, F_1 \rangle = \langle m, f, S \rangle.$$

Level 1 Attack (II)

$$F_i = m + r_i S + s_i f, \quad i \in \{0, 1\}.$$

Lemma

$$J = \langle f, S \rangle + \langle F_0, F_1 \rangle = \langle m, f, S \rangle.$$

Normal Form

$\mathrm{NF}_J(\cdot)$: $\mathbb{F}_p$-linear application $\mathbb{F}_p[x, y, t] \to \mathbb{F}_p[x, y, t]$.
$\mathrm{Ker}(\mathrm{NF}_J) = J$.
Can be computed when a **Gröbner basis** of J is known.

$$F_i = m + r_i S + s_i f, \quad i \in \{0, 1\}.$$

Lemma

$$J = \langle f, S \rangle + \langle F_0, F_1 \rangle = \langle m, f, S \rangle.$$

Normal Form

$\mathrm{NF}_J(\cdot)$: $\mathbb{F}_p$-linear application $\mathbb{F}_p[x, y, t] \longrightarrow \mathbb{F}_p[x, y, t]$.
$\mathrm{Ker}(\mathrm{NF}_J) = J$.
Can be computed when a **Gröbner basis** of J is known.

The **support** of $m(x, y, t)$ is known (Γ_m).

$$m = \sum_{u \in \Gamma_m} \lambda_u u.$$

$$\begin{aligned} m \in J \quad &\Rightarrow \quad \mathrm{NF}_J(m) = 0. \\ &\Rightarrow \quad \sum_{u \in \Gamma_m} \lambda_u \mathrm{NF}_J(u) = 0. \end{aligned}$$

1: **Compute** $\mathrm{GB}(\langle F_0 - F_1, S\rangle \cap \mathbb{F}_p[y,t]) = \{Q(y,t)\}$.

2: **Factor** $Q = \prod Q_i(y,t)$.
Let $Q_0(y,t) \in \mathbb{F}_p[y,t]$ be an irreducible factor with highest degree with respect to y.

3: Compute a **Gröbner basis** of the ideal $J = \langle F_0, F_1, S, Q_0\rangle$.

4: Solve the **linear system** over $\mathbb{F}_p$

$$\sum_{u \in \Gamma_m} \lambda_u \mathrm{NF}_J(u) = 0.$$

Magma code

```
R<x,y,t>:=PolynomialRing(GF(p),3,"grevlex");
Res:=Resultant(R!(F0-F1),R!X,x);
F:=Factorization(Res);
maxdeg:=Max([Degree(R!f[1],R!y) :  f in F]);
exists(Q0){f[1]:f in F| Degree(R!f[1],R!y) eq maxdeg};
J:=Ideal([R!Q0,R!X,R!F0,R!F1]);
Groebner(J);
Coeffm:=PolynomialRing(GF(p),#Lambda_m*(deg_t+1));
R2<x,y,t>:=PolynomialRing(Coeffm,3);
plaintext:=&+[Coeffm.((i-1)*(deg_t+1)+j)*
              R2!NormalForm(R!x^Lambda_m[i][1]*
              R!y^Lambda_m[i][2]*R!t^(j-1),J) :
              i in [1..#Lambda_m], j in [1..deg_t+1]];
V:=Variety(Ideal(Coefficients(plaintext)));
```

Toy example ($p = 17$, $d = 3$, $w = 5$): broken in **136** seconds.

Principle: polynomials have high degree in t and **low degree** in x, y
$\longrightarrow$ **compute in** $\mathbb{F}_p(t)[x, y]$.

Principle: polynomials have high degree in t and **low degree** in x, y
$\longrightarrow$ **compute in** $\mathbb{F}_p(t)[x, y]$.

Problem: in $\mathbb{F}_p(t)[x, y], \langle m, f, S \rangle = \mathbb{F}_p(t)[x, y]$
$\longrightarrow$ the final **linear system** has an **infinite** number of solutions.

Principle: polynomials have high degree in t and **low degree** in x, y
$\longrightarrow$ **compute in** $\mathbb{F}_p(t)[x, y]$.

Problem: in $\mathbb{F}_p(t)[x, y]$, $\langle m, f, S \rangle = \mathbb{F}_p(t)[x, y]$
$\longrightarrow$ the final **linear system** has an **infinite** number of solutions.

Solution: "deform" the ideal $\langle m, f, S \rangle$ by adding a new variable:

$$J' = \langle f, S \rangle + \langle F_0 + z, F_1 + z \rangle = \langle m + z, f, S \rangle \subset \mathbb{K}(t)[x, y, z].$$

Principle: polynomials have high degree in t and **low degree** in x, y
$\longrightarrow$ **compute in** $\mathbb{F}_p(t)[x, y]$.

Problem: in $\mathbb{F}_p(t)[x, y]$, $\langle m, f, S \rangle = \mathbb{F}_p(t)[x, y]$
$\longrightarrow$ the final **linear system** has an **infinite** number of solutions.

Solution: "deform" the ideal $\langle m, f, S \rangle$ by adding a new variable:

$$J' = \langle f, S \rangle + \langle F_0 + z, F_1 + z \rangle = \langle m + z, f, S \rangle \subset \mathbb{K}(t)[x, y, z].$$

Then apply the same **strategy**:

$$\mathrm{NF}_{J'}(m + z) = 0.$$

Solving the resulting **linear system** yields the **plaintext**.

Principle: polynomials have high degree in t and **low degree** in x, y
$\longrightarrow$ **compute in** $\mathbb{F}_p(t)[x, y]$.

Problem: in $\mathbb{F}_p(t)[x, y]$, $\langle m, f, S \rangle = \mathbb{F}_p(t)[x, y]$
$\longrightarrow$ the final **linear system** has an **infinite** number of solutions.

Solution: "deform" the ideal $\langle m, f, S \rangle$ by adding a new variable:

$$J' = \langle f, S \rangle + \langle F_0 + z, F_1 + z \rangle = \langle m + z, f, S \rangle \subset \mathbb{K}(t)[x, y, z].$$

Then apply the same **strategy**:

$$\mathrm{NF}_{J'}(m + z) = 0.$$

Solving the resulting **linear system** yields the **plaintext**.

Toy example ($p = 17$, $d = 3$, $w = 5$): broken in **74** seconds.

Level 3 Attack

Level 2 Attack is **Faster** than Level 1 Attack but... coefficients in $\mathbb{F}_p(t)$ are **big** during **intermediate computations**.

Level 2 Attack is **Faster** than Level 1 Attack but... coefficients in $\mathbb{F}_p(t)$ are **big** during **intermediate computations**.

Principle: **multi-modular approach**.

For several irreducible $P_\ell(t) \in \mathbb{F}_p[t]$:

- Compute in $\mathbb{F}_{p^{\deg(P_\ell)}}[x, y] = (\mathbb{F}_p[t]/P_\ell(t))[x, y]$.
  $\rightarrow$ yields $m(x, y, t) \mod P_\ell(t)$.
- Use the **CRT** to retrieve $m(x, y, t) = m(x, y, t) \mod \prod_\ell P_\ell(t)$.

Level 2 Attack is **Faster** than Level 1 Attack but... coefficients in $\mathbb{F}_p(t)$ are **big** during **intermediate computations**.

Principle: **multi-modular approach**.

For several irreducible $P_\ell(t) \in \mathbb{F}_p[t]$:

- Compute in $\mathbb{F}_{p^{\deg(P_\ell)}}[x, y] = (\mathbb{F}_p[t]/P_\ell(t))[x, y]$.
  $\rightarrow$ yields $m(x, y, t) \mod P_\ell(t)$.
- Use the **CRT** to retrieve $m(x, y, t) = m(x, y, t) \mod \prod_\ell P_\ell(t)$.

Toy example ($p = 17$, $d = 3$, $w = 5$): broken in **0.05** seconds.

1: Choose $n \approx \deg_t(m)\log(p)/C$ irreducible polynomials of degree $\approx C/\log(p)$ such that $\sum \deg(P_\ell) > \deg_t(m)$.
2: **for** i from 1 to n **do**
3: $\quad \mathbb{K} = \mathbb{F}_p[t]/(P_\ell)$.
4: $\quad$ Compute $\mathrm{Res}_x(F_0 - F_1, S) \in \mathbb{K}[y]$.
5: $\quad$ Factor $\mathrm{Res}_x(F_0 - F_1, S)$.
Let $Q_0(y) \in \mathbb{K}[y]$ be an irreducible factor of highest degree in y.
6: $\quad$ Compute a GB of the ideal
$J' = \langle F_0 + z, F_1 + z, S, Q_0 \rangle \subset \mathbb{K}[x, y, z]$.
7: $\quad$ Solve the linear system over $\mathbb{K}$:

$$\mathrm{NF}_{J'}(z) + \sum_{(i,j)\in\Lambda_m} m_{ij}(t)\mathrm{NF}_{J'}(x^i y^j) = 0.$$

8: $\quad$ Retrieve $m \mod P_\ell = \sum_{(i,j)\in\Lambda_m} m_{ij}(t)x^i y^j$.
9: **end for**
10: Use the CRT to get $m = m \mod \prod P_\ell$.

Complexity of the Level 3 Attack

- **Number of loops**: $wd \log(p)/C$.

Complexity of the Level 3 Attack

- **Number of loops**: $wd \log(p)/C$.
  - Computation of the **resultant**: $\mathcal{O}(w^3)$.

Complexity of the Level 3 Attack

- **Number of loops**: $wd \log(p)/C$.
  - Computation of the **resultant**: $\mathcal{O}(w^3)$.
  - **Factorization** (Cantor-Zassenhaus algorithm): $\widetilde{\mathcal{O}}(w^4 + w^2 C)$.

Complexity of the Level 3 Attack

- **Number of loops**: $wd \log(p)/C$.
  - Computation of the **resultant**: $\mathcal{O}(w^3)$.
  - **Factorization** (Cantor-Zassenhaus algorithm): $\widetilde{\mathcal{O}}(w^4 + w^2 C)$.
  - **Gröbner basis** computation (Faugère F_4/F_5): $\mathcal{O}(w^6)$ (degree of regularity estimated with the Macaulay bound)).

Complexity of the Level 3 Attack

- **Number of loops**: $wd \log(p)/C$.
  - Computation of the **resultant**: $\mathcal{O}(w^3)$.
  - **Factorization** (Cantor-Zassenhaus algorithm): $\widetilde{\mathcal{O}}(w^4 + w^2 C)$.
  - **Gröbner basis** computation (Faugère F_4/F_5): $\mathcal{O}(w^6)$ (degree of regularity estimated with the Macaulay bound)).
- **CRT**: $\widetilde{\mathcal{O}}(wd \log(p)/C)$.

- **Number of loops**: $wd\log(p)/C$.
  - Computation of the **resultant**: $\mathcal{O}(w^3)$.
  - **Factorization** (Cantor-Zassenhaus algorithm): $\widetilde{\mathcal{O}}(w^4 + w^2 C)$.
  - **Gröbner basis** computation (Faugère F_4/F_5): $\mathcal{O}(w^6)$ (degree of regularity estimated with the Macaulay bound)).
- **CRT**: $\widetilde{\mathcal{O}}(wd\log(p)/C)$.

Theorem

The total **binary complexity** of the Level 3 Attack is upper bounded by:

$$\widetilde{\mathcal{O}}(dw^7\log(p)).$$

$\longrightarrow$ quasi-linear in $d\log(p)$ which is the size of the **secret key**.

Experimental results (I) – increasing d and p

p	d	w	size of public key	size of secret key	t_{res}	t_{fact}	t_{GB}	t_{total}	security bound
2	**50**	**5**	**310 bits**	**102 bits**	**0.02s**	**0.02s**	**0.01s**	**0.05s**	$\mathbf{2^{102}}$
2	100	5	560 bits	202 bits	0.03s	0.02s	0.02s	0.07s	2^{202}
2	400	5	2060 bits	802 bits	0.1s	0.1s	0.1s	0.30s	2^{802}
2	1600	5	8060 bits	3202 bits	0.3s	0.3s	0.4s	1.0s	2^{3202}
2	5000	5	25060 bits	10002 bits	0.8s	1.3s	0.8s	3.0s	2^{10002}
17	50	5	1267 bits	409 bits	0.2s	2.4s	0.4s	3.0s	2^{409}
17	400	5	8420 bits	3270 bits	1.45s	27.7s	3.9s	33.1s	2^{3270}
17	800	5	16595 bits	6500 bits	3.1s	70s	9.5s	83s	2^{6500}
10007	500	5	34019 bits	13289 bits	29s	217s	64s	310s	2^{13289}

Experimental results (II) – increasing w

p	d	w	size of public key	size of secret key	t_{res}	t_{fact}	t_{GB}	t_{LinSys}	t_{total}	security bound
2	**50**	**5**	**310 bits**	**102 bits**	**0.02s**	**0.02s**	**0.01s**	**0.001s**	**0.05s**	$\mathbf{2^{102}}$
2	50	15	810 bits	102 bits	0.7s	0.3s	4.4s	0.03s	5.4s	2^{102}
2	50	25	1310 bits	102 bits	3s	1s	32s	0.2s	37s	2^{102}
2	50	35	1810 bits	102 bits	10s	3s	260s	1s	274s	2^{102}
2	50	45	2310 bits	102 bits	30s	7s	1352s	4s	1393s	2^{102}
2	50	55	2810 bits	102 bits	70s	12s	4619s	13s	4714s	2^{102}
2	50	65	3310 bits	102 bits	147s	22s	12408s	27s	12604s	2^{102}
2	50	75	3810 bits	102 bits	288s	38s	37900s	56s	38280s	2^{102}

Conclusion

- Description of the underlying **algebraic structure**.
- **Algebraic cryptanalysis** of ASC by using tools from **Computer Algebra** (Gröbner bases, resultants, efficient CRT, decomposition of ideals, ...).
- Breaks the **recommended parameters** in **0.05 seconds**.
- Often faster than the **legal decryption algorithm**.

Perspectives

- Still no efficient algorithm to solve the **Section Finding Problem (SFP)**.

→ **SFP-based** multivariate crypto ?
Signature ?
Authentification ?
...