Multivariate Public Key Cryptography

Jintai Ding

University of Cincinnati & Southern Chinese University of technology

Feb. 2, 2011

Outline

Happy Chinese New Year!

Thank the organizers, in particular, (Alex Myasnikov)^2.

Thank the generous support of the Charles Phelps Taft Memorial Fund and the Taft Family.

Outline

PKC and Quantum computer

- 2519590847565789349402718324004839857142928212620403202777713783604366202070759555626401852588078440691829064124951508218929855914917618450280848912007284499268739280728777673597141834727026189637501497182469116507761337985909570009733045974880842840179742910064245869181719511874612151517265463228221686998754918242243363725908514186546204357679842338718477444792073993423658482382428119816381501067481045166037730605620161967625613384414360383390441495263443219011465754445417842402092461651572335077870774981712577246796292638635637328991215483143816789988504044536402352738195137863656439121201039712282212072035

What is this number?

PKC and Quantum computer

- The number for Microsoft updates

PKC and Quantum computer

- The number for Microsoft updates
- Digital signature based on RSA

PKC and Quantum computer

- Mathematics behind: integer factorization

$$n = pq.$$

$$15 = 3 \times 5.$$

PKC and Quantum computer

- Mathematics behind: integer factorization

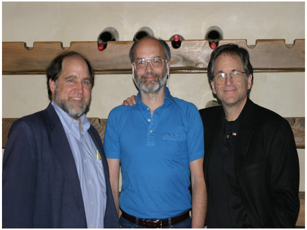
$$n = pq.$$

$$15 = 3 \times 5.$$

- The concept behind:

Public key Cryptography

PKC and Quantum computer

-  RSA – 2003 Turing prize

Diffie-Hellman – inventors of the idea of PKC

PKC

- What is PKC?

PKC

- What is PKC?
- Traditionally the information is symmetric.

PKC

- What is PKC?
- Traditionally the information is symmetric.
- PKC is asymmetric

PKC

- What is PKC?
- Traditionally the information is symmetric.
- PKC is asymmetric
- There are two sets of keys, one public and one private

PKC

- What is PKC?
- Traditionally the information is symmetric.
- PKC is asymmetric
- There are two sets of keys, one public and one private
- Encryption: Public is for encryption and private for decryption

PKC

- What is PKC?
- Traditionally the information is symmetric.
- PKC is asymmetric
- There are two sets of keys, one public and one private
- Encryption: Public is for encryption and private for decryption
- Digital Signature: Public is for verification and private for signing

PKC

- What is PKC?
- Traditionally the information is symmetric.
- PKC is asymmetric
- There are two sets of keys, one public and one private
- Encryption: Public is for encryption and private for decryption
- Digital Signature: Public is for verification and private for signing
- RSA: n is public and p,q is private.

PKC

- What is PKC?
- Traditionally the information is symmetric.
- PKC is asymmetric
- There are two sets of keys, one public and one private
- Encryption: Public is for encryption and private for decryption
- Digital Signature: Public is for verification and private for signing
- RSA: n is public and p,q is private.
- One knows how to factor n, one can defeat RSA

PKC and Quantum computer

- Quantum computer: using basic particles and quantum mechanics principles to perform computations


R. Feynman

PKC and Quantum computer

- Quantum computer: using basic particles and quantum mechanics principles to perform computations

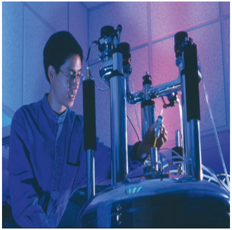
R. Feynman

- In 1995, Peter Shor at IBM showed theoretically that it can solve a family of mathematical problems including factoring.

PKC and Quantum computer

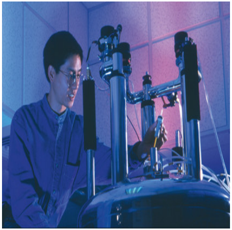
- Can quantum computer really work?

PKC and Quantum computer

- Can quantum computer really work?
- Isaac Chuang
  15 million dollars to show that

$$15 = 3 \times 5.$$

PKC and Quantum computer

- Can quantum computer really work?
- Isaac Chuang
  15 million dollars to show that
  $$15 = 3 \times 5.$$
- The problem of scaling
  People have different opinions.

PKC and Quantum computer

- PQC – to prepare for the future of quantum computer world

PKC and Quantum computer

- PQC – to prepare for the future of quantum computer world
- Ubiquitous computing world .

PQC

- **Post-quantum cryptography**
  Public key cryptosystems that potentially could resist the future quantum computer attacks. Currently there are 4 main families.

PQC

- **Post-quantum cryptography**
  Public key cryptosystems that potentially could resist the future quantum computer attacks. Currently there are 4 main families.
- 1)Code-based public key cryptography
  Error correcting codes

PQC

- **Post-quantum cryptography**
  Public key cryptosystems that potentially could resist the future quantum computer attacks. Currently there are 4 main families.
- 1)Code-based public key cryptography
  Error correcting codes
- 2) Hash-based public key cryptography
  Hash-tree construction

PQC

- **Post-quantum cryptography**
  Public key cryptosystems that potentially could resist the future quantum computer attacks. Currently there are 4 main families.
- 1)Code-based public key cryptography
  Error correcting codes
- 2) Hash-based public key cryptography
  Hash-tree construction
- 3) Lattice-based public key cryptography
  Shortest and nearest vector problems

PQC

- **Post-quantum cryptography**
  Public key cryptosystems that potentially could resist the future quantum computer attacks. Currently there are 4 main families.
- 1)Code-based public key cryptography
  Error correcting codes
- 2) Hash-based public key cryptography
  Hash-tree construction
- 3) Lattice-based public key cryptography
  Shortest and nearest vector problems
- 4) Multivariate Public Key Cryptography

What is a MPKC?

- Multivariate Public Key Cryptosystems
  - *Cryptosystems, whose public keys are a set of multivariate functions*

What is a MPKC?

- Multivariate Public Key Cryptosystems
  - *Cryptosystems, whose public keys are a set of multivariate functions*
- The public key is given as:

$$G(x_1, ..., x_n) = (G_1(x_1, ..., x_n), ..., G_m(x_1, ..., x_n)).$$

  Here the G_i are multivariate $(x_1, ..., x_n)$ polynomials over a finite field.

Encryption

- Any plaintext $M = (x_1', ..., x_n')$ has the ciphertext:

$$G(M) = G(x_1', ..., x_n') = (y_1', ..., y_m').$$

Encryption

- Any plaintext $M = (x'_1, ..., x'_n)$ has the ciphertext:

$$G(M) = G(x'_1, ..., x'_n) = (y'_1, ..., y'_m).$$

- To decrypt the ciphertext $(y'_1, ..., y'_n)$, one needs to know a secret (**the secret key**), so that one can invert the map to find the plaintext $(x'_1, ..., x'_n)$.

Toy example

- We use the finite field $k = GF[2]/(x^2 + x + 1)$ with 2^2 elements.

Toy example

- We use the finite field $k = GF[2]/(x^2 + x + 1)$ with 2^2 elements.
- We denote the elements of the field by the set $\{\mathit{0}, \mathit{1}, \mathit{2}, \mathit{3}\}$ to simplify the notation.
  Here $\mathit{0}$ represent the 0 in k, $\mathit{1}$ for 1, $\mathit{2}$ for x, and $\mathit{3}$ for $1 + x$.
  In this case, $\mathit{1} + \mathit{3} = \mathit{2}$ and $\mathit{2} * \mathit{3} = \mathit{1}$.

A toy example

- $$\begin{aligned} G_0(x_1, x_2, x_3) &= 1 + x_2 + 2x_0x_2 + 3x_1^2 + 3x_1x_2 + x_2^2 \\ G_1(x_1, x_2, x_3) &= 1 + 3x_0 + 2x_1 + x_2 + x_0^2 + x_0x_1 + 3x_0x_2 + x_1^2 \\ G_2(x_1, x_2, x_3) &= 3x_2 + x_0^2 + 3x_1^2 + x_1x_2 + 3x_2^2 \end{aligned}$$

A toy example

- $$\begin{aligned} G_0(x_1,x_2,x_3) = & \quad 1 + x_2 + 2x_0x_2 + 3x_1^2 + 3x_1x_2 + x_2^2 \\ G_1(x_1,x_2,x_3) = & \quad 1 + 3x_0 + 2x_1 + x_2 + x_0^2 + x_0x_1 + 3x_0x_2 + x_1^2 \\ G_2(x_1,x_2,x_3) = & \quad 3x_2 + x_0^2 + 3x_1^2 + x_1x_2 + 3x_2^2 \end{aligned}$$

- For example, if the plaintext is: $x_0 = 1$, $x_1 = 2$, $x_2 = 3$, then we can plug into G_1, G_2 and G_3 to get the ciphertext $y_0 = 0$, $y_1 = 0$, $y_2 = 1$.

A toy example

- $$\begin{aligned} G_0(x_1, x_2, x_3) &= \quad 1 + x_2 + 2x_0x_2 + 3x_1^2 + 3x_1x_2 + x_2^2 \\ G_1(x_1, x_2, x_3) &= \quad 1 + 3x_0 + 2x_1 + x_2 + x_0^2 + x_0x_1 + 3x_0x_2 + x_1^2 \\ G_2(x_1, x_2, x_3) &= \quad 3x_2 + x_0^2 + 3x_1^2 + x_1x_2 + 3x_2^2 \end{aligned}$$
- For example, if the plaintext is: $x_0 = 1$, $x_1 = 2$, $x_2 = 3$, then we can plug into G_1, G_2 and G_3 to get the ciphertext $y_0 = 0$, $y_1 = 0$, $y_2 = 1$.
- This is a bijective map and we can invert it easily. This example is based on the Matsumoto-Imai cryptosystem.

Theoretical Foundation

- Direct attack is to solve the set of equations:

$$G(M) = G(x_1, ..., x_n) = (y_1', ..., y_m').$$

Theoretical Foundation

- Direct attack is to solve the set of equations:

$$G(M) = G(x_1, ..., x_n) = (y'_1, ..., y'_m).$$

- *- Solving a set of n randomly chosen equations (nonlinear) with n variables is NP-complete, though this does not necessarily ensure the security of the systems.*

Quadratic Constructions

- *1) Efficiency considerations lead to mainly quadratic constructions.*

$$G_l(x_1, ..x_n) = \sum_{i,j} \alpha_{lij} x_i x_j + \sum_i \beta_{li} x_i + \gamma_l.$$

Quadratic Constructions

- *1) Efficiency considerations lead to mainly quadratic constructions.*

$$G_l(x_1, ..x_n) = \sum_{i,j} \alpha_{lij} x_i x_j + \sum_i \beta_{li} x_i + \gamma_l.$$

- *2) Mathematical structure consideration: Any set of high degree polynomial equations can be reduced to a set of quadratic equations.*

$$x_1 x_2 x_3 = 5,$$

is equivalent to

$$\begin{aligned} x_1 x_2 - y &= 0 \\ y x_3 &= 5. \end{aligned}$$

The view from the history of Mathematics(Diffie in Paris)

- RSA – Number Theory – the 18th century mathematics

The view from the history of Mathematics(Diffie in Paris)

- RSA – Number Theory – the 18th century mathematics
- ECC – Theory of Elliptic Curves – the 19th century mathematics

The view from the history of Mathematics(Diffie in Paris)

- RSA – Number Theory – the 18th century mathematics
- ECC – Theory of Elliptic Curves – the 19th century mathematics
- Multivariate Public key cryptosystem – Algebraic Geometry – the 20th century mathematics
  Algebraic Geometry – Theory of Polynomial Rings

Early works

- Early attempts by Diffie, Fell, Tsujii, Matsumoto, Imai, Ong, Schnorr, Shamir etc

Early works

- Early attempts by Diffie, Fell, Tsujii, Matsumoto, Imai, Ong, Schnorr, Shamir etc
- Fast development in the late 1990s.

Outline

Multivariate Signature schemes

- **Public key**:
  $G(x_1, \ldots, x_n) = (g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$.

Multivariate Signature schemes

- **Public key**:
  $G(x_1, \ldots, x_n) = (g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$.
- **Private key**: a way to compute G^{-1}.

Multivariate Signature schemes

- **Public key**:
  $G(x_1, \ldots, x_n) = (g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$.
- **Private key**: a way to compute G^{-1}.

- **Signing a hash of a document**:

Multivariate Signature schemes

- **Public key**:
  $G(x_1, \ldots, x_n) = (g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$.
- **Private key**: a way to compute G^{-1}.

- **Signing a hash of a document**:
  $(x_1, \ldots, x_n) \in G^{-1}(y_1, \ldots, y_m)$.

Multivariate Signature schemes

- **Public key**:
  $G(x_1, \dots, x_n) = (g_1(x_1, \dots, x_n), \dots, g_m(x_1, \dots, x_n))$.
- **Private key**: a way to compute G^{-1}.

- **Signing a hash of a document**:
  $(x_1, \dots, x_n) \in G^{-1}(y_1, \dots, y_m)$.
- **Verifying**: $(y_1, \dots, y_m) \stackrel{?}{=} G(x_1, \dots, x_n)$.

k, a small finite field.

A toy example over GF(3)

$$
\begin{aligned}
G_1(x_1, x_2, x_3) &= 1 + x_3 + x_1x_2 + x_3^2 \\
G_2(x_1, x_2, x_3) &= 2 + x_1 + 2x_2x_3 + x_2 \\
G_3(x_1, x_2, x_3) &= 1 + x_2 + x_1x_3 + x_1^2
\end{aligned}
$$

Hash:

$(y_1, y_2, y_3) = (0, 1, 1)$.

A toy example over GF(3)

$$
\begin{aligned}
G_1(x_1, x_2, x_3) &= 1 + x_3 + x_1x_2 + x_3^2 \\
G_2(x_1, x_2, x_3) &= 2 + x_1 + 2x_2x_3 + x_2 \\
G_3(x_1, x_2, x_3) &= 1 + x_2 + x_1x_3 + x_1^2
\end{aligned}
$$

Hash:

$(y_1, y_2, y_3) = (0, 1, 1)$.

A signature: $(x_1, x_2, x_3) = (2, 0, 1)$

A toy example over GF(3)

$$
\begin{aligned}
G_1(x_1, x_2, x_3) &= 1 + x_3 + x_1x_2 + x_3^2 \\
G_2(x_1, x_2, x_3) &= 2 + x_1 + 2x_2x_3 + x_2 \\
G_3(x_1, x_2, x_3) &= 1 + x_2 + x_1x_3 + x_1^2
\end{aligned}
$$

Hash: $(y_1, y_2, y_3) = (0, 1, 1)$.

A signature: $(x_1, x_2, x_3) = (2, 0, 1)$

$$
\begin{aligned}
G_1(2, 0, 1) &= 1 + 1 + 2 \times 0 + 1 = 0 \\
G_2(2, 0, 1) &= 2 + 2 + 2 \times 0 \times 1 + 0 = 1 \\
G_3(2, 0, 1) &= 1 + 0 + 2 \times 1 + 1 = 1
\end{aligned}
$$

Security: polynomial solving.

- Signature for $(y_1, y_2, y_3) = (0, 0, 0)$?

Security: polynomial solving.

- Signature for $(y_1, y_2, y_3) = (0, 0, 0)$?

$$
\begin{aligned}
G_1(x_1, x_2, x_3) &= 1 + x_3 + x_1 x_2 + x_3^2 = 0 \\
G_2(x_1, x_2, x_3) &= 2 + x_1 + 2x_2 x_3 + x_2 = 0 \\
G_3(x_1, x_2, x_3) &= 1 + x_2 + x_1 x_3 + x_1^2 = 0
\end{aligned}
$$

Security: polynomial solving.

- Signature for $(y_1, y_2, y_3) = (0, 0, 0)$?

$$
\begin{aligned}
G_1(x_1, x_2, x_3) &= 1 + x_3 + x_1x_2 + x_3^2 = 0 \\
G_2(x_1, x_2, x_3) &= 2 + x_1 + 2x_2x_3 + x_2 = 0 \\
G_3(x_1, x_2, x_3) &= 1 + x_2 + x_1x_3 + x_1^2 = 0
\end{aligned}
$$

- Direct attack: difficulty of solving a set of nonlinear polynomial equations over a finite field.

How to construct G?

- A scheme by Kipnis, Patarin and Goubin 1999. (Eurocrypt 1999)

How to construct G?

- A scheme by Kipnis, Patarin and Goubin 1999. (Eurocrypt 1999)
- $G = F \circ L$.
  F: nonlinear, easy to compute F^{-1}.
  L: invertible linear, to hide the structure of F.

Unbalanced Oil-vinegar (uov) schemes

- $F = (f_1(x_1, .., x_o, x'_1, ..., x'_v), \cdots, f_o(x_1, .., x_o, x'_1, ..., x'_v))$.

Unbalanced Oil-vinegar (uov) schemes

- $F = (f_1(x_1, .., x_o, x'_1, ..., x'_v), \cdots, f_o(x_1, .., x_o, x'_1, ..., x'_v))$.
- $$f_l(x_1, ., x_o, x'_1, ., x'_v) = \sum a_{lij} x_i x'_j + \sum b_{lij} x'_i x'_j + \sum c_{li} x_i + \sum d_{li} x'_i + e_l$$

  Oil variables: $x_1, ..., x_o$.

  Vinegar variables: $x'_1, ..., x'_v$.

How to invert F?

$$f_l(x_1, .., x_o, \underbrace{x'_1, .., x'_v}_{\textbf{fix the values}}) =$$

$$\sum a_{lij} x_i x'_j + \sum b_{lij} x'_i x'_j + \sum c_{li} x_i + \sum d_{li} x'_i + e_l.$$

How to invert F?

$$f_l(x_1, .., x_o, x'_1, .., x'_v) =$$
$$\sum a_{lij} x_i x'_j + \sum b_{lij} x'_i x'_j + \sum c_{li} x_i + \sum d_{li} x'_i + e_l.$$

How to invert F?

$$f_l(x_1, .., x_o, x'_1, .., x'_v) = \\ \sum a_{lij} x_i x'_j + \sum b_{lij} x'_i x'_j + \sum c_{li} x_i + \sum d_{li} x'_i + e_l.$$

- F: linear in Oil variables: $x_1, .., x_o$.

  $\Longrightarrow F$: easy to invert.

Security analysis

- $v = o$ and $v >> o$ not secure

Security analysis

- $v = o$ and $v >> o$ not secure
- $v = 2o, 3o$

Security analysis

- $v = o$ and $v >> o$ not secure
- $v = 2o, 3o$
- Direct attacks does not work.

Security analysis

- The mathematical problem to find equivalent secret keys — find the common null subspace spaces of a set of quadratic forms.

$$\begin{array}{cccccc}
0 & .. & 0 & * & .. & * \\
. & . & 0 & * & .. & * \\
. & . & 0 & * & .. & * \\
. & . & . & * & .. & * \\
0 & .. & 0 & * & .. & * \\
* & .. & * & * & .. & * \\
* & .. & * & * & .. & *
\end{array}$$

Security analysis

- The mathematical problem to find equivalent secret keys — find the common null subspace spaces of a set of quadratic forms.

$$\begin{matrix} 0 & .. & 0 & * & .. & * \\ . & . & 0 & * & .. & * \\ . & . & 0 & * & .. & * \\ . & . & . & * & .. & * \\ 0 & .. & 0 & * & .. & * \\ * & .. & * & * & .. & * \\ * & .. & * & * & .. & * \end{matrix}$$

- The problem above can also be transformed into solving a set of quadratic equations.

Rainbow – Ding, Schmidt, Yang etc – ACNS 05,06,07,08

- Make F "small" without reducing security.

Rainbow – Ding, Schmidt, Yang etc – ACNS 05,06,07,08

- Make F "small" without reducing security.

$$G = \underbrace{L_1}_{\textbf{Hide the separation}} \circ F \circ \underbrace{L_2}_{\textbf{Hide } L_1 \circ F}.$$

$$F = (F_1, F_2).$$

Rainbow – Ding, Schmidt, Yang etc – ACNS 05,06,07,08

- Make F "small" without reducing security.

$$G = \underbrace{L_1}_{\textbf{Hide the separation}} \circ F \circ \underbrace{L_2}_{\textbf{Hide } L_1 \circ F}.$$

$$F = (F_1, F_2).$$

- Rainbow(18,12,12) over GF(2^8).

$F_1 : o_1 = 12, v_1 = 18.$
$F_2 : o_2 = 12, v_2 = 12 + 18 = 30.$

Rainbow

- Rainbow(18,12,12)

Rainbow

- Rainbow(18,12,12)
-

Signature 400 bits	**Hash 336 bits**

Implementations

- IC for Rainbow: 804 cycles
  A joint work of Cincinnati and Bochum.(ASAP 2008)

Implementations

- IC for Rainbow: 804 cycles
  A joint work of Cincinnati and Bochum.(ASAP 2008)

- FPGA implementation by the research group of Professor Paar at Bochum (CHES 2009)
  Beat ECC in area and speed.

Side channel attack on Rainbow

- Natural Side channel attack resistance.

Side channel attack on Rainbow

- Natural Side channel attack resistance.
- Further optimizations.

Side channel attack on Rainbow

- Natural Side channel attack resistance.
- Further optimizations.
- Real implementations — Yang and Cheng In Taiwan.

Side channel attack on Rainbow

- Natural Side channel attack resistance.
- Further optimizations.
- Real implementations — Yang and Cheng In Taiwan.
- **A good candidate for light-weight crypto for small devices like RFID.**

Security

Security

- UOV: not broken since 1999.
- Rainbow – MinRank problem

Pros and Cons

Pros and Cons

- Computationally very efficient
- Large public key size

Outline

Notation

- k is a small finite field with $|k| = q$

Notation

- k is a small finite field with $|k| = q$
- $\bar{K} = k[x]/(g(x))$, a degree n extension of k.

Notation

- k is a small finite field with $|k| = q$
- $\bar{K} = k[x]/(g(x))$, a degree n extension of k.
- The standard k-linear invertible map $\phi : \bar{K} \longrightarrow k^n$, and $\phi^{-1} : k^n \longrightarrow \bar{K}$.

The idea of "BIG" field

- Proposed in 1988 by Matsumoto-Imai.

The idea of "BIG" field

- Proposed in 1988 by Matsumoto-Imai.
- Build up a map F over $\bar{K}$:

$$\bar{F} = L_1 \circ \phi \circ F \circ \phi^{-1} \circ L_2.$$

where the L_i are randomly chosen invertible affine maps over k^n

The idea of "BIG" field

- Proposed in 1988 by Matsumoto-Imai.
- Build up a map F over $\bar{K}$:

$$\bar{F} = L_1 \circ \phi \circ F \circ \phi^{-1} \circ L_2.$$

  where the L_i are randomly chosen invertible affine maps over k^n
- The L_i are used to "hide" $\bar{F}$.

The idea of "BIG" field

- Proposed in 1988 by Matsumoto-Imai.
- Build up a map F over $\bar{K}$:

$$\bar{F} = L_1 \circ \phi \circ F \circ \phi^{-1} \circ L_2.$$

  where the L_i are randomly chosen invertible affine maps over k^n
- The L_i are used to "hide" $\bar{F}$.
- IP problem.

Encryption

- The MI construction:

$$F : X \longmapsto X^{q^\theta + 1}.$$

Encryption

- The MI construction:

$$F : X \longmapsto X^{q^\theta+1}.$$

- Let $\tilde{F}(x_1, \ldots, x_n) = \phi \circ F \circ \phi^{-1}(x_1, \ldots, x_n) = (\tilde{F}_1, \ldots, \tilde{F}_n)$.

Encryption

- The MI construction:

$$F : X \longmapsto X^{q^\theta+1}.$$

- Let $\tilde{F}(x_1, \ldots, x_n) = \phi \circ F \circ \phi^{-1}(x_1, \ldots, x_n) = (\tilde{F}_1, \ldots, \tilde{F}_n)$.
- The $\tilde{F}_i = \tilde{F}_i(x_1, \ldots, x_n)$ are quadratic polynomials in n variables. Why quadratic?

$$X^{q^\theta+1} = X^{q^\theta} \times X.$$

Decryption

- The condition: $\gcd\left(q^{\theta}+1, q^{n}-1\right)=1$, ensures the invertibility of the map for purposes of decryption.
  It requires that k must be of characteristic 2.

Decryption

- The condition: $\gcd\left(q^{\theta}+1, q^{n}-1\right)=1$, ensures the invertibility of the map for purposes of decryption.
  It requires that k must be of characteristic 2.
- $F^{-1}(X)=X^{t}$ such that:

$$t \times\left(q^{\theta}+1\right) \equiv 1 \pmod{q^{n}-1}.$$

Decryption

- The condition: $\gcd(q^\theta + 1, q^n - 1) = 1$, ensures the invertibility of the map for purposes of decryption.
  It requires that k must be of characteristic 2.
- $F^{-1}(X) = X^t$ such that:

$$t \times (q^\theta + 1) \equiv 1 \pmod{q^n - 1}.$$

- The public key includes the field structure of k, θ and $\bar{F} = (\bar{F}_1, .., \bar{F}_n)$. The secret keys are L_1 and L_2.

Decryption

- The condition: $\gcd(q^\theta + 1, q^n - 1) = 1$, ensures the invertibility of the map for purposes of decryption. It requires that k must be of characteristic 2.
- $F^{-1}(X) = X^t$ such that:

$$t \times (q^\theta + 1) \equiv 1 \pmod{q^n - 1}.$$

- The public key includes the field structure of k, θ and $\bar{F} = (\bar{F}_1, .., \bar{F}_n)$. The secret keys are L_1 and L_2.
- The first toy example is produced by setting $n = 3$ and $\theta = 2$.

Decryption

- The condition: $\gcd(q^\theta + 1, q^n - 1) = 1$, ensures the invertibility of the map for purposes of decryption.
  It requires that k must be of characteristic 2.
- $F^{-1}(X) = X^t$ such that:

$$t \times (q^\theta + 1) \equiv 1 \pmod{q^n - 1}.$$

- The public key includes the field structure of k, θ and $\bar{F} = (\bar{F}_1, .., \bar{F}_n)$. The secret keys are L_1 and L_2.
- The first toy example is produced by setting $n = 3$ and $\theta = 2$.
- This scheme is defeated by linearization equation method by Patarin 1995.

HFE by Patarin etc

- The only difference from MI is that F is replaced by a new map given by:

$$F(X) = \sum_{i,j=0}^{D} a_{ij} X^{q^i + q^j} + \sum_{i=0}^{D} b_i X^{q^i} + c.$$

HFE by Patarin etc

- The only difference from MI is that F is replaced by a new map given by:

$$F(X) = \sum_{i,j=0}^{D} a_{ij} X^{q^i + q^j} + \sum_{i=0}^{D} b_i X^{q^i} + c.$$

- Due to the work of Kipnis and Shamir, Faugere, Joux, D cannot be too small. Therefore, the system is much slower.

HFE by Patarin etc

- The only difference from MI is that F is replaced by a new map given by:

$$F(X) = \sum_{i,j=0}^{D} a_{ij} X^{q^i + q^j} + \sum_{i=0}^{D} b_i X^{q^i} + c.$$

- Due to the work of Kipnis and Shamir, Faugere, Joux, D cannot be too small. Therefore, the system is much slower.
- D can not too large due to the inversion of using Berlakemp algorithms of solving one variable equations.

Internal Perturbation

- (Internal) Perturbation was introduced at PKC 2004 as a general method to improve the security of multivariate public key cryptosystems.

Internal Perturbation

- (Internal) Perturbation was introduced at PKC 2004 as a general method to improve the security of multivariate public key cryptosystems.
- Construction – small-scale "noise" is added to the system in a controlled way so as to not fundamentally alter the main structure, but yet substantially increase the "entropy."

Internal Perturbation

- Let r be a small integer and

$$z_1(x_1, \ldots, x_n) = \sum_{j=1}^{n} \alpha_{j1} x_j + \beta_1$$

$$\vdots$$

$$z_r(x_1, \ldots, x_n) = \sum_{j=1}^{n} \alpha_{jr} x_j + \beta_r$$

be a set of randomly chosen affine linear functions in the x_i over k^n such that the $z_j - \beta_j$ are linearly independent.

Internal Perturbation

- Let r be a small integer and

$$z_1(x_1,\ldots,x_n) = \sum_{j=1}^{n} \alpha_{j1} x_j + \beta_1$$

$$\vdots$$

$$z_r(x_1,\ldots,x_n) = \sum_{j=1}^{n} \alpha_{jr} x_j + \beta_r$$

  be a set of randomly chosen affine linear functions in the x_i over k^n such that the $z_j - \beta_j$ are linearly independent.
- We can use these linear functions to create quadratic "perturbation" in HFE (including MI) systems.

IP of MI

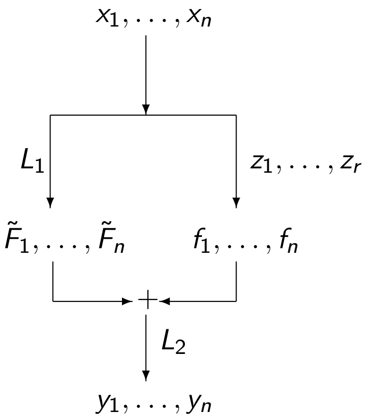


Figure: Structure of Perturbation of the Matsumoto-Imai System.

Decryption

- We need to a search of size of q^r, therefore slower.

Decryption

- We need to a search of size of q^r, therefore slower.
- We need to use Plus Method, **Adding random polynomial**, to help it to resist differential attacks.

Decryption

- We need to a search of size of q^r, therefore slower.
- We need to use Plus Method, **Adding random polynomial**, to help it to resist differential attacks.
- Despite the cost of the search, it is still efficient.

Efficient schemes

- PMI+

Efficient schemes

- PMI+
- IPHFE+ (odd characteristics)

Efficient schemes

- PMI+
- IPHFE+ (odd characteristics)
- IPMHFE+ (odd characteristics)

Other works

- Quartz – HFEV- – very short signature

Other works

- Quartz – HFEV- – very short signature
- MHFEv- short but much more efficinet schemes

Other works

- Quartz – HFEV- – very short signature
- MHFEv- short but much more efficinet schemes
- MFE, TTM

Outline

Direct attacks

- New polynomial solving algorithms, MXL, MGB, ZZ.

Direct attacks

- New polynomial solving algorithms, MXL, MGB, ZZ.
- Complexity analysis – recent work of Dubois, Gamma

Direct attacks

- New polynomial solving algorithms, MXL, MGB, ZZ.
- Complexity analysis – recent work of Dubois, Gamma
- SAT solver is not really a threat but needs more understanding

Direct attacks

- New polynomial solving algorithms, MXL, MGB, ZZ.
- Complexity analysis – recent work of Dubois, Gamma
- SAT solver is not really a threat but needs more understanding
- The connection with algebraic cryptanalysis of symmetric ciphers

Direct attacks

- New polynomial solving algorithms, MXL, MGB, ZZ.
- Complexity analysis – recent work of Dubois, Gamma
- SAT solver is not really a threat but needs more understanding
- The connection with algebraic cryptanalysis of symmetric ciphers
- Quantum computer attacks?

Provable security

- A very difficult question

Provable security

- A very difficult question
- Some new results are coming out.

New constructions

- New algebraic structure to explore
  Heindl, Gao – Diophantine Equations

New constructions

- New algebraic structure to explore
  Heindl, Gao – Diophantine Equations
- Other structures

The end

- Thank you very much!
- Questions?