Multivariate Public Key Cryptography and its Cryptanalysis

Jintai Ding

University of Cincinnati

Jintai.Ding@gmail.com

Quantum Cryptanalysis, Simons Institute, 02.2020

Overview

The Threat of Quantum Computers

- Quantum computer: using quantum mechanics principles to perform computations.
- Peter Shor's Algorithm to defeat RSA and ECC.
- Post-quantum cryptography, new cryptosystems that can resist quantum attacks.

NIST Call for Post-Quantum Cryptography Standardization

- NIST call for proposals of new, post-quantum cryptosystems (Dec 2016)
- Three criteria: Security, Cost, Algorithm and Implementation Characteristics
- Nine signature schemes left in Round 2

  Among them, 4 of them are multivariate signatures.

  Short signatures (Rainbow: 48 bytes), fastest signing and verifying, relatively large public key size (tens of Kbs) (except MQDSS).

Signature Schemes

Mathematical scheme for verifying the authenticity of digital messages or documents.

- Key generation: private key, public key.
- Signing: given a message and a private key, produces a signature.
- Verifying: given the message, public key and signature, either accepts or rejects the message's claim to authenticity.

Multivariate Signature schemes

- **Public key**: $\mathcal{P}(x_1, \cdots, x_n) = (p_1(x_1, \cdots, x_n), \cdots, p_m(x_1, \cdots, x_n))$. Here p_i are multivariate polynomials over a finite field.
- **Private key** A way to compute $\mathcal{P}^{-1}$.
- **Signing a hash of a document:**
  $(x_1, \cdots, x_n) \in \mathcal{P}^{-1}(y_1, \cdots, y_m)$.
- **Verifying:**
  $(y_1, \cdots, y_m) \stackrel{?}{=} \mathcal{P}(x_1, \cdots, x_n)$

Theoretical Foundation

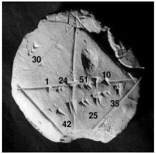
- Direct attack is to solve the set of equations:

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

- *- 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.*

A quick historic overview

- Single variable quadratic equation – Babylonian around 1800 to 1600 BC



- Cubic and quartic equation – around 1500

Tartaglia  Cardano 

- Multivariate system– 1964-1965
  Buchberger : Gröobner Basis
  Hironaka: Standard basis

The hardness of the problem

- Single variable case – Galois's work.

  Newton method – continuous system
  Berlekamp's algorithm – finite field and low degree
- Multivariate case: NP-complete, the generic systems.
  Numerical solvers – continuous systems
  Finite field case

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
- 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

Oil Vinegar Signature Scheme

- Introduced by J. Patarin, 1997
- Inspired by linearization attack to Matsumoto-Imai cryptosystem
- $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$.
  $\mathcal{F}$: nonlinear, easy to compute $\mathcal{F}^{-1}$.
  $\mathcal{T}$: invertible linear, to hide the structure of $\mathcal{F}$.

Oil Vinegar Signature Scheme

- $\mathcal{F} = (f_1(x_1, \cdots, x_0, x'_1, \cdots, x'_v), \cdots, f_o(x_1, \cdots, x_0, x'_1, \cdots, x'_v))$.
- $f_k = \sum a_{i,j,k} x_i x'_j + \sum b_{i,j,k} x'_i x'_j + \sum c_{i,k} x_i + \sum d_{i,k} x'_i + e_k$
- Oil variables: $x_1, \cdots, x_o$

  Vinegar variables: $x'_1, \cdots, x'_v$.
- **Public Key:** $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$.
  Private Key: $\mathcal{T}$.

Oil Vinegar Signature Scheme

- $\mathcal{P}^{-1} = \mathcal{T}^{-1} \circ \mathcal{F}^{-1}$
- Fix values for vinegar variables $x'_1, \cdots, x'_v$.
- $f_k = \sum a_{i,j,k} x_i x'_j + \sum b_{i,j,k} x'_i x'_j + \sum c_{i,k} x_i + \sum d_{i,k} x'_i + e_k$
- $\mathcal{F}$: Linear system in oil variables $x_1, \cdots, x_o$.

Example I

Parameters: $o = v = 2$, $n = 6$, Field is $\mathbb{F}_7$.
Here are the central map $\mathcal{F}$ and the change of basis $\mathcal{T}$ in matrix form:

$$\mathcal{F}(\mathbf{x}) = \begin{cases} f_1(\mathbf{x}) = x_1x_3 + 4x_2x_3 + 3x_2x_4 + 3x_2 + 5x_3x_4 + 6x_3 + 3x_4 + 1, \\ f_2(\mathbf{x}) = 5x_1x_3 + 3x_1x_4 + 6x_2x_3 + 3x_2x_4 + 6x_2 + 2x_3^2 + x_3x_4 \\ \qquad\qquad +x_3 + x_4^2 + x_4 + 3 \end{cases}$$

$$\mathcal{T} = \begin{bmatrix} 5 & 4 & 6 & 2 \\ 1 & 0 & 6 & 2 \\ 4 & 6 & 2 & 0 \\ 0 & 5 & 4 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Example II

And here is $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$

$$P(\mathbf{x}) = \begin{cases} \tilde{f}_1(\mathbf{x}) = x_1^2 + 3x_1x_2 + 6x_1x_3 + 5x_1x_4 + 6x_1 + 6x_2^2 + 6x_2x_4 + 2x_2 \\ \qquad +4x_3^2 + 2x_3x_4 + 6x_4 + 1, \\ \tilde{f}_2(\mathbf{x}) = 2x_1^2 + 3x_1x_2 + 5x_1x_3 + 4x_1x_4 + 3x_1 + 6x_2^2 + 2x_2x_3 \\ \qquad +3x_2x_4 + 4x_2 + x_3x_4 + 5x_4 + 3 \end{cases}$$

Note that this appears to be a random quadratic system, but it is not!

Security Analysis and Efficiency

- $v = o$
  Defeated by Kipnis and Shamir using invariant subspace (1998).
- $v >> o$
  Finding a solution is generally easy
- $v = 2o, 3o$
  Direct attack does not work – the complexity is the same as if solving a random system!
- Reconcilation attack – finding keys is converted into a polynomial solving problem
- Less efficient
  Signature is at least twice the size of the document

Modifications

- Rainbow, J. Ding, D. Schmidt (2004)
  Multilayer version of UOV.
- **Public Key:** $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$.
  Private Key: $\mathcal{T}, \mathcal{S}, \mathcal{F}$.
  Reduces number of variables in the public key
  smaller key sizes
  smaller signatures
- A new MinRank attack
  a problem to find linear combinations of a set of matrices to achieve the minimum rank.
- Rainbow is a NIST round 2 candidate.

LUOV

- A modification of the original unbalanced oilvinegar scheme designed in 2017.
- Coefficients of the public key are from $\mathbb{F}_2$
- Shorten the size of public key.
- A NIST round 2 candidate but we broke the original submission to NIST with Subfield Differential attack.

Cryptanalysis Tools

- Direct attack
- Reconciliation attack
- MinRank Attack
- Subfield Differential attack

  All of them are reduced to solving polynomial equations.

How to solve multivariate systems?

We would like to solve:

$$F_1 = y_1, ..., F_m = y_m$$

- We in general like to look at

  $$F_1 - y_1 =, ..., F_m - y_m = 0$$

  Over the function ring: $k[x_1, ..., x_n]/ < x_1^q - x_1, ..., x_n^q - x_n >$, we need to find: $x_i - a_i = 0$.
- The first general method is Groebner basis method in 1960s, but the same idea was discussed by Hironaka earlier.
  S polynomial from leading terms of the polynomials
- Later the idea of using linear algebra
  Lazard etc
  Dense Linear Algebra

How to solve multivariate systems?

A different from the point of ideal and linear algebra

- The view of algebraic geometry for the case with only one solution: Ideal $< F_1 - y_1, ..., F_m - y_m >= \{h | h = \sum g_i(F_i - y_i)\} =$ Ideal $< x_1 - a_1, .., x_n - a_n >$.
  Over the function ring: $k[x_1, ..., x_n]/ < x_1^q - x_1, ..., x_n^q - x_n >$, we need to find:

$$x_i - a_i = \sum g_i(F_i - y_i).$$

- The significance of the field equations: $x_i^q = x_i$.
  Solutions over the finite field or its algebraic closure?

How to solve multivariate systems?

A different from the point of ideal and linear algebra

- The computation strategy:
  look for the desired polynomials through elements in the ideal via linear algebra
  Matrix with:
  a row – a polynomial, a column – a monomial
  Gaussian elimination on rows and essentially solve the equation: $MX = b'$, where
  $X = (x_1, x_2, ...x_n, x_1x_2, ...,$ (list of all monomials)), M, the polynomial coefficient matrix, b', the constant terms of the plynomials.
- The complexity – the size of the largest matrix

How to solve multivariate systems?

The simplest and the most direct way – the XL algorithm:

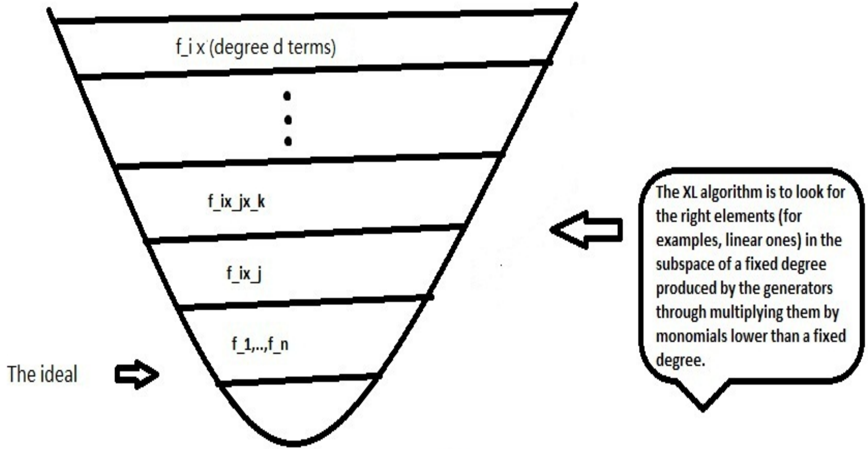


Fig. 1

The degree fall

- Rethinking the formula:

$$x_i - a_i = \sum g_i(\bar{F}_i - y_i).$$

  The degree of the L.F.S. must go down!

- The implication of degree fall — certain degeneration of the system:
  mutant
- The implication of mutant:
  Mutant XL and its variants.

Mutants

The degree must go down: **mutants** and mutant XL

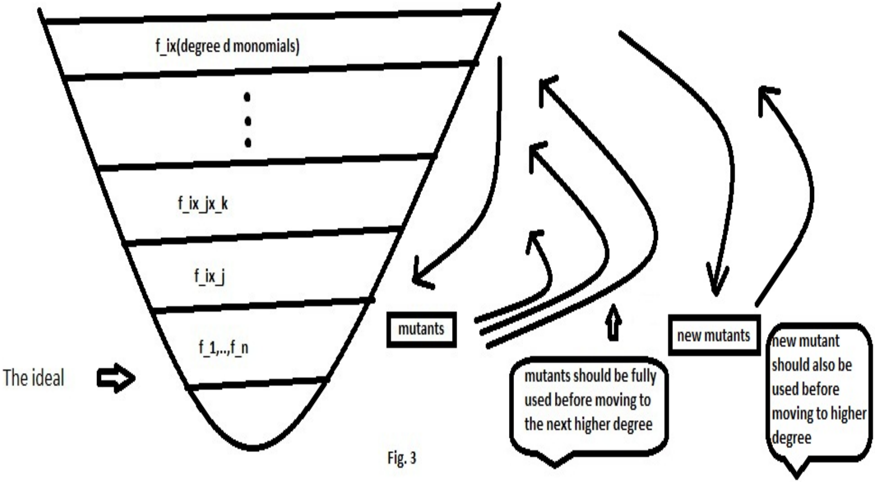


Fig. 3

The key concepts

- The solving degree: the degree at which the maximum matrix size is achieved.
- The mutant degree: the lowest degree at which a mutant appears
- The degeneration degree: the lowest degree where there is non-trivial degeneration of the top level of the polynomial system.
- Are they really different?
  $SD \geq MD \geq DD$
  The convention: *for non-degenerate systems, they are essentially the same.* A work of Ding and Schmidt: $SD - DD \leq 2$.

Degeneration Degree?

- For a regular system:
  Degree of Regularity
- The name change:
  Degeneration Degree
- A hard problem:
  bounds on the DD – complexity analysis
  Many works done in the area to lay a solid foundation for the security analysis of MPKCs. Degree of regularity of HFE systems by Ding, Hodges, Kleinjung, Yang etc
  Theory and experiments match very well !!
  Optimal choice of parameters.

Sparsity

- For XL, the linear system is sparse!
- One can Wiedemann or block Wiedemann method by Yang etc

Quantum attacks – Grover's Algorithm

- Square root speed up
- Relative large key size
  Large number of quantum bits.

HHL and Gao groups's work

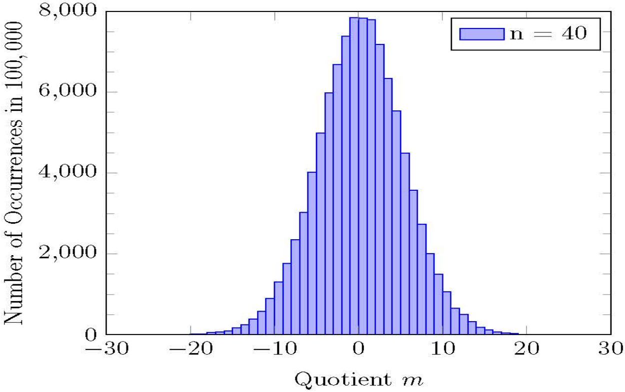
- HHL
  Harrow, Hassidim, and Lloyd 2009
  Solving a sparse linear system
  $AX = b$
  over real numbers
- Assumptions:
  1) Efficient way to compute or access none-zero terms in A and b
  2) The matrix A must be Hermitian
  3) The complexity depends on the condition number κ which is the ratio of the max and the min of the eigenvalues of A.
  The best complexity: $\mathcal{O}(d\kappa\text{poly}(\log(d\kappa/\varepsilon)))$, where d is the sparseness of A, and ε is the precision.

The idea of Gao etc

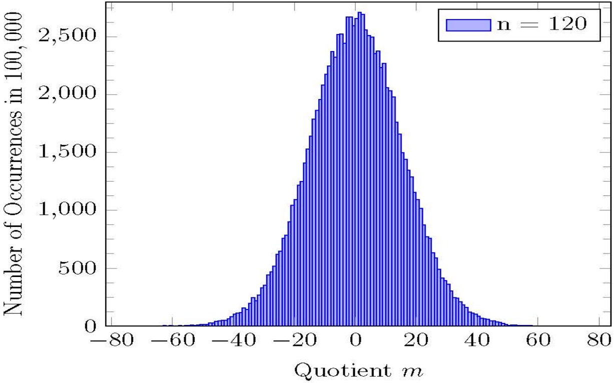
- Add modular part back
  $F(X) = 0 \bmod 2$
  becomes $F(X) = 2z$
  This idea was already developed by Ding, Schmidt etc in 2012. (https://eprint.iacr.org/2012/094.pdf)
- Then add
  $\prod_{a<i<b}(z - i) = 0$
  Very important to ensure the solution is unique otherwise we will have solution from extension field!
- Symmetrization of the Macauley matrix
  $MX = b$
  $M^T MX = Mb.$
- Then apply HHL
  The complexity is polynomial in terms of log of matrix size and conditional number.
  If the condition number is polynomial in n, we have polynomial algorithm.

The complexity

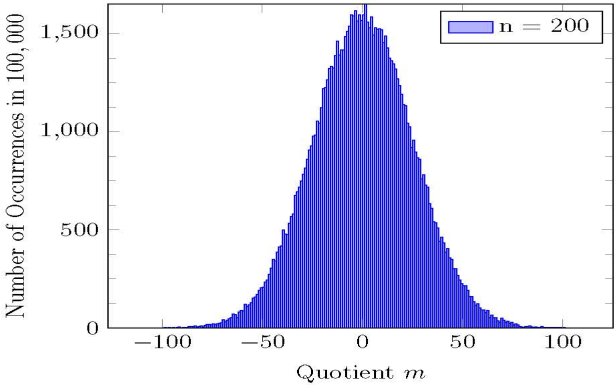
- The degree of regularity is high for MPKC is hight and the range of z is the same.
  For a random system, we expect the degree of be $n/8$
  The range of z is in general $[-8/n, 8/n]$.
- As long as the conditional number is small, we have a fast quantum algorithm.


Number of Occurrences in 100,000
8,000
6,000
4,000
2,000
0
−30
−20
−10
0
10
20
30
Quotient m
n = 40

$s = 5.19630$


n = 120
Number of Occurrences in 100,000
2,500
2,000
1,500
1,000
500
0
−80
−60
−40
−20
0
20
40
60
80
Quotient m

$s = 15.18424$


n = 200
Number of Occurrences in 100,000
1,500
1,000
500
0
−100
−50
0
50
100
Quotient m

$s = 25.12024$

A new way to estimate the conditional number – joint work with Vlad Gheorghiu

- we divide the system (M) into two parts
  1) the original equations: small coefficient:
  0, 1, -1
  2) the modular part: large and small coefficients:
  $\prod_{-n/8 \leq i \leq n/8} (z - i) = 0$

  has 1, and $((n/8)!)^2 \geq 2^n$

A new way to estimate the conditional number – joint work with Vlad Gheorghiu

- $M^T M$ is (semi)positive definite with large and small entries in the diagonal.
- Min(Eigenvalue of M^T) $\leq$ diagonal entries $\leq n^2/2$
  Max(Eigenvalue of M^T) $\geq$ diagonal entries $\geq 2^n$
  The conditional number is exponential in general.

The complexity

- Can we rescale the coefficients to reduce the large entries in $M^T M$ Our analysis shows that it is not the case because of the large spread of the the coefficients and re-scaling could cause very serious problems because the system becomes unstable.
- We can apply the same analysis to other attacks by Gao etc.

Thank you!

Questions to Jintai.Ding@gmail.com

Supported by Taft Fund, NIST and NSF