Current State of Multivariate Cryptography

Jintai Ding | University of Cincinnati
Albrecht Petzoldt | Kyushu University

A review of the current state of multivariate public-key cryptosystems reveals the most promising and secure multivariate schemes in digital signatures and public-key encryption.

The invention of public-key cryptosystems in the late 1970s was a fundamental breakthrough in modern cryptography. Since then, public-key cryptosystems have become an increasingly integral part of our communication networks. Today, the Internet and other communication systems rely principally on the Diffie-Hellman key exchange, RSA encryption, and digital signatures using digital signature algorithm (DSA), elliptic curve DSA, or related algorithms. The security of these cryptosystems depends on the difficulty of certain number theoretic problems, such as integer factorization or the elliptic curve discrete logarithm. However, in 1994, Peter Shor showed that quantum computers can solve each of these problems in polynomial time.[1] Therefore, as soon as large quantum computers become a reality, all cryptosystems based on such assumptions will be insecure.

In the past several years, a large international community has emerged to address this issue. The hope is that our public-key infrastructure might remain intact by utilizing new quantum-resistant primitives. In the academic world, this new research field is called postquantum cryptography.

In this article, we focus on the best existing multivariate public-key cryptosystem (MPKC) candidates for signatures and encryption. (Due to space constraints, we don't give a complete overview of the history of multivariate cryptography and the variety of existing schemes.) Furthermore, we give an overview of the security of multivariate schemes and discuss their advantages and disadvantages.

Background

In August 2015, the NSA published a webpage announcing preliminary plans for transitioning to quantum-resistant algorithms (www.iad.gov/iad/programs/iad-initiatives/cnsa-suite.cfm). In December 2016, NIST announced a call for proposals for quantum-resistant algorithms (www.nist.gov/pqcrypto). In addition, government organizations like the European Commission and the Japanese Society for the Promotion of Science finance research programs such as PQCRYPTO, SAFECRYPTO, and CryptoMathCrest to enhance postquantum cryptography research. Due to these initiatives, the effort to develop quantum-resistant technologies—particularly postquantum cryptosystems—is becoming a central research area in information security. Currently, four main families of public-key cryptosystems have the potential to resist quantum computer attacks, namely:

1540-7993/17/$33.00 © 2017 IEEE

- hash-based signature schemes of the Diffie-Lamport-Merkle type,
- lattice-based public-key cryptosystems (for instance, NTRU [N-th degree truncated polynomial ring] and BLISS [Bimodal Lattice Signature Scheme]),
- code-based public-key cryptosystems (in particular, the McEliece encryption scheme), and
- MPKCs.

For efficiency reasons, the MPKC public key is usually a system of quadratic polynomials in several variables over a small finite field K with q elements.

$$\begin{aligned} p^{(1)}(x_1 ..., x_n) &= \sum_{i=1}^{n} p_{ij}^{(1)} x_i \cdot x_j + \sum_{i=1}^{n} p_i^{(1)} x_i + p_0^{(1)}, \\ p^{(2)}(x_1 ..., x_n) &= \sum_{i=1}^{n} p_{ij}^{(2)} x_i \cdot x_j + \sum_{i=1}^{n} p_i^{(2)} x_i + p_0^{(2)}, \\ &..., \\ p^{(m)}(x_1 ..., x_n) &= \sum_{i=1}^{n} p_{ij}^{(m)} x_i \cdot x_j + \sum_{i=1}^{n} p_i^{(m)} x_i + p_0^{(m)}. \end{aligned} \tag{1}$$

The security of multivariate schemes is based on the multivariate quadratic polynomial (MQ) problem: given m quadratic polynomials $p^{(1)}(x)$, …, $p^{(m)}(x)$ in the n variables x_1, …, x_n—as shown in Equation 1—find a vector $\tilde{x} = (\tilde{x}_1 ..., \tilde{x}_n)$ such that $p^{(1)}(\tilde{x}) = ... = p^{(m)}(\tilde{x}) = 0$.

The MQ problem was proven to be nondeterministic polynomial time (NP)-hard over any field and is believed to be hard on average for both classical and quantum computers.[2] This is the security basis of MPKCs.

To build a public-key cryptosystem on the basis of the MQ problem, we start with an easily invertible quadratic map F: $K^n \to K^m$ (central map). To hide the central map's structure in the public key, we combine F with two invertible affine maps S: $K^m \to K^m$ and T: $K^n \to K^n$. The scheme's public key is the composed map $P = S \circ F \circ T$: $K^n \to K^m$; the private key consists of the three maps S, F, and T.

Figure 1 shows the standard process of encryption/decryption or signature generation/verification.

Construction of MPKCs

An encryption scheme requires that the public-key map is injective to ensure the decryption process outputs a unique plaintext. This is why $m \geq n$.

To encrypt a message $z \in K^n$, we evaluate the public key to get the ciphertext $w = P(z) \in K^m$. To decrypt a ciphertext $w \in K^n$, we compute recursively $x = S^{-1}(w) \in K^m$, $y = F^{-1}(x) \in K^n$, and $z = T^{-1}(y)$. The plaintext corresponding to the ciphertext w is given by $z \in K^n$.

A signature scheme requires that the public-key map is surjective to ensure that one can sign any document. This is why $m \leq n$.

Decryption/signature generation

$$\begin{tikzcd} w \in K^m \arrow[r, "S^{-1}"] & x \in K^m \arrow[r, "F^{-1}"] & y \in K^n \arrow[r, "T^{-1}"] & z \in K^n \arrow[lll, bend left, "P"] \end{tikzcd}$$

Encryption/signature verification

Figure 1. Standard workflow of multivariate public-key cryptosystems.

To generate a signature for a message d, we use a hash function H: $\{0,1\}^* \to K^m$ to compute the hash value $w = H(d) \in K^m$. After that, we compute recursively $x = S^{-1}(w) \in K^m$, $y = F^{-1}(x) \in K^n$ and $z = T^{-1}(y)$. The signature of the message d is given by $z \in K^n$. Here, $F^{-1}(x)$ means finding one (of possibly many) preimage of x under the central map F.

To check the authenticity of a signature $z \in K^n$, we compute the hash value $w = H(d) \in K^m$ of the message d and evaluate the public map to obtain $w' = P(z) \in K^m$. If $w' = w$ holds, the signature is accepted; otherwise, it's rejected.

The problem of recovering the private key from the public key is equivalent to finding the composition of P in $P = S \circ F \circ T$. This problem is known as extended isomorphism of polynomials. In spirit, this is similar to the case of RSA, in which one must find the factorization of the RSA modulus into two prime numbers.

In the mid-1980s, Whitfield Diffie, Harriet Fell, Shigeo Tsuji, Adi Shamir, and others began researching MPKC but weren't very successful at the time. The real breakthrough was the C* cryptosystem, proposed by Tsutomu Matsumoto and Hideki Imai in 1988,[3] which used the new idea of a BigField scheme wherein the central map F is defined over an extension field E of K.

This work stimulated fast development in MPKC, which led to the development of the Oil and Vinegar (OV) and HFEv– (Hidden Field Equations with Vinegar and Minus) families of signature schemes, which have sustained 15 to 20 years of attacks and therefore are believed to offer a high level of security. Although many practical multivariate signature schemes exist, the development of secure and efficient multivariate encryption schemes appeared to be a much harder task. Until recently, there were only a few secure multivariate encryption schemes, such as PMI+ (Perturbed Matsumoto-Imai with Plus) and IPHFE+ (Internally Perturbed HFE with Plus), which are much less efficient than the signature schemes. However, in 2013, a new and efficient scheme based on simple matrix multiplications appeared: the SimpleMatrix encryption scheme. The scheme is very new, but it's very simple and easy to understand and therefore has great potential.

Signature Schemes

Here, we give an overview of the most promising multivariate signature schemes. We describe two schemes—unbalanced OV (UOV) and Rainbow—and provide a short overview of other schemes.

Oil and Vinegar Signature Schemes

Let K be a finite field and o and v be integers. We define $n = o + v$, $V = \{1, \dots, v\}$ and $O = \{v + 1, \dots, n\}$. We call $x_1, \dots, x_v$ *vinegar variables* and $x_v + 1, \dots, x_n$ *oil variables*. For $o = v$, the scheme is called balanced OV; for $v > o$, we speak of the UOV signature scheme.

Key generation. The central map $F: K^n \to K^o$ of the (U)OV signature scheme consists of o quadratic polynomials $f^{(1)}, \dots, f^{(o)}$ of the form

$$f^{(k)} = \sum_{i\in V}\sum_{j\in V} a_{ij}^{(k)} x_i \cdot x_j + \sum_{i\in V}\sum_{j\in O} b_{ij}^{(k)} x_i \cdot x_j + \sum_{i\in V\cup O} c_i^{(k)} x_i + d^{(k)}$$

with randomly chosen coefficients $a_{ij}^{(k)}, b_{ij}^{(k)}, c_i^{(k)}$, and $d^{(k)} \in K$.

Note that the polynomials $f^{(1)}, \dots, f^{(o)}$ contain no quadratic terms $x_i \cdot x_j$ with both $i,j \in O$. We will use this later to invert F.

To hide the structure of F in the public key, we combine F with one invertible affine map $T: K^n \to K^n$. Therefore, the scheme's public key has the form $P = F \circ T: K^m \to K^o$; the private key consists of the maps F and T.

Inversion of the central map. To find a preimage $x \in K^n$ of $y \in K^o$ under the central map F, we choose randomly the values of the vinegar variables $x_1, \dots, x_v$ and substitute them into the polynomials $f^{(1)}, \dots, f^{(o)}$. Due to the special structure of the central polynomials, we obtain by this strategy o linear polynomials $\tilde{f}^{(1)}, \dots, \tilde{f}^{(o)}$ in the o oil variables $x_{v+1}, \dots, x_n$. We can solve the resulting linear system

$$\tilde{f}^{(k)}(x_{v+1}, \dots, x_n) = y_k$$

by Gaussian elimination, where $k = 1, \dots, o$. If the system has no solution, we choose other values for the vinegar variables $x_1, \dots, x_v$ and try again.

Toy example. Let $K = \mathrm{GF}(7)$ (GF is Galois field) and $o = v = 2$. Let the central map $F = (f^{(1)}, f^{(2)})$ of our (balanced) OV instance be given by

$$\begin{aligned} f^{(1)}(x_1, \dots, x_4) = {} & 2x_1^2 + 3x_1 \cdot x_2 + 6x_1 \cdot x_3 + x_1 \cdot x_4 \\ & + 4x_2^2 + 5x_2 \cdot x_4 + 3x_1 + 2x_2 + 5x_3 \\ & + x_4 + 6, \end{aligned}$$

$$\begin{aligned} f^{(2)}(x_1, \dots, x_4) = {} & 3x_1^2 + 6x_1 \cdot x_2 + 5x_1 \cdot x_4 + 3x_2^2 \\ & + 5x_2 \cdot x_3 + x_2 \cdot x_4 + 2x_1 + 5x_2 \\ & + 4x_3 + 2x_4 + 1. \end{aligned}$$

To find a preimage $x = (x_1 \dots, x_4)$ of $y = (3,4)$ under the central map F, we choose random values for x_1 and x_2, for example, $(x_1, x_2) = (1,4)$, and substitute them into $f^{(1)}$ and $f^{(2)}$. By doing so, we obtain

$$\tilde{f}^{(1)}(x_1, \dots, x_4) = 4x_3 + x_4 + 4,$$

$$\tilde{f}^{(2)}(x_1, \dots, x_4) = 3x_3 + 4x_4.$$

By solving the linear system $\tilde{f}^{(1)} = y_1 = 3, \tilde{f}^{(2)} = y_2 = 4$, we obtain $(x_3, x_4) = (1,2)$. Therefore, the required preimage is $x = (1,4,1,2) \in K^4$.

Signature generation. To generate a signature $z \in K^n$ for a message d, we use a hash function $H: \{0,1\}^* \to K^o$ to compute $w = H(d) \in K^o$ and perform the following steps:

- compute a preimage $x \in K^n$ of w under the central map as we described earlier, and
- compute the signature $z \in K^n$ of the document d by $z = T^{-1}(x)$.

Signature verification. To check the authenticity of a signature $z \in K^n$, we use the hash function H to compute $w = H(d) \in K^o$ and compute $w = P(z)$. If $w' = w$ holds, the signature is accepted; otherwise, it's rejected.

Security. In the first version of the scheme (balanced OV[4]), equal values of o and v were chosen. However, this makes the scheme weak against an attack proposed by Aviad Kipnis and Adi Shamir in "Cryptanalysis of the Oil-Vinegar Signature Scheme."[5] To avoid this, Kipnis and his colleagues suggested using $v \geq 2o$ (UOV).[6] For suitable parameter sets, the UOV signature scheme has resisted cryptanalysis for 20 years and is therefore believed to offer high security.

Table 1 shows practical parameters for UOV. Roughly speaking, UOV parameters $(o,v) = (k,2k)$ lead to $2k$ bits of security over GF(16) and to approximately $2.6 \cdot (k - 2) + 12$ bits of security over GF(256).[7] The corresponding hash and signature sizes are $4k$ respectively $12k$ bits over GF(16) and $8k$ respectively $24k$ bits over GF(256).

Rainbow

The Rainbow signature scheme, as proposed by Jintai Ding and Dieter Schmidt in "Rainbow, a New Multivariate Polynomial Signature Scheme," can be seen as a multilayer version of UOV.[8] By their modifications,

Table 1. Parameters and key sizes of current multivariate signature schemes.[7,13]

Security level (bits)	Scheme parameters	Public-key size (Kbytes)	Private-key size (Kbytes)	Hash size (bits)	Signature size (bits)
80	Unbalanced Oil and Vinegar (UOV) (GF(2^8), 28, 56)	99.9	95.8	224	672
	Rainbow (GF(2^8), 17, 13, 13)	25.1	19.9	208	344
	Hidden Field Equations with Vinegar and Minus (HFEv–) (GF(7), 62, 8, 2, 2)	47.1	2.9	168	192
	Gui (GF(2), 95, 9, 5, 5)	60.1	3.0	—*	120
100	UOV (GF(2^8), 35, 70)	193.8	183.2	280	840
	Rainbow (GF(28), 26, 17, 16)	59.0	44.4	264	472
	HFEv– (GF(7), 78, 8, 3, 3)	93.5	4.5	210	243
120	Gui (GF(2), 127, 9, 4, 4)	139.1	5.2	—*	163
128	UOV (GF(2^8), 45, 90)	409.4	381.8	360	1,080
	Rainbow (GF(2^8), 36, 22, 21)	136.1	101.3	344	632
	HFEv– (GF(7), 100, 8, 4, 4)	65.2	2.8	264	296

** Gui can be instantiated with hash functions of arbitrary length.*[13]

Ding and Schmidt were able to reduce key and signature sizes as well as improve UOV performance. The scheme can be described as follows.

Let K be a finite field and $0 < v_1 < v_2 < \ldots < v_{u+1} = n$ be a sequence of integers. We set $V_i = \{1, \ldots, v_i\}$, $O_i = \{v_i + 1, \ldots, v_{i+1}\}$ and $o_i = v_{i+1} - v_i$ $(i = 1, \ldots, u)$.

Key generation. The Rainbow signature scheme's central map F consists of $m = n - v_1$ quadratic polynomials $f^{(v_1+1)}, \ldots, f^{(n)}$ of the form

$$f^{(k)} = \sum_{i\in V_l}\sum_{j\in V_l} a_{ij}^{(k)} x_i \cdot x_j + \sum_{i\in V_l}\sum_{j\in O_l} b_{ij}^{(k)} x_i \cdot x_j + \sum_{i\in V_l \cup O_l} c_i^{(k)} x_i + d^{(k)} \quad (2)$$
,

where $l \in \{1, \ldots, u\}$ is the only integer such that $k \in O_l$. Note that, in every polynomial $f^{(k)}$ with $k \in O_l$, there is no quadratic term $x_i \cdot x_j$ with both $i,j \in O_l$. Therefore, if we substitute the variables x_i $(i \in V_l)$ into the equations $f^{(k)}$ $(k \in O_l)$, we obtain a system of o_l linear equations in the o_l variables x_i $(i \in O_l)$. We will use this during the scheme's signature generation process.

To hide the structure of F in the public key, we compose it with two invertible affine or linear maps S: $K^m \rightarrow K^m$ and T: $K^n \rightarrow K^n$. Hence, Rainbow's public key has the form $P = S \circ F \circ T$: $K^n \rightarrow K^m$; its private key consists of the three maps S, F, and T.

Inversion of the central map. Because the Rainbow central map consists of several layers of UOV, it can be inverted by inverting the single UOV layers recursively (see Algorithm 1). The variables of the ith layer are hereby used as the vinegar variables of the $i + 1$th layer.

Algorithm 1: inversion of the Rainbow central map.

Input: Rainbow central map $F = (f^{(v_1+1)}, \ldots, f^{(n)})$ vector $y \in K^m$.

Output: vector $x \in K^n$ with $F(x) = y$.

```
Choose random values for the variables $x_1, \ldots, x_{v_1}$ and substitute these values into the polynomials $f^{(i)}$ $(i = v_1 + 1, \ldots, n)$.
for $l = 1$ to $u$ do
    Perform Gaussian elimination on the polynomials $f^{(i)}$ $(i \in O_l)$ to get the values of the variables $x_i$ $(i \in O_l)$.
    Substitute the values of $x_i$ $(i \in O_l)$ into the polynomials $f^{(i)}$ $(i = v_1 + 1, \ldots, n)$.
end for.
```

$$f^{(3)} = x_1^2 + 3x_1 \cdot x_2 + 5x_1 \cdot x_3 + 6x_1 \cdot x_4 + 2x_2^2 = 6x_2 \cdot x_3 + 4x - 2 \cdot x_4 + 2x_2 + 6x_3 + 2x_4 + 5,$$
$$f^{(4)} = 2x_1^2 + x_1 \cdot x_2 + x_1 \cdot x_3 + 3x_1 \cdot x_4 + 4x_1 + x_2^2 + x_2 \cdot x_3 + 4x_2 \cdot x_4 + 6x_2 + x_4,$$
$$f^{(5)} = 2x_1^2 + 3x_1 \cdot x_2 + 3x_1 \cdot x_3 + 3x_1 \cdot x_4 + x_1x_5 + 3x_1x_6 + 6x_1 + 4x_2^2 + x_2x_3 + 4x_2x_4 + x_2x_5 + 3x_2x_6 + 3x_2 + 3x_3x_4 + x_3x_5 + 2x_3x_6 + 2x_3 + 3x_4x_5 \cdot x_5 + 6x_6,$$
$$f^{(6)} = 2x_1^2 + 5x_1x_2 + x_1x_3 + 5x_1x_4 + 5x_1x_6 + 6x_1 + 5x_2^2 + 3x_2x_3 + 5x_2x_5 + 4x_2x_6x_2 + 3x_3^2 + 5x_3x_4 + 4x_3x_5 + 2x_3x_6 + 4x_3 + x_4^2 + 6x_4x_5 + 4x_4 + 4x_5 + x_6 + 2.$$

Figure 2. Central map of our Rainbow instance.

Toy example. Let K = GF(7). We consider a Rainbow instance with two layers, $(v_1, o_1, o_2) = (2,2,2)$ and central map $F = (f^{(3)}, \ldots, f^{(6)})$, as seen in Figure 2.

Let's assume that we want to find a preimage $x \in K^6$ of $y = (6,2,0,5) \in K^4$ under the map F. To do this, we choose random values for the vinegar variables x_1 and x_2, for example, $(x_1, x_2) = (0,1)$ and substitute them into the polynomials $f^{(3)}, \ldots, f^{(6)}$. By doing so, we get

$$\tilde{f}^{(3)} = 5x_3 + 6x_4 + 2,$$

$$\tilde{f}^{(4)} = x_3 + 5x_4,$$

$$\tilde{f}^{(5)} = 3x_3x_4 + x_3x_5 + 2x_3x_6 + 3x_4x_5 + 4x_4 + 2x_5 + 2x_6,$$

$$\tilde{f}^{(6)} = 3x_3^2 + 5x_3x_4 + 4x_3x_5 + 2x_3x_6 + x_4^2 + 6x_4x_5 + 3x_4x_6 + 4x_4 + 2x_5 + 5x_6 + 1.$$

By setting $\tilde{f}^{(3)} = y_1$ and $\tilde{f}^{(4)} = y_2 = 2$, we obtain $(x_3, x_4) = (3,4)$. Substituting these values into $\tilde{f}^{(5)}$ and $\tilde{f}^{(6)}$ yields

$$\overset{\approx}{f}^{(5)} = 3x_5 + x_6 + 5,$$

$$\overset{\approx}{f}^{(6)} = 3x_5 + 2x_6 + 1.$$

By setting $\overset{\approx}{f}^{(5)} = y_3 = 0$ and $\overset{\approx}{f}^{(6)} = y_4 = 5$, we obtain $(x_5, x_6) = (0,2)$. Altogether, we get the preimage $x = (0,1,3,4,0,2) \in K^6$.

Signature generation. To generate a signature for a message d, we use a hash function H: $\{0,1\}^* \to K^m$ to compute the hash value $w = H(d) \in K^m$ and perform the following three steps:

- compute $x = S^{-1}(w) \in K^m$,
- compute a preimage $y \in K^n$ of x under the central map F as shown in Algorithm 1, and
- compute the signature $z \in K^n$ by $z = T^{-1}(y)$.

Signature verification. To check if $z \in K^n$ is a valid signature for the message d, we compute $w = H(d) \in K^m$ and $w' = P(z)$. If $w' = w$ holds, the signature is accepted; otherwise, it's rejected.

Security. As the Rainbow signature scheme can be seen as an extension of the widely studied UOV signature scheme, major parts of the security analysis of UOV relate to Rainbow, too. However, the additional structure of the Rainbow central map enables several new attack strategies, such as the MinRank attack and the Rainbow-Band-Separation attack.[9] This attack aims to find linear maps that transform the public polynomials into quadratic maps of the form of Equation 2, which then can be used to forge signatures. The linear maps S and T can be recovered by solving systems of multivariate nonlinear equations.

Due to these additional attack possibilities, Rainbow parameter selection is challenging. Table 1 shows practical parameter sets for the Rainbow signature scheme. Roughly speaking, to achieve a security level of k bits for a Rainbow scheme with two layers, we need to choose the parameters (v_1, o_1, o_2) to be about $(k/4, k/4, k/4)$ for GF(16) and $(4s/3, s, s)$ with $s = (k-12)/5$ over GF(256).[7] The corresponding hash and signature sizes are $2k$ bits respectively $3k$ bits over GF(16) and $16s$ bits respectively $27s$ bits over GF(256).

Efficiency and implementation. Both UOV and Rainbow require only simple operations such as matrix vector multiplication and matrix inversion over a small finite field. Therefore, these schemes are very easy to implement and can be used on embedded devices. Compared to UOV, Rainbow reduces the number of variables in the system, which leads to smaller key sizes, shorter signatures, and better performance. This makes Rainbow one of the fastest signature schemes (see bench.cr.yp.to).

Other Schemes

Other important examples of multivariate signature schemes include HFEv, Gui, and MQDSS (Multivariate Quadratic Digital Signature Scheme).

HFE variants. The HFE scheme is a multivariate scheme of the BigField family, which was originally proposed as a candidate for a multivariate encryption scheme.[10] After the basic scheme was broken due to direct and rank attacks, several secure HFE variants for signature schemes, including HFE–, HFEv, and HFEv–, have been proposed. These schemes aim to remove some of the equations from the HFE public key (HFE–) and to parametrize the HFE central map by introducing additional vinegar variables (HFEv). HFEv– combines these two ideas and provides the best combination of security and efficiency of these HFE variants.[11] HFEv– is one of the most studied multivariate schemes and is therefore believed to offer high security.[12,13]

Gui. The Gui signature scheme, proposed by Albrecht Petzoldt and his colleagues in "Design Principles for HFEv-Based Signature Schemes," is an extension of the HFEv– signature scheme.[13] Due to a specially designed signature generation algorithm, it's possible to create secure 120-bit-long signatures (80-bit security), which are the shortest signatures of all currently existing signature schemes (both classical and postquantum). Another result of this special signature generation process is that, in contrast to other multivariate schemes, the output length of the hash function in use isn't fixed. This makes it easier to switch to other hash functions.

MQDSS. The MQDSS signature scheme is one of the few provable secure multivariate schemes;[14] that is, the scheme's security depends only on the hardness of the MQ problem. However, regarding signature sizes and performance, it can't compete with other multivariate schemes such as Rainbow and HFEv–.

Others. There are several other multivariate signature schemes, such as PFLASH[15] and TTS.[16] Due to space constraints, we can't cover these schemes here and refer readers to the original papers.

Encryption Schemes

Here, we give an overview of multivariate schemes for encryption. The recently proposed SimpleMatrix scheme is currently the most promising candidate for a multivariate encryption scheme.

SimpleMatrix

The SimpleMatrix (or ABC) encryption scheme, as proposed by Chengdong Tao and his colleagues in "Simple Matrix Scheme for Encryption," can be described as follows.[17]

Let K be a finite field with q elements and s be an integer. We set $n = s^2$ and $m = 2n$. To generate the key, we define three matrices A, B, and C in the form

$$A = \begin{pmatrix} x_1 & x_2 & \cdots & x_s \\ x_{s+1} & x_{s+2} & \cdots & x_{2s} \\ \vdots & \vdots & & \vdots \\ x_{n-s+1} & x_{n-s+2} & \cdots & x_n \end{pmatrix}$$

$$B = \begin{pmatrix} b_1 & b_2 & \cdots & b_s \\ b_{s+1} & b_{s+2} & \cdots & b_{2s} \\ \vdots & \vdots & & \vdots \\ b_{n-s+1} & b_{n-s+2} & \cdots & b_n \end{pmatrix}$$

$$C = \begin{pmatrix} c_1 & c_2 & \cdots & c_s \\ c_{s+1} & c_{s+2} & \cdots & c_{2s} \\ \vdots & \vdots & & \vdots \\ c_{n-s+1} & c_{n-s+2} & \cdots & c_n \end{pmatrix}.$$

Here, $x_1, \ldots, x_n$ are the linear monomials of the multivariate polynomial ring $F[x_1, \ldots, x_n]$, whereas $b_1, \ldots, b_n$ and $c_1, \ldots, c_n$ are randomly chosen linear combinations of $x_1, \ldots, x_n$. We compute two $s \times s$ matrices E_1 and E_2 containing quadratic polynomials by $E_1 = A \cdot B$ and $E_2 = A \cdot C$. The scheme's central map F consists of the m components of E_1 and E_2. The scheme's public key is the composed map $P = S \circ F \circ T$: $K^n \to K^m$ with two randomly chosen invertible linear maps S: $K^m \to K^m$ and T: $K^n \to K^n$; the private key consists of the matrices B and C and the linear maps S and T.

To encrypt a message $z \in K^n$, we simply compute $w = P(z) \in K^m$. To decrypt a ciphertext $w \in K^m$, we perform the following three steps. First, compute $x = S^{-1}(w)$. The elements of the vector $x \in K^m$ are written into matrices E_1 and E_2, shown as follows:

$$\tilde{E}_1 = \begin{pmatrix} x_1 & x_2 & \cdots & x_s \\ x_{s+1} & x_{s+2} & \cdots & x_{2s} \\ \vdots & \vdots & & \vdots \\ x_{n-s+1} & x_{n-x+2} & \cdots & x_n \end{pmatrix}$$

$$\tilde{E}_2 = \begin{pmatrix} x_{n+1} & x_{n+2} & \cdots & x_{n+s} \\ x_{n+s+1} & x_{n+s+2} & \cdots & x_{n+2s} \\ \vdots & \vdots & & \vdots \\ x_{m-s+1} & x_{m-x+2} & \cdots & x_m \end{pmatrix}..$$

Next, find a vector $y = (y_1, \ldots, y_n) \in K^n$ such that $F(y) = x$. To do this, we assume that the matrix $\bar{A} = A(y)$ is invertible. We consider the relations $\bar{A}^{-1} \cdot \bar{E}_1 - B = 0$ and $\bar{A}^{-1} \cdot \bar{E}_2 - C = 0$ and interpret the elements of $\bar{A}^{-1}$ as new variables $r_1, \ldots, r_n$ and therefore get m linear equations in the m variables $r_1, \ldots, r_n$, $y_1, \ldots, y_n$. Hence, the values of $y_1, \ldots, y_n$ can be recovered by Gaussian elimination.

Finally, we compute the plaintext $z \in F^n$ by $z = T^{-1}(y_1, \ldots, y_n)$.

The linear system in the second step might have multiple solutions $y^{(1)}, \ldots, y^{(l)}$. In this case, we must perform the third step of the decryption process for each of these solutions to get a set of possible plaintexts $z^{(1)}, \ldots, z^{(l)}$. By encrypting these plaintexts, we can test which of them corresponds to the given ciphertext w.

If, in the second step of the decryption process, the matrix $\bar{A} = A(y)$ isn't invertible, the plaintext z can't be recovered (decryption failure). This happens with a probability of about $1/q$.

To decrease the probability of decryption failures, we use the SimpleMatrix scheme over large fields K (for instance, $K \in \{\mathrm{GF}(2^{16}),\mathrm{GF}(2^{32})\}$). Furthermore, several techniques have been proposed to reduce the probability of decryption failures. However, a general solution to this problem is still missing. On the other hand, public-key encryption schemes are used mainly for the key establishment of symmetric ciphers such as Advanced Encryption Standard. Therefore, if one key can't be transmitted correctly, it's easy to replace it with another plaintext.

The SimpleMatrix encryption scheme's security has been carefully studied in "An Asymptotical Optimal Attack on the ABC Multivariate Encryption Scheme."[18] Table 2 shows practical parameters for the SimpleMatrix scheme.

Other Schemes

Developing secure and efficient multivariate encryption schemes is difficult, and many proposed schemes have been broken. Of the BigField family's multivariate encryption schemes, only the PMI+[19] and the IPHFE+ schemes survived; however, owing to several modifications needed to secure the schemes, they aren't very efficient. Recently, the HFE– scheme has been considered as a candidate for a multivariate encryption scheme for small values of the minus parameter. However, similar to IPHFE+ and PMI+, its efficiency is quite bad. Furthermore, due to a new attack,[20] the security of HFE– as an encryption scheme seems to be questionable. In the past few years, several new candidates for multivariate encryption schemes have been proposed. Besides the SimpleMatrix scheme, there are the SRP[21] (Square, Rainbow, Plus) and the ZHFE[22] (Zhuang-Zi Hidden Field Equations) schemes. We refer the reader to the original papers to learn more about these schemes.

Security of Multivariate Schemes

Most multivariate public-key schemes don't have a formal security proof, but we've built very strong theoretical and practical security analysis tools. The theoretical analysis matches the experimental results, which isn't necessarily the case for other families of postquantum cryptosystems. Here, we give a very brief introduction to the main cryptanalysis techniques for MPKCs, which are the basis for how we select the parameters for the MPKCs.

Attacks against multivariate schemes can be divided into two main groups: direct and structural. In direct attacks, one tries to solve the public system $P(z) = w$ directly as an instance of the MQ problem. The most common way to do this is by a Gröbner basis attack, such as Faugères F4 and F5 algorithms.[23] For $m = n$, the complexity of these algorithms is exponential in the number of equations. Table 3 shows the minimal number of equations in a determined system ($m = n$) needed to reach given security levels for different underlying fields.

In a structural attack, one tries to utilize the special structure of a multivariate scheme's central map to recover the composition of the public key into $P = S \circ F \circ T$. Two well-known examples for such an attack are rank and differential attacks. The MinRank attack aims to find a linear combination of the quadratic forms associated to the public-key polynomials of low rank.[24] Such a linear combination corresponds to a central polynomial. By finding linear combinations of low rank, it's possible to recover the multivariate cryptosystem's private key. A differential attack searches for symmetries or invariants of the differential $G(x,y) = P(x + y) - P(x) - P(y) + P(0)$ of the public key.[18] These invariants can be used to analyze the scheme's structure and recover the central map.

Comparison

Compared to other postquantum cryptosystems, multivariate schemes offer several advantages.

- *Speed*. Multivariate schemes can be implemented very efficiently and outperform most of their competitors.[25]
- *Modest computational requirements*. Multivariate schemes require only simple arithmetic operations, such as addition and multiplication, over small, finite fields and therefore can be efficiently implemented on low-cost devices like smartcards and RFID chips,[26] which makes multivariate cryptosystems a promising candidate for Internet of Things security.
- *Short signatures*. Multivariate signature schemes offer very short signatures of a few hundred bits—much shorter than the signatures of other (postquantum and classical) signature schemes.

The main drawback of multivariate schemes is the large size of the public keys. The public-key size of an MPKC is typically about 10 to 100 Kbytes—much

Table 2. Parameters and key sizes of current multivariate encryption schemes.[17]

Security level (bits)	Scheme parameters	Public-key size (Kbytes)	Private-key size (Kbytes)	Plaintext size (bits)	Ciphertext size (bits)
80	SimpleMatrix (GF(2^{16}), 7)	244.0	33.3	784	1,568
100	SimpleMatrix (GF(2^{16}), 8)	536.3	56.6	1,024	2,048
128	SimpleMatrix (GF(2^{16}), 9)	1,076.7	90.5	1,296	2,592

Table 3. Minimal number of equations needed to reach given security levels.[7]

	No. of equations		
Security level (bits)	GF(16)	GF(31)	GF(256)
80	30	28	26
100	39	36	33
128	51	48	43
192	80	75	68
256	110	103	93

larger than that of classical schemes such as RSA and lattice-based cryptosystems. Similar to other families of postquantum cryptosystems, the security of some multivariate schemes is still not completely understood. In terms of provable security, there are few rigorous proofs that reduce the security of multivariate schemes to hard mathematical problems, like the MQ problem. However, in terms of practical cryptanalysis, multivariate public-key cryptosystems have a very solid foundation in the sense that the theoretical estimates of the attack complexities line up nicely with the experimental results,[12,13] which is a very different situation from the case of LLL (Lenstra, Lenstra, Lovacs)-type lattice-reduction attacks on lattice-based cryptosystems. Again, some multivariate schemes such as Rainbow and HFEv– have withstood rigorous cryptanalysis for more than 15 years. Therefore, despite the lack of provable security, we feel very confident that some of the schemes—in particular the signature schemes—are truly viable choices for postquantum cryptographic standards. ■

References

1. P. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM J. Computing*, vol. 26, no. 5, 1997, pp. 1484–1509.
2. M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, 1979.
3. T. Matsumoto and H. Imai, "Public Quadratic Polynomial-Tuples for Efficient Signature Verification and Message Encryption," *Advances in Cryptology*, LNCS 330, Springer, 1988, pp. 419–453.
4. J. Patarin, "The Oil and Vinegar Signature Scheme," Dagstuhl Workshop on Cryptography, 1997.
5. A. Kipnis and A. Shamir, "Cryptanalysis of the Oil-Vinegar Signature Scheme," *Advances in Cryptology* (CRYPTO 99), LNCS 1462, Springer, 1999, pp. 257–266.
6. A. Kipnis, J. Patarin, and L. Goubin, "Unbalanced Oil and Vinegar Signature Scheme," *Advances in Cryptology* (EUROCRYPT 99), LNCS 1592, Springer, 1999, pp. 206–222.
7. A. Petzoldt, "Selecting and Reducing Key Sizes for Multivariate Cryptography," PhD thesis, Dept. Computer Science, Technische Universität Darmstadt, 2013; tuprints.ulb.tudarmstadt.de/3523/1/thesis.pdf.
8. J. Ding and D.S. Schmidt, "Rainbow, a New Multivariate Polynomial Signature Scheme," *Applied Cryptography and Network Security* (ACNS 05), LNCS 3531, Springer, 2005, pp. 164–175.
9. J. Ding et al., "New Differential-Algebraic Attacks and Reparametrization of Rainbow," *Applied Cryptography*

and Network Security (ACNS 08), LNCS 5037, Springer, 2008, pp. 242–257.

10. J. Patarin, "Hidden Field Equations (HFE) and Isomorphism of Polynomials (IP): Two New Families of Assymetric Algorithms," *Advances in Cryptology* (EUROCRYPT 96), LNCS 1070, Springer, 1996, pp. 33–48.
11. J. Patarin, N. Courtois, and L. Goubin, "QUARTZ, 128-Bit Long Digital Signatures," *Topics in Cryptology* (CTRSA 01), LNCS 2020, Springer, 2001, pp. 282–297.
12. J. Ding and B.Y. Yang, "Degree of Regularity for HFEv and HFEv-," *Post-Quantum Cryptography* (PQCrypto 13), LNCS 7932, Springer, 2013, pp. 52–66.
13. A. Petzoldt et al., "Design Principles for HFEv-Based Signature Schemes," *Advances in Cryptology*, Part I (ASIACRYPT 15), LNCS 9452, Springer, 2015, pp. 311–334.
14. M.S. Chen et al., "From 5-Pass Mqbased Identification to MQbased Signature," *Advances in Cryptology* (ASIACRYPT 16), LNCS 10032, Springer, 2016, pp. 135–165.
15. J. Ding et al., "Could SFLASH Be Repaired?," *Automata, Languages, and Programming* (ICALP 08), LNCS 5126, Springer, 2008, pp. 691–701.
16. B.Y. Yang and J.-M. Chen, "Building Secure Tame-Like Multivariate Public-Key Cryptosystems: The New TTS," *Information Security and Privacy* (ACISP 05), LNCS 3574, Springer, 2005, pp. 518–531.
17. C. Tao et al., "Simple Matrix Scheme for Encryption," *Post-Quantum Cryptography* (PQCrypto 13), LNCS 7932, Springer, 2013, pp. 231–242.
18. D. Moody, A. Perlner, and D. Smith-Tone, "An Asymptotical Optimal Attack on the ABC Multivariate Encryption Scheme," *Post-Quantum Cryptography* (PQCrypto 14), LNCS 8772, Springer, 2014, pp. 190–196.
19. J. Ding, "A New Variant of the Matsumoto-Imai Cryptosystem through Perturbation," *Public Key Cryptography* (PKC 04), LNCS 2947, Springer, 2004, pp. 305–318.
20. J. Vates and D. Smith Tone, "Key Recovery Attack for All Parameters of HFE-," *Proc. 8th Int'l Workshop Post-Quantum Cryptography* (PQCrypto 17), LNCS 10346, Springer, 2017, pp. 272–288.
21. T. Yasuda and K. Sakurai, "A Multivariate Encryption Scheme with Rainbow," *Information Security and Cryptology* (ICISC 15), LNCS 9543, Springer, 2015, pp. 222–236.
22. J. Porras, J. Baena, and J. Ding, "ZHFE, a New Multivariate Public Key Encryption Scheme," *Post-Quantum Cryptography* (PQCrypto 14), LNCS 8772, Springer, 2014, pp. 229–245.
23. J.C. Faugere, "A New Efficient Algorithm for Computing Groebner Bases (F4)," *J. Pure and Applied Algebra*, vol. 139, nos. 1–3, 1999, pp. 61–88.
24. D. Coppersmith, J. Stern, and S. Vaudenay, "Attacks on the Birational Signature Scheme," *Advances in Cryptology* (CRYPTO 94), LNCS 773, Springer, 1994, pp. 435–443.
25. A.I.T. Chen et al., "SSE Implementation of Multivariate PKCs on Modern x86 CPUs," *Cryptographic Hardware and Embedded Systems* (CHES 09), LNCS 5747, Springer, 2009, pp. 33–48.
26. P. Czypek, S. Heyse, and E. Thomae, "Efficient Implementations of MQPKS on Constrained Devices," *Cryptographic Hardware and Embedded Systems* (CHES 12), LNCS 7428, Springer, 2012, pp. 374–389.

Jintai Ding is a professor in the Department of Mathematical Sciences at the University of Cincinnati. His main research interests are in cryptography, computational algebra, and information security. Ding received a PhD in mathematics from Yale. He was a co-chair of the Second International Workshop on Postquantum Cryptography. Contact him at jintai.ding@gmail.com.

Albrecht Petzoldt works at NIST. His main research interests comprise multivariate cryptography and postquantum digital signature schemes. Petzoldt received a PhD in computer science from Technische Universität Darmstadt. Contact him at albrecht.petzoldt@googlemail.com.