MULTIVARIABLE PUBLIC–KEY CRYPTOSYSTEMS

JINTAI DING, DIETER SCHMIDT

Keywords: public-key, multivariable polynomials

1. Introduction

Recently Landau and Diffie gave in a series of articles in the Notices of the American Mathematical Society [DL02, Lan01, Lan00a, Lan00b] and in the American Mathematical Monthly [Lan04] excellent expositions on how the theory of multivariable polynomials are used in cryptography. However they covered only half of the story. They covered only the theory of polynomials in symmetric or secret cryptography. There is another half of the story, namely the story about the theory of multivariable polynomials in asymmetric or public key cryptosystems. The importance of the theory of multivariable cryptosystems is manifested in the recent selection of **Sflash** by the Information Society Technologies (IST) Programme of the European Commission for the New European Schemes for Signatures, Integrity, and Encryption project (NESSIE) [NES] as one of the security standards for low-cost smart card. **Sflash** belongs to one of the families of public key cryptosystems, which have been developed in the last ten years.

2. Motivation

As we all know, the revolutionary idea of a public key cryptosystem has fundamentally changed our modern communication system. It was initiated by Diffie and Hellman [DH76] when they provided a protocol for a public key exchange. The first practical realization of a public cryptosystem is the famous RSA cryptosystem by Rivest, Shamir and Adleman [RSA82, RSA78].

In a public key cryptosystem the key consists of two different parts, a public key and a secret key. The public key is accessible to anyone, and it is used either to encrypt a message or to verify the authenticity of an electronic signature. The secret key is used either to decrypt an encrypted message or to produce an electronic signature. This asymmetric design allows one to communicate securely over an open communication channel without any prior exchange of a secret key. For the symmetric key cryptosystem the two parties who want to communicate securely with each other must have the same (symmetric) key and the two parties must have agreed on this key somehow earlier, or they may have used a public key exchange protocol. RSA and other public key cryptosystems avoid this need to exchange keys and they have become a great success to serve the needs of the Internet and our society in terms of providing security and privacy.

However the RSA system also has its weakness. The RSA cryptosystem requires an integer $N = pq$ as a product of two large prime numbers p and q. The security of the system relies on the fact that we do not have a fast algorithm for factoring large integers. Due to the fast development in the field of integer factorization, N should have at least 1024 bits or roughly 340 decimal digits in order to be considered secure today. This requires that in the encryption process one must perform about 1024 times multiplications modular N of two integers with 1024 bits. This is a huge amount of calculation. It makes the communication process slow and inefficient. In practical communication systems, public key systems never stand alone. They are often used for sending keys for another symmetric cryptosystem.

Recently an unexpected threat to the RSA system appeared. Shor [Sho99] developed an algorithm that could factor the integer N on a quantum computer, where the time for factoring increases in a polynomial form with the number of digits of N. This means that if a quantum computer can be built, then the RSA systems are no longer secure. There is a tremendous amount of effort devoted to develop quantum computers. Though, we still do not have a suitable quantum computer for this job, it implies that there exists a strong motivation to search for other more efficient and secure (if there are any) cryptosystems.

3. Multivariable public key cryptosystems

The search for public key cryptosystems has gone into many different directions, for example, elliptic curve cryptosystem, where the structure of elliptic curve is used, lattice cryptosystem, where the structure of lattices is used and many others. Multivariable public key cryptosystem is one of these direction. The building blocks of such systems are multivariable polynomials, in particular, quadratic polynomials. The method relies on the proven theorem that solving a set of multivariable polynomial equations over a finite field is in general an NP-hard problem.

Roughly speaking, the security of the RSA type of cryptosystems relies on the complexity of integer factorization and is based on mathematics developed in the 17th and 18th centuries, namely number theory. Elliptic curve cryptosystems uses the mathematics of the 19th century. Multivariable cryptosystems tries to go one step further, by applying the mathematics of the 20th century, that is, algebraic geometry.

The existing multivariable cryptosystems can roughly be divided into explicit cryptosystems and implicit cryptosystems. Both can be used for one of two purposes: 1) encryption or 2) electronic signature. For encryption schemes all maps must be invertible, so that given an encrypted message we can find uniquely the original message. For signature schemes this requirement can be relaxed as it suffices to see if the signature matches one of a few possible preimages.

We will use $X = (x_1, ..., x_n)$ to denote the standard coordinate system in k^n, and $Y = (y_1, ..., y_m)$ to denote the standard coordinate system in k^m, where k is a suitable finite field.

For encryption we will always use $X' = (x'_1, ..., x'_n)$ to denote an element in k^n, which we will treat as a plaintext (unencrypted secret message), and $Y' = (y'_1, ..., y'_m)$ to denote an element in k^m a ciphertext (encrypted secret message).

In case of electronic signature we will always use $Y' = (y'_1, ..., y'_m)$ to denote an element in k^m as the message to be signed and $X' = (x'_1, ..., x'_n)$ to denote an element in k^n, which is then the electronic signature of the message Y'.

3.1. **Explicit systems.**

In an explicit multivariable public key cryptosystem, we have a map F from k^n to k^m such that

$$\begin{aligned} F(x_1, ..., x_n) &= (F_1(x_1, ..., x_n), ..., F_m(x_1, ..., x_n)) = \\ Y &= (y_1, ..., y_m), \end{aligned}$$

where $F_i(x_1, ..., x_n)$ is a polynomial in $x_1, ..., x_n$.

The key construction for this type of system is that we first build a map f from k^n to k^m such that

$$f(x_1, ..., x_n) = (f_1(x_1, ..., x_n), ..., f_m(x_1, ..., x_n)),$$

where $f_i(x_1, ..., x_n)$ is a polynomial in $x_1, ..., x_n$, and the equation

$$f(x_1, ..., x_n) = (f_1(x_1, ..., x_n), ..., f_m(x_1, ..., x_n)) = (a_1, ..., a_m),$$

can be solved easily. In other words we can find the pre-image of f easily. Note that here f^{-1} means finding the pre-image and does not have the strict mathematical meaning of denoting the inverse of f.

Then F is constructed as:

$$F = L_1 \circ f \circ L_2, \tag{1}$$

where L_1 is a randomly chosen affine invertible linear map from k^m to k^m, $L_1(x_1, .., x_m) = X \times A_1 + C_1$, A_1 is an $m \times m$ invertible matrix and $C_1 \in k^m$; and L_2 is an (affine) invertible linear map from k^n to k^n, $L_2(x_1, .., x_n) = X \times A_2 + C_2$, A_2 is an $n \times n$ invertible matrix and $C_2 \in k^n$.

In this case, the public key consists of the m polynomial components of F and the field structure of k. The secret key mainly consists of L_1 and L_2. The key idea is that L_1 and L_2 serve the purpose of "hiding" the map f, which otherwise could be solved easily. In some systems the function f may be well known, whereas in others f itself might be kept secret.

In order to encrypt a message X', one calculates $F(X')$. To decrypt a message Y', one solves the equation

$$F(x_1, ..., x_n) = Y'. \tag{2}$$

In the case of electronic signature, to sign a message Y', one solves the equation (2), whose solution we denote by X'. To verify if it is a legitimate signature, one just needs to check if indeed

$$F(x'_1, ..., x'_n) = Y'.$$

Due to the design, we can see that we can find the pre-image of Y' by applying in order $(L_1)^{-1}$, f^{-1} and $(L_2)^{-1}$.

3.2. **Implicit systems.** In an implicit multivariable public key cryptosystem, we have a set of l equations in the form of

$$\begin{gathered} H(X, Y) = H(x_1, ..., x_n, y_1, ..., y_m) = \\ (H_1(x_1, ..., x_n, y_1, ..., y_m), ..., H_l(x_1, ..., x_n, y_1, .., y_m)) = (0, ..., 0), \end{gathered} \tag{3}$$

where $H_i(x_1, ..., x_n, y_1, ..., y_m)$ is a polynomial in $x_1, ..., x_n, y_1, .., y_m$.

The key construction here is to build first an equation in the form of

$$\begin{gathered} h(X, Y) = h(x_1, ..., x_n, y_1, ..., y_m) = \\ (h_1(x_1, ..., x_n, y_1, ..., y_m), ..., h_l(x_1, ..., x_n, y_1, .., y_m)) = (0, ..., 0), \end{gathered}$$

where $h_i(x_1, ..., x_n, y_1, ..., y_m)$ is a polynomial in $x_1, ..., x_n, y_1, .., y_m$. There are two requirements:

- For any given specific element X', we can easily solve the equation

$$h(x'_1, ..., x'_n, y_1, ..., y_m) = (0, ..., 0), \tag{4}$$

  whose solution we denote by $Y' = (y'_1, ..., y'_m)$, and
- for any given specific element Y', we can easily solve the equation

$$h(x_1, ..., x_n, y'_1, ..., y'_m) = (0, ..., 0), \tag{5}$$

  whose solution we denote by $X' = (x'_1, ..., x'_n)$.

In most cases, (4) is actually a set of linear equations and (5) is a set of specially designed nonlinear equations.

Then the equation H is constructed as

$$H = L_3 \circ h(L_2(X), L_1(Y)) = (0, ..., 0)$$

where L_1 and L_2 are defined as in the explicit case and L_3 is an invertible linear map from k^l to k^l.

In order to encrypt a message X', one plugs X' into the equation (3). Then one solves the equation:

$$H(X', Y) = H(x_1, ..., x_n, y_1, ..., y_m) = (0, ..., 0),$$

and the solution will be denoted by Y', which is the encrypted message, the ciphertext.

In order to decrypt the message Y', one first calculates first $\bar{Y}' = L_2^{-1}(Y')$, then plugs $\bar{Y}'$ into the equation (5). Then one solves the equation:

$$h(X, \bar{Y}') = h(x_1, ..., x_n, \bar{y}'_1, ..., \bar{y}'_m) = (0, ..., 0),$$

The solution will be denoted by $\bar{y}'$. The plaintext is given by $Y' = (L_2)^{-1}(\bar{y}')$.

For an electronic signature, in order to sign a message Y', one goes through the decryption process above to find an element X' in k^n. To verify if it is a legitimate signature, one just needs to check if indeed

$$H(x'_1, ..., x'_n, y'_1, ..., y'_m) = (0, ..., 0).$$

In this case, the public key consists of the l polynomial components of H and the field structure of k. The secret key mainly consists of L_1, L_2 and L_3. Depending on the case the equation $h(X, Y) = (0, ..., 0)$ is either known or can be made a part of the secret key.

Again the key idea is that L_1, L_2, L_3 serve the purpose to "hide" the equation $h(X, Y) = 0$, which otherwise could be easily solved for a given value of Y.

3.3. **Basic Security and Efficiency Assumptions.**

The most important concerns for multivariable cryptosystems are their security and efficiency. We will discuss the basic aspects of these issues in the context of encryption systems as the case of signature schemes is very similar.

Any encryption process basically applies a map from k^n to k^m to an element in k^n and the decryption process is to find its "inverse", that is, to solve the equation (2). *This means that the equation (2) must be hard to solve*, which is basically ensured by the well-known fact that the Groebner Basis method in general is of exponential complexity and therefore not very efficient. If indeed the encryption has an inverse, which can be expressed itself as a polynomial map, then we must ensure that *this inverse map must have a very high degree*, otherwise one can use the public key to generate enough pairs of plaintext and ciphertext to find the inverse easily [Dic92]. From the construction itself we must also ensure that *it is hard to factorize the encryption map in the specific form of (1).* This is in general difficult because factorization of multivariable maps is an extremely hard topic, partially due to the famous Jacobian conjecture about invertible maps.

Surely any public key cryptosystems is intended for practical applications. This requires that the whole encryption and decryption process must be performed efficiently. The public key is a set of multivariable polynomials, which first has to be transmitted and stored and then values of these polynomials have to be calculated. Thus, these polynomial components F_i must be of a small degree (but not linear, otherwise the system will be unusable.) This means the best choices are quadratic polynomials. Would a system with higher degree polynomials be more secure? The answer is essentially no. The reason is the well-known mathematical trick that we can transform any set of multivariable polynomial equations into another set of quadratic multivariable polynomial equations but with more variables and equations.

4. Multivariable cryptosystems

4.1. The first examples.

The first construction of multivariable signature cryptosystem was given in [OSS84]. This system is based on a quadratic equation

$$x_1^2 + kx_2^2 = m \bmod n, \tag{6}$$

where n is a large composite integer that is difficult to factorize. To sign any message m, one needs to find one of the many (about n) solution pairs (x_1, x_2), which is easy if one knows the factorization of n. The public key is essentially the integer n and the equation (6). Because the security relies on the factorization of n, in some sense, this system is still in the shadow of the RSA cryptosystem, though it initiated the idea of multivariable cryptosystem. Unfortunately shortly afterwards, Pollard and Schnorr [PS87] broke this cryptosystem. They found an algorithm to solve the equation (6) for any given m without the use of the factorization of n. In particular when k and m are relatively prime to n, a solution can be found easily; and with the assumption of the generalized Riemann hypothesis, a solution can be found by a probabilistic algorithm with a complexity of $O\{(\log n)^2 |\log\log|k||\}$ of $O(\log n)$-bit integer operations.

4.2. Triangular cryptosystems.

Another early attempt to build a multivariable cryptosystem was by Diffie and Fell [FD86]. Their idea is to build a cryptosystem using the composition of many invertible linear maps and simple triangular maps in the form of

$$T(x_1, ..., x_n) = (x_1 + g(x_2, ..., x_n), x_2, ..., x_n), \tag{7}$$

where g_i is a polynomial. Clearly T is invertible and therefore the decryption process can be done. However due to consideration of efficiency, in particular, the key size, the authors themselves concluded that *"there seems no way to build such a system that is both secure and has a public key of practical size".*

Here one should notice that the simple triangular map above belongs to the family of de Jonquières maps defined as

$$\begin{aligned} J(x_1, .., x_n) &= (x_1 + g_1(x_2, \dots, x_m), x_2 + g_2(x_3, \dots, x_n), \\ & \qquad \dots, x_{n-1} + g_{n-1}(x_n), x_n), \end{aligned} \tag{8}$$

where g_i are polynomial functions. Clearly J can be easily inverted. All invertible affine linear maps over k^n and the de Jonquières maps are called tame transformations in algebraic geometry. They also include all transformations which are formed by composition of tame transformations. Tame transformation are elements of the group of automorphism of the polynomial ring $k[x_1, ..., x_n]$. Elements, which are in this group and are not tame, are called wild. This topic is closely related to the famous Jacobian conjecture in algebraic geometry, which essentially asks, if the Jacobian of a multivariable map, which is a nonzero constant, also implies that the map is indeed invertible. This is a long standing difficult question in mathematics. Even to find if a map is tame is a very difficult problem [Nag72].

One can see that de Jonquières maps have two types, one is upper triangular as the one above and similarly we can also define the lower triangular type. Triangular construction was not pursued again until 10 years later when Moh [Moh99] suggested a construction where the quadratic map f is given by

$$f = J_u \circ J_l \circ I(x_1, ..., x_n). \tag{9}$$

Here J_u is a k^m upper triangular de Jonquières map and J_l is a k^m lower triangular de Jonquières map and the linear map I is the embedding of k^n into k^m: $I(x_1, ..., x_n) = (x_1, ..., x_n, 0, 0, ..., 0)$. The main achievement of such a construction is that f is a quadratic function and that any linear combination of the components of f can not produce a linear function. The trick of this construction is actually the use of the map I. One can see that

$$\begin{aligned} J_l \circ I(x_1, ..., x_n) &= (x_1, x_2 + g_1(x_1), ..., x_n + g_{n-1}(x_1, ..., x_{n-1}), \\ &\quad\ g_n(x_1, .., x_n), ..., g_{m-1}(x_1, ..., x_n)), \end{aligned}$$

which gives us the freedom to choose any g_i, $i = n, .., m-1$. This method is named the tame transformation method (TTM). A few examples of such constructions were given and a family of challenges with monetary award was set up in a web (www.usdsi.com). Despite the inventor's claim that TTM systems are very secure from all standard attacks, shortly afterwards Courtois and Goubin [GC00] used the method of Minrank to attack this system. This method searches for the matrix of the minimum rank among the space of linear span of a few given matrices. What they did first is associate a quadratic polynomial with a bilinear form and therefore its corresponding matrices, then they used the Minrank method to work on these matrices. They easily broke one of the challenges and claimed that the TTM systems could not work due to the Minrank attack method. However the inventor of TTM refuted this claim and together with Chen gave a new implementation of his scheme in [CM01].

In [DH] another method was found to defeat the first TTM implementation scheme of [Moh99]. This attack method can also be applied to other TTM implementation schemes [CGJ02]. Later, Ding and Schmidt [DS03a, DS03b] found out that actually all currently existing implementation schemes for the TTM cryptosystem have a common defect that could make them insecure. The conclusion comes from observing that we can extend the linearization method by Patarin [Pat95] to attack all current TTM implementation schemes. The problem lies in the fact that although the TTM constructions is a very original idea, the existing constructions of the TTM cryptosystem are not done in a systematic way and no explanation is given why and how they work. From what we can see at the moment simple TTM systems do not work. More sophisticated constructions are needed and they may require deep insight from algebraic geometry.

Attempts were made to use a similar but simpler idea for signature schemes, which was called a TTS (tamed transformation signature.) This system is essentially the result of an application of the Minus method in [Sha98] for a tame transformation. A few of them were suggested mainly by Chen and his collaborators [YC03, CYP02]. One sees fairly easily that these systems can also be defeated by the method using the descending chain of the ranks of quadratic forms by Coppersmith, Stern, and Vaudenay [CSV97].

4.3. **Matsumoto-Imai systems.** Another idea to design a multivariable cryptosystem was started by Matsumoto and Imai [MI88], where the key idea is that one should use a map $\bar{f}$ over a large field $\bar{K}$, a degree n extension of a finite field k (with characteristic 2). Through a map ϕ which identifies $\bar{K}$ as k^n first, one would identify this map as a multivariable polynomial map f from k^n to k^n:

$$f = \phi \circ \bar{f} \circ \phi^{-1}. \tag{10}$$

Then, one would "hide" this map f by composing from both sides by two invertible affine linear maps L_1 and L_2 on k^n. The map $\bar{f}$ suggested by Matsumoto and Imai is the map

$$\bar{f} : X \longmapsto X^{1+q^i}, \tag{11}$$

where q is the number of elements in k, X is an element in $\bar{K}$, k is of characteristic 2, and such that $\gcd(1+q^i, q^n-1) = 1$. The last condition ensures that the map $\bar{f}$ can be easily inverted. The inverse of the map f is given by

$$\bar{f}^{-1}(X) = X^t, \tag{12}$$

where $t(1+q^i) = 1 \bmod (q^n-1)$. This ensures that we can decrypt any secret message easily by the inverse. One more important thing is that the map f is actually quadratic due to the property of the Frobenius map $X \to X^{q^i}$.

However Patarin [Pat95] found out that for this family of cryptosystem, due to the properties of the map $\bar{f}$, the cipher satisfies a large number of linearly independent equations of the following form:

$$\begin{aligned} \sum a_{ij} f_i(x_1,...,x_n) x_j + \sum a_i f_i(x_1,...,x_n) + \sum b_j x_j + c = \\ = \sum a_{ij} y_i x_j + \sum a_i y_i + \sum b_j x_j + c = 0, \end{aligned} \tag{13}$$

which are called the linearization equations. In this case, if we are given values of the secret message, the values of $y_i = f_i$, they will produce linear equations satisfied by the secret message component x_i, which therefore allows us to find x_i easily.

Though the original idea of Matsumoto-Imai failed, it has inspired many new designs most of them coming from Patarin and his collaborators.

4.4. **Minus-Plus generalizations of the Matsumoto-Imai system.** It is a simple idea, as one takes out a few of the quadratic polynomial components of $\bar{F}$ (Minus method), and/or one adds (Plus method) a few randomly chosen quadratic polynomials [PGC98]. The Minus method was first suggested in [Sha98]. The main reason to take the "Minus" action is to improve the security. Shamir [Sha98] realized that even if a set of equations is easy to solve, if the number of equations is reduced (Minus), the new set of equations could be much harder or impossible to solve. The Minus (only) method is very suitable for signature schemes, because it does not require that a document has a unique signature unlike the case of decryption process. Sflash [ACDG03, PCG01a] is a Matsumoto-Imai-Minus cryptosystem. Although the original submission of Sflash to the NESSIE project had a minor design flaw it made it to the final selection. The design flaw was pointed out in [GM02] where a property of difference equations related to permutation polynomials was used to search for the missing polynomials. The author of Sflash recently improved the system and suggested a new version [CGP03].

4.5. **Hidden Field Equation Method.** (HFE) This method is suggested by Patarin to be the strongest [Pat95, CDF03]. In this case, the difference from the original Matsumoto-Imai system is that $\bar{f}$ is replaced by the more general map

$$\bar{f} : X \longmapsto \sum_{i,j}^{A} a_{ij} X^{q^i+q^j} + \sum_{i}^{A} b_i X^{q^i} + c, \tag{14}$$

where the coefficients are chosen at random. The decryption process involves solving the equation $\bar{f} = Y'$ for X. This can be done for example with the algorithm of Berlekamp, whose time complexity is proportional to the cube of the degree of $\bar{f}$. If the degree of $\bar{f}$ is large then the system is too slow due to the process of solving the polynomial equation in the decryption process, but Kipnis and Shamir [KS99] show that the degree can not be too small either. The key of their attack [KS99] is that they realize that one can lift any k^n to k^n map to be a map $\bar{K}$ to $\bar{K}$ map. Then they treat the quadratic part of $\bar{f}$ as a bilinear from and use the Minrank method to attack the system. These findings have been confirmed by [Cou01, FJ03].

4.6. **Vinegar-Oil method.** The Oil and Vinegar schemes [Pat97] and the unbalance Oil and Vinegar schemes [KPG99] are suitable for signature.

Let o and v be two constant integers. Let $x_1, ..., x_o$ be o variables, which we call oil variables and $\hat{x}_1, ..., \hat{x}_v$ be v variables, which we call vinegar variables.

Let f be a map from k^{o+v} to k^o and

$$f(x_1, .., x_o, \hat{x}_1, ..., \hat{x}_v) = (f_1(x_1, .., x_o, \hat{x}_1, ..., \hat{x}_v), ..., f_o(x_1, .., x_o, \hat{x}_1..., \hat{x}_v)),$$

$$f_l(x_1, .., x_o, \hat{x}_1, ..., \hat{x}_v) = \sum_{i,j=1}^{o,v} a_{lij} x_i \hat{x}_j + \sum_{i,j=1}^{v} b_{lij} \hat{x}_i \hat{x}_j + \sum_{i=1}^{o} c_{li} x_i + \sum_{j=1}^{v} d_{lj} \hat{x}_j + e_l,$$

where all coefficients are randomly chosen from the field k. Here we notice that there are no quadratic terms of oil variables, which means the oil variables and vinegar variables are not fully mixed (like oil and vinegar) and this explains the name of this scheme.

The cipher F is constructed as usual:

$$F = L_1 \circ f \circ L_2,$$

where L_1 and L_2 are invertible affine linear maps on the corresponding spaces. In some way, here the change of basis is a process to "mix" fully oil and vinegar, so one can not see what is oil and what is vinegar.

The case $o = v$ is the original Oil and Vinegar signature scheme, and when $o < v$, it is the unbalance Oil and Vinegar signature scheme. The public key are the polynomial components of F, namely F is itself, not the composition components, and the field structure of k. The secret key consists of the linear maps L_i and the map f.

Given a message $Y' = (y'_1, ..., y'_o)$, to sign it, we need to try to find a vector $X' = (x'_1, ..., x'_{o+v})$ such that $F(X') = Y'$.

With the secret key it can be done easily. Since L_1 and L_2 can be inverted, we only need to find a way to "invert" f, namely to find a pre-image. To invert f one first guesses all the values of $\hat{x}_i$, namely all the vinegar variables. Thus, one obtains a set of o linear equations with o variables. With a very high probability it has a solution. If it does not have a solution, one tries another set of values of the vinegar variables until one finds a pre-image of a given element in k^o. This process can be done easily.

To check if X' is indeed a legitimate signature for Y', we only need to get the public map f and check if indeed $F(X') = Y'$.

To make a forgery of a signature, one needs to solve the equation $F(X') = Y'$. It turns out that it can be done easily if q^{v-o} is small due to attack by Kipnis and Shamir and later [KPG99]. The basic idea here is that we treat each f_i as a bilinear form. The corresponding matrix is in the form of $\binom{0 \quad *}{* \quad *}$. This reduces the problem to finding a basis change for a set of bilinear forms. This construction is inspired by the Minus method of Shamir [Sha98] and the idea of linearization equation.

4.7. **HFEV.** It is also a possible to combine different constructions like the HFE and VO schemes. In the same paper [KPG99] where the unbalance Oil and Vinegar scheme was presented a new scheme called HFEV was suggested. The basic idea is to add on top of the HFE method a few new variables to make the system more complicated. This method essentially replaces F with an even more complicated function:

$$F : (X, \bar{X}) \longmapsto \tag{15}$$

$$\sum_{i,j}^{A,A} a_{ij} X^{q^i+q^j} + \sum_{i,j}^{A,B} b_{i,j} X^{q^i} \bar{X}^{q^j} + \sum_{i,j}^{B,B} \alpha_{ij} \bar{X}^{q^i+q^j} + \sum_{i}^{A} b_i X^{q^i} + \sum_{i}^{B} \beta'_i \bar{X}^{q^i} + c,$$

where the new vinegar variables given by $\bar{X}$ are of a small dimension when viewed in k^n, that is, $\phi^{-1}(\bar{X}) = (\bar{x}_1, ..., \bar{x}_v, 0, ..., 0)$ with v a small number.

These new variables are also mixed in a special way with the original variables. When the new variables are given specific values, equation (15) reverts back to form (14) but with different coefficients. The decryption process requires an exhaustive search on the added small number of (vinegar) variable. For the signature case the search becomes a random selection, which has a good probability to succeed each time, and it continues until a correct answer is found. We [DS05] recently observed that the attack in [KS99] can also be applied here to actually eliminate the small number of added variables and attack the system. The basic idea is to use the algebraic method to find a way to purge out the vinegar variables. We also [DS05] apply the internal perturbation to the HFE cryptosystem, which works even better than the Matsumoto-Imai cryptosystem.

A signature scheme Quartz was proposed as a HFE-Minus scheme. It has a very short signature of 128 bits [PCG01b]. It still stands against all existing attacks.

4.8. **Perturbation.** From a very general point of view, the methods above (the HFEV and Oil-Vinegar method) can also be interpreted as an extension of a commonly used idea in mathematics and physics, namely perturbation. A good way to deal with a continuous system often is to "perturb" the system at a minimum scale. The HFEV can be viewed as a perturbation of the HFE method by the newly added vinegar variables. However, because of the "Oil-Vinegar" idea it is in some sense more of an "external" perturbation, as a few new (external) variables (Vinegar) are introduced.

This inspired a new idea of internal perturbation [Din04]. The idea was applied to the Matsumoto-Imai system. The new multivariable cryptosystem is called the Perturbed Matsumoto-Imai (PMI) system. A practical scheme was suggested as an implementation of this idea and resulted in a 136 bits open-key cryptosystem.

The theoretical idea of "internal" perturbation is very general and can be applied to all existing multivariable cryptosystems. It appears to be better to perturb the HFE method "internally" rather than by the "external" Oil-Vinegar scheme, as the vector $\bar{X}$ in (15) is replaced by variables from a subspace inside the original k^n and one does not have to introduce any new variables. The security is improved because it is impossible to purge out the perturbation. The reason for this is exactly due to the fact that it is internal and fully mixed into the system unlike in the case of Oil-Vinegar mixing. This idea can also be combined with the Minus method for the construction of signature schemes.

4.9. **Implicit and other systems.** The implicit systems are not as well developed, as most of the research has been devoted to explicit systems. There are two existing families of implicit systems and they are called Little Dragon and Dragon [Pat96, Kob98]. Little Dragon is a simplified version of Dragon. These constructions are very much inspired by the linearization equations and the Matsumoto-Imai cryptosystems and they are essentially a combination of these two ideas. The little Dragon is defeated by Coppersmith and the Dragon can be defeated by the same method as in [KS99].

5. Illustrative example

All crypto system require extensive computations. Therefore, it is difficult to give a simple and at the same time meaningful example in order to illustrate a method. This is true in particular for multivariate cryptosystems where many lines of text are needed to display the quadratic equations.

The following toy example for a Matsumoto–Imai system will use the finite field $k = GF[2]/(x^2 + x + 1)$, with 2^2 elements, which we will denote 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, $\mathit{3}$ for $1 + x$. In this case, $\mathit{1} + \mathit{3} = \mathit{2}$ and $\mathit{2} * \mathit{3} = \mathit{1}$.

For the larger field we use $\bar{K} = k[y]/(y^3 + y + \mathit{1})$. With $n = 3$ the only option for θ is $\theta = 2$. The Matsumoto-Imai map and its inverse is thus

$$\bar{f}(X) = X^{1+4^2} \qquad\qquad \bar{f}^{-1}(X) = X^{26}$$

The mappings L_1 and L_2 in (1) represent the secret keys and we select for L_1

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \mathit{3} & \mathit{2} & \mathit{2} \\ \mathit{2} & \mathit{1} & \mathit{0} \\ \mathit{1} & \mathit{0} & \mathit{1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \mathit{0} \\ \mathit{1} \\ \mathit{2} \end{bmatrix},$$

and for L_2

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \mathit{1} & \mathit{0} & \mathit{2} \\ \mathit{0} & \mathit{1} & \mathit{2} \\ \mathit{1} & \mathit{2} & \mathit{0} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \mathit{2} \\ \mathit{3} \\ \mathit{3} \end{bmatrix}.$$

In constructing the public key we also denote by $(x_0, x_1, x_2) \in k^3$ the message. The map $\phi \circ L_2$ is given by

$$u = \mathit{2} + x_0 + \mathit{2}x_2 + (\mathit{3} + x_1 + \mathit{2}x_2)y + (\mathit{3} + x_0 + \mathit{2}x_1)y^2.$$

Next compute u^{4^2}, which is easily done, since $(u(y))^{16} = u(y^{16})$ in the finite field k. Now compute the product

$$\begin{aligned} &v = u^{16}u = \mathit{1} + \mathit{3}x_0 + \mathit{2}x_1 + x_2 + x_0x_1 + \mathit{2}x_0x_2 + \mathit{3}x_1x_2 \\ &+(\mathit{2} + \mathit{2}x_0 + x_1 + \mathit{3}x_2 + x_0^2 + \mathit{3}x_0x_1 + x_1^2 + x_1x_2)y \\ &+(\mathit{3} + \mathit{3}x_0 + \mathit{2}x_1 + \mathit{2}x_2 + x_0^2 + x_0x_1 + \mathit{2}x_0x_2 + \mathit{3}x_1^2 + \mathit{2}x_1x_2 + \mathit{3}x_2^2)y^2. \end{aligned}$$

Finally compose it with L_1 to obtain

$$\begin{aligned} y_0 = f_0(x_1, x_2, x_3) &= \mathit{1} + x_2 + \mathit{2}x_0x_2 + \mathit{3}x_1^2 + \mathit{3}x_1x_2 + x_2^2, \\ y_1 = f_1(x_1, x_2, x_3) &= \mathit{1} + \mathit{3}x_0 + \mathit{2}x_1 + x_2 + x_0^2 + x_0x_1 + \mathit{3}x_0x_2 + x_1^2, \\ y_2 = f_2(x_1, x_2, x_3) &= \mathit{3}x_2 + x_0^2 + \mathit{3}x_1^2 + x_1x_2 + \mathit{3}x_2^2. \end{aligned}$$

This is the public key together with the field structure of k.

For example, given the plaintext $x_0 = 1,\ x_1 = 2,\ x_2 = 3$, we plug them into f_1, f_2, f_3, and it produces the ciphertext $y_0 = 0$, $y_1 = 0$, $y_2 = 1$.

In order to recover the plaintext, we evaluate it with the inverse function $L_2^{-1} \circ \bar{f}^{-1} \circ L_1^{-1}$. Since the private key can be used directly, the function is evaluated in three stages. The exponentiation required by $\bar{f}^{-1}$ can be done by the 'square and multiply (binary) method'. This method is not always optimal and it can be time consuming when the exponent is very large. Fortunately it is possible to find faster ways to carry out the exponentiation by exploiting the form of the exponent. A practical system has to use values of n and θ where this can be done.

In order to illustrate the Minus method, as it is used for example in Sflash, the linear map L_1 is combined with a projection. Here we select a projection onto the first two coordinates so that the public key is now

$$\begin{aligned} y_0 = f_0(x_1, x_2, x_3) &= 1 + x_2 + 2x_0x_2 + 3x_1^2 + 3x_1x_2 + x_2^2, \\ y_1 = f_1(x_1, x_2, x_3) &= 1 + 3x_0 + 2x_1 + x_2 + x_0^2 + x_0x_1 + 3x_0x_2 + x_1^2, \end{aligned}$$

together with the field structure of k. The system can no longer be used for encryption, but it is very useful for signing a message. In our case a message would have to be compressed or hashed onto the first two components of the vector $\mathbf{y} = (y_0, y_1, y_2)$, that is into 4 bits. The third component y_2 can be selected at random at each signing and can be treated as a personal secret key. Signing the message $\mathbf{y}$ means computing first

$$\mathbf{x} = L_2^{-1} \circ f^{-1} \circ L_1^{-1}(\mathbf{y}).$$

Verifying a signature means that when the cipher $\mathbf{x} = (x_0, x_1, x_2)$ is substituted into the public key the original message y_0 and y_1 is recovered.

In our toy example let the message be $y_0 = 1$ and $y_1 = 3$. The personal key $y_2 = 0$ produces the cipher $\mathbf{x} = (2, 0, 2)$, the key $y_2 = 1$ produces $\mathbf{x} = (1, 0, 0)$, the key $y_2 = 2$ gives $\mathbf{x} = (0, 1, 1)$, and $y_2 = 3$ gives $\mathbf{x} = (2, 0, 0)$. In all four cases, when these values are plugged into the public key, the message $y_0 = 1$ and $y_1 = 3$ is recovered and with it the signature is verified.

6. Computational Complexity

Surely one of the main purposes of cryptography is to develop systems that can be used in real life. In general, the key size of multivariable cryptosystem are much larger than the key size for RSA cryptosystems. A public key of 1024 bits is recommended today for RSA and it requires only a storage of 128 bytes. But with it one also needs a computer program to do the extensive computations efficiently. Sflash^{v_3} is a Matsumoto–Imai–Minus system. It uses the finite field $k = Z_2\,[x]\,/(x^7 + x + 1)$ and is defined as the mapping $F^- \ :\ k^{67} \to k^{56}$. The notation F^- indicates that 11 equations have been removed from the function F, which is constructed as usual by

$$F = L_1 \circ f \circ L_2.$$

Here L_1 and L_2 are two invertible affine linear transformation and $\bar{f}$ as defined in (10) and (11) is given by

$$\bar{f}(X) = X^{1+128^{33}}. \tag{16}$$

The public key consists therefore of 56 quadratic polynomials in 67 variables with coefficients in k. Each quadratic polynomial will have $67 \times 34 + 67 + 1$ coefficients. This requires 128.3 KB of storage if each coefficient is stored in a single byte, and it can be reduced to 112.3 KB if only 7 bits are used for each coefficient. This is a fair amount of storage, but it should not be a problem for PC's or smart cards.

For the secret key it suffices to store the 9144 coefficients of the two linear transformations and to have a way to evaluate the function (12). Since $t =$ 768146788955706749847833964023909031926328283136065735939753188884 195798396896924678452624605588574118228052288763643673842596138395 343249344 it appears to be lot of work, but looking at this number in hexadecimal $t = (102040810204081020408102040810204081020408102040810204 08101fbf7efdfbf7efdfbf7efdfbf7efdfbf7efdfbf7efdfbf7efdfbf7efdfc0)_{16}$ one sees that there are repeated patterns and that the function (12) can be evaluated much more efficiently than by the standard square and multiply method.

The computational complexity for multivariable cryptosystems are typically much less than for the RSA system. For example in Sflash^{v3} the evaluation of the quadratic polynomials is done in the field k. Multiplications can be done either with the help of a table with 128×128 entries or with the help of logarithms for which an array with only 127 entries is needed. Addition of elements in k is even simpler as it is the exclusive or of the two numbers in binary.

7. The Future

Though indeed many of the multivariable cryptosystem are broken, many still stand and the new designs are getting stronger and stronger. Since the theory behind multivariable cryptosystems matured quickly in the last decade, there exists a great potential for the practical applications of these ideas. Since so many different schemes have been proposed in recent years our bibliography is not comprehensive, and we apologize to those whose method we have not mentioned. Also interesting schemes like [PL97, YDL01] have not been mentioned here, although they are related to multivariable cryptosystems they use different ideas.

From the mathematical point of view, one can see that the developments in the area of multivariable cryptosystem, also brought a lot of progress in the related mathematical fields. Just by looking at the attack methods alone, we see many new and old ideas: linearization equations [KS99, DS03b], relinearization equations [KS99], use of ideals in the XL method [CP03], Minrank method [Kob98], new Groebner basis [FJ03], quadratic forms on finite fields [Pat95, CKPS00], theory of permutation polynomials [GM02].

The development of the research in multivariable cryptosystems, we believe, will continue to be a strong drive to develop further the theory of functions over finite fields, especially in terms of computational complexity. In the field of multivariable cryptosystem, there are many unsolved problems (see www.minrank.org). There also exist many ideas that today can only be verified by computations and not by a mathematical argument. We believe, new mathematical insights especially insight from algebraic geometry will be fundamental to deal with these problems.

References

[ACDG03] Mehdi-Laurent Akkar, Nicolas T. Courtois, Romain Duteuil, and Louis Goubin. A fast and secure implementation of Sflash. In *PKC-2003, LNCS*, volume 2567, pages 267–278. Springer, 2003.

[CDF03] Nicolas T. Courtois, Magnus Daum, and Patrick Felke. On the security of HFE, HFEv- and Quartz. In *PKC-2003, LNCS*, volume 2567, pages 337–350. Springer, 2003.

[CGJ02] G. Chou, J. Guan, and Chen J. A systematic construction of a Q_{2^k}-model in TTM. *Comm. Algebra*, 30:551–562, 2002.

[CGP03] Nicolas Courtois, Louis Goubin, and Jacques Patarin. Sflashv3, a fast asymmetric signature scheme, 2003. http://eprint.iacr.org.

[CKPS00] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In B. Preenel, editor, *Advances in cryptology, Eurocrypt 2000, LNCS*, volume 1807, pages 392–407. Springer, 2000.

[CM01] J. Chen and T. Moh. On the Goubin-Courtois attack on TTM. *Cryptology ePrint Archive*, 72, 2001. http://eprint.iacr.org/2001/072.

[Cou01] Nicolas T. Courtois. The security of hidden field equations (HFE). In C. Naccache, editor, *Progress in cryptology, CT-RSA, LNCS*, volume 2020, pages 266–281. Springer, 2001.

[CP03] Nicolas Courtois and Jacques Patarin. About the XL algorithm over $GF(2)$. In *LNCS*, volume 2612, pages 141–157. Springer, 2003.

[CSV97] D. Coppersmith, J. Stern, and S. Vaudenay. The security of the birational permutation signature schemes. *J. Cryptology*, 10(3):207–221, 1997.

[CYP02] J. Chen, B. Yang, and B. Peng. Tame transformation signatures with Topsy-Yurvy Hashes. In *IWAP'02*, 2002.

[DH] Jintai Ding and Timothy Hodges. Cryptanalysis of an implementation scheme of TTM. http://eprint.iacr.org.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[Dic92] Matthew Dickerson. The inverse of an automorphism in polynomial time. *J. Symbolic Comput.*, 13(2):209–220, 1992.

[Din04] Jintai Ding. A new variant of the Matsumoto-Imai cryptosystem through perturbation. In F. Bao, R. Deng, and J. Zhou, editors, *Public Key Cryptosystems, PKC-2004, LNCS*, volume 2947, pages 305–318. Springer, 2004.

[DL02] W. Diffie and S. Landau. September 11th did not change cryptography policy. *Notices of the American Mathematical Society*, 49:450–454, 2002.

[DS03a] J. Ding and D. S. Schmidt. A common defect of the TTM cryptosystem. In *Proceedings of the technical track of the ACNS'03, ICISA Press*, pages 68–78, 2003. http://eprint.iacr.org.

[DS03b] J. Ding and D. S. Schmidt. The new TTM implementation is not secure. In H. Niederreiter K.Q. Feng and C.P. Xing, editors, *Proceedings of International*

Workshop on Coding, Cryptography and Combinatorics (CCC 2003), pages 106–121, 2003.

[DS05] Jintai Ding and D. S. Schmidt. Cryptanalysis of HFEV and the internal perturbation of HFE. In *The 8th International Workshop on Practice and Theory in Public-Key Cryptography, Jan. 2005, Switzerland (PKC'05)*, page accepted. Springer, 2005.

[FD86] Harriet Fell and Whitfield Diffie. Analysis of a public key approach based on polynomial substitution. In *Advances in cryptology—CRYPTO '85 (Santa Barbara, Calif.), LNCS*, volume 218, pages 340–349. Springer, 1986.

[FJ03] Jean-Charles Faugère and Antoine Joux. Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases. In Dan Boneh, editor, *Advances in cryptology – CRYPTO 2003, LNCS*, volume 2729, pages 44–60. Springer, 2003.

[GC00] L. Goubin and N. Courtois. Cryptanalysis of the TTM cryptosystem. *LNCS, Springer Verlag*, 1976:44–57, 2000.

[GM02] Henri Gilbert and Marine Minie. Cryptanalysis of SFLASH. In L. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002, LNCS*, volume 2332, pages 288–298. Springer, 2002.

[Kob98] N. Koblitz. *Algebraic aspects of cryptography.* Springer Verlag, 1998.

[KPG99] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced oil and vinegar signature schemes. In *Eurocrypt'99, LNCS*, volume 1592, pages 206–222. Springer, 1999.

[KS99] Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In M. Wiener, editor, *Advances in cryptology – Crypto '99, LNCS*, volume 1666, pages 19–30. Springer, 1999.

[Lan00a] S. Landau. Communications security for the twenty-first century: the advanced encryption standard. *Notices of the AMS*, 47(4):450–459, April 2000.

[Lan00b] S. Landau. Standing the test of time: the data encryption standard. *Notices of the AMS*, 47(3):341–349, March 2000.

[Lan01] S. Landau. Advanced encryption standard choice is rijndael. *Notices of the AMS*, 48(1):38, January 2001.

[Lan04] S. Landau. Polynomials in the nation's service: Using algebra to design the advanced encryption standard. *American Mathematical Monthly*, 111:89–117, 2004.

[MI88] T. Matsumoto and H. Imai. Public quadratic polynomial-tuples for efficient signature verification and message encryption. In C. G. Guenther, editor, *Advances in cryptology – EUROCRYPT '88, LNCS*, volume 330, pages 419–453. Springer, 1988.

[Moh99] T. T. Moh. A fast public key system with signature and master key functions. *Lecture Notes at EE department of Stanford University.*, May 1999. http://www.usdsi.com/ttm.html.

[Nag72] M. Nagata. *On Automorphism Group of $K[x, y]$*, volume 5 of *Lectures on Mathematics.* Kyoto University, Kinokuniya, Tokyo, 1972.

[NES] NESSIE. European project IST-1999-12324 on New European Schemes for Signature, Integrity and Encryption. http://www.cryptonessie.org.

[OSS84] H. Ong, C.-P. Schnorr, and A. Shamir. Signatures through approximate representations by quadratic forms. In *Advances in cryptology, Crypto '83*, pages 117–131. Plenum Publ., 1984.

[Pat95] J. Patarin. Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88. In D. Coppersmith, editor, *Advances in Cryptology – Crypto '95, LNCS*, volume 963, pages 248–261, 1995.

[Pat96] J. Patarin. Asymmetric cryptography with a hidden monomial. In N. Koblitz, editor, *Advances in cryptology, CRYPTO '96, LNCS*, volume 1109, pages 45–60. Springer, 1996.

[Pat97] J. Patarin. The oil and vinegar signature scheme. *Dagstuhl Workshop on Cryptography, September 1997*, 1997.

[PCG01a] Jacques Patarin, Nicolas Courtois, and Louis Goubin. Flash, a fast multivariate signature algorithm. In *LNCS*, volume 2020, pages 298–307. Springer, 2001.

[PCG01b] Jacques Patarin, Nicolas Courtois, and Louis Goubin. QUARTZ, 128-bit long digital signatures. In C. Naccache, editor, *Progress in cryptology, CT-RSA, LNCS*, volume 2020, pages 282–297. Springer, 2001.

[PGC98] Jacques Patarin, Louis Goubin, and Nicolas Courtois. C^*_{-+} and HM: variations around two schemes of T. Matsumoto and H. Imai. In K. Ohta and D. Pei, editors, *ASIACRYPT'98, LNCS*, volume 1514, pages 35–50. Springer, 1998.

[PL97] J. Patarin and Goubin L. Asymmetric cryptography with S-boxes. In Han Y., Okamoto T., and Qing S., editors, *Proceedings of ICICS'97, LNCS*, volume 1334, pages 369–380. Springer, 1997.

[PS87] John M. Pollard and Claus-P. Schnorr. An efficient solution of the congruence $x^2 + ky^2 = m \pmod{n}$. *IEEE Trans. Inform. Theory*, 33(5):702–709, 1987.

[RSA78] Ronald Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *ACM*, 21(2):120–126, 1978.

[RSA82] Ronald Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public key cryptosystems. secure communications and asymmetric cryptosystems. In G Simmons, editor, *AAAS Sel. Sympos. Ser.*, volume 69, pages 217–239. Westview Press, 1982.

[Sha98] Adi Shamir. Efficient signature schemes based on birational permutations. In *LNCS, Advances in cryptology – CRYPTO '98 (Santa Barbara, CA, 1998)*, volume 1462, pages 257–266. Springer, 1998.

[Sho99] Peter Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.*, 41(2):303–332, 1999.

[YC03] B. Yang and J. Chen. A more secure and efficacious TTS signature scheme. *ICISC'03*, 2003. http://eprint.iacr.org.

[YDL01] Dingfeng Ye, Zongduo Dai, and Kwok-Yan Lam. Decomposing attacks on asymmetric cryptography based on mapping compositions. *J. Cryptology*, 14(2):137–150, 2001.

Department of Mathematical Sciences, Department of Electrical and Computer Engineering and Computer Science, University of Cincinnati, Cincinnati, OH, 45220, USA, ding@math.uc.edu