MQ Challenge: Hardness Evaluation of Solving Multivariate Quadratic Problems *

Takanori Yasuda[1], Xavier Dahan[1,6], Yun-Ju Huang[1,2],
Tsuyoshi Takagi[1,3,4], and Kouichi Sakurai[1,5]

[1] Institute of Systems, Information Technologies and Nanotechnologies
[2] Graduate school of Mathematics, Kyushu University
[3] CREST, Japan Science and Technology Agency
[4] Institute of Mathematics for Industry, Kyushu University
[5] Department of Informatics, Kyushu University
[6] Leading Graduate School Promotion Center, Ochanomizu University

Abstract. Multivariate Quadratic polynomial (MQ) problem serve as the basis of security for potentially post-quantum cryptosystems. The hardness of solving MQ problem depends on a number of parameters, most importantly the number of variables and the degree of the polynomials, as well as the number of equations, the size of the base field etc. We investigate the relation among these parameters and the hardness of solving MQ problem, in order to construct hard instances of MQ problem. These instances are used to create a challenge, which may be helpful in determining appropriate parameters for multivariate public key cryptosystems, and stimulate further the research in solving MQ problem.

Keywords: Post-quantum cryptography, Multivariate public-key cryptosystems, Gröbner basis

1 Introduction

Multivariate public-key cryptosystems [14, 33] (MPKC for short) are candidates for post-quantum cryptography. MPKC are schemes that use multivariate polynomial maps as public keys. The security of MPKC is thus based on the one-wayness of the multivariate polynomial maps. In the same vein, QUAD [4] is a stream cipher (symmetric key cryptography) whose security is guaranteed by the one-wayness of the multivariate polynomial maps.

The one-wayness of multivariate polynomial maps resides in the difficulty to find solutions of a system of multivariate polynomial equations (MP problem). In particular, if the multivariate polynomials involved in a MP problem consist only of quadratic polynomials, the problem is called MQ problem:

$$\begin{aligned}
f_1(x_1,\ldots,x_n) &= \sum_{1\le i\le j\le n} a_{ij}^{(1)} x_i x_j + \sum_{1\le i\le n} b_i^{(1)} x_i + c^{(1)} = d_1,\\
f_2(x_1,\ldots,x_n) &= \sum_{1\le i\le j\le n} a_{ij}^{(2)} x_i x_j + \sum_{1\le i\le n} b_i^{(2)} x_i + c^{(2)} = d_2,\\
&\vdots\\
f_m(x_1,\ldots,x_n) &= \sum_{1\le i\le j\le n} a_{ij}^{(m)} x_i x_j + \sum_{1\le i\le n} b_i^{(m)} x_i + c^{(m)} = d_m,
\end{aligned}$$

* This result was presented at the NIST Workshop on Cybersecurity in a Post-Quantum World held in Washington, D.C. on April 2-3, 2015.

where m is the number of equations, n the number of variables, and $a_{ij}^{(1)}, b_i^{(1)}, c^{(1)}, \ldots$ are all elements in a finite field F.

Since many schemes in MPKC make use only of quadratic polynomials, the analysis for solving MQ problems is important. In this paper, we present a sequence of MQ problems of different level of difficulty, which we propose as a world wide challenge. The construction of these MQ problems is based both on theoretical and on practical considerations.

MPKC can be used both in encryption schemes and in signature schemes. The original idea of MPKC was presented by Matsumoto and Imai [26], and their scheme is commonly referred to as the *MI scheme*. After MI scheme was proposed, several encryption systems were proposed such as HFE [28] and ℓ-IC [17]. Unfortunately, most of them including MI, HFE and ℓ-IC were broken after several security analyses [27, 24, 22]. Nonetheless, recently, two new encryption schemes have been proposed. First, the "simple matrix" scheme (ABC scheme) [32] which was presented at PQCrypto 2013 is an encryption scheme using matrix operations whose components consist of elements in a finite field of small size. An enhanced extension of this scheme was proposed at PQCrypto 2014 [16]. Second, the ZHFE scheme [30], also proposed at PQCrypto 2014, is an enhancement of the HFE scheme.

On the other hand, UOV [23] is a signature scheme using polynomials with a distinction on the variables into two kinds: oil variables and vinegar variables. Rainbow [15] is the "multilayered version" scheme of UOV. The framework of Rainbow, using (commutative) polynomial rings, has been extended to non-commutative rings. The security of this scheme was analyzed in [35]. The structure of the associated MQ problems for encryption schemes and signature schemes are substantially different. For encryption schemes, the associated MQ problem verifies $m \geq n$ (overdetermined), while for signature schemes, the associated MQ problem has $m \leq n$ (underdetermined). Therefore, we must prepare difficult problems of two kinds *e.g.* when $m \geq n$ and when $m \leq n$. In addition, we distinguish finite fields F of characteristic 2 and of odd characteristic, regarding that the approach to solve the MQ problems is quite different.

On the other hand, the specificity of MQ problem over $GF(2)$ has attracted the attention of several researchers in cryptography. Concerning signatures, in the extended version of [28], Patarin introduced two HFE challenges (coming with a prize of US $ 500 for attacking any of them). HFE challenges include MQ problems over $GF(2)$ saw many researchers trying to solve the problems [13]. Moreover, the solving technique of MQ problem is also applied to other cryptography. The technique is used to the security analysis of the block cipher [12]. Therefore, our instances of MQ problem include those over $GF(2)$.

When creating challenge, one of the most important matter is to guarantee the fairness of challenging problem. We mean here that the problem is evenly difficult for anybody, including the creator himself. ECC challenge [18] and Lattice challenge [25] have been built in this way. Indeed, instances of elliptic curve discrete logarithm problems and short vector problems can be created without knowing the solution. However, this is not the case for the RSA challenge [31]. Generating a composite number without knowing its factors is indeed difficult. Therefore, RSA challenge may become an unfair contest. It is also difficult to create MQ problem without knowing a solution in advance. However, we want to create MQ problem whose fairness is guaranteed. In order to achieve this, we considered two strategies. One is the use of systems of equations with completely random coefficients. Generally, these systems may not have solutions. However, the underdetermined systems have at least one solution with high probability. Another is a construction from a random solution. For the overdetermined systems, we use this method.

Two fundamental tools for solving MP problem are Gröbner bases and XL. It is known that the "degree of regularity" which is an invariant of a MQ problem is deeply related to the cost for computing a Gröbner basis corresponding to the MQ problem [2]. To determine appropriate parameters for MPKC's schemes, it is necessary to assess the practical difficulty of this problem as

precisely as possible. We have experimented solving several MQ problems with small parameters, and by combining the theoretical complexity bounds involving the degree of regularity, we have extrapolated the results of these experiments to create hard instances of MQ problems.

2 Fundamental Structure of MPKC

In both cases of encryption and of signature under MPKC, a multivariate quadratic polynomial map whose inverse map can be computed easily is required. Such a polynomial map is called a *central map*. Given a central map $G : K^n \to K^m$, a multivariate quadratic polynomial map $F : K^n \to K^m$ of the form $F = L \circ G \circ R$ can be constructed by where L and R are affine transformations on K^m and K^n, respectively. For a person who does not know the central map G, nor the two affine transformations L, R, the map F must look like a multivariate quadratic polynomial map chosen randomly. If so, F plays the role of a trapdoor one-way function, and thus would be the public key. The private key would consist of the central map G and of the affine transformations L and R. Hereafter in this section, we review in more details MPKC schemes, with the distinction encryption/signature.

2.1 Encryption Case

From the feature of an encryption scheme, G and F both must be (almost) injective. This fact imposes that $m \geq n$. For example ABC scheme and ZHFE scheme both use parameters such that $m = 2n$. Encryption and decryption are described as follows:
Encryption A plain text M is selected from K^n. An encryptor computes $C = F(M) \in K^m$. This is the associated cipher text.
Decryption The decryptor computes $E_1 = L^{-1}(C)$, $E_2 = G^{-1}(E_1)$, $E = R^{-1}(E_2)$ in this order. Then E coincides with M.

2.2 Signature Case

From the feature of a signature scheme, G and F both must be surjective. This fact imposes that $m \leq n$. UOV and Rainbow satisfy this property. UOV often uses parameters such that $n = 2m$, justified by security reasons. For Rainbow, the recommended parameters are estimated in [29] and $n \approx 1.5m$ is considered to be suitable for Rainbow. In a signature scheme, signature generation and verification are performed as follows:
Signature generation A message M is selected from K^m. The signer computes $S_1 = L^{-1}(M)$, $S_2 = G^{-1}(S_1)$, $S = R^{-1}(S_2)$ in this order. Then S is the associated signature.
Verification A verifier computes $F(S) \in K^m$, and checks if $M = F(S)$, in which case The signature is accepted.

3 General Attack against MQ Problem

Let $\mathbb{F}_q$ denote the finite field of order q and $\mathbb{F}_q[x_1, \ldots, x_n]$ the polynomial ring over $\mathbb{F}_q$ with n variables. For any $\mathbf{f} = (f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)) \in \mathbb{F}_q[x_1, \ldots, x_n]^m$, *MP problem* indicates the following computational hard problem:

Question: find a common zero $\mathbf{x_0} \in \mathbb{F}_q^n$ of the polynomials $f_1, \ldots, f_m$.

If the degree of all $f_1, \ldots, f_m$ are equal to 2, the corresponding MP problem is called *MQ problem*. A fundamental tool to solve MQ problem are the Gröbner bases. The historical method for computing Gröbner bases was introduced by Buchberger [6, 7]. Faugère made major improvements upon Buchberger's algorithm with the introduction of F_4 and F_5 [19, 20], which are often as to today considered the best algorithms for Gröbner bases computation.

3.1 Complexity of F_5 Algorithm

Definition 1. *Let $(h_1, \ldots, h_m) \in \mathbb{F}_q[x_1, \ldots, x_n]^m$ be homogeneous polynomials. The degree of regularity of a homogeneous ideal $\mathcal{I} = \langle h_1, \ldots, h_m \rangle$ is defined by*

$$d_{reg} = \min \left\{ d \geq 0 \;\middle|\; \dim_{\mathbb{F}_q}(\{f \in \mathcal{I}, \deg(f) = d\}) = \binom{n+d-1}{d} \right\}.$$

For non-homogeneous polynomials $(f_1, \ldots, f_m) \in \mathbb{F}_q[x_1, \ldots, x_n]^m$, the degree of regularity is defined by that of the ideal $\langle f_1^h, \ldots, f_m^h \rangle$, where f_i^h is the homogeneous part of f_i of highest degree. Note that in this last case, it is not an invariant of the ideal, but is attached to the polynomial system.

For a MQ problem with zero dimensional solution variety, the complexity of F_5 algorithm is given by the following.

Proposition 1. *[2] The complexity of computing a Gröbner basis of a zero-dimensional system of m equations in n variables with F_5 is:*

$$\mathcal{O}\left(\left(m \cdot \binom{n+d_{reg}}{d_{reg}}\right)^{\omega}\right)$$

where d_{reg} is the degree of regularity of the system and $2 \leq \omega \leq 3$ is the linear algebra constant.

Recall that random *underdetermined* systems are regular. The concept of semi-regularity was introduced to formalize "random systems", in the case of overdetermined systems. (though this fact is not proven in general, the *Fröberg's conjecture* has been widely observed in practice).

Definition 2. *Let $(h_1, \ldots, h_m) \in \mathbb{F}_q[x_1, \ldots, x_n]^m$ be homogeneous polynomials of respective degree $d_1, \ldots, d_m$. This sequence is semi-regular if*

- $\langle h_1, \ldots, h_m \rangle \neq \mathbb{F}_q[x_1, \ldots, x_n]$,
- *for all $1 \leq i \leq m$ and $g \in \mathbb{F}_q[x_1, \ldots, x_n]$,*

$$\deg(g \cdot h_i) < d_{reg} \text{ and } g \cdot h_i \in \langle h_1, \ldots, h_{i-1} \rangle \Rightarrow g \in \langle h_1, \ldots, h_{i-1} \rangle.$$

For a general system $(f_1, \ldots, f_m) \in \mathbb{F}_q[x_1, \ldots, x_n]^m$, this sequence is semi-regular *if the sequence $(f_1^h, \ldots, f_m^h)$ is, where f_i^h is the homogeneous part of f_i of highest degree.*

The degree of regularity of semi-regular sequences can be computed explicitly.

Proposition 2. *The degree of regularity of a semi-regular sequence $h_1, \ldots, h_m$ of respective degree $d_1, \ldots, d_m$ is given by the first non-positive coefficient of*

$$\sum_{k \geq 0} c_k z^k = \frac{\prod_{i=1}^{m}(1 - z^{d_i})}{(1-z)^n}.$$

3.2 Complexity of Hybrid Approach

Bettale et al. [2] proposed an attack against general MP problems defined over finite field of medium size from 2^2 to 2^{24} elements, which was called *hybrid approach*. This technique mixes exhaustive search and Gröbner bases computation. The raw idea was previously introduced in the context of XL solvers in [11].

Concerning systems of dimension zero defined over a finite field of medium size, the hybrid approach consists in choosing k variables, evaluate them at randomly chosen values, and solve this system in $n-k$ variables. The choice of k is delicate, but when making some reasonable assumptions (see Hypothesis 1 below), the authors of [2] succeed to provide a theoretical optimal choice of k depending of the data of the system. This outcome allows them to apply this hybrid approach to forge signatures based on a MPKC scheme (namely TMRS and UOV signatures, see [11, Section 4.1 & 4.2]) by solving underdetermined systems, where traditional solving approaches failed.

Despite we have not put into practice this approach in the experiments, we have followed the same strategy to solve underdetermined systems (Type IV,V,VI): evaluation of $m-n$ variables to reduce to a zero-dimensional systems. To solve the zero-dimensional system, we have used a standard approach rather than the hybrid, which we plan to do in the future.

Proposition 3. *For* $\mathbf{f} = (f_1(x_1,\ldots,x_n),\ldots,f_m(x_1,\ldots,x_n)) \in \mathbb{F}_q[x_1,\ldots,x_n]^m$*, let* $d_{reg}(k)$ *be the maximum degree of regularity of all the systems:*

$$\left\{\{f_1(x_1,\ldots,x_{n-k},v_1,\ldots,v_k),\ldots,f_m(x_1,\ldots,x_{n-k},v_1,\ldots,v_k)\} \mid (v_1,\ldots,v_k) \in \mathbb{F}_q^k\right\}$$

If the system is zero-dimensional, the complexity of the hybrid approach is bounded from above by

$$\mathcal{O}\left(\min_{0\le k\le n}\left\{q^k\cdot\left(m^\omega\cdot\binom{n-k+d_{reg}(k)-1}{d_{reg}(k)}^\omega+(n-k)d^{(n-k)\omega}\right)\right\}\right)$$

where $2\le\omega\le 3$.

The optimal choice of k is usually small from 1 to 4 or 5 for common MQ problems. If the base field is $GF(2)$, it is better to add the field equations to the system than to resort to the hybrid approach.

In order to grasp the asymptotic behavior of the hybrid approach, we assume a regularity condition set in [2].

Hypothesis 1 Let $\{f_1,\ldots,f_m\}\subset\mathbb{F}_q[x_1,\ldots,x_n]$ be polynomials of respective degrees $d_1\ge\cdots\ge d_m$. Let $\beta_{\min}$, $0<\beta_{\min}<1$ be a value that will be specified later. Then, for any k, $0\le k\le\lceil\beta_{\min}n\rceil$, and for each vector $(v_1,\ldots,v_m)\in\mathbb{F}_q^k$, the system,

$$\left\{\{f_1(x_1,\ldots,x_{n-k},v_1,\ldots,v_k),\ldots,f_m(x_1,\ldots,x_{n-k},v_1,\ldots,v_k)\} \mid (v_1,\ldots,v_k) \in \mathbb{F}_q^k\right\}$$

is semi-regular for n large enough.

4 Construction of the Challenge

We explain how to create MQ problems. The parameters that need to be set for a MQ problem are the size of base field q, the number of variables n and the number of equations m. As for base fields, we treat $GF(2)$ as a special case. Otherwise we consider $GF(31)$ and $GF(2^8)$ because the situation changes according to whether the characteristic of the base field is two or not. These two fields are often used as base fields in many papers [9, 29].

For most encryption schemes, overdetermined system (*i.e.* $m\ge n$) are used because the multivariate polynomial function appearing in the MQ problem underlying the schemes must be injective. For example, the ABC scheme, ZHFE scheme and the QUAD cipher have all been set to $m=2n$. As for signature schemes, underdetermined systems, *i.e.* which verify $m\le n$,

are used because the associated multivariate polynomial functions have to be surjective. Since Rainbow is a signature scheme which enhances UOV, we treat Rainbow as a representative of signature scheme in MPKC. In the case of Rainbow with 2 layers, number of polynomials m and of variables n are often set to $n \approx 1.5m$. Therefore, in our challenge, we set problems of six types, described in Table 1

Table 1. Types of MQ problem

Type	(m, n)	base field	target
I	$m = 2n$	$GF(2)$	encryption
II	$m = 2n$	$GF(2^8)$	encryption
III	$m = 2n$	$GF(31)$	encryption
IV	$n \approx 1.5m$	$GF(2)$	signature
V	$n \approx 1.5m$	$GF(2^8)$	signature
VI	$n \approx 1.5m$	$GF(31)$	signature

We consider two construction methods of MQ problem. One is for encryption and another is for signature. For encryption, $m \geq n$, however, for every system $m > n$, the probability is about $\frac{1}{q^{m-n}}$ to have a solution, and if we set $m = n$, it is the case which had the same time complexity with the signature case. Therefore, we construct the system corresponding to encryption scheme with a random solution by adjusting the constant coefficients. For the signature case, we construct the system with all random coefficients.

Algorithm 1: Construction of MQ problem for type I, II, III

Step 1 Fix parameters n and $m = 2n$ with base field over $F = GF(2), GF(2^8)$ or $GF(31)$.
Step 2 Select randomly a vector x_0 in F^n.
Step 3 Select randomly $a_{ij}^{(k)}, b_i^{(k)}$ for all i, j, k.
Step 4 Compute $c^{(k)}$ such that the associated system of equations has a solution x_0.

Algorithm 2: Construction of MQ problem for type IV, V, VI

Step 1 Fix parameters m and $n = 1.5m$ with base field over $F = GF(2), GF(2^8)$ or $GF(31)$.
Step 2 Select randomly $a_{ij}^{(k)}, b_i^{(k)}, c^{(k)}$ for all i, j, k.

5 Experiments

In this section we will present our experimental results of Type I, Type II, Type III, Type IV, Type V and Type VI systems. In order to get a general analysis to the system, we didn't apply any specific technique in solving system and used plain attack only. The experiments were all conducted on a CPU with four 6-cores Intel® Xeon® CPU E5-4617, running at 2.9GHz with an Intel® smart cache of 15MB. The Operating System was Linux Mint 15 Olivia with kernel version GNU/Linux 3.8.0-19-generic x86_64 and 1TB memory. The programming platform was Magma V2.19-9 in its 64-bit version. We provide average results on 10 experiments. The time unit is the second, and the memory unit is a MB. All algorithms were implemented in Magma, and we used the *Variety* function of Magma to compute the solutions with Gröbner bases, which has no significant time difference from *GroebnerBasis* function. All the parameters for the experiments

Table 2. Experimental results of Type I, Type II and Type III

n	m	Type I			Type II			Type III		
		Deg_{reg}	time	memory	Deg_{reg}	time	memory	Deg_{reg}	time	memory
7	14	3	0.001	32.09	3	0.442	32.09	3	0.414	32.09
8	16	3	0.001	32.09	3	0.25	32.09	3	0.167	32.09
9	18	4	0.003	32.09	4	0.207	32.09	4	0.23	32.09
10	20	4	0.008	32.09	4	0.295	32.09	4	0.37	32.09
11	22	4	0.013	32.09	4	0.432	32.09	4	0.314	32.09
12	24	4	0.021	32.09	4	0.24	32.09	4	0.261	32.09
13	26	4	0.041	32.09	4	0.47	32.09	4	0.582	32.09
14	28	4	0.08625	32.09	4	0.559	32.09	4	0.573	32.09
15	30	4	0.163	32.09	4	0.864	32.09	4	1.319	32.09
16	32	5	0.393	64.12	5	0.864	32.09	5	1.319	32.09
17	34	5	0.821	64.12	5	7.427	128.19	5	21.486	96.16
18	36	5	1.565	96.16	5	16.627	192.25	5	53.836	128.19
19	38	5	3.426	128.19	5	36.796	274.66	5	122.647	192.25
20	40	5	6.715	192.25	5	74.733	440.25	5	254.797	320.38
21	42	5	14.101	259.513	5	161.195	649.78	5	543.629	512.56
22	44	5	34.463	394.049	5	507.531	979.34	5	1717.623	809.78
23	46	5	58.006	704.75	5	967.727	1656.528	5	3542.895	1240.798
24	48	6	268.445	4397.994	6	9268.363	10681.12			
25	50	6	658.157	7724.88						
26	52	6	1437.111	13043.162						
27	54	6	2882.882	27617.278						
28	56	6	6084.231	34366.371						
29	58	6	12521.942	48814.859						

Table 3. Experimental results of Type IV, Type V and Type VI

n	m	Type IV			Type V			Type VI		
		Deg_{reg}	time	memory	Deg_{reg}	time	memory	Deg_{reg}	time	memory
11	7	8	1.261	32.09	9	2.597	32.09	9	1.981	32.09
12	8	9.3	10.122	32.09	10	30.318	32.09	9.9	18.502	32.09
14	9	10.1	127.182	32.09	11	337.327	64.12	11	377.944	64.12
15	10	11.3	1449.08	94.84	12	3446.797	136.69	12	5075.393	136.69
17	11	12.5	28786.837	292.751						

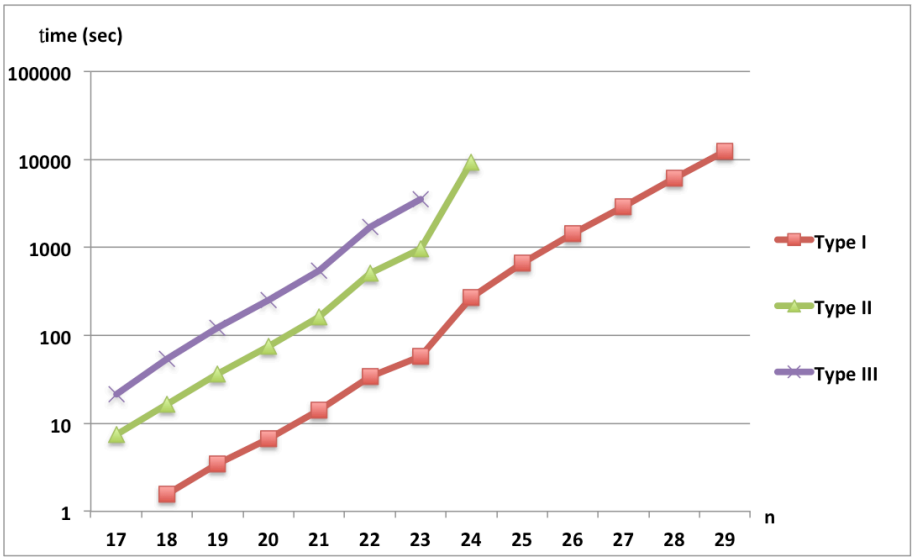

Fig. 1. Time comparison for Type I, II, and III

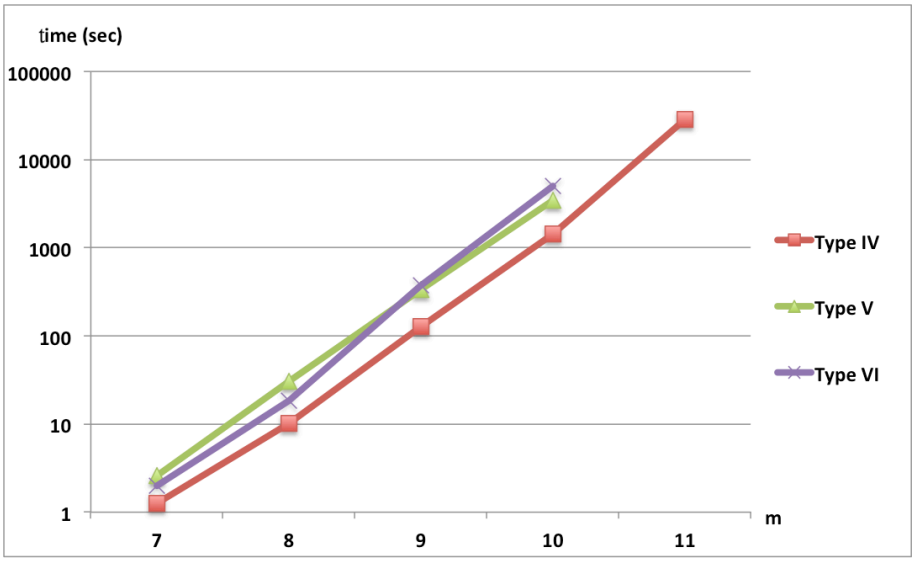

Fig. 2. Time comparison for Type IV, V and VI

applied here were only toy example, and the total time of the experiments would not exceed one week.

For every experiment, we generated a random multivariate quadratic system with coefficients in random uniform distribution with specific format. The input format is described in the Appendix. Then we read the input file and solve the system using *Variety()* function of Magma.

Table 2 shows the experimental results of Type I, Type II and Type III. Deg_{reg} represents the degree of regularity, which indicates the largest degree appeared during the process of computing a Gröbner basis. In this case, we simulate an encryption scheme without knowing any structure of it. Since in this case the system is overdetermined, we adjust the constant coefficients to make sure that the system has a solution. Actually for an overdetermined system, even if $m = n + 1$, the probability to have a solution is quite small. Let the number of variables to be n, the number of equations to be m, the size of the coefficient field to be q, then the probability of an overdetermined system to have a solution is roughly $\frac{1}{q^{m-n}}$. For systems of Type I, defined over the binary field, we add the field equations to the system.

Table 3 shows the experimental results of Type IV, Type V and Type VI. In this case, we simulate an signature scheme without knowing any structure of it. Note that in this case, since $n = 1.5m$, there are more free variables in the MQ system than the equations and thus the system is underdetermined. Generally when we solve such system, we assign random values to the free variables and solve the remained determined system. We use this technique here. Hence, the system may have the same time complexity and degree of regularity with that of the system $n = m$. For systems of Type IV, defined over the binary field, we add the field equations to the system.

Fig. 1 and Fig. 2 show the time comparison for the encryption case and the signature case respectively. Note that the x-axis indicates the number of variables n for systems of type I, II and III, whereas it indicates the number of equations m for systems of type IV, V and VI.

6 The Challenge

In Section 4, we explained the method for constructing the MQ challenge, which was used in Section 5 to construct toy examples of MQ problems. The timings obtained on these toy examples serve an experimental basis to determine parameters yielding harder instances of MQ systems, and which would require at least a month to be solved. More precisely, at least one month when equipped with a similar computational environment described in the beginning of Section 5, and using "plain" techniques (but possibly adapted to the different six Types of systems). This is because the systems are essentially random.

This challenge does not include MQ systems which are *sparse*. However, sparse systems form an important class of MQ problems, since it provides more efficiency for encryption and verification of MPKC. We plan to include this kind of challenge in the future.

In creating MQ challenges, we choose starting values of parameters m, n such that the time required for solving systems with the same parameters (extrapolation) exceeds a month.

In the boolean case, where systems and solutions are taken over the field $GF(2)$, exhaustive search and variants are the most efficient *practical* solvers (as for theoretical complexity, the improved hybrid approach in [3] gives a better worst case bound for some instances). A software library libFES [36] dedicated to solving system of polynomial equations over $GF(2)$ by exhaustive search has been created. In the homepage of this project [36], comparative benchmarks are provided between libFES and other solvers including Gröbner bases on Magma. The type of systems that these benchmarks treat are square, *i.e.* when $m = n$. Since in Type I case $m = 2n$, the exhaustive search for Type I requires twice the time of the exhaustive search against MQ problem of n equations and n variables. In this way, it is possible to estimate the time of the

Table 4. Minimum of m and n

Type	I	II	III	IV	V	VI
(m, n)	$(110, 55)$	$(70, 35)$	$(68, 34)$	$(55, 82)$	$(16, 24)$	$(16, 24)$

exhaustive search by using the result of the benchmarks displayed in the homepage of libFES. According to this estimation, we have that when $n = 55$ and $m = 110$ in Type I case, the exhaustive search takes about a month. On the other hand, in our experiment, when $n = 35$ and $m = 70$, the estimation given by Gröbner basis computation is also about a month. Comparing our experiments to the estimation of libFES, leads to the following conclusion: for a fixed couple (m, n), the of Gröbner basis computation time becomes about $1,000,000$ times that of libFES exhaustive search. In Type IV case, parameters m, n are set such that $n = 1.5m$. If we substitute random values to a number of variables equal to $1.5m - m = 0.5m$, this MQ problem of Type IV can be reduced to MQ problem of square type. By using the result of libFES, we can estimate that when $n = 82$ and $m = 55$ in Type IV case, the exhaustive search takes about a month.

Next consider the case of base field equal to $GF(2^8)$ and to $GF(31)$. For m, n of small sizes, the XL method is effective against MQ problems, as well as is the Hybrid approach. The paper [10] presents experimental results of the XL method. We make use of these results and of our experiments of type II, III, V and VI to estimate the starting parameters of MQ challenges. If we try to solve MQ problem using the Hybrid approach, it is necessary to estimate a suitable number of variables for substitution. According to theoretical computation of the paper [2], in practice this number is rarely larger than two. Therefore we choose (m, n) as starting parameter of MQ challenges such that the time for solving the MQ problem with $(m - 2, n - 3)$ using Magma F4 algorithm is over a month. From the experiments of Type II and III, the gradient of the time of attack with respect to n is about 2.31. For Type V and VI, the gradient is about 10. Using these gradients, we estimated the parameter m, n for which the time of attack of the MQ problem is over a month.

In case of Type I, II and III systems, the difficulty is graduated by the number of variables: between two consecutive problems, the harder has one more variable than the easier one. Therefore, according to Table 2 to solve a new challenge, it would take twice the time required to solve the previous one. On the other hand, in case of Type IV, V and VI systems, the difficulty is graduated by the number of equations: between two consecutive problems, the harder has one more equation than the easier one. Therefore, according to Table 3 the difficulty in term of computation time increases by a factor 10 between two such problems.

MQ Challenge

For quadratic polynomials f_i $(i = 1, 2, \ldots, m)$ of n variables over a finite field F, consider the following polynomial system:

$$S : \begin{cases} f_1(x_1, \ldots, x_n) = 0 \\ f_2(x_1, \ldots, x_n) = 0 \\ \qquad\qquad \vdots \\ f_m(x_1, \ldots, x_n) = 0 \end{cases}$$

Here, the system has one of parameters of the following six types:

Type I: $m = 2n$, $F = GF(2)$, Type IV: $n \approx 1.5m$, $F = GF(2)$,
Type II: $m = 2n$, $F = GF(2^8)$, Type V: $n \approx 1.5m$, $F = GF(2^8)$,
Type III: $m = 2n$, $F = GF(31)$, Type VI: $n \approx 1.5m$, $F = GF(31)$.

The goal is to find a solution $\mathbf{v} = (v_1, \ldots, v_n) \in F^n$ of the system S.
The challenge is hosted at `https://www.mqchallenge.org/`

Acknowledgments

We should like to thank Kouichiro Akiyama, Johannes Buchmann, Chen-Mou Cheng, Jintai Ding, Ryo Fujita, Keisuke Hakuta, Ludovic Perret, Albrecht Petzoldt, Shigeo Tsujii, and Bo-Yin Yang for their helpful comments and suggestions.

This work was supported by "Strategic Information and Communications R&D Promotion Programme (SCOPE), no. 0159-0091", Ministry of Internal Affairs and Communications, Japan.

References

1. Bernstein D.J., Buchmann J. and Dahmen E., "Post Quantum Cryptography", Springer, Heidelberg 2009.
2. Bettale L., Faugère J.C. and Perret L., "Hybrid Approach for Solving Multivariate Systems over Finite Fields", Journal of Mathematical Cryptology, vol. 2, pp. 1–22, 2008.
3. Bardet M., Faugère J.-C., Salvy B. and Spaenlehauer P.-J. On the complexity of solving quadratic boolean systems. *Journal of Complexity*, vol. 29(1), pp. 53–75, 2013.
4. Berbain C. , Gilbert H., and Patarin J., "QUAD: A Practical Stream Cipher with Provable Security", Eurocrypt 2006, Springer LNCS vol. 4004, pp. 109-128, 2006.
5. Bouillaguet C., Chen H-C., Cheng, C.-M., Chou T., Niederhagen R., Shamir A. and Yang B.-Y. Fast Exhaustive Search for Polynomial Systems in $\mathbb{F}_2$. CHES 2010, LNCS vol. 6225, pp. 203–218, 2010.
6. Buchberger B., "Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal", Ph.D. thesis, University of Innsbruch, 1965.
7. Buchberger B., Collins G.E., Loos R.G.K (Edts), in cooperation with Albrecht R. "Computer Algebra Symbolic and Algebraic Computation", 296pages, (2nd edition) Springer-Verlag (Wien New-York), 1982-83.
8. Braeken A., Wolf C. and Preneel B., "A Study of the Security of Unbalanced Oil and Vinegar Signature Schemes", CT-RSA 2005, Springer LNCS vol. 3375, pp. 29–43, 2005.
9. Chen A. I.-T., Chen M.-S., Chen T.-R., Cheng C.-M., Ding J., Kuo E. L.-H., Lee F. Y.-S., and Yang B.-Y., "SSE Implementation of Multivariate PKCs on Modern x86 CPUs", CHES 2009, Springer LNCS vol. 5747, pp. 33–48, 2009.
10. Cheng C.-M., Chou T., Niederhagen R., and Yang B.-Y., "Solving Quadratic Equations with XL on Parallel Architectures", CHES 2012, Springer LNCS vol. 7428, pp. 356–373, 2012.
11. Courtois N., Klimov A., Patarin J., Shamir A., "Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations", Eurocrypt 2000, Springer LNCS, vol. 1807, pp. 392–407, 2000.
12. Courtois N. and Pieprzyk J. "Cryptanalysis of Block Ciphers with Overdefined Systems of Equations", Asiacrypt 2002, Springer LNCS vol. 2501, pp. 267–287, 2002.
13. Courtois N., "The security of Hidden Field Equations (HFE)", CT-RSA 2001, LNCS Springer vol. 2020, pp. 266–281, 2001.
14. Ding J., Gower J. E. and Schmidt D. S., "Multivariate Public Key Cryptosystems", Advances in Information Security 25, Springer, 2006.
15. Ding J. and Schmidt D., "Rainbow, a New Multivariable Polynomial Signature Scheme", ACNS 2005, Springer LNCS vol. 3531, pp. 164–175, 2005.
16. Ding J., Petzoldt A. and Wang L-C., "The Cubic Simple Matrix Encryption Scheme", PQCrypto 2014, Springer LNCS vol. 8772, pp. 76–87, 2014.
17. Ding J., Wolf C., and Yang B.-Y., "ℓ-Invertible Cycles for Multivariate Quadratic (MQ) Public key Cryptography", PKC 2007, Springer LNCS vol. 4450, pp. 266-281, 2007.
18. The Certicom ECC Challenge: `https://www.certicom.com/index.php/the-certicom-ecc-challenge`
19. Faugère J.C., "A New Efficient Algorithm for Computing Groebner Bases (F4)", Journal of Pure and Applied Algebra, vol. 139, pp. 61–88, 1999.
20. Faugère J.C., "A New Efficient Algorithm for Computing Groebner Bases (F5)", In Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, ISSAC 2002, pp. 75–83, 2002.

21. Faugère J.C., "On the Security of UOV", In Proceedings of First International Conference on Symbolic Computation and Cryptography, SCC 2008, pp. 103–109, 2008.
22. Fouque P.-A., Macario-Rat G., Perret L., and Stern J., "Total Break of the ℓ-IC Signature Scheme", PKC 2008, Springer LNCS vol. 4939, pp. 1–17, 2008.
23. Kipnis A., Patarin J. and Goubin, L., "Unbalanced Oil and Vinegar Schemes", Eurocrypt'99, Springer LNCS vol. 1592, pp. 206–222, 1999.
24. Kipnis A. and Shamir A., "Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization", CRYPTO'99, Springer LNCS vol. 1666, pp. 19–30, 1999.
25. The Lattice Challenge: `http://www.latticechallenge.org/`
26. Matsumoto T. and Imai H., "Public Quadratic Polynomial-tuples for Efficient Signature Verification and Message Encryption", Eurocrypt'88, Springer LNCS vol. 330, pp. 419–453, 1988.
27. Patarin J., "Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88", CRYPTO'95, Springer LNCS vol. 963, pp. 248–261, 1995.
28. Patarin J., "Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms", Eurocrypt'96, Springer LNCS vol. 1070, pp. 33–48, 1996.
29. Petzoldt A., Bulygin S., Buchmann J., "Selecting Parameters for the Rainbow Signature Scheme", PQCrypto 2010, Springer LNCS vol. 6061, pp. 218–240, 2010.
30. Porras J., Baena J., and Ding J., "ZHFE, a New Multivariate Public Key Encryption Schem", PQCrypto 2014, Springer LNCS vol. 8772, pp. 229–245, 2014.
31. The RSA Factoring Challenge: `http://www.emc.com/emc-plus/rsa-labs/historical/the-rsa-factoring-challenge.htm`
32. Tao C., Diene A., Tang S., Ding J., "Simple Matrix Scheme for Encryption", PQCrypto 2013, Springer LNCS vol. 7932, pp. 231–242, 2013.
33. Wolf C., "Introduction to Multivariate Quadratic Public Key Systems and their Applications", In Proceedings of YACC 2006 - Yet Another Conference on Cryptography, Porquerolles, France, pp. 44–55, 2006.
34. Yang B.-Y. and Chen J.-M., "All in the XL Family, Theory and Practice", ICISC 2004, Springer LNCS, vol. 3506, pp. 67–86, 2005.
35. Yasuda T., Takagi T., Sakurai K., "Security of Multivariate Signature Scheme Using Non-commutative Rings", IEICE Transactions, vol. 97-A(1) pp. 245–252, 2014.
36. Homepage of the libFES library, `http://www.lifl.fr/~bouillag/fes/`.

A Input format of the MQ challenge

A.1 $GF(2)$

```
Galois Field : GF(2)
Number of variables (n) : 4
Number of polynomials (m) : 8
Order : graded reverse lex order

**********************
1 1 0 0 1 1 0 0 0 0 1 0 0 1 0 ;
0 0 0 0 1 1 0 1 0 1 0 1 0 1 1 ;
1 0 1 1 0 1 1 0 1 0 0 0 1 1 1 ;
0 0 0 1 0 0 0 0 1 0 1 1 1 0 0 ;
0 0 0 1 0 0 0 0 0 0 1 1 0 1 1 ;
0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 ;
1 0 1 0 0 1 1 1 1 0 1 0 1 1 1 ;
0 1 0 0 1 0 1 0 0 0 0 0 0 0 1 ;
```

The text box above is an example of a MQ challenge system over $GF(2)$. As we can see in the example, are specified the coefficient field, the number of variables, the number of equations in the system, and the monomial order chosen for the MQ system. Every line ends up with the ; symbol. Data coming after the ********** line in he input file rerepesent polynomials. These data are made up of the coefficients of the polynomials in the monomial basis, with respect to the monomial order indicated. For the graded reverse lexicographic order, let x_1, x_2, x_3 and x_4 to be the four variables so that $x_1 > x_2 > x_3 > x_4$. The monomials are then ordered as follows: $x_1^2 > x_1x_2 > x_1x_3 > x_1x_4 > x_2^2 > x_2x_3 > x_2x_4 > x_3^2 > x_3x_4 > x_4^2 > x_1 > x_2 > x_3 > x_4 > 1$. Thus, the first polynomial is $x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3^2 + x_3x_4 + x_4^2 + x_1 + x_2 + 1$ and the second polynomial is $x_1x_2 + x_1x_3 + x_2x_4 + x_3^2 + x_3 + 1$.

A.2 $GF(2^8)$

```
Galois Field : GF(2)[x] / x^8 + x^4 + x^3 + x^2 + 1
Number of variables (n) : 4
Number of polynomials (m) : 8
Order : graded reverse lex order

**********
de 8d 73 3b f0 46 88 50 ca 7b dc 9d 22 cd b2 ;
e1 f7 ac 25 ed b9 74 9b 7b d4 94 4f e6 b5 e0 ;
f2 cf c3 5d c4 cd a1 aa 20 51 85 4b dd b1 bc ;
08 4e 21 48 7e bc 7a ad de d5 0c b3 00 4f 5c ;
81 0e 98 4d 3c 38 d3 07 48 f3 5a 52 27 fc 91 ;
ee 27 47 c9 82 21 99 31 0e cd c4 b8 69 69 9e ;
7b ce 96 c5 37 6a ce 34 ca 19 73 8f 30 34 b6 ;
90 65 bc d0 02 77 6c af 1d 7f 1c 29 9c 55 60 ;
```

The text box above is an example of a MQ challenge system over $GF(2^8)$. Most of the format is similar to the example of $GF(2)$. In the case of $GF(2^8)$, the input file also displays the structure of the field by specifying the irreducible polynomial it uses to define the field extension. The hex representation of the coefficient is the polynomial representation of the $GF(2^8)$ element. For example, the first coefficient **de** indicates **1101 1110**, which refers to $x^7+x^6+x^4+x^3+x^2+x \in GF(2)[x]/x^8+x^4+x^3+x^2+1$. Similarly, the second coefficient **8d** indicates **1000 1101**, which refers to $x^7 + x^3 + x^2 + 1 \in GF(2)[x]/x^8 + x^4 + x^3 + x^2 + 1$.

A.3 $GF(31)$

```
Galois Field : GF(31)
Number of variables (n) : 4
Number of polynomials (m) : 8
Order : graded reverse lex order

********************
29 20 25 28 4 7 10 28 8 13 14 29 19 30 8 ;
24 20 3 27 25 28 30 3 23 6 23 25 3 2 18 ;
4 29 29 31 0 19 7 24 18 8 9 23 24 8 27 ;
28 4 4 4 17 16 3 25 14 2 1 6 30 8 16 ;
6 1 11 17 3 1 14 14 6 29 3 23 27 18 22 ;
25 19 7 0 1 14 28 27 6 11 13 26 29 14 24 ;
12 21 28 2 21 25 0 12 1 29 27 7 23 23 14 ;
1 28 21 15 11 30 23 7 9 26 10 29 2 0 7 ;
```

The text box above is an example of a MQ challenge system over $GF(31)$. All the format is similar to the example of $GF(2)$ and $GF(2^8)$. Every number is a coefficient in $GF(31)$.