MQQ-SIG - An Ultra-Fast and Provably CMA Resistant Digital Signature Scheme

Danilo Gligoroski, Rune Steinsmo Ødegard, Rune Erlend Jensen, Ludovic Perret, Jean-Charles Faugère, Svein Johan Knapskog, Smile Markovski

MQQ-SIG

An Ultra-Fast and Provably CMA Resistant Digital Signature Scheme

Danilo Gligoroski[1], Rune Steinsmo Ødegård[2], Rune Erlend Jensen[2],
Ludovic Perret[3], Jean-Charles Faugère[3],
Svein Johan Knapskog[2], and Smile Markovski[4]

[1] Department of Telematics,
The Norwegian University of Science and Technology (NTNU),
O.S. Bragstads plass 2E, N-7491 Trondheim, Norway
danilog@item.ntnu.no
[2] Centre for Quantifiable Quality of Service in Communication Systems, NTNU,
O.S. Bragstads plass 2E, N-7491 Trondheim, Norway
knapskog@Q2S.ntnu.no, rune.odegard@q2s.ntnu.no, runeerle@stud.ntnu.no
[3] INRIA, Paris-Rocquencourt Center, SALSA Project
UPMC Univ. Paris 06, UMR 7606, LIP6, F-75005, Paris, France
CNRS, UMR 7606, LIP6, F-75005, Paris, France
jean-charles.faugere@inria.fr, ludovic.perret@lip6.fr
[4] "Ss Cyril and Methodius" University,
Faculty of Natural Sciences and Mathematics, Institute of Informatics,
P.O. Box 162, 1000 Skopje, Macedonia
smile@ii.edu.mk

Abstract. We present MQQ-SIG, a signature scheme based on "*Multivariate Quadratic Quasigroups*". The MQQ-SIG signature scheme has a public key consisting of $\frac{n}{2}$ quadratic polynomials in n variables where $n = 160, 192, 224$ or 256. Under the assumption that solving systems of $\frac{n}{2}$ MQQ's equations in n variables is as hard as solving systems of random quadratic equations, we prove that in the random oracle model our signature scheme is CMA (Chosen-Message Attack) resistant.

From efficiency point of view, the signing and verification processes of MQQ-SIG are three orders of magnitude faster than RSA or ECDSA. Compared with other MQ signing schemes, MQQ-SIG has both advantages and disadvantages. Advantages are that it has more than three times smaller private keys (from 401 to 593 bytes), and the signing process is an order of magnitude faster than other MQ schemes. That makes it very suitable for implementation in smart cards and other embedded systems. However, MQQ-SIG has a big public key (from 125 to 512 Kb) and it is not suitable for systems where the size of the public key has to be small.

Keywords: Public Key Cryptography, Ultra-Fast Public Key Cryptography, Multivariate Quadratic Polynomials, Quasigroup String Transformations, Multivariate Quadratic Quasigroup.

L. Chen, M. Yung, and L. Zhu (Eds.): INTRUST 2011, LNCS 7222, pp. 184–203, 2012.

1 Introduction

Multivariate quadratic schemes (MQ schemes) are an active research area since their introduction more than 26 years ago in the papers of Matsumoto and Imai [25,31]. They have a lot of performance advantages over classical public key schemes based on integer factorization (RSA) and on the discrete logarithm problem in the additive group of points defined by elliptic curves over finite fields (ECC), but they have also one additional advantage: there are no known quantum algorithms that would break MQ schemes faster than generic brute force attacks.

We can say that MQ schemes can be generally divided in five types of schemes that conceptually differ in the construction of the nonlinear quadratic part of the scheme. There is a nice (but a little bit older survey from 2005) [49] that covers the first four classes of multivariate quadratic public key cryptosystems: MIA [25], STS [44,33,23], HFE [36] and UOV [28].

The fifth scheme MQQ was introduced in [21,22] in 2008. MQQ is based on the theory of quasigroups and quasigroup string transformations. Since it had interesting performance characteristics, it immediately attracted the attention of cryptographers trying to attack it. It was first successfully cryptanalysed independently by Perret [39] using Gröbner basis approach, and Mohamed et al. using MutantXL [35]. Later, improved cryptanalysis by Faugère et al. in [17] explained exactly why the MQQ systems are so easy to solve in practice.

In this paper we describe a digital signature variant of MQQ (called MQQ-SIG). To thwart previous successful attacks, we propose to use the *minus modifier*, i.e. to remove some equations of the public key. More specifically, we remove $\frac{1}{2}$ of the public equations of the original MQQ public key algorithm. We also present numerical (experimental) evidence that gives us arguments to believe that Gröbner bases approach (and having in mind that MutantXL approach is equivalent) is ineffective in solving the remaining known equations.

Thus, based on the assumption that solving $\frac{n}{2}$ quadratic MQQ's equations with n variables is as hard as solving systems of random quadratic equations, we show that in the random oracle model our signature scheme is provably CMA resistant.

The properties of MQQ-SIG digital signature scheme can be briefly summarized as:

- In the random oracle model it is provably CMA resistant under the assumption that solving $\frac{n}{2}$ MQQ's quadratic equations with n variables is as hard as solving systems of random equations;
- Its conjectured security level is at least $2^{\frac{n}{2}}$;
- The length of the signature is $2n$ bits where ($n = 160, 192, 224$ or 256);
- The size of the private key is between 401 and 593 bytes.
- The size of the public key is between 125 and 512 Kb.
- In software, its signing speed is in the range of 300–3,500 times faster than the most popular public key schemes, and 5 to 20 times faster than other multivariate quadratic schemes with equivalent security parameters;
- Its verification speed is comparable to the speed of other multivariate quadratic PKCs;

- In hardware, its signing or verification speed can be more than 10,000 times faster than the most popular public key schemes;
- In 8-bit MCUs, smart cards and RFIDs, it is hundreds or thousands times faster than the most popular public key signature schemes;

2 Preliminaries - Quasigroups and Multivariate Quadratic Quasigroups

Here we give a brief overview of quasigroups and quasigroup string transformations. A more detailed explanation can be found in [5,12,47].

Definition 1. *A quasigroup $(Q, *)$ is a groupoid satisfying the law*

$$(\forall u, v \in Q)(\exists! x, y \in Q) \quad u * x = v \ \& \ y * u = v. \tag{1}$$

This implies the cancelation laws $x*y = x*z \implies y = z$, $y*x = z*x \implies y = z$. Note also that the equations $a * x = b$, $y * a = b$ have unique solutions x, y for each $a, b \in Q$. Given a quasigroup $(Q, *)$ five so called "*parastrophes*" (or "*conjugate operations*") can be adjoint to $*$. Here, we use only two of them – denoted by $\backslash$ and $/$, – defined by

$$x * y = z \iff y = x \setminus z \iff x = z/y \tag{2}$$

Then $(Q, \backslash)$ and $(Q, /)$ are quasigroups too and the algebra $(Q, *, \backslash, /)$ satisfies the identities

$$x \setminus (x * y) = y, \quad (x * y)/y = x, \quad x * (x \setminus y) = y, \quad (x/y) * y = x \tag{3}$$

Conversely, if an algebra $(Q, *, \backslash, /)$ with three binary operations satisfies the identities (3), then $(Q, *)$, $(Q, \backslash)$, $(Q, /)$ are quasigroups and (2) holds.

In what follows we will work with finite quasigroups of order 2^d i.e. where $|Q| = 2^d$. To define a multivariate quadratic PKC for our purpose, we will use the following result.

Lemma 1 ([21,22]). *For every quasigroup $(Q, *)$ of order 2^d and for each bijection $Q \to \{0, 1 \ldots, 2^d - 1\}$ there are a uniquely determined vector valued Boolean functions $*_{vv}$ and d uniquely determined $2d$-ary Boolean functions $f_1, f_2, \ldots, f_d$ such that for each $a, b, c \in Q$ the operation $a * b = c$ is represented by*

$$*_{vv}(x_1, \ldots, x_d, y_1, \ldots, y_d) = \big(f_1(x_1, \ldots, x_d, y_1, \ldots, y_d), \ldots, f_d(x_1, \ldots, x_d, y_1, \ldots, y_d)\big). \tag{4}$$

Recall that each k-ary Boolean function $f(x_1, \ldots, x_k)$ can be represented in a unique way by its algebraic normal form (ANF), i.e., as a sum of products $\mathrm{ANF}(f) = \alpha_0 + \sum_{i=1}^{k} \alpha_i x_i + \sum_{1 \le i < j \le k} \alpha_{i,j} x_i x_j + \sum_{1 \le i < j < s \le k} \alpha_{i,j,s} x_i x_j x_s + \ldots$, where the coefficients $\alpha_0, \alpha_i, \alpha_{i,j}, \ldots$ are in the set $\{0, 1\}$ and the addition and multiplication are in the field $GF(2)$.

The ANFs of the functions f_i defined in Lemma 1 give us information about the complexity of the quasigroup $(Q, *)$ via the degrees of the Boolean functions f_i. In general, for a randomly generated quasigroup of order 2^d, $d \ge 4$, the degrees are higher than 2. Such quasigroups are not quadratic and thus are not suitable for our construction of multivariate quadratic PKC.

Definition 2. *A quasigroup $(Q, *)$ of order 2^d is called Multivariate Quadratic Quasigroup (MQQ) of type $Quad_{d-k}Lin_k$ if exactly $d-k$ of the polynomials f_i are of degree 2 (i.e., are quadratic) and k of them are of degree 1 (i.e., are linear), where $0 \leq k < d$.*

In [21,22] the authors give sufficient conditions a quasigroup to be a MQQ as well as an algorithm for finding MQQs up to the order of 2^5. That work was later extended in [10] for constructing MQQs of order 2^d for any d. The common characteristic of the MQQs produced by those two methods is that the quasigroups are bilinear. Namely, the equations (4) describing a multivariate quadratic quasigroup $(Q, *)$ can be expressed in the following form:

$$\mathbf{A_1} \cdot (y_1, \ldots, y_d)^T + \mathbf{b_1} \equiv \mathbf{A_2} \cdot (x_1, \ldots, x_d)^T + \mathbf{b_2} \tag{5}$$

where $\mathbf{A_1} = [f_{ij}]_{d \times d}$ is a $d \times d$ matrix and $\mathbf{b_1} = [u_i]_{d \times 1}$ is a $d \times 1$ vector of linear Boolean expressions of the variables $x_1, \ldots, x_d$, while $\mathbf{A_2} = [g_{ij}]_{d \times d}$ is a $d \times d$ matrix and $\mathbf{b_2} = [v_i]_{d \times 1}$ is a $d \times 1$ vector of linear Boolean expressions of the variables $y_1, \ldots, y_d$.

A Multivariate Quadratic Quasigroup (MQQ) $*$ of order 2^d used in MQQ-SIG can be described shortly by the following expression:

$$\mathbf{x} * \mathbf{y} \equiv \mathbf{B} \cdot \mathbf{U}(\mathbf{x}) \cdot \mathbf{A_2} \cdot \mathbf{y} + \mathbf{B} \cdot \mathbf{A_1} \cdot \mathbf{x} + \mathbf{c} \tag{6}$$

where $\mathbf{x} = (x_1, \ldots, x_d)$, $\mathbf{y} = (y_1, \ldots, y_d)$, the matrices $\mathbf{A_1}$, $\mathbf{A_2}$ and $\mathbf{B}$ are nonsingular of size $d \times d$ in $GF(2)$, the vector $\mathbf{c}$ is a random d-dimensional vector with elements in $GF(2)$ and all of them are generated by a uniformly random process. The matrix $\mathbf{U}(\mathbf{x})$ is an upper triangular matrix with all diagonal elements equal to 1, and the elements above the main diagonal are linear expressions of the variables of $\mathbf{x} = (x_1, \ldots, x_d)$. It is computed by the following expression:

$$\mathbf{U}(\mathbf{x}) = I + \sum_{i=1}^{d-1} \mathbf{U}_i \cdot \mathbf{A_1} \cdot \mathbf{x}, \tag{7}$$

where the matrices $\mathbf{U}_i$ have all elements 0 except the elements in the rows from $\{1, \ldots, i\}$ that are strictly above the main diagonal. Those elements can be either 0 or 1 generated by a uniformly random process.

Additionally, we require the quasigroups to satisfy the following two conditions:

$$\forall i \in \{1, \ldots, d\},\ Rank(\mathbf{B}_{f_i}) \geq 2d - 4, \tag{8a}$$

$$\exists j \in \{1, \ldots, d\},\ Rank(\mathbf{B}_{f_j}) = 2d - 2 \tag{8b}$$

where the matrices $\mathbf{B}_{f_i}$ are $2d \times 2d$ Boolean matrices defined from the expressions f_i as

$$\mathbf{B}_{f_i} = [b_{j,k}],\ b_{j,d+k} = b_{d+k,j} = 1, \text{ iff } x_j y_k \text{ is a term in } f_i. \tag{9}$$

The reasons why we need the additional conditions (8a) and (8b) will be explained in the beginning of the Section 5.

Proposition 1. *For $d = 8$, a multivariate quadratic quasigroup that satisfies the conditions (6), ..., (9) can be encoded in a unique way with 81 bytes.* □

3 Description of the MQQ-SIG Digital Signature Scheme

Our scheme can be expressed as a $(\frac{1}{2})$ truncation of a typical multivariate quadratic system:

$$\mathbf{S} \circ P' \circ \mathbf{S}' : \{0,1\}^n \to \{0,1\}^n,$$

where $\mathbf{S}' = \mathbf{S} \cdot \mathbf{x} + \mathbf{v}$ (i.e. $\mathbf{S}'$ is a bijective affine transformation), $\mathbf{S}$ is a nonsingular linear transformation, and $P' : \{0,1\}^n \to \{0,1\}^n$ is a central bijective multivariate quadratic mapping defined in Table 1. It is graphically presented in Fig. 1.

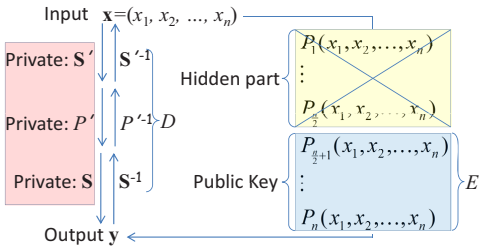

Fig. 1. A graphical presentation of our MQ "minus" scheme

The graphical presentation of the construction of the central mapping P' using the quasigroup operation $*$ is shown in Fig. 2, and its inverse P'^{-1} constructed with the parastrophe operations $\backslash$ and $/$ is shown in Fig. 3.

Fig. 2. A graphical presentation of the construction of the central bijective multivariate quadratic mapping P'

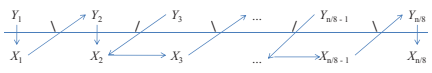

Fig. 3. A graphical presentation of the construction of the inverse central mapping P'^{-1} with parastrophe operations

The generation of the public and private key is defined in Table 2.

Let us denote by $D(\mathbf{y})$ the composition of inverse operations $\mathbf{S}^{-1}$, P'^{-1} and $\mathbf{S}'^{-1}$ on vector $\mathbf{y}$ i.e. $D(\mathbf{y}) \equiv \mathbf{S}^{-1}(P'^{-1}(\mathbf{S}'^{-1}(\mathbf{y})))$. Also, let us denote by $E(\mathbf{x})$ the mapping of a vector $\mathbf{x}$ with the public polynomials $P_i(x_1, \ldots, x_n)$ $i = 1+\frac{n}{2}, \ldots, n$. Both signing and verification for MQQ-SIG are graphically presented on Fig. 4 while the algorithmic steps for the signing procedure are presented in details in Table 3, and the verification steps in Table 4.

4 Design Rationale

4.1 Nonsingular Boolean Matrices in MQQ-SIG

The nonsingular Boolean matrices that are used in MQQ-SIG are generated in a specific way. In general, we need n^2 bits to store a randomly generated

Table 1. Definition of the central bijective multivariate quadratic mapping $P' : \{0,1\}^n \to \{0,1\}^n$

The central Bijective multivariate quadratic mapping $P'(\mathbf{x})$
Input. A vector $\mathbf{x} = (f_1, \ldots, f_n)$ of n linear Boolean functions of n variables. We implicitly suppose that a multivariate quadratic quasigroup $*$ is previously defined, and that $n = 32 \times k$, with $k \in \{5, 6, 7, 8\}$ already fixed.
Output. 8 linear expressions $P'_i(x_1, \ldots, x_n), i = 1, \ldots, 8$ and $n-8$ multivariate quadratic polynomials $P'_i(x_1, \ldots, x_n), i = 9, \ldots, n$
1. Represent a vector $\mathbf{x} = (f_1, \ldots, f_n)$ of n linear Boolean functions of n variables $x_1, \ldots, x_n$, as a string $\mathbf{x} = X_1 \ldots X_{\frac{n}{8}}$ where X_i are vectors of dimension 8; 2. Compute $\mathbf{y} = Y_1 \ldots Y_{\frac{n}{8}}$ where: $Y_1 = X_1$, $Y_{j+1} = X_j * X_{j+1}$, for even $j = 2, 4, \ldots$, and $Y_{j+1} = X_{j+1} * X_j$, for odd $j = 3, 5, \ldots$ 3. Output: $\mathbf{y}$.

Table 2. Generation of the public and the private key

Generation of the public and the private key for MQQ-SIG scheme.
Input. Integer n, where $n = 32 \times k$ and $k \in \{5, 6, 7, 8\}$.
Output. A public key $\mathbf{P}$ given by $\frac{n}{2}$ multivariate quadratic polynomials $P_i(x_1, \ldots, x_n)$, $i = 1 + \frac{n}{2}, \ldots, n$, and a private key given by two permutations σ_0^0 and σ_0^1 on $\{1, \ldots, n\}$, and 81 bytes for encoding a quasigroup $*$.
1. Generate an MQQ $*$ according to equations (6) $\ldots$ (9). 2. Generate a nonsingular $n \times n$ Boolean matrix $\mathbf{S}$ and affine transformation $\mathbf{S}'$ according to equations (10), $\ldots$, (13). 3. Compute $\mathbf{y} = \mathbf{S}(P'(\mathbf{S}'(\mathbf{x})))$, where $\mathbf{x} = (x_1, \ldots, x_n)$. 4. Output: The public key is $\mathbf{y}$ as $\frac{n}{2}$ multivariate quadratic polynomials $P_i(x_1, \ldots, x_n)$ $i = 1 + \frac{n}{2}, \ldots, n$, and the private key is the tuple $(\sigma_0^0, \sigma_0^1, *)$.

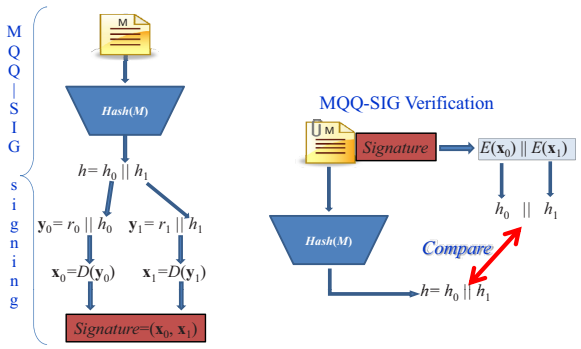

Fig. 4. A graphical presentation of the signing and verification process with MQQ-SIG

nonsingular Boolean matrix of size $n \times n$. In our case we need to store $\mathbf{S}^{-1}$ because we need it in the process of signing. With our proposed sizes for $n = 160, 192, 224, 256$, storing $\mathbf{S}^{-1}$ would require between 3.125 and 8.0 Kbytes.

The idea of reducing the size of the keys in MQ schemes by using circulant matrices has been applied previously in several works [51,46,40]. Instead of using one circulant matrix, we use two. The rationale why and how we construct the private linear (affine) transformations from them is given in what follows.

In order to compress the private information for the linear and affine transformations we define nonsingular matrices $\mathbf{S}$ by the following expression:

$$\mathbf{S}^{-1} = \bigoplus_{i=0}^{\frac{n}{16}} I_{\sigma_i^0} \oplus \bigoplus_{i=0}^{\frac{n}{16}+3} I_{\sigma_i^1}, \tag{10}$$

Table 3. Digital signing

Signing with a private key $(\sigma_0^0, \sigma_0^1, *)$
Input. A document M to be signed.
Output. A signature $\mathbf{sig} = (\mathbf{x}_0, \mathbf{x}_1)$.
1. Compute the pair $h = h_0 \| h_1 \leftarrow Hash(M)$, where $Hash()$ is the standardized hash function. Here we assume that the output of the hash function is n bits, and that h_0 and h_1 are $\frac{n}{2}$ bits long. 2. Set $\mathbf{y}_0 = r_0 \| h_0$ and $\mathbf{y}_1 = r_1 \| h_1$, where the values r_0 and r_1 are $\frac{n}{2}$-bit values chosen uniformly at random. 3. Compute $\mathbf{x}_0 = D(\mathbf{y}_0)$ and $\mathbf{x}_1 = D(\mathbf{y}_1)$. 4. The MQQ-SIG digital signature of the document M is the pair $\mathbf{sig} = (\mathbf{x}_0, \mathbf{x}_1)$.

Table 4. Digital verification

Signature verification with a public key $\mathbf{P} = \{P_i(x_1, \ldots, x_n) \mid i = 1 + \frac{n}{2}, \ldots, n\}$
Input. A document M and its signature $\mathbf{sig} = (\mathbf{x}_0, \mathbf{x}_1)$.
Output. TRUE or FALSE.
1. Compute $h = h_0 \| h_1 = Hash(M)$, where M is the signed message, and $Hash()$ is the standardized hash function. 2. Compute $\mathbf{z}_0 = E(\mathbf{x}_0)$ and $\mathbf{z}_1 = E(\mathbf{x}_1)$. 3. If $\mathbf{z}_0 = h_0$ and $\mathbf{z}_1 = h_1$ then return TRUE, else return FALSE.

where $I_{\sigma_i^0}, i = \{0, 1, 2, \ldots, \frac{n}{16}\}$ and $I_{\sigma_i^1}, i = \{0, 1, 2, \ldots, \frac{n}{16} + 1\}$ are permutation matrices of size n, the operation $\oplus$ is a "bitwise exclusive or" of the elements in the permutation matrices and permutations σ_i^0 and σ_i^1 are permutations on n elements. They are defined by the following expressions:

$$\begin{cases} \sigma_0^0 - \text{random permutation on } \{1, 2, \ldots n\}, \\ \sigma_i^0 = RotateLeft(\sigma_{i-1}^0, 8), \text{ for } i = 1, \ldots, \frac{n}{16}, \\ \sigma_0^1 - \text{random permutation on } \{1, 2, \ldots n\}, \\ \sigma_i^1 = RotateLeft(\sigma_{i-1}^1, 8), \text{ for } i = 1, \ldots, \frac{n}{16} + 1, \end{cases} \tag{11}$$

We chose the permutations σ_0^0 and σ_0^1 such that the expression (10) gives a non-singular matrix $\mathbf{S}^{-1}$ (and $\mathbf{S} = (\mathbf{S}^{-1})^{-1}$). From $\mathbf{S}$ we will obtain the affine transformation

$$\mathbf{S}'(\mathbf{x}) = \mathbf{S} \cdot \mathbf{x} + \mathbf{v}, \tag{12}$$

where the vector $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ is an n–dimensional Boolean vector defined from the values of the permutation $\sigma_0^1 = (s_1, s_2, \ldots, s_n)$ by the following expression:

$$v_i = \left(\left(\frac{\left(\left(s_{1+\lfloor \frac{i-1}{8} \rfloor} \right) \bmod 16 \right) \times 16}{2^{(8-i) \bmod 8}} \right) + \left(\frac{s_{65+\lfloor \frac{i-1}{8} \rfloor}}{2^{(8-i) \bmod 8}} \right) \right) \bmod 2. \tag{13}$$

In words: we construct the bits of the vector $\mathbf{v}$ by constructing two arrays. The first array is constructed by taking the four least significant bits of the values $s_1, \ldots, s_{\frac{n}{8}}$ and each of them is shifted by four positions to the left. The second array is just simple extraction of the values $s_{65}, \ldots, s_{65+\frac{n}{8}}$. Finally we XOR respectively those two arrays of values in order to produce the vector $\mathbf{v}$ of n bits. *Although the expression (13) looks complex, it is chosen specifically to be very fast in software and hardware.*

Proposition 2. *The linear transformation* $\mathbf{S}^{-1}$ *can be encoded in a unique way with* $2n$ *bytes.* □

The reasons why we decided to use two permutations σ_0^0 and σ_0^1 in order to define the matrix $\mathbf{S}^{-1}$ as in (10) are due to the fact that the inverse matrix of any circulant matrix is again circulant [11]. Thus, if we would use a circulant matrix $\mathbf{S}^{-1}$, its inverse $\mathbf{S}$ that is used in the production of the public key would be also circulant. From a cryptographic point of view, we wanted to avoid the circular property of $\mathbf{S}$ since its strong regularity. This strong regularity might affect the randomness of the multivariate quadratic expressions in the public key. We have made a tradeoff between the totaly non-circulant matrix $\mathbf{S}$ generated completely by a uniformly distributed random process which will cost a lot in terms of space, and the regular circulant matrices, by using two circulant matrices that are combined as it is described in the expression (10). The obtained $\mathbf{S}$ from $\mathbf{S}^{-1}$ is without the circulant regularity, and still we can store it in just $2n$ bytes.

To illustrate our technique for producing non-circulant matrices $\mathbf{S}^{-1}$ and $\mathbf{S}$ we give the following baby example with $n = 16$ and where rotations to the left are performed by 2 positions.

Let $\sigma_0^0 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 0 & 6 & 2 & 5 & 15 & 8 & 11 & 12 & 1 & 9 & 14 & 3 & 10 & 7 & 4 & 13 \end{pmatrix}$ and $\sigma_0^1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 12 & 5 & 14 & 2 & 6 & 7 & 9 & 0 & 10 & 11 & 8 & 4 & 1 & 15 & 13 & 3 \end{pmatrix}$. Since this is a baby example, we have to adopt the expression (10) for this smaller value of n. The adopted expression is: $\mathbf{S}^{-1} = \bigoplus_{i=0}^{2} I_{\sigma_i^0} \oplus \bigoplus_{i=0}^{3} I_{\sigma_i^1}$ and we get

$$\mathbf{S}^{-1} = \begin{pmatrix} 1&0&1&0&0&0&1&0&0&1&0&0&1&0&1&1 \\ 1&0&1&0&0&0&1&1&1&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&1&0&0&1&1&1&0&0&1&1 \\ 1&0&1&0&0&1&0&1&1&0&0&1&1&0&0&0 \\ 0&1&0&0&0&0&1&0&1&1&1&1&0&0&0&1 \\ 1&0&0&0&1&0&0&1&1&1&0&1&1&0&0&0 \\ 0&0&0&0&0&0&0&0&1&1&1&1&0&0&1&0 \\ 1&0&0&1&1&0&0&0&0&1&0&1&1&0&0&1 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&1&1&0 \\ 0&0&0&0&1&0&0&1&0&1&0&1&0&0&0&1 \\ 0&1&0&0&1&0&0&0&1&0&1&0&1&1&1&0 \\ 0&0&0&0&1&1&0&1&0&0&0&0&0&1&0&1 \\ 1&1&0&0&1&0&0&0&0&0&1&0&1&1&1&0 \\ 0&0&1&1&0&1&1&1&0&0&0&0&0&1&0&1 \\ 1&0&1&0&1&0&1&0&0&0&0&0&1&1&1&0 \\ 0&0&1&1&0&0&1&1&0&0&0&0&0&1&0&0 \end{pmatrix} \quad \text{and} \quad \mathbf{S} = \begin{pmatrix} 1&0&0&1&1&1&0&0&0&1&1&1&0&0&0&0 \\ 1&0&1&1&1&1&0&0&1&0&0&0&0&1&1&1 \\ 1&0&1&1&0&0&1&1&0&0&0&0&0&1&1&0 \\ 0&1&1&0&0&1&1&1&0&0&1&0&1&0&0&0 \\ 1&0&1&1&1&0&1&0&1&0&0&1&1&0&1&0 \\ 0&1&1&0&0&0&1&1&1&1&1&1&0&1&1&1 \\ 0&1&1&0&1&1&1&0&1&0&1&0&0&0&0&0 \\ 1&0&0&1&1&1&0&1&1&0&0&0&1&1&1&0 \\ 1&0&0&1&1&1&0&0&0&1&0&1&1&0&0&0 \\ 0&1&1&1&1&1&1&0&0&1&1&0&0&1&1&1 \\ 0&1&1&0&0&0&0&1&0&0&1&0&1&0&1&1 \\ 0&0&1&1&1&0&1&0&1&1&0&0&1&1&0&1 \\ 0&1&1&0&0&1&1&1&1&0&0&1&0&1&1&0 \\ 0&0&1&0&0&1&1&1&0&0&0&0&0&0&0&1 \\ 1&0&1&1&1&0&1&1&1&1&0&1&1&0&0&1 \\ 0&1&1&0&0&0&1&1&1&1&1&1&1&1&1&0 \end{pmatrix}$$

Note that $\mathbf{S}$ is not a circulant matrix.

4.2 Choosing the Order and Characteristics of Quasigroups

In the original MQQ proposal [21,22], the authors used several different multivariate quasigroups of order 2^5. That design decision was mainly done because the authors did not know how to construct MQ quasigroups of bigger order.

In the meantime, Chen et al., in [10] and Samardjiska et al., in [42] have found ways how to construct MQQs of arbitrary order 2^d. Thus, we have decided to use quasigroups of order 2^8. That decision was made in order to match the byte size of 8 bits. This enables efficient implementations of MQQ-SIG even on tiny industrial 8-bit MCUs, as well as on high end systems (PCs or workstations). The left and right parastrophes can be pre-computed each taking 64KBytes. These pre-computed parastrophes can speedup the signing phase at least 10 times, but using pre-computed parastrophes of size 2^d where $d > 8$ simply becomes too costly.

Without going into details of the different characteristics of MQQs produced by methods described in [10] and [42] we can say that for encoding MQQs as described in [42] we need 256 bytes, while for MQQs from [10] we need just 81 bytes (see Proposition 1). This is due to the fact that MQQs in [10] have bi-linear nature, while MQQs constructed in [42] are based on T-functions and generally are not bi-linear.

We have performed experiments with both types of MQQs and after removing $\frac{n}{2}$ MQQ's expressions from the public key, we have not observed any security consequences of using the bi-linear MQQs from [10]. That fact combined with the fact that the knowledge of MQQ is a part of the private key, and that the encoding of MQQs from [10] needs just 81 bytes (versus 256 bytes for MQQs from [42]), was the decisive argument in favor of MQQs defined in [10].

In our design we use affine transformation $\mathbf{S}'$ instead of the linear one $\mathbf{S}$, and we also use a non-zero vector $\mathbf{c}$ in the quasigroup construction. The reasons for this is that without $\mathbf{S}'$ our scheme would have the zeroth vector as a fixed point and the same is true for a quasigroup that has $\mathbf{c} = \mathbf{0}$. We consider that these properties are unnecessary and easily avoidable weaknesses.

5 Security Analysis of the Algorithm

In this section we will describe all the security analysis we have performed during the design of MQQ-SIG. First we want to emphasize that MQQ-SIG similarly as the original MQQ is still resistant against the well know attacks such as: Patarin's chosen plaintext attack on MIA scheme [37], the attacks with differential cryptanalysis that were proposed by Fouque, Granboulan and Stern in [19], solving the isomorphism of polynomials with one secret done by Perret and others in [38,18,9] and MinRank attacks. For the resistance against MinRank attacks we want to note that the minimal rank r of the matrices $\mathbf{B}_{f_i}$ for the nonlinear part of our scheme have to fulfil the conditions (8a, 8b), thus at least one of the ranks is 14 and all of the ranks are at least 12. Additionally, it is not known how to extend the MinRank attack to our scheme, since some equations of the public-key have been removed. In [8], it has been proved that the attack can be extended when 1 equation is removed in HFE. However, the attack can not be applied in our context when $\frac{n}{2}$ equations are removed.

We suggest the reader to see [21,22] for the arguments why MQQ-SIG is resistant against these attacks.

5.1 Experiments with Gröbner Bases

The public key encryption algorithm MQQ introduced in [21,22] was quickly shown to be weak against algebraic cryptanalysis. It was broken both by Perret [39] using Gröbner basis approach, and by Emam Mohamed et al [35] using MutantXL. Later Faugère et al [17] explained why the MQQ systems are so easy to solve in practice. To understand their results we must first introduce to concept of degree of regularity.

As explained in [17], the complexity of computing a Gröbner basis of an ideal depends on the maximum degree of the polynomials appearing during the computation. This degree, called *degree of regularity*, is the key parameter for understanding the complexity of a Gröbner basis computation [3]. Indeed, the complexity of the computation is polynomial in the degree of regularity D_{reg}, more precisely the complexity is:

$$\mathcal{O}(n^{\omega D_{\text{reg}}}), \tag{14}$$

which basically correspond to the complexity of reducing a matrix of size $\approx n^{D_{\text{reg}}}$. Here $2 < \omega \leq 3$ is the "linear algebra constant", and n the number of variables of the system. Note that D_{reg} is a function of n and also of the number of equations m. The relation between D_{reg}, n and m depends on the specific system of equations. This relation is well understood for regular (and semi-regular) systems of equations [1,2,3,4]. However, as soon as the system has some kind of structure, this degree is much more difficult to predict.

In [17], the authors showed that the degree of regularity of the original public key algorithm MQQ was bounded from above by a small constant. Having in mind the successful and very efficient way how Gröbner bases and XL methods are solving the full systems of MQQ equations, we want to ensure that MQQ-SIG does not have a similar small bound on the degree of regularity. A classical way to avoid this is to remove some equations of the system. Indeed, an under-defined system of equations ($n > m$) will have an exponential number of solutions. This is an issue since the complexity of Gröbner bases is also related to the number of solutions [15]. To circumvent this problem, a solution is to fix $n - m$ variables (or more [7]). However, as soon as sufficiently many variables were fixed, we observed that the new system behaved as a "random" system of equations of the same size. This has been also observed and used in the hybrid approach [7].

To confirm this behavior in our context, we have performed experiments on MQQ-SIG equations systems of reduced sizes. The observed degree of regularity is compared to the expected degree of regularity for a random multivariate system of the same size. The strategy for choosing S has changed during the course of our experiments. The experiments where performed with random Boolean matrices. However, from a security against Gröbner bases attack point of view, the most important feature is that we ensure that the 8 linear expressions are removed from the equations set. Below is our experimental strategy for small-scale version of MQQ-SIG equation systems in n variables:

1. Repeat:
2. Generate a bijective multivariate quadratic mapping $P'_i(x_1, \ldots, x_n), i = 1, \ldots, n$
3. Remove the 8 linear expressions $P'_i(x_1, \ldots, x_n), i = 1, \ldots, 8$
4. Multiply with random nonsingular Boolean matrices S_R and T_R, $\mathbf{P} = \mathbf{S_R} \circ P' \circ \mathbf{T_R}$.
5. For $j = 8$ to $j = \frac{n}{2}$ do:
   (a) Remove the last $8 - j$ equations from $\mathbf{P}$.
   (b) Set a random Boolean vector $(x_{n-j+1}, \ldots, x_n) \in \{0,1\}^j$
   (c) Obtain a system $\mathbf{P_1} = \{P_i(x_1, \ldots, x_{n-j}) \mid i = 1, \ldots, n-j\}$ of $n - j$ equations with $n - j$ variables $(x_1, \ldots, x_{n-j})$
   (d) Call $F_4(\mathbf{P_1})$ algorithm from Magma, to find a Gröbner basis for the system $\mathbf{P_1}$, and measure the degree of regularity.
6. Compute the average degree of regularity.

Table 5. The average degree of regularity for a MQQ signature system in V variables with R equations removed. In parentheses, the expected degree of regularity for a random system of size $V - R$.

R/V	16	24	32	40	48	56	64
8	**3**, **00**(**3**)	3,33 (5)	3,75 (6)	4,15 (6)	4,30 (7)	5 (8)	6 (9)
9		3,09 (4)	3,97 (5)	4,05 (6)	4,10 (7)	4 (8)	4 (9)
10		3,74 (4)	4,00 (5)	4,04 (6)	4,30 (7)	4 (8)	5 (9)
11		3,87 (4)	4,01 (5)	4,56 (6)	4,90 (7)	5 (8)	5 (9)
12		**3**, **93**(**4**)	4,06 (5)	5,00 (6)	5,00 (7)	5 (8)	5 (9)
13			4,33 (5)	5,00 (6)	5,00 (7)	5 (8)	· (9)
14			4,48 (5)	5,00 (6)	5,50 (7)	6 (8)	· (9)
15			4,46 (5)	5,00 (6)	5,60 (7)	6 (8)	· (8)
16			**4**, **21**(**5**)	5,00 (6)	5,60 (6)	6 (7)	· (8)
17				5,00 (5)	5,90 (6)	· (7)	· (8)
18				5,00 (5)	5,90 (6)	· (7)	· (8)
19				5,00 (5)	6,00 (6)	· (7)	· (8)
20				**5**, **00**(**5**)	6,00 (6)	· (7)	· (8)
21					6,00 (6)	· (7)	· (8)
22					6,00 (6)	· (7)	· (8)
23					6,00 (6)	· (7)	· (8)
24					**6**, **00**(**6**)	6 (6)	· (7)
25						6 (6)	· (7)
26						6 (6)	· (7)
27						6 (6)	· (7)
28						**6**(**6**)	· (7)
29							· (7)
30							· (7)
31							· (7)
32							**6**(**6**)

We have performed 100 experiments for 16, 24,32 and 40 variables. Due to the complexity, the experiments have only been repeated 10 times for 48 variables and just once for 56 and 64 variables. For 56 and 64 variables many of the instances either required more than the 1TB RAM our system has, or did not finish after about 1 month of computation. These instances are marked with a · in the table. We also experienced that 72 variables with 36 equations removed did not finish after about a month of computation. The experiments were done with Magma 2.17-3's implementation [30] of the F_4[16] algorithm on a workstation with 32 cores based on Intel Xeon 2.27GHz, with 1TB of RAM memory. The results of these experiments are listed in Table 5. In the table the expected degree of regularity for a random system of equations over $GF(2)$ in $V - R$ variables are also listed in parentheses. These numbers have been calculated using the formula provided in [2]. From the table we see that the bigger percentage of equations we remove from the system, the closer the measured degree of regularity is to a random system of equations. The reason for this is that we are removing crucial relations among terms, thus rendering the remaining sets of equations as random sets of multivariate equations. It is then natural to formulate the following conjecture:

Conjecture 1. For every full set of public key equations produced by MQQ as defined in steps 1–3 in Table 2, removing $\frac{n}{2}$ of the equations, makes the remaining set of $\frac{n}{2}$ multivariate quadratic equations to act as a set of $\frac{n}{2}$ random multivariate quadratic equations with $\frac{n}{2}$ variables in $GF(2)$.

5.2 The Size of the Pool of MQQs of Order 2^8

It is very important to address the question of the size of the set of MQQs of order 2^8 that we use in our MQQ-SIG scheme. In [10], Chen et al., gave a lower bound on the number of MQQs of order 2^8. That number is projected to 2^{273}. However, we are using additional conditions (8). By a heuristical measuring we have obtained that approximately one in 2^7 randomly generated MQQs of order 2^8 complies with the conditions (8). That means that the lower bound of the size of the pool of MQQs of order 2^8 is 2^{266}.

5.3 Secret Key Leakage Scenarios

Originally this attack was presented to us by an anonymous reviewer of an earlier variant of our scheme submitted to WCC 2011. We would like to express *big acknowledgement* to that anonymous reviewer.

In a previous version of our scheme instead of $\mathbf{y} = r_0||h_0$, the value $\mathbf{y} = h$ obtained as the output of the hashing procedure is n bits long, and the signature part is $\mathbf{x} = D(\mathbf{y})$. The following Chosen Message Attack could then be launched. An attacker asks for signatures of $1+n+\binom{n}{2}+O(1)$ messages i.e. he will have the triplets $(M_i, \mathbf{x}_i, \mathbf{y}_i \equiv Hash(M_i))$. He will then attempt to recover the missing $\frac{n}{2}$ equations in the public key. Given the missing equations he can successfully launch an efficient Gröbner bases attack.

Consider the extraction of the first missing equation $y_1 = P_1(x_1, \ldots, x_n)$, which can be expressed in a general form as:

$$y_1 = d_0 + d_1x_1 + d_2x_2 + \ldots + d_nx_n + d_{n+1}x_1x_2 + \ldots + d_{2n+1}x_2x_3 + \ldots + d_{1+n+\binom{n}{2}}x_{n-1}x_n. \tag{15}$$

Since the attacker knows the values of $1 + n + \binom{n}{2} + O(1)$ triplets $(M_i, \mathbf{x}_i, \mathbf{y}_i)$, from the equation (15), with high probability, he can obtain a full rank linear system of equations with $1 + n + \binom{n}{2}$ unknown variables d_j. Additionally and most importantly he knows the corresponding values $y_1^{(i)}$ for every of the values $\mathbf{y}_i = (y_1^{(i)}, \ldots, y_n^{(i)})$. Thus, by solving the obtained linear system of equations he can recover the values of the coefficients d_j i.e. he can recover the first missing equation. The extraction of other hidden equations is similar.

This attack is easily mitigated by our strategy to construct the values $\mathbf{y}_0 = r_0||h_0$ and $\mathbf{y}_1 = r_1||h_1$ where r_0 and r_1 are strings of $\frac{n}{2}$ randomly generated bits with every signing invocation, and $h = h_0||h_1$ is the hash output that is digesting the message M.

We formulate the previous discussion about the leakage of the private key in the non-randomized MQQ-SIG and its prevention by the following two lemmas:

Lemma 2. *For any MQQ signature scheme with K expressions removed, if the signatures for the messages M are obtained as $\mathbf{x} = D(\mathbf{y})$, where $\mathbf{y} = Hash(M)$, the extraction of the removed part has complexity of $O(Kn^2)$.* □

Lemma 3. *For the MQQ-SIG signature scheme as defined in steps 1–3 in Table 2, by removing $\frac{n}{2}$ of the expressions, an attack for extraction of the removed part as in Lemma 2 has complexity of $O(2^{n^3})$.*

Proof. Since the signature for a message M has two parts $\mathbf{x}_0$ and $\mathbf{x}_1$ that are computed as $\mathbf{x}_0 = D(r_0||h_0)$ and $\mathbf{x}_1 = D(r_1||h_1)$ where $h = h_0||h_1 = Hash(M)$, and the values r_0 and r_1 are $\frac{n}{2}$-bit values chosen uniformly at random for every particular procedure of signing, the extraction technique from Lemma 2 can give the correct extraction of the hidden part if and only if for all $O(n^2)$ queries, the random values r_0 and r_1 are known to the attacker. Having in mind that for every produced signature the values r_0 and r_1 are unknown, fresh, uniformly distributed random values, the probability of guessing their values is $2^{-\frac{n}{2}} \times 2^{-\frac{n}{2}} = 2^{-n}$. For all $O(n^2)$ queries this gives us a total probability of $(2^{-n})^{n^2} = 2^{-n^3}$, i.e. the complexity for extracting the hidden part is $O(2^{n^3})$. □

5.4 MQQ-SIG Is Provably CMA Resistant

We will use the following definition of security against chosen message attack [27]:

Definition 3. *Signature scheme (Gen, Sign, Vrfy) is **existentially unforgeable under a chosen-message attack** if for all probabilistic, polynomial-time adversaries A, the success probability of A in the following experiment is negligible (as a function of k):*

1. *The key-generation algorithm Gen(1^k) is run to obtain a pair of keys (pk, sk)*
2. *A is given pk and allowed to interact with a signing oracle $Sign_{sk}(\cdot)$, requesting signatures on as many messages as it likes. Let M denote the set of messages queried to the signing oracle by A.*
3. *Eventually, A outputs (m, σ)*
4. *A **succeeds** if $Vrfy_{pk}(m, \sigma) = 1$ and $m \notin M$*

It is well known that solving multivariate quadratic polynomials is an NP-complete problem (see for instance [20]). This theorem is repeated below.

Theorem 1 ([20]). *Let $P_i(x_1, \ldots, x_n), 1 \leq i \leq m$ be a collection of polynomials over GF[2]. The problem of finding $u_1, \ldots, u_n$ such that $P_i(u_1, \ldots, u_n) = 0$ for $1 \leq i \leq m$ remains NP-complete even if none of the polynomials has a term involving more than two variables or if there is just one polynomial.*

Theorem 2. *MQQ-SIG is CMA resistant in the random oracle model under the assumptions that solving $\frac{n}{2}$ MQQ equations with n variables is as hard as solving systems of $\frac{n}{2}$ random multivariate quadratic equations.*

In what follows we give a sketch of the proof and the ideas how to use the fact that the verification of the MQQ-SIG signatures depends on the values h_0 and h_1 that are each $\frac{n}{2}$ bits long. This fact implies that a chosen message attack on MQQ-SIG would need either at least $2^{\frac{n}{2}}$ pairs of messages in order

to find a collision of the used hash function or to solve the system of $\frac{n}{2}$ random multivariate quadratic equations with n variables. A formal proof showing the strict reduction from the CMA-resistance of the scheme to the assumption that solving $\frac{n}{2}$ MQQ equations with n variables is as hard as solving systems of $\frac{n}{2}$ random multivariate quadratic equations with n variables will be given in the extended version of this paper.

Proof. (sketch) The security parameter input to the generating algorithm is $k = \frac{n}{2}$, which controls the number of equations over $GF(2)$ and is directly connected with the value n: the output size of the hash function.

Given the assumption that solving $\frac{n}{2}$ MQQ equations with n variables is as hard as solving systems of $\frac{n}{2}$ random multivariate quadratic equations, there are no structural weaknesses of the MQQ equations that can be exploited to solve the system faster then solving $\frac{n}{2}$ random multivariate quadratic equations. This means the adversary has basically three strategies of breaking MQQ-SIG:

1. To find a collision in the hash digest $(h_0||h_1)$ of length $n = 2k$.
2. To solve two systems of $\frac{n}{2}$ MQ equations with n Boolean variables.
3. Some combination of the two above.

Strategy 1: Breaking with the strategy 1 means finding a collision for a random oracle with a $n = 2k$ bit output. Interacting with the signing oracle will not help the adversary for this instance, since he is only interested in the output of the random oracle. By the generic birthday attack the adversary needs $O(2^k)$ queries to the random oracle to find a collision for the whole digest. The probability for a polynomial time adversary to break the signature scheme by finding a collision in the digest is therefore negligible in k.

Strategy 2: Under the assumption that solving the $\frac{n}{2}$ MQQ equations in n variables is as hard as solving k MQ equations in k variables, we know by Theorem 1 that the probability the adversary solves either of the equations with the strategy 2 is negligible in k. However, to prove that the signature scheme is CMA, we must also show that querying the signing oracle gives the adversary no significant advantage in solving the equations. There are two ways the signing oracle might leak information.

(a) Signing leaks information about the hidden equations:
In Lemma 3 we proved that extracting information about the removed part has complexity $O(2^{n^3})$. With our security parameter of $k = \frac{n}{2}$ this is out of reach for a polynomially bound adversary.

(b) Signing leaks some other information that can help solve the equation system:
Consider the following game where the adversary does not have access to the random oracle. The adversary asks for a signature for a chosen message M. The signing oracle then flips a coin.
   I If the coin land on heads the signing oracle outputs the digest $H(M) = (h_0||h_1)$, and the corresponding signature $(\mathbf{x}_0, \mathbf{x}_1)$.

II If the coin lands on tails the signing oracle outputs the evaluation of the encryption function in some random numbers $(E(r_0), E(r_1))$, and the corresponding random numbers (r_0, r_1).

The adversary is then asked if the coin is heads or tails.

Since by the definition of random oracles the output of $H(M)$ is independent of M, it should be clear that the adversary has no way of winning the game above. This illustrates that from the adversary point of view, there is no difference between querying the signing oracle and evaluating the known equations on random inputs. The fact that the adversary actually has access to the random oracle does not change this conclusion because the adversary has no control over the output of the random oracle.

To summarize this means that signing reveals no information about the hidden equations, and leaks no other information that can be used to solve the equations. The signature scheme is therefore CMA with respect to the strategy number 2.

Strategy 3: First note that finding a $k - l$, $1 \leq l \leq k$, bit collision in, for instance h_0, will not help computing the corresponding $\mathbf{x}_0$. The reason for this is the nature of the random MQ equations, where the solution to the system will drastically change by just flipping one output bit. Namely, each output bit depends on average on $\frac{k(k-1)}{2}$ combination of all pairs of variables. This means that the best for the adversary in strategy attack number 3 is to find a collision in either h_0 or h_1, and to solve the equation system for the part that a solution is not known. This requires "just" $O(2^{\frac{k}{2}-1})$ calls to the random oracle. However, the adversary still needs to solve a system of k equations in $2k$ variables, proven to be CMA resistant by the arguments under the attack strategy number 2. □

5.5 Non-applicability of Successful Attacks against STS on MQQ-SIG

An anonymous reviewer for IMACC 2011 (to whom we express *big acknowledgement*) has pointed out an interesting comment that MQQ-SIG scheme looks similar as STS schemes and thus the successful attacks that have broken STS schemes may also break MQQ-SIG. Here we explain the crucial and essential differences between STS and MQQ-SIG schemes and the non-applicability of successful attacks against STS on MQQ-SIG.

The Stepwise Triangular Scheme was introduced by Wolf et al., [48] as a generalization of earlier multivariate quadratic schemes, such as [45,34,24,26]. The main purpose of the generalization in [48] was to show how all these schemes, and the whole STS family in general, is either insecure or impractical. The general attacks presented exploit the chain of kernels introduced by the triangular structure of the hidden polynomials.

There are at least two important reasons why this attack is not applicable on MQQ-SIG. First, even tough the kernel of two adjacent sub-blocks share half of each others variables, the triangular structure of the hidden polynomials does not result in a chain of kernels. The production of the public key in MQQ-SIG is essentially parallel and chained for the whole n-dimentional space, while the

production of the public key in STS is essentially sequential with increasingly larger embedded subspaces. It is this structure that the attacks on STS exploit.

The second reason is that the attacks linearly combine the public key expressions in order to get ranks within certain values. Non-applicability of these attacks against MQQ-SIG is due to the fact that half of the public key expressions are removed, and linearly combining the remaining half in order to obtain low ranks does not necessarily produce vectors from the kernel of the transformation T^{-1}.

6 Operating Characteristics

In this section we discuss the sizes of the private and public key as well as the number of operations for verification and signing.

Table 6. Comparison between RSA, ECDSA, and several MQ schemes: MQQ-SIG, Rainbow, TTS and 3ICP. Operations have been performed in 64-bit mode of operation on Intel Core i7 920X machine running at 2 GHz.

Security level (power of 2)	Algorithm	KeyGen	Sign 59 bytes (CPU cycles)	Verify (CPU cycles)	Private key size (bytes)	Public key size (bytes)	Signature size (bytes)
80	RSA1024	102,869,553	2,213,112	60,084	1024	128	128
	ECDSA160	1,201,188	944,364	1,083,060	60	40	40
	MQQSIG160	799,501,482	6,534	92,232	401	137,408	40
	RainbowBinary256181212	30,311,648	38,784	43,800	23,408	30,240	42
96	RSA1536	322,324,721	5,452,076	87,516	1536	192	192
	ECDSA192	1,799,284	1,390,560	1,662,664	72	48	48
	MQQSIG192	800,724,096	7,938	138,972	465	222,360	48
112	RSA2048	786,466,598	11,020,696	125,776	2048	256	256
	ECDSA224	2,022,896	1,555,740	1,821,348	84	56	56
	MQQSIG224	1,107,486,126	9,492	184,392	529	352,828	56
128	RSA3072	2,719,353,538	31,941,760	230,536	3072	384	384
	ECDSA256	2,296,976	1,780,524	2,085,588	96	64	64
	MQQSIG256	1,501,955,022	9,138	218,700	593	526,368	64
	TTS6440	60,827,704	84,892	76,224	16,608	57,600	43
	3ICP	15,520,100	1,641,032	60,856	12,768	35,712	36

6.1 The Size of the Public and the Private Key

Since the public key consists of $\frac{n}{2}$ randomly generated multivariate quadratic equations, the size of the public key follows the rules given in [49]. So, for n bit blocks the size of the public key is $0.5 \times n \times (1 + \frac{n(n+1)}{2})$ bits. The private key of our scheme is the tuple $(\sigma_0^0, \sigma_0^1, *)$. The corresponding memory size needed for storage of the private key is $2n + 81$ bytes.

In Table 6, there are two columns for the size of the private and public key and as we can see for MQQ-SIG the size of the public key for $n \in \{160, 192, 224, 256\}$ is in the range from 125 up to 521 KBytes.

We want to emphasize that recently Samardjiska and Chen in [43] have proposed extension of their algorithms for construction of MQQs over arbitrary finite fields and that by their construction it is possible to reduce the huge public key size of MQQ-SIG to be in the range 2.3 – 8.8 Kbytes.

6.2 Performance of the Software Implementation of the MQQ-SIG Algorithm

We have implemented MQQ-SIG in C for the SUPERCOP benchmarking system [6] and tested it together with the corresponding RSA [41] and ECC [32,29] (actually ECDSA) and several other multivariate quadratic systems such as: Rainbow [14], enhanced TTS [50] and 3ICP [13]. In Table 6 we give the comparison of the mentioned signatures schemes where the measurements were performed in 64-bit mode of operation on Intel Core i7 920X machine running at 2 GHz. Although, our C code is not yet optimized for the key generation part, we expect that the performance of key generation part to be the most time consuming part of our algorithm.

From the Table 6 it is clear that in signing of 59 bytes MQQ-SIG is faster than RSA in the range from 300 up to 3500 times, and is faster than ECDSA in the range from 140 up to 200 times. If we exclude the time for hashing the messages, signing operations in MQQ-SIG in Table 6 take from 2,500 up to 5,000 cycles. MQQ-SIG is also significantly faster than other multivariate methods such as Rainbow, TTS or 3ICP and that performance advantage in the signing procedure is in the range from 5 to 20 times.

The verification speed in our code is not optimized so far. We expect the optimized verification speed of MQQ-SIG to be in the range of Rainbow, TTS and 3ICP.

7 Conclusions

We have constructed a multivariate quadratic digital signature scheme MQQ-SIG based on multivariate quadratic quasigroups.

By learning about the weaknesses of the previous attempt to design a multivariate quadratic scheme based on quasigroups - MQQ, by analyzing the successful attacks on all existing MQ schemas, and by our experimentally supported assumption that solving $\frac{n}{2}$ quadratic polynomials with n variables is as hard as solving random systems of equations, we have designed a digital signature scheme that in the random oracle model is provably CMA resistant and that we believe is strong enough to attract the attention of the cryptographic community.

The efficiency of producing digital signatures of our scheme outperforms all the existing signature schemes (RSA, ECDSA and other MQ schemes) in the range from 5 up to 3,500 times. The speed of verification of our scheme is similar to the other MQ schemes. However the MQQ-SIG scheme that was described in this paper has an unpractically big public key. The ongoing research efforts are in this direction and soon we can expect MQQ-SIG variants with significantly smaller public keys.

We believe that its superior performance will allow an employment of strong and fast authentication protocols based on the paradigm of the public key cryptography in many new areas of our modern society.

Acknowledgements. We would like to thank anonymous reviewers of the INTRUST 2011 conference for their useful comments that improved the text of the paper. The work described in this paper has been supported by the Commission of the European Communities through the ICT program under contract ICT-2007-216676 (ECRYPT-II). J.-C. Faugère and L. Perret are also supported by the French ANR under the Computer Algebra and Cryptography (CAC) project (ANR-09-JCJCJ-0064-01).

References

1. Bardet, M.: Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie. PhD thesis, Université de Paris VI (2004)
2. Bardet, M., Faugère, J.-C., Salvy, B.: Complexity study of Gröbner basis computation. Technical report, INRIA (2002), `http://www.inria.fr/rrrt/rr-5049.html`
3. Bardet, M., Faugère, J.-C., Salvy, B.: On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In: Proc. International Conference on Polynomial System Solving (ICPSS), pp. 71–75 (2004)
4. Bardet, M., Faugère, J.-C., Salvy, B., Yang, B.-Y.: Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. In: Proc. of MEGA 2005, Eighth International Symposium on Effective Methods in Algebraic Geometry (2005)
5. Belousov, V.D.: Osnovi teorii kvazigrup i lup, Nauka, Moscow (1967) (in russian)
6. Bernstein, D.J., Lange, T. (eds.): eBACS: ECRYPT benchmarking of cryptographic systems (accessed January 12, 2011)
7. Bettale, L., Faugère, J.-C., Perret, L.: Hybrid approach for solving multivariate systems over finite fields. Journal of Mathematical Cryptology 3(3), 177–197 (2009)
8. Bettale, L., Faugère, J.-C., Perret, L.: Cryptanalysis of Multivariate and Odd-Characteristic HFE Variants. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 441–458. Springer, Heidelberg (2011)
9. Bouillaguet, C., Faugère, J.-C., Fouque, P.-A., Perret, L.: Practical Cryptanalysis of the Identification Scheme Based on the Isomorphism of Polynomial with One Secret Problem. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 473–493. Springer, Heidelberg (2011)
10. Chen, Y., Knapskog, S.J., Gligoroski, D.: Multivariate quadratic quasigroups (MQQs): Construction, bounds and complexity. In: Inscrypt, 6th International Conference on Information Security and Cryptology. Science Press of China (October 2010)
11. Davis, P.J.: CirculantMatrices. AMS Chelsea Publishing (1994)
12. Denes, J., Keedwell, A.D.: Latin squares and their applications. Academic Press, New York (1974)
13. Ding, J., Wolf, C., Yang, B.-Y.: -Invertible Cycles for Ultivariate Uadratic (q) Public Key Cryptography. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 266–281. Springer, Heidelberg (2007)

14. Ding, J., Yang, B.-Y., Chen, C.-H.O., Chen, M.-S., Cheng, C.-M.: New Differential-Algebraic Attacks and Reparametrization of Rainbow. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 242–257. Springer, Heidelberg (2008)
15. Faugère, J.C., Gianni, P., Lazard, D., Mora, T.: Efficient computation of zero-dimensional Gröbner bases by change of ordering. J. Symb. Comput. 16, 329–344 (1993)
16. Faugere, J.-C.: A new efficient algorithm for computing Gröbner basis, F4 (2000), http://citeseer.ist.psu.edu/faugere00new.html
17. Faugère, J.-C., Ødegård, R.S., Perret, L., Gligoroski, D.: Analysis of the MQQ Public Key Cryptosystem. In: Heng, S.-H., Wright, R.N., Goi, B.-M. (eds.) CANS 2010. LNCS, vol. 6467, pp. 169–183. Springer, Heidelberg (2010)
18. Faugère, J.-C., Perret, L.: Polynomial Equivalence Problems: Algorithmic and Theoretical Aspects. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 30–47. Springer, Heidelberg (2006)
19. Fouque, P.-A., Granboulan, L., Stern, J.: Differential Cryptanalysis for Multivariate Schemes. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 341–353. Springer, Heidelberg (2005)
20. Garey, M.R., Johnson, D.S.: Computers and Intractability. A guide to the theory of NP-Completeness. Bell Telephone Laberatories, Incoperated (1979)
21. Gligoroski, D., Markovski, S., Knapskog, S.J.: Public key block cipher based on multivariate quadratic quasigroups. Cryptology ePrint Archive, Report 2008/320
22. Gligoroski, D., Markovski, S., Knapskog, S.J.: Multivariate quadratic trapdoor functions based on multivariate quadratic quasigroups. In: MATH 2008: Proceedings of the American Conference on Applied Mathematics, pp. 44–49. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point (2008)
23. Goubin, L., Courtois, N.T.: Cryptanalysis of the TTM Cryptosystem. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 44–57. Springer, Heidelberg (2000)
24. Goubin, L., Courtois, N.T., Schlumbergersema, C.: Cryptanalysis of the TTM Cryptosystem. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 44–57. Springer, Heidelberg (2000)
25. Imai, H., Matsumoto, T.: Algebraic Methods for Constructing Asymmetric Cryptosystems. In: Calmet, J. (ed.) AAECC-3. LNCS, vol. 229, pp. 108–119. Springer, Heidelberg (1986)
26. Kasahara, M., Sakai, R.: A construction of public key cryptosystem for realizing ciphertext of size 100 bit and digital signature scheme. IEICE Transactions 87-A(1), 102–109 (2004)
27. Katz, J.: Digital Signatures. Springer (2010)
28. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 19–30. Springer, Heidelberg (1999)
29. Koblitz, N.: Elliptic Curve Cryptosystems. Mathematics of Computation 48(177), 203–209 (1987)
30. MAGMA. High performance software for algebra, number theory, and geometry — a large commercial software package, http://magma.maths.usyd.edu.au
31. Matsumoto, T., Imai, H.: Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 419–453. Springer, Heidelberg (1988)
32. Miller, V.S.: Use of Elliptic Curves in Cryptography. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 417–426. Springer, Heidelberg (1986)

33. Moh, T.: A public key system with signature and master key functions. Communications in Algebra (1999)
34. Moh, T.: A public key system with signature and master key functions (1999)
35. Mohamed, M.S.E., Ding, J., Buchmann, J., Werner, F.: Algebraic Attack on the MQQ Public Key Cryptosystem. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 392–401. Springer, Heidelberg (2009)
36. Patarin, J.: Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 33–48. Springer, Heidelberg (1996)
37. Patarin, J.: Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt 98. Des. Codes Cryptography 20, 175–209 (2000)
38. Perret, L.: A Fast Cryptanalysis of the Isomorphism of Polynomials with One Secret Problem. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 354–370. Springer, Heidelberg (2005)
39. Perret, L.: Personal e-mail communication with Danilo Gligoroski (2008)
40. Petzoldt, A., Bulygin, S., Buchmann, J.: Cyclicrainbow - a multivariate signature scheme with a partially cyclic public key based on rainbow. Cryptology ePrint Archive, Report 2010/424 (2010), http://eprint.iacr.org/
41. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21, 120–126 (1978)
42. Samardjiska, S., Markovski, S., Gligoroski, D.: Multivariate quasigroups defined by t-functions. In: Proceedings of SCC 2010 - The 2nd International Conference on Symbolic Computation and Cryptography (2010)
43. Samardjiska, S., Chen, Y., Gligoroski, D.: Construction of multivariate quadratic quasigroups (mqqs) in arbitrary galois fields. In: Proceedings of the International Conference on Information Assurance and Security (IAS) 2011, Malacca, Malaysia (2011)
44. Shamir, A.: Efficient Signature Schemes Based on Birational Permutations. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 1–12. Springer, Heidelberg (1994)
45. Shamir, A.: Efficient Signature Schemes Based on Birational Permutations. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 1–12. Springer, Heidelberg (1994)
46. Singh, R.P., Sarma, B.K., Saikia, A.: Public key cryptography using permutation p-polynomials over finite fields. Cryptology ePrint Archive, Report 2009/208 (2009), http://eprint.iacr.org/
47. Smith, J.D.H.: An introduction to quasigroups and their representations. Chapman & Hall/CRC (2007)
48. Wolf, C., Braeken, A., Preneel, B.: On the security of stepwise triangular systems. Des. Codes Cryptography 40, 285–302 (2006)
49. Wolf, C., Preneel, B.: Taxonomy of public key schemes based on the problem of multivariate quadratic equations. Cryptology ePrint Archive, Report 2005/077 (2005)
50. Yang, B.-Y., Chen, J.-M.: Building Secure Tame-like Multivariate Public-Key Cryptosystems: The New TTS. In: Boyd, C., González Nieto, J.M. (eds.) ACISP 2005. LNCS, vol. 3574, pp. 518–531. Springer, Heidelberg (2005)
51. Yang, B.-Y., Cheng, C.-M., Chen, B.-R., Chen, J.-M.: Implementing Minimized Multivariate PKC on Low-Resource Embedded Systems. In: Clark, J.A., Paige, R.F., Polack, F.A.C., Brooke, P.J. (eds.) SPC 2006. LNCS, vol. 3934, pp. 73–88. Springer, Heidelberg (2006)