The Digital Signature Scheme MQQ-SIG

Intellectual Property Statement and Technical Description

First published: 10 October 2010, Last update: 6 December 2010

Danilo Gligoroski[1] and Rune Steinsmo Ødegård[2] and Rune Erlend Jensen[2] and Ludovic Perret[4] and Jean-Charles Faugère[5] and Svein Johan Knapskog[2] and Smile Markovski[3]

[1] Department of Telematics, Faculty of Information Technology, Mathematics and Electrical Engineering, The Norwegian University of Science and Technology (NTNU), O.S.Bragstads plass 2E, N-7491 Trondheim, NORWAY, danilog@item.ntnu.no

[2] Norwegian University of Science and Technology Centre for Quantifiable Quality of Service in Communication Systems. O.S. Bragstads plass 2E, N-7491 Trondheim, NORWAY, knapskog@Q2S.ntnu.no, rune.odegard@q2s.ntnu.no, runeerle@stud.ntnu.no

[3] "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

[4] Pierre and Marie Curie University - Paris, Laboratory of Computer Sciences, Paris 6, 104 avenue du Président Kennedy 75016 Paris FRANCE, ludovic.perret@lip6.fr

[5] UPMC, Université Paris 06, LIP6 INRIA, Centre Paris-Rocquencourt, SALSA Project-team CNRS, UMR 7606, LIP6 4, place Jussieu 75252 Paris, Cedex 5, FRANCE jean-charles.faugere@inria.fr

Abstract: This document contains the Intellectual Property Statement and the technical description of the MQQ-SIG - a new public key digital signature scheme. The complete scientific publication covering the design rationale and the security analysis will be given in a separate publication. MQQ-SIG consists of $n - \frac{n}{4}$ quadratic polynomials with n Boolean variables where $n = 160$, 196, 224 or 256.

Keywords: Public Key Cryptosystems, Fast signature generation, Multivariate Quadratic Polynomials, Quasigroup String Transformations, Multivariate Quadratic Quasigroup

1 Intellectual Property Statement

We, the seven names given in the title of this document and undersigned on this statement, the authors and designers of MQQ-SIG digital signature scheme, do hereby agree to grant any interested party an irrevocable, royalty free licence to practice, implement and use MQQ-SIG digital signature scheme, provided our roles as authors and designers of the MQQ-SIG digital signature scheme are recognized by the interested party as authors and designers of the MQQ-SIG digital signature scheme.

Name	Signature	Place	Date
1. Danilo Gligoroski	________________________	Trondheim	__________
2. Svein Johan Knapskog	________________________	Trondheim	__________
3. Smile Markovski	________________________	Skopje	__________
4. Rune Steinsmo Ødegård	________________________	Trondheim	__________
5. Rune Erlend Jensen	________________________	Trondheim	__________
6. Ludovic Perret	________________________	Paris	__________
7. Jean-Charles Faugère	________________________	Paris	__________

2 Description of the MQQ-SIG digital signature scheme

A generic description for our scheme can be expressed as a $\frac{3}{4}$ 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' is a bijective multivariate quadratic mapping on $\{0,1\}^n$.

The bijective multivariate quadratic mapping $P' : \{0,1\}^n \to \{0,1\}^n$ is defined in Table 1.

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

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

The algorithm for generating the public and private key is defined in the Table 2.

Algorithm for generating Public and Private key for the MQQ-SIG scheme
Input: Integer n, where $n = 32 \times k$ and $k \in \{5, 6, 7, 8\}$.
Output: Public key $\mathbf{P}$: $n - \frac{n}{4}$ multivariate quadratic polynomials $P_i(x_1, \dots, x_n)$, $i = 1 + \frac{n}{4}, \dots, n$, Private key: Two permutations σ_1 and σ_K of the numbers $\{1, \dots, n\}$, and 81 bytes for encoding a quasigroup $*$.
1. Generate an MQQ $*$ according to equations (1) ... (4). 2. Generate a nonsingular $n \times n$ Boolean matrix $\mathbf{S}$ and affine transformation $\mathbf{S}'$ according to equations (5), ..., (11). 3. Compute $\mathbf{y} = \mathbf{S}(P'(\mathbf{S}'(\mathbf{x})))$, where $\mathbf{x} = (x_1, \dots, x_n)$. 4. Output: The public key is $\mathbf{y}$ as $n - \frac{n}{4}$ multivariate quadratic polynomials $P_i(x_1, \dots, x_n)$ $i = 1 + \frac{n}{4}, \dots, n$, and the private key is the tuple $(\sigma_1, \sigma_K, *)$.

Table 2. Generating the public and private key

The algorithm for signing by the private key $(\sigma_1, \sigma_K, *)$ is defined in Table 3.

Algorithm for digital signature with the private key $(\sigma_1, \sigma_K, *)$
Input: A document M to be signed.
Output: A signature $\mathbf{sig} = (x_1, \dots, x_n)$.
1. Compute $\mathbf{y} = (y_1, \dots, y_n) = H(M)\|_n$, where M is the message to be signed, $H()$ is a standardized cryptographic hash function such as SHA-1, or SHA-2, with a hash output of not less than n bits. The notation $H(M)\|_n$ denotes the least significant n bits from the hash output $H(M)$. 2. Set $\mathbf{y}' = \mathbf{S}^{-1}(\mathbf{y})$. 3. Represent $\mathbf{y}'$ as $\mathbf{y}' = Y_1 \dots Y_{\frac{n}{8}}$ where Y_i are Boolean vectors of dimension 8. 4. By using the left and right parastrophes $\backslash$ and $/$ of the quasigroup $*$ compute $\mathbf{x}' = X_1 \dots X_{\frac{n}{8}}$, such that: $X_1 = Y_1$, $X_j = X_{j-1} \backslash Y_j$, for even $j = 2, 4, \dots$, and $X_j = Y_j / X_{j-1}$, for odd $j = 3, 5, \dots$. 5. Compute $\mathbf{x} = \mathbf{S}^{-1}(\mathbf{x}') + \mathbf{v} = (x_1, \dots, x_n)$. 6. The MQQ-SIG digital signature of the document M is the vector $\mathbf{sig} = (x_1, \dots, x_n)$.

Table 3. Digital signing

The algorithm for signature verification with the public key $\mathbf{P} = \{P_i(x_1, \dots, x_n) \mid i = 1 + \frac{n}{4}, \dots, n\}$ is given in Table 4.

Algorithm for signature verification with a public key $\mathbf{P} = \{P_i(x_1, \dots, x_n) \mid i = 1 + \frac{n}{4}, \dots, n\}$
Input: A document M and its signature $\mathbf{sig} = (x_1, \dots, x_n)$.
Output: TRUE or FALSE.
1. Compute $\mathbf{y} = (y_{1+\frac{n}{4}}, \dots, y_n) = H(M)\vert_{n-\frac{n}{4}}$, where M is the signed message, $H()$ is a standardized cryptographic hash function such as SHA-1, or SHA-2, with a hash output of not less than n bits, and the notation $H(M)\vert_{n-\frac{n}{4}}$ denotes the least significant $n - \frac{n}{4}$ bits from the hash output $H(M)$. 2. Compute $\mathbf{z} = (z_{1+\frac{n}{4}}, \dots, z_n) = \mathbf{P}(\mathbf{sig})$. 3. If $\mathbf{z} = \mathbf{y}$ then return TRUE, else return FALSE.

Table 4. Digital verification

3 Multivariate Quadratic Quasigroups

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

$$\mathbf{x} * \mathbf{y} \equiv \mathbf{B} \cdot \mathbf{U}(\mathbf{x}) \cdot \mathbf{A_2} \cdot \mathbf{y} + \mathbf{B} \cdot \mathbf{A_1} \cdot \mathbf{x} + \mathbf{c} \tag{1}$$

where $\mathbf{x} = (x_1, \dots, x_d)$, $\mathbf{y} = (y_1, \dots, y_d)$, the matrices $\mathbf{A_1}$, $\mathbf{A_2}$ and $\mathbf{B}$ are nonsingular in $GF(2)$, of size $d \times d$, 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 $\mathbf{x} = (x_1, \dots, x_d)$. It is computed by the following expression:

$$\mathbf{U}(\mathbf{x}) = I + \sum_{i=1}^{d-1} \mathbf{U}_i \cdot \mathbf{A_1} \cdot \mathbf{x}, \tag{2}$$

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

Once we have a multivariate quadratic quasigroup

$$*_{vv}(x_1, \dots, x_d, y_1, \dots, y_d) = (f_1(x_1, \dots, x_d, y_1, \dots, y_d), ..., f_d(x_1, \dots, x_d, y_1, \dots, y_d))$$

we will be interested in those quasigroups that will satisfy the following conditions:

$$\forall i \in \{1, \dots, d\}, Rank(\mathbf{B}_{f_i}) \geq 2d - 4, \tag{3a}$$

$$\exists j \in \{1, \dots, d\}, \quad Rank(\mathbf{B}_{f_j}) = 2d - 2 \tag{3b}$$

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

$$\mathbf{B}_{f_i} = [b_{j,k}],\ b_{j,d+k} = b_{d+k,j} = 1, \text{ iff } x_j y_k \text{ is a term in } f_i. \tag{4}$$

Proposition 1. *For $d = 8$, a multivariate quadratic quasigroup that satisfies the conditions (1), ..., (4) can be encoded in a unique way with 81 bytes.*

4 Nonsingular Boolean matrices in MQQ-SIG

In MQQ-SIG the nonsingular matrices $\mathbf{S}$ are defined by the following expression:

$$\mathbf{S}^{-1} = \sum_{i=1}^{K} I_{\sigma_i}, \tag{5}$$

where $I_{\sigma_i},\ i = \{1, 2, \ldots, K\}$ are permutation matrices of size $n = 32 \times k$ and where permutations σ_i are permutations on n elements. They are defined by the following expressions:

$$K = \begin{cases} \quad k \quad, \text{ if } k \text{ is odd,} \\ k+1 \text{ , if } k \text{ is even} \end{cases} \tag{6}$$

$$\begin{cases} \sigma_1 & - \text{ random permutation on } \{1, 2, \ldots n\} \text{ satisfying the condition (8),} \\ \sigma_2 & = \mathit{RotateLeft}(\sigma_1, 32) \text{ satisfying the condition (8),} \\ \sigma_3 & = \mathit{RotateLeft}(\sigma_2, 64) \text{ satisfying the condition (8),} \\ \sigma_j & = \mathit{RotateLeft}(\sigma_{j-1}, 32), \text{ for } j = 4, \ldots, K-1, \text{ satisfying the condition (8),} \\ \sigma_K & - \text{ random permutation on } \{1, 2, \ldots n\} \text{ satisfying the condition (8)} \end{cases} \tag{7}$$

$$\sigma_\nu = \begin{pmatrix} 1 & 2 & \ldots & 8 & 9 & \ldots n-1 & n \\ s_1^{(\nu)} & s_2^{(\nu)} & \ldots & s_8^{(\nu)} & s_9^{(\nu)} & \ldots\ s_{n-1}^{(\nu)} & s_n^{(\nu)} \end{pmatrix}, \ \{s_1^{(\nu)}, s_2^{(\nu)}, \ldots, s_8^{(\nu)}\} \bigcap \{1, 2, \ldots, 8\} = \emptyset \tag{8}$$

where $\mathit{RotateLeft}(\sigma, l)$ denotes a permutation obtained from the permutation σ by rotating it to the left for l positions.

We require an additional condition to be fulfilled by the permutations $\sigma_1, \ldots, \sigma_K$:

$$L = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_{K-1} \\ \sigma_K \end{bmatrix}, \text{ is a Latin Rectangle.} \tag{9}$$

Once we have a nonsingular matrix $\mathbf{S}^{-1}$ we will compute its inverse obtaining

$$\mathbf{S} = (\mathbf{S}^{-1})^{-1}$$

and from there we will obtain the affine transformation

$$\mathbf{S}'(\mathbf{x}) = \mathbf{S} \cdot \mathbf{x} + \mathbf{v}, \tag{10}$$

where the vector $\mathbf{v}$ is n–dimensional Boolean vector defined from the values of the permutation σ_K by the following expression:

$$\mathbf{v} = (v_1, v_2, \ldots, v_n), \text{where } \ v_i = \left(\left(\frac{\left(\left(s^{(K)}_{1+\lfloor \frac{i-1}{8} \rfloor} \right) \bmod 16 \right) \times 16}{2^{(8-i) \bmod 8}} \right) + \left(\frac{s^{(K)}_{65+\lfloor \frac{i-1}{8} \rfloor}}{2^{(8-i) \bmod 8}} \right) \right) \bmod 2. \tag{11}$$

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 $s_1^{(K)}$, …, $s_{\frac{n}{8}}^{(K)}$ and each of them is shifted by four positions to the left. The second array is just simple extraction of the values $s_{65}^{(K)}$, …, $s_{65+\frac{n}{8}}^{(K)}$. Finally we XOR correspondingly those two arrays of values in order to produce the vector $\mathbf{v}$ of n bits.

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

5 Characteristics of the MQQ-SIG digital signature scheme

The main characteristics of our MQQ-SIG digital signature scheme can be briefly summarized as follows:

- there is no message expansion;
- the length of the signature is n bits where ($n = 160, 192, 224$ or 256);
- its conjectured security level is $2^{\frac{n}{2}}$;
- its verification speed is comparable to the speed of other multivariate quadratic PKCs;
- in software its signing speed is in the range of 300–7,000 times faster than RSA and ECC schemes;
- in hardware its signing or verification speed is more than 10,000 times faster than RSA and ECC schemes;
- it is also well suited for producing short signatures in smart cards and RFIDs;

5.1 The size of the public and the private key

The size of the public key is $0.75 \times n \times (1 + \frac{n(n+1)}{2})$ bits. The private key of our scheme is the tuple $(\sigma_1, \sigma_K, *)$. The corresponding memory size needed for storage of the private key is $2n + 81$ bytes. In Table 5 we give the size of the public key (in KBytes) and the size of the private key (in bytes) for $n \in \{160, 192, 224, 256\}$.

n	Size of the public key (KBytes)	Size of the private key (bytes)
160	188.69	401
192	325.71	465
224	516.82	529
256	771.02	593

Table 5. Memory size in KBytes for the public key and in bytes for the private key

Security in bits	Algorithm	KeyGen	Signing of 59 bytes	Verification of a signature of 59 bytes
80	RSA1024	102,869,553	2,230,848	61,116
	ECC160	1,201,188	1,284,800	1,476,196
	MQQSIG160	2,468,300,868	3,948	134,860
96	RSA1536	322,324,721	7,346,420	123,140
	ECC192	1,799,284	1,895,752	2,242,988
	MQQSIG192	4,224,748,748	4,756	80,332
112	RSA2048	786,466,598	14,815,324	174,792
	ECC224	2,022,896	2,108,556	2,501,108
	MQQSIG224	6,884,572,576	5,488	117,828
128	RSA3072	2,719,353,538	31,941,760	315,904
	ECC256	2,296,976	2,418,968	2,833,856
	MQQSIG256	11,536,103,292	4,672	254,164

Table 6. Comparison between performance of RSA, ECC and MQQ in CPU cycles in 64-bit mode of operation on Intel Core i7 920X machine running at 2 GHz.

5.2 Performance of the software and hardware implementation of the MQQ-SIG algorithm

We have implemented MQQ-SIG in C for the SUPERCOP benchmarking system `http://bench.cr.yp.to/supercop.html` and tested it together with the corresponding RSA and ECC. In Table 6 we give the comparison of MQQ-SIG with RSA and ECC in 64-bit mode of operation on Intel Core i7 920X machine running at 2 GHz. The numbers in the table represent CPU cycles. Although, our C code is not yet optimized for the key generation part, we expect that the performance of key generation part would be the most time consuming part of our algorithm.

On the other hand, from the Table 6 it is clear that in signing of 59 bytes MQQ-SIG is faster than RSA in the range from 565 up to 6836 times, and is faster than ECC in the range from 325 up to 517 times.

The verification speed in our code is not so distinctively faster than the corresponding RSA and ECC since it is programmed for one core. We expect that the high parallelizable nature of MQQ-SIG can be used to achieve much higher speeds in multicore systems (CPUs or GPUs).