The Multivariate Probabilistic Encryption Scheme MQQ-ENC

Danilo Gligoroski and Simona Samardjiska

Department of Telematics,
Norwegian University of Science and Technology, Trondheim, Norway
danilog@item.ntnu.no, simonas@item.ntnu.no

Abstract. We propose a new multivariate probabilistic encryption scheme with decryption errors MQQ-ENC that belongs to the family of MQQ-based public key schemes. Similarly to MQQ-SIG, the trapdoor is constructed using quasigroup string transformations with multivariate quadratic quasigroups, and a minus modifier with relatively small and fixed number of removed equations. To make the decryption possible and also efficient, we use a universal hash function to eliminate possibly wrong plaintext candidates. We show that, in this way, the probability of erroneous decryption becomes negligible.

MQQ-ENC is defined over the fields $\mathbb{F}_{2^k}$ for any $k \geq 1$, and can easily be extended to any $\mathbb{F}_{p^k}$, for prime p. One important difference from MQQ-SIG is that in MQQ-ENC we use left MQQs (LMQQs) instead of bilinear MQQs. Our choice can be justified by our extensive experimental analysis that showed the superiority of the LMQQs over the bilinear MQQs for the design of MQQ-ENC.

We apply the standard cryptanalytic techniques on MQQ-ENC, and from the results, we pose a plausible conjecture that the instances of the MQQ-ENC trapdoor are hard instances with respect to the MQ problem. Under this assumption, we adapt the Kobara-Imai conversion of the McEliece scheme for MQQ-ENC and prove that it provides IND–CCA security despite the negligible probability of decryption errors.

We also recommend concrete parameters for MQQ-ENC for encryption of blocks of 128 bits for a security level of $\mathcal{O}(2^{128})$.

Keywords – Multivariate Quadratic Public Key Cryptosystems, Multivariate Quadratic Quasigroup MQQ, Left Multivariate Quadratic Quasigroup LMQQ, Probabilistic encryption with decryption errors, One way encryption, IND–CCA security.

1 Introduction

Multivariate public key cryptography has been a vibrant research area for more than 20 years. Apart from the pure scientific interest in new algorithms, it has attracted a special attention as one of the alternatives to the public key algorithms based on the integer factorization and discrete log problem such as RSA or ECC. Multivariate schemes have a system of multivariate quadratic (MQ) polynomials over a finite field as a public key, and their security relies on the hardness of solving a system of MQ equations over the field. In general this is an instance of a well known NP-complete problem - the MQ problem [29]. There are three aspects that make MQ schemes attractive research field:

1. Multivariate schemes may offer a post-quantum security, since, so far, no quantum algorithm is known for solving systems of MQ equations;
2. Recent breakthroughs [56, 30, 14, 37, 25] in the development of index calculus techniques for the DLP for elliptic curves can lead to development of new index calculus techniques that furthermore can make practical the attacks on ECC with the current set of standardized parameters;
3. MQ schemes have performance advantages over the classical algorithms, especially on multicore architectures, because of their highly parallelizable nature.

Traditionally, the MQ public key cryptosystems are divided in four groups, arising from the four basic schemes MI [36], HFE [47], STS [63] and UOV [38]. They all share the general form of a

MQ scheme, however differ in the construction of the secret internal transformation. The first two are also known as mixed field schemes because they use a ground field and an extension field to construct the trapdoor. The last two are single field systems, and the trapdoor is constructed only in one field with some specific structure. A nice (but now a bit old) survey on these four types can be found in [64].

The successful cryptanalysis of most of the variants of the basic types initiated an emergence of schemes that for one reason or another do not fall in any of the described types. Such is the MQQ cryptosystem [31] and its successor the MQQ-SIG signature scheme [32]. They are single field schemes, and in the construction use elements from quasigroup theory. The internal mapping is a so called quasigroup string transformation [31] of quasigroups represented in multivariate form or MQQs (Multivariate Quadratic Quasigroups). The first scheme MQQ, which is an encryption scheme showed excellent performance characteristics, however, it was soon cryptanalyzed both by using Gröbner basis approach [50] and MutantXL [45]. A deeper explanation of the weaknesses was later given in [24].

To immunize against these successful attacks, in the recent design of MQQ-SIG [32] the authors apply the minus modifier to the original MQQ scheme. Since half of the polynomials are removed, the new design is suitable only for a signature scheme. MQQ-SIG has excellent performance in signing, but still has a big public key, because it is defined over $\mathbb{F}_2$. With the results from [55], where a construction of MQQs over bigger fields was provided, it is possible to extend the design of MQQ-SIG to $\mathbb{F}_{2^k}, k > 1$, and therefore, substantially reduce the size of the public key.

1.1 Our results

We propose a new encryption scheme MQQ-ENC in the family of MQQ-based public key schemes. It has many similarities with the MQQ-SIG scheme. It uses quasigroup string transformations for the internal secret mapping P' and uses specially constructed affine mappings S and T from two circulant matrices, as in MQQ-SIG. On the other hand we have introduced several new ideas.

First of all, MQQ-ENC is a probabilistic encryption scheme with decryption errors. The trapdoor is constructed using a minus modifier with fixed number of removed equations. To make the decryption possible we use a universal hash function to eliminate possibly wrong plaintext candidates. This however can introduce errors, but we show that the probability of an erroneous decryption is negligible. Second, MQQ-ENC is defined over the field $\mathbb{F}_{2^k}$ for any $k \geq 1$, and can be easily extended to any $\mathbb{F}_{p^k}$, for p prime. Third, and most notable, important changes have been made regarding the choice of the building blocks of the internal transformation - the MQQs. Instead of bilinear MQQs as used in MQQ-SIG, we use left MQQs (LMQQs). The main difference between the two structures is that in LMQQs only the left translation is a bijection, and in MQQs both the left and the right translations are bijections. We made an extensive experimental analysis that confirms the superiority of the LMQQs over the bilinear MQQs for the design of MQQ-ENC.

We applied the currently used cryptanalytic techniques to the MQQ-ENC scheme, and as a result of the analysis, we can pose a plausible conjecture that the instances of the MQQ-ENC trapdoor are hard instances with respect to the MQ problem. Under this assumption, we adapt the Kobara-Imai conversion of the McEliece PKE scheme [40] for MQQ-ENC and prove that it provides IND–CCA security despite decryption errors.

1.2 Organization of the paper

In Section 2 we provide the basic definitions for public key encryption with decryption errors and MQ cryptosystems. In Section 3 we explain the notion of left quasigroups and left multivariate quadratic quasigroups over $\mathbb{F}_{2^k}$, and provide the needed algorithms for their construction, for finding the parastrophe of the secret LMQQ (needed for decryption) as well as the algorithms for creating the affine mappings S and T. Section 4 gives the description of the encryption scheme MQQ-ENC. In Section 5 we analyze the security of the trapdoor function of MQQ-ENC, and give the recommended parameters that offer security of $\mathcal{O}(2^{128})$. We describe the conversion for MQQ-ENC that provides $\mathsf{IND\text{-}CCA}$ security in Section 6. The operating characteristics of a preliminary implementation of the scheme are given in Section 7. We conclude the paper in Section 8.

2 Preliminaries

2.1 Probabilistic encryption with correctness errors

We give a definition of probabilistic public-key encryption (PKE) with decryption errors based on [21].

Definition 1. *A public key encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is a triplet of algorithms for key generation, encryption and decryption associated with two finite sets Mspace(n), Coins$(n) \subseteq \{0,1\}^*, n \in \mathbb{N}$ a security parameter, where:*

- *$\mathcal{G}$ called the key-generation algorithm is a probabilistic algorithm that on input 1^n (and internal random coins), outputs the public key and secret key pair $(\mathsf{pk}, \mathsf{sk})$.*
- *$\mathcal{E}$ called the encryption algorithm is a probabilistic algorithm that on input a public key pk, a message $m \in$ Mspace(n) and a random $r \in$ Coins(n) outputs $c = \mathcal{E}_{\mathsf{pk}}(m, r)$ as the ciphertext.*
- *$\mathcal{D}$ called the decryption algorithm is a deterministic algorithm that takes as input a secret key sk and a ciphertext c, and outputs either a message m' (which may fail to equal the original message m) or $\perp$ to indicate invalid.*

The standard definition of probabilistic public-key encryption, due to Goldwasser and Micali [33] requires perfect correctness, i.e. given a valid ciphertext, the decryption algorithm always outputs the original plaintext. Dwork et al. [21], relax the notion of public-key encryption, to allow errors in the decryption process. They give the following definition for α-correct public-key encryption scheme.

Definition 2. *For any function $\alpha : \mathbb{N} \to [0,1]$, a PKE scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is all-keys α-correct if for every pair $(\mathsf{pk}, \mathsf{sk})$ generated by $\mathcal{G}$ on input 1^n*

$$Pr\left[m \leftarrow Mspace(n); r \leftarrow Coins(n) | \mathcal{D}_{\mathsf{sk}}(\mathcal{E}_{\mathsf{pk}}(m, r)) \neq m\right] \leq 1 - \alpha(n).$$

The standard security notions of one-way encryption (OWE) and indistinguishability of encryption ($\mathsf{IND\text{-}CPA}$, $\mathsf{IND\text{-}CCA}$) due to space limitations are not included. We refer the reader to [4, 54] or for ex. [13, 28] for these definitions.

2.2 Multivariate Quadratic (MQ) cryptosystems

Let $\mathbb{F}_{p^k}$ be a finite field of order p^k, where p is prime. We will consider the n-tuples $(u_1, u_2, \ldots, u_n) \in \mathbb{F}^n_{p^k}$, where $n \in \mathbb{N}$, as column vectors and use the notation $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ when appropriate. Also, we will denote by $\mathbf{x}$ the column vector $(x_1, \ldots, x_n)$ over $\mathbb{F}_{p^k}[x_1, \ldots, x_n]$.

Let $P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), ..., p_m(\mathbf{x})) \in \mathbb{F}^m_q[x_1, x_2, \ldots, x_n]$ be a system of m polynomials of degree $d, d \geq 2$. Let $\mathbf{v} \in \mathbb{F}^m_{p^k}$. The problem of Simultaneous Multivariate Equations over the field $\mathbb{F}_{p^k}$ consists of finding a solution $\mathbf{u} \in \mathbb{F}^n_{p^k}$ to the system of equations $\mathbf{v} = (p_1(\mathbf{x}), p_2(\mathbf{x}), ..., p_m(\mathbf{x}))$

over $\mathbb{F}_{p^k}$. It has been shown that for every $d \geq 2$ this problem is NP complete [29]. The case of $d = 2$ has been most exploited and it is called the MQ-problem (MQ - Multivariate Quadratic).

A typical MQ public key scheme relies on the knowledge of a trapdoor for a particular system of polynomials $P \in \mathbb{F}_{p^k}^m[x_1, \ldots, x_n]$. Usually, the system of polynomials P is created as a composition of three polynomial transformations: two affine mappings S and T and one quadratic P' as $P(\mathbf{x}) = T \circ P' \circ S(\mathbf{x})$. The mappings S and T are usually randomly chosen, and serve as a sort of mask that hides the structure of P'. They are part of the private key s. The private key may also contain other secret parameters that allow creation, but also easy inversion of the transformation P'. Without loss of generality, we can assume that the private key is $s = (S, P', T)$.

3 The building blocks of MQQ-ENC

3.1 Left Quasigroups

Definition 3. *The groupoid (Q, q) is called a left (right) quasigroup if the mapping $L_{q,a}(x) = q(a, x)$ $(R_{q,a}(x) = q(x, a))$, is a permutation for every $a \in Q$. If (Q, q) is both left and right quasigroup, than it is simply called a quasigroup. A finite (left/right) quasigroup of n elements is said to be a (left/right) quasigroup of order n.*

Definition 4. *Given a (left) quasigroup (Q, q) a new (left) quasigroup operation $q_\backslash$, called a left parastrophe operation, can be defined on the set Q by $q_\backslash(x, y) = z \Leftrightarrow q(x, z) = y$. The two operations satisfy the identities $q(x, q_\backslash(x, y)) = y$ and $q_\backslash(x, q(x, y)) = y$.*

Definition 5. *Two (left) quasigroups (Q, q_1) and (Q, q_2) are said to be isotopic, if there exist bijections $\alpha, \beta, \gamma : Q \to Q$ such that for all $a, b \in Q$, $\gamma(q_1(a, b)) = q_2(\alpha(a), \beta(b))$. We denote the isotopy by (α, β, γ).*

3.2 Left Multivariate Quadratic Quasigroups (LMQQs)

Let (Q, q) be a left quasigroup of order p^{kd}. We say that (Q, q) is a Left Multivariate Quadratic Quasigroup (LMQQ) if q can be represented as a function $q = (q^{(1)}, q^{(2)}, \ldots, q^{(d)}) : \mathbb{F}_{p^k}^{2d} \to \mathbb{F}_{p^k}^d$, where for every $s = 1, \ldots, d$, $q^{(s)}$ is a quadratic polynomial over $\mathbb{F}_{p^k}$. For simplicity, we take that $Q = \mathbb{F}_{p^k}^d$.

Note that this definition is in accordance with [31] where the notion of Multivariate Quadratic Quasigroups (MQQ) defined using operations over $\mathbb{F}_2$ was introduced. A generalization of MQQs defined over arbitrary finite fields was made in [55]. As a natural extension, appropriate results hold for LMQQs, so from [55], we have that the following theorem holds.

Theorem 1. *The function $q_0 = (q^{(1)}, q^{(2)}, \ldots, q^{(d)}) : \mathbb{F}_{p^k}^{2d} \to \mathbb{F}_{p^k}^d$ such that for every $s = 1, \ldots, d$, the component $q_0^{(s)}$ is of the form*

$$
\begin{aligned}
q_0^{(s)}(x_1, \ldots, x_d, y_1, \ldots, y_d) = p^{(s)}(y_s) + \sum_{1 \leq i,j \leq d} \alpha_{i,j}^{(s)} x_i x_j + \sum_{s < i,j \leq d} \beta_{i,j}^{(s)} y_i y_j + \\
+ \sum_{1 \leq i \leq d, s < j \leq d} \gamma_{i,j}^{(s)} x_i y_j + \sum_{1 \leq i \leq d} \delta_i^{(s)} x_i + \sum_{s < i \leq d} \epsilon_i^{(s)} y_i + \eta^{(s)}, \qquad (1)
\end{aligned}
$$

where $p^{(s)}(x) = ax$, $a \neq 0$, or $p^{(s)}(x) = ax^2$, $a \neq 0$, $p = 2$, defines an LMQQ $(\mathbb{F}_{p^k}^d, q_0)$ of order p^{kd}.

For the purpose of MQQ-ENC, we construct an LMQQ using the following algorithm.

Algorithm: CreateLMQQ(d, p, k)**:**

Input $d, p, k \in \mathbb{N}$, where p is prime.

1. For all $s \in \{1, \dots, d\}$ generate at random from $\mathbb{F}_{p^k}$ the coefficients:
   - $\alpha_{i,j}^{(s)}$, $\delta_i^{(s)}$, for all i, j, $1 \le i, j \le d$, and $\beta_{i,j}^{(s)}$, $\epsilon_i^{(s)}$, for all i, j, $s < i, j \le d$,
   - $\gamma_{i,j}^{(s)}$, for all i, j, $1 \le i \le d, s < j \le d$, and the constant term $\eta^{(s)}$.
2. For all $s \in \{1, \dots, d\}$
   - If $p = 2$ generate at random a bit $r \in \mathbb{F}_2$, otherwise set $r = 0$.
   - Choose at random $a^{(s)} \in \mathbb{F}_{p^k} \setminus \{0\}$. If $r = 0$ set $p^{(s)} = a^{(s)} x_s$, otherwise set $p^{(s)} = a^{(s)} x_s^2$.
3. For all $s \in \{1, \dots, d\}$ construct $q_0^{(s)}(\mathbf{x}, \mathbf{y})$ given by (1), and the LMQQ $q_0 = (q_0^{(1)}, q_0^{(2)}, \dots, q_0^{(d)})$.
4. Generate at random over $\mathbb{F}_{p^k}$, $d \times d$ nonsingular matrices $\mathbf{D}, \mathbf{D_y}$, and vectors $\mathbf{c}, \mathbf{c_y}$ of dimension d.

Output the quintet $(q_0, \mathbf{D}^{-1}, \mathbf{D_y}^{-1}, \mathbf{c}, \mathbf{c_y})$ and the LMQQ of order p^{kd}: $q(\mathbf{x}, \mathbf{y}) = \mathbf{D} \cdot q_0(\mathbf{x}, \mathbf{D_y} \cdot \mathbf{y} + \mathbf{c_y}) + \mathbf{c}$.

Note that the two LMQQs q and q_0 are isotopic.

We also need an Algorithm for finding the parastrophe operation $q_\backslash$ of q that will be used in the decryption process. However, in general, finding the explicit polynomial form of $q_\backslash$ is both time and memory consuming process, since the parastrophe can be of any degree $deg, 2 \le deg \le d$. Instead of finding the explicit form of $q_\backslash$ and using it to evaluate $\mathbf{y} = q_\backslash(\mathbf{u}, \mathbf{v})$ for given $\mathbf{u}, \mathbf{v} \in \mathbb{F}_{p^k}$, $k \ge 1$, we will use the left quasigroup operation q to determine $\mathbf{y}$, based on the identity $q_\backslash(\mathbf{u}, \mathbf{v}) = \mathbf{y} \Leftrightarrow q(\mathbf{u}, \mathbf{y}) = \mathbf{v}$. In other words, we reduce the problem of evaluating $q_\backslash$, to solving the system of d quadratic equations in d variables $y_1, y_2, \dots, y_d$ over $\mathbb{F}_{p^k}$

$$q(\mathbf{u}, \mathbf{y}) = \mathbf{v} \tag{2}$$

Even though this is also a non trivial problem in general, the specific structure of the LMQQs in use, allows this system to be solved in polynomial time, very efficiently and fast. We have the following algorithm for solving (2).

Algorithm: $\mathbf{Q}_\backslash(\mathbf{u}, \mathbf{v}, d, p, k, q_0, \mathbf{D}^{-1}, \mathbf{D_y}^{-1}, \mathbf{c}, \mathbf{c_y})$

Input $d, p, k \in \mathbb{N}$, where p is prime

1. Compute $\mathbf{v}_1 = \mathbf{D}^{-1} \cdot (\mathbf{v} - \mathbf{c})$.
2. Solve the system $q_0(\mathbf{u}, \mathbf{y}_1) = \mathbf{v}_1$ for the unknown $\mathbf{y}_1$.
3. Compute $\mathbf{y} = \mathbf{D_y}^{-1} \cdot (\mathbf{y}_1 - \mathbf{c_y})$.

Output $\mathbf{y}$ as solution to (2).

3.3 The Affine mappings S and T

A standard part of any MQ system are two affine mappings S and T that are usually generated at random from the class of all affine mappings on n variables over the underlying field $\mathbb{F}_{p^k}$. However, the affine mappings S and T for MQQ-ENC are constructed in a special way that reduces the storage space and furthermore significantly speeds up the decryption process. S and T are constructed using a combination of two circulant matrices, an idea already implemented in MQQ-SIG but over the prime field $GF(2)$ [32]. Similar ideas for reducing the private key have also been applied in [51, 57, 61].

For the purpose of MQQ-ENC, we modified the procedure for creating S and T given in [32] in several ways, in order to be appropriate for a bigger field, but also to fit the security requirements of MQQ-ENC. The detailed construction over any $\mathbb{F}_{p^k}$ is given trough the next algorithm.

Algorithm: CreateST(n, p, k, r_1, r_2, rem)

Input $n, p, k, r_1, r_2, rem \in \mathbb{N}$, where p is prime.

1. Generate at random two permutations σ_1 and σ_2 on the set $\{1,\ldots,n\}$. Create the permutation matrices $\mathbf{P}_{\sigma_1}$ and $\mathbf{P}_{\sigma_2}$ over $\mathbb{F}_{p^k}$.
2. Create the permutation matrices $\mathbf{P}_{\rho_i^{(1)}}$ and $\mathbf{P}_{\rho_j^{(2)}}$ over $\mathbb{F}_{p^k}$, for all $i \in \{0,\ldots,r_1\}, j \in \{0,\ldots,r_2\}$, where $\rho_i^{(1)}, \rho_j^{(2)}$ are rotations defined for all $l \in \{1,\ldots,n\}$ by:
$$\rho_i^{(1)}(l) = (l + i\left\lfloor \frac{n}{r_1} \right\rfloor - 1) \pmod{n} - 1, \quad \rho_j^{(2)}(l) = (l + j\left\lfloor \frac{n}{r_2} \right\rfloor - 1) \pmod{n} - 1.$$
3. Generate at random $rem \times n$ matrix $\mathbf{M}_0 = \left[m_{i,j}^{(0)}\right]_{rem \times n}$ over the set $\{0,1\} \subset \mathbb{F}_{p^k}$, such that all the columns are nonzero.
4. For a fixed ordering of $\mathbb{F}_{p^k}\backslash\{0\}$ create a repetitive array $(\alpha_i)_n$ of length n of the elements of $\mathbb{F}_{p^k}\backslash\{0\}$. Create a matrix $\mathbf{M} = \left[\alpha_{\sigma_1(j)} \cdot m_{i,j}^{(0)}\right]_{rem \times n}$. Create a matrix $\mathbf{I_M}$ by replacing the last rem rows of the identity matrix $\mathbf{I}_n$ by $\mathbf{M}$.
5. Generate at random from $\mathbb{F}_{p^k}\backslash\{0\}$ two arrays $(a_i^{(1)})_{r_1+1}$ $(a_i^{(2)})_{r_2+1}$ of lengths r_1+1 and r_2+1 respectively. Compute the matrices
$$\mathbf{S}'_{inv} = \sum_{i=0}^{r_1} a_i^{(1)} \cdot \mathbf{P}_{\rho_i^{(1)}} \cdot \mathbf{P}_{\sigma_2} + \sum_{j=0}^{r_2} a_j^{(2)} \cdot \mathbf{P}_{\rho_j^{(2)}} \cdot \mathbf{P}_{\sigma_1}, \quad \mathbf{T}'_{inv} = \sum_{i=0}^{r_1} a_i^{(1)} \cdot \mathbf{P}_{\rho_i^{(1)}} \cdot \mathbf{P}_{\sigma_1} \cdot \mathbf{I_M} + \sum_{j=0}^{r_2} a_j^{(2)} \cdot \mathbf{P}_{\rho_j^{(2)}} \cdot \mathbf{P}_{\sigma_2} \cdot \mathbf{I_M}$$
6. Let $\mathbf{SubT}'_{inv}$ be the $rem \times n$ matrix of the last rem rows of $\mathbf{T}'_{inv}$. If $\mathbf{SubT}'_{inv}$ has a zero column or $det(\mathbf{T}'_{inv}) = 0$ or $det(\mathbf{S}'_{inv}) = 0$, **then** go to Step 1, **else** create the matrices $\mathbf{S} = ((\mathbf{S}'_{inv})^\top)^{-1}$ and $\mathbf{T} = ((\mathbf{T}'_{inv})^\top)^{-1}$, and the column vector $\mathbf{v}_s = (\alpha_{\sigma_1(1)} \cdot \alpha_{\sigma_2(1)}, \ldots, \alpha_{\sigma_1(n)} \cdot \alpha_{\sigma_2(n)})^\top$.
7. Let $S(\mathbf{x}) = \mathbf{S} \cdot \mathbf{x} + \mathbf{v}_s$, and $T(\mathbf{x}) = \mathbf{T} \cdot \mathbf{x}$.

Output the quintet $(\sigma_1, \sigma_2, \mathbf{M}_0, (a_i^{(1)})_{r_1+1}, (a_i^{(2)})_{r_2+1})$ and the mappings S, T.

---

4 Description of MQQ-ENC

Let $n \in \mathbb{N}$ be divisible by 8. For the purpose of describing MQQ-ENC, we will use the algorithms given in the previous section. Note that, although they are valid for any characteristic p of the field, in the rest of the paper we will focus on the case when $p = 2$. Similar arguments can be given for any $\mathbb{F}_{p^k}$.

The MQQ-ENC cryptosystem is defined as a triplet of probabilistic algorithms MQQ-ENC= $(\mathcal{G}^{MQQ}, \mathcal{E}^{MQQ}, \mathcal{D}^{MQQ})$, associated to a message space $Mspace(nk) = \{0,1\}^{nk/2}$, and random coins $Coins(nk) = \{0,1\}^{nk/4}$, such that:

Key-Generation algorithm $\mathcal{G}^{MQQ}$

Input: 1^{nk},

1. Run **CreateST**$(n, 2, k, r_1, r_2, rem)$ to obtain $(\sigma_1, \sigma_2, \mathbf{M}_0, (a_i^{(1)})_{r_1+1}, (a_i^{(2)})_{r_2+1})$ and the affine mappings S and T.
2. Run **CreateLMQQ**$(8, 2, k)$ to obtain $(q_0, \mathbf{D}, \mathbf{D_y}, \mathbf{c}, \mathbf{c_y})$ and q.
3. Represent the vector $(x_1, x_2, \ldots, x_n)$ of variables over $\mathbb{F}_{2^k}$ as a vector $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n/8})$ of variables over $\mathbb{F}_{2^k}^8$, where $\mathbf{x}_i = (x_{8i-7}, x_{8i-6}, \ldots, x_{8i})$.
4. Define a mapping $P' : \mathbb{F}_{2^k}^n \to \mathbb{F}_{2^k}^n$ (a quasigroup string transformation) by:
$$(y_1, \ldots, y_n) = P'(x_1, \ldots, x_n) \Leftrightarrow (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{n/8}) = (q(11\ldots1, \mathbf{x}_1), q(\mathbf{x}_1, \mathbf{x}_2), \ldots, q(\mathbf{x}_{n/8-1}, \mathbf{x}_{n/8})) \qquad (3)$$
5. Construct the mapping $P_{full} : \mathbb{F}_{2^k}^n \to \mathbb{F}_{2^k}^n$ as $P_{full} = T \circ P' \circ S$. We use the notation $P_{full} = (p_1, p_2, \ldots, p_n)$, where $p_i(x_1, \ldots, x_n)$, $1 \le i \le n$.
6. The vector of polynomials $P : \mathbb{F}_{2^k}^n \to \mathbb{F}_{2^k}^{n-rem}$ is obtained by removing the last rem coordinates from P_{full}, i.e. $P = (p_1, p_2, \ldots, p_{n-rem})$.
7. Choose a universal hash function $H : \{0,1\}^{3nk/4} \to \{0,1\}^{nk/4}$.
8. Set $\mathsf{pk} = (P, H)$, and $\mathsf{sk} = (\sigma_1, \sigma_2, \mathbf{M}_0, (a_i^{(1)})_{r_1+1}, (a_i^{(2)})_{r_2+1}, q_0, \mathbf{D}^{-1}, \mathbf{D}_\mathbf{y}^{-1}, \mathbf{c}, \mathbf{c_y})$.

Output: Public private key pair $(\mathsf{pk}, \mathsf{sk})$.

Encryption algorithm $\mathcal{E}^{MQQ}$

Input: Public key $\mathsf{pk} = (P, H)$ and plaintext message $m = \{m_1, m_2, \ldots, m_{n/2}\} \in Mspace(nk)$,

1. Generate a random string $r = \{r_1, r_2, \dots, r_{n/4}\} \in Coins(nk)$.
2. Evaluate $H(m,r) = H(m_1, m_2, \dots, m_{n/2}, r_1, r_2, \dots, r_{n/4})$. Let $H(m,r) = (h_1, h_2, \dots, h_{n/4})$.
3. Evaluate $P(m, r, H(m,r)) = P(m_1, \dots, m_{n/2}, r_1, \dots, r_{n/4}, h_1, \dots, h_{n/4})$.

Output: Ciphertext $c = P(m, r, H(m,r))$.

Decryption algorithm $\mathcal{D}^{MQQ}$

Input: Private key $\mathsf{sk} = (\sigma_1, \sigma_2, \mathbf{M}_0, (a_i^{(1)})_{r_1+1}, (a_i^{(2)})_{r_2+1}, q_0, \mathbf{D}^{-1}, \mathbf{D_y}^{-1}, \mathbf{c}, \mathbf{c_y})$ and cipher $c = (c_1, \dots, c_{n-rem}) \in \mathbb{F}_{2^k}^{n-rem}$,

For all $(c_{n-rem+1}, c_{n-rem+2}, \dots, c_n) \in \mathbb{F}_{2^k}^{rem}$ **do**

1. Evaluate $(m'_1, m'_2, \dots, m'_n) = S^{-1} \circ P'^{-1} \circ T^{-1}(c_1, c_2, \dots, c_n)$, where P'^{-1} is evaluated by:
$$(u_1, u_2, \dots, u_n) = P'^{-1}(v_1, v_2, \dots, v_n) \Leftrightarrow$$
$$(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{n/8}) = (q_\backslash(\mathbf{u}_0, \mathbf{v}_1), q_\backslash(\mathbf{u}_1, \mathbf{v}_2), q_\backslash(\mathbf{u}_2, \mathbf{v}_3), \dots, q_\backslash(\mathbf{u}_{n/8-1}, \mathbf{v}_{n/8})). \quad (4)$$
Here, $\mathbf{u}_0 = (11\dots1)$, and for every $i \in \{0, \dots, n/8-1\}$, $\mathbf{u}_{i+1} = q_\backslash(\mathbf{u}_i, \mathbf{v}_{i+1})$ is evaluated by running the Algorithm $\mathbf{Q}_\backslash(\mathbf{u}_i, \mathbf{v}_{i+1}, 8, 2, k, q_0, \mathbf{D}^{-1}, \mathbf{D_y}^{-1}, \mathbf{c}, \mathbf{c_y})$.
The vector $(u_1, \dots, u_n)$ over $\mathbb{F}_{2^k}$ is represented as a vector $(\mathbf{u}_1, \dots, \mathbf{u}_{n/8})$ over $\mathbb{F}_{2^k}^8$, where $\mathbf{u}_i = (u_{8i-7}, u_{8i-6}, \dots, u_{8i})$.
Analogously, the same is done for the vector $(v_1, v_2, \dots, v_n)$.
2. **If** $H(m'_1, m'_2, \dots, m'_{3n/4}) = (m'_{3n/4+1}, m'_{3n/4+2}, \dots, m'_n)$ **then break**;

End for;

Output: Plaintext m' or $\perp$ if the above test failed for all $(c_{n-rem+1}, \dots, c_n) \in \mathbb{F}_{2^k}^{rem}$.

Remark 1. An algorithm for fast evaluation of T^{-1} and S^{-1} will be provided in an extended version of the paper.

Theorem 2. *MQQ-ENC$= (\mathcal{G}^{MQQ}, \mathcal{E}^{MQQ}, \mathcal{D}^{MQQ})$ is all-keys $(1 - \frac{1}{2^{nk/4-rem\cdot k+1}})$-correct public key encryption scheme.*

The proof is given in Appendix A.

5 Security of the trapdoor function of MQQ-ENC

In this section we analyze the security of the trapdoor function of MQQ-ENC, i.e. of the system of polynomials P. We consider the classical MQ cryptanalytic techniques. Based on this analysis we recommend parameters for practical implementation that offer a security level of $\mathcal{O}(2^{128})$.

5.1 Direct algebraic attack

In MQ public key cryptosystems the standard inversion (or message recovery) attack reduces to solving the system $P(x_1, x_2, \dots, x_n) = (c_1, c_2, \dots, c_m)$ of quadratic equations over $\mathbb{F}_{p^k}$ for a given ciphertext $(c_1, c_2, \dots, c_m)$. Today, the most powerful method for solving algebraic systems is to compute the Gröbner basis of the system by the Faugère's algorithms F_4 [22] or F_5 [23] (or by variants of XL [12] and MutantXL [16] algorithms).

A good measure of the complexity of computing a Gröbner basis of an ideal I is the *degree of regularity* (D_{reg}), which is the maximal degree of the polynomials appearing during the computation (see for ex. [1]). In fact the complexity is polynomial in D_{reg} and is $\mathcal{O}\left(\binom{n+D_{reg}}{n}^\omega\right)$ where ω is the linear algebra constant. For randomly generated systems of equations, the behavior of D_{reg} as a function of n is quite well understood [1, 2]. For systems that involve a particular algorithm and structure, this behavior is different and hard to predict. Thus, en exhaustive experimental analysis can be a good indicator of the nature of D_{reg}.

For completeness and justification of the design of MQQ-ENC, it is important to note that in the signature scheme MQQ-SIG [32], resistance against Gröbner basis attack was achieved by applying the minus modifier. The experimental results in [32] for D_{reg} showed that it behaves in the same way as for a randomly generated system. We note that the modification applied in MQQ-SIG is suitable only for a signature scheme, since half of the polynomials of the public system are removed. In the case of an encryption scheme, only a small number of polynomials can be removed, otherwise the decryption becomes inefficient or even infeasible.

MQQ-ENC has many similarities with MQQ-SIG, mainly in the overall design of the internal transformation. Also the order of the MQQs used is the same. The main difference, is that, in MQQ-ENC, we use left MQQs (Only the left translation is bijective), whereas, in MQQ-SIG, MQQs of bilinear nature were used. In the case of MQQ-SIG, the bilinear MQQs do not pose a threat to the scheme. Our analysis, however, showed that much more superior for use in MQQ-ENC are the left MQQs constructed using Algorithm **CreateLMQQ**. In Appendix B, we give a comparison of the experimental results of a Gröbner basis cryptanalysis on the system P of MQQ-ENC as described in the paper (i.e. using left MQQs) and a system the same as MQQ-ENC but using bilinear MQQs.

The experiments of Gröbner basis cryptanalysis on MQQ-ENC were done with Magma 2.17-3's implementation [42] of the F_4 [22] algorithm on a workstation with 32 cores based on Intel Xeon 2.27GHz, with 1TB of RAM memory. They can be summarized in the following two groups:

Gröbner basis cryptanalysis on MQQ-ENC defined over $\mathbb{F}_2$ These experiments are our referent ones for investigating the immunity of MQQ-ENC against inversion attack. This is mainly because the case of $\mathbb{F}_2$ scales good, and more experiments for different small values of n can be made. We performed 100 experiments for each of the cases of 24, 32, 40 and 48 variables, but because of the great time complexity, we were able to have only 10 experiments finished for each of the cases of 56 and 64 variables. In the experiments we took that the number of removed equations rem is 8. This value was chosen as a trade off between efficiency of the decryption and security against direct algebraic attack. The results of the experiments are given in Table 1.

n	$D_{reg,rand}$	$D_{reg,\mathrm{MQQ-ENC}}$	Time(sec)	Memory(MB)
24	5	4.27	0.13	22.56
32	6	4.98	55.66	811.00
40	6	5.06	2 058.76	13 755.86
48	7	5.67	21 981.07	89 987.15
56	8	7.50	128 644.55	308 728.40
64	9	8.10	771 861.06	1 372 192.13

Table 1. The average Degree of regularity, Time and Memory complexity observed in solving MQQ-ENC system of $n-8$ variables over $\mathbb{F}_2$. $D_{reg,rand}$ is the expected degree of regularity of a random system of $n-8$ variables

We note that, since there are more variables than polynomials, in all our experiments we used the standard technique to first fix $rem = 8$ variables to randomly chosen elements of $\mathbb{F}_2$, and then solve the system. Thus the results given in Table 1, actually show the time and memory needed to solve such a system of $n-8$ variables. Also, $D_{reg,rand}$ is the expected degree of regularity for a randomly generated system of size $n - rem = n - 8$.

The results of Table 1 show that although the degree of regularity $D_{reg,\mathrm{MQQ-ENC}}$ of MQQ-ENC does not reach the value of a random system, the difference is approximately constant for all investigated n. This indicates that it grows together with the $D_{reg,rand}$ of a random system, and that the two are approximately linearly dependent. Thus we may conjecture the following relation:

$$D_{reg,\mathrm{MQQ-ENC}} \approx D_{reg,rand} - C, \quad \text{for a small constant } 0 \leq C \leq 2. \tag{5}$$

The results also show expected exponential growth of the time and space complexity.

Gröbner basis cryptanalysis on MQQ-ENC defined over $\mathbb{F}_{2^k}$ In this more general case, making experiments was only possible for $k = 2$ and only for 24 and 32 variables. The number of removed equations had to be scaled too, and the value of $rem = 4$ was chosen, so that the load for the decryption would be the same as in the case of $\mathbb{F}_2$. As expected, this led to less time needed than for the appropriate values of n over $\mathbb{F}_2$, i.e. 48 and 64. Actually, the times over $\mathbb{F}_{2^2}$ were approximately two times smaller than those over $\mathbb{F}_2$. Although, we have too little information to predict what will happen over bigger fields, a plausible conjecture is the following:

$$TimeGr\ddot{o}bner(\mathbb{F}_2, n) \approx 2^k \cdot TimeGr\ddot{o}bner(\mathbb{F}_{2^k}, \frac{n}{k}) \tag{6}$$

We further performed experiments with less removed variables, i.e. 1, 2 and 3, to be able to predict the trend for different (bigger) values of rem. The number of experiments performed is 100 for all values given, except for the case of $n = 32$, for 3 removed equations (29 experiments) and 4 removed equations (6 experiments). The results showed that when more variables are removed even though the system is of smaller size, the difficulty of solving it raises substantially. More than 4 removed equations will probably create even a bigger difficulty, however this will reduce the decryption speed as well. The results are given in Table 2.

	$n = 24$		$n = 32$		$n = 24$, S, T random	
rem	Time (sec)	Memory (MB)	Time (sec)	Memory (MB)	Time (sec)	Memory (MB)
1	3.45	37.04	172.44	574.08	3.68	28.79
2	57.86	68.61	4 392.05	1 447.01	47.79	74.14
3	1 429.68	381.60	238 265.45	9 061.32	1 361.56	369.84
4	11 952.02	2 616.65	378 470.09	12 083.12	11 412.38	2 286.47

Table 2. The average Time and Memory complexity of solving MQQ-ENC system of $n - rem$ variables over $\mathbb{F}_4$.

Another important type of experiments that we performed was to investigate the impact of the specially constructed affine mappings S and T. For the case of $n = 24$, $r_1 = 10$, $r_2 = 13$ and $rem \in \{1, 2, 3, 4\}$ we compared the results for randomly generated affine mappings, and the ones used in MQQ-ENC. The comparison (given in Table 2) clearly shows that there is no difference regarding direct algebraic attack between the two cases. This furthermore justifies the use of the designed mappings.

5.2 Rank attacks on Stepwise Triangular Systems

Rank attacks are types of structural attacks that are very powerful against the family of triangular MQ schemes. A rank attack has first been successfully applied against the Birational Permutation scheme [10], defined over a large finite ring [58]. Afterwards, the idea was translated into finite fields in [34] where is was used to break the TTM scheme [44], as well as the more general TPM scheme, introduced in the paper. Another generalization was made in [63], where two general attacks were described for the so called Stepwise Triangular Systems (STS). The private key of a STS system consists of invertible affine mappings S and T over $\mathbb{F}_{p^k}^n$ and $\mathbb{F}_{p^k}^m$ respectively, and the internal quadratic mapping $P' : \mathbb{F}_{p^k}^n \to \mathbb{F}_{p^k}^m$ has a stepwise triangular structure: For each layer $l \in \{1, \dots, L\}$ the polynomials of the current layer contain only the variables from the previous and the current layer. The number of new variables and polynomials in each layer can be different.

The key observation for the rank attacks is that

$$\sum_{j=1}^{m} \tau_{ij}\hat{\mathbf{P}}_i = \mathbf{S}\hat{\mathbf{P}}'_i\mathbf{S}^\top \text{ and thus } Rank(\sum_{j=1}^{m} \tau_{ij}\hat{\mathbf{P}}_i) = Rank(\mathbf{S}\hat{\mathbf{P}}'_i\mathbf{S}^\top), \tag{7}$$

where $\hat{\mathbf{P}}_i$ and $\hat{\mathbf{P}}'_i$ are the matrices of the homogeneous part of the public P_i and private P'_i polynomial respectively (for a description of how this matrix is formed, see for ex. [63]), and $\mathbf{S}$ and $\mathbf{T}^{-1} = [\tau_{ij}]$ are the invertible matrices of the mappings S and T^{-1}. Now, the matrix $\mathbf{T}^{-1}$ can be retrieved (or an equivalent one) by finding linear combinations of the public key matrices $\hat{\mathbf{P}}_i$. However, the following two conditions have to be met.

First of all, the kernels of the matrices $\hat{\mathbf{P}}'_i$ must form a chain of kernels $ker'_L \subset \cdots \subset ker'_1$, where ker'_l is the common kernel for the matrices from the layer l.

If this is not true, and for example, all or most of the layers have kernels of the same dimension, it is not clear how new vectors from each subsequent kernel can be found. Note that, this inapplicability of the attack for such schemes was mentioned in [63] for the case of the Enhanced TTS [62]. In [62], the attack was adapted for some very special types of tame-like systems without chain of kernels, but this adaptation strongly depends on the tame structure of the schemes, and is still not applicable in the more general case.

Second, in order to use the rank equation (7), the mapping $\mathbf{T}$ must be invertible. Otherwise, (7) doesn't hold at all. For example, the simplest case is when the public key is formed from a bijective transformation by removing some of the polynomials. Then, a linear combination of the remaining polynomials can have a very different and hard to predict rank compared to a linear combination of the full system. Note that in [10] a successful rank attack was mounted against the Birational Permutation scheme even when one polynomial from the public key had been removed (as a countermeasure). However, the attack works only for the special structure of this scheme, and highly complicates even for that scheme when more than one equation is removed. A similar approach is not effective when more polynomials are removed, or when the structure of one removed polynomial is more general. Interestingly, the general TPM [34] and the general STS [63] both assume that $\mathbf{T}$ is always a bijection ([34] considers "-" modifier, but the equations are removed from the internal P' and not from P). We should note that recently, in [59], the layered structure and bijective S and T were exploited to mount algebraic key recovery attack on Enhanced STS [60] Enhanced TTS [62], and Rainbow schemes [17] by finding simpler equivalent keys.

The MQQ-ENC scheme has some similarities with STS schemes, but also has some essential differences. Indeed, each of the quasigroup evaluations can be considered as a step or a layer. From (3) we have that $\mathbf{y}_1 = q(11\ldots1, \mathbf{x}_1)$ and $\mathbf{y}_i = q(\mathbf{x}_{i-1}, \mathbf{x}_i)$, for $i \in \{2, \ldots, n/8\}$. Thus each of the P' polynomials depends on at most 16 variables. This structure allows a fast decryption, similarly as in the case of STS, but unlike STS, there is no chain of kernels ($\mathbf{y}_i$ does not depend on $\mathbf{x}_1, \ldots, \mathbf{x}_{i-2}$).

Also, as *rem* polynomials of the public P are removed, the second condition is also not satisfied. Even more, the matrix $\mathbf{T}^{-1}$ is of special structure, that insures that there isn't a single linear combination of the remaining $n - rem$ polynomials for which (7) holds. Namely, the matrix formed by the last *rem* columns of $\mathbf{T}^{-1}$ doesn't have a zero row.

As a result of these properties of MQQ-ENC, and based on the previous discussion, we can conclude that a rank attack as known so far, can not be mounted on MQQ-ENC.

5.3 Rank attacks on mixed field systems

Rank attacks were also proven to be powerful in the cryptanalysis of the family of "mixed field" schemes [39, 6, 7]. In the case of MQQ-ENC, which has a completely different nature, both the shape of the internal secret transformation P' and its degree are not known. Furthermore, when a minus modifier is applied, the attacks [6, 7] are possible only for MultiHFE and not for HFE, since it works only when the number of variables in the extension field is strictly bigger than the removed equations (in HFE the number of variables in the big field is 1). This makes it very unsuitable for MQQ-ENC, even if the public and secret keys are represented as univariate polynomials over the big field.

5.4 Differential attacks

In [26], Fouque et al. introduced a new technique in multivariate cryptanalysis, based on differentials. They used it to successfully break the perturbed version of the MI scheme PMI [15].

The differential of a quadratic mapping $\mathbf{G} : \mathbb{F}_{p^k}^n \to \mathbb{F}_{p^k}^m$ over a finite field $\mathbb{F}_{p^k}$ at a point $\mathbf{a} \in \mathbb{F}_{p^k}^n$ is the linear mapping defined by $\mathbf{D_{G,a}}(\mathbf{x}) = \mathbf{G}(\mathbf{x}+\mathbf{a}) - \mathbf{G}(\mathbf{x}) - \mathbf{G}(\mathbf{a}) + \mathbf{G}(\mathbf{0})$.

The idea behind the attack is to study the distribution of the rank of the differential, and deduce information about the internal structure. This was also used by Dubois et al. first to create an efficient distinguisher for HFE in [19], and later to extend the idea for attacking the perturbed version of HFE in [20]. In both attacks, the authors used the fact that the internal polynomial of HFE in the extension field is of known, small degree. The mentioned technique, was again applied to break all the versions of the NESSIE proposal SFLASH [48] in [18]. Here, the special structure of the MI scheme was proven to be revealing enough even when a "minus" modifier has been applied, a strengthening usually considered to protect against rank attacks.

From the published work so far that uses the technique of differential cryptanalysis in MQ schemes, it is clear that the method is especially suitable for attacking schemes from the MI family of MQ systems. Our scheme MQQ-ENC is not in this family. The polynomial of the internal mapping over the extension field is not known and does not need to be of small degree. This makes the attack not applicable to MQQ-ENC.

5.5 MQQ-ENC recommended parameters for security level of $\mathcal{O}(2^{128})$

We find that the appropriate finite fields for defining MQQ-ENC are $\mathbb{F}_{2^k}$ where $k \in \{1, 2, 4, 8\}$, mainly because of natural suitability with operations and register sizes on standard CPUs. For these underlying fields, our recommended values for the parameters $n \cdot k$ and rem are $n \cdot k \geq 256$ and $8 \leq rem \cdot k \leq 16$. Also, for r_1 and r_2 used in the creation of the mappings S and T we recommend the values $r_1 = 10$ and $r_2 = 13$. We chose these parameters based on the arguments and experiments for the direct algebraic attack, since, as argumented in the previous subsections, it is our opinion that the other described attacks in their usual form can not be efficiently mounted against MQQ-ENC.

We should note that for bigger values of $n \cdot k$ and $rem \cdot k$ the security will be better, but the efficiency will be reduced. Thus, in essence, the parameters for practical implementation are $n \cdot k = 256$ and $rem \cdot k = 8$. In this case, the message space is $Mspace(nk) = \{0,1\}^{nk/2} = \{0,1\}^{128}$ i.e. MQQ-ENC encrypts messages m of length 128 bits using random coins r of length $\frac{nk}{4} = 64$ bits. The output of the universal hash function $H(m, r)$, used for reducing decryption errors, is also 64 bits.

For these parameters, and based on the conjectured relations (5) and (6) given in Section 5.1, we have the following complexity estimate.

Proposition 1. *Under the assumptions (5) and (6), the complexity of finding a Gröbner basis of MQQ-ENC over $\mathbb{F}_{2^k}$ where $k \in \{1, 2, 4, 8\}$, with the recommended parameters $n \cdot k = 256$ and $rem \cdot k = 8$ is $\mathcal{O}(2^{235-k})$.*

The proof is given in Appendix A.

In the decision of the parameters for MQQ-ENC, we took into account the recent work of Bardet et al. [3] and their algorithm BooleanSolve for solving MQ systems over $\mathbb{F}_2$, which is a variant of the hybrid approach [8]. Thus we have the following:

Proposition 2. *Under the assumption (5), the complexity of* BooleanSolve *for solving MQQ-ENC over* $\mathbb{F}_2$ *implemented with the recommended parameters is* $\mathcal{O}(2^{179})$.

The proof is given in Appendix A.

The estimated complexities for solving an MQ system arising from MQQ-ENC from Propositions 1 and 2 for the recommended parameters of MQQ-ENC are well over $\mathcal{O}(2^{128})$. Nevertheless, we claim a security level of $\mathcal{O}(2^{128})$ for the recommended parameters of MQQ-ENC. This may seem as too conservative or an overdefinition. However, *it is always a good practice to have a security margin for a cryptographic scheme, since the "attacks always get better, they never get worse".*

6 Security of the public key encryption scheme MQQ-ENC

6.1 Security assumption

Recall that the MQ-problem consisted of finding a solution to a system of polynomials randomly chosen from the set $\mathcal{P}_{MQ}$ of all systems of m polynomials in n variables over a finite field $\mathbb{F}_{p^k}$. However, if a system is drawn randomly from a different set, the MQ-problem does not have to be hard, as the successful cryptanalysis of many previous proposals has shown.

In the previous sections we made a thorough analysis of the underlying trapdoor function P of MQQ-ENC. Based on it we can conclude that for the right parameters, the currently known cryptanalytic techniques can not successfully invert P, nor find the included trapdoor. Thus, we can pose the following plausible assumption:

MQQ-assumption: *Let* n *be sufficiently large,* $1 \leq k \leq 8$ *and* $8 \leq rem \cdot k \leq 16$. *Let* $\mathcal{P}_{MQQ}$ *be the set of all public systems* P *that can be obtained using the algorithm* $\mathcal{G}^{MQQ}(1^{nk})$.
The MQ-problem is hard for a randomly chosen P *from* $\mathcal{P}_{MQQ}$.

A direct implication of this assumption is that MQQ-ENC is secure in the weak sense of OWE.

Theorem 3. *The public key encryption scheme MQQ-ENC is* OWE*-secure under the MQQ-assumption.* □

6.2 An IND – CCA secure encryption scheme from MQQ-ENC

In the known literature there are many generic proposals for converting a given PKE scheme, that satisfies weaker notions of security, to another that is secure in the sense of adaptive IND–CCA in the random oracle model. Some of them require stronger security assumptions such as IND–CPA security ([27]), but some provide the highest level of security even for PKE schemes that are secure in the weakest sense of OWE ([5, 28, 40, 49, 52]).

However, most of these conversions are proven to supply the intended level of security only in the case of perfect decryption. As it turns out, when a PKE scheme is susceptible to errors, the security proofs of these conversions might fail. This is best illustrated by the attack to the NTRU scheme [53], where the imperfect decryption of NTRU is exploited to totally break various enhancements of the scheme, that are otherwise sound in the perfect decryption scenario.

To our knowledge, the problem of securing an error prone scheme has been addressed in two papers [21] and [13]. In [21] the authors propose a new conversion, especially suitable for such schemes, and in [13] it is argumented that 3-round OAEP [49] (and a new probabilistic version of it) can actually be used in the error prone environment. Both conversions do not aim to convert the scheme into one that has perfect decryption but rather to make it very hard for the attacker to find situations where decryption errors occur and to exploit them to break the system. Also both conversions require the probability of decryption error to be negligible.

In [40] a special conversion for achieving adaptive IND–CCA security of the McEliece cryptosystem [43] was proposed under the assumption that the McEliece system is OWE. This conversion is especially efficient and suitable for the McEliece system compared to other generic transformations. Interestingly enough, we find it especially suitable for the MQQ-ENC as well. Even more, it can immunize against decryption errors, and make it infeasible for the attacker to try to exploit them. In essence, the arguments that this is true are in the same line as those given in [21].

We first give a short description of the Kobara-Imai conversion [40], adapted for our parameters.

Conversion K-I(MQQ-ENC):

- Let the **Key-Generation algorithm** $\mathcal{G}^{MQQ}_{K-I}$ be the same as $\mathcal{G}^{MQQ}$.
- Let $H_1 : \{0,1\}^{nk} \to \{0,1\}^{nk/4}$ and $G : \{0,1\}^{nk/4} \to \{0,1\}^{nk}$ be two pseudo random functions.
- The **Encryption algorithm** $\mathcal{E}^{MQQ}_{K-I}$ works as follows:
  1. For message $m \in Mspace(nk)$, and random $r \in \{0,1\}^{nk/2}$, let $z = H_1(r||m)$.
  2. Let $y = G(z) \oplus (r||m)$. Represent $y = y_1||y_2$, where $y_1, y_2 \in \{0,1\}^{nk/2}$.

  Output $c = (c_1, c_2) = (\mathcal{E}^{MQQ}(y_1, z), y_2)$ as the ciphertext.
- The **Decryption algorithm** $\mathcal{D}^{MQQ}_{K-I}$ works as follows:
  1. For given ciphertext $c = (c_1, c_2)$, let $(y_1, z) = \mathcal{D}'^{MQQ}(c_1)$, where $\mathcal{D}'^{MQQ}$ is the same as $\mathcal{D}^{MQQ}$ except that it outputs the first $3nk/4$ bits instead of the first $nk/2$ bits, obtained during the procedure. If c_1 is not a valid ciphertext for $\mathcal{D}^{MQQ}$ break and output $\perp$.
  2. Let $r||m = G(z) \oplus (y_1||c_2)$.

  If $H_1(r||m) = z$ **output** m as the plaintext, otherwise output $\perp$.

Proposition 3. *A standard adversary in an* IND–CCA *indistinguishability game against K-I(MQQ-ENC) that has access to the random oracles H_1 and G can find a decryption error with probability at most:*

$$P_{error} \leq q_{H_1}/2^{nk/4-rem\cdot k+1} + q_{H_1}q_G/2^{nk/4}$$

where q_{H_1} and q_G are the total number of calls to the oracles H_1 and G, respectively.

The proof is given in Appendix A.

The arguments that the Kobara-Imai conversion of MQQ-ENC provides indistinguishability in the sense of IND–CCA are the same as in the case of the McEliece system [40] for which it was originally proposed. As they can easily be translated in our case, and since we have proven that the adversary can not make use of the erroneous decryption we have the following theorem.

Theorem 4. *The public key encryption scheme K-I(MQQ-ENC) is* IND–CCA *secure under the MQQ-assumption.* □

7 Operating characteristics of MQQ-ENC

The public key $\mathsf{pk} = (P, H)$ of MQQ-ENC consists of a system of $n - rem$ polynomials in n variables over $\mathbb{F}_{2^k}$, and a universal hash function H. We will not count the size of encoding the universal hash function as a part of the public key and it can be any short-output universal hash function like MMH [35], NH [9], *digest()* [46], GHASH or PolyQ [41]. Thus, the size of the public key of MQQ-ENC in bytes is given by the formula:

$$\text{MQQ-ENC\_PublicKeySize}(k,n,rem) = \begin{cases} \frac{1}{8}(n-rem)(1+\frac{n(n+1)}{2}), & \text{if } k=1, \\ \frac{k}{8}(n-rem)(1+\frac{n(n+3)}{2}), & \text{if } k>1 \end{cases}$$

The private key is $\mathsf{sk} = (\sigma_1, \sigma_2, \mathbf{M}_0, (a_i^{(1)})_{r_1+1}, (a_i^{(2)})_{r_2+1}, q_0, \mathbf{D}^{-1}, \mathbf{D}_{\mathbf{y}}^{-1}, \mathbf{c}, \mathbf{c_y})$. The first five terms are needed for the creation of the secret S and T and we need $2n + \frac{rem \cdot n}{8} + \frac{k}{8} \cdot (r_1 + r_2 + 2)$ bytes to store them for $k > 1$ and $2n + \frac{rem \cdot n}{8}$ for $k = 1$, since for $\mathbb{F}_2$ we don't need the arrays $(a_i^{(1)})_{r_1+1}$ and $(a_i^{(2)})_{r_2+1}$. The last 5 terms are the building blocks of the secret LMQQ, and the needed memory in bytes is $\frac{1}{8}\left(\sum_{s=1}^{8}(1+8+\binom{8}{2}+8\cdot(8-s)+8-s+\binom{8-s}{2})+2\cdot(8^2+8)\right) = 93{,}5$ Bytes for $k=1$, and $\frac{k}{8}\left(\sum_{s=1}^{8}(2+2\cdot 8+\binom{8}{2}+8\cdot(8-s)+2(8-s)+\binom{8-s}{2})+2\cdot(8^2+8)\right) = 106k$ Bytes for $k>1$.

Table 3 shows the values of the size of the public and private key in bytes of the MQQ-ENC system for a message space $\{0,1\}^{128}$ and parameters: $k \in \{1,2,4,8\}$, $rem = 8/k$, $r_1 = 10$ and $r_2 = 13$.

The table also includes the performance of an initial C code performed on a 4 core Intel Nehalem i7 920X CPU architecture running at 2 GHz. Note that the execution of the code was on one core only, and because of the highly parallelizable nature of MQQ-ENC, almost linear speedups are possible if more cores are used. An optimization of the code will be one of our goals in the following period.

	k	n	rem	Public Key (Bytes)	Private Key (Bytes)	Key generation (cycles)	Encryption (cycles)	Decryption (cycles)
MQQ ENC	1	256	8	1 019 807	862	4 062 832 500	140 364	838 656
	2	128	4	259 935	539	1 041 751 923	93 576	645 120
	4	64	2	66 495	581	274 145 242	66 840	496 246
	8	32	1	17 391	941	74 093 308	51 415	381 728

Table 3. Operating characteristics of MQQ-ENC for encrypt./decrypt. of a message 128 bits long over different fields.

8 Conclusions

In this paper we proposed a new multivariate encryption scheme MQQ-ENC from the MQQ-family of public key schemes. The scheme is probabilistic encryption with negligible probability of decryption errors, that is achieved using a universal hash function. It is defined over the fields $\mathbb{F}_{2^k}$ for any $k \geq 1$, and can easily be extended to any $\mathbb{F}_{p^k}$, for prime p. MQQ-ENC shares the overall design of its predecessor MQQ-SIG, however, we introduce Left MQQs as a base for construction of the internal secret transformation. An extensive experimental analysis showed that LMQQs show much better characteristics compared to the previously used bilinear MQQs in the design of MQQ-SIG. We analyzed the applicability of the standard cryptanalytic techniques for MQ schemes. Based on the analysis, we conjectured that the instances of the MQQ-ENC trapdoor are hard instances with respect to the MQ problem. Under this assumption, we adapted the Kobara-Imai conversion of the McEliece PKE scheme for MQQ-ENC and showed that it immunizes against decryption errors and provides IND–CCA security. The operating characteristics of an unoptimized C-implementation are comparable to other multivariate schemes.

A part of our future work will be an optimized implementation of the algorithm, as well as a version where the redundancy now defined by the use of a universal hash function is replaced by some predetermined secret value. In that way we think that by a proper use of list-decoding techniques, MQQ-ENC can be turned into an efficient public-key block cipher.

References

1. M. Bardet, J.-C. Faugère, and B. Salvy. On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In Proc. of Internat. Conf. on Polynomial System Solving (ICPSS), pp. 71-75, 2004.
2. M. Bardet, J.-C. Faugère, B. Salvy, and B.-Y. Yang. Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. In Proc. of MEGA 2005, 8–th Internat. Symp. on Effective Methods in Alg. Geometry, 2005.
3. M. Bardet, J.-C. Faugère, B. Salvy and P.-J. Spaenlehauer. On the Complexity of Solving Quadratic Boolean Systems. arXiv:1112.6263, 2011, `http://arxiv.org/abs/1112.6263`
4. M. Bellare, A. Desai, D. Pointcheval, P. Rogaway, Relations among notions of security for public-key encryption schemes, in Advances in Cryptology - CRYPTO '98, LNCS, vol. 1462, pp. 26-45, Springer, 1998.
5. M. Bellare, P. Rogaway. Optimal Asymmetric Encryption. In Proc. EUROCRYPT '94, LNCS 950, pp. 92-111, 1995.
6. L. Bettale, J.-C. Faugère, L. Perret. Cryptanalysis of Multivariate and Odd-Characteristic HFE Variants. In Proc. PKC '11, pp. 441–458, 2011.
7. L. Bettale, J.-C. Faugère, L. Perret: Cryptanalysis of HFE, Multi-HFE and Variants for Odd and Even Characteristic. IACR Cryptology ePrint Archive 2011: 399 (2011)
8. L. Bettale, J.-C. Faugère, and L. Perret. Hybrid approach for solving multivariate systems over finite fields. Journal of Mathematical Cryptology, 3:177197, 2009.
9. J. Black, S. Halevi, H. Krawczyk, T. Krovetz, P. Rogaway. UMAC: Fast and Secure Message Authentication. CRYPTO, LNCS vol. 1666, pp. 216-233, 1999.
10. D. Coppersmith, J. Stern and S. Vaudenay. The Security of the Birational Permutation Signature Schemes, Journal of Cryptology, vol.10. pp. 207–221, 1997.
11. Courtois NT (2001) Effcient zero-knowledge authentication based on a linear algebra problem MinRank. In: Advances in Cryptology - ASIACRYPT 2001, Springer, LNCS, vol 2248, pp 402–421
12. N. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations. In Proc. EUROCRYPT '00, LNCS vol. 1807, pp. 392–407, 2000.
13. Y. Cui, K. Kobara, and H. Imai. On Achieving Chosen Ciphertext Security with Decryption Errors. AAECC 2006, LNCS 3857, pp. 173-182, Springer-Verlag 2006.
14. C. Diem. On the discrete logarithm problem in elliptic curves. *Compos. Math.*, 147(1):75–104, 2011.
15. J. Ding. A new variant of the Matsumoto-Imai Cryptosystem through Perturbation. In PKC '04, LNCS 2947, pp. 305-318. Springer-Verlag, 2004.
16. J. Ding, J. Buchmann, M. S. E. Mohamed, W. S. A. Moahmed, and R.-P. Weinmann. MutantXL. In Proc. of 1st international conference on Symbolic Computation and Cryptography (SCC08), pp. 16–22, Beijing, China, 2008.
17. J. Ding and D. Schmidt. Rainbow, a new multivariable polynomial signature scheme. In Conference on Applied Cryptography and Network Security ACNS 2005, LNCS vol. 3531, pp. 164-175, Springer, 2005.
18. V. Dubois, P.-A. Fouque, A. Shamir, and J. Stern. Practical Cryptanalysis of SFLASH. In Advances in Cryptology - CRYPTO '07, LNCS vol. 4622, pp. 1–12. Springer, 2007.
19. V. Dubois, L. Granboulan, and J. Stern. An Efficient Provable Distinguisher for HFE. In Proc. of ICALP (2), LNCS vol. 4052, pp. 156–167. Springer, 2006.
20. V. Dubois, L. Granboulan, and J. Stern. Cryptanalysis of HFE with Internal Perturbation. In Proc. of PKC '07, LNCS vol. 4450, pp. 249–265. Springer, 2007.
21. C. Dwork, M. Naor, and O. Reingold. Immunizing encryption schemes from decryption errors. In Proc. of EUROCRYPT '04, LNCS vol. 3027, pp. 342-360, 2004.
22. J.-C. Faugère. A new efficient algorithm for computing Gröbner basis (F4). Journal of Pure and Applied Algebra, 139:61-88, 1999.
23. J.-C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero F5. In Proc. of ISSAC, pp. 75-83. ACM Press, July 2002.
24. J.-C. Faugère, R.S. Ødegård, L. Perret, and D. Gligoroski. Analysis of the MQQ public key cryptosystem. In Proc. of *CANS*, LNCS vol. 6467, pp. 169–183. Springer, 2010.
25. J.-C. Faugère, L. Perret, C. Petit and G. Renault. Improving the Complexity of Index Calculus Algorithms in Elliptic Curves over Binary Fields, EUROCRYPT 2012, LNCS vol. 7237, pp. 27-44, 2012.
26. P.-A. Fouque, L. Granboulan, and J. Stern. Differential cryptanalysis for multivariate schemes. In Proc. of *Eurocrypt 2005*, LNCS, vol. 3494, pp. 341–353. Springer, 2005.
27. E. Fujisaki and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. In Proc. of PKC '99, LNCS 1560, pp. 53–68, 1999.
28. E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In Proc. of CRYPTO '99, LNCS 1666, pp. 535-554, 1999.
29. M. R. Garey and D. S. Johnson. Computers and Intractability - A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, 1979.

30. P. Gaudry. Index calculus for Abelian varieties of small dimension and the elliptic curve discrete logarithm problem. J. Symbolic Comput., 44(12):1690–1702, 2008.
31. D. Gligoroski, S. Markovski, and S. J. Knapskog. Multivariate quadratic trapdoor functions based on multivariate quadratic quasigroups. In *MATH'08: Proc. of the American Conference on Applied Mathematics*, pp. 44–49, 2008.
32. D. Gligoroski, R. S. Ødegård, R. E. Jensen, L. Perret, J.-C. Faugère, S. J. Knapskog, and S. Markovski. MQQ-SIG, an ultra-fast and provably CMA resistant digital signature scheme. In Proc. of INTRUST 2011, LNCS vol. 7222, pp. 184-203, 2012.
33. S. Goldwasser and S. Micali, Probabilistic Encryption, J. of Comp. and System Sci., vol. 28, pp. 270–299, 1984.
34. L. Goubin and N. T. Courtois. Cryptanalysis of the TTM Cryptosystem. In Proc. of ASIACRYPT '00, LNCS 1976, pp. 44-57, Springer-Verlag 2000.
35. S. Halevi and H. Krawczyk. MMH: Software Message Authentication in the Gbit/second Rates. FSE, LNCS vol. 1267, pp. 172-189, 1997.
36. H. Imai and T. Matsumoto. Algebraic methods for constructing asymmetric cryptosystems. In Proc. of the 3rd Internat. Conf. on Algebraic Algorithms and Error-Correcting Codes, AAECC-3, pp. 108–119, Springer-Verlag, 1986.
37. A. Joux and V. Vitse. Cover and Decomposition Index Calculus on Elliptic Curves made practical. Application to a seemingly secure curve over F_{p^6}, Cryptology ePrint Archive, Report 2011/020, 2011.
38. A. Kipnis, J. Patarin, and L. Goubin. Unbalanced Oil and Vinegar signature schemes. In Advances in Cryptology EUROCRYPT 1999, LNCS vol. 1592, pp. 206-222, Springer, 1999.
39. A. Kipnis and A. Shamir. Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization. In Advances in Cryptology Crypto '99, LNCS vol. 1666, pp. 19-30. Springer Verlag, 1999.
40. K. Kobara and H. Imai. Semantically Secure McEliece Public-Key Cryptosystems Conversions for McEliece PKC In Proc. of PKC 2001, LNCS vol. 1992, pp. 19-35, Springer-Verlag 2001.
41. T. Krovetz and P. Rogaway. Fast Universal Hashing with Small Keys and no Preprocessing: the PolyR Construction. ICICS, LNCS vol. 2015, pp. 73-89, 2000.
42. MAGMA. High performance software for algebra, number theory, and geometry `http://magma.maths.usyd.edu.au`
43. R. J. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. In DSNP Report 1978.
44. T.T. Moh. A public key system with signature and master key functions. Comm. in Algebra, 27, 2207–2222, 1999.
45. M.S. Mohamed, J. Ding, J. Buchmann, and F. Werner. Algebraic attack on the MQQ public key cryptosystem. In *Proc. of CANS '09*, pp. 392–401, Springer-Verlag, 2009.
46. L. H. Nguyen and A. W. Roscoe. Short-output universal hash functions and their use in fast and secure message authentication. Cryptology ePrint Archive: Report 2011/116. `http://eprint.iacr.org/`
47. J. Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In Proc. of EUROCRYPT '96, pp. 33–48, Springer-Verlag, 1996.
48. J. Patarin, N. Courtois, and L. Goubin. FLASH, a fast multivariate signature algorithm. In Proc. of The Cryptographers Track at RSA, CT-RSA 2001, pp. 298-307, Springer-Verlag, 2001.
49. D.H. Phan and D. Pointcheval. Chosen-Ciphertext Security without Redundancy. In Proc. of Asiacrypt '03, LNCS vol. 2894, pp. 1–18, Springer-Verlag, 2003.
50. L. Perret. Personal e-mail communication with Danilo Gligoroski, 2008.
51. A. Petzoldt, S. Bulygin, and J. Buchmann. 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/`.
52. D. Pointcheval. Chosen-Ciphertext Security for Any One-Way Cryptosystem. In Proc. of PKC '00, LNCS 1751, pp. 129-146. SpringerVerlag, 2000.
53. J. Proos, Imperfect Decryption and an Attack on the NTRU Encryption Scheme, IACR Crypt. Archive, 2003/02.
54. C. Rackoff, D. Simon, Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Advances in CryptologyCRYPTO '91, LNCS vol. 576, pp. 433-444, Springer, 1992.
55. S. Samardjiska, Y. Chen and D. Gligoroski. Construction of Multivariate Quadratic Quasigroups (MQQs) in arbitrary Galois fields. In IEEE Proc. of the 7th Internat. Conf. on Information Assurance and Security, IAS 2011, pp. 314–319.
56. I. A. Semaev. Summation polynomials and the discrete logarithm problem on elliptic curves. Cryptology ePrint Archive, Report 2004/031, 2004.
57. R.P. Singh, B.K.Sarma, and A.Saikia. Public key cryptography using permutation p-polynomials over finite fields. Cryptology ePrint Archive, Report 2009/208, 2009. `http://eprint.iacr.org/`.
58. A. Shamir. Efficient Signature Schemes Based on Birational Permutations. In Proc. of CRYPTO '93, LNCS vol. 773, pp. 1–12, Springer-Verlag, 1993.
59. E. Thomae. A Generalization of the Rainbow Band Separation Attack and its Applications to Multivariate Schemes, Cryptology ePrint Archive, Report 2012/223, 2012. `http://eprint.iacr.org/`.

60. S. Tsujii, M. Gotaishi, K. Tadaki, and R. Fujita. Proposal of a signature scheme based on STS trapdoor. In Proc. of Post-Quantum Cryptography, LNCS vol.6061, pp.201–217, Springer, 2010.
61. B-Y. Yang, C-M. Cheng, B-R. Chen, and J.-M. Chen. Implementing minimized multivariate PKC on low-resource embedded systems. In Security in Pervasive Computing, SPC 2006, LNCS vol. 3934 , pp. 73-88. Springer, 2006.
62. B-Y. Yang and J-M. Chen. Rank attacks and defense in Tame like multivariate PKC's. Cryptology ePrint Archive, Report 2004/061, `http://eprint.iacr.org/`.
63. C. Wolf, A. Braeken, and B. Preneel. On the security of stepwise triangular systems. Des. Codes Cryptography, 40:285-302, 2006.
64. C. Wolf and B. Preneel. Taxonomy of public key schemes based on the problem of multivariate quadratic equations. Cryptology ePrint Archive, Report 2005/077, 2005.

A Proofs

Proof of Proposition 1

Proof. The complexity of finding a Gröbner basis of a system of n polynomials with n variables over $\mathbb{F}_2$ is $\mathcal{O}\left(\binom{n+D_{reg}}{n}^{\omega}\right)$ ([1]). The degree of regularity $D_{reg,rand}$ for a randomly generated system can be found using the approximation from [1]

$$D_{reg,rand}(n) \approx 0.09n + 1.00n^{\frac{1}{3}} - 1.58 \tag{8}$$

For $n = 256 - 8 = 248$ we have $D_{reg,rand}(248) \approx 27$. Applying the relation (5) where we take $C = 2$, we get that $D_{reg,\mathrm{MQQ-ENC}}(248) \approx 25$. Taking into account the sparsity of the matrices obtained during the execution of the F_4 algorithm, we can take $w = 2$ [59]. Now, the complexity of finding a Gröbner basis of the public key of MQQ-ENC over $\mathbb{F}_2$ for the given parameters can be found to be $\mathcal{O}(2^{234})$. From the relation (6) for other values of $k \in \{2, 4, 8\}$ we have that the complexity is $\mathcal{O}(2^{234-(k-1)}) = \mathcal{O}(2^{235-k})$. □

Proof of Proposition 2

Proof. For a random MQ system over $\mathbb{F}_2$ the complexity of the Las Vegas probabilistic variant of BooleanSolve is $\mathcal{O}(2^{0.792n})$. The public key of MQQ-ENC defined over $\mathbb{F}_2$ is clearly not a random MQ system, thus the expression $\mathcal{O}(2^{0.792n})$ is not directly applicable for $n = 256 - 8$. However, as there is a relation between the degree of regularity of a random MQ system and MQQ-ENC public key expressed via (5), and from the proof of Proposition 1 we can assume that the public key of MQQ-ENC acts as a random system with $D_{reg,rand}(n') \approx D_{reg,\mathrm{MQQ-ENC}}(248) \approx 25$. Now the solution of $D_{\mathrm{reg,rand}}(n') \approx 25$ is $n' = 227$. Replacing $n = n' = 227$ in $\mathcal{O}(2^{0.792n})$ gives us a complexity of $\mathcal{O}(2^{179})$ of the Las Vegas probabilistic variant of BooleanSolve for solving MQQ-ENC with the recommended parameters over $\mathbb{F}_2$. □

Proof of Theorem 2

Proof. The decryption of MQQ-ENC can be seen as a sequence of at most $\eta = 2^{rem \cdot k}$ independent experiments, each of which yields success with probability $p = \frac{1}{2^{nk/4}}$. For simplicity, we assume that all the experiments are executed, and that the first successful experiment gives the output of the decryption.

Then, the probability π that a decryption error occurs is $\pi = \sum_{i=2}^{\eta} P(B_i|A)P(C_i)$, where A is the event that there is at least one successful experiment, B_i is the event that there are exactly i

successful experiments, and C_i is the event that in the event of B_i, the successful experiment that leads to correct decryption is not the first in the sequence of successful experiments. Than we have:

$$\pi < \sum_{i=2}^{\eta} P(B_i|A) = \sum_{i=2}^{\eta} \frac{P(B_i \cap A)}{P(A)} = \sum_{i=2}^{\eta} \frac{P(B_i)}{P(A)} = \frac{1 - P(B_0) - P(B_1)}{1 - P(B_0)} = \frac{1 - (1-p)^\eta - \eta \cdot p \cdot (1-p)^{\eta-1}}{1 - (1-p)^\eta}$$

It is not hard to see that for a given η, the function $f(p) = \dfrac{\eta \cdot p}{2} - \dfrac{1 - (1-p)^\eta - \eta \cdot p \cdot (1-p)^{\eta-1}}{1 - (1-p)^\eta}$ is continuous on the interval $(0, 1)$, monotonically increasing, and that $\lim_{p \to 0^+} f(p) = 0^+$.

Hence, $\pi < \dfrac{\eta \cdot p}{2}$, and the claim follows.

Proof of Proposition 3

Proof. The idea is to show that no matter what strategy the adversary applies, it is still very improbable to find a pair (y_1, z) that leads to erroneous decryption, i.e., $\mathcal{D}^{MQQ}(\mathcal{E}^{MQQ}(y_1, z)) \neq y_1$ (such pairs will be called bad pairs [21]). This will provide insurance that even if the adversary knows a smart way of using the decryption errors in an attack, it is so hard to find them, that the attack becomes infeasible.

First of all, let the adversary just pick at random candidate ciphertexts (c_1, c_2) to query the decryption oracle $\mathcal{D}_{K-I}^{MQQ}$. If c_1 is not a valid ciphertext for $\mathcal{D}^{MQQ}$, then it will be rejected. The probability that the adversary finds a valid c_1 in this way is bounded by $2^{nk/4 - rem \cdot k}$. Still, nothing guarantees that c_2 is valid, and that it will not be rejected. (In order to be accepted by $\mathcal{D}_{K-I}^{MQQ}$ it has to satisfy $H_1(G(z) \oplus (y_1||c_2)) = z$.)

Instead of doing this, since the goal of the adversary is to find valid ciphertexts that will decrypt wrongly, he can use a strategy that always produces valid c. Hence, the security rests on the hardness of finding a bad pair that further on satisfies $H_1(G(z) \oplus (y_1||c_2)) = z$.

Let q_{H_1} and q_G be the total number of calls to the oracles H_1 and G, respectively.

A natural construction of bad pair satisfying the constrains is the following: Pick r and m at random. Let $z = H_1(r||m)$, and $(y_1||c_2) = G(z) \oplus (r||m)$. Now, the constrains are satisfied, and this will of course produce a valid ciphertext. The probability that this cipher is decrypted wrongly, no matter how r and m are chosen, is at most $1/2^{nk/4 - rem \cdot k + 1}$. Thus, the total probability for all calls to H_1 for this purpose is at most $q_{H_1}/2^{nk/4 - rem \cdot k + 1}$.

From another point of view, the adversary can use the calls to G made during the history of the game in the following way: Let G be called on inputs $x_1, x_2, \ldots, x_{q_G}$. Let $r||m$ be arbitrary. Check whether $H_1(r||m) \in \{x_1, x_2, \ldots, x_{q_G}\}$. This happens with probability $q_G/2^{nk/4}$. Let's say this happens, and for some x_j we have that $H_1(r||m) = x_j$. We put $z = x_j$ and $y_1||c_2 = (r||m) \oplus G(z)$. Again the constrains are satisfied, and in this case we get a total probability $q_{H_1} q_G / 2^{nk/4}$.

Now the claim follows.

B Comparison of experimental results of Gröbner basis attack on MQQ-ENC using Left MQQs and bilinear MQQs

During our experiments, we created a version of the system, called MQQ-ENC$_{bl}$, where instead of LMQQs created by Algorithm **CreateLMQQ**, we used bilinear MQQs from the design of MQQ-SIG [32]. All other characteristics are the same as in MQQ-ENC. Table 4 shows a comparison of the experimental results of a Gróbner basis attack on MQQ-ENC and MQQ-ENC$_{bl}$. A visual representation of the degrees of regularity of the two systems compared to the one of a random system is given in Figure 1.

n	$D_{reg,rand}$	$D_{reg,\mathrm{MQQ-ENC}}$	$D_{reg,\mathrm{MQQ-ENC}_{bl}}$	Time(sec) MQQ-ENC	Time(sec) MQQ-ENC$_{bl}$	Memory(MB) MQQ-ENC	Memory(MB) MQQ-ENC$_{bl}$
24	5	4.27	3.58	0.13	0.01	22.56	15.00
32	6	4.98	4.02	55.66	0.38	811.00	31.93
40	6	5.06	4.17	2 058.76	7.70	13 755.86	198.49
48	7	5.67	4.34	21 981.07	45.53	89 987.15	654.26
56	8	7.50	4.42	128 644.55	191.25	308 728.40	2267.71
64	9	8.10	4.49	771 861.06	724.73	1 372 192.13	6170.68
72	10	-	4.70	-	1208.17	-	8877.79
80	11	-	4.78	-	1735.03	-	17014.11
88	12	-	4.85	-	671.39	-	10587.19
96	13	-	4.78	-	1293.82	-	16581.81

Table 4. A comparison of the average Degree of regularity, Time and Memory complexity of solving MQQ-ENC and MQQ-ENC$_{bl}$ systems of $n - 8$ variables over $\mathbb{F}_2$. $D_{reg,rand}$ is the expected degree of regularity of a random system of $n - 8$ variables

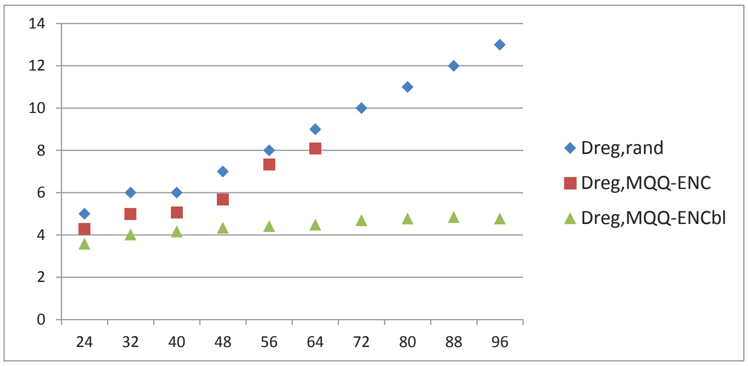

Fig. 1. A graphical comparison of the average Degree of regularity of the MQQ-ENC and MQQ-ENC$_{bl}$ systems of $n - 8$ variables over $\mathbb{F}_2$. $D_{reg,rand}$ is the expected degree of regularity of a random system of $n - 8$ variables