Multivariate Quadratic Trapdoor Functions Based on Multivariate Quadratic Quasigroups

DANILO GLIGOROSKI
Centre for Quantifiable Quality of Service in Communication Systems Norwegian University of Science and Technology O.S. Bragstads plass 2E, Trondheim NORWAY

SMILE MARKOVSKI
Ss “Cyril and Methodius” University Institute of Informatics Faculty of Sciences Gazi baba bb. P.O.Box 162, Skopje REPUBLIC OF MACEDONIA

SVEIN JOHAN KNAPSKOG
Centre for Quantifiable Quality of Service in Communication Systems Norwegian University of Science and Technology O.S. Bragstads plass 2E, Trondheim NORWAY

Abstract: We have designed a new class of multivariate quadratic trapdoor functions. The trapdoor functions are generated by quasigroup string transformations based on a class of quasigroups called multivariate quadratic quasigroups (MQQ). The public key schemes using these trapdoor functions are bijective mappings, they do not perform message expansions and can be used both for encryption and signatures. The public key consist of \( n \) quadratic polynomials with \( n \) variables where \( n = 140, 160, 180, \ldots \). A particular characteristic of our public key scheme is that it is very fast; it has the speed of decryption/signature generation as a typical symmetric block cipher.

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

1 Introduction

Public key cryptographic schemes based on multivariate quadratic polynomials have been known in cryptography for more than 20 years.

There are 4 basic trapdoor functions that are based on multivariate quadratic polynomials.

The first MQ scheme called MIA is that of Matsumoto and Imai [1] from 1985. That scheme was broken in 1995 by Patarin [2].

The second MQ scheme called STS (Stepwise Triangular Scheme) was first introduced in 1993 by Shamir [3] in the variant called Birational Permutation Schemes and was successfully broken by Coppersmith et al. in the same year [4]. In 1999 a MQ scheme which is a variant of STS called TTM was proposed by Moh [5]. That scheme was broken in 2000 by Goubin and Courtois [6]. The whole STS class has been broken by Wolf, Braeken and Preneel in 2004 [7].

The third MQ scheme called HFE (Hidden Field Equations) was designed by Patarin [8] in 1996. Basic HFE was broken by solving instances of the MinRank problem, by Kipnis and Shamir [9] in 1999.

The forth scheme called UOV (Unbalanced Oil and Vinegar) was proposed in 1999 by Kipnis et al. [10] and is a generalization of the original Oil and Vinegar scheme of Patarin [11] from 1997. The basic variant of UOV has been broken in [12].

Our results. We have designed a new class of MQ trapdoor functions based on the theory of quasigroups and quasigroup string transformations. Our PKC scheme is: 1. A deterministic one-to-one mapping; 2. There is no message expansion; 3. It has one parameter \( n = 140, 160, 180, \ldots \) – the bit length of the encrypted block; 4. Its conjectured security level when \( n \geq 140 \) bits is \( 2^{2n/2} \); 5. Its encryption speed is comparable to the speed of other multivariate quadratic PKCs; 6. Its decryption/signature speed is as a typical symmetric block cipher (i.e. in the range of 500–1000 times faster than the most popular public key schemes); 7. It is well suited for short signatures and smart card implementations.

The paper is composed like this: In section 2 we give preliminaries, define a new special class of so called Multivariate Quadratic Quasigroups (MQQ) and we describe a public-key cryptosystem based on MQQs. Its operating characteristics are given in Section 3. We discuss the security of our PKC in section 4. Conclusions are given in section 5.

2 Preliminaries

Quasigroup string transformations. Here we give a brief overview of quasigroups and quasigroup string transformations. A more detailed explanation is found in [13, 14].
**Definition 1** A quasigroup \((Q, \ast)\) is a groupoid satisfying the law

\[(\forall u, v \in Q)(\exists! x, y \in Q)u \ast x = v, \quad y \ast u = v. \tag{1}\]

It follows from (1) that for each \(a, b \in Q\) there is a unique \(x \in Q\) such that \(a \ast x = b\). Then we denote \(x = a \setminus b\) where \(\setminus\) is a binary operation in \(Q\) (called a left parastrophe of \(\ast\) and the groupoid \((Q, \setminus)\) is a quasigroup too. The algebra \((Q, \ast, \setminus)\) satisfies the identities

\[x \setminus (x \ast y) = y, \quad x \ast (x \setminus y) = y. \tag{2}\]

![Figure 1: Graphical representations of the \(e_l, \ast\) and \(d_l, \ast\) transformations](image)

Consider an alphabet (i.e., a finite set) \(Q\), and denote by \(Q^+\) the set of all nonempty words (i.e., finite strings) formed by the elements of \(Q\). In this paper, depending on the context, we will use two notations for the elements of \(Q^+: a_1a_2\ldots a_n\) and \((a_1, a_2, \ldots, a_n)\), where \(a_i \in Q\). Let \(\ast\) be a quasigroup operation on the set \(Q\). For each \(l \in Q\) we define two functions \(e_{l, \ast}, d_{l, \ast}: Q^+ \rightarrow Q^+\) as follows:

**Definition 2** Let \(a_i \in Q\), \(M = a_1a_2\ldots a_n\). Then

\[e_{l, \ast}(M) = b_1b_2\ldots b_n \iff \Rightarrow b_1 = l \ast a_1, \quad b_2 = b_1 \ast a_2, \ldots, \quad b_n = b_{n-1} \ast a_n, \quad d_{l, \ast}(M) = c_1c_2\ldots c_n \iff \]

\[\quad \Rightarrow c_1 = l \ast a_1, \quad c_2 = a_1 \ast a_2, \ldots, \quad c_n = a_{n-1} \ast a_n, \quad i.e., \quad b_{i+1} = b_i \ast a_{i+1} \quad \text{and} \quad c_{i+1} = a_i \ast a_{i+1} \quad \text{for each} \quad i = 0, 1, \ldots, n - 1, \quad \text{where} \quad b_0 = a_0 = l.\]

The functions \(e_{l, \ast}\) and \(d_{l, \ast}\) are called the \(e\)-transformation and the \(d\)-transformation of \(Q^+\) based on the operation \(\ast\) with leader \(l\) respectively, and their graphical representations are shown in Fig. 1.

**Theorem 1** If \((Q, \ast)\) is a finite quasigroup, then \(e_{l, \ast}\) and \(d_{l, \ast}\) are mutually inverse permutations of \(Q^+\), i.e.,

\[d_{l, \ast}(e_{l, \ast}(M)) = M = e_{l, \ast}(d_{l, \ast}(M)) \forall l \in Q \quad \text{and for every string} \quad M \in Q^+. \]

Quasigroups as vector valued Boolean functions. To define a multivariate quadratic PKC for our purpose, we will use the presentation of finite quasigroups \((Q, \ast)\) of order \(2^d\) by vector valued Boolean functions (v.v.b.f.). In what follows we will represent \(a \in Q\) by their \(d\)-bit representation, i.e., \(a \equiv x_1x_2\ldots x_d\) or, sometimes, \(a \equiv (x_1, x_2, \ldots, x_d)\). Now, the binary operation \(\ast\) on \(Q\) can be seen as a vector valued operation \(\ast_{vv}: \{0, 1\}^{2d} \rightarrow \{0, 1\}^{d}\) defined as:

\[a \ast b = c \iff \ast_{vv}(x_1, \ldots, x_d, y_1, \ldots, y_d) = (z_1, z_2, \ldots, z_d)\]

where \(x_1\ldots x_d, y_1\ldots y_d, z_1\ldots z_d\) are binary representations of \(a, b, c\).

Each \(z_i\) depends on the bits \(x_1, \ldots, x_d, y_1, \ldots, y_d\) and is uniquely determined by them. So, each \(z_i\) can be seen as a \(2\)-ary Boolean function \(z_i = f_i(x_1, \ldots, x_d, y_1, \ldots, y_d)\), where \(f_i : \{0, 1\}^{2d} \rightarrow \{0, 1\}\) strictly depends on, and is uniquely determined by \(x, \ast\).

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

\[ANF(f) = \alpha_0 + \sum_{i=1}^{k} \alpha_i x_i + \sum_{1 \leq i < j \leq k} \alpha_{i,j} x_i x_j + \sum_{1 \leq i < j < s < k} \alpha_{i,j,s} x_i x_j x_s + \ldots\]

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 \(GF(2)\). In the rest of the text we will abuse the notation and identify the Boolean function \(f\) and its ANF, i.e., we will take \(f = ANF(f)\). We say a polynomial \(f(x_1, \ldots, x_k)\) when we consider the arguments of \(f\) to be indeterminate variables \(x_1, x_2, \ldots, x_k\).

The ANFs of the functions \(f_i\) give us information about the complexity of the quasigroup \((Q, \ast)\) via the degrees of the Boolean functions \(f_i\). It can be observed that the degrees of the polynomials \(ANF(f_i)\) rise with the order of the quasigroup. In general, for a randomly generated quasigroup of order \(2^d\), \(d \geq 4\), the degrees are higher than 2. Such quasigroups are not suitable for our construction of multivariate quadratic PKC.
Multivariate Quadratic Quasigroups. We define a special class of quasigroups, called multivariate quadratic quasigroups (MQQs) that can be of different types.

**Definition 3** A quasigroup \((Q, \ast)\) 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 \(0 \leq k < d\).

Table 1: Definition of MQQ\((d, k)\)

<table>
<thead>
<tr>
<th>(MQQ(d, k))</th>
<th>(Input:) Integer (d) and integer (k), (0 \leq k &lt; d)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Output:</strong></td>
<td>a quasigroup of order (2^d) of type (Quad_{d-k}Lin_k)</td>
</tr>
<tr>
<td>1. Randomly generate a (d \times d) matrix (A_1) of linear Boolean expressions of variables (x_1, \ldots, x_d), such that (\text{Det}(A_1) = 1) in (GF(2)) and the number (#\text{Const}) of constants 0 or 1 in the matrix (A_1) satisfies the inequality (kd \leq #\text{Const} &lt; (k + 1)d).</td>
<td></td>
</tr>
<tr>
<td>2. Randomly generate a (d \times d) matrix (B_1) of linear Boolean expressions of variables (x_1, \ldots, x_d).</td>
<td></td>
</tr>
<tr>
<td>3. Compute the vector (*<em>{v1} = A_1 \cdot x_2 + b_1), where (x_2 = (x_d+1, \ldots, x</em>{2d})^T).</td>
<td></td>
</tr>
<tr>
<td>4. Represent (<em>_{v2}) as (</em>_{v2} = A_2 \cdot x_1 + b_2), where (x_1 = (x_0, \ldots, x_1)^T).</td>
<td></td>
</tr>
<tr>
<td>5. if ((\text{Det}(A_2) = 1) in (GF(2))) and (<em><em>{v2}) is of type (Quad</em>{d-k}Lin_k) then Return(</em>_{v2}), else Go To 1.</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Definition of \(P'(n)\)

<table>
<thead>
<tr>
<th>(P'(n))</th>
<th>(Input:) Integer (n), where (n = 5k), (k \geq 28), and a vector (x = (f_1, \ldots, f_n)) of (n) linear Boolean functions of (n) variables.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Output:</strong></td>
<td>Eight quasigroups (*_1, \ldots, *_8) and (n) multivariate quadratic polynomials (P'_i(x_1, \ldots, x_n), i = 1, \ldots, n)</td>
</tr>
<tr>
<td><strong>Preprocessing phase</strong>enerate two large sets (Quad_{d}Lin_1) and (Quad_{d}Lin_0) with more than (2^{20}) elements each of MQQs of type (Quad_{d}Lin_1) and of type (Quad_{d}Lin_0) such that the minimal rank of their quadratic polynomials when represented in matrix form is at least 8; Transform (by permuting the coordinates) all quasigroups in the set (Quad_{d}Lin_1) such that their first coordinate is linear.</td>
<td></td>
</tr>
<tr>
<td>1. Represent a vector (x = (f_1, \ldots, f_n)) of (n) linear Boolean functions of (n) variables (x_1, \ldots, x_n), as a string (x = X_1 \ldots X_k) where (X_i) are vectors of dimension 5;</td>
<td></td>
</tr>
<tr>
<td>2. Pick randomly two different quasigroups (*_1, <em><em>2 \in Quad</em>{d}Lin_1) and six different quasigroups (</em>_3, *_4, *_5, *_6, *_7, *<em>8 \in Quad</em>{d}Lin_0).</td>
<td></td>
</tr>
<tr>
<td>3. Compute (y_1 = Y_1 \ldots Y_7) where: (Y_1 = X_1), (Y_2 = X_1 \ast_1 X_2, Y_3 = X_2 \ast_2 X_3) and (Y_{i+1} = Y_i \ast_{(i+2)}(mod\ 6)) (X_i).</td>
<td></td>
</tr>
<tr>
<td>4. Form a 7-dimensional vector (Z = Y_1</td>
<td></td>
</tr>
<tr>
<td>5. Transform (Z) by the bijection of Dobbertin i.e. (W = \text{Dob}(Z)).</td>
<td></td>
</tr>
<tr>
<td>6. Set (Y_1 = (W_1, W_2, W_3, W_4, W_5)), (Y_{2,1} = W_6, Y_{3,1} = W_7).</td>
<td></td>
</tr>
<tr>
<td>7. Output: Quasigroups (*_1, \ldots, *_8) and (n) as multivariate quadratic polynomials (P'_i(x_1, \ldots, x_n), i = 1, \ldots, n).</td>
<td></td>
</tr>
</tbody>
</table>

where \(x_2 = (x_{d+1}, \ldots, x_{2d})^T\) are unknown bits, \(c = (c_1, \ldots, c_d)^T\) and \(a_i, c_i\) are given bits. We have the linear system in \(GF(2)\) of kind

\[
A'_1 \cdot x_2 + b'_1 = c
\]

(7)

where \(A'_1\) and \(b'_1\) are the valuations of \(A_1\) and \(b_1\) over the vector \((a_1, \ldots, a_d)\). Since \(\text{Det}(A_1) = 1\), it follows that \(\text{Det}(A'_1) = 1\) too, so the linear system (7) has a unique solution \(x_2 = (A'_1)^{-1} \cdot (c - b'_1)\). In the same manner a unique solution of the equation \(*_{v2}(x_1, \ldots, x_d, a_{d+1}, \ldots, a_{2d}) = c\) can be found, and \(*_{v2}\) is a v.v.b.f. of a quasigroup operation \(*\) on the set \(Q = \{0, 1, \ldots, 2^d - 1\}\). The quasigroup \((Q, *)\) is MQQ since the vector \(A_1 \cdot x_2\) has as elements multivariate quadratic polynomials. □

By using Theorem 2 we define the procedure \(MQQ(d, k)\) for producing MQQs of order \(2^d\) and type \(Quad_{d-k}Lin_k\).

Note that the procedure \(MQQ(d, k)\) is a randomized algorithm for finding MQQs of order \(2^d\) and type \(Quad_{d-k}Lin_k\). For \(d = 5\) the average number of attempts for finding MQQs of type \(Quad_{d}Lin_1\) is
around $2^{15}$ and for finding MQQs of type $Quad_5 Lin_0$ is around $2^{16}$. However, MQQ(6,0) did not give us any MQQ of order $2^9$. Finding MQQs of orders $2^d$, $d \geq 6$, we consider as an open research problem.

The definition of MQQs implies the following theorem:

**Theorem 3** Let $x_1 = (f_1, \ldots, f_d)$ and $x_2 = (f_{d+1}, \ldots, f_{2d})$ be two $d$-dimensional vectors of linear Boolean functions of variables $x_1, \ldots, x_d$. Let $(Q, *)$ be a multivariate quadratic quasigroup of type $Quad_d-Lin_k$. If $x_1 * x_2 = (g_1, \ldots, g_d)$ then at most $d - k$ of the polynomials $g_i$ are multivariate quadratic and at least $k$ polynomials are linear. □

We want to emphasize that in a process of a random generation of MQQs of type $Quad_d-Lin_k$, usually the number of quadratic polynomials is exactly $d - k$, and the number of linear polynomials is exactly $k$. However, there are rare cases when all quadratic terms can cancel each other, and the number of linear polynomials will be bigger than $k$ while the number of quadratic polynomials will be less than $d - k$. Nevertheless, these cases, if they occur, can be easily detected, and quasigroups with such properties can be omitted from consideration as candidates for the private key.

**The bijection of Dobbertin.** By using only MQQs, some of the coordinate functions will remain linear. In order to make a trapdoor bijective function $\{0, 1\}^n \rightarrow \{0, 1\}^n$ that is multivariate quadratic in all of its coordinates we use the bijection of Dobbertin.

Dobbertin has proved [15] that the function $\text{Dob}(X) = X^{2n+1} + X^3 + X$ is a bijection in $GF(2^{2n+1})$. Moreover it is multivariate quadratic too. In our design of MQQ public key cryptosystem we use the bijection of Dobbertin for $m = 3$.

**Description of the algorithm.** A generic description for our scheme can be expressed as: $T \circ P' \circ T : \{0, 1\}^n \rightarrow \{0, 1\}^n$ where $T$ is a nonsingular linear transformation, and $P'$ is a bijective multivariate quadratic mapping on $\{0, 1\}^n$.

The algorithm for the mapping $P' : \{0, 1\}^n \rightarrow \{0, 1\}^n$ is defined in Table 2.

The algorithms for generating the public and private key and for decryption/signing by the use of the private key $(T, *_1, \ldots, *_8)$ are given above. The algorithm for encryption with the public key is straightforward application of the set of $n$ multivariate polynomials $P(x_1, \ldots, x_n)$, $i = 1, \ldots, n$ over a vector $x = (x_1, \ldots, x_n)$, i.e., $y = P(x)$.

### Table 3: Generation of Public and Private key for the MQQ scheme

**Algorithm for generating Public and Private key for the MQQ scheme**

**Input:** Integer $n$, where $n = 5k$ and $k \geq 28$.

**Output:** Public key: $n$ multivariate quadratic polynomials $P_i(x_1, \ldots, x_n), i = 1, \ldots, n$.

Private key: Nonsingular Boolean matrix $T$ of order $n \times n$ and eight quasigroups $*_1, \ldots, *_8$.

1. Generate a nonsingular $n \times n$ Boolean matrix $T$ (uniformly at random).
2. Call the procedure for definition of $P' : \{0, 1\}^n \rightarrow \{0, 1\}^n$ and from there also obtain the quasigroups $*_1, \ldots, *_8$.
3. Compute $y = T(P'(T(x)))$ where $x = (x_1, \ldots, x_n)$.
4. Output: The public key is $y$ as $n$ multivariate quadratic polynomials $P_i(x_1, \ldots, x_n), i = 1, \ldots, n$, and the private key is the tuple $(T, *_1, \ldots, *_8)$.

### Table 4: Algorithm for decryption or signing

**Algorithm for decryption/signing with the private key $(T, *_1, \ldots, *_8)$**

**Input:** A vector $y = (y_1, \ldots, y_8)$.

**Output:** A vector $x = (x_1, \ldots, x_n)$ such that $P(x) = y$.

1. Set $y' = T^{-1}(y)$.
2. Set $W = (y_1, y_2, y_3, y_4, y_5, y_6, y_11)$.
3. Compute $Z = (Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7) = \text{Dob}^{-1}(W)$.
4. Set $y'_1 \leftarrow Z_1, y'_2 \leftarrow Z_2, y'_3 \leftarrow Z_3, y'_4 \leftarrow Z_4, y'_5 \leftarrow Z_5, y'_6 \leftarrow Z_6, y'_11 \leftarrow Z_7$.
5. Represent $y'$ as $y' = Y_1 \ldots Y_k$ where $Y_i$ are vectors of dimension 5.
6. By using the left parastrophes $\\setminus$ of the quasigroups $*_i, i = 1, \ldots, 8$, obtain $x' = X_1 \ldots X_k$, such that: $X_1 = Y_1, X_2 = X_1 \setminus Y_2, X_3 = X_2 \setminus Y_3$ and $X_i = X_{i-1} \setminus_{m+(i+2)\text{mod} 8} Y_i$.
7. Compute $x = T^{-1}(x')$.

### 3 Operating characteristics

In this section we will discuss the size of the private and the public key as well as the number of operations per byte for encryption and decryption.

The size of the public and the private key. Since the public key consists of $n$ multivariate quadratic equations, and they appear to be randomly generated, the size of the public key follows the rules given in [16]. So, for $n$ bit blocks the size of the public key is $n \times \left(1 + \frac{n(n+1)}{2}\right)$ bits. In the Table 5 we give the size of the public key for $n \in \{140, 160, 180, 200\}$ in KBytes.

The private key of our scheme is the tuple $(T, *_1, \ldots, *_8)$. The corresponding memory size needed for storage of $T$ is $n^2$ bits. The memory size for the quasigroups $(*_1, \ldots, *_8)$, (actually for their parastrophes), is $8 \times 32 \times 32 \times 5 = 40960$ bits. For
the storage of the inverse table of the bijection of Dobbertin we need additional 896 bits.

In total, the size of the private key expressed in Kb is \(\frac{1}{2\pi}(n^2 + 40960 + 896)\). In the second column of the Table 5 we give the size of the private key for \(n \in \{140, 160, 180, 200\}\) in KBytes.

The number of operations for encryption and decryption. In order to obtain an independent measure for the operating speed of our scheme, we will express the speed of encryption and decryption/signing as the number of operations per processed byte. We will also take into account three widespread microprocessor architectures: 8-bit, 32-bit and 64-bit architectures.

Since the public part of our scheme follows the typical paradigm of the MQ public key cryptosystems, its speed of encryption is the same as (or similar to) the speed of other MQ systems. That means that the encryption is done after \(O(n^3)\) logical AND and logical XOR operations.

The actual speed of any multivariate quadratic PKC when encryption is performed on 32-bit or 64-bit microprocessor architectures, using internal parallelism of the modern CPUs, as well as techniques of bit slicing can result in an encryption process which is at least two orders of magnitude faster than RSA/DH or ECC encryption for systems with equivalent security levels.

If we assume that AND or XOR operations can be executed in one cycle (without taking into account that modern 32-bit and 64-bit CPUs actually can perform several such operations in parallel), then non-optimized encryption of any general \(n\)-bit variant of any multivariate quadratic PKC scheme have a speed of \(\frac{16}{n} \left[ \frac{n}{Arch} \right] (1 + \frac{n(n+1)}{2})\) operations per byte where \(Arch = 8, 32, 64\).

The speed of decryption/signing in the class of multivariate quadratic PKCs is not so uniformly distributed as it is for encryption. The number of operations for particular parts of the process of decryption of our scheme can be summarized in the following list:

- a) Two linear operation by the matrix \(T^{-1}\) that takes \(2n \left[ \frac{n}{Arch} \right] \) operations;
- b) One lookup operation at the table of the bijection of Dobbertin;
- c) Exactly \(\frac{n}{5} - 1\) lookup operations at the quasi-group parastrophes.

The total number of operations per byte can be computed by the expression \(\frac{8}{n} (2n \left[ \frac{n}{Arch} \right] + \frac{n}{5})\) and are given in the Table 6 and Table 7.

<table>
<thead>
<tr>
<th>(n)</th>
<th>Size of the public key (KBytes)</th>
<th>Size of the private key (KBytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>140</td>
<td>168.69</td>
<td>7.50</td>
</tr>
<tr>
<td>160</td>
<td>251.58</td>
<td>8.23</td>
</tr>
<tr>
<td>180</td>
<td>357.96</td>
<td>9.06</td>
</tr>
<tr>
<td>200</td>
<td>490.75</td>
<td>9.99</td>
</tr>
</tbody>
</table>

Table 5: Memory size in Kbytes for the public key and the private key.

<table>
<thead>
<tr>
<th>(n)</th>
<th>Operations per encrypted byte</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Arch=8-bit</td>
</tr>
<tr>
<td>140</td>
<td>20306</td>
</tr>
<tr>
<td>160</td>
<td>25762</td>
</tr>
<tr>
<td>180</td>
<td>33306</td>
</tr>
<tr>
<td>200</td>
<td>40202</td>
</tr>
</tbody>
</table>

Table 6: Estimated operations per encrypted byte, for different \(n\) and 8, 32 or 64 bit architectures.

<table>
<thead>
<tr>
<th>(n)</th>
<th>Operations per decrypted byte</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Arch=8-bit</td>
</tr>
<tr>
<td>140</td>
<td>289.65</td>
</tr>
<tr>
<td>160</td>
<td>321.65</td>
</tr>
<tr>
<td>180</td>
<td>369.64</td>
</tr>
<tr>
<td>200</td>
<td>401.64</td>
</tr>
</tbody>
</table>

Table 7: Estimated operations per decrypted byte, for different \(n\) and 8, 32 or 64 bit architectures.

4 Security characteristics of the algorithm

In this section we will give our projections of the strength of the scheme with \(n\) variables. A more detailed security analysis such as the size of the pool of MQQs of order \(2^5\), and the resistance against different kinds of attacks such as: chosen plaintext attack, attack using isomorphism of polynomials, attack using the solution of the MinRank problem, attacks using differential cryptanalysis, and XL and Gröbner basis attacks, will be given in the extended version of this paper.

As a general claim for our MQQ scheme with \(n\) variables we say that its strength is \(2^{2n}\). We base our claims on the analysis of the power of the methods using Gröbner basis to solve random multivariate quadratic systems of equations.

Namely, the authors of [17] give a formula for computing the upper bound for the efficiency of the Gröbner basis attacks. Based on that analysis we are giving here the Table 8 with the projected complexity for solving random multivariate quadratic systems of equations by Gröbner basis algorithms for different number of variables \(n\). Based on that projection in the second row we are giving the projection for the strength of our PKC scheme.
2, and then to use quasigroup string transformations to construct bijective trapdoor multivariate quadratic polynomials.

We have constructed a public key cryptosystem MQQ against Gr¨obner basis attacks.

Based on the complexity of Gr¨obner basis attacks that are currently the most efficient way for solving multivariate quadratic equations we project that our MQQ scheme with 5 variables has strength of $2^{23}$.

Table 8: Complexity of the Gr¨obner basis attacks for different number of variables $n$ and the strength of MQQ against Gr¨obner basis attacks.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$2^{87}$</th>
<th>$2^{99}$</th>
<th>$2^{112}$</th>
<th>$2^{125}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity of Gr¨obner basis attacks</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strength of our MQQ PKC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5 Conclusion

We have constructed a public key cryptosystem MQQ by using quasigroups. The main idea is to represent quasigroups as vector valued Boolean functions, to find a class of quasigroups that have degree at most 2, and then to use quasigroup string transformations to construct bijective trapdoor multivariate quadratic polynomials.

The speed of encryption of our scheme is similar to other MQ schemes, and the speed of decryption is in the range of 500–1000 times faster than the most popular public key schemes.

Based on the complexity of Gr¨obner basis attacks that are currently the most efficient way for solving multivariate quadratic equations we project that our MQQ scheme with $n$ variables have strength of $2^{23}$.

It is an open research problem to count the number of MQQs of order $2^n$ and to find ways to construct MQQs of higher order (for example of order $2^3$). Finding such quasigroups will increase the security and will speed up the process of decryption/signing even more.

References: