Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/070548

An Algorithm for Judging and Generating Bilinear Multivariate Quadratic Quasigroups

Ying Zhang[*] *and Huisheng Zhang*

Department of Mathematics, Dalian Maritime University, Dalian 116026, China

Received: 10 Feb. 2013, Revised: 12 Jun. 2013, Accepted: 15 Jun. 2013
Published online: 1 Sep. 2013

Abstract: Multivariate Quadratic Quasigroups (MQQs) as a suitable class of quasigroups for multivariate public key cryptography recently have been an important mathematical tool in information security field. In this paper, we propose a necessary and sufficient condition to verify whether a quasigroup given by its multiplication table is a bilinear MQQ, which shows that checking whether an arbitrary quasigroup is a bilinear MQQ is equivalent to solving a simple matrix equation. Based on this condition, a deterministic algorithm is proposed to judge whether a given quasigroup is a bilinear MQQ and then obtain the corresponding Boolean functions if it is. An example is given to show the validity of our results.

Keywords: Quasigroup, Multivariate quadratic quasigroup, Vector-valued Boolean functions

1 Introduction

The most popular public-key cryptosystems rely on some computationally intractable problems such as the factoring of large primes in RSA [1] and the calculation of the discrete logarithm in ECC [2]. These algorithms are assumed to become insecure in the era of quantum computers [3]. Therefore, several solutions for the post-quantum cryptography have been proposed in literatures such as hash-based [4], code-based [5], lattice-based [6], and MQ-based (multivariate-quadratic-equation-based) cryptosystems [7]. Among these, MQ-based schemes are promising candidates for post-quantum cryptography since they are fast and suitable to devices that are short of computing and memory resources. The current proposals for MQ-based schemes basically can be classified into four main categories: Matsumoto and Imai (MIA) [8], Stepwise Triangular Scheme (STS) [9], Hidden Field Equations (HFE) [10], and Unbalanced Oil and Vinegar (UOV) [11]. However, though various MQ-based public key cryptosystems have been proposed, only a few of them are relatively safe and most have been broken.

Recently, based on multivariate quadratic quasigroups (MQQs), Gligoroski et al. [12] proposed a new class of MQ-based schemes called MQQ scheme [12] to counter the weaknesses observed in the existing MQ-based systems. The scheme combines the advantages of both the MQ polynomials and quasigroups. On the one hand, as it only needs the basic operations of XOR and AND between bits during the encryption and decryption processes, MQQ scheme performs several orders of magnitude faster than the traditional public key algorithms like RSA, DH or ECC [13]. On the other hand, the key space of MQQ scheme is large enough against an exhaustive key-search attack [12]. Moreover, this scheme offers flexibility in its implementation from parallelization point of view [13]. In a recent work [14], MQQ scheme has been successfully used in wireless sensor network. Though the original MQQ scheme is considered broken now [15], a new proposed signature scheme MQQ-SIG is not vulnerable on the existing successful attacks on MQQ scheme [16] by removing certain percentage of the public key.

As the basic step for MQQ scheme, the generation of MQQs is an important and hard task. In order to increase the security and speed up the process of encryption and decryption, the order of MQQs is desired to be high. As MQQs are a subclass of quasigroups, a natural idea is to test whether quasigroups are MQQs. In the pioneering work [12], Gligoroski et al. established a sufficient condition of generating an MQQ for a given quasigroup. Based on this condition, they brought out a randomized generation algorithm for MQQs. However, this algorithm

[*] Corresponding author e-mail: zhgyg77@sina.com

is time-consuming and can only generate MQQs of order $2^d (d \leq 5)$. Subsequently, Ahlawat et al. [17] proposed an improved algorithm to generate MQQs, and checked the existence of MQQs from $d = 2$ to $d = 14$. Recently, Chen et al. [18] simplified the sufficient condition in [12] and gave an efficient algorithm for generating bilinear MQQs (a subclass of MQQs) of any order 2^d. In addition, Samardjiska et al. [19] and Christov [20] also proposed new algorithms and theory for generating MQQs respectively.

Though several construction methods for MQQs have been proposed, a basic question remains interesting: how to judge whether or not an arbitrary quasigroup is a bilinear MQQ and find the corresponding Boolean functions if it is? This question was answered in [21], where a standard method was proposed to determine the unique Boolean representation of any function defined over a set of 2^d elements from the truth table. However, this method is not specifically designed for MQQs and does not make full use of the algebra relationships among truth tables. Thus, there remains great room for improvement. In this paper, we shall propose a necessary and sufficient condition to determine whether a quasigroup given by its multiplication table is a bilinear MQQ. Then based on this condition, a deterministic algorithm is proposed to check whether the given quasigroup is bilinear and generate the Boolean functions if it is. The advantages of this algorithm over the existing algorithms can be summarized as follows: (i) In contrast with the standard method [21], our algorithm is dedicated to recognize bilinear MQQs and has a better running time; (ii) Compared with the existing MQQs-generating algorithms, it is suitable for quasigroups of any order 2^d and all the bilinear MQQs theoretically can be generated.

The rest of the paper is organized as follows. Section 2 recalls the original MQQ generation scheme [12]. Section 3 gives a necessary and sufficient condition and an algorithm to determine whether or not a given quasigroup is a bilinear MQQ. An explicit example is presented in Section 4. Finally, we conclude the paper in Section 5.

2 Original MQQ generation scheme

Unless otherwise defined in this paper, additions and multiplications are operated in the binary field GF(2).

Definition 2.1 *(Definition 1 in [18]) A quasigroup $(Q,*)$ is a set Q with a binary operation $*$ such that for any $a,b \in Q$, there exist unique x,y:*

$$x * a = b; a * y = b. \tag{1}$$

We call that a quasigroup $(Q,*)$ is of order n if the set Q has n elements. Consider a finite quasigroup $(Q,*)$ of order 2^d. One can choose a bijection $Q \rightarrow \{0,1,\cdots,2^d-1\}$ and represent $a \in Q$ by a unique $d-$bit sequence $(x_1,x_2,\cdots,x_d)$. Now the binary operation $*$ on Q can be considered as a vector valued operation $*_{vv} : \{0,1\}^{2d} \rightarrow \{0,1\}^d$ defined as

$$a * b = c \Leftrightarrow *_{vv}(x_1,\cdots,x_d,x_{d+1},\cdots,x_{2d}) = (z_1,\cdots,z_d), \tag{2}$$

where $x_1,\cdots,x_d,x_{d+1},\cdots,x_{2d}$ and $z_1,\cdots,z_d$ are binary representation of a,b and c, respectively. It is easy to see that each $z_s (1 \leq s \leq d)$ depends on the $2d$ bits $x_1,\cdots,x_{2d}$. Thus each z_s can be regarded as a $2d-$ary Boolean function $z_s = f_s(x_1,\cdots,x_{2d})$, here $f_s : \{0,1\}^{2d} \rightarrow \{0,1\}$ is determined by $*$. As stated in [12], we have the following lemma.

Lemma 2.1 *(Lemma 1 in [12]) For every quasigroup $(Q,*)$ of order 2^d and for each bijection $Q \rightarrow \{0,\cdots,2^d-1\}$, there are a uniquely determined vector valued Boolean function $*vv$ and d uniquely determined $2d-$ary Boolean functions $f_1,\cdots,f_d$ such that for each $a,b,c \in Q$*

$$\begin{aligned} &a * b = c \Longleftrightarrow *vv(x_1,\cdots,x_d,y_1,\cdots,y_d) = \\ &(f_1(x_1,\cdots,x_d,y_1,\cdots,y_d),\cdots,f_d(x_1,\cdots,x_d,y_1,\cdots,y_d)). \end{aligned} \tag{3}$$

In general, for a randomly generated quasigroup of order $2^d (d \geq 4)$, the degrees of Boolean functions are usually higher than 2. Such quasigroups are not suitable for the construction of multivariate quadratic public-key cryptosystem.

Definition 2.2 *(Definition 3 in [12]) A quasigroup $(Q,*)$ of order 2^d is called Multivariate Quadratic Quasigroup (MQQ) of type $Quad_{d-k}Lin_k$ if exactly $d-k$ of the polynomials f_s are of degree 2 and k of them are of degree 1, where $0 \leq k < d$.*

In [12], a sufficient condition is proposed for a quasigroup $(Q,*)$ to be an MQQ.

Proposition 2.2 *(Theorem 2 in [12]) Let $A_1 = [f_{st}]_{d\times d}$ and $A_2 = [g_{st}]_{d\times d}$ be two $d \times d$ matrices of linear Boolean expressions, and let $b_1 = [u_s]_{d\times 1}$ and $b_2 = [v_s]_{d\times 1}$ be two $d \times 1$ vectors of linear or quadratic Boolean expressions. Let the functions f_{st} and u_s depend only on variables $x_1,\cdots,x_d$, and let the functions g_{st} and v_s depend only on variables $x_{d+1},\cdots,x_{2d}$. If*

$$Det(A_1) = Det(A_2) = 1 \ \text{in GF(2)} \tag{4}$$

and if

$$\begin{aligned} &A_1 \cdot (x_{d+1},x_{d+2},\cdots,x_{2d})^T + b_1 \\ &= A_2 \cdot (x_1,x_2,\cdots,x_d)^T + b_2, \end{aligned} \tag{5}$$

then the vector valued operation

$$*vv(x_1,\cdots,x_{2d}) = A_1 \cdot (x_{d+1},\cdots,x_{2d})^T + b_1$$

defines a quasigroup $(Q,*)$ of order 2^d that is MQQ.

If the vector valued Boolean functions defining the MQQ in Proposition 2.2 have no terms of the form $x_s x_t$ with $s,t \leq d$ or $s,t > d$ [22], we call such MQQs as bilinear MQQs in order to differ from other MQQs.

Proposition 2.2 not only proposes a sufficient condition for a quasigroup to be an MQQ, but also provides an approach to finding MQQs. According to

Proposition 2.2, to find an MQQ, one needs the appropriate A_1, A_2, b_1, and b_2, which are time-consuming to choose. Later, Chen simplified this sufficient condition such that only a matrix and a vector need to be determined. However, as those methods are based on the sufficient conditions for a quasigroup to be a bilinear MQQ, the bilinear MQQs they generated are only a subset of the bilinear MQQs. Thus, it is desirable to establish such a necessary and sufficient condition, which is the main job of the next section.

3 New algorithm for justifying and generating bilinear MQQs

In this section, we give a necessary and sufficient condition for a given quasigroup to be a bilinear MQQ, and then use this condition to propose an algorithm for verifying whether a quasigroup is a bilinear MQQ and generating the corresponding Boolean functions if it is.

3.1 A necessary and sufficient condition

Definition 3.1 (see [23]) *Given an $m \times n$ matrix $A = (a_{ij})$, $\overline{vec}(A)$ is a vector defined as*

$$\overline{vec}(A) = (a_{11}, \cdots, a_{1n}, a_{21}, \cdots, a_{2n}, \cdots, a_{m1}, \cdots, a_{mn})^T.$$

Lemma 3.1 (see [23]) *Let $A \in R^{m\times p}, B \in R^{q\times n}, X \in R^{p\times q}$, then*

$$\overline{vec}(AXB) = (A \otimes B^T)\overline{vec}(X),$$

where $\otimes$ denotes tensor product.

For convenience of presentation, we introduce some notations as follows. Let a quasigroup $(Q, *)$ of order 2^d be given by the multiplication scheme in Table 1,

Table 1: A quasigroup $(Q, *)$ of order 2^d

$*$	0	1	2	$\cdots$	2^d-1
0	$q_0^{(0)}$	$q_1^{(0)}$	$q_2^{(0)}$	$\cdots$	$q_{2^d-1}^{(0)}$
1	$q_0^{(1)}$	$q_1^{(1)}$	$q_2^{(1)}$	$\cdots$	$q_{2^d-1}^{(1)}$
2	$q_0^{(2)}$	$q_1^{(2)}$	$q_2^{(2)}$	$\cdots$	$q_{2^d-1}^{(2)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
2^d-1	$q_0^{(2^d-1)}$	$q_1^{(2^d-1)}$	$q_2^{(2^d-1)}$	$\cdots$	$q_{2^d-1}^{(2^d-1)}$

where $q_i^{(j)} \in Q$, $(i, j = 0, 1, \cdots, 2^d - 1)$. For given i and $\forall j \neq j'$, we have $q_i^{(j)} \neq q_i^{(j')}$; for given j and $\forall i \neq i'$, we have $q_i^{(j)} \neq q_{i'}^{(j)}$. Collect the elements of Table 1 into an vector

$$\Big(q_0^{(0)}, q_1^{(0)}, \cdots, q_{2^d-1}^{(0)}, q_0^{(1)}, q_1^{(1)}, \cdots, q_{2^d-1}^{(1)}, \\ \cdots, q_0^{(2^d-1)}, q_1^{(2^d-1)}, \cdots, q_{2^d-1}^{(2^d-1)}\Big)^T \tag{6}$$

and convert every element of the vector into a d-bit binary sequence, then we obtain a $2^{2d} \times d$ Boolean matrix $[b_1, \cdots, b_d]$, where every $b_s (s = 1, \cdots, d)$ is 2^{2d} dimensional column vector.

According to Lemma 2.1, whether a given quasigroup is a bilinear MQQ mainly lies in whether there is $2d$-ary bilinear Boolean function set $\{f_1, f_2, \cdots, f_d\}$, which have no terms of the form $x_s x_t$ with $s, t \leq d$ or $s, t > d$, satisfying Table 1. Note that, $\forall s (1 \leq s \leq d)$, bilinear Boolean function $f_s(x_1, \cdots, x_d, x_{d+1}, \cdots, x_{2d})$ can be written in the form

$$f_s = (1, x_1, \cdots, x_d)\mathscr{A}_s \begin{pmatrix} 1 \\ x_{d+1} \\ \vdots \\ x_{2d} \end{pmatrix}, (s = 1, 2, \cdots, d), \tag{7}$$

where $\mathscr{A}_s$ is a matrix of order $d+1$ over binary field GF(2). By (3), (7), and Table 1, when $(x_1, \cdots, x_d)$ and $(x_{d+1}, \cdots, x_{2d})^T$ in f_s are respectively assigned the ergodic d-bit binary sequence of $\{0, 1, \cdots, 2^d - 1\}$ in turn, we have

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix} \mathscr{A}_s \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 1 \end{pmatrix} = (p_{ji})_{2^d \times 2^d}, \tag{8}$$

where p_{ji} is the sth bit of the binary representation of $q_i^{(j)}$. Let

$$\mathscr{Q}_d = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}_{2^d \times (d+1)}, \tag{9}$$

then (8) can be rewritten by

$$\overline{vec}(\mathscr{Q}_d \mathscr{A}_s \mathscr{Q}_d^T) = b_s. \tag{10}$$

By Lemma 3.1, (10) can be reshaped into

$$(\mathscr{Q}_d \otimes \mathscr{Q}_d)\overline{vec}(\mathscr{A}_s) = b_s. \tag{11}$$

Thus, the given quasigroup in Table 1 is a bilinear MQQ iff there is a set of matrices $\{\mathscr{A}_1, \cdots, \mathscr{A}_d\}$ satisfying the following matrix equation

$$(\mathscr{Q}_d \otimes \mathscr{Q}_d)[\overline{vec}(\mathscr{A}_1), \cdots, \overline{vec}(\mathscr{A}_d)] = [b_1, \cdots, b_d]. \tag{12}$$

It is easy to see

$$\text{rank}(\mathscr{Q}_d \otimes \mathscr{Q}_d) = (d+1)^2.$$

So if the matrix equation (12) has solution, then the solution matrix must be unique. Furthermore, $\mathscr{A}_s (s = 1, \cdots, d)$ can be obtained by solving the matrix equation (12), and then $\{f_1, f_2, \cdots, f_d\}$ can be achieved by (7).

By now we have proved the following necessary and sufficient condition that a given quasigroup is a bilinear MQQ.

Theorem 3.1 *For a given quasigroup* $(Q,*)$ *of order* 2^d *and binary bijection* $Q \to \{0,1,\cdots,2^d-1\}$, $(Q,*)$ *is a bilinear MQQ of type* $Quad_{d-k}Lin_k$ *if and only if the matrix equation (12) has solution. Furthermore,* $f_s(s=1,2,\cdots,d)$ *obtained by (7) are just d Boolean polynomials of the bilinear MQQ, and their degrees are not more than 2.*

3.2 New algorithm for justifying and generating bilinear MQQs

According to Theorem 3.1, the problem of checking whether a given quasigroup is a bilinear MQQ is reduced to solving the matrix equation (12). Thus the solving algorithm can be chosen independently. In this subsection, we give such an efficient algorithm according to the special structure of Equation (12).

Using Gaussian elimination method, $\mathscr{Q}_d \otimes \mathscr{Q}_d$ can be simplified to a matrix with only $(d+1)^2$ nonzero rows. Noticing that the matrix $\mathscr{Q}_d \otimes \mathscr{Q}_d$ is fixed for all the quasigroups of order 2^d, our algorithm only needs to apply the above elimination steps to $[b_1,\cdots,b_d]$. Then we can judge whether the matrix equation (12) has solution and achieve the solution easily if it exists. Thus, for a given quasigroup and binary bijection, whether the quasigroup is a bilinear MQQ is determined and the corresponding bilinear Boolean functions are obtained if it is. The detailed algorithm is given as follows.

Algorithm 1: algorithm for checking whether a given quasigroup is a bilinear MQQ and generating the corresponding Boolean functions if it is.

```
1. Write the given quasigroup in a vector with the form of (6).
2. Convert every element of the vector into a d-bit binary sequence, then a $2^{2d} \times d$ Boolean matrix $[b_1,\cdots,b_d]$ is obtained, where every $b_s(s=1,\cdots,d)$ is $2^{2d}$ dimensional column vector. Let $r_{i+j\cdot 2^d}$ be on behalf of the $(1+i+j\cdot 2^d)$ th row vector of $[b_1,\cdots,b_d]$, where $i,j=0,1,\cdots,2^d-1$.
3. For $j=1,2,\cdots,2^d-1$
     for $i=0,1,\cdots,2^d-1$
       DO   $r_{i+j\cdot 2^d} := r_{i+j\cdot 2^d} \oplus r_i$,
         where $\oplus$ represents addition modulo 2
4. For $j=0,1,\cdots,2^d-1$
     if $j=2^{b_t}+2^{b_{t-1}}+\cdots+2^{b_1}$,
       where $d-1 \geq b_t > \cdots > b_1 \geq 0, t>1$
       for $i=0,1,\cdots,2^d-1$
         DO   $r_{i+j\cdot 2^d} := r_{i+j\cdot 2^d} \oplus r_{i+2^{b_t}\cdot 2^d}$
                $\oplus r_{i+2^{b_{t-1}}\cdot 2^d} \oplus \cdots \oplus r_{i+2^{b_1}\cdot 2^d}$
5. For $j=0,1,\cdots,2^d-1$
     if $j=2^b$ or 0, where $b=0,1,\cdots,d-1$
       for $i=1,2,\cdots,2^d-1$
         DO   $r_{i+j\cdot 2^d} := r_{i+j\cdot 2^d} \oplus r_{j\cdot 2^d}$
6. For $j=0,1,\cdots,2^d-1$
     if $j=2^b$ or 0, where $b=0,1,\cdots,d-1$
       for $i=0,1,\cdots,2^d-1$
         if $i=2^{b_t}+2^{b_{t-1}}+\cdots+2^{b_1}$,
           where $d-1 \geq b_t > \cdots > b_1 \geq 0, t>1$
           DO   $r_{i+j\cdot 2^d} := r_{i+j\cdot 2^d} \oplus r_{2^{b_t}+j\cdot 2^d}$
                  $\oplus r_{2^{b_{t-1}}+j\cdot 2^d} \oplus \cdots \oplus r_{2^{b_1}+j\cdot 2^d}$
7. For $i,j=0,1,\cdots,2^d-1$
     if $j \neq 2^b$ or 0, or if $i \neq 2^b$ or 0, $r_{i+j\cdot 2^d} \neq 0$
       output "no bilinear MQQ"
     else
       pass
8. For $j=0,2^{d-1},2^{d-2},\cdots,2,1$
     for $i=0,2^{d-1},2^{d-2},\cdots,2,1$
       output"$r_{i+j\cdot 2^d}$"
9. Write out $[\overline{vec}(\mathscr{A}_1),\cdots,\overline{vec}(\mathscr{A}_d)]$ and $\{\mathscr{A}_1,\cdots,\mathscr{A}_d\}$.
10. Compute $\{f_1,\cdots,f_d\}$ by (7).
```

3.3 The complexity of new algorithm

The algorithm amounts to perform a Gaussian elimination on a $2^{2d} \times d$ matrix $[b_1,\cdots,b_d]$. The operations used in the algorithm are only addition modulo 2 (XOR), and those operations are performed mainly in the third to sixth steps. Specifically, for a quasigroup of order 2^d, the third step takes $(2^{2d}-2^d)\cdot d$ operations, the fourth step takes $(2^{d-1}-1)\cdot 2^d \cdot d^2$ operations, the fifth step takes $(2^d-1)\cdot(d+1)\cdot d$ operations, and the sixth step takes $(2^{d-1}-1)\cdot(d+1)\cdot d^2$ operations. Thus, the total operations that the new algorithm needs is

$$d\cdot 2^{2d} + d^2\cdot 2^{2d-1} + (d^3+d^2)\cdot 2^{d-1} - d^3 - 2d^2 - d.$$

Suppose that a computer can execute 10^9 basic operations per second. Take the generation/judgment of MQQs of order 2^5 for an example, the new algorithm only needs to take 20140 times of operations, which cost about 20 microseconds. The speed is rather fast.

Remark 1 We mention a related work in [21], where a standard method was proposed to determine the unique Boolean representation of any function defined over a set of 2^d elements from the truth table. By splitting the multiplication table of a given quasigroup of order 2^d into d truth tables, d functions $f_i(x_1,\cdots,x_{2d})(i=1,\cdots,d)$ from $GF(2)^{2d}$ to $GF(2)$ corresponding to d truth tables can be found respectively. The standard method requires $O(d^2\cdot 2^{2d})$ operations to check whether a quasigroup is a bilinear MQQ. As a comparison, our algorithm needs only $O(d^2\cdot 2^{2d-1})$ operations.

Remark 2 Compared with the existing MQQs-generating methods which are based on sufficient conditions for a quasigroup to be a bilinear MQQ, our algorithm is based on such a necessary and sufficient condition. As a result, more bilinear MQQs may be obtained by our algorithm. For example, Chen [18] claims that they can find 256 bilinear MQQs of order 2^2, but using our algorithm we have checked that all 576 quasigroups of order 2^2 are bilinear MQQs. However, we also mention that, with the increasing of the order, the probability for a quasigroup to be a bilinear MQQ is decreasing. In contrast with the fact that all quasigroups of order 2^2 are bilinear MQQs, in a numerical experiment we found 9 bilinear MQQs by randomly trying 30000 quasigroups of order 2^3.

4 An example

In this section, we will use an example to show the validity of our results. The example appeared in [12]. Here we use it to show how Algorithm 1 works. A quasigroup $(Q,*)$ of order 2^3 and its corresponding 3-bit binary sequences are given in Table 2.

Table 2: A quasigroup $(Q,*)$ of order 2^3 and its binary sequences

$*$	0	1	2	3	4	5	6	7
0	3	2	6	7	1	0	4	5
1	5	3	7	1	0	6	2	4
2	0	6	3	5	4	2	7	1
3	6	7	2	3	5	4	1	0
4	7	1	4	2	3	5	0	6
5	1	0	5	4	2	3	6	7
6	4	5	1	0	6	7	3	2
7	2	4	0	6	7	1	5	3

$*$	000	001	010	011	100	101	110	111
000	011	010	110	111	001	000	100	101
001	101	011	111	001	000	110	010	100
010	000	110	011	101	100	010	111	001
011	110	111	010	011	101	100	001	000
100	111	001	100	010	011	101	000	110
101	001	000	101	100	010	011	110	111
110	100	101	001	000	110	111	011	010
111	010	100	000	110	111	001	101	011

By Theorem 3.1, the problem of finding 6-ary quadratic Boolean functions set $\{f_1, f_2, f_3\}$ satisfying (3) transforms into the problem of finding 4×4 matrices set $\{\mathscr{A}_1,\mathscr{A}_2,\mathscr{A}_3\}$. Further, whether $\{\mathscr{A}_1,\mathscr{A}_2,\mathscr{A}_3\}$ exists or not relies on whether the matrix equation

$$(\mathscr{Q}_3\otimes\mathscr{Q}_3)[\overline{vec}(\mathscr{A}_1),\overline{vec}(\mathscr{A}_2),\overline{vec}(\mathscr{A}_3)] = [b_1,b_2,b_3]$$

has solution. By our new algorithm for generating MQQs, we get that

$$[\overline{vec}(\mathscr{A}_1),\overline{vec}(\mathscr{A}_2),\overline{vec}(\mathscr{A}_3)] = \begin{pmatrix} 0&1&1\\ 0&1&0\\ 1&0&1\\ 0&0&1\\ 1&0&0\\ 1&1&0\\ 1&1&0\\ 1&1&1\\ 0&1&1\\ 1&1&0\\ 1&1&0\\ 1&1&1\\ 1&1&0\\ 1&1&1\\ 1&1&1\\ 1&1&1 \end{pmatrix}.$$

So, the corresponding bilinear Boolean functions are achieved as follows:

$$\begin{aligned} f_1 &= (1, x_1, x_2, x_3)\,\mathscr{A}_1 \begin{pmatrix}1\\x_4\\x_5\\x_6\end{pmatrix}\\ &= (1, x_1, x_2, x_3) \begin{pmatrix}0&0&1&0\\1&1&1&1\\0&1&1&1\\1&1&1&1\end{pmatrix} \begin{pmatrix}1\\x_4\\x_5\\x_6\end{pmatrix}\\ &= x_1+x_3+x_1x_4+x_2x_4+x_3x_4+x_5+x_1x_5+x_2x_5\\ &\quad +x_3x_5+x_1x_6+x_2x_6+x_3x_6, \end{aligned} \tag{13}$$

$$\begin{aligned} f_2 &= (1, x_1, x_2, x_3)\,\mathscr{A}_2 \begin{pmatrix}1\\x_4\\x_5\\x_6\end{pmatrix}\\ &= (1, x_1, x_2, x_3) \begin{pmatrix}1&1&0&0\\0&1&1&1\\1&1&1&1\\1&1&1&1\end{pmatrix} \begin{pmatrix}1\\x_4\\x_5\\x_6\end{pmatrix}\\ &= 1+x_2+x_3+x_4+x_1x_4+x_2x_4+x_3x_4+x_1x_5\\ &\quad +x_2x_5+x_3x_5+x_1x_6+x_2x_6+x_3x_6, \end{aligned} \tag{14}$$

$$\begin{aligned} f_3 &= (1, x_1, x_2, x_3)\,\mathscr{A}_3 \begin{pmatrix}1\\x_4\\x_5\\x_6\end{pmatrix}\\ &= (1, x_1, x_2, x_3) \begin{pmatrix}1&0&1&1\\0&0&0&1\\1&0&0&1\\0&1&1&1\end{pmatrix} \begin{pmatrix}1\\x_4\\x_5\\x_6\end{pmatrix}\\ &= 1+x_2+x_3x_4+x_5+x_3x_5+x_6+x_1x_6+x_2x_6\\ &\quad +x_3x_6. \end{aligned} \tag{15}$$

This is a bilinear MQQ of type $Quad_3Lin_0$. The above result is coincident with the result in [12].

5 Conclusions

This paper reports new theory and algorithm of justifying and generating bilinear Multivariate Quadratic Quasigroups. We first establish a necessary and sufficient condition to test whether a quasigroup given by its multiplication table is a bilinear MQQ. Then, based on this condition, we propose an algorithm to judge whether the given quasigroup is a bilinear MQQ and obtain the corresponding Boolean functions if it is. Moreover, an example is given to show the validity of our results.

The new algorithm has a better running time than the standard method [21] and theoretically can obtain all the bilinear MQQs. The main problem for our algorithm is that the probability for a quasigroup to be a bilinear MQQ is decreasing with the increasing of the order. Thus, our algorithm can be a useful complement to the existing MQQs-generating algorithms.

Acknowledgement

The authors wish to thank the anonymous reviewers for their helpful comments and valuable suggestions regarding this paper. This work was supported in part by the Fundamental Research Funds for the Central Universities (no. 2012TD032).

References

[1] R. Rivest, A. Shamir, and L. Adleman, Communications of the ACM, **21**, 120 (1978).
[2] N. Koblitz, Elliptic curve cryptosystems, Mathematics of Computation, **48**, 203 (1987).
[3] P. W. Shor, Proc. 35th Annual Symposium on Foundation of Computer Science, 124 (1994).
[4] R. Merkle, LNCS, **435**, 218 (1989).
[5] R. J. McEliece, DSN Progress Report, **114**, 42-44 (1978).
[6] A. K. Lenstra, H. W. Lenstra, L. Lovasz, Mathematische Annalen, **261**, 515 (1982).
[7] H. Fell and W. Diffie, LNCS, **218**, 340 (1986).
[8] H. Imai and T. Matsumoto, LNCS, **29**, 108 (1985).
[9] A. Shamir, LNCS, **773**, 1 (1993).
[10] J. Patarin, LNCS, **1070**, 33 (1996).
[11] A. Kipnis, J. Patarin, and L. Goubin, LNCS, **1592**, 206 (1999).
[12] D. Gligoroski, S. Markovski, and S. J. Knapskog, Cryptology ePrint Archive, Report 320, (2008).
[13] M. E. Hadedy, D. Gligoroski, and S. J. Knapskog, Proc. Reconfigurable Computing and FPGAs, 427 (2008).
[14] R. J. M. Maia, P. S. L. M. Barreto, and B. T. D. Oliveira, LNCS, **6480**, 64 (2010).
[15] M. S. Mohamed, J. T. Ding, J. Buchmann, and F. Werner, LNCS, **5888**, 392 (2009).
[16] D. Gligoroski, R. S. Ødegård, R. E. Jensen, L. Perret, J. C. Faugre, S. J. Knapskog, and S. Markovski, LNCS, **7222**, 184 (2012).
[17] R. Ahlawat, K. Gupta, and S. K. Pal, Proc. Mathematics in Defence, (2009).
[18] Y. L. Chen, S. J. Knapskog, and D. Gligoroski, Proc. 6th Information Security and Cryptology, (2010).
[19] S. Samardjiska, S. Markovski, and D. Gligoroski, Proc. Symbolic Computation and Cryptography, 117 (2010).
[20] A. Christov, In: Charles University in Prague, Ph.D. (2009).
[21] C. J. A. Jansen, In: Technical University of Delft, Ph.D. (1989).
[22] J.C. Faugère, R. S. Ødegård, L. Perret and D. Gligoroski, LNCS **6467**, 169 (2010).
[23] G. H. Golub and C. F. V. Loan, Matrix Computations. In: Johns Hopkins Studies in the Mathematical Sciences, (Johns Hopkins University Press, USA,) (1996).

Ying Zhang is with Department of Mathematics, Dalian Maritime University. Her research interests include algebra, cryptography, and coding theory.

Huisheng Zhang received the MS degree from Xiamen University in 2003 and PhD degree from Dalian University of Technology in 2009. He is currently an associate professor of Dalian Maritime University. His research interests include neural networks, signal processing, learning theory, and coding theory.