SBIM(Q) - a Multivariate Polynomial Trapdoor Function over the Field of Rational Numbers

Smile Markovski[1], Aleksandra Mileva[2], and Vesna Dimitrova[1]

[1]Faculty of Computer Science and Engineering,
University "Ss Cyril and Methodius", Skopje, Republic of Macedonia
{smile.markovski,vesna.dimitrova}@finki.edu.mk,
[2]Faculty of Computer Science,
University "Goce Delčev", Štip, Republic of Macedonia
aleksandra.mileva@ugd.edu.mk

Abstract. In this paper we define a trapdoor function called SBIM(Q) by using multivariate polynomials over the field of rational numbers $\mathbb{Q}$. The public key consists of $2n$ multivariate polynomials with $3n$ variables $y_1, \ldots, y_n,\ z_1, \ldots, z_{2n}$. The y_i variables take care for the information content, while the z_i variables are for redundant information. Thus, for encryption of a plaintext of n rational numbers, a ciphertext of $2n$ rational numbers is used. The security is based on the fact that there are infinitely many solutions of a system with $2n$ polynomial equations of $3n$ unknowns.
The public key is designed by quasigroup transformations obtained from quasigroups presented in matrix form. The quasigroups presented in matrix form allow numerical as well as symbolic computations, and here we exploit that possibility. The private key consists of several $1 \times n$ and $n \times n$ matrices over $\mathbb{Q}$, and one $2n \times 2n$ matrix.

Keywords: trap-door function, public key, private key, encryption, decryption, matrix form of quasigroup, quasigroup transformations, bipermutations

1 Introduction

A function $f : A \to B$ is said to be one way function if for each $x \in A$ it can be effectively computed the image $f(x)$, but for any $y \in B$ finding a preimage $x \in A$, such that $f(x) = y$, has to be computationally infeasible. In other words, computation of $f(x)$ can be realized with an algorithm of polynomial complexity, and computation of a preimage of y can be realized only with algorithms of exponential complexity. A trapdoor function is a one way function such that the preimage can be effectively computed when some additional (usually secret) information is known. A trapdoor function has important applications in cryptography, especially for designing public key infrastructures, and for many others cryptographic primitives. For that purposes a trapdoor function has to be real time computable and it has to use small instance of the computer memory.

There are not many trapdoor functions. The most popular is the RSA [7]. In this paper we present a new trapdoor function, called SBIM(Q). It is defined over the field of rational numbers $\mathbb{Q}$, but its construction can be used over any infinite field. The public key of SBIM(Q) consists of a set of multivariate polynomials. The first such scheme was constructed by Matsumoto and Imai [4] in 1985. After that, several other systems of that type were designed, unfortunately almost all of them were broken. An excellent survey article for these schemes is written by Wolf and Preneel [9].

The construction of SBIM(Q) is based on so called quasigroup string transformations. In Section 2 we give the needed definitions and properties (without proofs), in order to make clear the construction of SBIM(Q). For more details one may consult [5], [8].

The private key of SBIM(Q) consists of several nonsingular matrices, and the public key consists of a system of multivariate polynomial equations. The encryption is by evaluating the polynomial expressions, and decryption is by several multiplications of vectors by matrices.

In Section 2 we define the notion of quasigroup bi-permutations and the string transformations defined by them. In Section 3 the construction of SBIM(Q) is given, as well as the encryption and decryption functions. Section 4 contains an example in all details. Some issues on security, implementation, and optimization are discussed in Section 5. Section 6 contains conclusions and future work.

2 Quasigroup bi-permutations

A quasigroup is a groupoid (G, f), where G is a nonempty set and $f : G \times G \to G$ is a binary operation that satisfies the property each one of the equations $f(a, x) = b$ and $f(y, a) = b$ to have a unique solution x, respectively y; then we also say that f is a quasigroup operation. When G is a finite set, the main body of the Cayley table of the quasigroup (G, f) represents a Latin square, i.e., a matrix with rows and columns that are permutations of G. It follows that if we fix one component of f, then the mappings $f(x, \cdot) : \{x\} \times G \to G$ and $f(\cdot, y)$ are permutations of the set G. That is why we say that the binary operation f is a (quasigroup) bi-permutation.

Given a quasigroup (G, f) two new operations $f^{(23)}$ and $f^{(13)}$, called parastrophes, can be derived from the operation f as follows:

$$f(x, y) = z \;\Leftrightarrow\; f^{(23)}(x, z) = y \;\Leftrightarrow\; f^{(13)}(z, y) = x. \tag{1}$$

Then $(G, f^{(23)})$ and $(G, f^{(13)})$ are also quasigroups and the algebra $(G, f, f^{(23)}, f^{(13)})$ satisfies the identities

$$\begin{array}{ll} f^{(23)}(x, f(x, y)) = y, & f(x, f^{(23)}(x, y)) = y, \\ f^{(13)}(f(x, y), y) = x, & f(f^{(13)}(x, y), y) = x. \end{array} \tag{2}$$

In the sequel, when there will be no confusion, we will write $a_1 a_2 \ldots a_n$ instead of $(a_1, a_2, \ldots, a_n)$.

Quasigroup string transformations are defined on the set $G^n = \{a_1a_2\ldots \ldots a_n |\ a_i \in G\},\ n \geq 1$, by using quasigroup operations $f_1, f_2, \ldots, f_n$ on the set G. Some of the operations f_i, and even all of them, can be equal. We define four types of transformations. Let $l \in G$ be a fixed element, called a leader. For every $a_i, b_i \in G$, $e-$ and $d-$transformations are defined as follows:

$$\begin{aligned} e_l(a_1a_2\ldots a_n) = b_1b_2\ldots b_n \Leftrightarrow b_{i+1} = f_{i+1}(b_i, a_{i+1}),\\ d_l(a_1a_2\ldots a_n) = b_1b_2\ldots b_n \Leftrightarrow b_{i+1} = f_{i+1}(a_i, a_{i+1}), \end{aligned} \tag{3}$$

for each $i = 0, 1, \ldots, n-1$, where $b_0 = a_0 = l$. The $e'-$ and $d'-$transformations are defined similarly, in reverse way:

$$\begin{aligned} e'_l(a_1a_2\ldots a_n) = b_1b_2\ldots b_n \Leftrightarrow b_i = f_{n+1-i}(a_i, b_{i+1}),\\ d'_l(a_1a_2\ldots a_n) = b_1b_2\ldots b_n \Leftrightarrow b_i = f_{n+1-i}(a_i, a_{i+1}), \end{aligned} \tag{4}$$

for each $i = n, n-1, \ldots, 2, 1$, where $b_{n+1} = a_{n+1} = l$.

By using the identities (2), we have that

$$d_l(e_l(a_1\ldots a_n)) = e_l(d_l(a_1\ldots a_n)) = a_1\ldots a_n$$

when e_l is defined by the sequence of quasigroup operations $f_1, \ldots, f_{n-1}, f_n$ and d_l is defined by the sequence of quasigroup operations $f_1^{(23)}, f_2^{(23)}, \ldots, f_n^{(23)}$. Also, when e'_l is defined by $f_1, \ldots, f_{n-1}, f_n$ and d'_l is defined by $f_1^{(13)}, f_2^{(13)}, \ldots \ldots, f_n^{(13)}$ we have $d'_l(e'_l(a_1\ldots a_n)) = e'_l(d'_l(a_1\ldots a_n)) = a_1\ldots a_n$ as well. This means that e_l, d_l, e'_l and d'_l are permutations on G^n, such that e_l, d_l and e'_l, d'_l are mutually inverse.

The next theorem shows that $e-$ and $e'-$ transformations are useful for obtaining pseudo-randomness of the transformed strings.

Theorem 1. [5] *Consider an arbitrary string $\alpha = a_1a_2\ldots a_k$ where $a_i \in G$, and let β be obtained after k applications of $e-$ or $e'-$transformations on α. If k is an enough large integer then, for each $1 \leq t \leq k$, the distribution of substrings of β of length t is uniform.* (We note that for $t > k$ the distribution of substrings of β of length t is not uniform.)

A construction of quasigroup bi-permutations is given by the next theorem.

Theorem 2. *Let A and B be nonsingular $m \times m$ matrices and let C be $1 \times m$ matrix over a field F. Then the mapping*

$$f(a_1, \ldots, a_m; b_1\ldots, b_m) = (a_1, \ldots, a_m)\cdot A + (b_1, \ldots, b_m)\cdot B + C, \tag{5}$$

where $a_i, b_i \in F$, is a quasigroup bi-permutation on F^m. The parastrophic operations $f^{(13)}$ and $f^{(23)}$ of f are defined as follows:

$$\begin{aligned} f^{(13)}(\mathbf{x};\mathbf{y}) = \mathbf{x}\cdot A^{-1} + \mathbf{y}\cdot(-B\cdot A^{-1}) - C\cdot A^{-1},\\ f^{(23)}(\mathbf{x};\mathbf{y}) = \mathbf{x}\cdot(-A\cdot B^{-1}) + \mathbf{y}\cdot B^{-1} - C\cdot B^{-1}, \end{aligned} \tag{6}$$

where $\mathbf{x} = a_1\ldots a_m,\ \mathbf{y} = b_1\ldots b_m \in F^m$.

The presentation of the bi-permutations f in matrix form (5) allows f to be used for symbolic computations. Namely, instead of elements $a_i, b_i \in F$, we can use any expressions E_i, H_i valuable in F for getting a new expression $f(E_1, \ldots, E_m; H_1, \ldots, H_m)$. Thus, in the sequel we will use polynomials E_i, H_i and then $f(E_1, \ldots, E_m; H_1, \ldots, H_m)$ will be polynomials too.

3 Construction of SBIM(Q)

Further on we work with the field of rational numbers $\mathbb{Q}$. We define SBIM(Q) over $\mathbb{Q}$ in several steps. We use as a parameter a positive integer n.

Choosing polynomials. Denote by $y_1, \ldots, y_n, z_1, \ldots, z_{2n}$ variables on $\mathbb{Q}$. Choose randomly n multivariate polynomials $Y_1, Y_2, \ldots, Y_n$ on $\mathbb{Q}$ with variables $y_1, \ldots, y_n$ such that the system of equations

$$\begin{aligned} Y_1(y_1, \ldots, y_n) &= b_1, \\ Y_2(y_1, \ldots, y_n) &= b_2, \\ &\ldots\ldots\ldots\ldots \\ Y_n(y_1, \ldots, y_n) &= b_n, \end{aligned} \tag{7}$$

for any given $b_i \in \mathbb{Q}$, has unique solution $y_1 = a_1, \ldots, y_n = a_n,\ a_i \in \mathbb{R}$. (As usual, $\mathbb{R}$ denotes the field of real numbers.) One trivial way to choose the polynomials Y_i is by taking a nonsingular $n \times n$ matrix S over $\mathbb{Q}$ and then $(Y_1, Y_2, \ldots, Y_n) = (y_1, \ldots, y_n) \cdot S$. Another simple example, for instance for $n = 4$, is by taking $Y_1 = y_1 + 3y_2,\ Y_2 = y_3^3 - 4,\ Y_3 = y_4 - y_1,\ Y_4 = y_4^5$.

Next, we choose randomly n multivariate polynomials $Y_{n+1}, Y_{n+2}, \ldots, Y_{2n}$ on $\mathbb{Q}$ with variables $y_1, \ldots, y_n, z_1, \ldots, z_{2n}$.

Applying transformation. Let π be a random permutation on the set of integers $\{1, 2, \ldots, 2n\}$, and let denote $(X_1, \ldots, X_{2n}) = (Y_{\pi(1)}, \ldots, Y_{\pi(2n)})$, $\mathbf{x} = (X_1, X_2, \ldots, X_n)$ and $\mathbf{y} = (X_{n+1}, X_{n+2}, \ldots, X_{2n})$. We apply $e-$ and $e'-$transformations on $(X_1, X_2, \ldots, X_{2n})$ for producing new polynomials as follows.

The first $e-$transformation. At first we use an e-transformation defined by a random leader $\mathbf{l}_1 = (l_{11}, \ldots, l_{1n}) \in \mathbb{Q}^n$ and two quasigroup bi-permutations f_1 and f_2 defined by random nonsingular $n \times n$ matrices A_i, B_i as following:

$$f_1(\mathbf{l}_1; \mathbf{x}) = \mathbf{l}_1 \cdot A_1 + \mathbf{x} \cdot B_1, \quad f_2(\mathbf{x}'; \mathbf{y}) = \mathbf{x}' \cdot A_2 + \mathbf{y} \cdot B_2,$$

where $\mathbf{x}' = f_1(\mathbf{l}_1; \mathbf{x})$. Let denote $\mathbf{y}' = f_2(\mathbf{x}'; \mathbf{y})$.

The second $e'-$transformation. As second transformation we use an e'-transformation defined by a random leader $\mathbf{l}_2 = (l_{21}, \ldots, l_{2n})$ and two quasigroup bi-permutations f_3 and f_4 defined by random nonsingular $n \times n$ matrices A_i, B_i as following:

$$f_3(\mathbf{y}'; \mathbf{l}_2) = \mathbf{y}' \cdot A_3 + \mathbf{l}_2 \cdot B_3, \quad f_4(\mathbf{x}'; \mathbf{y}'') = \mathbf{x}' \cdot A_4 + \mathbf{y}'' \cdot B_4,$$

where $\mathbf{y}'' = f_3(\mathbf{y}'; \mathbf{l}_2)$. Let denote $\mathbf{x}'' = f_4(\mathbf{x}'; \mathbf{y}'')$.

The next transformations. The previous two transformations are obligatory. We can make p ($p \geq 0$) new $e-$ or $e'-$transformations in the same manner as before. It is choice of ours what type of transformation and in which order will be applied. We start by transforming the $n-$tuples of polynomials $\mathbf{x}'', \mathbf{y}''$. For $p = 1$ we can take either an $e-$ or an $e'-$transformation, for $p = 2$ we can take either $e-, e-$, or $e-, e'-$, or $e'-, e-$ or $e'-, e'-$ transformations, and so on. For that purposes we have to take new random leaders $\mathbf{l}_i$ and new random matrices A_i, B_i. (In that case more secure system will be obtained, the price being paid with more complex private key. If we want to keep the private key enough small, we can reuse the previous bi-permutations $f_1, \ldots, f_4$ and/or leaders.)

The public key and the encryption. Let $p \geq 0$ additional transformations were applied. Then the last transformation was done by some leader $\mathbf{l}_{2+p}$ and bi-permutations f_{3+p} and f_{4+p}, applied on some $n-$tuples of polynomials $\mathbf{u}$ and $\mathbf{v}$. In the case when the last transformation was an $e-$transformation, we denote $(Z_1, \ldots, Z_n) = f_{3+p}(\mathbf{l}_{2+p}; \mathbf{u})$ and $(Z_{n+1}, \ldots, Z_{2n}) = f_{4+p}((Z_1, \ldots, Z_n); \mathbf{v})$. In the case of an $e'-$transformation, we denote $(Z_1, \ldots, Z_n) = f_{3+p}(\mathbf{v}; \mathbf{l}_{2+p})$ and $(Z_{n+1}, \ldots, Z_{2n}) = f_{4+p}(\mathbf{u}; (Z_1, \ldots, Z_n))$. We choose randomly a nonsingular $2n \times 2n$ matrix R on $\mathbb{Q}$ and we denote $(A_1, A_2, \ldots, A_{2n}) = (Z_1, \ldots, Z_{2n}) \cdot R$. Then the **public key** consists of the $2n-$tuple of polynomials $(A_1, A_2, \ldots, A_{2n})$. Note that each $A_i = A_i(y_1, \ldots, y_n, z_1, \ldots, z_{2n})$ is a multivariate polynomial on $\mathbb{Q}$ with $3n$ variables.

The encryption of a message $M = (m_1, m_2, \ldots, m_n) \in \mathbb{Q}^n$ is as follows. Choose random elements $r_1, r_2, \ldots, r_{2n} \in \mathbb{Q}$ and evaluate the polynomials A_i by taking $y_j = m_j,\ z_k = r_k$. The **ciphertext** is the $2n-$tuple $(c_1, c_2, \ldots, c_{2n})$, where

$$\begin{aligned} c_1 &= A_1(m_1, \ldots, m_n, r_1, \ldots, r_{2n}), \\ c_2 &= A_2(m_1, \ldots, m_n, r_1, \ldots, r_{2n}), \\ &\ldots\ldots\ldots\ldots \\ c_{2n} &= A_{2n}(m_1, \ldots, m_n, r_1, \ldots, r_{2n}). \end{aligned}$$

We note that a plaintext of n rational numbers is encrypted with $2n$ rational numbers, we can say that the information efficiency is 50%.

The private key. The private key consists of the permutation π, of all leaders $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3, \ldots$ and all matrices A_i, B_i, R used for producing the public key. The leaders and the matrices may not be all different. It is obligatory to be used at least two different leaders and at least four different matrices for the bi-permutations. By choosing suitable number of leaders and matrices an optimal private key can be designed.

Decryption. The decryption is done by applying $d-$ and $d'-$transformations. The public key was built up in an upward-down way. The decryption is in downward-up way.

Given a ciphertext $(c_1, \ldots, c_{2n})$, we first apply the inverse matrix R^{-1} and obtain $(t_1, \ldots, t_{2n}) = (c_1, \ldots, c_{2n}) \cdot R^{-1}$. After that we produce two n-tuples $C_1 = (t_1, \ldots\ \ldots, t_n)$ and $C_2 = (t_{n+1}, \ldots, t_{2n})$. We have to possible cases.

The case of a $d-$transformation. Let the last applied transformation for obtaining the polynomials $(Z_1, Z_2, \ldots, Z_{2n})$ have been an $e-$transformation, defined by a leader $\mathbf{l}_{2+p}$ and bi-permutations f_{3+p} and f_{4+p}. Then we apply a $d-$transformation on C_1, C_2 by using the leader $\mathbf{l}_{2+p}$ and the parastrophes $f_{3+p}^{(23)}$ and $f_{4+p}^{(23)}$. We obtain two new n-tuples of rational numbers D_1 and D_2 as follows:

$$D_1 = f_{3+p}^{(23)}(\mathbf{l}_{2+p}; C_1), \quad D_2 = f_{4+p}^{(23)}(C_1; C_2).$$

The case of a $d'-$transformation. Let the last applied transformation for obtaining the polynomials $(Z_1, Z_2, \ldots, Z_{2n})$ have been an $e'-$transformation, defined by a leader $\mathbf{l}_{2+p}$ and bi-permutations f_{3+p} and f_{4+p}. Then we apply a $d'-$transformation on C_1, C_2 by using the leader $\mathbf{l}_{2+p}$ and the parastrophes $f_{3+p}^{(13)}$ and $f_{4+p}^{(13)}$. We obtain two new n-tuples of rational numbers D_1 and D_2 as follows:

$$D_1 = f_{3+p}^{(13)}(C_2; \mathbf{l}_{2+p}), \quad D_2 = f_{4+p}^{(13)}(C_1; C_2).$$

What we have done is replacing the $n-$tuples of polynomials with $n-$tuples of rational numbers. We apply these $d-$ or $d'-$transformations from downward-up way and after each application we will obtain n-tuples of rational numbers. Eventually, at the end, instead of starting n-tuples of polynomials $\mathbf{x} = (X_1, \ldots, X_n)$ and $\mathbf{y} = (X_{n+1}, \ldots, X_{2n})$ we obtain n-tuples of rational numbers $\mathbf{a} = (a_1, \ldots, a_n)$ and $\mathbf{b} = (a_{n+1}, \ldots, a_{2n})$.

Finally, we apply the inverse permutation π^{-1} on $(a_1, a_2, \ldots, a_{2n})$ to get $(b_1 = a_{\pi^{-1}(1)}, \ldots, b_{2n} = a_{\pi^{-1}(2n)})$. Then we form the system of equations (7) with the rational numbers $b_1, b_2, \ldots, b_n$. The solution of the system (7) is the message $M = (m_1, m_2, \ldots, m_n)$.

4 Example

Here we give an example to illustrate the previous constructions.

We take $n = 2$ and so we use the variables y_1, y_2, z_1, z_2, z_3 and z_4. We choose polynomials $Y_1 = y_1 - 2y_2$, $Y_2 = y_1^3 - 2$, $Y_3 = y_1^3 + y_2^2 + z_1^3 + 3y_1z_4 + 2y_2z_2 + y_1 + y_2 - z_1 - z_4$, $Y_4 = -z_2^4 - y_2z_1 + 2y_1 + z_1 - z_3 - z_4$ and a permutation $\pi = (3, 2, 1, 4)$. We have $X_1 = Y_3$, $X_2 = Y_2$, $X_3 = Y_1$, $X_4 = Y_4$ and $\mathbf{x} = (X_1, X_2)$, $\mathbf{y} = (X_3, X_4)$. We also take $p = 0$.

For the transformations we use the leaders $\mathbf{l}_1 = (-1, 1)$ and $\mathbf{l}_2 = (2, -1)$ and the following bi-permutations:

$$f_1(\mathbf{x}, \mathbf{y}) = \mathbf{x}\begin{pmatrix} 1 & -1 \\ 2 & -1 \end{pmatrix} + \mathbf{y}\begin{pmatrix} 0 & 3 \\ 1 & 0 \end{pmatrix}, \quad f_2(\mathbf{x}, \mathbf{y}) = \mathbf{x}\begin{pmatrix} -1 & 0 \\ 1 & 1 \end{pmatrix} + \mathbf{y}\begin{pmatrix} 2 & 1 \\ -1 & -1 \end{pmatrix},$$

$$f_3(\mathbf{x}, \mathbf{y}) = \mathbf{x}\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} + \mathbf{y}\begin{pmatrix} 3 & 5 \\ 1 & 2 \end{pmatrix}, \quad f_4(\mathbf{x}, \mathbf{y}) = \mathbf{x}\begin{pmatrix} 1 & -2 \\ 1 & 1 \end{pmatrix} + \mathbf{y}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

The first $e-$transformation is defined by $\mathbf{l}_1$, f_1 and f_2, and from $\mathbf{x}, \mathbf{y}$ we obtain $\mathbf{x}', \mathbf{y}'$ as follows:

$\mathbf{x}' = f_1(\mathbf{l}_1;\ \mathbf{x}) = f_1(-1,1;\ X_1,X_2) = (1,0) + (X_2,3X_1) = (1+X_2,3X_1),$

$\mathbf{y}' = f_2(1+X_2,3X_1;\ X_3,X_4) = (-1-X_2+3X_1,3X_1)+(2X_3-X_4,X_3-X_4)$

$= (-1+3X_1-X_2+2X_3-X_4,3X_1+X_3-X_4).$

Next, we have to apply an $e'-$transformation and we use $\mathbf{l}_2$, f_3 and f_4. Then $\mathbf{x}',\mathbf{y}'$ will be transformed into $\mathbf{x}'',\mathbf{y}''$, where

$\mathbf{y}'' = f_3(\mathbf{y}';\mathbf{l}_2) = f_3(-1+3X_1-X_2+2X_3-X_4,3X_1+X_3-X_4;\ 2,-1)$

$= (1-3X_1+X_2-2X_3+X_4,-3X_1-X_3+X_4)+(5,8)$

$= (6-3X_1+X_2-2X_3+X_4,\ 8-3X_1-X_3+X_4),$

$\mathbf{x}'' = f_4(\mathbf{x}';\mathbf{y}'') = f_4(1+X_2,3X_1;\ 6-3X_1+X_2-2X_3+X_4,8-3X_1-X_3+X_4)$

$= (1+3X_1+X_2,-2(1+X_2)+3X_1)+$

$+(6-3X_1+X_2-2X_3+X_4,8-3X_1-X_3+X_4)$

$= (7+2X_2-2X_3+X_4,\ 6-2X_2-X_3+X_4).$

We obtain $Z_1 = 7+2X_2-2X_3+X_4$, $Z_2 = 6-2X_2-X_3+X_4$, $Z_3 = 6-3X_1+X_2-2X_3+X_4$, $Z_4 = 8-3X_1-X_3+X_4$. Let take a 4×4 nonsingular matrix $R = \begin{pmatrix} 2 & -1 & 0 & -3 \\ 1 & 2 & -1 & -1 \\ 0 & 3 & 2 & 0 \\ -3 & -1 & -1 & 4 \end{pmatrix}$ and compute $(A_1,\ldots,A_{2n}) = (Z_1,\ldots,Z_{2n})\cdot R$.

We get the public key

$A_1 = -4+9X_1+2X_2-2X_3,$

$A_2 = 15-6X_1-3X_2-5X_3+3X_4,$

$A_3 = -2-3X_1+4X_2-2X_3,$

$A_4 = 5-12X_1-4X_2+3X_3.$

After replacement of variables we get the final form of the public key:

$A_1 = -8+7y_1+13y_2-9z_1-9z_4+11y_1^3+9y_2^3+9z_1^3+27y_1z_4+18y_2z_2,$

$A_2 = 21-5y_1+4y_2+9z_1-3z_3+3z_4-9y_1^3-6y_2^3-6z_1^3-3z_2^4-18y_1z_4-3y_2z_1-12y_2z_2,$

$A_3 = -10-5y_1+y_2+3z_1+3z_4+y_1^3-3y_2^3-3z_1^3-9y_1z_4-6y_2z_2,$

$A_4 = 13-9y_1-18y_2+12z_1+12z_4-16y_1^3-12y_2^3-12z_1^3-36y_1z_4-24y_2z_2.$

Given a message $M(1,1)$ (so $y_1 = y_2 = 1$), we choose redundant elements $z_1 = z_2 = z_3 = 0$, $z_4 = 1$ and we compute the ciphertext (c_1,c_2,c_3,c_4) as following:

$c_1 = A_1(1,1,0,0,0,1) = 50,$

$c_2 = A_2(1,1,0,0,0,1) = -10,$

$c_3 = A_3(1,1,0,0,0,1) = -22,$

$c_4 = A_4(1,1,0,0,0,1) = -66.$

Hence, $(50,-10,-22,-66)$ is the ciphertext of the plaintext $(1,1)$.

Let *Allice* obtained the ciphertext $(50,-10,-22,-66)$. She does not know the message and the redundant elements. *Allice* will make a decryption of the ciphertext as follows. First she compute $(50,-10,-22,-66)\cdot R^{-1} = (8,10,-10,-8)$ and obtain that $(t_1,t_2,t_3,t_4) = (8,10,-10,-8)$.

She also needs the parastrophes $f_1^{(23)}$, $f_2^{(23)}$, $f_3^{(13)}$, $f_4^{(13)}$ for decryption purposes. Recall that the parastrophes of $f(\mathbf{x};\mathbf{y}) = \mathbf{x}A + \mathbf{y}B$ are $f^{(13)}(\mathbf{x};\mathbf{y}) = \mathbf{x}A^{-1} + \mathbf{y}(-BA^{-1})$ and $f^{(23)}(\mathbf{x};\mathbf{y}) = \mathbf{x}(-AB^{-1}) + \mathbf{y}B^{-1}$. Then the needed parastrophes are the following:

$$f_1^{(23)}(\mathbf{x};\mathbf{y}) = \mathbf{x}\begin{pmatrix}1/3 & -1\\ 1/3 & -2\end{pmatrix} + \mathbf{y}\begin{pmatrix}0 & 1\\ 1/3 & 0\end{pmatrix},$$

$$f_2^{(23)}(\mathbf{x};\mathbf{y}) = \mathbf{x}\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix} + \mathbf{y}\begin{pmatrix}1 & 1\\ -1 & -2\end{pmatrix},$$

$$f_3^{(13)}(\mathbf{x};\mathbf{y}) = \mathbf{x}\begin{pmatrix}-1 & 0\\ 0 & -1\end{pmatrix} + \mathbf{y}\begin{pmatrix}3 & 5\\ 1 & 2\end{pmatrix},$$

$$f_4^{(13)}(\mathbf{x};\mathbf{y}) = \mathbf{x}\begin{pmatrix}1/3 & 2/3\\ -1/3 & 1/3\end{pmatrix} + \mathbf{y}\begin{pmatrix}-1/3 & -2/3\\ 1/3 & -1/3\end{pmatrix}.$$

From the 4-tuple $(t_1, t_2, t_3, t_4) = (8, 10, -10, -8)$ *Allice* makes the pairs $C_1 = (8, 10)$ and $C_2 = (-10, -8)$. Since the last applied transformation for producing the public key was an $e'-$transformation, she applies on C_1 and C_2 the $d'-$transformation defined by $\mathbf{l}_2$, $f_3^{(13)}$ and $f_4^{(13)}$. She computes

$D_2 = f_3^{(13)}(C_2; \mathbf{l}_2) = f_3^{(13)}(10, -8;\ 2, -1) = (15, 16)$ and
$D_1 = f_4^{(13)}(C_1; C_2) = f_4^{(13)}(8, 10;\ 10, -8) = (0, 18)$.

The previous transformation before the last one was an $e-$transformation. So, *Allice* applies the $d-$transformation defined by $\mathbf{l}_1$, $f_1^{(23)}$ and $f_2^{(23)}$ on D_1 and D_2 and she computes

$\mathbf{x} = f_1^{(23)}(\mathbf{l}_1; D_1) = f_1^{(23)}(-1, 1; 0, 18) = (6, -1)$,
$\mathbf{y} = f_2^{(23)}(D_1; D_2) = f_2^{(23)}(0, 18;\ 15, 16) = (-1, 1)$.

Then *Allice* knows that $X_1 = 6$, $X_2 = -1$, $X_3 = -1$, $X_4 = 1$ and by the inverse permutation π^{-1} she get that $X_3 = Y_1 = y_1 - 2y_2 = -1$ and $X_2 = Y_2 = y_1^3 - 2 = -1$. From that *Allice* discovers the sent message M, since $(y_1 = 1, y_2 = 1)$ is the solution of the system of polynomial equations

$$\begin{cases} y_1^3 - 2 = -1, \\ y_1 - 2y_2 = -1. \end{cases}$$

5 Discussion

Here we will discuss security, implementation, optimization and other issues.

5.1 Security

The security of the SBIM(Q) is based mainly on the infinity number of solutions of a system of $2n$ polynomial equations with $3n$ unknowns. If an attacker have

a ciphertext $(c_1, c_2, \ldots, c_{2n})$, he/she can make the system

$$\begin{aligned} A_1(y_1, \ldots, y_n, z_1, \ldots, z_{2n}) &= c_1, \\ A_2(y_1, \ldots, y_n, z_1, \ldots, z_{2n}) &= c_2, \\ &\ldots\ldots\ldots\ldots \\ A_{2n}(y_1, \ldots, y_n, z_1, \ldots, z_{2n}) &= c_{2n}. \end{aligned} \tag{8}$$

When the public key is carefully produced, the system (8) has infinitely many rational solutions for the unknowns $y_1, y_2, \ldots, y_n$, so the attacker cannot compute the plaintext in that way. Even when the ciphertext consists of integers and the attacker supposes that the plaintext consists only of integers too, well defined public key can have infinitely many integers solutions. Hence, by this attack, an attacker can only guess the plaintext.

The attacker can only guess the private key as well, since the key space is infinite.

The public key was produced by using linear bi-permutations and we find that it is the weakest part of the construction of SBIM(Q). We could not realized an attack by using this weakness, although it may be possible.

5.2 Implementation

The public key has $2n$ polynomials of $3n$ variables. If we bounded the degrees of the polynomial by 2, the polynomials may have up to $\frac{(3n)!(3n-1)}{2} + 6n + 2$ members. So, the choice of the starting polynomials should be suitably made for obtaining desired public key. We think that for $n = 4$ quite secure public key can be designed, with quadratic polynomials only.

An obvious question is why we are not using linear polynomials only? The reason is that in that case the system is not secure.

The presented example in Section 5 shows that the choice of the bi-permutations have to be carefully done. Choosing simpler matrices A_i, B_i may produce weak public key. On the other hand, applying more $e-$ and $e'-$transformations will produce better public key.

Caution: After any construction of SBIM(Q), it has to be checked if the system of equations (8) has infinitely many solutions for the variables $y_1, \ldots, y_n$.

5.3 Performance

The performance of the system depends on the chosen polynomials mainly. For example, let take $n = 4$ and let we work with polynomials of degree 2 only. Those polynomials have at most 91 members (1 constant + 12 linear + 78 quadratic). Then for getting a ciphertext we have to make no more than $8 \cdot 12 = 96$ additions and $8 \cdot 156 = 1248$ multiplications of rational numbers. Let assume that 4 transformations were applied for getting the public key. So, 8 4×4 matrices and 4 1×4 vectors as leaders were used. For computing the plaintext we have firstly to apply the inverse matrix R^{-1}, then 16 multiplications and 12

additions of rational numbers will be made. For transformations we have to make 8 multiplications of vectors by matrices, so 128 multiplications and 96 additions of rational numbers have to be made, and 8 additions of vectors by vectors, so 32 additions more have to be made. Hence, altogether, 140 additions and 144 multiplication of rational numbers have to be made for recovering the plaintext, i.e., the message.

5.4 Optimization

Some optimization of the system can be made.

The number of variables can be reduced. Namely, instead of using $3n$ variables for building the polynomials $Y_{n+1}, Y_{n+2}, \ldots, Y_{3n}$, we can reduce the number of polynomials to $2n+m$, where $0 < m < n$. In that case we have to check that the system of equations (8) has infinitely many solutions for the variables $y_1, \ldots, y_n$.

We mentioned that the system is not secure if only linear polynomials are used. Still, we can use linear polynomials to make the system more simpler. For example, many of the polynomials $Y_1, Y_2, \ldots, Y_n$ can be linear, and some of the polynomials $Y_{n+1}, Y_{n+2}, \ldots, Y_{3n}$ too.

One obvious question is why we have used quasigroup transformations? A quite simpler way to produce the public key of the same type is by taking a nonsingular $2n \times 2n$ matrix T and to compute $(A_1, \ldots, A_{2n}) = (Y_1, \ldots, Y_{2n}) \cdot T$. In this case the system is not secure, since then the system (8) can be easily solved for the variables $y_1, \ldots, y_n$ (although the redundant variables $z_1, \ldots, z_n$ may have infinitely many solutions).

6 Conclusion

SBIM(Q) is a public key cryptsystem of multivariate polynomial type. We found that SBIM(Q) has a moderate speed, comparable to today standard public key cryptsystems. Also, we found that a version of SBIM(Q) for $n = 2$ can be implemented in hardware useful for embedded systems. The encryption of SBIM(Q) can be easily parallelized, since each of the $2n$ polynomials of the public key can be independently computed.

References

1. Courtois, N., Klimov, A., Patarin, J., Shamir, A.: Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations. In: Preenel, B. (Ed.) Advances in Cryptology-Eurocrypt 2000. LNCS, vol. 1807, pp. 392–407. Springer-Verlag Berlin Heidelberg (2000).
2. Diffie, W., Hellman, M.: New Directions in Cryptography. IEEE Trans. Inform. Theory IT-22 (6), 644–654 (1976).
3. Gligoroski, D., Markovski, S., Knapskog, S.J.: Multivariate quadratic trapdoor functions based on multivariate quadratic quasigroups. In: American Conference on Applied Mathematics. Harvard, March 2008, USA.

4. Imai, H., Matsumoto, T.: Algebraic methods for constructing asymmetric cryptosysytems. In: Calmet, J. (Ed.) 3rd Intern. Conf. AAECC-3. LNCS, vol. 229, pp. 108–119. Springer Berlin Heidelberg (1986).
5. Markovski, S., Gligoroski, D., Bakeva, V.: Quasigroup String Processing: Part 1. Contributions, Sec. Math. Tech. Sci., MANU, XX 1- 2, 13–28 (1999).
6. Patarin, J.: Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of asymmetric algorithms. In: Maurer, U.M. (Ed.) Advances in Cryptology - EUROCRYPT 1996. LNCS, vol. 1070, pp. 33–48. Springer (1996).
7. Rivest, R., Shamir, A., Adleman, L.: A Method for Obtaining Digital Signatures and Public Key Cryptosystems. Comm. ACM 21(2), 120–126 (1978).
8. Siljanoska, M., Mihova, M., Markovski, S.: Matrix presentation of quasigroup of order 4. In: 10th Conference for Informatics and Information Technology (CIIT 2013), Bitola (2013).
9. Wolf, C., Preneel, B.: Taxonomy of Public Key Schemes based on the problem of Multivariate Quadratic equations. Cryptology ePrint Archive, Report 2005/077, (2005).