PUBLIC KEY CRYPTOGRAPHY USING PERMUTATION P-POLYNOMIALS

Rajesh P Singh · B.K.Sarma · A.Saikia

Received: date / Accepted: date

Abstract In this paper we propose an efficient multivariate public key cryptosystem based on permutation p-polynomials over finite fields. We first characterize a class of permutation p-polynomials over finite fields F_{q^m} and then construct a trapdoor function using this class of permutation p-polynomials. The complexity of encryption in our public key cryptosystem is $O(m^3)$ multiplication which is equivalent to other multivariate public key cryptosystems. However the decryption is much faster than other multivariate public key cryptosystems. In decryption we need $O(m^2)$ left cyclic shifts and $O(m^2)$ xor operations.

Keywords Multivariate Cryptography · Permutation Polynomials · Linear Polynomials

1 Introduction

Public key cryptography is used in e-commerce for authentication and secure communication. The most widely used cryptosystems RSA and ECC (elliptic

Rajesh P Singh
Dpartment of Mathematics, Indian Institute of Technology, Guwahati, Assam (India), Pin 781039
Tel.: +91-258-2646
Fax: +91-258-2649
E-mail: r.pratap@iitg.ernet.in

B.K.Sarma
Dpartment of Mathematics, Indian Institute of Technology, Guwahati, Assam (India), Pin 781039
E-mail: bks@iitg.ernet.in

A.Saikia
Dpartment of Mathematics, Indian Institute of Technology, Guwahati, Assam (India), Pin 781039
E-mail: a.saikia@iitg.ernet.in

curve cryptosystems) are based on the problem of integer factorization and discrete logarithm respectively. Improvements in factorization algorithm and computation power demands larger bit size in RSA key. At present the recommended key size is of 1024 bits which may have to be increased to 4096 bits by 2015 [1]. Larger key size makes RSA less efficient for practical applications. ECC are more efficient as compared to RSA, but its shortest signature is of 320 bits which is still long for many applications [2]. Although RSA and ECC have these drawbacks, they are still not broken. But in 1999 Peter Shor [4] discovered a polynomial time algorithm for integer factorization and computation of discrete logarithm on quantum computers. Thus once we have quantum computers the cryptosystems based on these problems can no longer be considered secure. So there is a strong motivation to develop public key cryptosystems based on problems which are secure on both conventional and quantum computers. Multivariate cryptography can be a possible option applicable to both conventional and quantum computers (see [9]). In multivariate cryptography the public key cryptosystems are based on the problem of solving system of nonlinear equations which is proven to be NP-complete. MIC*, the first practical public key cryptosystem based on this problem was proposed in 1988 by T. Matsumoto and H. Imai (see [12]). The MIC* cryptosystem was based on the idea of hiding a monomial x^{2^l+1} by two invertible affine transformations. This cryptosystem was more efficient than RSA and ECC. Unfortunately this cryptosystem was broken by Patarin in 1995[13]. In 1996 [14] Patarin gave a generalization of MIC* cryptosystem called HFE. However in HFE the secret key computation was not as efficient as in the original MIC* cryptosystem. The basic instance of HFE was broken in 1999[16]. The attack uses a simple fact that every homogeneous quadratic multivariate polynomial has a matrix representation. Using this representation a highly over defined system of equations can be obtained which can be solved by a new technique called relinearization [16]. Other possible attacks on the HFE scheme can be found in [17], [18] and [19]. Patarin [15] investigated whether it is possible to repair MIC* with the same kind of easy secret key computations. He designed some cryptosystems known as Dragons with multivariate polynomials of total degree 3 or 4 in public key (instead of 2) with enhanced security and with efficiency comparable to MIC*. In Dragon cryptosystems the public key was of mixed type of total degree 3 which is quadratic in plaintext variables and linear in ciphertext variables. However Patarin found [15] that Dragon scheme with one hidden monomial is insecure.

A public key scheme based on the composition of tame transformation methods (TTM) was proposed in 1999[23]. This scheme has been broken in 2000[24], where the cryptanalysis is reduced to an instance of the Min-Rank problem that can be solved within a reasonable time. In 2004 Ding [20] proposed a perturbed variant of MIC* cryptosystem called PMI. The PMI system attempts to increase the complexity of the secret key computations in order to increase security, using a system of r arbitrary quadratic equations over F_q with the assumption that $r << n$, where n is the bitsize. The PMI Cryptosystem was broken by Fouque, Granboulan and Stern [21]. The trick of the attack on PMI is to use differential cryptanalysis to reduce the PMI system to the MIC* system. A cryptosystem called Medium Field Equation (MFE) was proposed in 2006[25] and was broken by Ding in 2007[26] using high order linearization equation attack. For a detailed introduction of multivariate public key cryptography, we refer the interested readers to [9]. An interesting introduction of hidden monomial cryptosystems can be found in reference[10].

Designing secure and efficient multivariate public key cryptosystem continues to be a challenging area of research in recent years. In this paper we present a new method for designing efficient multivariate public key cryptosystem by overcoming all the known attacks. We are using permutation p-polynomials to construct a non-linear trapdoor function. Like Dragon cryptosystems the public key in our cryptosystem is of mixed type but it is possible to reduce the public key size by writing it as two sets of quadratic multivariate polynomials [15]. In our cryptosystem the decryption is possible by using only $O(m^2)$ left cyclic shifts and $O(m^2)$ xor operations and this results in much faster decryption. The complexity of encryption is equivalent to other multivariate public key cryptosystems that is $O(m^3)$ multiplications, where m is the bit size. The outline of our paper is as follows. In section 2 we give preliminaries and in section 3 we characterize a class of permutation p-polynomials. Then in section 4 we present our cryptosystem. In section 5 we give the security analysis of our cryptosystem and in section 6 we discuss the efficiency of our cryptosystem. We compare our cryptosystem with HFE in section 7.Finally we conclude in section 8.

2 Preliminaries

Let q be a prime power and let $\mathbb{F}_q$ denote the finite field of order q. We will denote an extension of $\mathbb{F}_q$ of degree m by $\mathbb{F}_{q^m}$. An element $\vartheta \in \mathbb{F}_{q^m}$ is said to be normal over $\mathbb{F}_q$ if the elements $\vartheta, \vartheta^q, \ldots, \vartheta^{q^{m-1}}$ form a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. In that case the set $B = \{\vartheta, \vartheta^q, \ldots, \vartheta^{q^{m-1}}\}$ is called a normal basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Any element x of $\mathbb{F}_{q^m}$ can be expressed as $x = \sum_{i=0}^{m-1} x_i \vartheta^{q^i}$ where $x_i \in \mathbb{F}_q$. Thus $\mathbb{F}_{q^m}$ can be identified by $\mathbb{F}_q^m$, the set of all m-tuples over $\mathbb{F}_q$, and $x \in \mathbb{F}_{q^m}$ can be written as $(x_0, x_1, \ldots x_{m-1})$. If we take the normal basis representation of finite field $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$, then the operation $x \mapsto x^q$ is $(x_{m-1}, x_0, \ldots, x_{m-2})$ which is just one left cyclic shift of $(x_0, x_1, \ldots, x_{m-1})$. Hence the cost of exponentiating by q is negligible. From now on we will take normal basis representation of finite field $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ with respect to normal basis B. When $q = 2$, we define the weight of x as the number of 1's in $(x_0, x_1, \ldots, x_{m-1})$, and denote it by $w(x)$.

A polynomial f over $\mathbb{F}_q$ is called a *permutation polynomial* of $\mathbb{F}_q$ if the polynomial f induces a one-one map on $\mathbb{F}_q$ onto itself. Permutation polynomials have been a subject of study for almost one and a half century see [6], [7] and Chapter 7 of [8]. A polynomial $L(x) \in \mathbb{F}_{q^m}[x]$ is called a *p- polynomial* or *linearized polynomial* over $\mathbb{F}_q$ if

$$L(x) = \sum_{i=0}^{k} \alpha_i x^{q^i}. \tag{1}$$

The p-polynomial $L(x)$ satisfies the following: $L(\beta+\gamma) = L(\beta) + L(\gamma)$ and $L(a\beta) = aL(\beta)$ for all $\beta, \gamma \in \mathbb{F}_{q^m}$ and $a \in \mathbb{F}_q$. Thus, $L : x \mapsto L(x)$ is a linear operator of the vector space $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Consequently, $L(x)$ is a permutation polynomial of $\mathbb{F}_{q^m}$ if and only if 0 is the only root of $L(x)$ in $\mathbb{F}_{q^m}$.

Corresponding to an element $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{m-1})$ of the finite field $\mathbb{F}_{q^m}$, we define a p-polynomial $L_\alpha(x)$ on $\mathbb{F}_{q^m}$ as

$$L_\alpha(x) = \sum_{i=0}^{m-1} \alpha_i x^{q^i}. \tag{2}$$

It is known that each function on $\mathbb{F}_{q^m}$ is given by a unique polynomial of degree at most $q^m - 1$ (see chapter 7 of [8]). Since the polynomial $L_\alpha(x)$ is of degree at most $q^m - 1$, the distinct polynomials $L_\alpha(x)$ are all distinct as functions on $\mathbb{F}_{q^m}$.

Definition 1 Suppose $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{m-1})$ and $\beta = (\beta_0, \beta_1, \ldots, \beta_{m-1})$, $\alpha_i, \beta_i \in \mathbb{F}_q$, are two elements of finite fields $\mathbb{F}_{q^m}$. We define the *convolution* $\alpha * \beta$ of α and β by

$$\alpha * \beta = (\gamma_0, \gamma_1, \ldots, \gamma_{m-1})$$

where

$$\gamma_k = \sum_{i=0}^{m-1} \alpha_{i \bmod m} \beta_{(k-i) \bmod m}.$$

Suppose $L_\alpha \circ L_\beta$ denotes the composition of linearized polynomials L_α and L_β. Then it can be easily verified that $L_\alpha \circ L_\beta = L_{\alpha * \beta}$. Therefore we can conclude that the linearized polynomials $L_\alpha(x)$ form a semigroup with identity. Let $\mathcal{L}(m)$ denote the group of all invertible linearized polynomials $L_\alpha(x)$ over $\mathbb{F}_{q^m}$. In section 3, we will identify $\mathcal{L}(m)$ with an appropriate subgroup of the general linear group $GL(m, \mathbb{F}_q)$. Moreover we will characterize elements of $\mathcal{L}(m)$ for certain values of m and thereby show that the groups $\mathcal{L}(m)$ are quite large.

3 Characterization of the group $\mathcal{L}(m)$, for $m = 2^k$

A characterization of a linearized polynomial to be a permutation was given by Dickson [3], which is as follows:

Theorem 1 [3] *The linearized polynomial*

$$L(x) = \sum_{s=0}^{m-1} c_s x^{q^s} \in \mathbb{F}_{q^m}[x]$$

is a permutation polynomial of $\mathbb{F}_{q^m}$ *if and only if*

$$\begin{vmatrix} c_0 & c_{m-1}^q & c_{m-2}^{q^2} & \cdots & c_1^{q^{m-1}} \\ c_1 & c_0^q & c_{m-1}^{q^2} & \cdots & c_2^{q^{m-1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{m-1} & c_{m-2}^q & c_{m-3}^{q^2} & \cdots & c_0^{q^{m-1}} \end{vmatrix} \neq 0. \tag{3}$$

An $m \times m$ matrix A over a field $\mathbb{F}$ is said to be *circulant* if it has the form

$$A = \begin{pmatrix} a_0 & a_{m-1} & a_{m-2} & \cdots & a_1 \\ a_1 & a_0 & a_{m-1} & \cdots & a_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m-1} & a_{m-2} & a_{m-3} & \cdots & a_0 \end{pmatrix}. \tag{4}$$

Let e_k denote the k^{th} column of the identity matrix I and R be the matrix $(e_2, e_3, \ldots, e_m, e_1)$ obtained by a permutation of columns of I. Clearly m is the least positive integer such that $R^m = I$. Let a denote the vector $(a_0, a_1, \ldots, a_{m-1})^T$ and A the circulant matrix as in equation (4). Then we have

$$A = (a, Ra, R^2a, \ldots, R^{m-1}a) \tag{5}$$

$$= a_0I + a_1R + a_2R^2 + \ldots + a_{m-1}R^{m-1}. \tag{6}$$

We will denote the circulant matrix A by $\mathrm{cir}(a_0, a_1, \ldots, a_{m-1})$. The product of any two circulant matrices A and B, where $A = \mathrm{cir}(a_0, a_1, a_2, \ldots, a_{m-1})$ and $B = \mathrm{cir}(b_0, b_1, b_2, \ldots, b_{m-1})$ is again circulant:

$$AB = \mathrm{cir}(d_0, d_1, d_2, \ldots, d_{m-1}),$$

$$\text{where } d_k = \sum_{i=0}^{m-1} a_{i (\mathrm{mod}\ m)} b_{k-i (\mathrm{mod}\ m)}. \tag{7}$$

It is known that the inverse of a nonsingular circulant matrix is circulant (see [5]). Thus the nonsingular circulant matrices over a field $\mathbb{F}$ form a subgroup of the general linear group of $\mathbb{F}$. In view of equation (5), we note that this group is abelian. We denote this group by $\mathcal{C}(\mathbb{F}, m)$.

Lemma 1 *For* $m \geq 1$*, the groups* $\mathcal{C}(\mathbb{F}_q, m)$ *and* $\mathcal{L}(m)$ *are isomorphic.*

Proof. We define a mapping $\phi : \mathcal{L}(m) \to \mathcal{C}(\mathbb{F}_q, m)$ as follows:

$$\phi(L_\alpha) = \mathrm{cir}(\alpha_0, \alpha_1, \ldots, \alpha_{m-1})$$

where $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{m-1})$. Since $\alpha_i \in \mathbb{F}_2$ in Theorem 1, the determinant in (3) is that of the circulant matrix $\mathrm{cir}(\alpha_0, \alpha_1, \ldots, \alpha_{m-1})$. Thus L_α is invertible if and only if $\mathrm{cir}(\alpha_0, \alpha_1, \ldots, \alpha_{m-1})$ is nonsingular. In other words, ϕ is a bijection. It follows from equation (7) that ϕ is a group homomorphism. □

Proposition 1 *If $m = 2^k$ for some $k \geq 0$, then the polynomial $L_\alpha(x)$ is a permutation of $\mathbb{F}_{q^m}$ if and only if $w(\alpha)$ is odd.*

Proof. Let $\alpha = \sum_{i=0}^{m-1} \alpha_i q^i$. If $w(\alpha)$ is even, then $L_\alpha(x)$ has 0 and 1 as roots, and therefore is not a permutation. Next suppose that $w(\alpha)$ is odd. Then

$$\begin{aligned}
\operatorname{cir}(\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{m-1})^{2^k} &= (\alpha_0 I + \alpha_1 R + \alpha_2 R^2, \ldots + \alpha_{m-1} R^{m-1})^{2^k} \\
&= \alpha_0 I + \alpha_1 R^{2^k} + \alpha_2 (R^{2^k})^2 + \ldots + \alpha_{m-1} (R^{2^k})^{m-1} \\
&= \alpha_0 I + \alpha_1 I + \alpha_2 I + \ldots + \alpha_{2^k - 1} I = I.
\end{aligned}$$

This implies that $\operatorname{cir}(\alpha_0, \alpha_1, \ldots, \alpha_{m-1})$ is invertible and therefore in view of Lemma 1, $L_\alpha(x)$ is a permutation polynomial. □

In the proof of above proposition we are taking $q = 2$, the proof is same for any q of the form $p^s, s \geq 1$ and $m = p^k$. In that case the condition '$w(\alpha)$ is odd' will be replaced by $\sum_{i=0}^{m-1} \alpha_i \neq 0$ in $\mathbb{F}_q$.

Corollary 1 *Let $\mathbb{F}_{q^m}$ is a finite field, with $m = 2^k$, $k \geq 0$. Let $L_\alpha^j(x)$ denote the jth times composition of $L_\alpha(x)$ with itself. If weight of α is odd, then the inverse polynomial of $L_\alpha(x)$ is $L_\alpha^{2^k - 1}(x)$.*

Proof. The result follows by noting that
$(cir(\alpha_0, \alpha_1, \ldots, \alpha_{m-1}))^{2^k} = I$. □

Lemma 1 implies, in particular, that the group $\mathcal{L}(m)$ is abelian. Since there are 2^{m-1} different α with odd weight, $\mathcal{L}(m)$ has order 2^{m-1} when $m = 2^k$. For $q = 2$, the converse of proposition 1 is true and can be seen in the following proposition.

Proposition 2 *Let the integer m be such that $L_\alpha(x)$, $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{m-1})$, is a permutation polynomial over $\mathbb{F}_{q^m}$ whenever $w(\alpha)$ is odd. Then $m = 2^k$ for some $k \geq 0$.*

Proof. Since $L_\alpha(x)$ is not a permutation polynomial when $w(\alpha)$ is even, we have 2^{m-1} as the order of $\mathcal{L}(m)$ and therefore that of $\mathcal{C}(\mathbb{F}_2, m)$.
Now $R = (e_2, e_3, \ldots, e_m, e_1)$ is an element of $\mathcal{C}(\mathbb{F}_2, m)$ of order m. Thus m divides 2^{m-1} and has the required form. □

Lemma 2 *$\mathbb{F}_{q^m}$ is finite field and α, β are two elements of $\mathbb{F}_{q^m}$. Let $\alpha = (\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{m-1})$ and $\beta = (\beta_0, \beta_1, \beta_2, \ldots, \beta_{m-1})$ then we have $L_\alpha(\beta) = L_\beta(\alpha)$.*

Proof. The proof of this lemma is consequence of the fact that
$\operatorname{cir}(\alpha_0, \alpha_1, \ldots, \alpha_{m-1})(\beta_0, \beta_1, \ldots, \beta_{m-1})^{\mathrm{T}} = \operatorname{cir}(\beta_0, \beta_1, \ldots, \beta_{m-1})(\alpha_0, \alpha_1, \ldots, \alpha_{m-1})^{\mathrm{T}}$ and $L_\alpha(\beta) = \operatorname{cir}(\alpha_0, \alpha_1, \ldots, \alpha_{m-1})(\beta_0, \beta_1, \ldots, \beta_{m-1})^{\mathrm{T}}$. □

Proposition 3 *Suppose $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{m-1})$ is an element of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Then α is normal element of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ if and only if $L_\alpha(x)$ is a permutation polynomial of $\mathbb{F}_{q^m}$.*

Proof. Suppose α is normal element of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Then for all $y \in \mathbb{F}_{q^m}$ there exist $x \in \mathbb{F}_{q^m}$ such that $y = L_x(\alpha)$ or $y = L_\alpha(x)$. This implies $L_\alpha(x)$ is a permutation of $\mathbb{F}_{q^m}$. Conversely suppose that $L_\alpha(x)$ is permutation of $\mathbb{F}_{q^m}$. Then $L_\alpha(x) = L_x(\alpha) = y$ has a unique solution for all $y \in \mathbb{F}_{q^m}$. This implies that α is a normal element $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. □

Thus we see that there is a one-one correspondence between normal elements of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ and the linear permutation polynomials of the form $L_\alpha(x)$. Using the above proposition we can easily count the normal elements of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$.

Corollary 2 *Let $\mathbb{F}_{q^m}$ is a finite field, with $m = p^k$, $k \geq 0$. Then total number of normal elements of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ is $q^m - q^{m-1}$, that is for $m = 2^k$, $q = 2$ the number of normal elements are 2^{m-1}.* □

Suppose α is normal element of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ and $f \in \mathcal{L}(m)$, then it can be easily verified that $f(\alpha)$ is also normal element. Thus in view of proposition 3, we can state the following corollary:

Corollary 3 *Suppose for $\alpha \in \mathbb{F}_{q^m}$, $L_\alpha(x)$ is permutation polynomial of $\mathbb{F}_{q^m}$ and $f \in \mathcal{L}(m)$ is any arbitrary element, then $L_{f(\alpha)}(x)$ is also a permutation polynomial of $\mathbb{F}_{q^m}$.* □

4 Public key Cryptosystem

In this section we present our multivariate public key cryptosystem using results from the previous section. Our cryptosystem will work in any arbitrary finite field $\mathbb{F}_{q^m}$, $m = p^k$. But for practical view point we need only $q = 2$, so we will assume that $q = 2$ and $m = 2^k$. To obtain the quadratic polynomials we use the convolution of bits. We have seen that convolution of binary bits is equivalent to the composition of corresponding p-polynomials. We know that composition of two permutation p-polynomial is a permutation p-polynomial so the convolution of two odd weight binary strings is an odd weight binary string. For $x \in \mathbb{F}_{q^m}$, $(x)^t$ denotes the t times convolution of x with itself. The set of all odd weight element of $\mathbb{F}_q^m$ is denoted by $O\mathbb{F}_q^m$. To describe our cryptosystem systematically we need the next two lemmas.

Lemma 3 *Suppose $x = (x_0, x_1, \ldots, x_{m-1})$ is an element of $\mathbb{F}_q^m$. If $(x)^t = (h_0, h_1, \ldots, h_{m-1})$, then h_i are non-linear functions of x_i of degree $w(t)$, where $w(t)$ denotes the Hamming weight of t.*

Proof: Since $x_i^2 = x_i$ and by definition of convolution, the bits of $(x)^2$ are linear function of x_i and it can be verified easily that if $(x)^2 = (c_0, c_1, \ldots, c_{m-1})$, then $c_{2i+1} = 0$ and $c_{2i} = x_i + x_{m/2+i}$. By definition of convolution the bits of $(x)^3$ will be quadratic multivariate polynomials. This implies that if $G = (g_0, g_1, \ldots, g_{m-1})$ and g_i are non linear polynomials of degree d and suppose $(G)^{2^l} = (g_0^{'}, g_1^{'}, \ldots, g_{m-1}^{'})$ and $(g_0^{''}, g_1^{''}, \ldots, g_{m-1}^{''})$ denotes the value of $(G)^{2^l+1}$, where $l \geq 1$ then the degrees of $g_i^{'}$ and $g_i^{''}$ are d and $d + 1$ respectively. This proves the lemma. □

Lemma 4 *The function defined by $h(x) = (x)^t$, where t is co-prime to m is a bijection from $O\mathbb{F}_q^m$ to $O\mathbb{F}_q^m$ itself.*

Proof. Since t and m are co-prime, there exist integers r and k such that $t.k = 1 + r.m$. Suppose $y = h(x) = (x)^t$, this implies that $L_y = L_{(x)^t} = L_x^t$ or $L_y^k = L_x^{rm+1}$. But by proposition 1 we know $(L_x)^m = L_\vartheta$, where L_ϑ is the identity mapping. Thus we have $L_x = L_y^k = L_{(y)^k}$ or $x = (y)^k$. This proves the lemma. □

4.1 Public Key Generation

Consider a message of $m-1$ bit string $(x_0, x_1, \ldots, x_{m-2})$, where m is of the form 2^k. We adjoin an additional bit x_{m-1} to make the weight odd. After decryption one can just ignore the last bit x_{m-1}. So we can assume that message $X = (x_0, x_1, \ldots, x_{m-1})$ is an m bit odd weight element of the finite field $\mathbb{F}_{2^m}$. Suppose $L_\alpha, L_\beta, L_\gamma$ and L_δ, L_η are elements of the group $\mathcal{L}(m)$ and L_ξ, L_ζ are elements of the group $\mathcal{L}(2m)$. Let π_1,π_2, π_3, π_4 and π_5 be random permutations of $\{0, 1, 2, \ldots, m-1\}$, and π_6, π_7 be random permutations of $\{0, 1, 2, \ldots, 2m-1\}$. Now compute $T_1^{'} = L_\alpha \circ \pi_1$, $T_2^{'} = L_\beta \circ \pi_2$, $T_3^{'} = L_\gamma \circ \pi_3$, $T_4^{'} = L_\delta \circ \pi_4$, $T_5^{'} = L_\eta \circ \pi_5$, $T_6^{'} = L_\xi \circ \pi_6$ and $T_7^{'} = L_\zeta \circ \pi_7$, where $L_{\alpha'} \circ \pi_i$ denotes the composition of $\mathrm{L}_{\alpha'}$ and the permutation π_i, for $1 \leq i \leq 7$ and $\alpha' \in \{\alpha, \beta, \gamma, \delta, \eta, \xi, \zeta\}$. Now define the affine transformation $T_r(X) = T_r^{'}(X) + \sigma_r$ for $1 \leq r \leq 7$ where σ_r for $1 \leq r \leq 5$ is an even weight element of $\mathbb{F}_{2^m}$ and σ_6, σ_7 are even weight element of $\mathbb{F}_{2^{2m}}$. Note that if X is an odd weight element of finite field $\mathbb{F}_{2^m}$, then $T_r^{'}(X)$ and $T_r(X)$ are also odd weight element of $\mathbb{F}_{2^m}$. Thus $T_r(X)$ is a bijection of $O\mathbb{F}_{2^m}$. Now compute $X^{'} = T_1(X)$, $X^{''} = T_2(X)$. Again compute $T_3\left((X^{'})^2 * X^{''}\right)$ and $T_4\left(X^{'} * X^{''}\right) + T_5\left((X^{'})^2 * X^{''}\right)$. Suppose the quadratic polynomials f_i and f_{m+i} denote the i^{th} bits of $T_3\left((X^{'})^2 * X^{''}\right)$ and $T_4\left(X^{'} * X^{''}\right) + T_5\left((X^{'})^2 * X^{''}\right)$ respectively in the normal basis representation. Suppose $\vartheta^{'}$ is the normal element of finite fields $\mathbb{F}_{2^{2m}}$ and $B^{'}$ denotes the normal basis of $\mathbb{F}_{2^{2m}}$ over $\mathbb{F}_2$ corresponding to the normal element $\vartheta^{'}$. Now consider the $2m$ bits $(f_0, f_1, \ldots, f_{2m-1})$ as an element of $\mathbb{F}_{2^{2m}}$ corresponding to the basis $B^{'}$. Ciphertext $Y = (y_0, y_1, \ldots, y_{2m-1})$ is an element of odd weight in $\mathbb{F}_{2^{2m}}$. Suppose $Z = T_6(Y)$. Suppose λ and σ are elements of $\mathbb{F}_{2^{2m}}$ of even and odd weight respectively. Then by lemma 4 the function $\lambda + \sigma * (Z)^{2m-1}$ is a bijection of $O\mathbb{F}_{2^{2m}}$. The relation between plaintext and ciphertext is:

$$\begin{aligned} &T_7(f_0, f_1, \ldots, f_{2m-1}) = \lambda + \sigma * (Z)^{2m-1} \\ &i.e., F(X) = \lambda + \sigma * [T_6(Y)]^{2m-1} \end{aligned} \tag{8}$$

From equation 8 and using corollary 1 and using the fact that convolution operation is distributive over addition in finite fields, that is, $a * (b + c) = a * b + a * c$ for $a, b, c \in \mathbb{F}_{2^{2m}}$, Alice has the following relation between the plaintext and ciphertext:

$$T_7\left(f_0, f_1, \ldots, f_{2m-1}\right) * Z + \lambda * Z + \sigma = 0 \tag{9}$$

Equation 9 gives the $2m$ polynomial equations of total degree 3 in variables $\{x_0, x_1, \ldots, x_{m-1}; y_0, y_1, \ldots, y_{2m-1}\}$ but only of degree 1 in variables y_i. Thus we get $2m$ equations of the form

$$\begin{aligned} &\sum a_{ijk}x_ix_jy_k + \sum b_{ij}x_iy_j + \sum c_{ij}x_ix_j + \sum d_ky_k \\ +& \sum e_kx_k + f_l \end{aligned} \tag{10}$$

Terms $a_{ijkl}x_ix_jy_k$ and $b_{ijl}x_iy_j$ and d_ky_k always will be there, others terms may not be there. The equation 10 is of degree three, so in one equation of the form 10, there will be $O(m^3)$ terms and we have $2m$ equations, so total size will be of $O(m^4)$, which is large. But it is possible to reduce the size of polynomial equations shown in 10 up to $O(m^3)$ by writing it as a two sets of public polynomials containing only quadratic terms (without changing the security since this can be done in polynomial time) see [15]. Thus the public key will be two sets of $2m$ quadratic equations of the form:

$$\sum g_ky_k + \sum b_{ij}x_iy_j + \sum d_ky_k + \sum e_kx_k + f_l$$

where

$$g_k = \sum h_{ijk}x_ix_j$$

The results in section 3 are true for any arbitrary prime power number q so the public key size can be further reduced by taking m which is not too large (for example m=32) and q which is not too small.

4.2 Secret Key

The linear transformations $(T_1, T_2, T_3, T_4, T_5, T_6, T_7)$ and finite fields elements (λ, σ) are the required secret keys.

4.3 Encryption

If Bob wants to send a message $M = (x_0, x_1, \ldots, x_{m-1})$ to Alice, he substitutes the plaintext vector in the public key and solves the resulting linear equations for the ciphertext $Y = (y_0, y_1, \ldots, y_{2m-1})$. Bob will get a unique ciphertext because our encryption function is injective. Given a ciphertext Y, the public equations are nonlinear in x_i. It follows from equation 8 that our encryption function is:

$$E(X) = Y = T_6^{-1}\Big[\left(\left(F(X) + \lambda\right) * (\sigma)^{2m-1}\right)^{2m-1}\Big],$$

where $F(X) = T_7(f_0, f_1, \ldots, f_{2m-1})$.
(note that $((t)^{2m-1})^{2m-1} = t$ in $\mathbb{F}_{2^m}$.)

Theorem 1 *The encryption function E is well defined and bijective from $O\mathbb{F}_{2^m}$ to $E(O\mathbb{F}_{2^m})$, where $E(O\mathbb{F}_{2^m})$ denotes the set of all imaged elements of $O\mathbb{F}_{2^{2m}}$.*

Proof. Suppose $X_1, X_2 \in O\mathbb{F}_{2^m}$. It is easy to verify that $X_1 = X_2$ implies $E(X_1) = E(X_2)$. Now we assume that $E(X_1) = E(X_2)$ that is
$T_6^{-1}\Big[\left((F(X_1)+\lambda)*(\sigma)^{2m-1}\right)^{2m-1}\Big] = T_6^{-1}\Big[\left((F(X_2)+\lambda)*(\sigma)^{2m-1}\right)^{2m-1}\Big]$
i.e., $\left((F(X_1)+\lambda)*(\sigma)^{2m-1}\right)^{2m-1} = \left((F(X_2)+\lambda)*(\sigma)^{2m-1}\right)^{2m-1}$.
Note that $(F(X_1)+\lambda)*(\sigma)^{2m-1}$ and $(F(X_2)+\lambda)*(\sigma)^{2m-1}$ are elements of $O\mathbb{F}_{2^{2m}}$, so by lemma 4 we have

$$(F(X_1)+\lambda)*(\sigma)^{2m-1} = (F(X_2)+\lambda)*(\sigma)^{2m-1}$$

Now taking the convolution both sides by σ and noting that $(\sigma)^{2m} = \vartheta'$, the identity of convolution. We have

$$F(X_1) = F(X_2)$$

i.e, $T_3\left((X_1')^2 * X_1''\right) = T_3\left((X_1')^2 * X_1''\right)$ and $T_4\left(X_1' * X_1''\right) + T_5\left((X_1')^2 * X_1''\right)$ $= T_4\left(X_2' * X_2''\right) + T_5\left((X_2')^2 * X_2''\right)$. From these two relations we have

$$(X_1')^2 * X_1'' = (X_2')^2 * X_2''$$

and

$$X_1' * X_1'' = X_2' * X_2''$$

that is

$$X_1' * (X_2' * X_2'') = X_2' * (X_2' * X_2'')$$

which implies

$$X_1' = X_2' \Longrightarrow X_1 = X_2.\square$$

4.4 Decryption

To recover the original message M from the ciphertext $Y = (y_0, y_1, \ldots, y_{2m-1})$ Alice uses her private key $(T_1, T_2, T_3, T_4, T_5, T_6, T_7, \lambda, \sigma)$ and the relation 8. First she computes $Z = T_6\,(Y)$. To compute $(Z)^{2m-1}$ efficiently she takes the linearized polynomial corresponding to Z and takes an element of group $\mathcal{L}(m)$ and its inverse, say L_α and L_α^{-1}, and then repeatedly computes $L_Z(\alpha)$, $L_Z^2(\alpha)$, $\ldots, L_Z^{2m-1}(\alpha)$. Note that $L_Z^{2m-1}(\alpha) = L_{(Z)^{2m-1}}(\alpha)$. But by lemma 2, we have, $L_{(Z)^{2m-1}}(\alpha) = L_\alpha((Z)^{2m-1})$. Thus $(Z)^{2m-1} = L_\alpha^{-1}\left(L_Z^{2m-1}(\alpha)\right)$. Thus computation of convolution of finite field elements can be done using only left cyclic shifts and xor operations and therefore it is very efficient. From now onwards we will be using this efficient technique to compute convolution. Now Alice computes $Z' = \lambda + \sigma * (Z)^{2m-1}$ and then $\Delta = T_7^{-1}(Z')$. Suppose $\Delta = (\delta_0, \delta_1, \ldots, \delta_{2m-1})$, $\Delta_1 = (\delta_0, \delta_1, \ldots, \delta_{m-1})$, $\Delta_2 = (\delta_m, \delta_{m+1}, \ldots, \delta_{2m-1})$. Now

Alice has $T_3\left((X^{'})^2 * X^{''}\right) = \varDelta_1$ and $\varDelta_2 = T_4\left(X^{'} * X^{''}\right) + T_5\left((X^{'})^2 * X^{''}\right)$. Now she computes $X^{'} * X^{''} = T_4^{-1}\left(\varDelta_2 + T_5\left(T_3^{-1}\left(\varDelta_1\right)\right)\right)$ and $(X^{'})^2 * X^{''} = T_3^{-1}\left(\varDelta_1\right)$ Now Alice does the following computations, she computes $L_{X' * X''}(\alpha)$ and $L_{(X')^2 * X''}(\alpha)$. Suppose $\theta = L_{X' * X''}(\alpha)$. Now Alice takes the linearized polynomial corresponding to θ that is L_θ and computes the inverse of L_θ. By corollary 1 we know that $L_\theta^{-1} = L_\theta^{m-1}$ and we can compute L_θ^{m-1} very efficiently by the procedure described above using only left cyclic shifts and xor operations on bits. Now we have $L_{(X')^2 * X''}(\alpha) = L_{X'}(\theta)$ and by lemma 2 we have $L_{X'}(\theta) = L_\theta(X^{'})$. Now Alice computes $X^{'} = L_\theta^{-1}(L_\theta(X^{'}))$ and $X = T_1^{-1}(X^{'})$ is the required secret message.

5 The Security of the proposed Cryptosystem

In this section we discuss the security of the proposed cryptosystem. In general it is very difficult to prove the security of a public key cryptosystem [31], [32]. For example if the public modulus of RSA is decomposed into its prime factors then the RSA is broken. However it is not proved that breaking RSA is equivalent to factoring its modulus, see [33]. In this section we will give some security arguments and evidence that our cryptosystem is secure. Most of the multivariate public key cryptosystems use the structure $t(f(s(x)))$, where t and s are secret invertible linear transformation and $f(x)$ is a quadratic non linear function. Hiding $f(x)$ by two linear transformations is not working very effectively (see the attack of Kipnis and Shamir on HFE [16]). We are using a different structure and we will prove that our structure is more secure than the $t(f(s(x)))$ structure. In our cryptosystem the function $f(x)$ is $(x * x * x, x * x + x * x * x)$ so $t(f(s(x))) = t\left(s(x) * s(x) * s(x),\ s(x) * s(x) + s(x) * s(x) * s(x)\right)$. We are taking simpler case, suppose we are not using the transformations T_3, T_4 and T_5 then in our structure, will be $T_7\left(F_1(X), F_2(X)\right)$, where $F_1(X) = T_1(x) * T_1(x) * T_2(x)$and $F_2(X) = T_1(x) * T_2(x) + T_1(x) * T_1(x) * T_2(x)$. It is clear that if $T_1 = T_2$ then our structure will be equivalent to $t(f(s(x)))$. Thus if it is possible to attack our structure then it is also possible to attack $t(f(s(x)))$ structure. This proves that our structure is more secure than the commonly used structure, that is $t(f(s(x)))$, in multivariate cryptography. Moreover our quadratic part of plaintext is hidden because in our cryptosystem the public polynomials are the m bit representation of $F(X) * Z + \lambda * Z + \sigma$ where $F(X) = T_7(f_0, f_1, \ldots, f_{2m-1})$ and $Z = T_6(Y)$. From $F(X) * Z + \lambda * Z + \sigma$ it is not possible to compute either $F(X)$, Z, λ and σ because $F(X) * Z$ is equivalent to the composition of corresponding p-polynomials and in general it is very difficult to decompose the composition of two functions. We are using affine transformations, so the bitwise representation of $F(X) * Z$ will give the terms of the form $d_k y_k + c_k$ also. So it is not possible to find λ and σ from the public key. Here we discuss some known attacks developed for multivariate cryptosystems and we will show that those attacks are not applicable to our cryptosystem. The attacks discussed in this section are Grobner basis, univariate polynomial representation, Linearization, Relinearization, XL and FXL algorithms.

5.1 Linearization Equation Attacks

Let $F = \{f_0, f_1, \ldots, f_{m-1}\}$ be any set of m polynomials in $\mathbb{F}_q[x_0, x_1, \ldots, x_{m-1}]$. A linearization equation for F is any polynomial in $\mathbb{F}_q[x_0, x_1, \ldots, x_{m-1} : y_0, y_1, \ldots, y_{m-1}]$ of the form

$$\sum_{i=0}^{m-1}\sum_{j=0}^{m-1} a_{ijl}x_iy_j + \sum_{i=0}^{m-1} b_{il}x_i + \sum_{j=0}^{m-1} c_{jl}y_j + d_l \tag{11}$$

where $l = 0, 1, \ldots, m-1$.
This attack was first successfully applied by Patarin in [13] to break the Matsumoto-Imai cryptosystem C^* [12]. The idea of Patarin was to notice that if a function is defined as $F : x \to x^{q^i+1}$, then a relation between plaintext $(x_0, x_1, \ldots, x_{m-1})$ and ciphertext $(y_0, y_1, \ldots,$
$y_{m-1})$ of the form shown in equations (11) can be established, where a_{ij}, b_i, c_j and d_l are unknown coefficients. By taking at least $(m+1)^2$ different plaintext and ciphertext pairs a linear system of equations can be established and solved. We are not taking any function of the form x^{q^i+1}. Moreover in our cryptosystem the plaintext and ciphertext are connected by the relation (8) and T_6, λ and σ are secrets. So in our case it is not possible to obtain a relation of the form (11). However, somebody can try to find a relation which is linear in x_i and nonlinear in y_j. We will prove that this line of attack is not possible as the degree of the inverse function is very high. From the relation (8) we have $(f_0, f_1, \ldots, f_{2m-1}) = T_7^{-1}(Z^{'})$. Note that $Z^{'} = \lambda + \sigma * (Z)^{2m-1}$ and $Z = T_6(Y)$ so $T_7^{-1}(Z^{'})$ will give $w(2m-1)$ degree non-linear polynomials in ciphertext variables. Suppose $T_7^{-1}(Z^{'}) = (Z_0, Z_1, \ldots, Z_{2m-1})$. Then we have the following relations between plaintext and ciphertext $T_3\left(X^{'} * X^{'} * X^{''}\right) = (Z_0, Z_1, \ldots, Z_{m-1})$ and $T_4\left(X^{'} * X^{''}\right) + T_5\left(X^{'} * X^{'} * X^{''}\right) = (Z_m, Z_{m+1}, \ldots, Z_{2m-1})$. Using these two relations one can get the following relation between the plaintext and ciphertext:

$$X^{'} * T_4^{-1} \circ T_5\left(Z^{'}\right) + T_4^{-1}\left(Z_m, Z_{m+1}, \ldots, Z_{2m-1}\right) = Z^{''} \tag{12}$$

here $X^{'} = T_1(X)$ and $Z^{''} = T_3^{-1}(Z_0, Z-1, \ldots, Z_{m-1})$ and $T_1, T_2, T_3, T_4, T_5, T_6$ are unknown linear transformations. Note that the relation (12) is of total degree $w(2m-1)+1$, $w(2m-1)$ degree in ciphertext and one degree in plaintext. Most crucially the degree of relation (12) is not constant but function of m. Thus to attack the cryptosystem we need Gaussisn reduction on $O\left(m^{w(2m-1)+1}\right)$ terms which is impractical for bit size greater than or equal to 64 because for $m = 64$, $w(2m-1)+1 = 8$.

5.2 Attacks with Differential Cryptanalysis

Differential cryptanalysis has been successfully used earlier to attack the symmetric cryptosystem. In recent years differential cryptanalysis has emerged as a powerful tool to attack the multivariate public key cryptosystems too. In 2005 [21] Fouque, Granboulan and Stern used differential cryptanalysis to attack the multivariate cryptosystems. The key point of this attack is that in case of quadratic polynomials the differential of public key is a linear map and its kernel or its rank can be analyzed to get some information on the secret key. For any multivariate quadratic function $G : \mathbb{F}_q^n \to \mathbb{F}_q^m$ the differential operator between any two points $x, k \in \mathbb{F}_q^n$ can be expressed as $L_{G,k}G(x+k) - G(x) - G(k) + G(0)$ and in fact that operator is a bilinear function. By knowing the public key of a given multivariate quadratic scheme and by knowing the information about the nonlinear part (x^{q^i+1}) they showed that for certain parameters it is possible to recover the kernel of $L_{G,k}$. This attack was successfully applied on Ding's cryptosystem [20] and afterwards using the same technique Dubois, Fouque, Shamir and Sterm in 2007 [27] have completely broken all versions of the SFLASH signature scheme proposed by Patarin, Courtois, and Goubin [22]. In our cryptosystem we are not using any polynomial of the form x^{q^i+1}. Moreover the public key in our system is not quadratic but of total degree 3, quadratic in plaintext variables and degree one in ciphertext variables. Substituting the ciphertext gives quadratic plaintext variables but in that case it will be different for different ciphertexts. So to attack our cryptosystem by the methods of [21] and [27] is not feasible.

5.3 Univariate polynomial representation of Multivariate Public Polynomials

As the encryption function is from finite field $\mathbb{F}_{2^m}$ to finite field $\mathbb{F}_{2^{2m}}$ so we can not directly represent the encryption function by a polynomial. But it is possible by introducing dummy variables $x_m, x_{m+1}, \ldots, x_{2m}$. In our cryptosystem the relation between the plaintext and ciphertext is $F(X) = \lambda + \sigma * (Z)^{2m-1}$, $F(X) = T_7(f_0, f_1, \ldots, f_{2m-1})$. We have $Y = T_6^{-1}(G(X))$ where, $G(X) = \left((F(X)+\lambda) * \sigma^{2m-1}\right)^{2m-1}$. Note that $F(X)$ is non linear of degree 2, so that $T_6^{-1}(G(X))$ will give $2m$ multivariate polynomials of degree $2.w(2m-1)$. By lemma 3.3 of [16] the degree of univariate polynomial representation is not constant but it is function of m. Thus the degree and the number of nonzero terms of the univariate polynomial representation of encryption function are both $O(m^m)$. The complexity of root finding algorithms e.g. Berlekamp algorithm, is polynomial in the degree of the polynomial. This results in an exponential time algorithm to find the roots of univariate polynomial. Therefore this approach is less efficient than the exhaustive search.

5.4 Grobner Basis Attacks

After substituting the ciphertext in public key one can get $2m$ quadratic equations in m variables and then Grobner basis techniques can be applied to solve the system. The classical algorithms for solving the system of multivariate equations is Buchberger's algorithm for constructing Grobner basis see [11]. Theoretically it can solve all the multivariate quadratic equations. However its complexity is exponential in the number of variables, although there is no closed-form formula for it. In the worst case the Buchberger's algorithm is known to run in double exponential time and on average its running time seems to be single exponential (see [28]). There are some efficient variants F_4 and F_5 of Buchberger's algorithm given by Jean-Charles Faugere (see [29] and [30]). The complexity of computing a Grobner basis for the public polynomials of the basic HFE scheme is not feasible using Buchberger's algorithm. However it is completely feasible using the algorithm F_5. The complexities of solving the public polynomials of several instances of the HFE using the algorithm F_5 are provided in [19]. Moreover it has been expressed in [19] "a crucial point in the cryptanalysis of HFE is the ability to distinguish a randomly algebraic system from an algebraic system coming from HFE". Instead of using any polynomial of special form we are using convolution operation to construct the public polynomials. Moreover our public key is of mixed type, this mean for different ciphertexts we will get different system of quadratic polynomial equations, so in our public key the quadratic polynomials looks random. We have already seen that the degree of univariate polynomial representation of encryption function is proportional to m. It is explained in [19] that in this case there does not seem to exist polynomial time algorithm to compute the Grobner basis. Hence to attack our cryptosystem by Grobner basis method is not feasible.

5.5 Relinearization, XL and FXL Algorithms

It is now clear that attack of [16] is not applicable to our cryptosystem. However the adversary may directly apply the Relinearization, XL or FXL algorithm. The main problem in applying the techniques to solve the quadratic equations directly is that our public key is of mixed type, this means for different ciphertexts we have to solve the different system of quadratic non linear equations. In the following we show that to attack the cryptosystem by this approach is not possible.

The Relinearization technique is developed in [16] for solving over defined system of quadratic equations. Unfortunately (or fortunately) it is shown in [28] that the Relinearization technique is not as efficient as one may expect since many of newly generated equations are dependent. Hence the XL (extended relinearization) technique has been proposed in [28]. It is claimed to be the best algorithm for solving over defined multivariate equations. However when the number of equations is $m+r$ for some $1 \leq r \leq m$ then XL has exponential complexity [28]. In our cryptosystem $r = m$. Hence the XL algorithm can not be directly used to attack our cryptosystem. A variant of the XL algorithm called FXL, was introduced in [28]. In this algorithm some variables are guessed to make the system slightly over defined. Then the XL algorithm is applied. The main question is how many variables must be guessed. Although more guesses make the system more unbalanced they add to the complexity of the algorithm. The optimum number of guesses is provided in [28]. Using this optimum value the FXL has the exponential complexity for solving the system of public polynomials in proposed cryptosystem. Hence The FXL algorithm is not applicable to our cryptosystem.

6 Complexity and number of operations for encryption and decryption

In this section we give complexity of the encryption and decryption of our cryptosystem.

6.1 Encryption

The public key in our cryptosystem consists of 2m equations of the form (10). There are $O(m^2)$ terms of the form $x_i x_j$ in each $2m$ equations of the public key so the complexity of evaluating public key at message block $x_0, x_1, \ldots, x_{m-1}$ is $O(m^3)$. The next step of encryption is to solve the $2m$ linear equation in $2m$ ciphertext variables $y_0, y_1, \ldots, y_{2m-1}$. This can be done efficiently by Gaussian elimination in $O(m^3)$ complexity. Hence the total complexity of encryption is $O(m^3)$.

6.2 Decryption

In our cryptosystem decryption is very fast. In decryption we are using the operations: permutation of bits, xor and left cyclic shifts of bits. In this section we will count the total number of operations to describe the exact efficiency of our cryptosystem. To operate T_i or T_i^{-1}, $0 \leq i \leq 5$ on a m bit string we need one permutation on bits and at most $m-2$ left cyclic shifts and m xor operations. To operate T_i or T_i^{-1}, for $i = 6, 7$ on a $2m$ bit string we need one permutation on $2m$ bits and at most $2m-2$ left cyclic shifts and $2m$ xor operations. To compute $(Z)^{2m-1}$, where Z is $2m$ bit string, we need at most $(2m-1)(2m-2) + 2m - 2$ left cyclic shifts and at most $(2m-1)^2 + 2m - 1$ xor operations. Thus to compute L_θ^{-1} where θ is m bit binary string we need at most $(m-1)(m-2) + m - 2$ left cyclic shifts and at most $(m-1)^2 + m - 1$ xor operations. Thus we see that in decryption we need $O(m^2)$ xor operations and $O(m^2)$ left cyclic shifts operations.

7 Comparison with HFE

In our cryptosystem the complexity of encryption is $O(m^3)$, i.e., equivalent to that of HFE. But the decryption is faster than HFE. In HFE the decryption is slow because one needs to compute the roots of a polynomial. The decryption complexity of HFE is $O\left(n^4 d^2 log(d)\right)$ where d is the degree of HFE polynomial. Note that for security reasons one can not take smaller degree. Due to this the decryption process in HFE is slow. In our cryptosystem we are using left cyclic shifts and xor operations resulting much faster decryption process. In our cryptosystem we need $O(m^2)$ left cyclic shifts and $O(m^2)$ xor operations to decrypt a message. Public key size of HFE is of $O(m^3)$ terms. In our cryptosystem public key size is bigger than HFE but it is also of $O(m^3)$ as it is possible to write public key as two sets of quadratic public polynomials. Secret key generation in our public key cryptosystem is faster than HFE because for secret keys we have to select random odd weight and even weight binary strings and random permutations.

8 Conclusion and Future Work

In this paper we show how permutation p-polynomials can be used to design an efficient public key cryptosystem. We characterize permutation p-polynomials over finite field $\mathbb{F}_{q^m}$ for $m = p^k$and use these to construct a trapdoor function. Computations with these polynomials are fast which makes them useful. As far as we know these permutation p-polynomials were never used before to design a public key cryptosystem. In our cryptosystem the public key is of mixed type of total degree three, two in plaintext variable and one in cipher text variable. However it is possible to reduce the public key size by writing it as a two sets of quadratic polynomials. Further investigations to develop an efficient cryptosystem using these permutation p-polynomials with public of two degree in plaintext variable can be an interesting topic of future work. The bit size in our public key cryptosystem is of the form 2^k. It is desirable to extend this cryptosystem for an arbitrary bit size with same level of efficiency. This can be done by characterizing permutation p-polynomials over finite field $\mathbb{F}_{q^m}$ for arbitrary m and by giving efficient method to find their inverse.

9 Acknowledgement

We are grateful to Professor Rana Barua of Indian Statistical Institute, Kolkata, India, for his valuable suggestions.

References

1. B. Schneier. Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd ed. New York: Wiley, 1996.

2. The Elliptic Curve Digital Signature Algorithm (ECDSA), ANSI X9.62 American National Standards Institute, 1998.
3. L.E. Dickson. The analytic representation of substitution on a power of a prime number of letters with a discussion of the linear group. *Ann. of Math.* 11, 65-120.
4. Peter Shor. Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Journal on Scientific Computing.* 26 (1997), 1484.
5. P. J. Davis, Circulant Matrices *American Mathematical Society, 1994*
6. R. Lidl and G. L. Mullen. When does a polynomial over a finite field permute the elements of the field? *The American Math. Monthly*, 95(3), 243-246, 1988.
7. R. Lidl and G. L. Mullen. When does a polynomial over a finite field permute the elements of the field? II *The American Math. Monthly*, 100(1), 71-74, 1993.
8. R. Lidl and H. Niederreiter. *Finite Fields.* Addision-Wesley, 1983.
9. Jintai Ding, Jason E. Gower, Dieter S. Schmidt. *Multivariate Public Key Cryptosystems.* Springer, 2006
10. Neal Koblitz. *Algebraic Aspects of Cryptography.* Springer, 1998
11. D. Cox, J. Little, and D. OShea,. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra,.* 2nd ed. New York: Springer-Verlag, 1997, Undergraduate Texts in Mathematics.
12. T. Matsumoto and H.Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. *Eurocrypt '88,* Springer-Verlag (1988), pp. 419453.
13. J.Patarin. Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt '88, . *Advances in Cryptology-Crypto '95,* Springer-Verlag, 248261.
14. J.Patarin. Hidden Field equations (HFE) and isomorphism of polynomials (IP): two new families of asymetric algorithms, *Advances in cryptology- Eurocrypt '96,* Springer-Verlag, pp. 3348.
15. J.Patarin. Asymmetric cryptography with a hidden monomial. *Advances in Cryptology-Crypto '96,* Springer-Verlag, pp. 4560.
16. Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization. *CRYPTO '99,* LNCS Vol. 1666, pp. 19-30, 1999.
17. N.T Courtois. The sequrity of Hidden Field equations (HFE), *in CT-RSA'01, '2001,* LNCS, vol. 2020 pp. 266281.
18. Nicolas T. Courtois, Magnus Daum, and Patrick Felke. On the Security of HFE, HFEv- and Quartz, *in PKC '2003,* LNCS, vol. 2567 pp. 337350.
19. Jean-Charles Faugere and Antoine Joux. Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Grobner Basis. *in CRYPTO '2003,* LNCS Vol. 2729, pp. 44-60, 2003.
20. J. Ding. A New Variant of the Matsumoto-Imai Cryptosystem through Perturbation. *in PKC04,* LNCS Vol. 2947, pp. 305-318, 2004.
21. Pierre-Alain Fouque, Louis Granboulan, and Jacques Sterm. Differential Cryptanalysis for Multivariate Schemes. *in EUROCRYPT 2005,* LNCS Vol. 3494, pp. 341-353, 2005.
22. J. Patarin, N. T. Courtois, and L. Goubin. FLASH, a fast multivariate signature algorithm. *in CT-RSA01, '2001,* LNCS Vol. 2020, pp. 298-307.
23. T.T.Moh. A public key system with signature and master key functions. *Commun. Algebra,* vol. 27, no. 5, pp. 2207-2222, 1999.
24. L. Goubin and N. T. Courtois. Cryptanalysis of the TTM cryptosystem. *in Adv. Cryptol.ASIACRYPT00, 2000* LNCS, vol. 1976, pp. 44-57.
25. Lih-ChungWang, Bo-yin Yang, Yuh-Hua Hu and Feipei Lai. A Medium- Field Multivariate Public key Encryption Scheme, *CT-RSA 2006: The Cryptographers Track at the RSA Conference 2006*, LNCS 3860, 132- 149, Springer, 2006
26. Jintai Ding, Lei Hu, Xuyun Nie, Jianyu Li, and John Wagner. High Order Linearization Equation (HOLE) Attack on Multivariate Public Key Cryptosystems. *PKC 2007* LNCS, vol. 4450, pp. 233-248, Springer, 2007.
27. Vivien Dubois, Pierre-Alain Fouque, Adi Shamir, Jacques Stern. Practical Cryptanalysis of Sflash. *in Advances in Cryptology-Crypto '2007,* LNCS Vol. 4622, pp. 1-12.
28. Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient Algorithm for Solving Overdefined System of Multivariate Polynomial Equations. *EUROCRYPT '2000,* LNCS Vol. 1807, pp. 392-407.
29. Faugere, Jean-Charles,. A New efficient algorithm for computing Grobner bases (F_4). *Journal of Pure and Applied Algebra '2002,* 139: 61-88

30. Faugere, Jean-Charles,. A New efficient algorithm for computing Grobner bases without reduction to zero (F_5). *International Symposium on Symbolic and Algebraic Computation - ISSAC '2002,* pages 75-83. ACM Press.
31. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. Handbook of Applied Cryptography. New York: CRC Press, 1997.
32. D. R. Stinson. Cryptography: Theory and Practice. Boca Raton, FL: CRC Press, 1995, The CRC Series on Discrete Mathematics and Its Applications.
33. S. Goldwasser and M. Bellare. Lecture Notes on Cryptography 2001 [Online]. Available: http://www.cs.ucsd.edu/users/mihir/papers/ gb.html.

10 Appendix

Example 1. $x+x^2+x^4$, $x+x^2+x^8$, $x^2+x^4+x^8$ are permutation polynomials of $\mathbb{F}_{2^4}$ and their inverses are $x+x^4+x^8$, $x+x^2+x^8$ and $x^2+x^4+x^8$ respectively. $x+x^2+x^4$, $x+x^2+x^8$ and $x+x^2+x^4+x^8+x^{16}$ are permutation polynomials of $\mathbb{F}_{2^8}$ and their inverses are $x^2+x^4+x^{16}+x^{32}+x^{128}$, $x+x^4+x^{32}+x^{64}+x^{128}$ and $x^2+x^8+x^{64}$ respectively.

Here is the toy example of our public key cryptosystem.

Example 2. We are taking the finite field $\mathbb{F}_{2^4}$ and $\lambda = 0$ and $\sigma = (1,0,0,0,0,0,0,0)$. Suppose $\vartheta^{'}$ is the normal element of $\mathbb{F}_{2^8}$ and we are taking the normal basis representation of $\mathbb{F}_{2^8}$ with respect to $\vartheta^{'}$. Suppose $T_1 = \pi_1 = \begin{pmatrix} 0\,1\,2\,3 \\ 2\,0\,3\,1 \end{pmatrix}$ and $T_2 = \pi_2 = \begin{pmatrix} 0\,1\,2\,3 \\ 3\,2\,1\,0 \end{pmatrix}$ and T_3, T_4, T_5 are $x+x^2+x^4,\ x^2+x^4+x^8,\ x+x^2+x^8$ respectively, $T_6 = \pi_6 = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6\,7 \\ 3\,1\,5\,0\,4\,2\,6\,7 \end{pmatrix}$ and $T_7 = \pi_7 \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6\,7 \\ 5\,3\,7\,2\,1\,0\,4\,6 \end{pmatrix}$.
Message $M = (x_0, x_1, x_2, x_3)$,
$T_1(M) = M^{'} = (x_2, x_0, x_3, x_1)$ and $T_2(M) = M^{'} = (x_3, x_2, x_0, x_1)$. We compute all the bits of $M^{'} * M^{''}$. Suppose $[M^{'} * M^{''}]_i$ denotes the ith bit of $M^{'} * M^{''}$. We obtain, $[M^{'} * M^{''}]_0 = x_2x_3 + x_0 + x_3x_1 + x_1x_2$, $[M^{'} * M^{''}]_1 = x_2 + x_1$, $[M^{'} * M^{''}]_2 = x_3 + x_1x_0 + x_2x_1 + x_0x_2$, $[M^{'} * M^{''}]_3 = x_2x_0 + x_0x_1 + x_2x_3 + x_1x_3$. Now compute all the bits $M^{'} * M^{'} * M^{''}$. The bits of $M^{'} * M^{'} * M^{''}$ are $[M^{'} * M^{'} * M^{''}]_0 = x_2x_3 + x_0x_1 + x_3 + x_1$, $[M^{'} * M^{'} * M^{''}]_1 = x_0 + x_2 + x_0x_1 + x_2x_3$, $[M^{'} * M^{'} * M^{''}]_2 = x_0x_3 + x_1x_2$, $[M^{'} * M^{'} * M^{''}]_3 = x_0x_3 + x_1x_2$. (f_0, f_1, f_2, f_3) denotes the bits of $T_3(M^{'} * M^{'} * M^{''})$ and (f_4, f_5, f_6, f_7) denotes the bits of $T_4(M^{'} * M^{''}) + T_5(M^{'} * M^{'} * M^{''})$, we have $f_0 = x_3x_2 + x_0x_1 + x_3 + x_1$, $f_1 = 1 + x_0x_3 + x_1x_2$, $f_2 = 1 + x_0x_3 + x_0x_1$, $f_3 = x_0 + x_2 + x_3x_2 + x_1x_2$, $f_4 = x_0 + x_0x_3 + x_3x_2 + x_1x_3$, $f_5 = 1 + x_0 + x_2x_3 + x_3x_0$, $f_6 = x_3 + x_2x_0 + x_1x_2 + x_0x_1$, $f_7 = 1 + x_0x_1 + x_0x_2 + x_3x_1 + x_2x_3$. Ciphertext $Y = (y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7)$ is an element of $\mathbb{F}_{2^{2m}}$. $T_6(Y) = Z = (y_3, y_1, y_5, y_0, y_4, y_2, y_6, y_7)$. Note that $\vartheta^{'} = (1,0,0,0,0,0,0,0)$. Suppose $P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7$ denote the bits of $T_7(f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7) * Z + \vartheta^{'}$. We compute all the P_i, so the required public key is :
$P_0 = 1 + y_0(x_0x_1 + x_1x_2) + y_1(x_0x_2) + y_2(x_0x_3 + x_2x_3) + y_3(x_2x_3 + x_0x_3) + y_4(x_0x_3 + x_1x_3) + y_5(x_2x_3 + x_0x_3 + x_1x_3) + y_6(x_1x_2 + x_2x_3 + x_1x_3) + y_7(x_0x_1 + x_1x_3)x_0y_3 + x_1y_3 + x_0y_7 + x_2y_7 + x_2y_6 + x_0y_6 + y_2 + y_4 + x_1y_0 + x_3y_0 + x_0y_5 + x_1y_1$
$P_1 = y_0(x_2x_3 + x_0x_3 + x_1x_3) + y_1(x_0x_3 + x_2x_3) + y_2(x_0x_3 + x_1x_3) + y_3(x_0x_1 +$

$x_1x_2)+y_4(x_0x_1+x_1x_2)+y_5(x_0x_2)+y_6(x_0x_3+x_2x_3)+y_7(x_2x_3+x_1x_2+x_0x_2+x_1x_3)+x_0y_0+x_0y_1+x_0y_3+x_0y_7+x_1y_1+x_1y_4+x_1y_5+x_2y_3+x_2y_7+x_3y_4+y_2+y_6$

$P_2 = y_0(x_0x_2)+y_1(x_0x_1+x_1x_2)+y_2(x_0x_1+x_1x_2)+y_3(x_2x_3+x_1x_3+x_0x_2+x_1x_2)+y_4(x_0x_3+x_2x_3+x_1x_3)+y_5(x_2x_3+x_0x_3)+y_6(x_0x_3+x_1x_3)+y_7(x_0x_3+x_2x_3)+x_0y_1+x_0y_3+x_0y_4+x_0y_5+x_1y_0+x_1y_2+x_1y_5+x_2y_1+x_2y_3+x_3y_2+y_6+y_7$

$P_3 = y_0(x_0x_3+x_2x_3)+y_1(x_0x_2+x_1x_2+x_2x_3)+y_2(x_0x_3+x_2x_3+x_1x_3)+y_3(x_2x_3+x_0x_3)+y_4(x_0x_2)+y_5(x_0x_1+x_1x_2)+y_6(x_0x_1+x_1x_2)+y_7(x_0x_3+x_1x_3)+x_0y_0+x_0y_1+x_0y_2+x_0y_5+x_1y_0+x_1y_4+x_1y_6+x_2y_1+x_2y_5+x_3y_6+y_3+y_7$

$P_4 = y_0(x_0x_1+x_1x_2)+y_1(x_2x_3+x_0x_3)+y_2(x_0x_2)+y_3(x_1x_3+x_0x_3)+y_4(x_0x_3+x_2x_3)+y_5(x_2x_3+x_0x_2+x_1x_3+x_1x_2)+y_6(x_1x_3+x_2x_3+x_0x_3)+y_7(x_0x_3+x_2x_3)+x_0y_0+x_0y_4+x_0y_5+x_0y_6+x_1y_4+x_1y_2+x_1y_7+x_2y_0+x_2y_5+x_3y_7+y_1+y_3$

$P_5 = y_0(x_0x_2+x_1x_2+x_1x_3+x_2x_3)+y_1(x_0x_3+x_1x_3)+y_2(x_0x_3+x_2x_3)+y_3(x_0x_1+x_1x_2)+y_4(x_0x_1+x_1x_2)+y_5(x_2x_3+x_0x_3)+y_6(x_0x_2)+y_7(x_2x_3+x_0x_3+x_1x_3)+x_0y_0+x_0y_2+x_0y_4+x_0y_7+x_1y_2+x_1y_3+x_1y_6+x_2y_0+x_2y_4+x_3y_3+y_1+y_5$

$P_6 = y_0(x_0x_3+x_2x_3)+y_1(x_0x_1+x_1x_2)+y_2(x_0x_1+x_1x_2)+y_3(x_2x_3+x_0x_3+x_1x_3)+y_4(x_0x_2+x_1x_3+x_2x_3+x_1x_2)+y_5(x_0x_3+x_1x_3)+y_6(x_0x_3+x_2x_3)+y_7(x_0x_2)+x_0y_2+x_0y_3+x_0y_4+x_0y_6+x_1y_1+x_1y_6+x_1y_7+x_2y_2+x_2y_4+x_3y_1+y_0+y_5$

$P_7 = y_0(x_0x_3+x_1x_3)+y_1(x_0x_3+x_1x_3+x_2x_3)+y_2(x_0x_2+x_2x_3+x_1x_2+x_1x_3)+y_3(x_0x_2)+y_4(x_2x_3+x_0x_3+x_0x_2)+y_5(x_1x_2+x_0x_1)+y_6(x_0x_1+x_1x_2)+y_7(x_2x_3+x_0x_3)+x_0y_1+x_0y_2+x_0y_6+x_0y_7+x_1y_3+x_1y_5+x_1y_7+x_2y_2+x_2y_6+x_3y_5+y_0+y_4$

The public key looks large, however it is possible to reduce the size of public key containing only quadratic terms. The public key can be written as two sets of public polynomials containing only quadratic terms. We have

$P_0^{'} = 1+y_0g_1+y_1g_0+y_2g_4+y_3g_4+y_4g_5+y_5g_3+y_6b+y_7g_5+x_0y_3+x_1y_3+x_0y_7+x_2y_7+x_2y_6+x_0y_6+y_2+y_4+x_1y_0+x_3y_0+x_0y_5+x_1y_1$

$P_1^{'} = y_0g_3+y_1g_4+y_2g_5+y_3g_1+y_4g_1+y_5g_0+y_6g_4+y_7g_2+x_0y_0+x_0y_1+x_0y_3+x_0y_7+x_1y_1+x_1y_4+x_1y_5+x_2y_3+x_2y_7+x_3y_4+y_2+y_6$

$P_2^{'} = y_0g_0+y_1g_1+y_2g_1+y_3g_2+y_4g_3+y_5g_4+y_6g_5+y_7g_4+x_0y_1+x_0y_3+x_0y_4+x_0y_5+x_1y_0+x_1y_2+x_1y_5+x_2y_1+x_2y_3+x_3y_2+y_6+y_7$

$P_3^{'} = y_0g_4+y_1g_6+y_2g_3+y_3g_4+y_4g_0+y_5g_1+y_6g_1+y_7g_5+x_0y_0+x_0y_1+x_0y_2+x_0y_5+x_1y_0+x_1y_4+x_1y_6+x_2y_1+x_2y_5+x_3y_6+y_3+y_7$

$P_4^{'} = y_0g_1+y_1g_4+y_2g_0+y_3g_5+y_4g_4+y_5g_2+y_6g_3+y_7g_4+x_0y_0+x_0y_4+x_0y_5+x_0y_6+x_1y_4+x_1y_2+x_1y_7+x_2y_0+x_2y_5+x_3y_7+y_1+y_3$

$P_5^{'} = y_0g_2+y_1g_5+y_2g_4+y_3g_1+y_4g_1+y_5g_4+y_6g_0+y_7g_3+x_0y_0+x_0y_2+x_0y_4+x_0y_7+x_1y_2+x_1y_3+x_1y_6+x_2y_0+x_2y_4+x_3y_3+y_1+y_5$

$P_6^{'} = y_0g_4+y_1g_1+y_2g_1+y_3g_3+y_4g_2+y_5g_5+y_6g_4+y_7a_0+x_0y_2+x_0y_3+x_0y_4+x_0y_6+x_1y_1+x_1y_6+x_1y_7+x_2y_2+x_2y_4+x_3y_1+y_0+y_5$

$P_7^{'} = y_0g_5+y_1g_3+y_2g_2+y_3g_0+y_4g_7+y_5g_1+y_6g_1+y_7g_4+x_0y_1+x_0y_2+x_0y_6+x_0y_7+x_1y_3+x_1y_5+x_1y_7+x_2y_2+x_2y_6+x_3y_5+y_0+y_4$

Where $g_0 = x_0x_2$, $g_1 = x_0x_1+x_1x_2$, $g_2 = x_2x_3+x_1x_2+x_0x_2+x_1x_3$, $g_3 = x_2x_3+x_0x_3+x_1x_3$, $g_4 = x_2x_3+x_0x_3$, $g_5 = x_0x_3+x_1x_3$, $g_6 = x_2x_3+x_1x_2+x_0x_2$, $g_7 = x_2x_3+x_0x_3+x_0x_2$ and $b = g_0+g_2$. Suppose $M = (0,0,0,1)$ is the plaintext message. Substituting this in above public equation we get linear equations, $y_2+y_4+y_0 = 1$, $y_2+y_4+y_6 = 0$, $y_2+y_6+y_7 = 0$, $y_3+y_6+y_7 = 0$, $y_1+y_3+y_7 = 0$, $y_1+y_3+y_5 = 0$,

$y_0 + y_1 + y_5 = 0$, $y_0 + y_4 + y_5 = 0$. Solving these linear equations by Gaussian-elimination we get $(y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7) = (0, 1, 0, 0, 1, 1, 1, 1)$, which is the required ciphertext.