Poly-Dragon: An efficient Multivariate Public Key Cryptosystem

Rajesh P Singh[*], A.Saikia[†], B.K.Sarma[‡]
Department of Mathematics
Indian Institute of Technology Guwahati
Guwahati -781039, India

December 1, 2009

Abstract

In this paper we propose an efficient multivariate public key cryptosystem. Public key of our cryptosystem contains polynomials of total degree three in plaintext and ciphertext variables, two in plaintext variables and one in ciphertext variables. However, it is possible to reduce the public key size by writing it as two sets of quadratic multivariate polynomials. The complexity of encryption in our public key cryptosystem is $O(n^3)$, where n is bit size, which is equivalent to other multivariate public key cryptosystems. For decryption we need only four exponentiations in the binary field. Our Public key algorithm is bijective and can be used for encryption as well as for signatures.

Keywords *Public Key Cryptography, Multivariate Cryptography, Little Dragon Cryptosystem, Big-Dragon Cryptosystem.*

1 Introduction

Public key cryptography has several practical applications, for example in e-commerce systems for authentication (electronic signatures) and for secure communication. The most widely used cryptosystems RSA and ECC (elliptic curve cryptosystems) are based on the problems of integer factorization and discrete logarithm respectively. Integer factorization and discrete logarithm problems are only believed to be hard but no proof is known for their NP-completeness or NP-hardness. Improvements in factorization algorithms and computation power demands larger bit size in RSA key which makes RSA less efficient for practical applications. Although RSA and ECC have some drawbacks, they are still not broken. But in 1999 [1] Peter Shor discovered the polynomial time algorithm for integer factorization and computation of discrete

[*] r.pratap@iitg.ernet.in

[†] a.saikia@iitg.ernet.in

[‡] bks@iitg.ernet.in

logarithm on quantum computers. Thus once we have quantum computers in the range of 1,000 bits, 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 is based on the problem of solving non-linear system of equations over finite fields which is proven to be NP-complete. Quantum computers do not seem to have any advantage on solving NP-complete problems, so multivariate cryptography can be a viable option applicable to both conventional and quantum computers. MIC*, the first practical public key cryptosystem based on this problem was proposed in 1988 [5] by T. Matsumoto and H. Imai. 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 [6]. In 1996 [7] 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 [9]. 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. Patarin [8] investigated whether it is possible to repair MIC* with the same kind of easy secret key computations. He designed some cryptosystems known as Dragons (Little Dragon and Big Dragon) 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 [8] found that Dragon scheme with one hidden monomial is insecure. Some more multivariate public key cryptosystems can be found in [11] and [12]. For a brief introduction of multivariate cryptography we refer the interested readers to [13]. An interesting introduction of hidden monomial cryptosystems can be found in [3].

Designing secure and efficient multivariate public key cryptosystems continues to be a challenging area of research in recent years. In this paper we present Poly-Dragon, an efficient multivariate public key Cryptosystem. Like Big Dragon cryptosystem the public key in our cryptosystem is of mixed type of total degree three, two in plaintext variables and one in ciphertext variables. However, it is possible to reduce the public key size by writing it as a two sets of quadratic multivariate polynomials (see [8]). The efficiency of our public key cryptosystem is equivalent to that of Big Dragon cryptosystem. The complexity of encryption or signature verification is equivalent to other multivariate public key cryptosystems that is $O(n^3)$ where n is the bit size. In decryption or in signature generation we need four exponentiations in the finite field $\mathbb{F}_{2^n}$ and this results in much faster decryption and signature generation. The outline of our paper is as follows. In Section 3 we present our cryptosystem and in Section 4 we give the security analysis of our cryptosystem. In Section 5 we discuss its efficiency.

2 Preliminaries

Let p be a prime, n be a positive integer, and $\mathbb{F}_q$ be the finite field of $q = p^n$ elements. A polynomial $f(x)$ in $\mathbb{F}_q[x]$ is said to be a *permutation polynomial*, if it is a bijection of $\mathbb{F}_q$ onto $\mathbb{F}_q$. A polynomial $f \in \mathbb{F}_q[x]$ is a permutation polynomial of $\mathbb{F}_q$ if and only if one of the following conditions holds:

1. the function f is onto;

2. the function f is one-to-one;

3. $f(x) = a$ has a solution in $\mathbb{F}_q$ for each $a \in \mathbb{F}_q$;

4. $f(x) = a$ has a unique solution in $\mathbb{F}_q$ for each $a \in \mathbb{F}_q$.

Lemma 2.1 *[2] (1) Every linear polynomial that is a polynomial of the form $ax+b$ with $a \neq 0$ over $\mathbb{F}_q$ is a permutation polynomial of $\mathbb{F}_q$.*
(2)The monomial x^k is a permutation polynomial of $\mathbb{F}_q$ if and only if $\gcd(k, q-1) = 1$.

A polynomial $L(x) \in \mathbb{F}_{q^m}[x]$ is called a *p-polynomial* or a *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, $L(x)$ 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 the only root of $L(x)$ in $\mathbb{F}_{q^m}$ is 0.

Let $B = \{\vartheta_1, \vartheta_2, \ldots, \vartheta_n\}$ be a basis of $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$. Any element $x \in \mathbb{F}_{2^n}$ can be uniquely expressed as $x = \sum_{i=0}^{m-1} x_i \vartheta_i$. Thus $\mathbb{F}_{2^n}$ can be identified by $\mathbb{F}_2^n$ and the element $x \in F_{2^n}$ by the n-tuple $(x_1, x_2, \ldots, x_n)$. The weight of $x = (x_1, x_2, \ldots, x_n)$, denoted by $\omega(x)$, is defined to be the number of 1's in x. Corresponding to an element $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$ of the finite field $\mathbb{F}_{2^n}$, we define a p-polynomial $L_\alpha(x)$ on $\mathbb{F}_{2^n}$ as

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

We use $Tr(x)$ to denote the trace function from the field $\mathbb{F}_{2^m}$ to $\mathbb{F}_2$, i.e.,

$$Tr(x) = x + x^2 + x^{2^2} + \ldots + x^{2^{2^{m-1}}}.$$

3 Some Permutation Polynomials

In this section, we generate some families of permutation polynomials over F_{2^n}, which will be used for the proposed cryptosystem.

Proposition 3.1 *Let n be an odd positive integer, and $\beta = (\beta_1, \beta_2, \ldots, \beta_n)$ be an element of F_{2^n} such that $\omega(\beta)$ is even and that 0 and 1 are the only roots of $L_\beta(x)$ in $\mathbb{F}_{2^n}$. Suppose k_1 and k_2 are non negative integers such that $\gcd(2^{k_1} + 2^{k_2}, 2^n - 1) = 1$. Let ℓ be any positive integer with $(2^{k_1} + 2^{k_2}).\ell \equiv 1$ mod $2^n - 1$ and γ be an element of $\mathbb{F}_{2^n}$ with $Tr(\gamma) = 1$. Then*

$$f(x) = (L_\beta(x) + \gamma)^\ell + Tr(x)$$

is a permutation polynomial of F_{2^n}.

Proof. Suppose x and y are distinct elements in F_{2^n} such that $f(x) = f(y)$. We claim that $Tr(x) \neq Tr(y)$. Otherwise, $f(x) = f(y)$ implies that $(L_\beta(x) + \gamma)^\ell = (L_\beta(x) + \gamma)^\ell$. Thus, we have $(L_\beta(x) + \gamma) = (L_\beta(x) + \gamma)$, that is, $L_\beta(x + y) = 0$. Since $x \neq y$ and $L_\beta(x)$ has only roots 0 and 1, we have $x + y = 1$. Then, $Tr(x + y) = 1$, since n is odd. This proves our claim. Without loss of generality we may assume that $Tr(x) = 0$ and $Tr(y) = 1$. Now, $f(x) = f(y)$ implies that

$$(L_\beta(x) + \gamma)^\ell = (L_\beta(y) + \gamma)^\ell + 1$$

and therefore

$$\begin{aligned} L_\beta(x) + \gamma &= [(L_\beta(y) + \gamma)^\ell + 1]^{2^{k_1}+2^{k_2}} \\ &= (L_\beta(y) + \gamma)^{\ell(2^{k_1}+2^{k_2})} + (L_\beta(y) + \gamma)^{\ell.2^{k_1}} + (L_\beta(y) + \gamma)^{\ell.2^{k_2}} + 1 \\ &= L_\beta(y) + \gamma + (L_\beta(y) + \gamma)^{\ell.2^{k_1}} + (L_\beta(y) + \gamma)^{\ell.2^{k_2}} + 1, \end{aligned}$$

since $(2^{k_1} + 2^{k_2}).\ell \equiv 1$ mod $2^n - 1$. Note that $Tr(L_{\beta(x)}) = 0$. Now applying the trace function to both sides of the above equation, we get

$$Tr(\gamma) = Tr(\gamma) + Tr(1),$$

which implies that $Tr(1) = 0$, a contradiction since n is odd. ■

The polynomial $x^2 + x$ has only roots 0 and 1 in $\mathbb{F}_{2^n}$. It is easy to see that 0 and 1 are the only roots of $x^{2^k} + x$ in $\mathbb{F}_{2^n}$ if and only if k and n are relatively prime. In the next lemma, we construct another linearized polynomial having only roots 0 and 1 in $\mathbb{F}_{2^n}$.

Lemma 3.1 *Let k be an even integer, with $\gcd(k, n) = 1$. Then, 0 and 1 are the only roots of $h(x) = \sum\limits_{i=0}^{k-1} x^{2^i}$ in F_{2^n}.*

Proof. Consider $h_1(x) = h(x)^2 + h(x) = h(x)(h(x) + 1)$. Clearly, each root of $h(x)$ is a root of $h_1(x)$. Moreover, $h_1(x) = x^{2^k} + x$ and has 0 and 1 as its only roots. ■

Lemma 3.2 *The polynomial $f(x) = x^{2^{k2^r}+2^r} + x^{2^{k2^r}} + x^{2^r}$, where r and k are positive integers, is a permutation polynomial of $\mathbb{F}_{2^n}$ if and only if $2^{k2^r} + 2^r$ and $2^n - 1$ are co-prime.*

Proof. First, note that there exist integers r and k such that $2^{2^r k} + 2^r$ and $2^n - 1$ are co-prime. It is known that composition of two polynomials is a permutation polynomial if and only if both the polynomials are permutation polynomials see chapter 7 of [2]). It is easy to check that $f(x+1) = x^{2^{k2^r}+2^r}+1$. By Lemma 3.1, $f(x+1)$ is a permutation polynomial if and only if $2^{2^r k}+2^r$ and 2^n-1 are co-prime. Since $x+1$ is always a permutation of $\mathbb{F}_{2^n}$, therefore $f(x)$ is a permutation polynomial of $\mathbb{F}_{2^n}$ if and only if $2^{k2^r}+2^r$ and 2^n-1 are co-prime. ■

As a consequence of above lemma we prove the following lemma, which will be used in public key cryptosystem.

Proposition 3.2 *The polynomial $g(x) = (x^{2^{k2^r}} + x^{2^r} + \alpha)^l + x$ is permutation polynomial of $\mathbb{F}_{2^n}$ if $Tr(\alpha) = 1$ and $(2^{k2^r} + 2^r).l = 1 \mod 2^n - 1$.*

Proof. Since $Tr(x^{2^{k2^r}} + x^{2^r} + \alpha) = Tr(\alpha) = 1$, $x^{2^{k2^r}} + x^{2^r} + \alpha \neq 0$ for all $x \in \mathbb{F}_{2^n}$. Let β be an element of $\mathbb{F}_{2^n}$. Then $g(x) = \beta$ implies

$$(x^{2^{k2^r}} + x^{2^r} + \alpha)^l = x + \beta.$$

Raising both sides to the power $2^{k2^r} + 2^r$, we get

$$(x^{2^{k2^r}} + x^{2^r} + \alpha) = (x + \beta)^{2^{k2^r}+2^r},$$

$$\text{i.e.,}\quad (x^{2^{k2^r}} + x^{2^r} + \alpha) + (x + \beta)^{2^{k2^r}+2^r} = 0.$$

Suppose $h(x) = (x^{2^{k2^r}} + x^{2^r} + \alpha) + (x+\beta)^{2^{k2^r}+2^r}$. It is enough to show that for any $\beta \in GF(2^n)$ the equation $h(x) = 0$ has a unique solution. Note that $h(x) = 0$ and $h(x+\beta) = 0$ have the same number of solutions. Now $h(x+\beta) = 0$ is equivalent to

$$x^{2^{k2^r}+2^r} + x^{2^{k2^r}} + x^{2^r} + \beta^{2^{k2^r}} + \beta^{2^r} + \alpha = 0.$$

By Lemma 3.2, $x^{2^{k2^r}+2^r} + x^{2^{k2^r}} + x^{2^r}$ is a permutation polynomial of $\mathbb{F}_{2^n}$. Hence the equation $h(x+\beta) = 0$ has a unique solution for any $\beta \in \mathbb{F}_{2^n}$. ■

4 The Cryptosystem Poly-Dragon

4.1 Public key generation.

For the public key cryptosystem we will use permutation polynomials $g(x) = (x^{2^{k2^r}} + x^{2^r} + \alpha)^l + x$ and $f(x) = (L_\beta(x) + \gamma)^\ell + Tr(x)$ obtained in Proposition 3.1 and Proposition 3.2. For quadratic public key size, we cannot take all the permutation polynomials of the form $(x^{2^{k2^r}} + x^{2^r} + \alpha)^l + x$. But the permutation polynomials in which l is of the form $2^t + 1$ or $2^t - 1$ can be used to design the multivariate public key cryptosystem with quadratic public key size. For l is of the form $2^t + 1$ it is not clear whether $g(x)$ is a permutation polynomial or not. But for $r = 0$, $n = 2m - 1$, $k = m$, $k_2 = m$, $k_1 = 0$, $l = 2^m - 1$ and $\ell = 2^m - 1$, $g(x)$ and $f(x)$ are permutation polynomials because in this case $2^{2^r k} + 2^r = 2^m + 1$, $2^{k_1} + 2^{k_2} = 2^m + 1$ and $(2^m - 1).(2^m + 1) = 1 \mod 2^n - 1$. So for public key generation we will take $g(x) = (x^{2^m} + x + \alpha)^{2^m-1} + x$ and $f(x) = (L_\beta(x) + \gamma)^{2^m-1} + Tr(x)$, where α, β, γ are secret. Suppose s and t are two invertible affine transformations. The relation between plaintext and ciphertext is $g(s(x)) = f(t(y))$, where variable x denotes the plaintext and y the ciphertext. Suppose $s(x) = u$ and $t(y) = v$. Thus we have the following relation between plaintext and ciphertext: $(u^{2^m} + u + \alpha)^{2^m - 1} + u = (L_\beta(v) + \gamma)^{2^m - 1} + Tr(v)$. Since $u^{2^m} + u + \alpha$ and $L_\beta(v) + \gamma$ are nonzero in the field $\mathbb{F}_{2^n}$, this relation gives

$$(u^{2^m} + u + \alpha)^{2^m}(L_\beta(v) + \gamma) + u(u^{2^m} + u + \alpha)(L_\beta(v) + \gamma) + (u^{2^m} + u + \alpha)(L_\beta(v) + \gamma)^{2^m}$$
$$+ Tr(v)(u^{2^m} + u + \alpha)(L_\beta(v) + \gamma) = 0$$

Suppose $Tr(v) = \zeta_y \in \{0, 1\}$. Using a fixed basis $B = \{\vartheta_1, \vartheta_2, \ldots, \vartheta_n\}$ of $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$, we identify the field $\mathbb{F}_{2^n}$ by $\mathbb{F}_2^n$, the set of all n-tuples over $\mathbb{F}_2$. Substituting $u = s(x)$ and $v = t(y)$, where $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$, we get n non-linear polynomial equations of the form

$$\sum a_{ijk} x_i x_j y_k + \sum b_{ij} x_i x_j + \sum (c_{ij} + \zeta_y) x_i y_j + \sum (d_k + \zeta_y) y_k + \sum (e_k + \zeta_y) x_k + f_l, \quad (3)$$

where $a_{ij}, b_{ij}, c_k, d_k, e_k \in \mathbb{F}_2$. These equations are of degree three and therefore each of them contains $O(n^3)$ terms. Since we have n equations, total size will be of $O(n^4)$, which is large. It is possible to reduce the size of polynomial equations (3) to $O(n^3)$ by writing it as two sets of public polynomials containing only quadratic terms without changing the security since this can be done in polynomial time (see [8]). Thus the public key will be two sets of n quadratic equations of the form:

$$\sum g_k y_k + \sum (b_{ij} + \zeta_y) x_i y_j + \sum (d_k + \zeta_y) y_k + \sum (e_k + \zeta_y) x_k + f_l,$$

where $g_k = \sum h_{ijk} x_i x_j$.

4.2 Secret Key

The invertible affine transformations (s, t) and the field elements α, β, γ are secret keys.

4.3 Encryption

If Bob wants to send a plaintext message $x = (x_1, x_2, \ldots, x_n)$ to Alice, he does the following:

1. Bob substitutes the plaintext $(x_1, x_2, \ldots, x_n)$ and $\zeta_y = 0$ in public key and gets n linear equations in ciphertext variables $y_1, y_2, \ldots, y_n$. Bob solve these linear equations by Gaussian elimination and gets $y^{'} = (y_1, y_2, \ldots, y_n)$

2. In the second step of encryption Bob substitutes the plaintext $(x_1, x_2, \ldots, x_n)$ and $\zeta_y = 1$ in public key and gets n linear equations in ciphertext variables $y_1, y_2, \ldots, y_n$. Now Bob solve these linear equations by Gaussian elimination method to gets $y^{''} = (y_1, y_2, \ldots, y_n)$.

3. The ordered pair $(y^{'}, y^{''})$ is the required ciphertext.

4.4 Decryption

Here we describe the decryption algorithm.
Input: Ciphertext $(y^{'}, y^{''})$ and secret parameters $(s, t, \alpha, \beta, \gamma)$.
Output: Message $(x_1, x_2, \ldots, x_n)$

```
1: $v_1 \leftarrow t(y^{'})$ and $v_2 \leftarrow t(y^{''})$
2: $z_1 \leftarrow L_\beta(v_1) + \gamma$ and $z_2 \leftarrow L_\beta(v_2) + \gamma$
3: $z_3^{'} \leftarrow (z_1)^{2^m-1}$ and $z_4^{'} \leftarrow (z_2)^{2^m-1}$
4: $z_3 \leftarrow z_3^{'} + Tr(v_1)$ and $z_4 \leftarrow z_4^{'} + Tr(v_2)$
5: $z_5 \leftarrow z_3^{2^m} + z_3 + \alpha + 1$ and $z_6 \leftarrow z_4^{2^m} + z_4 + \alpha + 1$
6: $z_7 \leftarrow z_5^{2^m-1}$ and $z_8 \leftarrow z_6^{2^m-1}$
7: $X_1 \leftarrow s^{-1}(z_3 + 1)$, $X_2 \leftarrow s^{-1}(z_4 + 1)$, $X_3 \leftarrow s^{-1}(z_3 + z_7 + 1)$ and $X_4 \leftarrow s^{-1}(z_4 + z_8 + 1)$
8: Return $(X_1, X_2, X_3, X_4)$
```

Between X_1, X_2, X_3, X_4 one X_i will be the correct message. There are only four choices for the message, and it will be easy to identify the correct one.

Theorem 4.1 *Given ciphertext, the decryption algorithm outputs a valid plaintext.*

Proof. We prove that the procedure described above outputs a valid plaintext. The relation between plaintext and ciphertext is $(u^{2^m} + u + \alpha)^{2^m-1} + u = (L_\beta(v) + \gamma)^{2^m-1} + Tr(v)$, or equivalently, $u^{2^m} + u + \alpha = (u + z)^{2^m+1}$, where $z = (L_\beta(v) + \gamma)^{2^m-1} + Tr(v)$ which can be converted to the form $(u + z + 1)^{2^m+1} + z + z^{2^m} + \alpha + 1 = 0$. There are only two possibilities either $u = z + 1$ or $u \neq z + 1$. If $u = z + 1$, then $x = s^{-1}(z + 1)$. If $u \neq z + 1$, then raising both sides power $2^m - 1$ in the relation $(u + z + 1)^{2^m+1} = z + z^{2^m} + \alpha + 1$, we get $(u + z + 1) = (z + z^{2^m} + \alpha + 1)^{2^m-1}$ or $u = z + 1 + (z + z^{2^m} + \alpha + 1)^{2^m-1}$ which implies $x = s^{-1}\left(z + 1 + \left(z + z^{2^m} + \alpha + 1\right)^{2^m-1}\right)$. ■

Example 4.1 Here is a toy example for our cryptosystem. We are considering the finite field $\mathbb{F}_{2^3}$, that is, $m = 2$ and $n = 3$. The polynomial x^3+x+1 is irreducible over $\mathbb{F}_2$. Suppose ϑ is the root of this polynomial in the extension field of $\mathbb{F}_2$, i.e., $\vartheta^3+\vartheta+1 = 0$. Using the basis $\{1, \vartheta, \vartheta^2\}$ the finite field $\mathbb{F}_{2^3}$ can be expressed as $\mathbb{F}_{2^3} = \{0, 1, \vartheta, \vartheta^2, 1+\vartheta, 1+\vartheta^2, \vartheta+\vartheta^2, 1+\vartheta+\vartheta^2\}$. We are taking $\alpha = \gamma = 1 + \vartheta + \vartheta^2$, as $tr(1 + \vartheta + \vartheta^2) \neq 0$, and $\beta = 1 + \vartheta$. Corresponding to $\beta = 1 + \vartheta$, $L_\beta = x + x^2$. We are taking invertible transformation $s(x) = A_1x + c_1$ and $t(x) = A_2x + c_2$, where

$$A_1 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, A_2 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \; c_1 = (1, 0, 1)^T \quad \text{and} \quad c_2 = (0, 1, 0)^T.$$

Suppose $x \in \mathbb{F}_{2^3}$, then x can be expressed as $x = x_1 + x_2\vartheta + x_3\vartheta^2$, where $x_i \in \mathbb{F}_2$. Taking $x = (x_1, x_2, x_3)^T$, we have $A_1x + c_1 = (x_1 + x_2 + 1, x_2 + x_3, x_3 + 1)^T$ and $A_2x + c_2 = (x_1 + x_2 + x_3, x_2 + x_3 + 1, x_3)^T$. For the plaintext variable $x = (x_1, x_2, x_3)$ the corresponding ciphertext variable is $y = (y_1, y_2, y_3)$. We have $u = (x_1 + x_2 + 1) + (x_2 + x_3)\vartheta + (x_3 + 1)\vartheta^2$ and $v = (y_1 + y_2 + y_3) + (y_2 + y_3 + 1)\vartheta + y_3\vartheta^2$. The relation between plaintext and ciphertext is

$$\begin{aligned}(u^{2^m} + u + \alpha)^{2^m}(L_\beta(v) + \gamma) + u(u^{2^m} + u + \alpha)(L_\beta(v) + \gamma)& \\ +(u^{2^m} + u + \alpha)(L_\beta(v) + \gamma)^{2^m} + Tr(v)(u^{2^m} + u + \alpha)(L_\beta(v) + \gamma) &= 0.\end{aligned}$$

Substituting u and v and $\alpha = \gamma = 1 + \vartheta + \vartheta^2$, and $L_\beta(x) = x + x^2$ and $Tr(v) = \zeta_y$ we have the following relation between plaintext and ciphertext:

$$\begin{aligned}\{1 + \zeta_y + (1 + \zeta_y)x_3y_3 + (1 + \zeta_y)x_3y_2 + \zeta_y x_2y_2 + x_1 + y_2 + x_2x_3 + x_1x_2y_2 +& \\ x_1x_3y_2 + x_1x_3y_3\} + \vartheta\{x_2 + (\zeta_y + 1)y_2 + y_3 + \zeta_y x_3 + \zeta_y x_3y_2 + (\zeta_y + 1)x_3y_3 +& \\ \zeta_y x_2y_3 + x_1x_3 + x_1y_2 + x_1x_3y_3 + x_1x_2y_2 + x_2y_2 + x_2x_3y_3\} +& \\ \vartheta^2\{1 + x_3 + (\zeta_y + 1)y_3 + \zeta_y x_2 + \zeta_y y_2 + (\zeta_y + 1)x_2y_2 + \zeta_y x_2y_3 + (\zeta_y + 1)x_3y_2 +& \\ x_3y_3 + x_1x_2 + x_1y_2 + x_1y_3 + x_1x_2y_2 + x_1x_2y_3 + x_1x_3y_2 + x_2x_3y_2\} &= 0,\end{aligned}$$

or, equivalently,

$$\begin{aligned}1 + \zeta_y + (1 + \zeta_y)x_3y_3 + (1 + \zeta_y)x_3y_2 + \zeta_y x_2y_2 +& \\ x_1 + y_2 + x_2x_3 + x_1x_2y_2 + x_1x_3y_2 + x_1x_3y_3 &= 0, \\ x_2 + (\zeta_y + 1)y_2 + y_3 + \zeta_y x_3 + \zeta_y x_3y_2 + (\zeta_y + 1)x_3y_3 + \zeta_y x_2y_3 +& \\ x_1x_3 + x_1y_2 + x_1x_3y_3 + x_1x_2y_2 + x_2y_2 + x_2x_3y_3 &= 0, \\ 1 + x_3 + (\zeta_y + 1)y_3 + \zeta_y x_2 + \zeta_y y_2 + (\zeta_y + 1)x_2y_2 + \zeta_y x_2y_3 + (\zeta_y + 1)x_3y_2 +& \\ +x_3y_3 + x_1x_2 + x_1y_2 + x_1y_3 + x_1x_2y_2 + x_1x_2y_3 + x_1x_3y_2 + x_2x_3y_2 &= 0.\end{aligned}$$

Above equations represent the required public key. Note that the above equations are non-linear in plaintext variables (x_1, x_2, x_3) and linear in ciphertext variables (y_1, y_2, y_3).

5 Security of the proposed Cryptosystem

In this section, we discuss the security of the proposed cryptosystem. In general, it is difficult to prove the security of a public key cryptosystem. 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. In this section we will give some security arguments and evidence that our cryptosystem is secure. We are using the polynomials $(x^{2^m} + x + \alpha)^{2^m-1} + x$, and $(L_\beta(x) + \gamma)^{2^m-1} + Tr(x)$ where α, β and γ are secret. Thus, if we write the polynomial $(x^{2^m} + x + \alpha)^{2^m-1} + x$ in the form $\sum_{i=0}^{d} p_i x^i$ then some of coefficients will be 0 and 1 and the rest will be some functions of α. Since α is secret, most of the coefficients of this polynomial are also secret. Similarly, if we write the polynomial $(L_\beta(x) + \gamma)^{2^m-1} + Tr(x)$ in the form $\sum_{i=0}^{d} p_i' x^i$ then all the coefficients of this polynomial are secret because β and γ both are secret. One important point is that the degrees of these polynomials are not constant but are functions of n, as $m = (n + 1)/2$. The Coppersmith-Patarin attack on Little Dragon cryptosystem [3] is due to the use of the monomial x^n to design the Little Dragon cryptosystem, so this attack is not applicable to our cryptosystem. 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 Linearization equation, Gröbner basis, univariate polynomial representation, Differential cryptanalysis, Relinearization, XL and FXL algorithms attacks.

5.1 Linearization Equation Attacks

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

$$\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a_{ijl}x_iy_j + \sum_{i=0}^{n-1} b_{il}x_i + \sum_{j=0}^{n-1} c_{jl}y_j + d_l, \tag{4}$$

where $l = 0, 1, \ldots, n-1$ and $y_i = f_i$.

This attack was first successfully applied by Patarin in [6] to break the Matsumoto-Imai cryptosystem C^* [5]. Patarin's idea 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 (4) can be obtained, 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 obtained and solved. We are not using the monomial x^{q^i+1}. Moreover, in our cryptosystem the plaintext and ciphertext are connected by the relation $(u^{2^m} + u + \alpha)^{2^m-1} + u = (L_\beta(v) + \gamma)^{2^m-1} + Tr(v)$ where α, β, γ are secret. So, in our case it is not possible to obtain a relation of the form (4). However, one 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. Suppose $(L_\beta(v) + \gamma)^{2^m-1} + Tr(v) = z$, then our relation between plaintext and ciphertext is $(u^{2^m} + u + \alpha)^{2^m-1} + u = z$ or equivalently we have $(u^{2^m} + u + \alpha) = (z + u)^{2^m+1}$ which is further equivalent to $(z + u + 1)^{2^m+1} = z^{2^m} + z + \alpha + 1$. If $z+u+1 \neq 0$ then raising both sides to the power 2^m-1 we get $(z+u+1) = (z^{2^m}+z+\alpha+1)^{2^m-1}$. If $z^{2^m} + z + \alpha + 1 \neq 0$, then we have the following relation between plaintext and ciphertext $(z + u + 1)(z^{2^m} + z + \alpha + 1) + (z^{2^m} + z + \alpha + 1)^{2^m} = 0$. This relation gives equations which are linear in plaintext variables x_i and non-linear in ciphertext variables y_i. Note that $z = (L_\beta(v)+\gamma)^{2^m-1}+Tr(v)$ will give non linear equations of degree $w(2^m-1)$, where $w(2^m-1)$ denotes the weight of 2^m-1 so the relation $(z+u+1)(z^{2^m}+z+\alpha+1)+(z^{2^m}+z+\alpha+1)^{2^m} = 0$ gives equations of non linear degree $2w(2^m-1)$ in ciphertext variables y_i. Thus, this line of attack is completely infeasible.

5.2 Attacks with Differential Cryptanalysis

Differential cryptanalysis has been successfully used earlier to attack symmetric cryptosystems. In recent years differential cryptanalysis has emerged as a powerful tool to attack the multivariate public key cryptosystems too. In 2005 [14] 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 MIC* cryptosystem and afterwards using the same technique Dubois, Fouque, Shamir and Sterm in 2007 [16] have completely broken all versions of the SFLASH signature scheme proposed by Patarin, Courtois, and Goubin [15]. In our cryptosystem instead of using monomial of the form x^{q^i+1}, we are using the polynomials $(x^{2^m} + x + \alpha)^{2^m-1} + x$ and $(L_\beta(v) + \gamma)^{2^m-1} + Tr(v)$, where α, β, γ are secret. Clearly the degree of these polynomial are not quadratic. Moreover, the public key in our cryptosystem is of mixed type. Substituting the ciphertext gives quadratic equations in plaintext variables but in that case, they will be different for different ciphertexts. So, attacking our cryptosystem by the methods of [14] and [16] is not feasible.

5.3 Gröbner Basis Attacks

After substituting the ciphertext in public key one can get n quadratic equations in n variables and then Gröbner basis techniques can be applied to solve the system. The classical algorithm for solving a system of multivariate equations is Buchberger's algorithm [4]. Although it can solve all the multivariate quadratic equations in theory, its complexity is exponential in the number of variables. We remark that there is no closed-form formula for its complexity. 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 [17]). There are some efficient variants F_4 and F_5 of Buchberger's algorithm given by Jean-Charles Faugere (see [19] and [20]). The complexity of computing a Gröbner basis by Buchberger's algorithm for the public key polynomials of the basic HFE scheme is too high to be feasible. 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 [10]. Moreover, it has been expressed in [10] "a crucial point in the cryptanalysis of HFE is the ability to distinguish a randomly algebraic system from an algebraic system coming from HFE". 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 are using a polynomial which has degree proportional to n. It is explained in [10] that in this case there does not seem to exist polynomial time algorithm to compute the Gröbner basis. Hence, Gröbner basis attacks on our cryptosystem are not feasible.

5.4 Relinearization, XL and FXL Algorithms.

Relinearization, XL or FXL algorithms [9], [17] are the techniques to solve an overdefined system of equations i.e., εn^2 equations in n variables, where $\varepsilon \geq 0$. To attack the HFE cryptosystem, first the equivalent quadratic polynomial representation of HFE public key was obtained and then using the matrix representation of quadratic polynomials, $O(n^2)$ polynomial equations in $O(n)$ variables were obtained [9]. The Relinearization and XL or FXL techniques are used to solve this system of equations. Note that our polynomials are not quadratic. Moreover, the degrees of our polynomials are not constant, but are functions of n. So, the attack as given in [9] is not feasible to our cryptosystem. Adversary can not use directly Relinearization, XL or FXL algorithms to attack our cryptosystem, because when the number of equations is equal to the number of variables, the complexities of these algorithms is 2^n.

6 Efficiency of the proposed cryptosystem

In this section we discuss complexities of the encryption and decryption of our cryptosystem.

6.1 Encryption

There are $O(n^2)$ terms of the form $x_i x_j$ in each n equations of the public key. So the complexity of evaluating public key at message block $x_1, \ldots, x_n$ is $O(n^3)$. The next step of encryption is to solve the n linear equations in n ciphertext variables $y_0, y_1, \ldots, y_n$. This can be done efficiently by Gaussian elimination with complexity $O(n^3)$. Hence the complexity of encryption is $O(n^3)$.

6.2 Decryption

For decryption in the proposed cryptosystem we need four exponentiations namely $z_3^{'} \leftarrow (z_1)^{2^m-1}$, $z_4^{'} \leftarrow (z_2)^{2^m-1}$, $z_7 \leftarrow z_5^{2^m-1}$ and $z_8 \leftarrow z_6^{2^m-1}$. So the complexity of decryption is equivalent to Dragon cryptosystems [3], [8]. Note that for exponentiation in finite fields $\mathbb{F}_{2^n}$, there are several efficient algorithms, so the exponentiation can be performed very efficiently. The exact complexity of exponentiation will depend on the algorithm used for exponentiation.

7 Conclusion

We have designed an efficient multivariate public key cryptosystem. Like in Big Dragon Cryptosystem, the public key is mixed type of total degree three, two in plaintext variables and one in ciphertext variables. The public key size can be reduced by writing it as two sets of quadratic equations. The efficiency of encryption is equivalent to those of other multivariate public key cryptosystems, that is, $O(n^3)$, where n is the bit size. For decryption we need four exponentiations in the binary field, and therefore decryption process is fast. Our algorithm is bijective and can be used for both encryption and signature generation. It is not clear at this stage how to design a secure cryptosystem like Little Dragon cryptosystem that is a cryptosystem having public key mixed type but quadratic. Further investigations to develop an

Little Dragon type cryptosystem using permutation polynomials over finite fields can be an interesting topic of future work.

8 Acknowledgement

The authors are grateful to Professor Lei Hu of Chinese Academy of Sciences, Beijing, China for his constructive suggestions and for critical evaluation of a previous work by the authors which inspired for the present work.

References

[1] Peter Shor. Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Journal on Scientific Computing.* 26 (1997), 1484.

[2] R. Lidl and H. Niederreiter. *Finite Fields.* Addision-Wesley, 1983.

[3] Neal Koblitz. *Algebraic Aspects of Cryptography.* Springer, 1998

[4] 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.

[5] T. Matsumoto and H.Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. *Eurocrypt '88,* Springer-Verlag (1988), pp. 419453.

[6] J.Patarin. Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt '88, *Advances in Cryptology-Crypto '95,* Springer-Verlag, 248261.

[7] 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.

[8] J.Patarin. Asymmetric cryptography with a hidden monomial. *Advances in Cryptology-Crypto '96,* Springer-Verlag, pp. 4560.

[9] Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization. *CRYPTO '99,* LNCS Vol. 1666, pp. 19-30, 1999.

[10] Jean-Charles Faugere and Antoine Joux. Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Basis. *in CRYPTO '2003,* LNCS Vol. 2729, pp. 44-60, 2003.

[11] 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

[12] Farshid Delgosha and Faramarz Fekri. Public-Key Cryptography Using Paraunitary Matrices, *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, VOL. 54, NO. 9, SEPTEMBER 2006.

[13] Christopher Wolf and Bart Preneel. Taxonomy of Public Key Schemes based on the problem of Multivariate Quadratic equations. *Cryptology eprint Archive,* 2005/077.

[14] Pierre-Alain Fouque, Louis Granboulan, and Jacques Sterm. Differential Cryptanalysis for Multivariate Schemes. *in EUROCRYPT 2005,* LNCS Vol. 3494, pp. 341-353, 2005.

[15] J. Patarin, N. T. Courtois, and L. Goubin. FLASH, a fast multivariate signature algorithm. *in CT-RSA01, '2001,* LNCS Vol. 2020, pp. 298-307.

[16] 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.

[17] 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.

[18] N.T. Courtois, J. Patarin. About the XL algorithm over $\mathbb{F}_2$. *CT-RSA'03, '2000,* LNCS Vol. 2612, pp. 141-157.

[19] Faugere, Jean-Charles,. A New efficient algorithm for computing Gröbner bases (F_4). *Journal of Pure and Applied Algebra '2002,* 139: 61-88

[20] Faugere, Jean-Charles,. A New efficient algorithm for computing Gröbner bases without reduction to zero (F_5). *International Symposium on Symbolic and Algebraic Computation - ISSAC '2002,* pages 75-83. ACM Press.