Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080559

Hexi McEliece Public Key Cryptosystem

K. Ilanthenral[*] *and K. S. Easwarakumar*

Department of Computer Science and Engineering, Anna University, Chennai 600 025, India

Received: 1 Oct. 2013, Revised: 29 Dec. 2013, Accepted: 30 Dec. 2013
Published online: 1 Sep. 2014

Abstract: This paper introduces a new class of hexi codes namely, hexi polynomial codes, hexi Rank Distance codes, hexi Maximum Rank Distance codes, hexi Goppa codes and hexi wild Goppa codes. These codes are useful to create variants of the McEliece public key cryptosystem known as the hexi McEliece public key cryptosystem and its variants; these cryptosystems are secure against attacks carried out on the existing variants of the McEliece public key cryptosystem. This newly introduced cryptosystem has better error correcting capacity and lesser time complexity making it more feasible to use. The security and possible attacks on these variants of the hexi McEliece public key cryptosystem are analysed.

Keywords: McEliece public key cryptosystem; Gabidulin - Paramonov - Tretjakov (GPT) public key cryptosystem; Wild McEliece cryptosystem; Goppa codes; Maximum Rank Distance (MRD) codes, hexi codes; hexi polynomial codes; hexi Maximum Rank Distance (MRD) codes, hexi wild Goppa codes, hexi McEliece public key cryptosystem

1 Introduction

The McEliece public key cryptosystem introduced by McEliece in the year 1978 [19], still remains unbroken. The public key cryptosystem is based on binary Goppa codes. Hexi codes were developed in 2013 for error correction in AES [14], further development of these codes is carried out in this paper. These codes are useful to create variants of the McEliece public key cryptosystem called the hexi McEliece public key cryptosystem and its variants. These public key cryptosystems are secure, have better error correcting capacity and lesser time complexity making it more advantageous to use. The organization of the rest of this paper is as follows. The history of the McEliece public key cryptosystem and its several variants are dealt in section two. Section three recalls hexi codes and introduces hexi polynomial codes, hexi Rank Distance (hexi RD) codes, hexi Maximum Rank Distance (hexi MRD) codes, hexi Goppa codes and hexi wild Goppa codes. The decoding, error detecting and error correcting capacity of these codes is discussed in section four. Section five introduces a few variants of the McEliece public key cryptosystems which are based on these new hexi codes; they are called the hexi McEliece public key cryptosystem and its variants. Section six deals with the possible attacks on the hexi McEliece public key cryptosystem and the resistance against these attacks. It also discusses the security of the cryptosystem. Section seven provides a comparison of the hexi McEliece public key cryptosystem with original McEliece public key cryptosystem, in terms of time complexity and error correcting capacity. Conclusions, suggestions and future direction of research are given in section eight.

2 Variants of the McEliece Cryptosystem

The original version of the McEliece public key cryptosystem which uses Goppa codes remains unbroken. It was the first public key cryptosystem based on coding theory, making it an important candidate for post quantum cryptography. In 1986, Niederreiter [21] proposed an equivalent to the McEliece public key cryptosystem, where the Goppa code was replaced by Generalised Reed Solomon (GRS) code. This proposal was proved to be insecure by Sidelnikov and Shestakov [28] in 1992. The variants of the McEliece public key cryptosystem that are used to create new cryptosystems are discussed here.

There have also been several attempts to create cryptographic protocols based on coding theory, the most successful protocols being the signature scheme by Courtois, Finiasz and Sendrier in 2001 [3] and the identification scheme by Stern in 1995 [29].

[*] Corresponding author e-mail: ilanthenral@gmail.com

2.1 The GPT public key cryptosystem

The public key cryptosystem based on Rank Distance error correcting codes, known as Gabidulin - Paramonov - Tretjakov (GPT) public key cryptosystem was introduced in 1991 [5,6]. Since the Rank Distance codes used in this system are well structured, attacks on the GPT public key cryptosystem are easier.

Gibson's attack: Gibson in a series of work [11,12] developed attacks that break the GPT public key cryptosystem. Several variants of the GPT public key cryptosystem were introduced to withstand Gibson's attack [7,22]. A rectangular row scramble matrix was used instead of a square matrix, it allowed to work with subcodes of Rank Distance codes which have more complicated structure. A modification of MRD codes was exploited to introduce the concept of a column scramble matrix. Reducible codes were introduced and also implemented to modify the GPT public key cryptosystem [8]. All these variants withstood Gibson's attack.

Overbeck's attack: An attack which was more effective than any of Gibson's attack, was proposed by Overbeck [23,24]. Many instances of the GPT cryptosystem were broken by Overbeck by using a generalization and development of one of Gibson's ideas. It was found in [9] that a proper column scrambler can be defined over the extension field without any violation of the standard mode of the public key cryptosystem. Overbeck's attack failed in this case. In 2009 [10], a proper choice of column scramblers over the extension field was taken to other variants of the GPT cryptosystem. This choice withstood both Gibson's and Overbeck's attacks.

2.2 Wild McEliece Cryptosystem

The wild McEliece public key cryptosystem [1,2,26] was proposed using wild Goppa codes, which are subfield codes over small F_q that have an increase in error correcting capability by a factor of about $q/(q-1)$. McEliece's construction using binary Goppa codes is a special case where $q = 2$ of this wild McEliece cryptosystem. The advantage of the wild Goppa codes is that they efficiently correct $\lfloor qt/2 \rfloor$ errors (or slightly more with the help of list decoding); for $q \in 3,4,\ldots,$ this is strikingly better than the performance of an irreducible polynomial of the same degree $(q-1)t$ namely correcting $\lfloor (q-1)t/2 \rfloor$ errors.

3 Hexi codes and related hexi codes

This section recalls some definitions regarding hexi codes and introduces other related codes like hexi polynomial codes, hexi Rank Distance codes, hexi MRD codes, hexi Goppa codes and hexi wild Goppa codes.

Table 1: Addition table ⊕ of the hexi field S

⊕	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	0	3	2	5	4	7	6	9	8	B	A	D	C	F	E
2	2	3	0	1	6	7	4	5	A	B	8	9	E	F	C	D
3	3	2	1	0	7	6	5	4	B	A	9	8	F	E	D	C
4	4	5	6	7	0	1	2	3	C	D	E	F	8	9	A	B
5	5	4	7	6	1	0	3	2	D	C	F	E	9	8	B	A
6	6	7	4	5	2	3	0	1	E	F	C	D	A	B	8	9
7	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8
8	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7
9	9	8	B	A	D	C	F	E	1	0	3	2	5	4	7	6
A	A	B	8	9	E	F	C	D	2	3	0	1	6	7	4	5
B	B	A	9	8	F	E	D	C	3	2	1	0	7	6	5	4
C	C	D	E	F	8	9	A	B	4	5	6	7	0	1	2	3
D	D	C	F	E	9	8	B	A	5	4	7	6	1	0	3	2
E	E	F	C	D	A	B	8	9	6	7	4	5	2	3	0	1
F	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

Table 2: Multiplication table ⊗ of the hexi field S

⊗	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	0	2	4	6	8	A	C	E	3	1	7	5	B	9	F	D
3	0	3	6	5	C	F	A	9	B	8	D	E	7	4	1	2
4	0	4	8	C	3	7	B	F	6	2	E	A	5	1	D	9
5	0	5	A	F	7	2	D	8	E	B	4	1	9	C	3	6
6	0	6	C	A	B	D	7	1	5	3	9	F	E	8	2	4
7	0	7	E	9	F	8	1	6	D	A	3	4	2	5	C	B
8	0	8	3	B	6	E	5	D	C	4	F	7	A	2	9	1
9	0	9	1	8	2	B	3	A	4	D	5	C	6	F	7	E
A	0	A	7	D	E	4	9	3	F	5	8	2	1	B	6	C
B	0	B	5	E	A	1	F	4	7	C	2	9	D	6	8	3
C	0	C	B	7	5	9	E	2	A	6	1	D	F	3	4	8
D	0	D	9	4	1	C	8	5	2	F	B	6	3	E	A	7
E	0	E	F	1	D	3	2	C	9	7	6	8	4	A	B	5
F	0	F	D	2	9	6	4	B	1	E	C	3	8	7	5	A

Hexi codes and hexi polynomial codes were introduced along with other hexi codes in 2013 [14]. The definition of hexi field, hexi code and hexi polynomial code are recalled.

Let $S = Z_{2^4}$ be a field of 16 elements which is isomorphic to

$$\frac{Z_2[x]}{\langle x^4 + x + 1 \rangle}$$

where $\langle x^4 + x + 1 \rangle$ is the ideal generated by the irreducible polynomial $x^4 + x + 1$ in $Z_2[x]$. Now the elements are given hexadecimal notation, where 0 = 0000, 1 = 0001, 2 = 0010, 3 = 0011, 4 = 0100, 5 = 0101, 6 = 0110, 7 = 0111, 8 = 1000, 9 = 1001, A = 1010, B = 1011, C = 1100, D = 1101, E = 1110 and F = 1111. In short $S = \{0,1,2,\ldots,9,A,\ldots,F\}$. Clearly $(S, \oplus, \otimes)$ is a field of order 16. The operator '⊕' denotes XOR modulo 2, is given in Table 1 and each element is inverse of itself with respect to ⊕. The operator '⊗' denotes multiplication modulo $x^4 + x + 1$ is given in Table 2. This operator '⊗' multiplication modulo $x^4 + x + 1$ was used in Mini AES in [27] and also described in [14]. This field is called as hexi field.

Let $V^n = \{(x_1 \ldots x_n) | x_i \in S;\ 1 \leq i \leq n\}$ be a n-dimensional vector space defined over S.

Definition 1.*A block code of length n with $(2^4)^k$ codewords is called a hexi (n, k) block code, denoted by C_H(n, k), if and only if its $(2^4)^k$ codewords form a k-dimensional subspace of the vector space V^n of all n tuples over the hexi field S.*

The method for generating the $C_H(n,k)$ code using the generator matrix G is as follows. G is given in the following matrix;

$$G = \begin{bmatrix} g_{00} & g_{01} & g_{02} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & g_{12} & \cdots & g_{1,n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \cdots & g_{k-1,n-1} \end{bmatrix}$$

$g_{i,j} \in S$; for $0 \leq i \leq k-1$ and $0 \leq j \leq n-1$. Consider u = $(u_0 \;\; u_1 \;\; \ldots u_{k-1})$, the message to be encoded, the corresponding codeword v is given by $v = u.G$. Every codeword v in $C_H(n,k)$ is a linear combination of k codewords.

Hexi polynomial codes are of two types, x^n + 1 and x^n + z ($z \in S \setminus \{0\}$ and $z \neq 1$). When x^n+ 1 is used, it forms a usual cyclic code, $g(x)$ is a polynomial which divides (x^n+ 1) and its coefficients are from S. To generate a $C_H(n,k)$ cyclic hexi code, consider only the polynomial of the form x^n + 1. Instead of x^n + 1, consider x^n + z ($z \in S \setminus \{0\}$); $z \neq 1$, then x^n + $z = g(x) \times h(x)$, $g(x)$ and $h(x)$ are polynomials belonging to $S[x]$. Let G be the generator matrix associated with generator polynomial $g(x)$. Let H be the parity check matrix associated with the parity check polynomial $h(x)$. The $C_H(n,k)$ hexi code is not cyclic. Clearly GH^T = (0). If $(x_1 \ldots x_n) \in C_H(n,k)$, then in general $(x_n \; x_1 \ldots x_{n-1}) \notin C_H(n,k)$.

$C_H(n,k)$, the hexi polynomial code generated by the polynomial $g(x)$ is defined as follows.

Definition 2.*Let x^n + $z \in S[x]$, $z \in S \setminus \{0, 1\}$, be a hexi polynomial in S[x]. If x^n + z = g(x) h(x) where g(x) is the hexi generator polynomial associated with the generator matrix G and h(x) is the hexi parity check polynomial associated with the parity check matrix H. If g(x) generates a code C_H(n, k), then C_H(n, k) is defined as the hexi polynomial code associated with the hexi generator polynomial g(x).*

Let $g(x) = g_0$ + $g_1 x$ + ... + $g_m x^m$ be the hexi generator polynomial, then the generator matrix G of the hexi polynomial code $C_H(n,k)$ is as follows:

$$G = \begin{bmatrix} g_0 & g_1 & \ldots & g_m & 0 & \ldots & 0 \\ 0 & g_0 & \ldots & g_{m-1} & g_m & \ldots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & g_0 & g_1 & \ldots & g_m \end{bmatrix}$$

$g_i \in S$; for $0 \leq i \leq m$. The rows of the generator matrix G are linearly independent and rank of G is k, the dimension of $C_H(n,k)$, k is the number of message symbols and m is the highest degree of the generator polynomial $g(x)$ and $n = m + k$, is the length of the codeword. A message $u = u_0 u_1 \ldots u_{k-1}$ can be represented as $u(x)$ = $u_0 x^0$ + $u_1 x^1$ + ... + $u_{k-1} x^{k-1}$ and can be encoded as $u(x) \times g(x)$.

An example of hexi polynomial code, decoding, error detection and error correcting capacity are discussed later in sections 4 and 5 of this paper. These hexi polynomial codes can be generated by splitting $x^n + z$ into two polynomials $g(x)$ and $h(x)$. Given n, m and k it is not possible to easily guess which polynomial has been used as the generating polynomial. These codes are not cyclic, making it harder to break.

A rank distance code which is defined over the hexi field S will be known as the hexi Rank Distance code.

Definition 3.*Let C_H be a (n,k) hexi Rank Distance code (hexi RD code). The hexi minimum distance of C_H is defined as $d_H = min\{r_H(x-y) | x,y \in C_H; x \neq y\}$ where $r_H(x)$ is rank of x. Since C_H is a linear k dimensional hexi subspace of the hexi rank distance space V_n, if $x,y \in C_H$ then $x - y \in C_H$.*

A linear (n,k) Rank Distance code with minimum distance d_H satisfies the bound $d_H \leq n - k + 1$. The hexi Maximum Rank Distance codes are analogous to the Maximum Rank Distance codes introduced by Gabidulin in 1985 [4], where the field F_q is replaced by S.

Definition 4.*Let C_H be a (n,k) hexi Rank Distance code, it is said to be a hexi Maximum Rank Distance code (hexi MRD code) if the minimum distance $d_H = n - k + 1$.*

Classical Goppa codes was introduced by Valery D. Goppa in 1970 [13], based on these codes, here in this paper hexi Goppa codes are introduced.

Fix a prime power $q = 2^4$, the order of the hexi field S; m a positive integer and n a positive integer such that $n \leq q^m = 2^{4^m}$; t be an integer where $t < n/m$; $a_1^H, \ldots, a_n^H$ be distinct elements in $F_{2^{4m}}$; and $g_H(x)$ be a special polynomial in $F_{2^{4m}}[x]$ of degree t such that $g_H(a_i) \neq 0$ for all i. The hexi codewords c_H = $(c_1^H \ldots c_n^H)$ in $F_{2^{4m}}^n$ with

$$\sum_{i=1}^{n} \frac{c_i^H}{x - a_i^H} \equiv 0 (mod \quad g_H(x)) \qquad i \in \mathbb{N} \qquad (1)$$

form a linear hexi code $\Gamma_{2^{4m}}^H$ $(a_1^H, \ldots, a_n^H, g_H)$ with length n and dimension $n - t$ over $F_{2^{4m}}$.

The hexi code $\Gamma_{2^{4m}}^H$ $(a_1^H, \ldots, a_n^H, g_H)$ is a special case of a generalised Reed Solomon hexi code. The restriction of a generalised Reed Solomon hexi code over $F_{2^{4m}}$ be a subfield of F_{2^4} and is called as an alternate hexi code, in general the restriction of a code to a smaller field is called a subfield subcode.

Definition 5.*The hexi Goppa code $\Gamma_{2^4}^H$ $(a_1^H, \ldots, a_n^H, g_H)$ with Goppa hexi polynomial $g_H(x)$ and support $a_1^H, \ldots, a_n^H$ is the restriction of $\Gamma_{2^{4m}}^H$ $(a_1^H, \ldots, a_n^H, g_H)$ to* *the field F_{2^4} = S, i.e., the set of elements $(c_1^H, \ldots, c_n^H)$ in $F_{2^4}^n$ that satisfy Equation 1.*

Note: Here the chosen hexi Goppa polynomial g_H does not vanish at the support elements $(a_1^H, \ldots, a_n^H)$, it is common to choose g_H to be a non linear irreducible element of $F_{2^{4m}}[x]$.

In this case $\Gamma_{2^4}^H$ $(a_1^H, \ldots, a_n^H, g_H)$ is defined as an irreducible hexi Goppa code. The hexi Goppa code $\Gamma_{2^4}^H(a_1^H, \ldots, a_n^H, g_H)$ is a hexi subfield hexi subcode of $\Gamma_{2^{4m}}(a_1^H, \ldots, a_n^H, g_H)$. The hexi dimension of $\Gamma_{2^4}^H(a_1^H, \ldots, a_n^H, g_H)$ is at least $n - mt$. The minimum distance of $\Gamma_{2^{4m}}$ $(a_1^H, \ldots, a_n^H, g_H)$ is $\geq t+1$ that is a consequence of hexi Goppa codes being part of the family of hexi alternate codes/ generalised Reed Solomon hexi codes.

Definition 6.*The hexi Goppa code of the form $\Gamma_{2^4}^H$ $(a_1^H, \ldots, a_n^H, g_H^{2^4-1})$ where g_H is an irreducible monic hexi polynomial in $F_{2^{4m}}[x]$ of degree t is called as hexi wild Goppa code.*

The decoding procedures of the above defined codes are described in the following section.

4 Decoding of hexi codes and related codes

The decoding, error detection and error correction capacity of hexi codes and hexi polynomial codes are discussed in detail in this section. Only methods that can be used to decode and perform error correction of other hexi codes are mentioned, as these newly introduced hexi codes are similar to their respective counterparts.

Some definitions are recalled from [14] to make this paper a self contained one. The Hamming metric of the hexi code is given in the following:

Definition 7.*For any 2 vectors x = $(x_1, \ldots, x_n)$ and y = $(y_1, \ldots, y_n)$ in V^n, the Hamming distance d(x, y) and Hamming weight w(x) are defined as follows:*

$$\begin{aligned} d(x,y) &= \mid \{x_i : x_i \neq y_i; x_i \in x; y_i \in y\} \mid \\ w(x) &= \mid \{x_i : x_i \neq 0; x_i \in x\} \mid . \end{aligned} \tag{2}$$

If C_H is a hexi code, the sum of two codewords is also a codeword in C_H. It follows that $d(x,y) = w(x+y)$, that is the Hamming distance between two codewords is equal to the Hamming weight of some other codeword.

Definition 8.*The minimum distance d_{min} of a hexi code C_H is defined as*

$$d_{min} = \min_{\substack{x,y \in C_H \\ x \neq y}} d(x,y). \tag{3}$$

The error correcting capacity of hexi code is discussed.

Table 3: Standard Array for Syndrome decoding

Coset Leaders	Codewords	Syndrome
$v_1 = 0$	$v_2 \ldots v_{2^{4k}}$	$s = 0$
e_2	$e_2 + v_2 \ldots e_2 + v_{2^{4k}}$	$e_2 H^T$
e_3	$e_3 + v_2 \ldots e_3 + v_{2^{4k}}$	$e_3 H^T$
$\vdots$	$\vdots$	$\vdots$
e_l	$e_l + v_2 \ldots e_l + v_{2^{4k}}$	$e_l H^T$
$\vdots$	$\vdots$	$\vdots$
$e_{2^{4(n-k)}}$	$e_{2^{4(n-k)}} + v_2$ $\ldots e_{2^{4(n-k)}} + v_{2^{4k}}$	$e_{2^{4(n-k)}} H^T$

Theorem 1.*[14] The number of errors a hexi code can correct is $t = \lfloor (d_{min} - 1)/2 \rfloor$, and this code can detect l errors where $t + l + 1 \leq d_{min}$ and $l > t$.*

Proof.: Proof is similar to that of linear block code [18]. Since the shift to hexadecimal system from binary does not alter the calculation of Hamming weight, Hamming distance or d_{min} and the error correcting capacity remains same.

Correction of errors in any code is a complicated process. There are 2^{4k} error patterns that result in same syndrome and the true error pattern e is just one of them. Determining the true error vector e is not easy. The coset leader method is used for error correction, by making use of the standard array and syndrome decoding described in [18]. The standard array is given by Table 3.

Here e_i's are coset leaders, $2 \leq i \leq 2^{4(n-k)}$; v_j's are non zero codewords, $2 \leq j \leq 2^{4k}$. The corrected codeword v_j is obtained by using the syndrome of the received codeword r. The coset leader e_i, related to the syndrome, is added to r to obtain the corrected codeword.

Decoding of Hexi Polynomial Codes

In case of hexi polynomial codes the original message u expressed in polynomial form as $u(x)$ can be encoded as $u(x) * g(x)$ where $g(x)$ is the generator polynomial. Thus without using the generator matrix G the encoding of the message can be carried out. Like in the case of decoding usual polynomial codes [17], the error detection and error correction of hexi polynomial codes can be done without the creation of standard array for syndrome decoding.

Let w be the received codeword, $w(x)$ is divided by the generator polynomial $g(x)$, if the division results with a reminder, it implies that an error has occurred.

To perform the error correction, the received codeword $w(x)$ is multiplied with the parity check polynomial $h(x)$. The resultant is then divided by $h(x)$. Since $w = v + e$, where $v(x)$ is the original codeword and $e(x)$ is the error. This division results in error $e(x)$ as the quotient, the original codeword is obtained by $w - e$. The message is later obtained by the division of the original codeword $v(x)$ by $g(x)$. The hexi polynomial code has a error correction capacity of $n - k$. The algorithm for error detection and error correction of hexi polynomial codes is given in Algorithm 1.

Algorithm 1: DECODING Algorithm for decoding and error correction of hexi polynomial codes

```
Input: $w(x)$ - Received codeword, $g(x)$ - generator
       polynomial, $h(x)$ - parity check polynomial
Output: $v(x)$ - Original codeword, $m$ - message
begin
    Compute $w(x)/g(x)$
    if $w(x)\% \ g(x) \neq 0$ then
        // Error is present in the $w(x)$
        $y(x) \leftarrow w(x) \times h(x) \bmod (x^n + z)$
        $e(x) \leftarrow y(x)/h(x)$
        $v(x) \leftarrow w(x) - e(x)$
        $m(x) \leftarrow v(x)/g(x)$
    else
        $m(x) \leftarrow w(x)/g(x)$
    end
    Return $m$
end
```

Decoding of Hexi MRD Codes and related codes

The hexi Rank Distance codes are analogous to Rank Distance codes which are analogous to Generalised Reed Solomon codes and can be decoded using the parity check matrices, or any other method that does fast decoding of the Rank Distance codes. Similarly the hexi Maximum Rank Distance codes are analogous to Maximum Rank Distance codes which are analogous to Generalised Reed Solomon codes and can be decoded using the parity check matrices, or any other method that does fast decoding of the Maximum Rank Distance codes.

Decoding of Hexi Goppa codes and related codes

The hexi Goppa codes are a special case of Goppa codes, $q = 2^4$, any decoding algorithm such as Patterson's algorithm [25] or list decoding which is used for decoding Goppa codes can be used for hexi Goppa codes, as it is analogous to Goppa codes. The hexi wild Goppa code is a wild Goppa code when the field F_{q^m}, $q = 2^4$ is taken, and this code can efficiently correct $\lfloor qt/2 \rfloor$ errors(slightly more errors when list decoding is used).

5 Variants of Hexi McEliece Cryptosystem

The hexi McEliece public key cryptosystem introduced in this paper is a variant of the classical McEliece public key cryptosystem proposed by McEliece in 1978 [19]. The original system made use of binary Goppa code. Here the hexi based McEliece public key cryptosystem makes use of codes defined over the hexi field S viz. hexi polynomial codes, hexi Maximum Rank Distance codes and hexi wild Goppa codes. Only hexi McEliece public key cryptosystem based on hexi polynomial code is discussed in detail.

5.1 Hexi McEliece Cryptosystem

The hexi McEliece public key cryptosystem is based on hexi polynomial code which is not cyclic in nature. Hexi polynomial code is used instead of binary Goppa code. Hexi GPT cryptosystem and hexi wild McEliece cryptosystem are also discussed. The hexi McEliece public key cryptosystem has better error correcting capacity and less time complexity.

The necessary parameters of the hexi McEliece cryptosystem based on hexi polynomial codes are as follows:
Let G be the generator matrix for a $C_H(n,k)$ hexi polynomial code based on the generator polynomial $g(x) = g_0 + g_1 x + \ldots + g_{n-k} x^{n-k}$ where $x^n + z \in S[x], z \in S \setminus \{0,\ 1\}$, and S_H be the $k \times k$ invertible matrix over the hexi field S. Let P be the $n \times n$ permutation matrix. The decoding of the message can be done in time complexity of $(n-k)lg(n-k)$, if $n = \Theta(n-k)$.
The public key for the cryptosystem will be given by G'

$$G' = S_H \times G \times P \tag{4}$$

where G' is a $k \times n$ matrix.
The error correcting capacity of hexi polynomial code $C_H(n,k)$ with generator matrix G is $n-k$. 2^{n-k} error patterns are generated, depending on the permutation matrix P. Error is added only to the parity elements in the resultant vector. Any of the error pattern e_p is selected and random error e_r of length n is taken. If i^{th} element of the error pattern is 1, then the i^{th} element of error e is the i^{th} element of error e_r, else it is set as 0.

The algorithm for encryption is given by the following Algorithm 2.

Algorithm 2: ENCRYPTION Algorithm for encryption in Hexi McEliece Cryptosystem

```
Input: $m$ - Message, $e_p$ - Error pattern, $e_r$ - Random error,
       $e$ - Final error, $G'$ - Public key
Output: $y$ - Ciphertext
begin
    Compute $m \times G'$
    Select error pattern $e_p$, random error $e_r$ with length $n$
    for $i \leftarrow 1$ to $n$ do
        if $e_{p_i} \neq 0$ then $e_i \leftarrow e_{r_i}$
        else $e_i \leftarrow 0$
    end
    Return $y \leftarrow m \times G' + e$
end
```

The decryption of the hexi McEliece public key cryptosystem is given by Algorithm 3. The decoding and error correction of the hexi polynomial code is given in Algorithm 1.

An example is discussed here in detail to illustrate the working of the hexi McEliece public key cryptosystem

Algorithm 3: DECRYPTION Algorithm for hexi McEliece Decryption

```
Input: $y_1$ - Ciphertext, $G, S_H, P$- Private key, $C_H(n,k)$ -
       Hexi code
Output: $m$ - Message
begin
    Compute $y_1 \times P^{-1}$
    Use decoding algorithm to remove error and obtain
    codeword $mS_HG$
    Compute $m_0$ such that $m_0 = mG$
    Calculate $m = m_0 S_H^{-1}$
    Return original message $m$
end
```

which is based on hexi polynomial code.

Consider the hexi polynomial $x^7 + F \in S[x]$. Let $x^7 + F = g(x) * h(x)$, where $g(x) = x^4 + Cx^3 + Fx^2 + A$ be the generator hexi polynomial and $h(x) = x^3 + Cx^2 + 8$ be the parity check hexi polynomial. The generator matrix of the hexi polynomial code $C_H(7,3)$ is given by G;

$$G = \begin{pmatrix} A & 0 & F & C & 1 & 0 & 0 \\ 0 & A & 0 & F & C & 1 & 0 \\ 0 & 0 & A & 0 & F & C & 1 \end{pmatrix}.$$

The 3×3 invertible matrix S_H is given below:

$$S_H = \begin{pmatrix} C & F & 0 \\ 0 & 9 & 0 \\ 0 & 4 & F \end{pmatrix}.$$

The 7×7 permutation matrix P is as follows:

$$P = \begin{pmatrix} 0&0&0&0&0&1&0 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0 \\ 0&0&0&1&0&0&0 \\ 0&1&0&0&0&0&0 \end{pmatrix}.$$

The public key $G' = S_H \times G \times P$ is given by

$$G' = \begin{pmatrix} 5 & 0 & 4 & F & C & 1 & 8 \\ E & 0 & 6 & 9 & 5 & 0 & 0 \\ 9 & F & F & C & E & 0 & C \end{pmatrix}.$$

Encryption: Let the original message m which needs to be encrypted be given by $m = (1\ 0\ A)$, then compute $m \times G'$.

$$m' = m \times G' = (0\ C\ 8\ E\ A\ 1\ 9).$$

The encrypted message is given by m'. Let the random error be $e_r = (A\ B\ 0\ 4\ 7\ 1\ 0)$ and the selected error pattern be $e_p = (0\ 0\ 0\ 0\ 1\ 1\ 1)$. Then the error e will be $e = (0\ 0\ 0\ 0\ 7\ 3\ 0)$. This error e is added to the encrypted message m'.

$$y = (0\ C\ 8\ E\ D\ 2\ 9) = (0\ 0\ 0\ 0\ 7\ 3\ 0) \oplus (0\ C\ 8\ E\ A\ 1\ 9).$$

y is the final encrypted message. The message y is sent to the receiver side by the user.

Decryption: The receiver will get the received codeword $y_1 = (0\ C\ 8\ E\ D\ 2\ 9)$. The decryption of the received codeword is got by computing $y_1 \times P^{-1}$, where P^{-1} is the inverse of the permutation matrix P, P^{-1} is given below:

$$P^{-1} = \begin{pmatrix} 0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&1 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&0&1&0 \\ 0&1&0&0&0&0&0 \\ 1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0 \end{pmatrix}.$$

The resultant of multiplying y_1 with inverse of permutation matrix is given by

$$w = y_1 \times P^{-1} = (2\ D\ 9\ 0\ 8\ E\ C).$$

The intermediate message w can be expressed as a hexi polynomial, $w(x) = Cx^6 + Ex^5 + 8x^4 + 9x^2 + Dx + 2$. The generator hexi polynomial is $g(x) = x^4 + Cx^3 + Fx^2 + A$ and the parity check hexi polynomial is $h(x) = x^3 + Cx^2 + 8$.

Using Algorithm 1, the decoding and error correcting of the intermediate message $w(x) = Cx^6 + Ex^5 + 8x^4 + 9x^2 + Dx + 2$ is done producing results $m_0 = C + x + Cx^2$ and $e = 7x + 3$. The message $m_0 = (C\ 1\ C)$ and the error $e = (0\ 0\ 0\ 0\ 0\ 7\ 3)$ are obtained. Since the error is added only to the parity bits, only polynomial division is enough to obtain the errors.

The inverse of the invertible matrix S_H is S_H^{-1} given below;

$$S_H^{-1} = \begin{pmatrix} A & B & 0 \\ 0 & 2 & 0 \\ 0 & C & 8 \end{pmatrix}.$$

The original message m is obtained by multiplying the intermediate message m_0 with the inverse of the invertible matrix

$$m = m_0 \times S_H^{-1} = (1\ 0\ A);$$

m is the original message obtained after decryption.

5.2 Hexi GPT Cryptosystem

Based on the GPT public key cryptosystem, a variant of the McEliece cryptosystem called as hexi GPT cryptosystem is created. The most secure variant of the GPT cryptosystem is taken so that it withstands several attacks.

The hexi GPT public key cryptosystem makes use of the hexi Maximum Rank Distance code for the generator matrix G in the public key G', and for S a proper choice of column scramblers over the extension field is taken as in [10]. Since hexi MRD codes are analogous to MRD codes, these are a special case of MRD codes where $q = 2^4$, these codes have the same error correcting capacity. This variant of the GPT public key cryptosystem can withstand both Gibson's and Overbeck's attacks.

5.3 Hexi Wild McEliece Cryptosystem

The hexi wild McEliece public key cryptosystem is based on Wild McEliece cryptosystem and uses the hexi wild Goppa code as the error correcting code. The use of hexi wild Goppa codes in the place of Goppa codes increases the error correcting capacity. The hexi wild McEliece cryptosystem is a special case of the wild McEliece cryptosystem where the field F_{2^4} is taken. This case has been analysed as in [26]. Hexi wild McEliece public key cryptosystem is as secure as the original McEliece public key cryptosystem and also is as secure as the wild McEliece public key cryptosystem.

6 Attacks on the Hexi McEliece Cryptosystem

The two main types of attacks on any code based cryptosystem are structural and decoding attacks. The structural attack exploits the structure of the underlying code, and usually they attempt to recover the secret key. The later can be used independently of the code structure and are thus called as generic attacks.

The possible attacks on the hexi McEliece public key cryptosystem and security of the system are analysed. Since the hexi variants of the GPT public key cryptosystem and wild McEliece public key cryptosystem are similar to their counterparts, their security and attacks have not been analysed in detail. The hexi McEliece cryptosystem is not dependent on Goppa codes, so many of the attacks carried out on the original McEliece cryptosystem due to the structure of Goppa codes might not be successful on the hexi McEliece cryptosystem which is based on hexi polynomial codes.

6.1 Structural attack

A structural attack consists of attempts to reconstruct a decoder for the code generated by the public key G'. If such an attempt is successful, then the private key, the generating matrix G is revealed and the cryptosystem is broken. In the past, most structural attacks against code-based cryptosystems have targeted specific classes of codes. The code structure was exploited in order to break cryptosystems which use these codes. Examples of structural attack include the Sidelnikov-Shestakov attack against the Niederreiter public key cryptosystem using Generalized Reed-Solomon codes [28] and Overbecks attack against rank-metric codes [23,24]. Here to break the cryptosystem, one must find the generator polynomial $g(x)$, given the message, one must try atleast $m!$ guesses. This is also not feasible for larger m. When m is as small as ten, nearly 3628800 guesses must be tried.

Algebraic attack
Since a large public key size is one of the drawbacks of code-based cryptography, there have been many proposals attempting to reduce the key size. Often, the authors used highly structured codes which can be stored more efficiently. Examples of highly structured codes include quasi cyclic and quasi dyadic codes, as well as Low Density Parity Check (LDPC) codes. Recently, there have been several attempts using structural attacks against such highly structured codes. Algebraic attacks cannot be carried out on this public key cryptosystem, since the hexi polynomial codes that are used are not cyclic or quasi cyclic or quasi dyadic and these codes do not have low degree algebraic equation for code support.

Gibson's and Overbeck's attack
The hexi GPT cryptosystem is based on the variant of GPT cryptosystem defined in [10]. This hexi MRD code based variant of the GPT cryptosystem can withstand both Gibson's and Overbeck's attacks in case a proper choice of column scramblers over the extension field is taken as in [10].

6.2 Decoding attack

A decoding attack consists of decoding the intercepted ciphertext. Information Set Decoding (ISD) and the Generalized Birthday Algorithm (GBA) are the two most important types of generic attacks against code-based cryptosystems.

Information set decoding
The information set decoding attack is a top threat against the original McEliece cryptosystem, in a generic decoding method. Information set decoding depends on syndrome decoding and systematic form of the generator matrix G to break the cryptosystem. But this newly introduced hexi McEliece cryptosystem does not depend on syndrome decoding. The generator matrix G of the hexi polynomial code that is used in this cryptosystem is not given in its systematic form, hence information set decoding attack cannot be easily carried out on the cryptosystem.

Generalized birthday algorithm
The generalised birthday algorithm is not as efficient as the information set decoding attack on code based cryptosystems. The CFS signature scheme [3] was attacked using this method. The method makes use of a very large lists. For a sufficiently large n, this cryptosystem is secure.

7 Comparison with existing system

The hexi McEliece cryptosystem is a variant of the McEliece Public Key Cryptosystem (McEliece PKC) that is based on hexi polynomial codes. The hexi polynomial codes used in the hexi McEliece cryptosystem has better error correcting capacity, then the originally used binary Goppa codes. It has an error correcting capacity of nearly $n-k$ (or m) errors. The time complexity for decoding of hexi polynomial codes is smaller than that of decoding Goppa codes. The time complexity for polynomial division which is the major part in decoding of hexi polynomial codes, is $O(m\ lg\ m)$ (where m is the highest degree of the generator polynomial $g(x)$) [15], whereas the decoding of Goppa codes takes polynomial time using Patterson's decoding algorithm [25].

In the example of McEliece cryptosystem provided by McEliece in [19], the size of the public key G' is 1024×524, the length of codeword n is 1024, the message length k is 524 binary elements and the error correcting capacity t is 50. For almost the same parameters the Hexi McEliece cryptosystem will have the size of the public key G' to be 1024×512, the length of codeword n is 512 and the message length k is 512 hexi symbols and the error correcting capacity t is 512, which is nearly ten times the error correcting capacity of the McEliece cryptosystem. Since the McEliece cryptosystem functions on binary Goppa codes whereas the hexi McEliece cryptosystem is based on hexi polynomial codes, the basic parameters (n,k) are decreased by 4, so that the storage size of G' is almost same. Then the public key G' is 256×128, the length of codeword n is 256, the message length k is 128 and the error correcting capacity t is 128. The error correcting capacity is still better. A comparison table is given in the following:

Table 4: Comparison table of the cryptosystems

Comparison	McEliece PKC	Hexi McEliece PKC	Hexi McEliece reduced
Error Correction(t)	50	512	128
Matrix Size (G')	1024×524	1024×512	256×128
Codeword (n)	1024	1024	256
Msg Symbols (k)	524	512	128
Time Complexity	Poly	m lg m	m lg m

It is clearly seen that the hexi McEliece cryptosystem has better error correcting capacity and less time complexity than the original McEliece cryptosystem. The hexi GPT cryptosystem is as secure as the variant of the GPT cryptosystem constructed in [10], since it is a special case of the same defined over the field F_{2^4}. The hexi wild McEliece cryptosystem is also a special case of wild McEliece cryptosystem and therefore it is secure.

8 Conclusion and future direction

In this paper hexi based variants of the McEliece public key cryptosystem are introduced. Several new codes like hexi polynomial codes, hexi RD codes, hexi MRD codes and hexi Goppa codes are defined in this paper. These codes were used to create different variants of the McEliece public key cryptosystem. The time complexity of decoding hexi polynomial code is $O(m\ lg\ m)$ in comparison with the polynomial time taken for decoding of the Goppa code. Based on this hexi polynomial code, the hexi McEliece public key cryptosystem was created in this paper. The security of the system was analysed and it is found that the system is almost as secure as the other variants of McEliece cryptosystem. The feasible attacks on this system was also analysed and the system is not vulnerable to several structural and decoding attacks.

This new hexi McEliece public key cryptosystem has a better error correcting capacity and a less time complexity than the existing McEliece public key cryptosystem.

Future direction

Creation of identification and signature scheme based on this hexi McEliece public key cryptosystem is under consideration.

References

[1] D.J. Bernstein, T. Lange and C. Peters, Wild McEliece, Selected Areas in Cryptography 17th International Workshop, SAC 2010, Canada, Aug 12-13, 2010, Revised Selected Papers, LNCS, **6544**, 143-158 (2010).

[2] D.J. Bernstein, T. Lange and C.Peters, Wild McEliece Incognito, PQCrypto 2011, Taiwan, Nov 29-Dec 02, 2011, Proceedings. LNCS, **7071**, 244-254 (2011).

[3] N. Courtois, M. Finiasz and N. Sendrier, How to achieve a McEliece based digital signature scheme, Advances in Cryptology - ASIACRYPT 2001, Springer Verlag, **2248**, 157-174 (2001).

[4] E.M. Gabidulin, Theory of Codes with Maximum Rank Distance, Problems of Information Transmission, **21**, 112 (1985).

[5] E.M. Gabidulin, A.V. Paramonov and O.V. Tretjakov, Ideals over a non commutative ring and their application in cryptography, Advances in cryptology - Eurocrypt'91, Editor D.W.Davies, LNCS, **547**, 482-489 (1991).

[6] E.M. Gabidulin, Public Key Cryptosystem based on Linear codes over Large Alphabets: Efficiency and Weakness, Codes and Ciphers, **11**, 17-32 (1995).

[7] E.M. Gabidulin and A.V. Ourivski, Improved GPT Public Key Cryptosystems, Coding, Communications, and Broadcasting, Research Studies Press, 73-102 (2000).

[8] E.M. Gabidulin, A.V. Ourivski, B. Honary and B. Ammar, Reducible Rank Codes and Their Applications to Cryptography, IEEE Transactions on Information Theory, **49**, 3289-3293 (2003).

[9] E. M. Gabidulin, Attacks and counter-attacks on the GPT public key cryptosystem, Designs, Codes and Cryptography, **48**, 171-177 (2008).

[10] E. M. Gabidulin, H. Rashwan and B.Honary, On improving security of GPT cryptosystems, Proceedings of the 2009 IEEE international conference on Symposium on Information Theory, **2**, 1110-1114 (2009).

[11] J.K. Gibson, Severely denting the Gabidulin version of the McEliece public key cryptosystem, Designs, Codes and Cryptography, **6**, 37-45 (1995).

[12] J.K. Gibson, The security of the Gabidulin public key cryptosystem, Advances in Cryptology Eurocrypt96, LNCS, **1070**, 212-223 (1996).

[13] V.D. Goppa, A new class of linear error correcting codes, Problemy Peredachi Informatsii, **6**, 24-30 (1970).

[14] K. Ilanthenral and K.S. Easwarakumar, Design of Hexi cipher for error correction–using Quasi Cyclic Partial hexi codes, Journal of Applied Mathematics and Information Sciences, **7**, 2063-2071 (2013).

[15] D.E. Knuth, Seminumerical Algorithms, Vol. **2** of The Art of Computer Programming, Addison-Wesley, (1998).

[16] Y.X. Li, R.H. Deng, and X.M. Wang, The equivalence of McElieces and Niederreiters public key crptosystems, IEEE Transaction on Information Theory, **40**, 271-273 (1994).

[17] R. Lidl and G. Pliz, Applied Abstract Algebra, Springer Verlag, (1984).

[18] S. Lin and D.J. Costello, Error Control Coding, Pearson, (2005).

[19] R.J. McEliece, A public-key cryptosystem based on algebraic coding theory, Jet Propulsion Laboratory DSN Progress Report, **4244**, 114-116 (1978).

[20] T.K. Moon, Error Correction Coding- Mathematical Methods and Algorithms, Wiley, (2005).

[21] H. Niederreiter, Knapsack-type cryptosystems and algebraic coding theory, Problems of Control and Information Theory, **15**, 159-166 (1986).

[22] A.V. Ourivski and E.M. Gabidulin, Column Scrambler for the GPT Cryptosystem, Discrete Applied Mathematics, **128**, 207-221 (2003).

[23] R. Overbeck, A new brute-force attack for GPT and variants, Proc. of Mycrypt2005, LNCS, **3517**, 5-63 (2005).

[24] R. Overbeck, Brute-force attacks Public Key Cryptosystem Based on Gabidulin codes, J. Cryptology, **21**, 280-301 (2008).

[25] N.J. Patterson, The algebraic decoding of Goppa codes, IEEE Transactions on Information Theory, **21**, 203-207 (1975).

[26] C. Peters. Curves, Codes, and Cryptography, Ph.D. diss, Technische Universiteit Eindhoven, (2011).

[27] R.C.Phan, Mini Advanced Encryption Standard (Mini AES): A Testbed for Cryptanalysis students, Cryptologia, **26**, 283-306 (2002).

[28] V.M. Sidelnikov and S.O. Shestakov, On insecurity of Cryptosystems based on generalized Reed Solomon codes, Discrete Mathematics and Applications, **2**, 439-444 (1992).

[29] J. Stern, Can one design a signature scheme based on error correcting codes?, ASIACRYPT94, LNCS, **917**, 424-426 (1995).

Ilanthenral Kandasamy is currently pursuing her Ph.D under the guidance of Prof. K. S. Easwarakumar at Department of Computer Science and Engineering at Anna University, Chennai. She received her B.Sc. in Mathematics from Madras University, Chennai and her Masters in Computer Science and Applications from Anna University, Chennai. Her research interests include Cryptography and Coding Theory.

K. S. Easwarakumar is a Professor at the Department of Computer Science and Engineering at Anna University, Chennai. He received his M.Tech in Computer and Information Sciences from Cochin University of Science and Technology, Cochin and Ph.D in Computer Science and Engineering from Indian Institute of Technology, Madras. His research interests include parallel and distributed computing, Data Structures and Algorithms, Graph Algorithms, Parallel Algorithms, Computational Geometry, Theoretical Computer Science and Molecular computing.