Ciphering Using Error-Correcting Codes

Mohammed S. EL-Atrash*
Fayik R. EL-Naowk**

الملخص

في هذا البحث سوف نقلد نظام المفتاح العام للشيفره للباحث ماك إليص وذلك لعمل مفتاح الشيفره الخاص المبني علي الترااميز مصححات الأخطاء. وعلي وجه الدقة سوف نستخدم تتابع من الترااميز $C_1, C_2, \ldots, C_r$ التي قدرتها التصحيحية $t_1, t_2, \ldots, t_r$ وأطوالها $n_1, n_2, \ldots, n_r$ ومصفوفاتها المولدة $G_1, G_2, \ldots, G_r$ التي رتبها $m\times n_1, n_1\times n_2, \ldots, n_{r-1}\times n_r$ علي الترتيب.

إذا كانت الرسالة P عبارة عن متجه ثنائي طوله m وبفرض أن $e_1, e_2, \ldots, e_r$ تتابع من الأخطاء والتي أطوالها $n_1, n_2, \ldots, n_r$ وأوزانها أقل أو تساوي $t_1, t_2, \ldots, t_r$ علي الترتيب ، فإن الشيفرة ستكون

$$C = (\ldots((P \times G_1 + e_1)G_2 + e_2)\ldots)G_r + e_r.$$

ABSTRACT

In this paper, we mimic the McEliece public key cryptosystem to build up our private key cryptosystem based on error-correcting codes. To be specific, we use a sequence of several different codes $C_1, C_2, \ldots, C_r$ that are $t_1, t_2, \ldots, t_r$ - error correcting codes of lengths $n_1, n_2, \ldots, n_r$, whose generating matrices are $G_1, G_2, \ldots, G_r$ of orders $m\times n_1, n_1\times n_2, \ldots, n_{r-1}\times n_r$, respectively. Let the message P be a binary vector of length m, and if we let $e_1, e_2, \ldots, e_r$ be a sequence of r errors of lengths $n_1, n_2, \ldots, n_r$ respectively, and of weights less than or equal $t_1, t_2, \ldots, t_r$, respectively. Then the ciphertext is

$$C = (\ldots((P \times G_1 + e_1)G_2 + e_2)\ldots)G_r + e_r.$$

Key words: Error correcting codes, ciphering, cryptography, McEliece cryptosystem.

* **Assoc. Prof. of Mathematics, department of mathematics, Islamic University-Gaza, Email: matrash@mail.iugaza.edu.**

** **Lecture Asst. of mathematics, department of mathematics, Al-Aqsa University-Gaza, Email: fayikrm@hotmail.com**

1. Introduction

Coding theory is now generally regarded as a mature science. However, cryptography on the other hand, is a science in the phase of early and rapid development. Coding theory has been used in cryptography by McEliece who has used Reed-Solomon codes in secret sharing [8], and in building up a public key cryptosystem [7]. This cryptosystem is based on error-correcting codes. The idea behind this scheme is to first select a particular code for which an efficient decoding algorithm is known. A description of the original code can serve as the private key, ehile a description of the transformed code serves as the public key.

Sidelnikov and Shestakov had constructed some cryptosystems based on generalized Reed-Solomon codes [9]. and by Massey who has used minimal codewords in secret sharing [5]. He also used codes and orthogonal arrays in local randomization [6].

The purpose of this paper is to use error-correcting codes to build up a symmetric private key cryptosystem.

This paper is organized as follows: Section 1 surveys the literature for the use of error correcting codes in cryptosystems. Sections 2 and 3 expose the elementary definitions and results of both coding theory and cryptography respectively in order to set up the notation being used through out this paper. Section 4 explains the main results of how to build up the cryptosystem. Section 5 studies the security level and complexity of the cryptosystem. Section 6 contains some concluding remarks.

2. *Basic Definitions in Coding Theory* [1, 2]

A *word* is a sequence of digits. *The length* of a word is the number of digits in the word.

Let $\beta = \{a_1, \ldots, a_q\}$ be a finite set called the alphabet, and let β^n be the set of all strings of length n over β. Any nonempty subset C of β^n is called *q-ary block code*. Each string in C is called a *codeword*. If $C \subset \beta^n$ contains M codewords then C has length n and size M and is denoted by (n, M)-code. A code whose $\beta = \{0, 1\}$ is called *binary code*. Commonly, alphabets have a group, a field or even a ring structures.

If v is a word in C that is sent and w in β^n is received, then $u = v + w$ is the *error pattern*, or the error.

Let a be a q-ary word of length n over an alphabet that has a group structure (that contains zero). The Hamming weight, or simply the weight of a is the number of non-zero components in a. We denote the weight of a by $w(a)$. The minimum weight of a code C is the minimum weight of all non-zero codewords in C and we denote it by $w(C)$.

Let x and y be two words of the same length. The Hamming distance or simply the distance between x and y is the number of positions in which x and y differ.

We denote this distance between x and y by $d(x, y)$. A code C is said to have minimum distance d if

$d = \min \{d(x, y): x, y \in C, x \neq y\}$ and we denote it by $d(C)$.

An (n, M, d)-code is a code of length n and size M and minimum distance d.

A code C is called a linear code if C is a vector subspace of a vector space $V(n, q)$ of dimension n over the field $GF(q)$. If C has dimension k over $GF(q)$, we say that C is an $[n, k]$-code, and if C has minimum distance d we say that C is an $[n, k, d]$-code.

We say that a code C corrects an error pattern u if, for all v in C; $u + v$ is closer to v than to any other word in C.

A code C is called a t-error-correcting code if C corrects all error patterns of weights at most t and does not correct at least one error pattern of weight $t+1$.

It can be easily verified that a linear $[n, k, d]$ code C corrects all error patterns e of t where $t = \left\lfloor \frac{d-1}{2} \right\rfloor$. We denote such a code by (n, k)-*t-error correcting code.*

Let C be a linear $[n, k]$-code, a $k \times n$ matrix G whose rows form a basis for C is called a *generator matrix* for C. If C is an $[n, k]$-code with generator matrix G, then the codewords in C are the linear combinations of the rows of G, i.e., $C = \{x\, G: x \in V(k, q)\}$.

3. *Basic Definitions and Notation in Cryptography* [3, 4]

A plain message is called *a plaintext.* The process of disguising a message in such a way as to hide its substance is called *encryption.* An encrypted message is called *ciphertext.* The process of turning ciphertext back into plaintext is called *decryption.* Plaintext is denoted by P. Ciphertext is denoted by C. The encryption function E operates on P to produce C. This means that E(P) = C. In the reverse process, the decryption function D operates on C to produce P; D(C) = P. Since the

whole point of encryption and then decryption of a message is to recover the original plaintext, the following identity must hold: D(E(P)) = P.

A key, denoted by k, is any thing that we keep secret such that without knowing it the decryption is infeasible process. *A cryptosystem* is the system consists of plaintext, ciphertext and the keys (encryption and decryption).

A symmetric key cryptosystem is a cryptosystem in which the decryption key can be calculated from the encryption key and vice versa. In most symmetric cryptosystems the encryption and decryption keys are the same.

Public key cryptosystem is a system in which the encryption key (*called public key*) is different from the decryption key and it is made public. Decryption key can not be calculated from the encryption key. The decryption key is often called the *private key*. A protocol is a set of instructions telling the sender and receiver what to do.

4. *The Main Results*

Although this system works in any finite field, we will concentrate only on the binary field.

Let C_i be a t_i - error correcting codes of lengths n_i, whose generating matrices are G_i of orders $n_{i-1} \times n_i$ respectively, where $i = 1, 2, \ldots r$.

Let the plaintext P be a binary vector of length n_0. Let e_i be a binary vector of n_i, and of weight t_i, where $i = 1, 2, \ldots r$.

We construct the following sequence of binary vectors

$$C_1 = P \times G_1 + e_1$$

$$C_i = C_{i-1} \times G_i + e_i, \text{ where } i = 1, 2, \ldots r.$$

We set the ciphertext $C = C_r$, and $C_0 = P$.

To decrypt the ciphertext C we first correct the error e_i using the error correcting code C_i to get $C_{i-1} \times G_i$, and then we use any method (Kramer rule, or variable elimination) to solve the linear system of equations:

$$b_{i-1} = C_{i-1} \times G_{i-1}$$

to get C_{i-1}, in the reversed order $i = r, r-1, \ldots, 1$.

Clearly, this generates a symmetric private key cryptosystem, for which the private key is the sequence of the codes C_i (i.e., the G_i) must be kept secret for all $i = 1, 2, \ldots, r$.

Example. Let C_1 be the linear code [7, 4, 3] whose generating matrix is

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Let C_2 be the linear code [15, 7, 5] whose generating matrix is

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

For instance, to encrypt the message

$P = 1010$

We multiply $P \times G_1$ to get $P \times G_1 = 1010010$.

We make one random error say $e_1 = 0000001$ then

$C_1 = 1010011$

We multiply C_1 by G_2 to get:

$C_1 \times G_2 = 101011110001001$.

Add 2-error randomly say $e_2 = 100001000000000$ to get the ciphertext

$C = C_1 \times G_2 + e_2 = 001010110001001$.

For decryption we reverse the process of encryption

This can be explained as follows:

First we know that $C = 001010110001001$ has 2 errors. By using the code C_2 we see that the codeword closest to C is

$w_1 = 101011110001001$.

Then solving the linear system $C_1 \times G_2 = w_1$ we can get $C_1 = 1010111$.

Now we need to eliminate 1-error using the code C_1, to get

$w_2 = P \times G_1 = 1010010$.

Now to find the message P we need to solve the linear system

$P \times G_1 = w_2$.

As the first three columns and the fifth column represent the 4×4 identity matrix then P is the first three entries of w_2 and the fifth entry, so, $P = 1010$.

5. *Security and Complexity of the Cryptosystem*

For the security level of this cryptosystem, we should say that it is not less than the security of McEliece public key cryptosystem, as we have here r different codes, in addition to the randomness of the r errors of weights t_i. We note that the number of errors possible for such system is $t_1 + t_2 + \ldots + t_r$ which can be large compared to the number of errors that can be made in the systems using one code, however, this is on the account of the message. To explain this we know that in our system of codes the message is n_0 bits and the ciphertext is n_r bits, while the number of errors is 3. We may use a one 3-error correcting code with dimension higher than n_0 and the same length n_r. This would imply that we can encipher a longer message to get the number of bits in the ciphertext.

As far as attacks on this system, we should say that for plaintext-ciphertext attack on such a system we need $n_0 \times n_1 + n_1 \times n_2 + \ldots + n_{r-1} \times n_r$ different plaintext-ciphertext messages in order to find all the coefficients in the r generating matrices considering all possible errors. The number of errors amounts to $\binom{n_1}{t_1} + \binom{n_2}{t_2} + \ldots + \binom{n_r}{t_r}$. Clearly such numbers are huge when we take n_i large numbers.

As for Complexity, we may use fixed permutation matrices T_i, and nonsingular matrices S_i in the following way:

$C_i = (C_{i-1} \times S_i \times G_i + e_i)T_i$, where $i = 1, 2, \ldots r$.

Example. Let C_1, C_2 be the linear codes used in the last example. Let T_1 be the permutation matrix

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

and T_2 be the permutation matrix

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Let S_1 be the nonsingular matrix

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

S_2 be the permutation matrix

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

For instance, to encrypt the message

$P = 1010$

We multiply $P \times S_1 = 0101$, then $P \times S_1 \times G_1 = 0101101$ We make one random error say $e_1 = 0000001$ then

$P \times S_1 \times G_1 + e_1 = 0101100$

We multiply $P \times S_1 \times G_1 + e_1$ by T_1 to get:

$C_1 = (P \times G_1 \times S_1 + e_1) \times T_1 = 1001100$

$C_1 \times S_2 = 0100110.$

$C_1 \times S_2 \times G_2 = 0100100100100 10$

Add 2-error randomly say $e_2 = 100001000000000$ to get the ciphertext

$C_1 \times S_2 \times G_2 + e_2 = 110011010010010$.

$C_2 = (C_1 \times S_2 \times G_2 + e_2) \times T_2 = 110011010010010$.

Thus C = 110011010010010.

For decryption we reverse the process of encryption

This can be explained as follows:

1. Multiply by T_2^{-1}
2. correct the error e_2 by using C_2
3. Solve the equation $C_1 \times S_2 \times G_2 = w_2$ to get $C_1 \times S_2$
4. Multiply by S_2^{-1} to get C_1.
5. Multiply by T_1^{-1}
6. correct the error e_1 by using C_1
7. Solve the equation $P \times S_1 \times G_1 = w_1$ to get $P \times S_1$
8. Multiply by S_1^{-1} to get P.

6. Conclusions

We have built up a cryptosystem based on error correcting codes similar to McEliece public key cryptosytem. It is more secure and more complicated. It is constructed by using a sequence of error correcting codes.

REFERENCES

Books

[1] Bierbrauer, J., Introduction to Codes and their use, MA 576, (1999).

[2] Roman, S., Coding and Information Theory, Springer-Verlag, (1992).

[3] Menezes, A. and Vanstone, S., Handbook of applied Cryptography, CRC Press (1996).

[4] Schneier, B., Applied Cryptography. Second edition, John Wiley and Sons, Inc., (1996).

Published Papers

[5] Massey, J.L. "*Minimal Codewords and Secret Sharing*", Proceedings 6th Joint Swedish-Russian International Workshop on Information Theory, pp. 276-279.

[6] Massey, J.L. "*Some applications of coding theory in Cryptography Codes and Cyphers*", Cryptography and Coding IV. Essex. England: Formare Ltd. 1995 pp. 33-47.

[7] McEliece, R.J. "*A public key Cryptosystem based on algebraic coding theory*" DSN Progress report 42-44 pp. 114-116. (1978)

[8] McEliece, R.J. and Sarwate, D.V. On Sharing Secrets and Reed-Solomon Codes. Communications of the ACM, 24, 583-584, (1981).

[9] Sidelnikov, V.M. and Shestakov, S.O. On cryptosystems based on generalized Reed-Solomocodes. Diskretnaya math, 4: 57-63, 1992, in Russian.