Engineering Journal of University of Qatar, Vol. 8, 1995, p. 65 - 81

CASCADED CODING SCHEMES FOR PUBLIC-KEY CRYPTOGRAPHY

Magdi M. Said El-Soudani
Electrical Engineering Department, College of Engineering,
University of Qatar, Doha, Qatar.

ABSTRACT

1976 Diffie and Hellman introduced the concept of public-key cryptography and in 1978, McEliece introduced the first public-key cryptosystem based on error correcting codes. Since that time, several methods have been proposed to use error correcting codes for cryptography either directly or indirectly. In this work we propose the use of cascaded codes in McEliece algorithm where cascading here means that one code is used after the other. Two or more codes are used in cascade to get high error correcting capabilities even with moderate length codes. This makes the system more useful over noisy channels. The structure of cascaded codes is in itself a good way to secure the data. Binary block codes are only considered in this work although other types of codes can be used. We discuss two different encryption schemes where normal and Tensor products of matrices are used to form the codes. The proposed schemes are more adequate for block encryption. Decoding is also performed in cascade to make use of the existing fast decoding algorithms available for each of the used codes. Therefore, the decryption process will be fast too compared with other schemes based on number theory. The selection of proper code parameters is discussed and the probability of correct recovery of transmitted messages is also found.

INTRODUCTION

We usually use error correcting codes in digital communication to detect and to correct transmission errors. Such codes are publicly known and normally they do not offer cryptographic protection. If the plaintext is encoded using an error correcting code, this will be considered as a further authentication of the transmitted message [1]. However, McEliece [2] proposed a public-key cryptosystem based on error correcting codes. Other works also suggest the use of error correcting codes for encryption and authentication such as [3] and [4]. In the present work, we propose the use of cascaded codes instead of Goppa codes

originally used in McEliece algorithm to provide both protection and security of information. This method is more adequate for block encryption and the code parameters can be selected to get lengths similar to those used in data encryption standard (DES) [5] for example. The paper is organized as follows; the following two sections give a brief review of the concepts of public-key cryptography and the properties of linear block codes which are only considered in this work although other codes can be used. Then, the details of McEliece public-key cryptosystem and the proposed two encryption schemes are covered in three sections. The code selection criteria and the system performance represent the third part of this paper. We conclude our work by comments on the advantages and disadvantages of the proposed schemes and their performance.

PUBLIC KEY ENCRYPTION

In public-key cryptosystem many people can encrypt messages in such a way that only one user can get them. Public-key cryptography makes secure communications possible even in communication networks with thousands of users [6,7]. This method is symmetric since it allows secret communication in both directions. Fig.1 shows a typical public key cryptosystem. Encryption employs an algorithm E and an encryption key K_e. Decryption process employs the algorithm D and the decryption key K_d. Both algorithms, E and D, are public. The decryption process is the inverse of the encryption process, so that the keys are related. They are derived from the same source (seed) K_s. Two public algorithms F_e and F_d are used to calculate the keys as shown in Fig.1. Only the intended receiver of the information should be able to decrypt it, and the decryption key K_d is kept secret by the receiver. The other key, K_e, is made public, enabling any one to encrypt data for one receiver to whom the key belongs. To keep the secret, the receiver must himself carry out the calculations F_e and F_d to create both keys, of which he keeps K_d strictly for his own use.

LINEAR BLOCK CODES

In this work we restrict ourselves to linear binary block codes. A linear block code is denoted generally by C(n,k,d) where n is the codeword length, k is the dimension of the code (or the number of information symbols), and d is its minimum distance. The codewords of a code C represent a subspace of the vector space of n-tuple over GF(2). All codewords are generated using a generator matrix **G** of dimension n * k. On the other hand, any n-tuple vector $\underline{c}$ is a codeword of the code generated by **G** if $\underline{c}\mathbf{H}^T = \underline{0}$; where **H** is an (n-k) * n matrix

called the parity check matrix and $\underline{0}$ is a zero vector of length n-k. called the parity check matrix and $\underline{0}$ is a zero vector of length n-k.

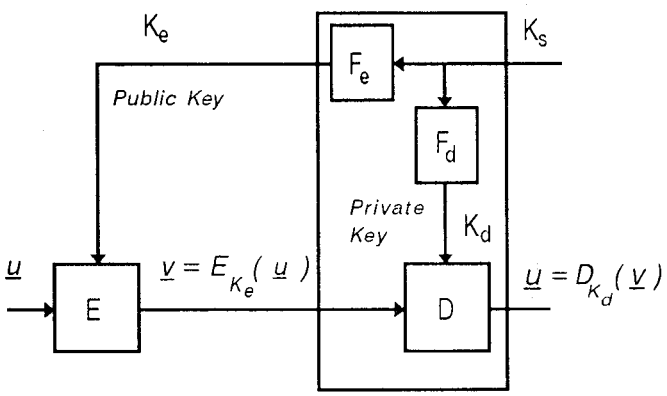

Fig.1: Public-key cryptography

As a result, $\mathbf{GH}^T = \mathbf{0}$, where **0** is zero matrix of dimension k * (n-k). The code is said to be systematic if the first k symbols of its codewords are the information symbols and the last n-k symbols are the redundant ones. This requires that the **G** matrix be in the form $\mathbf{G}=[\mathbf{I}_k|\mathbf{A}]$, where **A** is an arbitrary matrix of size k * (n-k) and $\mathbf{I}_k$ is the identity matrix of size k * k. Similarly, the parity check matrix **H** can be written in the form $\mathbf{H} = [\mathbf{A}^T|\mathbf{I}_{n-k}]$. The error correcting capability of the code C(n,k) is $t=\lfloor (d-1)/2 \rfloor$ where $\lfloor x \rfloor$ denotes the largest integer less or equal to x. However, a block code with random error correcting capability t can correct many error patterns of $t+1$ or more errors. Several decoding algorithms exist for linear block codes, some of them use the algebraic structure of the code, while others make use of the channel information to perform probabilistic decoding. More can be found about linear codes in [8,9].

MCELIECE PUBLIC-KEY ALGORITHM

Consider a binary message $\underline{m}$ of length k. If this message is encoded using the generator matrix **G** of the code C, the result is a codeword $\underline{c}$ of length n. In order to make it difficult for cryptanalyst to discover which code is being used, the matrix **G** is scrambled using a nonsingular matrix **S** of size k * k (this is equivalent

to combining the rows of **G** linearly), then the columns of the **G** matrix are rearranged using the permutation matrix **P** of size (n * n). Therefore, the generator matrix that the transmitter of the message actually uses is **S G P**, or **G'**. To encrypt a message $\underline{m}$, the transmitter encodes his data using **G'** and in addition he adds (modulo-2 addition) a certain pattern of error that only the receiver knows how to remove it through the use of a proper decoding algorithm. The added error pattern $\underline{e}$ is a locally generated random sequence of length n and weight $w_H(\underline{e}) \leq t$, with equality only for error-free channels. So in brief;

Private key : **G**, **P** and **S** .

Public key : **G'** and the code error correcting capability t.

Cryptogram : $\underline{v} = \underline{m}\,\mathbf{G'} + \underline{e}$.

Since the receiver knows the **S** and **P** matrices, the required vector $\underline{m}$ can be determined as shown bellow;

$$\underline{v}\,\mathbf{P}^{-1} = \underline{m}\,\mathbf{G'}\,\mathbf{P}^{-1} + \underline{e}\,\mathbf{P}^{-1}$$
$$= \underline{m}\,\mathbf{S}\,\mathbf{G} + \underline{e}'$$
$$= \underline{m}'\,\mathbf{G} + \underline{e}'$$

$\underline{m}'\,\mathbf{G}$ is a codeword of the code C and $\underline{e}'$ has the same weight as $\underline{e}$, assuming no more errors are introduced by the channel, so $\underline{e}'$ is still within the error correcting capability of the code C. Using an appropriate decoding algorithm, we can find the corresponding error pattern $\underline{e}'$, thence $\underline{m}$ can be found, where:

$$\underline{m} = \underline{m}'\,\mathbf{S}^{-1}$$

In the original algorithm, McEliece has used Goppa code because of its large value of t. Goppa code is a special case of alternant codes and can also be derived from BCH code [8]. McEliece algorithm is useful over error-free channels that is not the case of most real communication channels. To protect the information against channel error, the weight of the error vector deliberately added at the transmitter should be less than the error-correcting capability of the code. Fig.2 shows an additive white gaussian noise channel model assuming McEliece public-key cryptosystem is used. The errors introduced by the channel are independent and are denoted by the error vector $\underline{e}''$. The transmitted message $\underline{u}$ will be received correctly if $w_H(\underline{e}) + w_H(\underline{e}'') \leq t$.

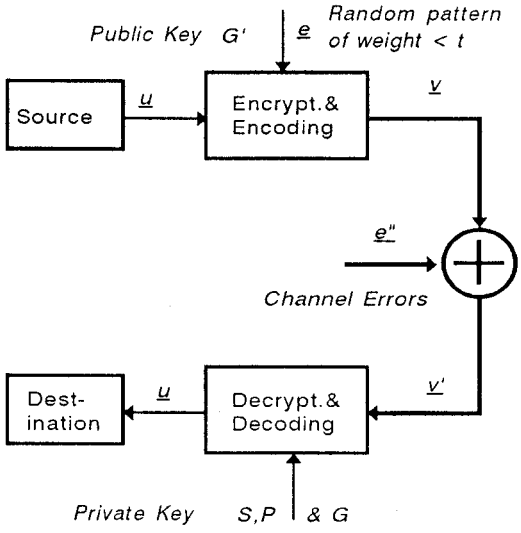

Fig.2: AWGN communication channel with public-key cryptosystem

ENCRYPTION USING NORMAL PRODUCT

In this scheme, two codes $C_1(n_1,k_1,d_1)$ and $C_2(n_2,k_2,d_2)$ are used one after the other such that the codewords of C_1 represents a subset of the information vectors of the second code C_2. This means that $k_2=n_1$. We refer to this code structure as the iterated code. The iterated code C_i has length n_2 and dimension k_1. The generator matrix of the iterated code will be the normal product of the two generator matrices, i.e., $\mathbf{G}_i = \mathbf{G}_1\mathbf{G}_2$. The minimum distance of the resultant code is still d_2. At the receiving end, the received block will be decoded first using the rules of C_2 followed by the rules of C_1. The decoding is successful if no errors are introduced by the channel or the originally added error vector $\underline{e}$ is of weight w_e that is less than t and the number of channel errors is less than t-w_e.

The cryptogram still have the same form as before;

$$\underline{v} = \underline{m}\ \mathbf{G}' + \underline{e}$$

where $\mathbf{G}' = \mathbf{S}\,\mathbf{G}_i\,\mathbf{P} = \mathbf{S}\,\mathbf{G}_1\mathbf{G}_2\,\mathbf{P}$

and in this case the scrambling matrix has dimension $k_1 * k_1$ and the permutation matrix will be of dimension $n_2 * n_2$ and $\underline{e}$ is a randomly generated vector of length n_2 and of weight $w_H(\underline{e}) \leq t_2$. Only $\mathbf{G}'$ is made public and the other

matrices are kept secret. At the receiving end these secret matrices together with decoding process are used to decrypt the received message. In brief;

Public key : **G'**, and error correcting capability t_2.

Private key : $\mathbf{G}_1$, $\mathbf{G}_2$, **S**, and **P**.

Cryptogram : $\underline{m}\ \mathbf{S}\,\mathbf{G}_1\mathbf{G}_2\,\mathbf{P} + \underline{e}$.

The decoding can be performed in two steps starting by decoding the n_2-bit block using decoding rules of C_2, then applying the decoding rules of C_1 on the resultant n_1-bit block. Due to the permutation process we assume that the remaining errors, either due to channel or due an inserted or altered text, affect the symbols independently. This scheme can be used either for data encryption or for authentication.

In order to find the possible number of keys; we have to consider all possible forms and combinations of scrambling, permutation, and generator matrices. This number is approximately;

$$N \approx 2^{k_i^2} |g_1| |g_2| n_2 ! \tag{1}$$

where $|g_i|$ is number of possible forms of generator matrix of the code C_i and is given by [4]:

$$|g_i| = \prod_{j=0}^{k_i-1} (2^{k_i} - 2^j) \tag{2}$$

This is mathematically true but from the coding point of view few combinations would represent real codes. As an example, consider the iterated code consists of the Hamming code (7,4,3) as the first code and the BCH code (15,7,5) as the second code. The resultant code C_i is an (15,4,5). The codewords of the this code represents a subset of the codewords of C_2. In this case the possible number of encoding rules for purpose of encryption is approximately 2.8×10^{35} for these two small codes.

In error-free communication channels, the error vector $\underline{e}$ used to encrypt the message can be identified after the decoding process. Therefore, if this error vector contains any information it can be recovered. We can make use of this error pattern in a different way. This can be achieved by using $\underline{e}$ as secondary message with weight less than t_2. Also we can add a secondary message by assuming the following structure of a cryptogram:

$$\underline{v} = (\underline{m}\,\mathbf{G}_1 + \underline{e}_1)\,\mathbf{S}_2\ \mathbf{G}_2\,\mathbf{P}_2 + \underline{e}_2 \tag{3}$$

where $\underline{m}_1$ represents the main or primary message, $\underline{e}_1$ is the secondary message and it is of length n_1 and $w_H(\underline{e}_1) \leq t_1$, and $\underline{e}_2$ is an error pattern of length n_2 and $w_H(\underline{e}_2) \leq t_2$. The error correcting capability of code C_1 should be large enough to make the secondary message of reasonable length. Using the same notation as before, we can write equation (3) as ;

$$\underline{v} = \underline{m}\mathbf{G}_1\mathbf{G}_2' + \underline{e}_1\mathbf{G}_2' + \underline{e}_2 \qquad (4)$$

where $\mathbf{G}_2' = \mathbf{S}_2\mathbf{G}_2\mathbf{P}_2$

This equation can be written in terms of primary and secondary keys as

$$\underline{v} = \underline{m}_1\mathbf{G}_p + \underline{e}_1\mathbf{G}_s + \underline{e}_2 \qquad (5)$$

where $\mathbf{G}_p = \mathbf{G}_1\mathbf{G}_s$ and $\mathbf{G}_s = \mathbf{G}_2'$

The addition of a secondary message requires the use of two different keys. At the receiving end, we start by applying the decoding rules of the second code C_2 in order to get the error pattern $\underline{e}_2$, then the decoding rules of the first code C_1 to get the primary message $\underline{m}_1$ and the secondary information $\underline{e}_1$. The secondary message is protected using single code C_2 while the primary message is protected using the two codes. The public key in this case consists of the matrices $\mathbf{G}_p$ and $\mathbf{G}_s$ together with t_1 and t_2. This will increase the probability for the cryptanalyst to find the original codes C_1 and C_2. Table-I gives examples of possible code combinations that can be used in this scheme. The parameters n,k, and d describe the overall resultant code, and R is the overall code rate. In order to satisfy the condition $k_2=n_1$, we get some new codes from known ones using the expurgating method [8]. Single parity check code is a good choice to get higher data rate. Only one of the two codes used in this case can be a single parity check code since its minimum distance is 2.

Table1: Examples of Codes for First Scheme

n_1	k_1	t_1	n_2	k_2	t_2	n	k	t	R
4	3	0	7	4	1	7	3	1	0.428
7	4	1	15	7	2	15	4	2	0.267
63	57	1	127	63	3	127	57	10	0.449
127	106	2	255	127	4	255	106	22	0.416

ENCRYPTION USING TENSOR PRODUCT

In this case we use two codes as before but without restrictions on their lengths or dimensions. Permutation is also performed in this case. The plaintext is divided into segments each of k_1 symbols or bits. Then, each segment is encoded using the first code C_1. k_2 codewords of the first code are stored, and they are divided into k_1 k_2-bit segments such that each segment does not contain more than one bit from each codeword of the previous stage. This similar to applying two decoding rules on a block of data of k_1 rows and k_2 columns. The resultant coded text is of length n_1n_2 , dimension k_1k_2, and minimum distance d_1d_2 and that is why this structure is called product code. As before, the cryptogram will be in the form:

$$\underline{v} = \underline{m}\,\mathbf{G}' + \underline{e}$$

and $$\mathbf{G}' = \mathbf{S}(\mathbf{G}_1 \otimes \mathbf{G}_2)\mathbf{P} \qquad (6)$$

where $\otimes$ indicates the Tensor (Kronecker) product of matrices [10], **S** and **P** are the same as before but with dimension $k_1k_2 * k_1k_2$ and $n_1n_2 * n_1n_2$ receptively. The plaintext or the information vector $\underline{m}$ is of length k_1k_2. The random error pattern $\underline{e}$ is of length n_1n_2 and weight $w_H(\underline{e}) \leq t$, where t is the error correcting capability of the resultant code. So in brief;

Public key : **G'**, and error correcting capability t_2.

Private key : $\mathbf{G}_1$, $\mathbf{G}_2$, **S**, and **P**.

Cryptogram : $\underline{m}\ \mathbf{S}\ (\mathbf{G}_1 \otimes \mathbf{G}_2)\ \mathbf{P} + \underline{e}$.

We can perform scrambling and permutation on each code separately in order to use smaller matrices. We can write the publicly known matrix in this case as:

$$\mathbf{G}' = \mathbf{S}_1\mathbf{G}_1\mathbf{P}_1 \otimes \mathbf{S}_2\mathbf{G}_2\mathbf{P}_2 \qquad (7)$$

From the properties of Tensor product , the above expression can be written in the form:

$$\mathbf{G}' = (\mathbf{S}_1 \otimes \mathbf{S}_2)\,(\mathbf{G}_1 \otimes \mathbf{G}_2)\,(\mathbf{P}_1 \otimes \mathbf{P}_2) \qquad (8)$$

In this way, we can use smaller scrambling and permutation matrices to get the same result as before. In this scheme, the information symbols are arranged in blocks of dimension $k_2 * k_1$. Let $\underline{x}_i$ represent the row information vectors of length k_1 for $i=1,2,..,k_2$. The information symbols are then rearranged in the form of vectors $\underline{y}_j$ of length k_2 such that $\underline{y}_j=[x_{ij}]$ where x_{ij} is the jth element of the vector $\underline{x}_i$, $i=1,..,k_2$ and $j=1,2...,k_1$. Assume that the scrambling matrix **S** is the unity matrix, the resultant codeword before column permutation will be:

$$\underline{z} = \left[\sum_{\oplus, i=1}^{k_1} g_{i1} \underline{y_i} G_2 \middle| \sum_{\oplus, i=1}^{k_1} g_{i2} \underline{y_i} G_2 \middle| \cdots\cdots\cdots \middle| \sum_{\oplus, i=1}^{k_1} g_{in_1} \underline{y_i} G_2 \right] \qquad (9)$$

where $\sum_{\oplus} \underline{x}$ indicates vector addition modulo-2, g_{ij}, i=1,2,..,k_1, j=1,2,..,n_1, is the (i,j) entry of the $\mathbf{G}_1$ matrix. This shows how the information symbols are arranged in the decoding process. This arrangement is a sort of interleaving and no two symbols of the same vector $\underline{x}_i$ exist in the same vector $\underline{y}_j$. Other interleaving methods can also be used.

As in the case of iterated code, the decoding can be performed in steps making use of the decoding procedure of each code separately. The **S** and **P** matrices together with the generator matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ are kept secret. Table-II gives examples of some codes which can be used in this scheme.

Table 2: Examples of Codes for Second Scheme

n_1	k_1	t_1	n_2	k_2	t_2	n	k	t	R
4	3	0	7	4	1	28	12	2	0.428
7	4	1	15	7	2	105	28	7	0.267
15	7	2	31	16	3	465	112	17	0.241
15	11	1	31	21	2	465	231	7	0.496

The choice of codes depends on several factors such as the required data rate, channel error rate, and the existence of proper decoding algorithms for each of the selected codes. In the next section we will discuss some of these factors.

CODE SELECTION

We gave examples of possible combinations of codes that can be used in each scheme. A proper selection of codes is the one that ensures the security of the system. McEliece investigated several attacks against the original system [11]. One of this attack is based on selecting k error free elements of the cryptogram $\underline{v}$ and to solve a set of k linear equations to recover the message. If k components of the error vector are zeros and the cryptanalyst succeeded in finding these

components, he can recover the information symbols. Since the number of non-zero elements in $\underline{e}$ is at maximum t (the equivalent error correcting capability of the code or combination of codes used) the probability of selecting k zero elements from $\underline{e}$ is

$$p_k = \frac{\binom{n-t}{k}}{\binom{n}{k}}$$

$$p_k = \prod_{i=0}^{t-1}\left(1 - \frac{k}{n-i}\right) \tag{10}$$

The above equation shows that in order to have a secure system the code rate k/n must be as large as possible ($0 < k/n < 1$) or the error-correcting capability of the code would be large enough to assure low value for p_k. This shows the tradeoff between the code error correcting capability and its rate. The above selection process requires the check of at least N_t ($2^n - 2^{n-k}$) n-bit vectors where N_t is the number of n-bit error pattern with t or fewer ones and it is equal to

$$N_t = \sum_{i=0}^{t}\binom{n}{i} \tag{11}$$

Therefore the average work necessary for this random selection attack will be proportional to the number of vectors to be checked. Fig.3 shows the average work for different code sets assuming the average of proportionality is unity. Interception of a cryptogram $\underline{v}$ will reduce the size of possible cryptograms by at most N_t. Cryptograms that are close to $\underline{v}$ will be accepted by the receiver with a high probability but they are not good choices as they are decoded to the same $\underline{m}$.

The all-zeros information messages represents another source of weakness in the proposed schemes. The weight of a cryptogram corresponding to a non-zero plaintext or information vector is always greater than or equal to $t + 1$. Therefore it is always possible to differentiate between non-zero and all zeros information vectors by finding the weight of the received vector. If the weight is less than or equal to t, then the transmitted information vector is the all-zeros vector. So it is recommended to avoid the use of all-zeros codeword, or the use of codes of same

length but with different dimension. In the above analyses we have considered one possible attack on the cryptogram to find the private key. These analyses simply illustrate the impact of the code parameters on the system security. Other possible attacks can be considered to test the system robustness against attacks. This will be the subject of a future work.

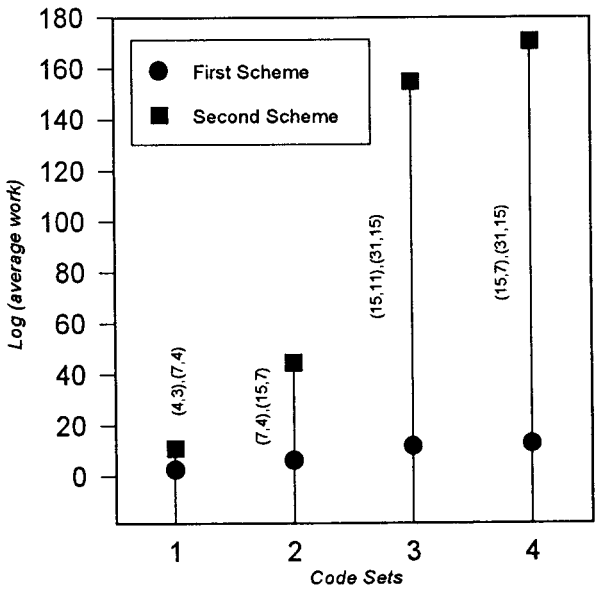

Fig. 3: Average work required for random selection attack for different code sets

SYSTEM PERFORMANCE

Another factors have to be considered in code selection, these are the channel error rate and the corresponding probability of unrecoverable error message. In what follows we will find the probability of incorrectly received message, i.e., the probability of unrecoverable message, in the two proposed schemes.

Assume an additive white gaussian noise communication channel and let the error rate be p_e. In the first scheme, the received message can be decoded correctly and the plaintext can be recovered in general if the number of errors introduced by the channel is less than t_2- $w_H(\underline{e})$. On the other hand, if the channel is noisy and no error vector has deliberately added by the transmitter and the transmitted codeword $\underline{c}_0$ (or $\underline{v}_0$ if multiplied by **G'** rather than **G**) is received as $\underline{c}_1$. $\underline{c}_0$ and $\underline{c}_1$ will differ at least in d positions and we can write the probability of a decoding error in favor of $\underline{c}_1$ as;

$$P(c_1 / c_0) = \sum_{i=\lceil d/2 \rceil}^{d} \binom{d}{i} p_e (1 - p_e)^{d-i} \qquad (12)$$

where $\lceil x \rceil$ indicates the least integer greater than x. This decoding error occurs if more than $\lceil d/2 \rceil + 1$ of the symbols where $\underline{c}_0$ ($\underline{v}_0$) and $\underline{c}_1$ ($\underline{v}_1$) differ are erroneous. A union bound on decoding error probability results in [12]:

$$P_E \leq \sum_{d=1}^{n} W_d P(v_d / v_o) \qquad (13)$$

where $\underline{c}_d$ ($\underline{v}_d$) is any word at a distance d from $\underline{c}_0$ ($\underline{v}_0$) , and W_d is the number of such words. In fact, W_d is a coefficient of the weight enumerating polynomial of the code. In the first scheme, the codewords of the first code represents a subset of the codewords of the second code. We cannot find the k_1 information symbols unless $d_1 > d_2$. To get the overall probability of incorrect message we can apply equation (12) after changing the code parameters and replacing the channel error probability p_e by p_e' which is the bit error probability after the first decoding stage. Unfortunately p_e' can only be found for codes whose weight enumerator polynomials are known such as the Hamming codes. An example of calculating the bit error probability after decoding is given in [12]

In the second coding scheme, we can follow the same procedure as before to find an upper bound for the decoding error probability using the overall code parameters n and d, where $n = n_1 n_2$ and $d = d_1 d_2$. On the other hand, if we are going to perform the decoding in two steps where we make use of the decoding rules for each code, the second decoder will be dependent on the first one. Assuming that we receive the cryptogram in the form of n_1 blocks, each of n_2 bits. If more than t_2 errors occur in any of these blocks, the decoder fails to find the corresponding plaintext (information bits) actually transmitted and produces another codeword which differs from the original one in at least d_2 positions. This in turns introduces more errors for the second decoder and increases the probability of

incorrect decoding. The same is true if the cryptogram is received in forms of n_1-bit blocks. If more than t_1 errors occur in any of these blocks, the nearest neighbor decoding will produce another codeword which at least at a distance d_1 from the transmitted one. Therefore more errors will be added to those already exist in the other blocks. This will definitely lead to errors in the information symbols. Let us consider the simple case where the plaintext cannot be found if any single bit of received cryptosystem is incorrectly decoded. We can write the probability of incorrectly received message, i.e., the probability of error P_E (assuming that $\underline{e}=\underline{0}$) as;

P_E = Prob. { (at least one of the n_1 n_2-bit blocks has more than t_2 errors) or (at least one of the n_2 n_1-bit blocks has more than t_1 errors)}

$$\begin{aligned} P_E &= P_1 + P_2 - P_1 \cdot P_2 \\ P_1 &= 1 - P_3^{n_1} \\ P_2 &= 1 - P_4^{n_2} \end{aligned} \tag{14}$$

where

$$P_3 = \sum_{i=0}^{t_2} \binom{n_2}{i} p_e^{\,i} (1-p_e)^{n_2 - i} \tag{15}$$

and

$$P_4 = \sum_{j=0}^{t_1} \binom{n_1}{j} p_e^{\,j} (1-p_e)^{n_1 - j} \tag{16}$$

From the above equations it is evident that in order to achieve small error probability over noisy channels, long encryption block sizes and large error correcting capability codes have to be used.

As explained in the previous section, the decoding will perform in cascade, therefore the two decoding processes will be dependent. The channel error rate will be different after the first stage, but due to symbol interleaving in the coding process, we assume that the errors affect each symbol independently. The scrambling and permutation processes do not change the error correcting capability of the code. Equation (14) gives an upper bound on the probability of unrecoverable messages. For illustrative purposes only, we compare the performance of the two proposed methods using small block codes since it is not possible to deal with usable block codes at very small channel error probabilities.

The results help to draw a conclusions based on these results. Fig. 4 shows the case where the first set of codes given in the first row of Table-I is used. Both schemes have the same data rate. At high channel error rates i.e., for $p_e > 0.1$, the performance of the first scheme is better than the second one. This is because the nearest neighbor decoding method introduces at least d errors if more than t errors occur as explained before. The second scheme performs better at a bit lower channel error rates because of its larger error correcting capability compared with the first scheme. Fig.5 shows improvement compared with Fig.4 because long codes with larger codes error correcting capabilities are used.

The above analyses help to select the codes that are adequate to the communication channels. For a given error rate or channel error probability, the code parameters have to be carefully selected such that the probability of error will be within the permissible range. It remains to say that the existence of fast decoding algorithm for the code is a decisive selection criterion.

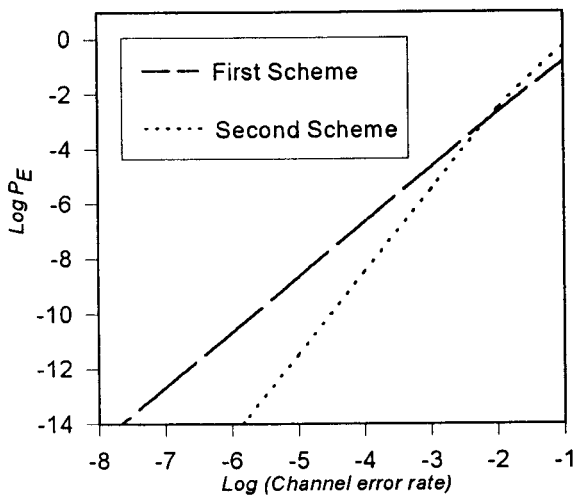

Fig.4: Probability of unrecoverable messages when (4,3) and (7,4) codes are used

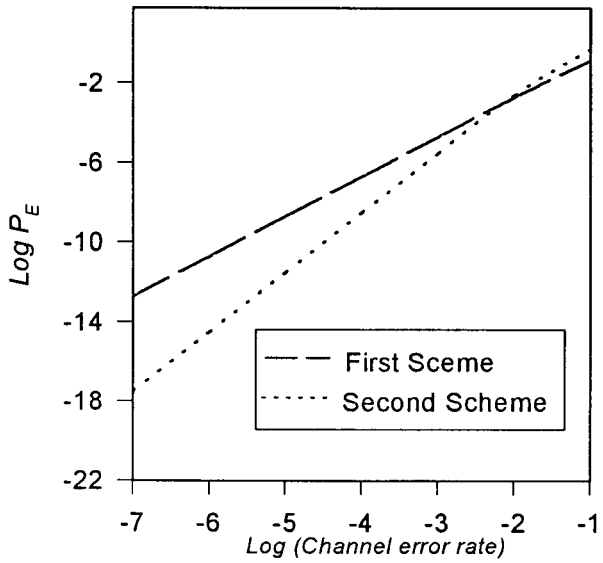

Fig..5: Probability of unrecoverable messages when (7,4) and (15,7) codes are used

CONCLUSION

McEliece public-key cryptosystem is one of the data encryption techniques that do not depend on unsolvable problems from number theory. On the contrary, the system is based on the coding theory, and on the fact that the decoding problem of a linear block code is an NP-complete. However, McEliece cryptosystem has not been widely used due to the size of the key, the structure of the system, and the low data rate. Also the use of minimum distance (nearest neighbor) decoding is not feasible for large codes. In this work we explain how the McEliece public-key algorithm can be used with codes other than Goppa codes. Small codes are combined together in a cascaded way to form long codes. The structure of the proposed encryption schemes is in itself a good way to protect the information from any opponent. This structure also increases the error correcting capability of the code so the system can also correct noise errors if the weight of the deliberately added error pattern (by the transmitter) is less than the overall error correcting capability of the structure. The cryptanalyst faces the problem of

decoding a corrupted codeword. If no information is available about the data structure, the cryptanalyst is forced to use inefficient decoding process. The large number of keys that can be used in the proposed schemes increases the difficulty of a successful attack. Even if a successful attack is made and the error pattern used to encrypt a particular block is discovered, a similar effort has to be made for each subsequent block of the message due to the change of the error pattern or a change in the key. The proposed schemes have the advantage of high speed encryption and decryption rates compared with the original algorithm as well as other systems based on number theory. This is true because fast encoding and decoding techniques are available for most of linear block codes especially those of small or moderate length. Non-binary codes can be used to get better error correcting capability. In this way we can overcome the main drawbacks of the original algorithm. Also, the proposed schemes carry with them more properties of error correcting codes than the original McEliece algorithm. The codes are chosen depending on the channel error rate as well as the required transmission or data rate. The trade-off between system security and its error correcting capability is clear. Long codes with high data rates are more secure. Meanwhile, high data rate codes have small error correcting capability so they cannot be used over noisy channels. Binary BCH codes are good candidates for both schemes because of their error correcting capability and the existence of fast decoding algorithms.

REFERENCES

1. **Diffie, W. and M. Hellman, 1979.** "Privacy and Authentication: An Introduction to Cryptography", Proceedings of IEEE, Vol. 67, No. 3, pp397-427.

2. **Brickely E. F. and Odlyzko A. M., 1992.** Cryptoanalysis, A Survey of Recent Results, Chapter 10 in "Contemporary Cryptography, The Science of Information Integrity", Simmons G. (editor), New York, IEEE Press.

3. **Agnew B., 1990.** "Cryptographic Systems Using Redundancy", IEEE Trans. Information Theory, Vol. IT-36, No. 1, pp.31-39.

4. **Safavi-Naini R. S. and Seberry J.R., 1991.** "Error-Correcting Codes for Authentication and Subliminal Channels", IEEE Trans. Information Theory, Vol IT-37, No.1 , pp.13-17.

5. **Federal Information Processing Standard (FIPS) Publications 46, 1977.** "Data Encryption Standard", January 15, pp.1-18.

6. **Diffie, W. and M. Hellman, 1976.** "New Directions in Cryptography", IEEE Trans. Inform. Theory, Vol. IT.22, No. 6, pp.644-654.

7. **Diffie, W. 1988.** "The First Ten Years of Public-Key Cryptography", Proceedings of IEEE, Vol. 76, No. 5, pp. 560-577.

8. **MacWilliams, F. and Sloane N., 1981.** "The Theory of Error Correcting Codes", Amsterdam, North Holland Pub., 3rd Printing.

9. **Peterson, W. and E. Weldon, 1972.** "Error Correcting Codes", Cambridge, MA, MIT Press.

10. **Grham, A., 1981.** "Kronecker Products and Matrix Calculus with Applications", Chichester, Ellis Horwood Pub.

11. **Adam, C. M. and Meijer H., 1989.** "Security-Related Comments Regarding McEliece's Public-Key Cryptosystem", IEEE Trans. Information Theory, Vol. 35, No. 2, pp. 454-455.

12. **Batttail, G., 1989.** "Coding for the Gaussian Channel: The Promise of Weighted-Output Decoding", International Journal of Satellite Communications, Vol. 7, pp 183-192.