JOINT SCHEMES FOR PHYSICAL LAYER SECURITY AND ERROR CORRECTION

Oluwayomi Bamidele Adamo, B.S., M.S.

Dissertation Prepared for the Degree of

DOCTOR OF PHILOSOPHY

UNIVERSITY OF NORTH TEXAS

August 2011

APPROVED:

Bill Buckles, Major Professor

Murali Varanasi, Co-Major Professor

Shengli Fu, Committee Member

Kamesh Namuduri, Committee Member

Mahadevan Gomathisankaran, Committee Member

Bill Buckles, Coordinator of Graduate Studies and

Ian Parberry, Interim Chair of the Department
of Computer Science and Engineering

Costas Tsatsoulis, Dean of College of Engineering

James Meernik, Interim Dean of the
Robert B. Toulouse School of Graduate Studies

CONTENTS

List of Figures

List of Tables

ACKNOWLEDGEMENT

I am very grateful to my major professors Dr. Murali Varanasi and Dr. Bill Buckles for being my advisors. I would especially like to thank Dr. Murali Varanasi for his guidance and support throughout my studies. His wealth of knowledge contributed immensely to my research work. I would also like to thank Dr. Kamesh Namuduri for his advice and spending time to guide me. I am grateful to my doctoral committee members, Dr. Shengli Fu and Dr. Mahadevan Gomathisankaran for their advice.

I would like to thank my wife, Monica Elise Adamo and my sons, Oluwayomi Wesley Bamidele Adamo and Wojuoluwa David Omotokunbo Adamo for their encouragement and patience throughout my research work. I would have not completed this degree without the continued encouragement from my father, Rev. Dr. David Tuesday Adamo who showed me the importance of education.

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.1. Motivation

Due to the rapid increase in the applications that can be carried out on portable wireless devices, it becomes necessary to secure data transmitted through these devices. Even though early work from [57] showed the existence of secrecy-achieving codes, data reliability and security schemes are still viewed as two different processes in a contemporary communication system. Error correction is carried out at the physical layer while security is performed at upper layers. Many security protocols today are designed and implemented with the assumptions that physical layer provides error free information. However with the emergence of resource constraint wireless devices and ad-hoc network, encryption at higher layers become difficult to implement. As a result, there has been a lot of interest in implementing encryption at the physical layer. For example, It was pointed out in [21] that the best and often the only way to secure data in a wireless sensor network is to encrypt the data using a secure encryption algorithm before it is transmitted over the airways. It was shown that the cost of a software-based encryption procedure could outweigh the risks of the transmission being intercepted because of the constraint nature of resources, memory and clock speeds on the sensor nodes. In recent years, protocols such as Wifi Protected Access (WPA), Wired Equivalent Privacy (WEP), Temporal Key Integrity Protocol which were implemented at upper layers are now implemented at lower layers such as the link layer. It was suggested in [10] that encryption in the link layer results in transmission delays which lead to retransmission in upper layers. This problem could be resolved by moving encryption and decryption to the physical layer.

Physical layer encryption was presented in [60] and [1]. However these encryption modules are visualized as separate modules from the error correction modules. Contrary to their models, we propose a joint scheme that combines encryption and error correction in one step for physical layer encryption. In such a case, the secrecy achieving characteristics of channel codes could be exploited. This leads to improved efficiency, speed and savings in hardware usage because of hardware reuse. This also give flexibility in terms of design and technology used for fabrication. It is also difficult to build lower layer analyzers in terms of attacks.

The conventional secure communication model with the sender (Alice), legitimate receiver (Bob) and the eavesdropper (Eve) [57] is shown in Fig. 1.1. Alice would like to send a confidential and reliable message u to Bob with whom she shares a secret key while making sure that Eve has no knowlege of u. She does this by encrypting u with the secret key k to obtain a ciphertext c. The ciphertext c is encoded by introducing redundancy into c to obtain x so that channel errors could be detected and corrected at the receiver by Bob. Upon receiving y, the legitimate receiver (Bob) decodes it to obtain $c^{'}$ and he then decrypts with the aid of k to obtain the message $u^{'}$ intended for Bob. It is important to note that eavesdropper (Eve) has knowledge of the decoder, hence she can obtain an error-free ciphertext as shown in Fig. 1.1. The knowledge of the decoder does not decrease the Shannon's entropy of u given $c^{'} = c$ which can be expressed as $H(u/c) = H(u)$.

In Fig. 1.1, we show the alternative secure communication model for our scheme. Here, Alice wants to send a message u to Bob with whom she shares a secret key while making sure that Eve has no knowlege of u. She does this by passing the message through the Error Correction Based Cipher (ECBC) to obtain encoded ciphertext x with the aid of secret key k. Upon receiving y, the legitimate receiver (Bob) decodes and decrypts in a single step using ECBC with the aid of k to obtain the message $u^{'}$ intended for Bob. The eavesdropper (Eve)

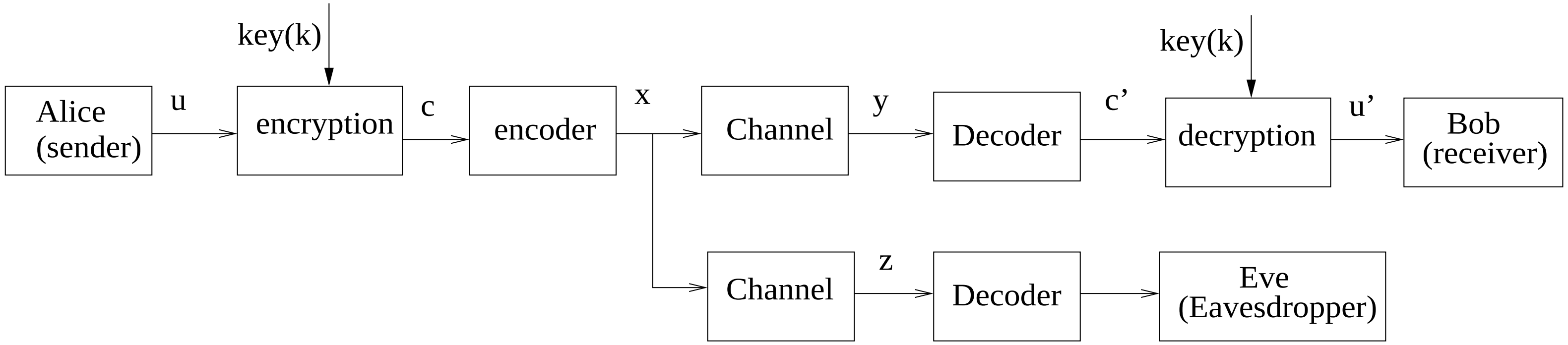

Figure 1.1. Block diagram of a conventional secure communication system model

does not have a knowledge of the key to ECBC, hence the ciphertext she receives is not error-free as shown in fig 1.2. The Shannon's entropy of u for our model is therefore larger than that of the conventional model which can be expressed as $(H(u/c'))_{ecbc} > (H(u/z))_{conventional}$.

This research combines the encryption and channel coding as one process thereby resulting in a potential reduction in hardware usage. The potential reduction in hardware usage will lead to an increase in power savings [32] as power consumption reduction and area efficiency are of utmost importance in modern wireless communication [41]. Also, there is no tradeoff between data reliability and security in the joint schemes presented in this dissertation as opposed to previous schemes which were presented in [22], [33], [43]. The cryptanalysis of the schemes, the result of the randomization test on the schemes and their hardware implementations will be presented. The joint schemes presented in this research carry out both encryption and error correction in a single step as opposed to two separate steps.

Several similar works have been done in the past. For example, the authors in [60] considered an architecture for physical layer encryption that first converts information sequences to longer channel codewords and then encrypts them using classical stream cipher. They pointed out that even though their architecture requires longer encryption sequences, it could use the natural randomness of the communication channel against known-plaintext. In our scheme, the order of the two processes is not of concern since they are done in one step by one unit. The authors in [2] pointed out how physical layer encryption is taking significant importance in wireless network security. They propose an efficient physical layer encryption that relies on implementation of Output Feedback (OFB) mode just after error correction.

The use of error correcting code as a public-key cryptosystem was first introduced by [33]. Mceliece scheme is based on algebraic coding theory using t-error correcting Goppa code [33]. However, his scheme requires large block length (n = 1000) in order to correct large number of errors (t = 50 bits). This results in very large computational overhead [43]. A private key algebraic-code using McEliece scheme was proposed in [44] where the generator

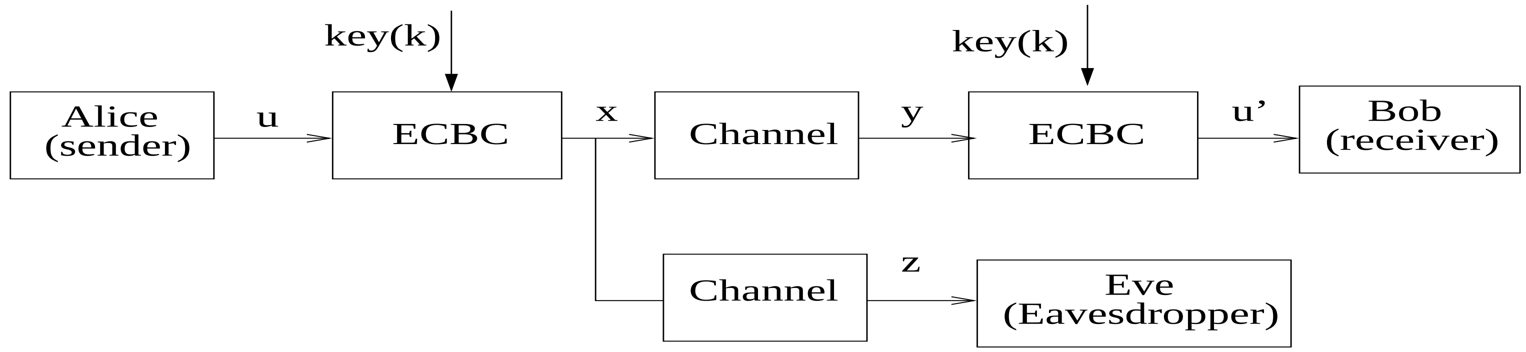

Figure 1.2. Block diagram of a secure communication system using ECBC model

matrix was made private. The scheme provides better security with simpler error correcting code thereby making it less computational intensive compared with McEliece. However, it was shown that the scheme could be easily broken by a chosen-plaintext attack [43]. They introduced a private key cryptosystem that requires simpler error correcting codes with distance ≤ 6 and block length n ≤ 250. If these schemes are used for error-correction based ciphers, there is a tradeoff between reliability and security. The Secret Error Correcting Code (SECC) using nonlinear preparata code was presented in [22]. The two schemes were shown to preserve full error correcting capability while providing data secrecy. The first scheme did not incorporate the error vector in the original McEliece cryptosystem and the second scheme did not incorporate the permutation process. The system was found vulnerable to the known plaintext attack conducted in [58]. However, in our scheme, the permutation process (P) is incorporated thereby increasing the security of our system. The non-linear function chosen also prevents the known-plaintext attack against the cipher. The most recent joint scheme for error correction and cryptography was presented in [10] where they used High Diffusion (HD) codes . They built their cipher using the structure of Advanced Encryption Standard (AES) [13] replacing the high diffusion layer of the AES with error correcting code. Though their scheme provides data security and error correction, it is higher in complexity compared to McEliece based scheme. They [10], [32] even confirmed that McEliece based schemes have advantage of low power consumption by using the same hardware components available for error correction for security. As a result, McEliece-like schemes are desired for a constrained environment. Our Error Correction based Cipher provides data reliability, integrity and security. The full error correcting capability of the error correcting code is preserved.

1.2. Background

The necessary concepts needed to understand this research is presented in this section.

1.2.1. *Data Reliability*

In order to provide reliability, linear codes are usually employed in the physical layer. In an (n, k) linear block code, the information sequence is divided into message blocks of k bits each. Each message block is represented as a binary k-tuple M=(m_1, m_2, m_3, ..., m_k) which implies that there are 2^k possible message blocks that will be unique. For data reliability, the encoder encodes each message block independently into an n-tuple codeword C = (c_1, c_2, c_3, ..., c_n). Hence, 2^k distinct messages correspond to 2^k distinct codewords at the output of the encoder. The ratio R = k/n is called the information rate of the code. It is important to note that linear block code is a subclass of all block codes. Codes could be broadly classified as block and convolutional codes. Encoding in convolutional codes are memory based and the mapping of message M to codeword C depends on both the current state of the encoder and previous message. Block codes can be further divided into linear and non-linear codes. as shown in Fig. 1.3. The mapping of message to codeword in linear block codes is usually predetermined by a generator matrix G. Examples of linear block codes are cyclic codes, Low Density Parity Check (LDPC) codes and perfect codes. Any valid codeword in a cyclic code is a cyclic shift of another codeword in the same cyclic code. Examples of cyclic codes are Reed Solomon (RS) codes and BCH codes. Examples of perfect codes are repetition code, Hamming code and Golay codes.

1.2.1.1. *Linear Block Codes.* [45]

In order to define linear codes, we will first define Hamming weight and Hamming distance.

Definition 1.1 (Hamming Weight). The Hamming weight of a binary n-tuple *u*, denoted as *w(u)*, is the number of nonzero components of *u*.

For example, Given *u* = (10001000) and *v* = (10001011), the Hamming weight of *u* is 2 and that of *v* is 4 i.e *w(u)* = 2 and *w(v)* = 4. We can express the Hamming distance between *u* and *v* as the Hamming weight of the sum of *v* and *w*, i.e

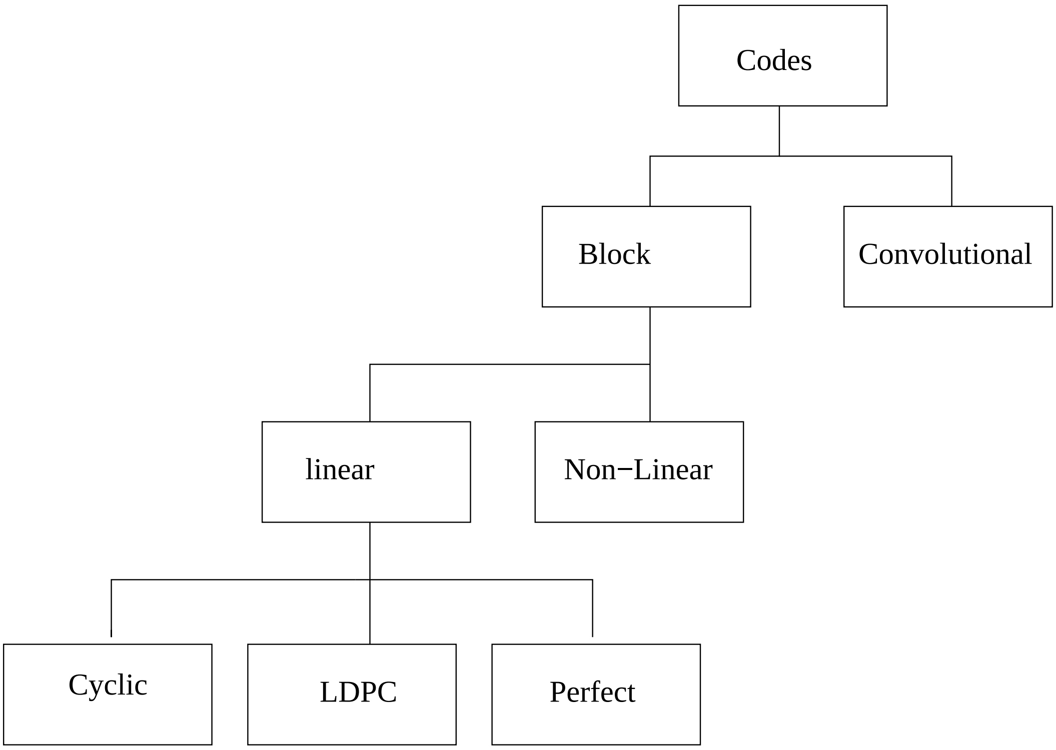

Figure 1.3. Block diagram Illustrating the Classification of Codes

Definition 1.2 (Hamming Distance). Hamming distance between n-tuple *u* and *v*, denoted as *d(w,v)* is the number of places where they differ.

For example, the Hamming distance between *u* and *v* is 2, where they differ in the sixth and the 7th places.

Definition 1.3 (Linear Block Code). [45] A block code of length *n* and 2^k codewords is called a linear (n, k) code if and only if its 2^k codewords form a k-dimensional subspace of the vector space of all the n tuples over the field GF(2).

In other words, a linear code is an error-correcting code in which any linear combination of codewords is another codeword of the code.

Definition 1.4 (Linear Block Code). [45] The minimum distance of a linear block code is equal to the minimum weight of its nonzero code words.

As a result, the minimum distance a linear block code C is its minimum weight. The hamming distance of any two n -bit codeword of an (n, k, d) code is of at least d. Since an (n,k) linear code forms a k-dimensional subspace of the vector space V_n of all binary n-tuples, it is possible to find k linearly independent codewords to span the code space. These k linearly independent codewords can be represented by a matrix G called the generator matrix. It is important to note that any k linearly independent codework of a (n,k) code can be used to form a generator matrix G. A linear code could thus be described by a generator matrix (G). A linear block code could possess a systematic structure where a codeword is divided into a message part and a reduntant checking part. The message part is k unaltered message digits and the redundant checking part is n-k parity check digits. The kxn generator matrix G that specifies a linear systematic (n,k) code can be expressed in the form:

$$G = [I_{k\times k} \; P_{k\times(n-k)}] \tag{1}$$

where $I_k \times k$ is a k x k identity matrix.

For any (kxn) generator matrix G of an (n,k) linear code, there is an (n-k)xn parity-check matrix H such that any vector in the row space G is orthogonal to every row of H and any vector that is orthogonal to every row of H is in the row space of G. We can thus describe (n,k) linear code generated by G in alternate way: An n-tuple c is a codeword in the code generated by G if and only if

$$cH^T = 0 \tag{2}$$

The matrix (H) is called a parity-check matrix of the code. This means that an element c which is a codeword of code C is a codeword *iff* Eqn. 5 is true. Also $GH^T = 0$. If the generator matrix G of a code C is given by $G = [IA]$ then $H = [-C^T I]$ is a parity-check matrix of code C. The set of vectors that are orthogonal to all vectors (codeword of C) is known as the dual code. The parity check matrix of a code C is a generator matrix of the dual code $C^{\perp}$ of C.

If c is a codeword that was transmitted in a noisy channel, the codeword $c' = c + E$ that is received at the receiver will be different from c because of error E = $(e_1, e_2, e_3......, e_n)$ due to the channel. The decoder calculates the syndrome ($S = H.c'^T$) of c' and if $S = 0$, then c' is a codeword. It is also possible that the error vector is identical to a nonzero codeword. In that case c'$H^T = 0$ and this means the error cannot be detected. On the other hand, if S $\neq$ 0, then c' is not a codeword and that means the presence of error has been detected and could possibly be corrected. To illustrate encoding and decoding process, we show the following example.

1.3. Example of Encoding and Decoding

The generator matrix G of a (7,4) linear code and its corresponding parity check matrix H is given below:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \tag{3}$$

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{4}$$

If a sender wants to send a message $(u) = 0101$ to a receiver, the message is encoded into a codeword c by performing:

$$c = u.G = 0101110 \tag{5}$$

If an error E=0000001 occur in the channel, the receiver will receive $c' = c$+E = 0101111. In order to decode the received word c', the receiver will compute the syndrome S=c.H^T = 001. Because the syndrome is identical to the seventh column of H, the error is detected at the seventh position. As a result, the error can be corrected.

A Maximal Distance Separable (MDS) code is a code that meets the singleton bound. The singleton bound gives the upper bound of the distance of a code C with given dimension. Given an (n, k, d) code, then the singleton bound states that $d \leq n - k + 1$. Every set of n-k columns of the parity check matrix of an MDS are linearly independent. Reed-Solomon codes form a class of MDS code [45].

A brief introduction to linear block codes has been presented in this section. A detailed presentation can be found in [45] and [40]

1.3.1. *Data Security*

Cryptography is used for protecting information transmitted through a communication network that uses communication satalites, microwave or landlines. Cryptography is the science and art of keeping a message secure. Cryptanalysis on the other hand is the science and art of uncovering the secrecy of a message. Cryptanalysts are individuals who practice cryptanalysis. Cryptocomplexity is the work factor required to break a cipher. A cipher is an algorithm for performing encryption or decryption. Encryption is the process of transforming a plaintext P to an unintelligible form called ciphertext C such that only those that possess special knowledge (key) can read it. Decryption on the hand is the process of transforming a ciphertext to a plaintext using a special knowledge (key). A cryptographic system contains five components [14]: a plaintext, ciphertext, key, encryption process and decryption process.

A cipher where the encryption key is different from the decryption key is called public key cipher. In this case, the key used for encryption is made public while the key used for decryption is made secret. Examples of public key ciphers include Elliptic curve, RSA, Diffie-Hellman, DSA and ELGAMAL. A cipher whose encryption and decryption keys are the same is called symmetric key or private key ciphers. Examples include Advanced Encryption Standard (AES), Blowfish, DES, Serpent and RC4 [42], [46]. A symmetric key cipher could be a stream cipher or a block cipher. In a stream cipher the plaintext digits are encrypted one at a time, and the transformation of successive digits varies during the encryption. In a block cipher a fixed-length groups of bits or block of bits are encrypted at a time [46].

The purpose of encryption is to keep the plaintext secret from the eavesdroppers. However, eavesdroppers always try to carryout cryptanalytic attacks on ciphertexts. In this case, the eavesdropper becomes the cryptanalyst. The Cryptanalytic attacks could be Ciphertext-only attack, Known-plaintext attack or Chosen-plaintext attack [47].

In Ciphertext-only attack, the attacker has several ciphertexts that were encrypted with the same algorithm. The work of the cryptanalyst is to try to deduce as many plaintexts or keys as possible. The ciphertext-only attack is the weakest attack and any cryptosystem that weak under the ciphertext-only attack is completely useless [14]. In the case of Known-plaintext attack, the crypatanalyst has acess to both the ciphertext and the plaintext from which the ciphertext was generated. The attacker tries to deduce the key used to encrypt the plaintext or the algorithm that could be used for the decryption of subsequent ciphertexts with the aid of the same key. The Chosen-plaintext attack is similar to the Known-plaintext attack except that the attacker can choose specific plaintext with the hope that they will reveal more information about the keys. Contemporary attacks such as linear and differential cryptanalysis fall under one of the attacks discussed earlier or their combinations. For example, linear crypanalysis is a Known-plaintext attack and differential cryptanalysis is a Chosen-plaintext attack. Each of the ciphers presented in this research will be analyzed under the attacks discussed earlier.

1.3.2. *Data Integrity and Authentication*

Cryptography could also be used to provide data integrity and authentication. If cryptography provide data integrity, then the receiver is able to determine whether the data has been modified. In the case of data authentication, the receiver is able to ascertain the origin of a data. If a pattern can be made dependent on a plaintext P such that it will be known with high probability if the pattern is changed whenever the recovered plaintext P' is different from P, then the pattern can be concatenated to the end of the message to detect any illegal modification to the plaintext [34]. The error propagation property of a cipher could be exploited to provide data integrity. In this case, due to block chaining effect, error due to a particular block of data will propagate to subsequent codeword. Hence, it shows that the codeword has been tampered with. In this case, the codeword is used as a checksum.

CHAPTER 2

RELATED RESEARCH

Even though encryption and error correction have been viewed as separate entities for many years, some researchers investigated the relationship between the two entities [54]. There have only been few successful attempts in the past to combine error correction and security. In this chapter, we briefly discuss some of the related research in combining error correction and security.

2.1. McEliece Public Key Cryptosystem

McEliece was the first to propose a public-key algebraic-code cryptosystem using Goppa codes [33] based on the result in [7] that the general decoding problem for linear codes is an NP-complete problem. In this cryptosystem, $G' = SGP$ is the public key where S is a (k × k) nonsingular matrix called the scrambler, G is a (k × n) generator matrix of a t-error correcting Goppa code and P is a random (n × n) permutation matrix. Encryption is carried out in the McEliece Public Key Cryptosystem based on Eqn. 6:

$$C = MG' + Z \tag{6}$$

where C: ciphertext of length n,

M: plaintext of length k,

Z: random error vector of length n whose hamming weight t' = t,

G': public-key,

S, G, P are private keys.

Decryption is carried out by using the private key keys S^{-1}, P^T and H^T where H is the parity check matrix of the Goppa code. The decryption process used in recovering the message M is illustrated below:
First, C' is computed using

$$C' = CP^T = M'G + ZP^T \tag{7}$$

where $M' = MS$. It is important to note that the hamming weight of ZP^T is the same as the original random error Z.
The syndrome is computed using Eqn. 8

$$C'H^T = M'GH^T + ZP^T H^T \tag{8}$$

M' is recovered by removing the error vector ZP^T using Patterson's algorithm [33]. The original plaintext M is then recovered from M' using Eqn. 9.

$$M = M'S^{-1} \tag{9}$$

Two attacks against the McEliece scheme were suggested in [33]. The cryptanalyst might factor G' to determine G. Patterson decoding algorithm could then be used to recover the plaintext. However, the method seems hopeless because of the astronomical number of choices for S, G, and P if n and t are large enough. The crypanalyst can also select k linearly independent columns of G' and solve to retrieve the plaintext from k components of C. It is important to note that the scheme is hard to break without the private keys S, G and P if t is large. One of the disadvantages of this scheme is that it has a low information rate. The scheme also requires large distance code and block lengths which leads to decryption overhead. Goppa code leads to the use of large matrices as keys.

McEliece scheme was not considered as a joint error correction and encryption code because the code could not correct any channel errors [32]. The scheme could however be used as a joint scheme by letting t' $<$ t. There has been much research done on improving the information rate and security of the McEliece scheme in [38], [48], [24], [53] and [30]. However, there is still information leakage in their scheme. For example, two attacks on the McEliece scheme were proposed in [8]. In McEliece scheme, even though error correcting code was utilized, the codes were not meant to correct channel errors. We could conclude that there is a tradeoff between error correction and security in the McEliece scheme. The advantage of algebraic based cryptosystem is that they are faster than contemporary cryptographic scheme like RSA. The drawback of the algebraic based scheme proposed by McEliece is that it requires very large key size and low information rate due to the Goppa code used. As a result, many researchers have conducted research with aim of reducing the key size and increasing the information rate [5], [39]. McEliece scheme thus suffers from low information rate, large key length and the inability to correct channel errors.

2.2. Private-Key Algebraic-Code Encryption

The private-key algebraic-code system was proposed in [44]. The scheme is a private key version of the McEliece public-key system where the public key in the McEliece scheme is made secret. As a result, simple linear codes ($d_{min} \leq 6$) is utilized. The scheme provided better security under ciphertext attacks but was shown to be weak under chosen plaintext and known plaintext attacks. A known-plaintext attack of their scheme is feasible by independently solving matrices for each column vector of the private key G'=SGP. A detailed chosen-plaintext attack on the Private-Key Algebraic-Code encryption is explained in [43].

2.3. Rao-Nam Scheme

The private-key algebraic-code cryptosystem was presented in [43]. In this scheme, a k-bit plaintext block M was enciphered into an n-bit ciphertext C using Eqn. 10:

$$C = (MSG + Z).P \tag{10}$$

where

S is an arbitrary (k × k) nonsingular matrix

G is an (n, k) code generator

P is a random (n × n) permutation matrix

Z is a n-bit error vector randomly selected from a repository of syndrome-error table

S, G and P are private keys.

In the scheme presented, decryption is achieved by using S^{-1}, H^T, and P^T. A summary of their scheme is shown below:

(i) $C' = CP^T = M'G +$ Z where $M' = M.S$

(ii) $C'H^T = M'GH^T + ZH^T = ZH^T$ which is the syndrome and this can be used to identify the error vector from syndrome-error table

(iii) Decoding is applied to obtain M' which is then multiplied by S^{-1} to obtain $M = M'S^{-1}$

In Rao-Nam scheme, any error vector can be introduced from the syndrome-error table during encryption. The drawback of this scheme is that any error due to the channel cannot be decoded. The number (N) of error vector depends on the rate of the code employed. In the scheme, $N = 2^{n-k} = 2^{n(1-k/n)}$. Their scheme is insecure against Struik-Tilburg (ST-type) chosen-plaintext attacks [50]. However, the author suggested that the use of a byte error correcting code or a non-linear code like preparata code could prevent ST-type attack. In the case of byte error correcting code, the size of error vector that can be used could be made large. In the case of the non-linear code, since the ST-type attack take advantage of the structure of linear codes, the use of non-linear code will deter the ST-type attack.

2.4. Private Key Cryptosystem by Sun

This scheme [52] is a private-key cryptosystem based on burst error correcting codes. Similar to previous schemes, the generator matrix G of a burst error correcting code $B(n, k, d, b))$ and the permutation matrix P are kept secret.

In the encryption scheme, a randomized burst sequence $E_{l,w}S$ is XORed with a plaintext M and then encoded with G. The result is further XORed with a burst error and then permuted with the aid of the permutation matrix to obtain the ciphertext. The ciphertext is calculated based on the Eqn. 11

$$C = ((M + E_{l,w}S)G + E_{l,w})P \tag{11}$$

where

+ is a modulo-two addition

$E_{l,w}$ is a random burst of length l with hamming weight w.

$w_{min} \leq$ w $\leq$ l $\leq$ b

w_{min} is a fixed number greater than t

t is the error correcting capacity of the code

b is the maximum burst length

In the decryption scheme, the ciphertext is permuted as shown in Eqn 12

$$C' = CP^{-1} = ((M + E_{l,w}S)G + E_{l,w}) \tag{12}$$

The decoding algorithm of the burst error correcting code is applied to the result to obtain C'' as whown in Eqn. 13.

$$C" = (M + E_{l,w}S) \tag{13}$$

The plaintext M is obtained by computing for M as shown in Eqn. 14.

$$M = (C" + E_{l,w}S) \tag{14}$$

where

$+$ is a modulo-two addition.

The strength of their scheme is based on the fact that the burst-error-correcting capacity of a binary linear block burst-error-correcting codes is, in general, larger than its random error-correcting capacity. The drawback of this scheme is that the original error correcting capacity of the error correcting code is not preserved because of the burst errors added to the plaintext.

2.5. Kak's Scheme

A joint encryption and error correction scheme that is based on decimal expansions of fractions known as D-sequences was proposed in [25]. They show that the encoding operation is equivalent to that of exponentiation in finite field, which is similar to encryption in public key cryptosystem. Their work did not attract further research because it was not scalable.

2.6. Godoy and Pereira's Scheme

This scheme [17] is a cryptographic system that combined error correction with cryptographic protection. The algorithm was proposed for a system that already has error correction implemented so that there will be no fundamental changes. The scheme is based on the property of a (n,k) linear block code where there exist a lot of possibilities to distribute k information symbols among the n symbols in a codeword. These possibilities increase when

$n \gg k$. New generator matrices were generated from existing generator matrices by using permutations. The security of the system was based on the secrecy and the change of the generator matrices. The drawback to the scheme is that the generator matrices of the given code are finite and countable. As a result, the system is susceptible to brute force attack on the generator matrix.

2.7. SECC Scheme by Hwang and Rao

Rao and Hwang in [22] proposed two schemes which they called Secret Error Correcting Code (SECC). Their scheme is a private key cryptosystem that uses a non-linear error correcting code called Preparata code [31]. The drawback of their scheme is that the error correcting capacity of the code used is reduced. In order to obtain a substantial capacity, the parameters of the system have to be very large. This leads to high computational complexity. Additionally, their cryptosystem was found vulnerable to the known plaintext attack [58].

2.8. Cryptcoding by Gligoroski, Knapskog and Andova

A joint error correction and encryption that is based on quasigroup (Latin Square) string transformation was proposed in [16]. In this scheme, every message is padded with bunch of zeros before the encoding and encryption. The decoding procedure is used to verify the presence of zeros. The received codeword symbols are continuously flipped until zero pad is recovered. It is important to note that the space of quasigroup gives the security of their technique. The drawback of this scheme is that the decoding scheme is extremely complicated and as a result cannot be used in a resource constrained network.

2.9. Error-Correcting Ciphers by Mathur, Narayan and Subbalakshmi

Mathur, Narayan and Subbalakshmi in [10] and [32] proposed two error correcting block ciphers called high diffusion cipher and pyramid cipher that uses the structure of the Advanced Encryption Standard (AES). They proposed a class of error-correcting codes called HD-codes with built-in security features that is used in the diffusion layer of the high diffusion cipher.

The high diffusion cipher is a ten round cipher which uses small high diffusion codes. The pyramid cipher uses a larger HD cipher. Their HD-cipher has imput size of 128 bits and ciphertext and key size of 288 bits. Though their scheme provides data security and error correction, the complexity of their scheme is higher than McEliece-based schemes. McEliece-like schemes are desired for resource-constrained network. The block size of the ciphertext in the HD cipher is greater than twice that of the Advanced Encryption Standard.

2.10. Su and Xiao's Scheme

The Su and Xiao in [51] also proposed an error correcting cipher that is based on the structure of Advanced Encryption Standard (AES). The cipher is also based on the wide trail strategy. Their cipher is a six round block cipher that encrypts 256 bits of plaintext to a 512 bits ciphertext. They utilized LDPC code in the diffusion layer of the AES structure. Even though they used 128-bit key, the size of their plaintext and ciphertext is twice that of the AES. There is also redundancy in the units that make up a round. This makes their scheme less suitable for resource constrained network.

2.11. Our Schemes

In this dissertation, three different joint schemes are presented. We broadly classify the three schemes into two groups; Error Correcting Based and Cipher Based schemes. In Chapter 3, two error correction based ciphers (ECBC) which we refer to as ECBC 1 and ECBC 2 are presented. The ECBC 1 is an algebraic-based private key cryptosystem that utilizes a block chaining-like structure. The ECBC 1 scheme is made up of a non-linear transformation (f), encoding process and a permutation transformation. The ECBC 1 scheme employs non-linear transformation to prevent a known plaintext attack. We employed S-Box for the non-linear transformation. A non-linear expansion function is also employed to prevent chosen plaintext. The Low Density Parity Check (LDPC) code which is a linear code with that has performance near capacity is employed in the encoding process of ECBC 1. We show that there is no tradeoff between error correction and security in this scheme. The cryptanalysis of the scheme is also presented in chapter 3.

The ECBC 2 is also presented in Chapter 3. The ECBC 2 is a McEliece-like private key algebraic based cryptosystem. Instead of using Goppa code with low information rate, or simple linear code, a non-linear code called Nordstrom-Robinson (NR) code is employed. The NR code has twice the codeword of a linear code with the same distance and length. This is an advantage because it increases the number of possible error vector that can be used in the cryptosystem. This is important because security of the McEliece-like scheme is based on the code generator (G) and the error vector. The NR code is linear in Z_4 because it is a binary image of an octacode. As a result, it can be used as though it is a linear code. This will result in simple decoding for the intended recepients of the cipher. However, we try to exploit this characteristic. We show that this scheme is secure against ST-type attack and other chosen plaintext attack. We present this scheme in Chapter 3.

The AES-McEliece Hybrid (AMH) cipher is presented in Chapter 4. This is a private key cryptosystem. This scheme takes 128 bits of input plaintext similar to the Advanced Encryption Standard. We take advantage of the characteristic of McEliece-like and AES-like schemes to produce the AMH cipher. This scheme is categorized as cipher based because it takes advantage of the structure of Advanced Encryption Standard (AES). The LDPC code is employed in this scheme. The AMH cipher results in less rounds (6) as opposed to the standalone AES with 10 rounds. The resistance against linear, differential cryptanalysis and Square attack is presented in Chapter 4. The AMH code has reduced complexity and higher security than AES. The AMH cipher is subjected to randomization test just like the AES and it passed all the test suite. A detailed analysis of AMH cipher is presented in Chapter 4.

The architecture and the implementation an ECBC scheme is presented in Chapter 5 while the architecture and implementation of AMH cipher is presented in chapter 6. In order to show that McEliece-like scheme results in less hardware usage, the result of the hardware consumption and speed are presented in chapter 5. It clearly shows that McEliece scheme are more efficient than conventional scheme (AES-like schemes). In Chapter 6, a joint scheme that combines error correction (McEliece-like scheme), security and modulation is presented. This leads to reduction of key in the original McEliece-like scheme since randomization is introduced in the modulation process. The LDPC code is employed in the joint scheme. Cryptanalysis of the scheme is also presented in Chapter 7. We present a summary of the dissersation and future research in Chapter 8.

CHAPTER 3

ERROR CORRECTION BASED CIPHER (ECBC)

3.1. ECBC Scheme 1

We present a private key algebraic based system for physical layer encryption called Error Correction Based Cipher (ECBC) that combines encryption and error correction into a single step. The scheme is based on the block chaining technique. In ECBC, a k-bit plaintext block M is enciphered into n-bit ciphertext block C. The s-box and the Low Density Parity Check code is employed in the construction of this scheme. A detailed explanation of ECBC is presented in this section.

- A stream of data is divided into k-bit blocks M_i, i = 1, 2, 3, ...N.
- Plaintext M_i is xored with a randomization vector to obtain d_i. The first plaintext block M_1 at time 1 is randomized by XORing it with a k-bit initialization vector (Q_0^*=initialization vector(IV)).
- A non-linear function f transforms d_i into X_i. The reason for the use of non-linear function will be explained in the cryptanalysis section of this paper.
- The output of the non-linear function X_i is encoded with the aid of the generator matrix G to obtain b_i. X_i is also stored in a register for obtaining a delay version which is then used to produce randomly generated vector Z_i with the aid of an expansion function (g).
- The encoded data b_i is permuted with the aid of permutation matrix P to produce Q_i. The first k-bit of Q_i is denoted as Q_i^* and is delayed with the aid of a register to produce Q_{i-1}^* which will be xored with the next block M_{i+1}.

- The randomly generated error vector Z_i is then added to Q_i to form ciphertext C_i which is then sent through the channel.

A ciphertext C_i is expressed as in equation 15

$$C_i = (X_i GP + Z_i) \tag{15}$$

where X_i = f($M_i + Q^*_{i-1}$)

The block diagram representing the encryption process of ECBC is shown in Fig. 3.1. Ciphertexts C_i for i = 1, 2, 3 .. etc can be expressed as:

$$C_1 = f(M_1 + Q_0)GP + Z_1 \tag{16}$$

where $Q_0 = IV_1$ and Z_1=$g(IV_2)$.

$$C_2 = f(M_2 + Q^*_1)GP + Z_2 \tag{17}$$

where Q_1 =f(M_1 + Q_0)GP and Z_2=$g(X_1)$, X_1=f(M_1 + Q_0).

$$C_3 = f(M_3 + Q^*_2)GP + Z_3 \tag{18}$$

.

.

.

$$C_i \;=\; f(M_i + Q^*_{i-1})\; GP + Z_i \tag{19}$$

where Q^*_{i-1} =f(M_{i-1} + Q_{i-2})GP and Z_i=$g(X_{i-1})$, X_{i-1}=f(M_{i-1} + Q^*_{i-2}).

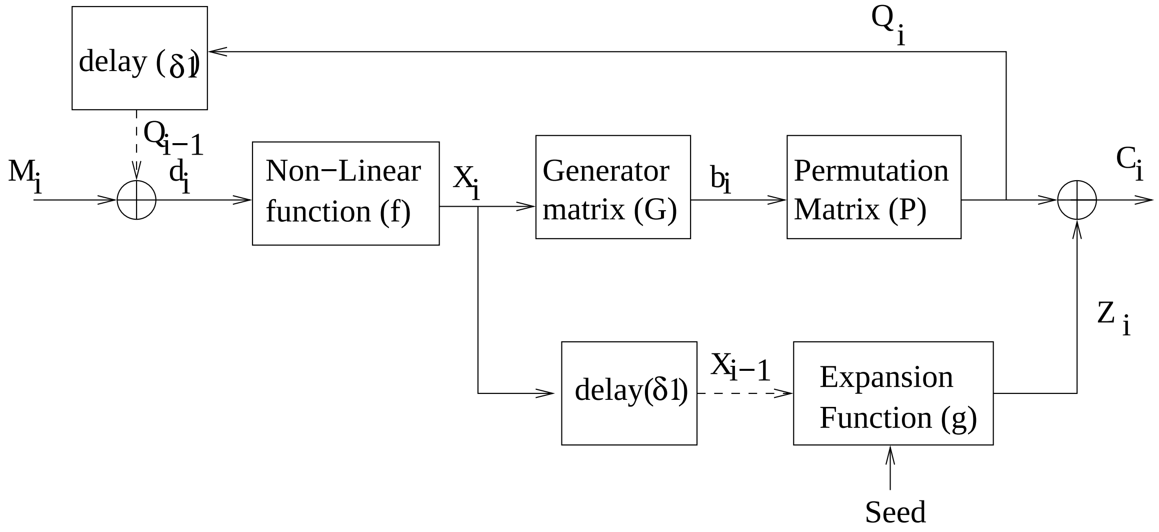

Figure 3.1. Block diagram of the proposed ECBC encryption scheme

The block chaining effect of this scheme allows the same plaintext block to be enciphered into different ciphertexts. Block chaining is a mechanism where each block of plaintext is XORed with the previous ciphertext block being encypted. Similarly, the decryption of a block of ciphertext depend on all the preceding ciphertext block. From the encryption algorithm, the cryptanalysis would be difficult. This because the cryptanalyst cannot construct a combinatorially equivalent generator matrix of the code from the ciphertexts because the ciphertexts are not codewords. Hence, the cryptanalyst cannot correct errors systematically. Also the cipher also employs double randomization since the plaintext is XORed with Q^*_{i-1} and the permuted codeword is XORed with Z_i. This also prevents construction of the generator matrix from the ciphertext.

For decryption, we assume that the receiver has to agree with the transmitter. This means that they have to agree on the initial Q_0 and X_0 vector (initialization vectors). For this section, we also assume that the decoding is done correctly in order to decrypt. The deccoding process is outlined below:

- The initialization vector is fed into the expansion function g to produce error vector Z_i
- The vector (Z_i) is XORed with the ciphertext C_i to produce Q_i.
- $Q_i$is multiplied by the transpose of the permutation matrix P to produce b_i.
- b_i is decoded into X_i
- The inverse of the nonlinear function f^{-1} is applied to X_i to produce d_i.
- d_i is XORed with Q_{i-1} to obtain the plaintext M_i.

The decryption process is shown mathematically in 20, 21, 22, 23, 24 and 25 The block diagram representing the decryption process is shown in Fig. 3.2.

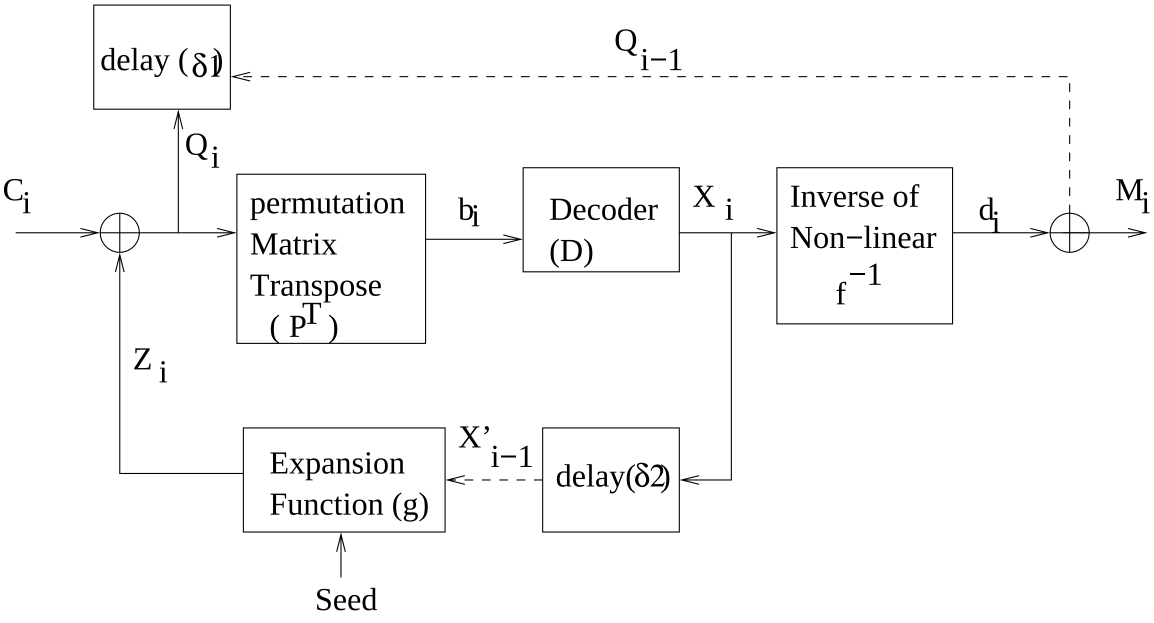

Figure 3.2. Block diagram of the proposed ECBC decryption scheme

To show the decryption process in a noiseless channel, let the received ciphertext be C_i (assuming no error due to the channel).

$$C_i = f(M_i + Q^*_{i-1})\ GP + Z_i \tag{20}$$

Applying the decryption process to 19

$$\begin{aligned} Q_i &= [f(M_i + Q^*_{i-1})\ GP + Z_i] + Z_i \\ &= f(M_i + Q^*_{i-1})\ GP \end{aligned} \tag{21}$$

Multiplying with the transpose of the permutation matrix

$$\begin{aligned} b_i &= [f(M_i + Q^*_{i-1})\ GP]P^T \\ &= f(M_i + Q^*_{i-1})\ G \end{aligned} \tag{22}$$

Applying the decoding algorithm to b_i depending on the code employed

$$X_i = f(M_i + Q^*_{i-1}) \tag{23}$$

Applying the inverse of the non-linear function f^{-1}

$$d_i = M_i + Q^*_{i-1} \tag{24}$$

Adding the error vector $Q{*}_{i-1}$ to d_i

$$\begin{aligned} d_i + Q^*_{i-1} &= M_i + Q^*_{i-1} + Q^*_{i-1} \\ &= M_i \end{aligned} \tag{25}$$

M_i is the message block i

For the case of noisy channel with error vector Z_c due to the channel, we assume that the Z_c is within the error correcting capability of the code. The received ciphertext with the channel error is

$$C_i = C_i + Z_c \tag{26}$$

From Eqn. 26 we know that:

$$C_i = [f(M_i + Q^*_{i-1})GP + Z_i] + Z_c \tag{27}$$

Applying the decryption process (we use $Q^{'}_i$ because of the effect of the channel error)

$$\begin{aligned} Q^{'}_i &= [f(M_i + Q^*_{i-1})GP + Z_i] + Z_c + Z_i \\ &= f(M_i + Q^*_{i-1})\ GP + Z_c \end{aligned} \tag{28}$$

Multiplying with the transpose of the permutation matrix

$$\begin{aligned} b_i &= [f(M_i + Q^*_{i-1})GP + Z_c]P^T \\ &= f(M_i + Q^*_{i-1})\ G + Z_e P^T \end{aligned} \tag{29}$$

Note: P^T does not change the weight of Z_e. Let W_H represent the hamming weight, hence

$$W_H(Z_e) = W_H(Z_e P^T) \tag{30}$$

Applying the decoding algorithm to b_i

$$X_i = f(M_i + Q^*_{i-1}) \tag{31}$$

Applying the inverse of the non-linear function f^{-1}

$$d_i = M_i + Q^*_{i-1} \tag{32}$$

Adding the error vector Q^*_{i-1} to d_i

$$\begin{aligned} d_i &= [M_i + Q^*_{i-1}] + Q^*_{i-1} \\ &= M_i \end{aligned} \tag{33}$$

The error-correction ability of the code is fully preserved for possible channel errors because error introduced intentionally at the sender can be removed because of synchronization of the initialization vector. Hence error due to the channel can be removed. In summary, the decryption process is shown in Fig. 3.2 and expressed mathematically:

$$D((C_i + Z_i)P^T)f^{-1} = M_i \tag{34}$$

where $Z_i = \mathrm{g}(X_{i-1})$

$$\begin{aligned} D((C_1 + Z_i)P^T) &= X_i \\ f^{-1}(X_i) + Q_{i-1} &= M_i \end{aligned} \tag{35}$$

In this scheme, errors due to intruders tampering which cannot be removed by the error-correcting code (error above the error correcting capability of the code) will propagate to the later blocks due to the block chaining technique. Hence, this scheme could be used as a checksum to detect illegal tampering or modification [34]. However, the transmitter will have to resend the data if the error-correcting code cannot correct the modification. Based on this features, ECBC does not only provide error detection and correction, but also data integrity.

3.2. Cryptanalysis of ECBC Scheme 1

Cryptanalysis will be more difficult because the same plaintext block will be encrypted to different ciphertext. The cryptanalyst cannot construct an equivalent generator matrix combinatorially [22], since the ciphertexts are not codewords, as a result, errors cannot be corrected systematically. We analyze the security that this scheme provides in this section.

In a case where X_i is fed forward and Q_{i-1} is not fed back, then the encryption process can be expressed as

$$C_i = f(M_i)GP + Z_i \quad (Z_1 = g(IV_2)) \tag{36}$$

$$C_{i+1} = f(M_{i+1})) \; GP + \mathrm{g}(\mathrm{f}(M_i)) \tag{37}$$

$$C_{i+2} = f(M_{i+2})) \; GP + \mathrm{g}(\mathrm{f}(M_{i+1})) \tag{38}$$

$$C_{i+3} = f(M_{i+3})) \; GP + \mathrm{g}(\mathrm{f}(M_{i+2})) \tag{39}$$

$$C_i = f(M_i)) \; GP + \mathrm{g}(X_{i-1}) \tag{40}$$

A chosen plaintext attack will break GP if the expansion function g is a linear function that has a left inverse based on the equations. To see this, let $M_i = M_{i+1}$, and $M_{i+2} = M_{i+3}$, then

$$C_{i+1} + C_{i+2} = f(M_{i+1}) + f(M_{i+2})GP \tag{41}$$

$$C_{i+2} + C_{i+3} = g(f(M_{i+1})) + g(f(M_{i+2})) \tag{42}$$

If g is linear

$$g(f(M_{i+1}))+g(f(M_{i+2})) \quad = \quad g(f(M_{i+1}))+f(M_{i+2}) \tag{43}$$

From Eqn. 43,

$$f(M_{i+1}) + f(M_{i+2}) = g^{-1}(C_{i+2} + C_{i+3}) \tag{44}$$

GP can be derived if the cryptanalyst could obtain *k* such distinct pairs. However, *GP* is a permutated version of *G* which increases the work factor of deriving *G*. This is one of features that differentiates the ECBC schemes from previous schemes. Also, if *g* is a secret nonlinear function, then this attack will not work at all and ECBC uses *g* as a nonlinear function.

We analyze the case where Q_i is fed back and

$$C_1^2 \text{ - } C_2^2 = (m_1^2 - m_2^2)\text{G'} + E_1 \text{ - } E_2 \tag{45}$$

X_i is not fed forward. The encryption sequence is shown below:

$$\begin{aligned}
C_1 &= (f(M_1 + Q_0)GP, \quad Q_0 = IV1 && (46)\\
C_2 &= f(M_2 + Q_1)GP && (47)\\
where\ \ Q_1 &= f(M_1\ + Q_0)GP) && \\
 & && (48)\\
C_3 &= (f(M_3 + Q_2)GP && (49)\\
 &. && \\
C_i &= f(M_i + Q_{i-1})GP && (50)\\
where\ \ Q_{i-1} &= f(M_{i-1}\ + Q_{i-2})GP) &&
\end{aligned}$$

The cryptanalyst would have to search for equivalent ciphertexts where $C_i = C_j$, as a result, $f(M_i + Q_{i-1}) = f(M_j + Q_{j-1})$ which means that $Q_i = Q_j$. If f is a linear transformation, then $C_{i+1} + C_{j+1} = f(M_{i+1})GP + f(M_{j+1})GP$, As a result fGP can be figured out by a known plaintext attack. However if f is a nonlinear transformation, the line of attack will not work. The cryptanalyst can collect k linearly independent equivalent codewords to construct G'=fGP which is combinatorially equivalent to G. It will be computationally infeasible to estimate the matrix G if k is large enough.

The ECBC scheme withstands chosen-plaintext attacks [50] because of the non-linear function f that transforms the plaintext. As a result, the cryptanalyst cannot construct unit vectors from chosen plaintext to construct the G.

Since we employed the LDPC code in this scheme, the complexity based on the random LDPC code used could be determined for an (256,3,6) ldpc code. Based on the algorithm used [51] the number of ways [18] of placing three ones in the first column could be expressed as:

$$C_3^{128} = 341376 \tag{51}$$

The number of ways of placing three ones in the second column is

$$C_3^{128} - C_2^3 - C_3^3 = C_3^{128} - 4 \tag{52}$$

$$C_3^{128} - 2 * (C_2^3 - C_3^3) = C_3^{128} - 8 \tag{53}$$

We can then express a generalized equation for the number of ways of placing ones in the Nth column as

$$C_3^{128} - (N-1) * (C_2^3 - C_3^3) = C_3^{128} - 4 * (N-1) \tag{54}$$

The time complexity that that the attacker quesses the H matrix of a random LDPC code is therefore 10^{3289} which is enormous. This value increases exponentially for a (N, j, k) LDPC code as N increases. This shows the advantange of the random LDPC code utilized in this system.

3.3. ECBC Scheme 2

In this section, we present a private key algebraic based cryptosystem that is based on McEliece Public Key Cryptosystem (MPKC). The cyptosystem is also based on Nordstrom Robinson code which is a non linear code that was first published by Nordstrom and Robinson in 1967 [35]. It has twice the number of codewords of any linear code with the same minimum distance (6) and length (16). The code had never been used because of the complexities involved in encoding and decoding because of absence of algebraic structure due to it non-linearity [26]. However, it is now known that the Nordstrom Robinsin (NR) code is the binary image of the octacode and the octacode is a linear code. It is defined as the extended cyclic code over Z_4 [19]. Based on this, Nordstrom-Robinson code could now be used as though they are linear. The non-linear binary code in Z_2 could be mapped into into a linear octacode in Z_4 [15]. The octacode is defined as an extended cyclic code over Z_4 and a code of length 7 can be generated by $g(X) = X^3 + 3X^2 + 2X + 3$ with the addition of an overall parity check to make the coordinates sum to zero, an octacode of length 8 could be generated over Z_4 [19]. Based on [15], [26], the generator (G) and parity check (H) matrices for the octacode (O_8) is derived and shown in Eqn. 55 and Eqn. 56. The code generated has 256 (4^4) codewords and is systematic.

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 3 & 1 & 2 & 1 \\ 0 & 1 & 0 & 0 & 1 & 2 & 3 & 1 \\ 0 & 0 & 1 & 0 & 3 & 3 & 3 & 2 \\ 0 & 0 & 0 & 1 & 2 & 3 & 1 & 1 \end{pmatrix} \tag{55}$$

$$H = \begin{pmatrix} 1 & 3 & 1 & 2 & 1 & 0 & 0 & 0 \\ 3 & 2 & 1 & 1 & 0 & 1 & 0 & 0 \\ 2 & 1 & 1 & 3 & 0 & 0 & 1 & 0 \\ 3 & 3 & 2 & 3 & 2 & 0 & 0 & 1 \end{pmatrix} \tag{56}$$

Since Nordstrom Robinson code is a binary image of the octacode, we take advantage of the mapping from Z_2 to Z_4 and vice-versa in this scheme 2. For example, if Alice wants to send a message to Bob, we can change the symbol mapped to each bit when changing from Z_2 to Z_4. If Eve does not know the mapping, more errors will be introduced into the received data. In other words, instead of taking advantage of the octacode, Eve will have to decode the message using the complicated decoder which might not work depending on the number of errors introduced into the message. In these scheme, Bob can take advantage of decoding the linear code while Eve will have to decode a non-linear code. Based on Wyner [57], the communication channel of Eve would have been degraded compared to that of Bob.

In order to encode and encrypt using the Norstrom-Robinson Code (Scheme 2), Alice would follow the following method:

(i) The message m in Z_2 is multiplied with a matrix called the scrambler (S) to obtain m'.

(ii) The scrambled message m' in Z_2 is mapped into m'' in Z_4 based on a key (k).

(iii) The m'' in Z_4 is then multiplied with the Nordstrom-Robinson Generator Matrix (G) in 55 to obtain codeword C''.

(iv) The C'' is then mapped back into Z_2 to obtain a different version of the codeword C using a key k.

(v) The codeword C is then multiplied with a permutation matrix (P) to obtain C'.

(vi) An Error vector (E) is then added to the permuted codeword (C') to obtain ciphertext (CT).

(vii) The ciphertext CT is then transmitted through the communication channel.

In order to decode and decrypt the received data, Bob would follow the following method:

(i) The received ciphertext (CT) is multiplied with the transpose of the permutation matrix to obtain CT'. This does not tamper with the weight of the error.

(ii) The CT' in Z_2 is mapped to Z_4 using a key k to obtain CT''.

(iii) The error vector (EP^T) in CT'' is removed using syndrome decoding introduced in [26] to obtain m''.

(iv) The m'' is mapped to Z_2 (C) from Z_4 using key k to obtain m'.

(v) The m' is multiplied with the scrambler to obtain m.

We can express the scheme 2 mathematically as shown in 57 where m_2 represent message in Z_2:

$$m_2 = ((m_2S)_4G)_2P + E \tag{57}$$

3.4. Cryptanalysis of ECBC Scheme 2

In terms of security, in order to provide higher security system, we keep S, G, and P private. The cryptanalyst could try to attack this scheme using known-plaintext attack by independently solving matrices for each column of the product SGP (private key). However this requires large pairs of message and ciphertext. This will be very difficult because of the periodic randomized mapping from Z_2 to Z_4. In order to prevent chosen plaintext attack, the multiplication of the message with the scrambler should be replaced with a non-linear function. Hence even if the cryptanalyst tries to use messages with hamming weight of 1, the attack will not succeed. A Chosen-plaintext attack with the previous scheme would look like this:

Assuming the message (m), permutation matrix (P) and the scrambler (S) are all in (Z_2) (however, in our scheme, G is in Z_4). The cryptanalyst can choose a plaintext m = (0000..00100..0) or can choose pairs of plaintext that differ by 1 as illustrated in Eqn. 58:

$$m_1 \text{ - } m_2 = (0000..00100..0) \tag{58}$$

if ciphertext 1 and 2 in Z_2 are expressed in Eqn. 59 and Eqn. 60:

$$C_1^2 = m_1^2 SGP + E_1^2 \tag{59}$$

$$C_2^2 = m_2^2 SGP + E_2^2 \tag{60}$$

if SGP = G' and a superscript of 2 means the data is in Z_2.

Then

$$C_1^2 \text{ - } C_2^2 = (m_1^2 - m_2^2)G' + E_1 \text{ - } E_2 \tag{61}$$

$$C_1^2 \text{ - } C_2^2 = (0000..00100..0)G' + E_1 \text{ - } E_2 \tag{62}$$

$$C_1^2 \text{ - } C_2^2 = g' + E_1 \text{ - } E_2 \tag{63}$$

where g' is the ith row vector of G'. The Hamming weight of the E_1 - E_2 will be at least 2t. The cryptanalyst can consider $(C_1 - C_2)$ as an estimate of g_1' if t ¡¡ n. The cryptanalyst can then use majority voting to determine for each position which ultimately give G'. However in our case, we use a non-linear function (f) in place of the scrambler as shown below:

(64) $$C_1^2 = (f(m_1^2)^4 G^4)_2 P^2 + E_1^2$$

(65) $$C_2^2 = f(m_2^2)^4 G^4)_2 P^2 + E_2^2$$

a superscript of 4 means the data is in Z_4. It is important to note that G' cannot just be factorized as in 61 since the function f is not linear as the previous case. Even if pairs of messages are different by 1, obtaining $C_1 - C_2$ will not result in an estimate of G. Also an estimate of G' will be in Z_2 which means the linear properties that exist in Z_4 does not apply. If the cryptanalyst will have to map the estimated G' into Z_4 which is based on a private seed.

Then

(66) $$C_1^2 - C_2^2 \neq (0000..00100..0)G' + E_1 - E_2$$

The number of choices for the permutation matrix p is n!. An exhaustive search for the S and P is considered hopeless. The attacks applied to linear code cryptosystems by [43] will not work because they depend on the linearity of the code. As a result they will fail with the use of non-linear code. The struik and Tilburg attack is applied to the system, as previously shown, the differences in error vector could be calculated but the technique will fail since the code used is non-linear. Since each of the methods suggested above fails due to non-linearity of the code or because of large work factor, the joint scheme appears to be secure.

CHAPTER 4

AES-MCELIECE HYBRID CIPHER (AMHC)

The AMHC is the third joint scheme presented in this research. Many researchers have presented joint schemes that are McEliece based. Even though these schemes were more efficient than AES based systems, most of them suffered from degradation in security due to use of simple codes. Most of them possesed tradeoff between security and error correction capability of the joint schemes. AES based schemes presented in the past involved the construction of new codes that resulted in increase in the complexity of the entire system. AES-like schemes have been known to have strong security based on the fact that they are rooted in the structure of the Advanced Encryption Standard. In order to take advantage of both schemes, we present a hybrid scheme that combines both AES-like and McEliece-like structure. The cipher is derived from AES and McEliece-like cryptosytem. The AMH cipher is an iterative algorithm that is made up of 6 rounds. The cipher takes 128-bit plaintext block and produce a 256-bit ciphertext block. The 128-bit plaintext block is divided into 16 bytes to form a 4 by 4 array called a state [23],[59],[2]. The AMH cipher uses a 128-bit AES key block. Each round of this cipher is broadly made up of Key Addition layer, Non-Linear (NL) layer and Linear (L) layer. The plaintext block is added with the initial key before been fed to the Non-linear layer of the AMH cipher as shown in Fig. 4.1. The key addition, NL layer and the linear layer form a round.

4.0.1. *Key Addition*

The first operation performed on the plaintext is the key addition operation as shown in Fig. 4.1. In this layer, the 128-bit initial key or expanded key is XORed with 128-bits of the plaintext. The key expansion algorithm of AES is used in this cipher.

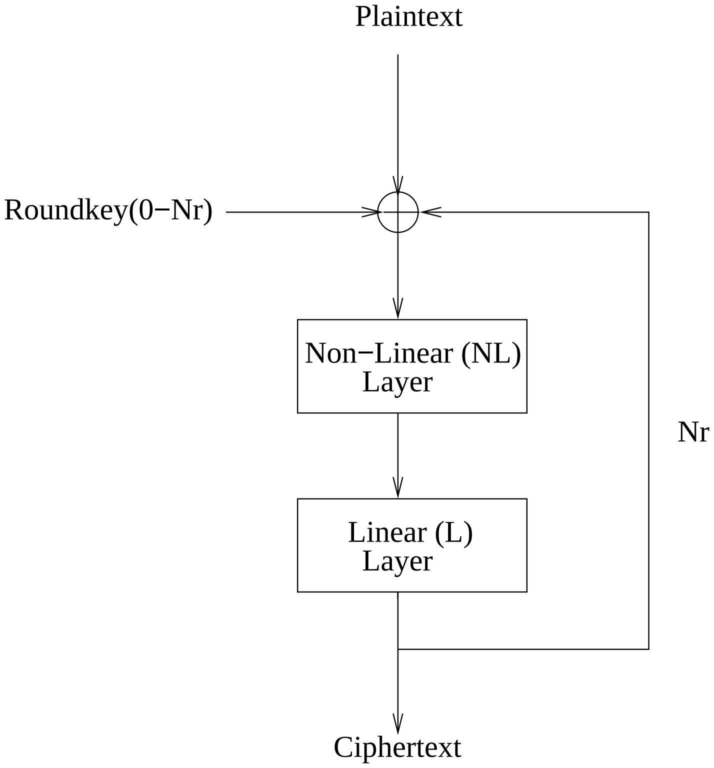

Figure 4.1. High Level Block Diagram of the AMH Encryption Scheme

4.0.2. *Non-Linear Layer*

This layer is responsible for the non-linear transformation of the states in the cipher. This layer ensures that the maximum input-output correlation amplitude and the difference propagation probability is as small as possible. The S-Box of Rijndael algorithm is used for this layer. The S-Box has maximum correlation amplitude of 2^{-3} and maximum difference propagation probability of 2^{-6} [13]. This values makes the sbox resistant to difference and linear cryptanalysis. The author in [36] further showed that the S-box function is highly non-linear. The S-Box is made up of the multiplicative inverse and invertible affine mapping operations. The mathematical representation of this layer is shown in Eqn. **??**.

4.0.3. *Linear Layer*

This layer carries out linear transformation on states of the cipher. The linear layer is made up of Shift-Gen and mix-perm operations. In the Shift-Gen operations, the ShiftRow operation of Rijndael algorithm is carried out for rounds 0 to Nr-1, where the rows of the states are shifted by their row numbers. In the last round, the state is encoded with the Low Density Parity Check code as illustrated in Fig. 4.2. This layer has to be diffusion optimal. The shiftrow operation of the AES was designed to be diffusion optimal [13] which makes the cipher resistant to truncated differential and saturation attacks. The Mix-Perm operation on the other hand is comprised of the MixColumn operation in Rijndael algorithm and permutation transformation. The MixColumn transformation is performed on a state for round 0 to Nr-1 and the result is multiplied by a permutation matrix (PM) in the last round in order to permut the state. The permutation operation has been shown to be diffustion optimal in [32]. The Mix-Perm block diagram is shown in Fig. 4.3. Based on Fig. 4.4, the AMH cipher is the combination of the first 5 rounds of Rijndael algorithm and the McEliece-like cryptosystem. This is illustrated in Fig. 4.5.

The AMH cipher is illustrated in Fig 4.4

Since diffusion property is very important in the resistance to linear and differential crypanalysis, we analyse the diffusion property of the ldpc encoding used in the shift-Gen layer. According to C. Shannon [49], diffusion is the quantitative spreading of information. Diffustion is the property of a function (cipher component) where the statistical property of the plaintext space is spread into the long range statistics of the output (ciphertext). It is therefore desired that the dependency on each plaintext bit is spread to as many cipher bits as possible. In the cipher presented, we examine the propagation ratio (R_p) [51] of the first Nr-1 rounds. Considering the byte propagation in Fig. 4.6 for the first Nr-1 rounds of the AMH cipher, the propagation ratio is 4^{Nr-1} where Nr is 7 for AMH cipher. From Fig. 4.7,

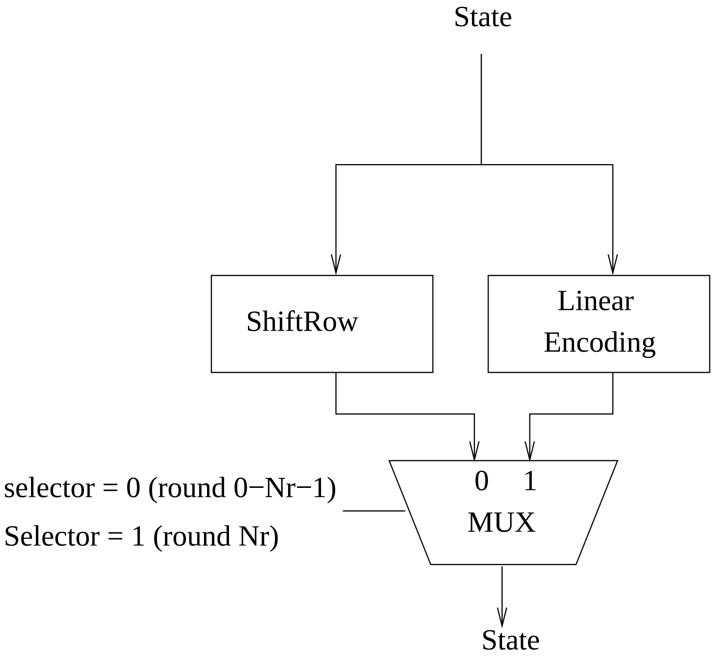

Figure 4.2. Block diagram of the Shift-Gen transformation

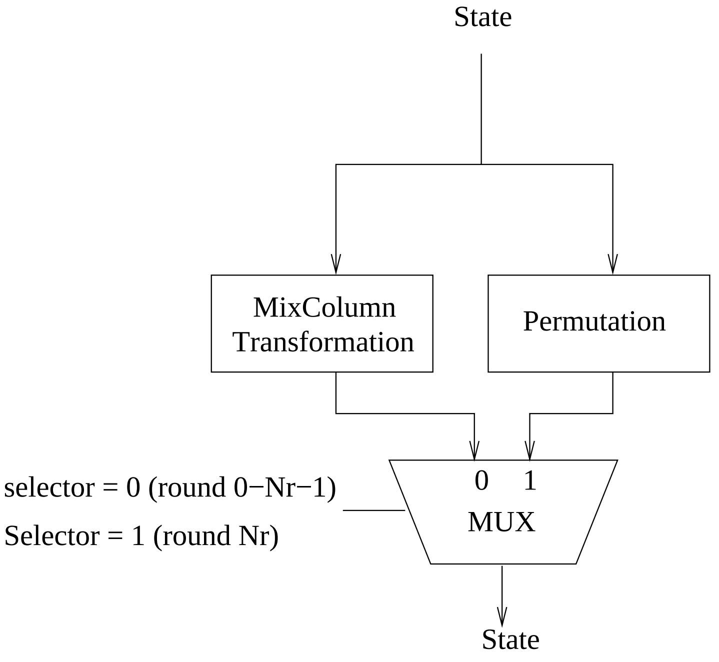

Figure 4.3. Block diagram of the Mix-Perm transformation

the propagation ratio of the last round of the AMH cipher is $128 > 4^3$. The propagation ratio of the entire AMH cipher is therefore $R_p > 4^6 * 4^3$. The propagation ratio of 10-round

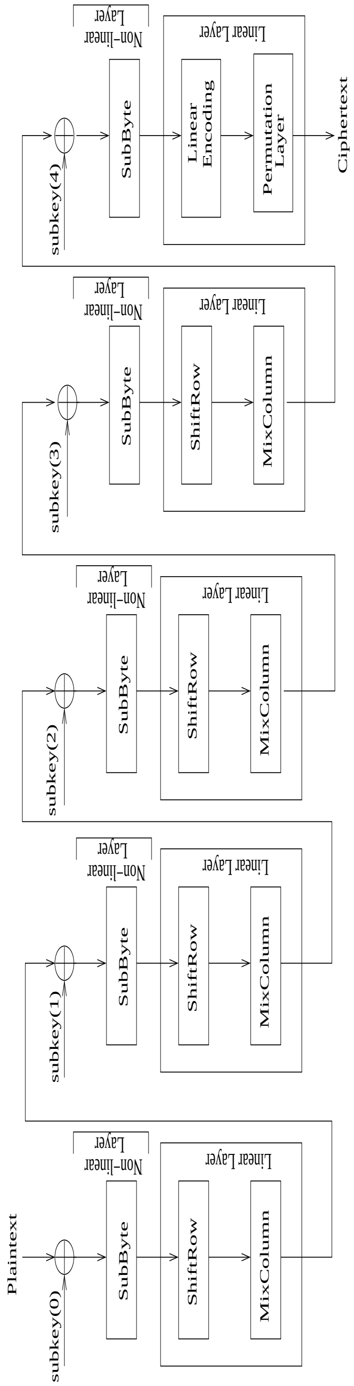

Figure 4.4. Block Diagram of the AMH Cipher

AES is 4^9, while the propagation ratio of the 7-round AMH cipher is greater than 4^9. As a result, the AMH cipher is more secure and more efficent than AES since it $R_p > 4^9$ and it takes less round to obtain the ciphertext than the AES.

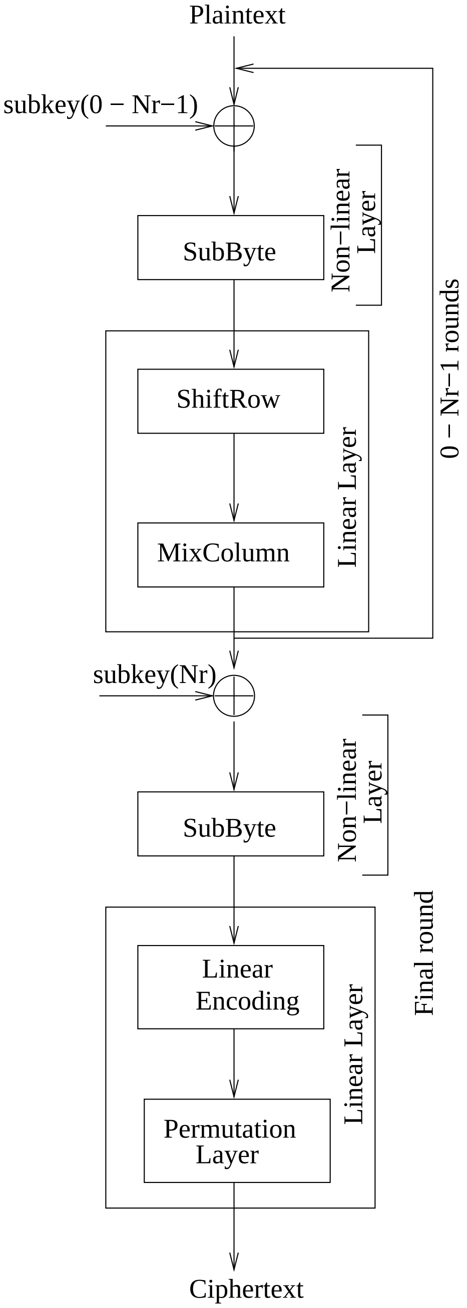

Figure 4.5. Block diagram of the AMH Encryption Scheme

For the decryption scheme of AMH cipher, the 256-bit ciphertext is multiplied with the permutation matrix and the result is decoded using linear decoding. In the case of AMH, LDPC decoding is employed. The permutation and linear decoding both form the linear layer of the AMH cipher. The inverse subByte is then applied to the output of the linear layer.

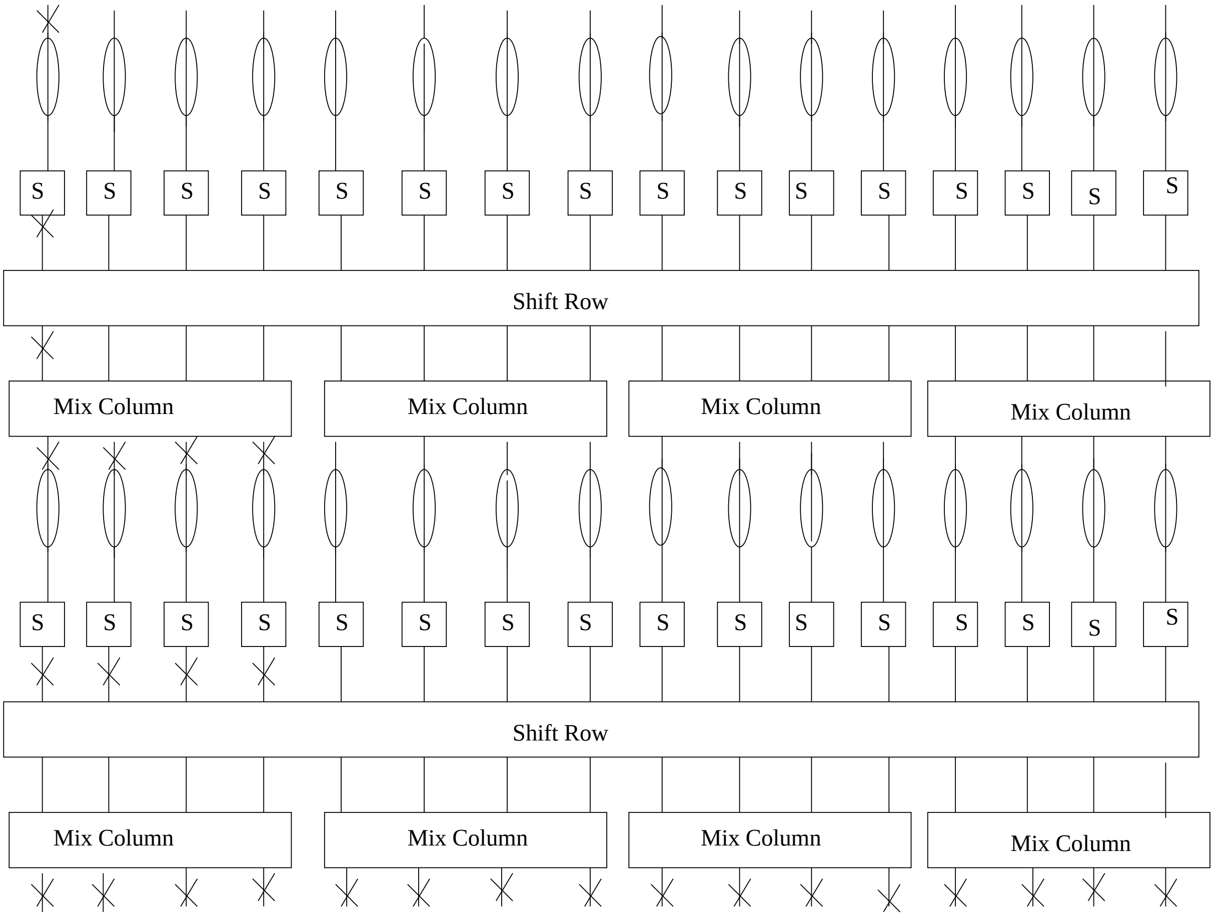

Figure 4.6. Active Byte Propagation in the Wide Trail Strategy

The result after the inverse subByte is XORed with Nr^{th} subkey. The inverse MixColumn operation is performed on the state and then Inv_ShiftRow transformation is applied to the resulting state. The inv_SubByte transformation is applied on the state and that is the non-linear layer. The Inv_MixColumn, Inv_shiftRow and Inv_SubByte tranformations are repeated Nr times on the states and the Nrth state is mixed with subkey(0) to produce the 128-bit plaintext.

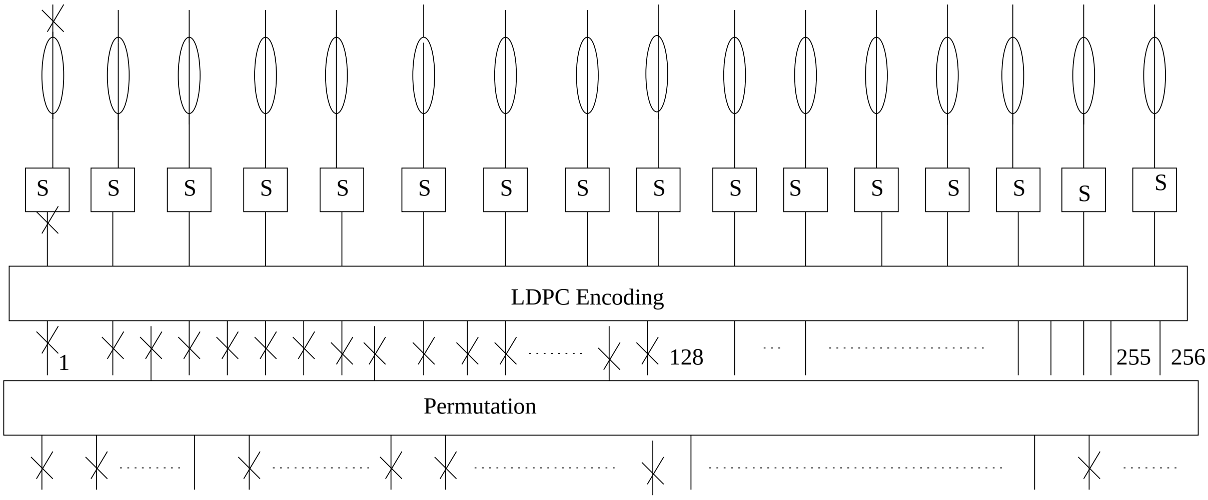

Figure 4.7. Bit Difference Propagation in the Last Round of AMH Cipher

4.1. Cryptanalysis

4.1.1. *Resistance to Differential and Linear Cryptanalysis*

A differential cryptanalysis is a chosen plaintext-ciphertext attack that utilizes the difference propagation property of a cipher to obtain the key bits. The statistical key information is deduced from the ciphertext blocks obtained by encrypting pairs of plaintext blocks with a specific bitwise difference under the target key. The difference propagation property of S-box is the relative number of all input pairs where for a given input difference results in a specific output difference. For example, consider a system with input X = [X1 X2 ... X_n] and output Y = [Y1 Y2 ... Y_n]. Let two intermediate inputs (states) to the cipher be X' and X" with the corresponding outputs Y' and Y", respectively. The input difference is given by ΔX = X' $\oplus$ X" where $\oplus$ represents a bit-wise exclusive-OR. A non-zero ΔX is called active byte. The concatenation of difference propagation of consecutive rounds across several rounds results in differential trail. It is possible for differential crypanalysis to break the cipher with complexity less than O(2^{-128}) if the the maximum possible propagation ratio over all rounds is significantly greater than 2^{-127}. Linear Cryptanalysis is a known-plaintext attack where a large number of plaintext-ciphertext pairs are used to determine the value of key bits. The

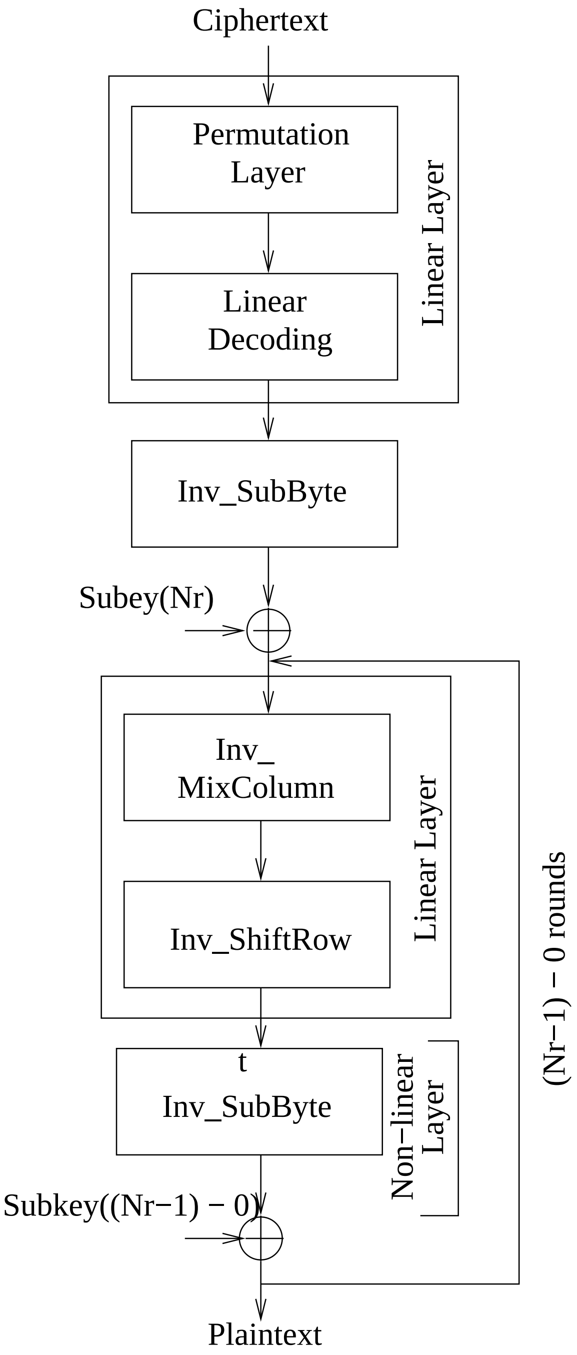

Figure 4.8. Block Diagram of the AMH Decryption Scheme

linearity of an active S-box used in the non-linear layer can be expressed as the maximum input-output correlation exhibited. The S-box used in the non-linear layer is the same as that of the AES and it has a maximum propagation ratio of 2^{-6} and a maximum input-output correlation of 2^3. As a result, the propagation ratio of the first 6 rounds is $2^{-6*(6*4)} = 2^{144}$ which is greater than 2^{-127}. This means that the complexity of the cryptanalysis will be greater than $O(2^{128})$. For the last round, a 128 x 256 LDPC generator matrix is employed whose propagation measure is 128. It is important to note that linear cryptanalysis can break the cipher with complexity less than $\mathrm{O}(2^{128})$ if the maximum possible correlation over all the rounds is significantly larger than 2^{-64}. Since the maximum input and output correlation for the S-box is 2^{-3}, then the maximum input-output correlation for the first 6 rounds of the AMH cipher is $2^{(4*6*-3)} = 2^{72}$ which is less than 2^{-64}. Combining this with that of the last round means that the input-output correlation will be far less than 2^{-64}. We can then conclude that the complexity of the linear cryptanalysis is greater than $\mathrm{O}(2^{128})$.

The complexity based on the random LDPC code used could be determined. Based on the algorithm used [51] the number of ways [18] that three ones could be placed in the first column could be expressed as:

$$C_3^{128} = 341376 \tag{67}$$

The number of ways of placing three ones in the second column is

$$C_3^{128} - C_2^3 - C_3^3 = C_3^{128} - 4 \tag{68}$$

$$C_3^{128} - 2*(C_2^3 - C_3^3) = C_3^{128} - 8 \tag{69}$$

We can then express a generalized equation for the number of ways of placing three ones in the Nth column as:

$$C_3^{128} - (N-1) * (C_2^3 - C_3^3) = C_3^{128} - 4 * (N-1) \tag{70}$$

The time complexity that that the attacke quesses the H matrix of a random LDPC code is therefore 10^{3289} which is greater than 2^{128}, which shows the advantange of the random LDPC code utilized in this system.

4.1.2. *Resistance Against Square Attack*

A square attack [12] is a chosen plaintext attack that takes advantage of byte-oriented cipher and has been applied to reduced version of AES. However, since only AMH's first 6 round are byte-oriented while the final round is bit-oriented, the application of the square attack to the AMH cipher will not be successful.

4.2. Radomization Test

We heuristically tested the AMHC by testing for randomness in the output. The AMHC was used as a pseudorandom number generator in counter mode. The TestU01 [27] was used to test the randomness of the output of AMHC in counter mode. We tested for p-values within the boundary $[10^{-4}, 1 - 10^{-4}]$. Any p-values lying outside this range is considered as failure, while the ones within the range is considered as pass. Table 1 lists the test suites, the number of tests in each suite and the results. The standard parameter refers to the inbuilt test parameters of test suites. A total of 319 tests were carried out and the AMHC passed all of them. It is important to point out that because the scheme passes the test of randomization, is not a guarantee that such a scheme is secure. However, it is important that a cryptographic scheme's output should be random. The result of randomization test using TestU01 is shown in Table 5.1

Table 4.1. Result of Test Suites on AMHC

Test Suite	Parameters	No. of Statistics	Results
Small Crush	Standard	15	Pass
Crush	Standard	144	Pass
Big Crush	Standard	160	Pass

CHAPTER 5

ARCHITECTURE AND IMPLEMENTATION OF ECBC

In this chapter, we present the architecture and implementation of the Error Correction Based Cipher I for prove of concept. The schemes presented in this research was implemented in software and hardware. To our knowledge, we are the first to implement a joint scheme for error correction and security in hardware. This helps to show the efficiency of a McEliece-like scheme compared to a non-McEliece scheme. In order to show that there is no degradation in the performance of the underlying error correction code in the scheme, we present the BER-SNR curve and this shows that the performance is not degraded as a result of the joint scheme. In addition, we also show results that the output of the ECBC scheme is random. The output of ECBC I is subjected to a battery of randomization test. Earlier McEliece-like schemes do not pass all these randomization based on our experimentation. The architecture of the ECBC scheme 1 for encryption is shown in Fig. 5.1. The shift register contains received stream data. Each block of data is shifted into the k-bit buffer. The output of the buffer (message block) is randomized with the output of the multiplexer MuxA through an XOR gate. The inputs to the multiplexer is a random Initial Vector (IV) and a delayed version of the permutated encoded data. The control unit outputs the selector (selA) for the multiplexer as shown in Fig. 5.1.

The output from the XOR gate is fed into the SBOX unit. The SBOX [13] represents the nonlinear function *f*. This is a function that computes the multiplicative inverse of each input byte of the state in GF (2^8) followed by affine transformation. It is a non-linear byte substitution and it is composed of two transformation:

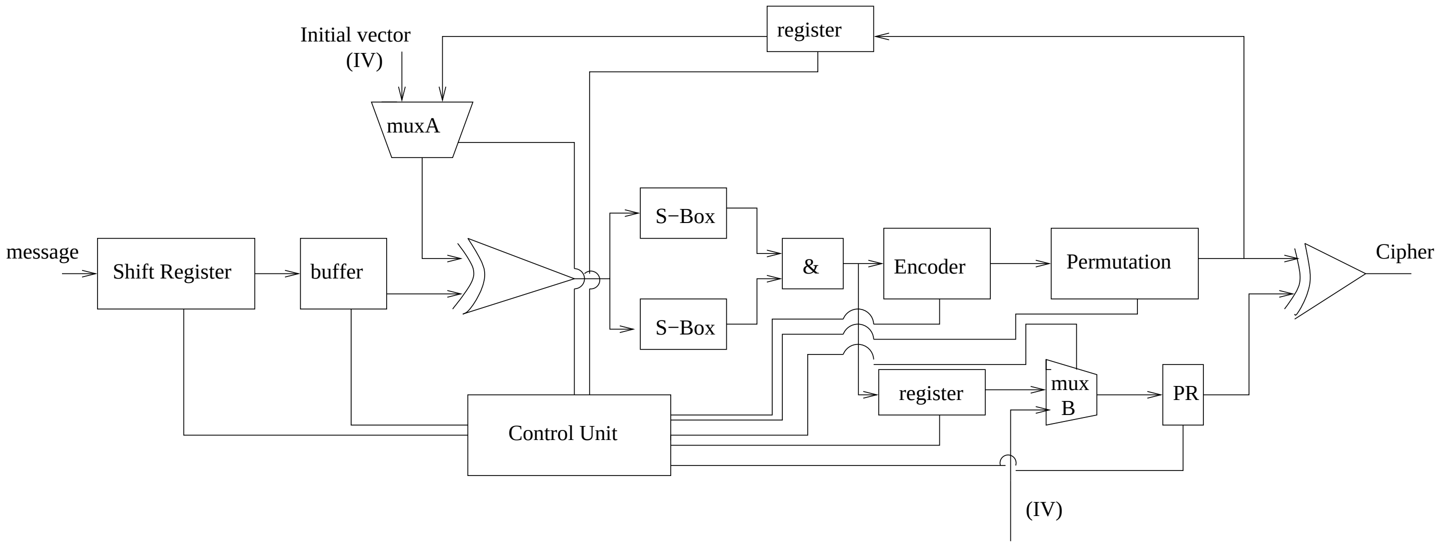

Figure 5.1. Architecture of proposed Error Correction Based Cipher

- multiplicative inverse in GF(2^8): This is the mapping of x → x^{-1}, where x^{-1} is the multiplicative inverse.
- Affine transformation over GF(2): x → Ax + b, where A and b are constants.

We implement the S-box with a multiplexer and lookup tables. In our implementation, given an n-bit input into the *f* function, n/8 s-boxes are applied to the n/8 bytes of data that makes up the input. Each of the n/8 bytes from the different s-boxes are substituted by the corresponding element in the s-boxes. Each of the byte output from each S-Boxes are then concatenated together to form a vector. The & sign is used to represent the concatenation unit. The architecture of the SBOX is shown in Fig. 5.2.

The concatenated output from the SBOX is encoded using the generator matrix (G) of Low Density Parity Check Code (LDPC). LDPC codes are linear block codes. They are codes that have received major attention in recent years because of their excellent performance and error correction capability. We used LDPC code because it has good diffusion property. It has good linearity relationship between code length and the minimum weight/code distance. The random LDPC code has higher security than QC LDPC codes [51]. An (n, k) LDPC code has *k* information bits and n codeword bits with code rate r = k/n. The parity check matrix *H* has a dimension of (n-k) x n. LDPC encoding is based on the property:

$$uH^T = 0 \tag{71}$$

where *u* is the n-bit codeword bits and *H* is the parity check matrix. The parity check matrix *H* can be expressed as:

$$H = [H_1 \ H_2] \tag{72}$$

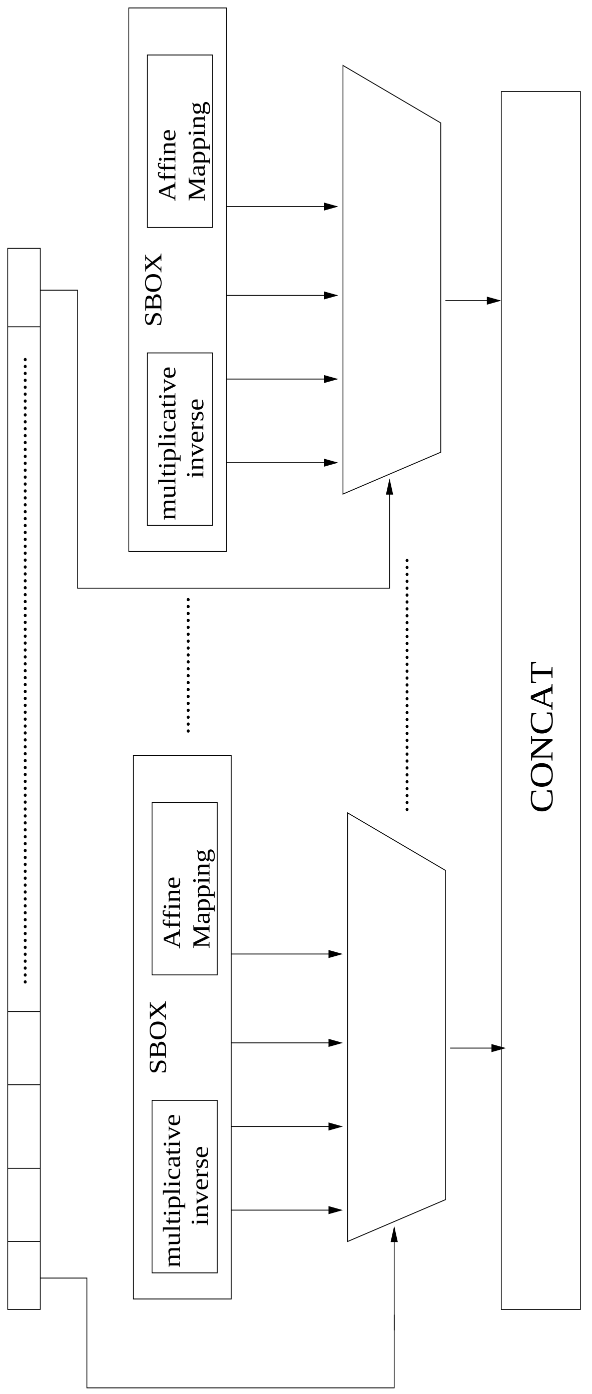

Figure 5.2. Architecture of S-Box

where H_1's dimension is (n-k) x (n-k) and H_2's is (n-k) x k. The information bit could be expressed as:

$$u = [p\ s] \tag{73}$$

where s is the k-bit information bits and p is the n-k parity bits. Based on Eq. 71:

$$H_1 \cdot p + H_2 \cdot s = 0 \tag{74}$$

since operation is in GF(2)

$$p = H_1^{-1} H_2 \cdot s \tag{75}$$

5.1. Encoder Unit

The architecture of the LDPC encoder is shown in Fig. 5.3. Each parity bit is obtained by matrix-vector multiplication of matrix H_2 with the output of the SBOX unit. Since matrix-vector multiplication operation is carried out in GF(2), Each row of matrix H_2 is ANDed with the vector output (s) from the SBOX. The outputs from the (n-k) AND gate are then xored together to produce the parity bit. Multiplication by H_1^{-1} is not not necessary if the H matrix is systematic. The codeword is reconstructed by concatenating the parity with the $kx1$ SBOX output with the aid of CodeWord Construction Unit.

5.2. Permutation Unit

The architecture of the permutation unit is shown in Fig. 5.4. The unit permutates the codeword output from the encoder unit. We attempt to explain the permutation unit in Fig. 5.4 in this section. Assuming we have a permutation matrix P shown below:

$$p = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{76}$$

The column numbers where 1s are located in the permutation matrix are stored instead of storing the 0s and the 1s. The row number (index) is used for referencing the column.

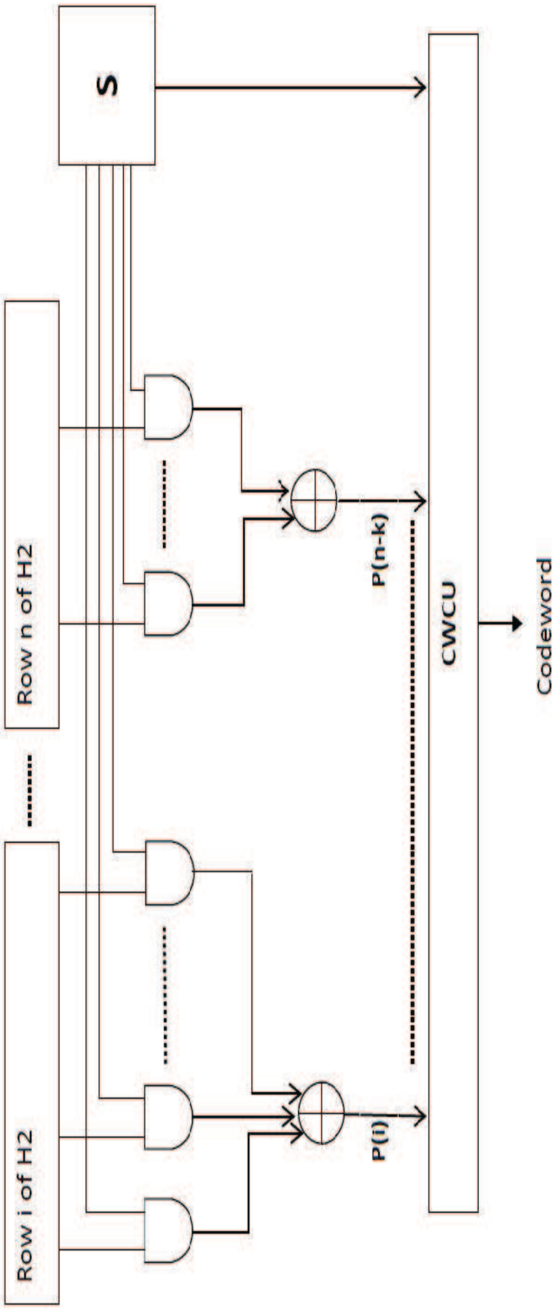

Figure 5.3. Architecture of the LDPC encoder unit

For example, from Fig. 5.4, for row 1 (index), 1 is located in column 2, in row 2, 1 is located in column 1, and row 3, 1 is located in column 3. The column numbers are used as selector for the multiplexer which in turn determines the output of the permutation unit. Each of the multiplexer is an n to 1 multiplexer.

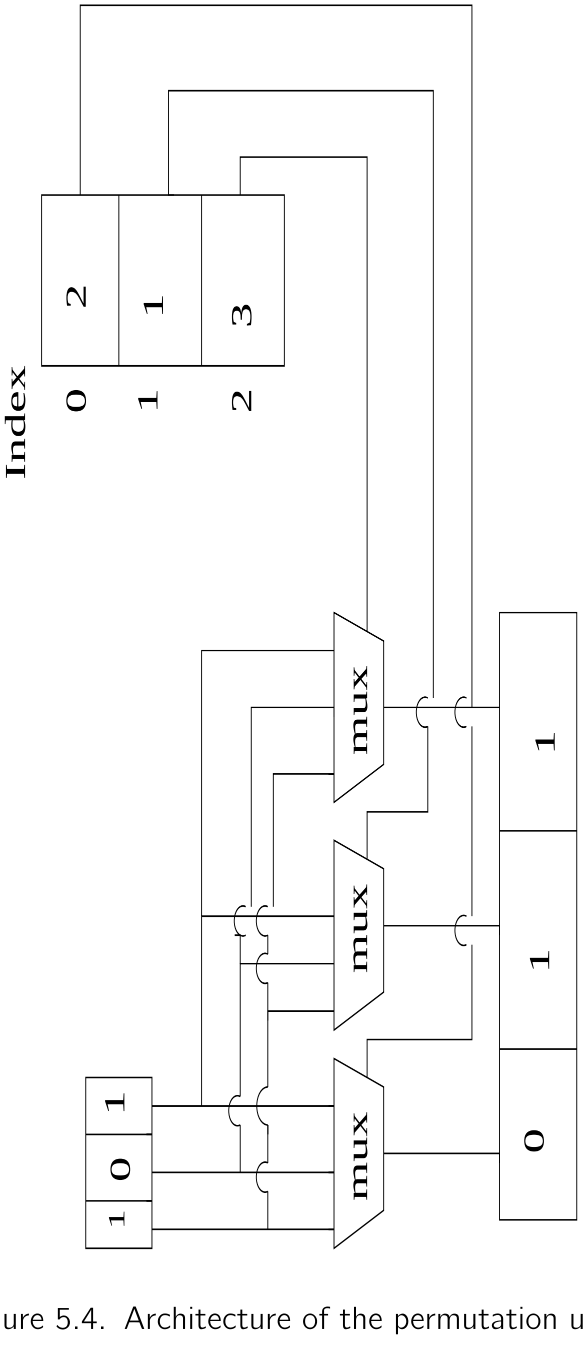

Figure 5.4. Architecture of the permutation unit

5.3. Control Unit

The control unit for the ECBC is modeled as a Finite State Machine (FSM) as shown in Fig. 5.5. The control unit has six states: Initial, Fetch, random, substitute, encode, permutation. The initial state is the first state after reset where zeros are written to all the registers. If start is asserted at the initial state, the present state becomes the fetch state. At the fetch state, a block of data is read into the buffer from the stream shift register. Control is transferred to the randomization state after the fetch state where the input block is randomized. The control unit also outputs the selector for MUXA at the random state. At the substitute state, control signal is sent to the the concat (&) unit in order to concatenate all the ouput block coming from the SBOX. In the encode state, codeword is generated by the encoder. The next state is the perm state where the codeword from the encoder is permutated. After perm state, an ecrypted data is produced and the present state becomes the fetch state where another block is fetched. The 5-staged piplelined architecture of the ECBC is shown in fig. 5.6. The figure include the 5 pipepline registers. This architecture helps to increase the throughput of ECBC.

5.4. Implementation and Result

The ECBC scheme was implemented in software for the purpose of verification and randomization test. The non-linear function f was implemented using s-box [13]. We used the generator matrix (G) of Low Density Parity Check (LDPC) Code for error correction. We used LDPC code because it has good diffusion property. It has good linearity relationship between code length and the minimum weight/code distance. The random LDPC code has higher security than QC LDPC codes [51]. The permuted output of the ECBC is xored with the output of a pseudorandom number. The key is the seed to a pseudorandom generator that generate a random sequence of bits that is xored with permutated codeword. In our case, KISS99 was used as a generator. We heuristically tested the ECBC by testing for

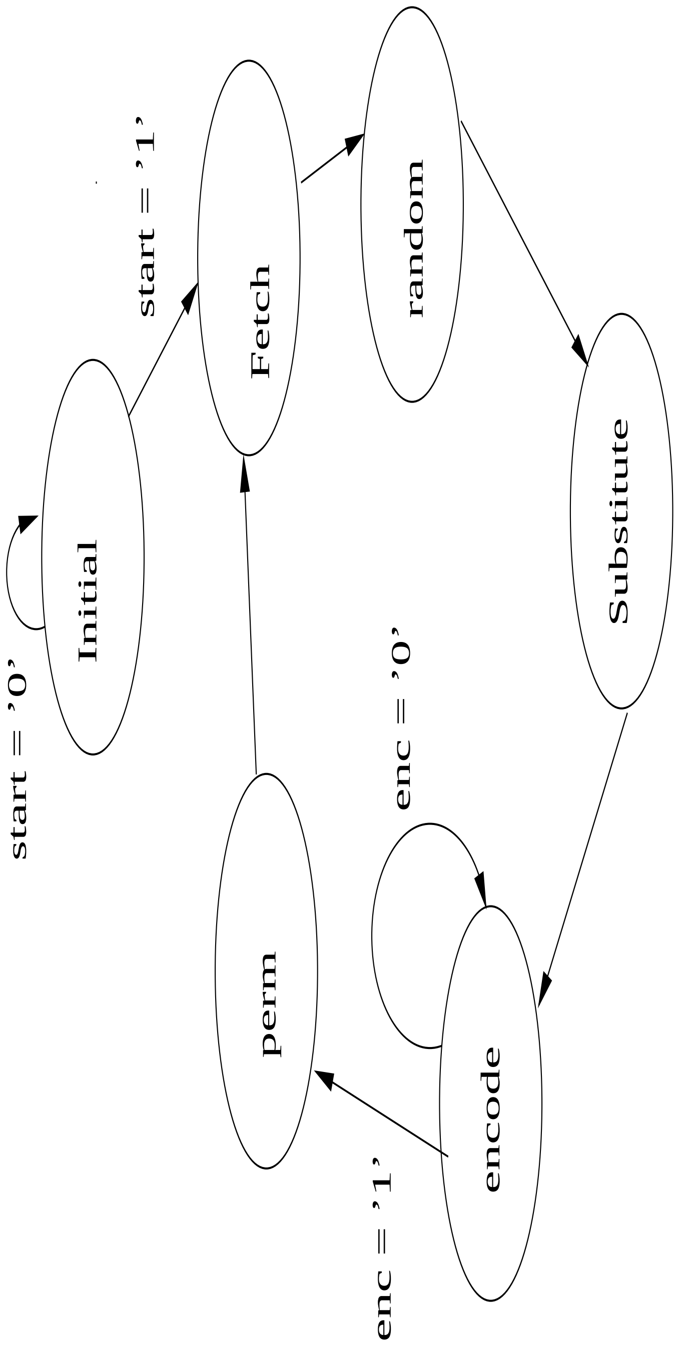

Figure 5.5. Control unit as a Finite State Machine (FSM)

randomness in the output. The ECBC was used as a pseudorandom number generator in counter mode. The TestU01 [27] was used to test the randomness of the output of ECBC in counter mode. We tested for p-values within the boundary $[10^{-4}, 1-10^{-4}]$. Any p-values

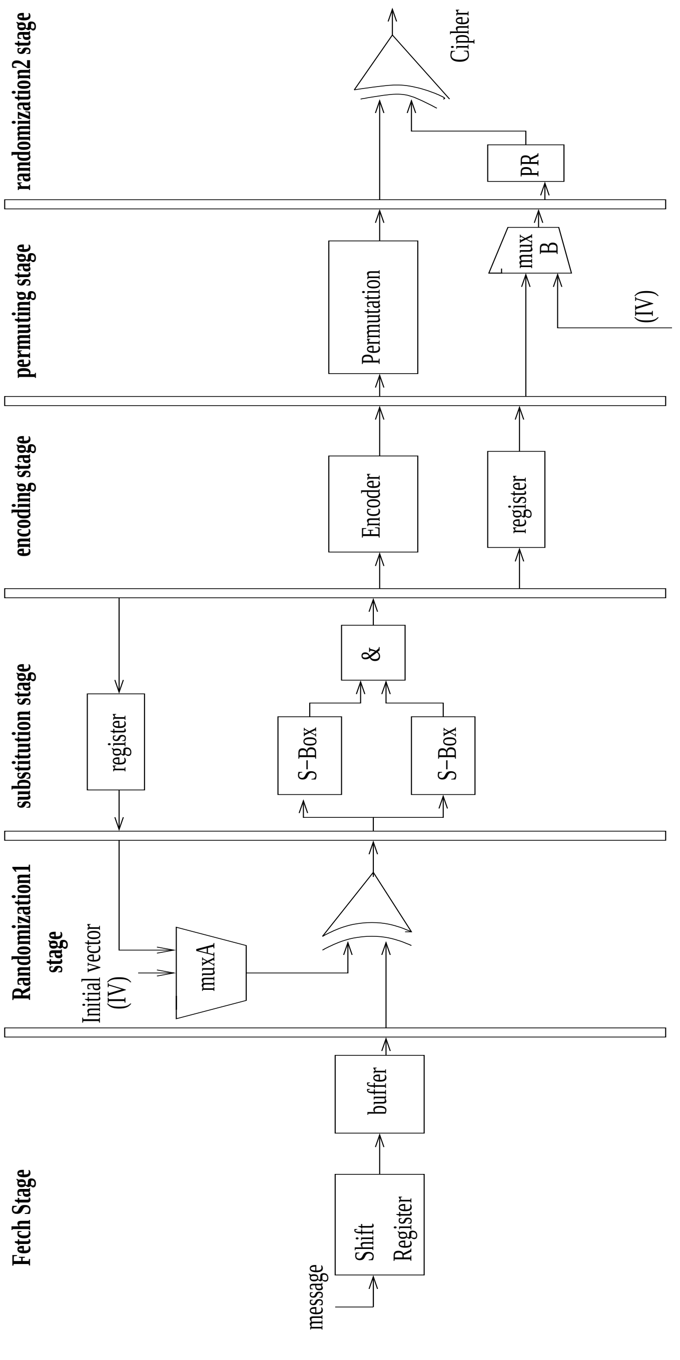

Figure 5.6. Architecture of the pipelined ECBC datapath

lying outside this range is considered as failure, while the ones within the range is considered as pass. Table 5.1 lists the test suites, the number of tests in each suite and the results. A total of 319 tests were carried out and the ECBC passed all of them. It is important to point out that because the scheme passes the test of randomization, is not a guarantee that such a scheme is secure. However, it is important that a cryptographic scheme's output should be random.

Table 5.1. Result of Test Suites on ECBC

Test Suite	Parameters	No. of Statistics	Results
Small Crush	Standard	15	Pass
Crush	Standard	144	Pass
Big Crush	Standard	160	Pass

We also plotted the graph of Bit Error rate (BER) against Signal-to-Noise Ratio SNR) in Fig. 5.7 to see the effect of ECBC system on the performance. The green curve(x) is for the case where ECBC is not used while the blue curve (o) is for the case where ECBC is part of the communication system. The graph shows that the performance is the same in both cases.

We implemented the ECBC on Field Programmable Gate Array (FPGA) on Xilinx Spartan 3E xc3s1200e-4ft256 using ISE Foundation 11.2. The result of implementation is shown in Table 5.2. For the non-pipelined architecture, 23% of the slices were used and has a maximum frequency of 130.924 MHz. For the piplined architecture, 26% of the slices were used and has a maximum frequency of 105 MHz. Even though maximum frequency reduced, the throughput for the piplined architecture is 8 Gb/S. The Non-ECBC method combines AES and LDPC as separate unit. These results show significant reduction in hardware usage.

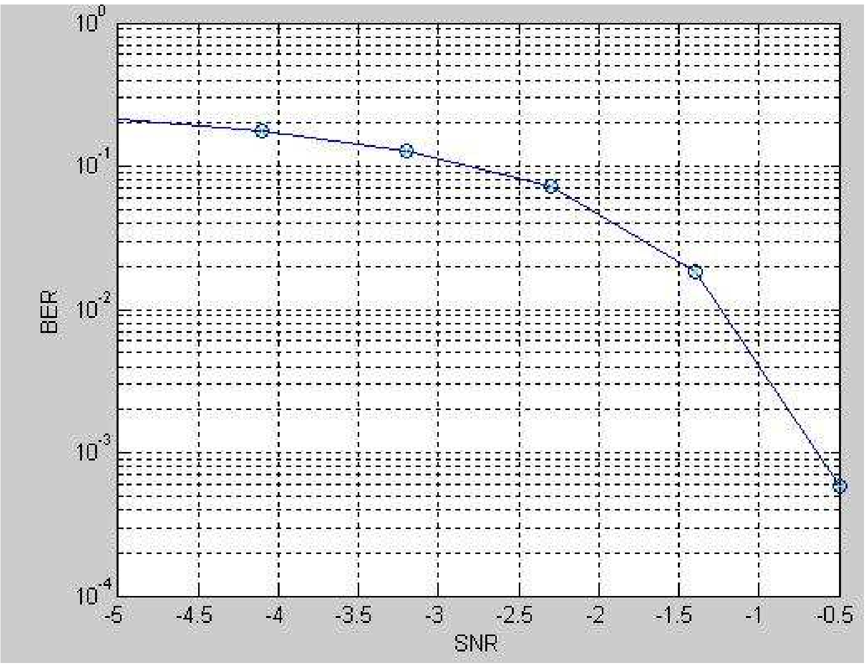

Figure 5.7. Plot of BER against SNR

Table 5.2. Result of FPGA Implementation

Parameters	non-pipelined ECBC	pipelined ECBC	Non-ECBC method
Bels	3328	3912	28178
Maximum frequency (MHz)	130	105.7	131
Total Flip flop and latches	1691	2058	33794

CHAPTER 6

ARCHITECTURE AND IMPLEMENTATION OF AMHC

In this chapter, the hardware architecture of the AMH cipher is presented. The architecture and implementation of the AMH cipher is presented for prove of concept. The schemes presented in this research was implemented in software and hardware. The architecture is aimed at achieving high throughput and reducing the required hardware resources [55]. The Top-level architecure of the AMH encryption is shown in Fig. 6.1. The main units in the top-level architecure are S-Box, ShiftRow, MixColumn, Encoder and permutation unit.

The 128-bit plaintext to be encrypted is temporarily stored in in a register and then fed into a multiplexer (MUX). The multiplexer selects between the plaintext or round states. For the first round, the input to the multiplxer is the original plaintext and at subsequent rounds, the input is the output of the previous round. The output of the multiplexer is then XORed with the 128-bit initial key. The initial key is also fed into a multiplexer. The multiplexer selects between the initial and the round key. The output of the XOR gates form a state and then fed into the S-Box unit. This unit implements the non-linear operation of this cipher. A detailed architecure will be discussed in this chapter. This unit is used for substitution where each by byte of the input is state is replaced with the affine mapping of its multiplicative inverse. The output from the S-Box unit is then fed into the ShiftRow and the Encoder units. For the first Nr-1 round, the multiplexer connected to both units select the output from the shiftRow unit. At the last Nrth round, multiplexer selects the output from the encoder unit. The output from the multiplexer is then fed into MixColumn unit and the permutation unit. For the first Nr-1 rounds, the multiplexer selects the output from the MixColumn, while at the Nrth round, the multiplexer selects the output from the permutation unit. In the first Nr-1 rounds, the output from the MixColumn unit is fed back into the input multiplexer while Nr-1 round keys are generated from the key schedule which is then XORed with the round inputs. A detailed archicture of the major unit will now be discussed.

6.1. S-Box Unit

The S-Box is used for substituting input bytes of a state with the affine mapping of its multiplicative inverse. In this transformation each block is replaced by its substitution in an S-Box table. The implementation of S-Box consists of mathematical function, the multiplicative inverse of each byte [20] [9]. The multiplicative inverse could be implemented using combinational logic however, the approach has large time and area requirements. The S-Box could also be implemented by storing the multiplicative inverses of bytes in GF(2^8) using look-up tables. In this work, the values obtained after taking the multiplicative inverse and applying the affine mapping are stored in the look-up table of the S-Box. The look-up tables cosumes more area more than the combinational logic implementation but access time is shorter. The architecture of the S-Box is shown in Fig. 6.2. All the byte values of a state can be obtained in parallel via an array of mux where the bytes are the selectors to the multiplexer as shown in Fig. 6.2. The individual outputs from the multiplexers are then concatenated together in a register to form a 128-bit output of the S-Box unit. The substitution operation that takes place with the aid of the S-Box is the only non-linear operation in AMH cipher.

6.2. ShiftRow Unit

The architecture of the ShiftRow is shown in Fig. 6.3. In this transformation the rows of the block state are shifted over different offsets. The number of shifts is determined by the row number. For example, row 0 is shifted 0 times, row 1 is shifted left once, row two is left shifted twice and the third row is left shifted thrice. The ShiftRow unit is implemented with the aid of multiplexers as shown in Fig. 6.3. Each row of a state is placed in a register and all the eight bytes of the row are fed into each multiplexer [56]. The row number determines the selector to each multiplexer. The outputs of all the multiplexers are now concatenated to form a 128-bit output. The entire hardware implementation of the shift row is shown in Fig. 6.3 where we avoided the use of shift register and instead used the combinational multiplexer.

6.3. MixColumn Unit

Each column of the state is considered as a polynomial over GF(2^8) for the MixColumn transformation. This column is multiplied with a constant polynomial C(x) or D(x) over a finite field in encryption. The hardware architecture of the MixColumn is shown in Fig. 6.4. In the hardware architecture, the input state is stored in a register and then each byte is simultaneously multiplied by 2 and 3 as shown in Fig. 6.4. The corresponding outputs from the multiplication by 2 and 3 are then xored with some state byte to form the output byte of the MixColumn unit. The multiplication of a byte (8-bit hexadecimal number by 2 or x: Left shift the 8-bit number by 1-bit This is done using a simple X-OR operation. If the most significant bit of the initial polynomial is 1, then the output is obtained by XORing the left shifted value with 1B else if the most significant bit of the initial polynomial before left shifting is 0, then the output is just the left shifted value.

6.4. Encoder Unit

The architecture of the LDPC encoder is shown in Fig. 6.5. Each parity bit is obtained by matrix-vector multiplication of matrix H_2 with the output of the SBOX unit. Since matrix-vector multiplication operation is carried out in GF(2), Each row of matrix H_2 is ANDed with the vector output (s) from the SBOX. The outputs from the (n-k) AND gate are then xored together to produce the parity bit. Multiplication by H_1^{-1} is not not necessary if the H matrix is systematic. The codeword is reconstructed by concatenating the parity with the $kx1$ SBOX output with the aid of CodeWord Construction Unit.

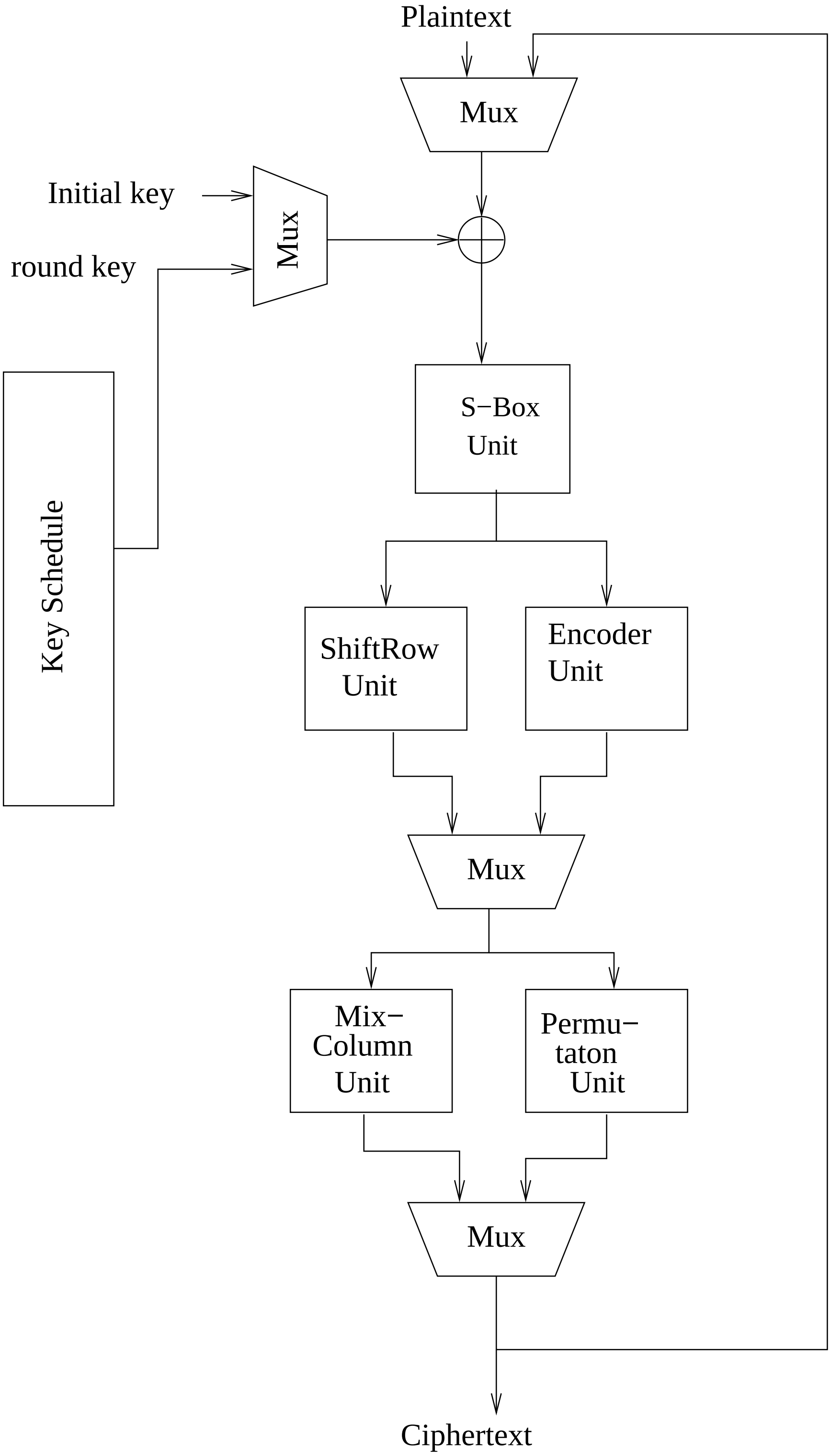

Figure 6.1. Top Level Architecture of AMH Cipher Encoder

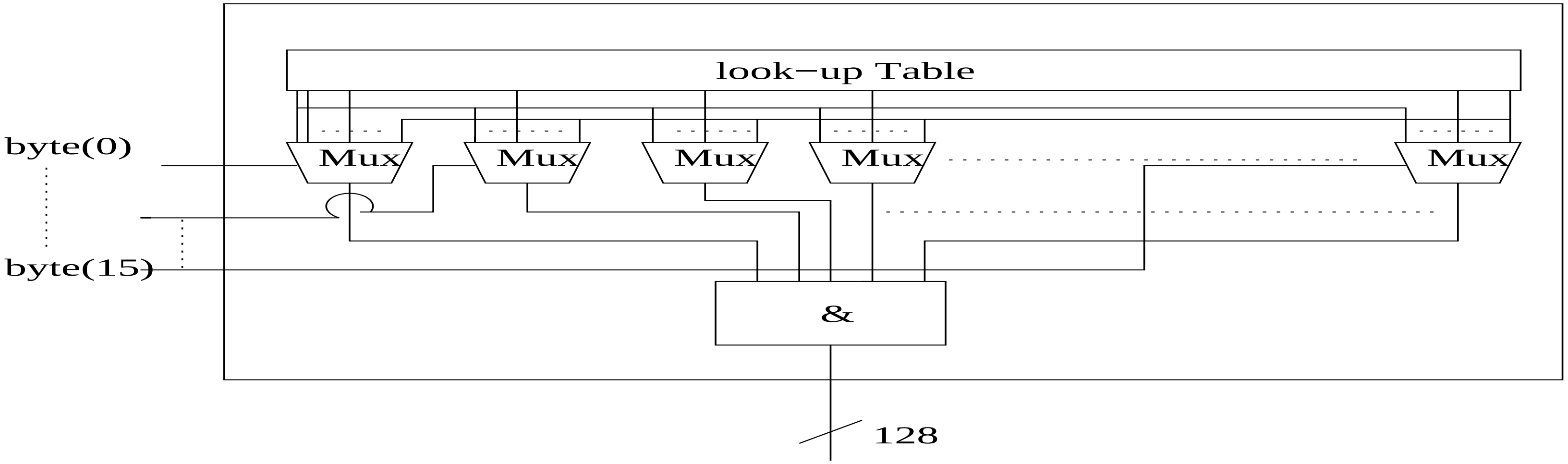

Figure 6.2. Architecture of S-Box

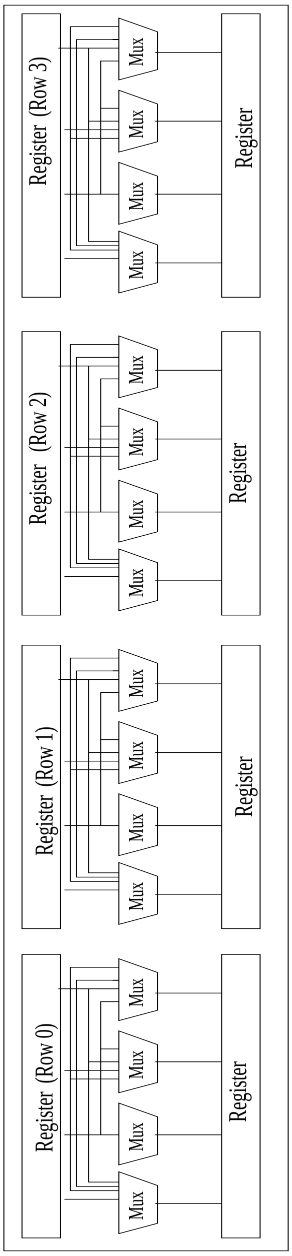

Figure 6.3. Architecture of ShiftRow

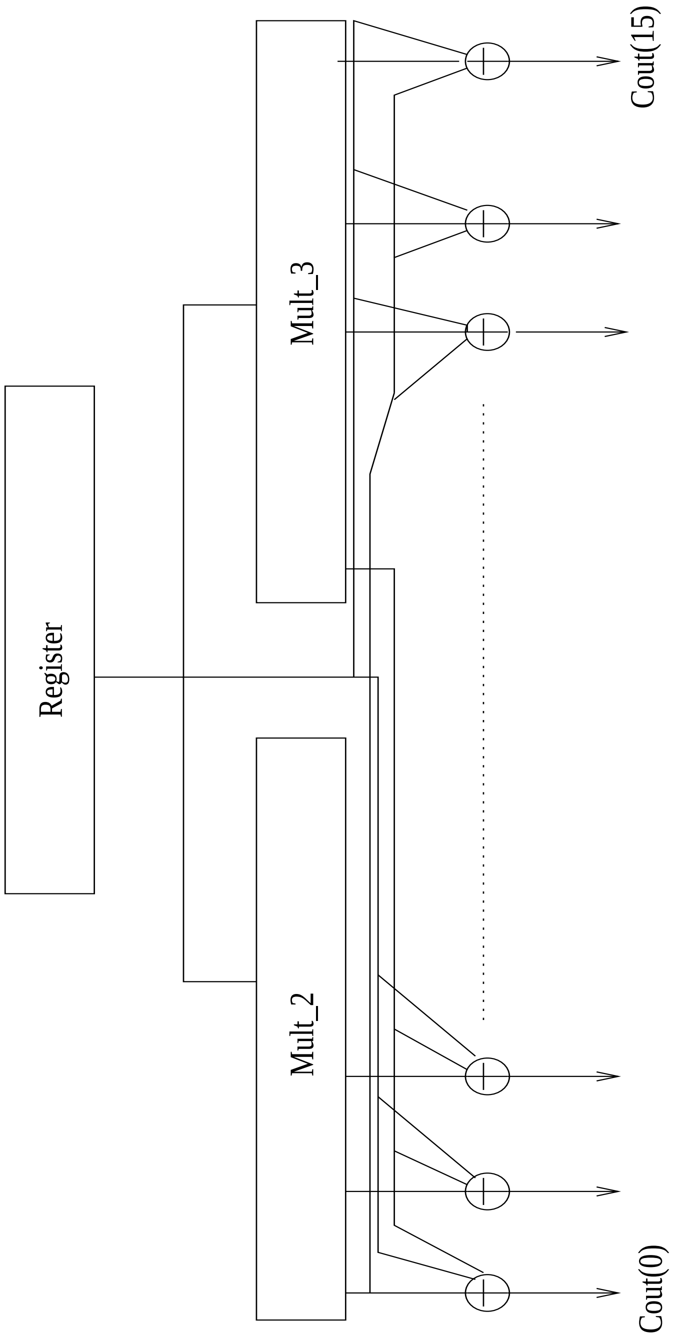

Figure 6.4. Architecture of MixColumn

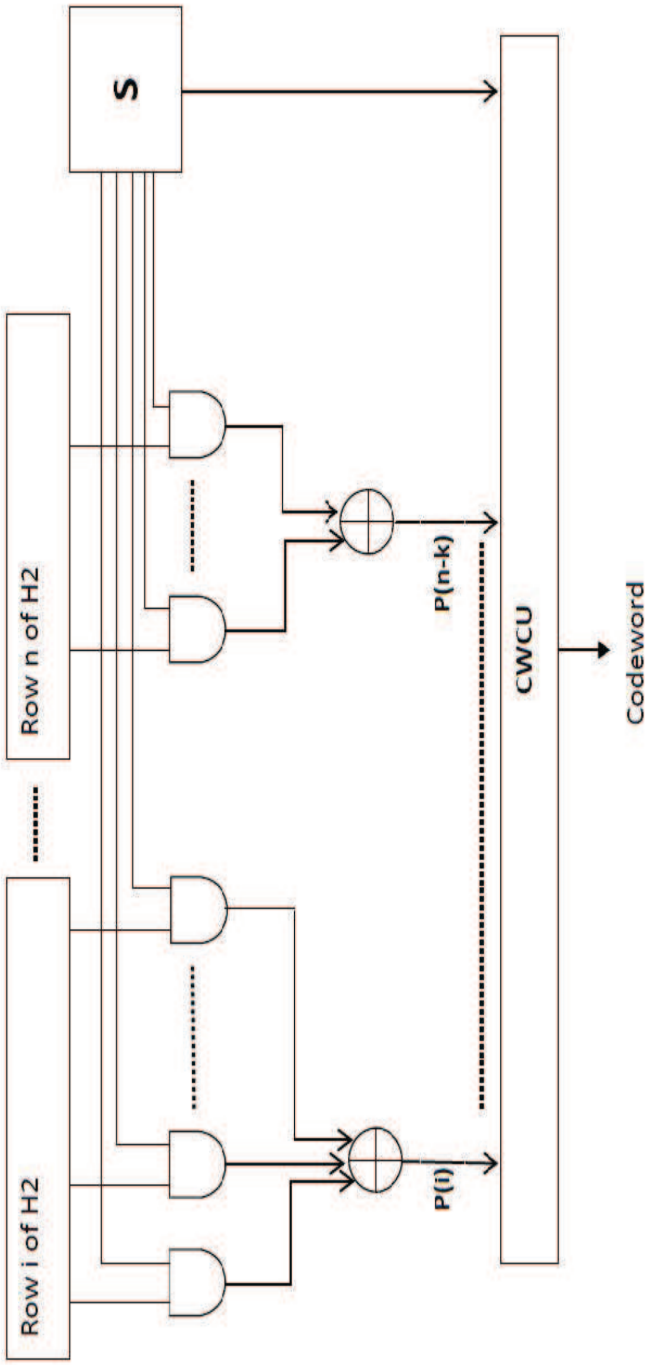

Figure 6.5. Architecture of the LDPC encoder unit

CHAPTER 7

JOINT ENCRYPTION, ERROR CORRECTION AND MODULATION (JEEM) SCHEME

7.1. Introduction

As part of the effort in providing security in the physical layer, we present a joint encryption, error correction and modulation scheme. The conventional modulation scheme are modified to provide random mapping of encoded information. This design is able to cater for speed, security and has the potential of reducing the key in situation whereby the security scheme is serving as a reinforcement of security in the upper layer. As a result, in this chapter, we present a scheme that combines encryption, error correction and modulation. Instead of fixed mapping of bits (information) to waveform, we present the design of a randomized mapping scheme with purpose of degrading the communication channel of the eavesdropper (Eve). We present several plots that proves the security of our system. The plots proves that the communication channel of the eavedropper is severely degraded. The scheme presented in this work does not compromise the full error correcting capability of the code used. The scheme also has the potential of reducing the key size of McEliece-like scheme.

This scheme is based on the structure of McElice cryptosystem. In the McEliece cryptosystem, a plaintext M is encrypted into a ciphertext C as shown in Eqn. 77

$$C = MG' + Z = MSGP + Z \tag{77}$$

where

C: ciphertext of length n,

M: plaintext of length k,

Z: random error vector of length n whose hamming weight t' = t,

G' = SGP: public-key,

G: Generator matrix of a t-error correcting code (Goppa for the case McEliece)

S, G, P are private keys.

In the JEEM, a plaintext M is multiplied by $G' = FGP$ where F is a non-linear function as opposed to the scramber in McEliece scheme. The result is encoded with the aid of the generator matrix and then permuted using the permutation matrix P as shown in Eqn. 78

$$C' = MG' = MFGP \tag{78}$$

Now, instead of performing a modulo 2 addition of C' with an error vector Z, we are proposing a random modulating function M_r, so that Eqn. 77 is replaced with an equivalent equation shown in Eqn. 79 and 80.

$$C = (MG')M_r = (MFGP)M_r \tag{79}$$

$$C = (MG')M_r = (MFGP)M_r \equiv MSGP + Z \tag{80}$$

In this scheme, the mapping process of the modulation is controlled by the error vectors. As a result, the modulation scheme is able to provide both randomization and modulation without compromizing the structure of the McEliece-like scheme. The modulation function could further be exploited to reduce the key size of the McEliece scheme [28]. The randomized modulation scheme is illustrated in Fig. 7.1.

The selector for multiplexer 1 (MUX 1) and multiplexer 2 (MUX 2) is controlled by the error vector. Instead of fixed mapping in BPSK where binary '1' is mapped to +sin ωc and binary '0' is mapped to -sin ωc, the mapping is based on the error vector that is an input to

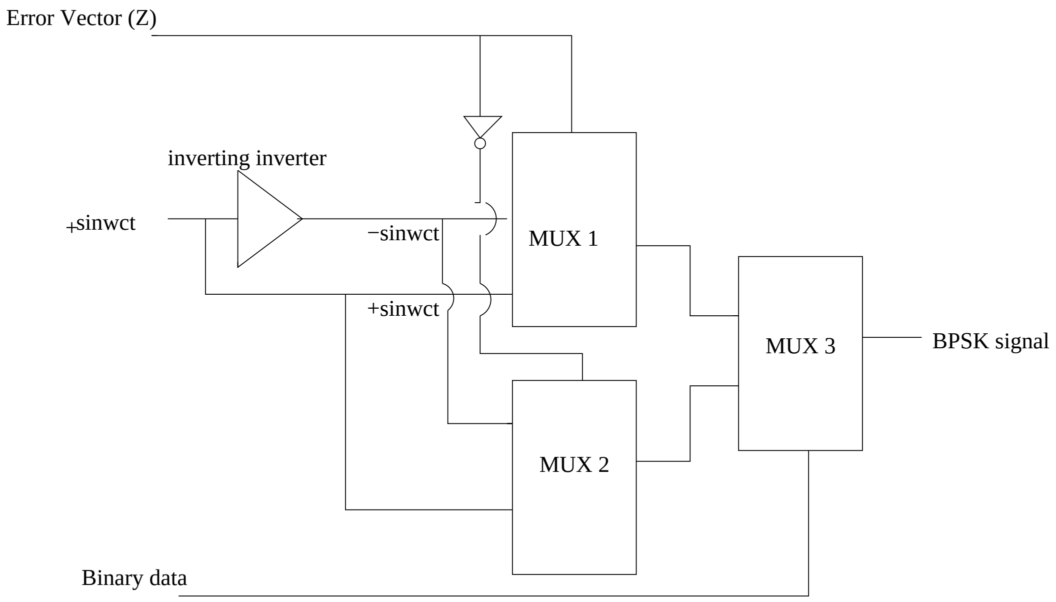

Figure 7.1. High Level Block Diagram of the Randomized Modulation Scheme

MUX1 and MUX2. The input to selector 2 is an inverted input to the selector 1. The binary data is the selector for multiplexer 3 (MUX3). The encryption process is illustrated in Fig. 7.2.

The BPSK constellation plot is also shown in Fig. 7.3.

7.2. Security of JEEM

It was shown by authors in [57], [37], [11] and [29] that it was exclusively possible to achieve a theoretically secure communication by means of coding at the physical layer if Eve has a worse channel then Bob. This could be achieved by degrading the Eve's communication channel in the wiretap channel. We use the Bit Error Rate (BER) over a message length as a measure of security as it is easy to analyze and measure. It has been argued in the past that a high BER at Eve can deliver improved resilience against eavesdropping if used with standard cryptographic techniques. In this case, the degraded channel offers physical layer security. In terms of brute force attacks, the attacker could try to enumerate the secret key until a meaningful message is achieved. This is similar to attempting to obtain the check matrix H

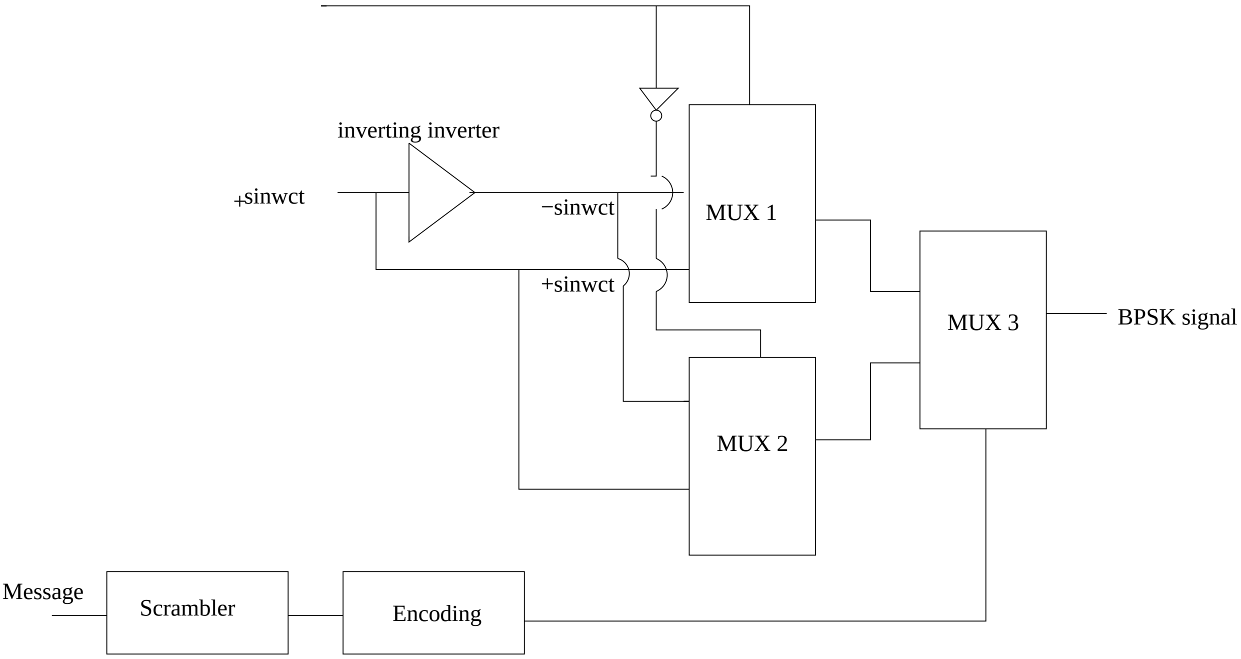

Figure 7.2. High Level Block Diagram of the encryption Randomized Modulation Scheme

of the LDPC codes and the error vectors. However if a QC-LDPC code is used, enumerating to obtain the check matrix is infeasible even if the parameter of such codes are made public as shown in [4], [6] and [3]. Depending on the code length of the code, enumerating to get the error vector is impractical. For example, for a (2044, 1024) LDPC code will be 2^{1024} and this will discourage brute force attack. For illustration purposes, we have simulated the performance (BER) of the communication channels for Bob and Eve (eavesdropper) in figs. 7.4, 7.5, 7.6 and 7.7. We assume

In fig. 7.4, we show the plot of BER against Signal to Noise Ratio (SNR) for both Bob and Eve (Eavesdropper). The communication is through Additive White Guassian Noise (AWGN) channel and the modulation scheme is the Binary Phase Shift Keying (BPSK). The red lines is the plot for Eve, the green line is for Bob and the blue line is the plot for the theorical case. The plot shows shows how degraded the performance of Eve's communication channel as opposed to Bobs. From the figure, the BER of Eve stays almost constant irrespective of the

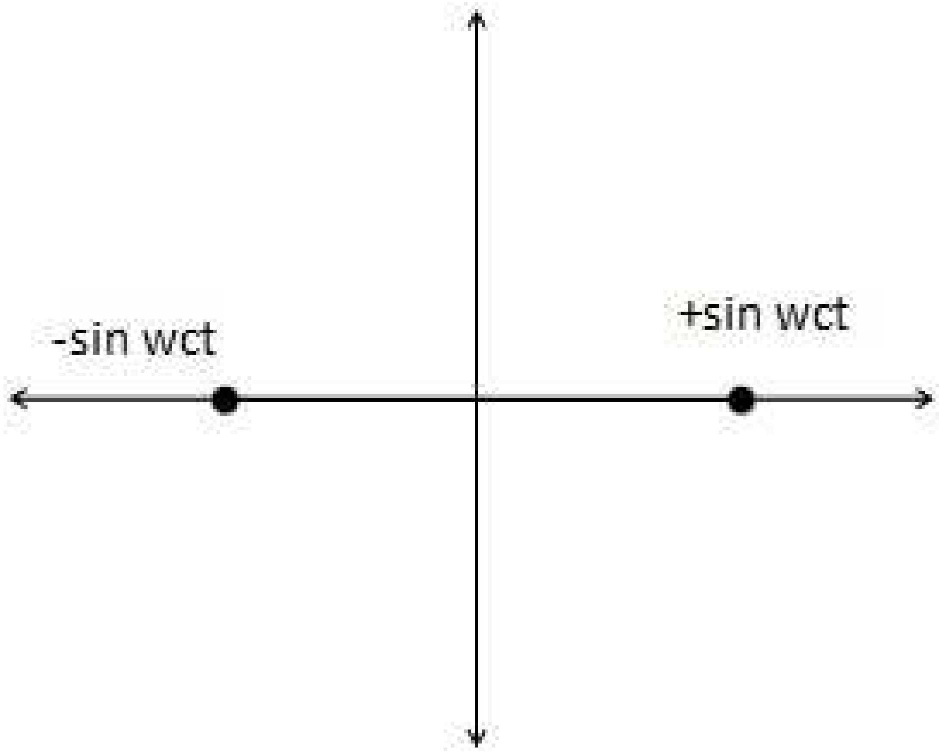

Figure 7.3. Illustration of a BPSK constellation

value of SNR. However in the case Bob, the BER reduces as the SNR increases. This proves that our scheme is secure and it will be very difficult for EVE to recover message because of degradation of her channel.

The figure in fig. 7.5 shows the plot of BER against SNR for both Bob and Eve (Eavesdropper) in an AWGN channel and the modulation scheme is the Quadrature Phase Shift Keying (QPSK). The plot shows how degraded the performance of Eve's communication channel as opposed to Bobs. The diamond points represent the plot for EVE, the circle is the plot for Bob and the dot points is the plot for the theoritical values. From the figure, the BER of Eve stays almost constant irrespective of the value of SNR. However in the case Bob, the BER reduces as the SNR increases. This proves that our scheme is secure and it will be very difficult for EVE to recover message because of degradation of her channel. This also show that the security is preserved irrespective of the modulation scheme.

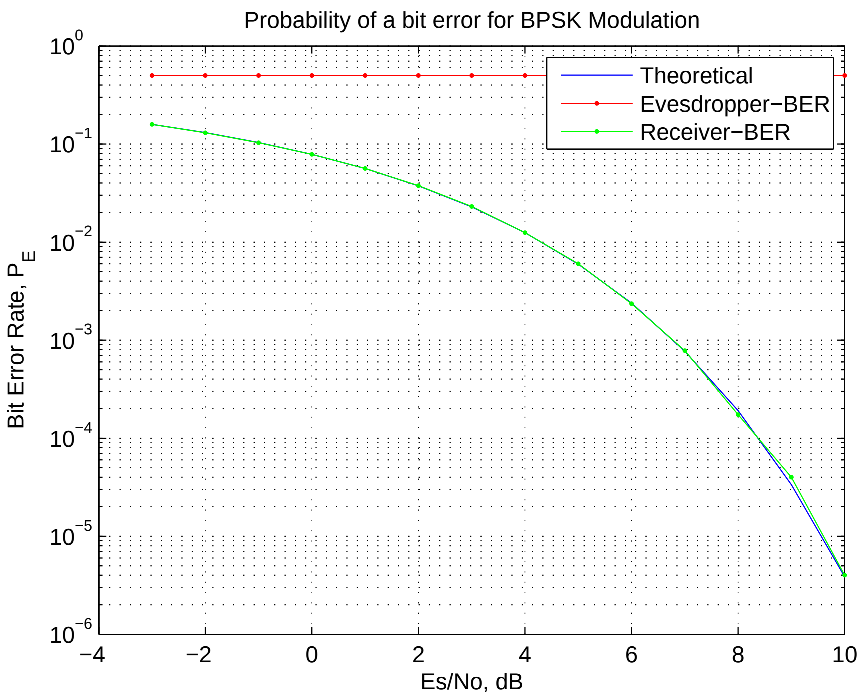

Figure 7.4. Plot of BER against SNR

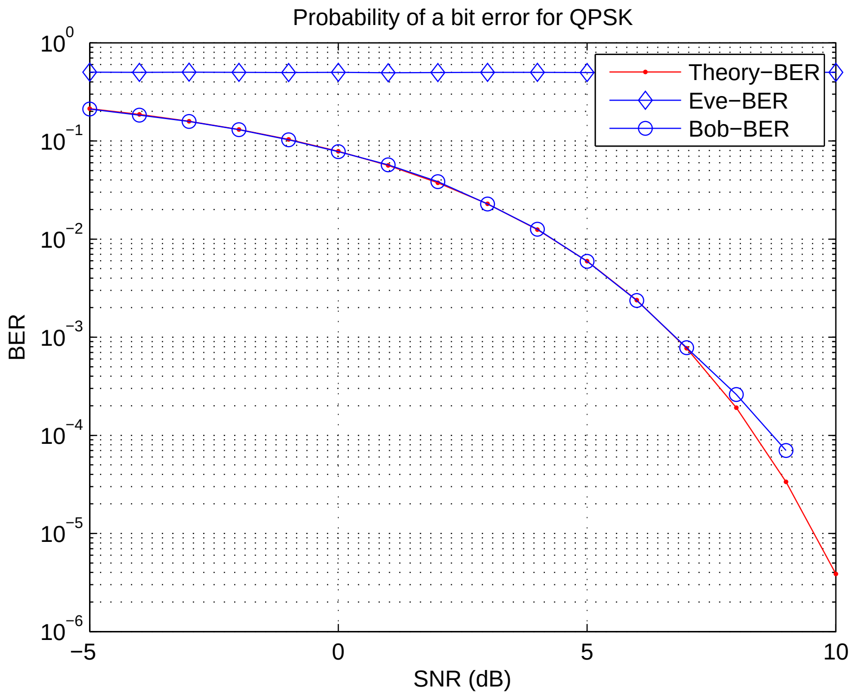

Figure 7.5. Plot of BER against SNR

The figure in fig. 7.6 shows the plot of symbol Error Rate (SER) against SNR for both Bob and Eve (Eavesdropper) in an AWGN channel and the modulation scheme is the Quadrature Phase Shift Keying (QPSK). The plot shows how degraded the performance of Eve's communication channel as opposed to that of Bob's. The diamond points represent the plot for Bob, the circle is the plot for Eve and the dot points is the plot for the theoritical values. From the figure, the SER of Eve stays almost constant irrespective of the value of SNR. However in the case Bob, the SER reduces as the SNR increases. This proves that our scheme is secure and it will be very difficult for EVE to recover message because of degradation of her channel. This also show that the security is preserved irrespective of the modulation scheme.

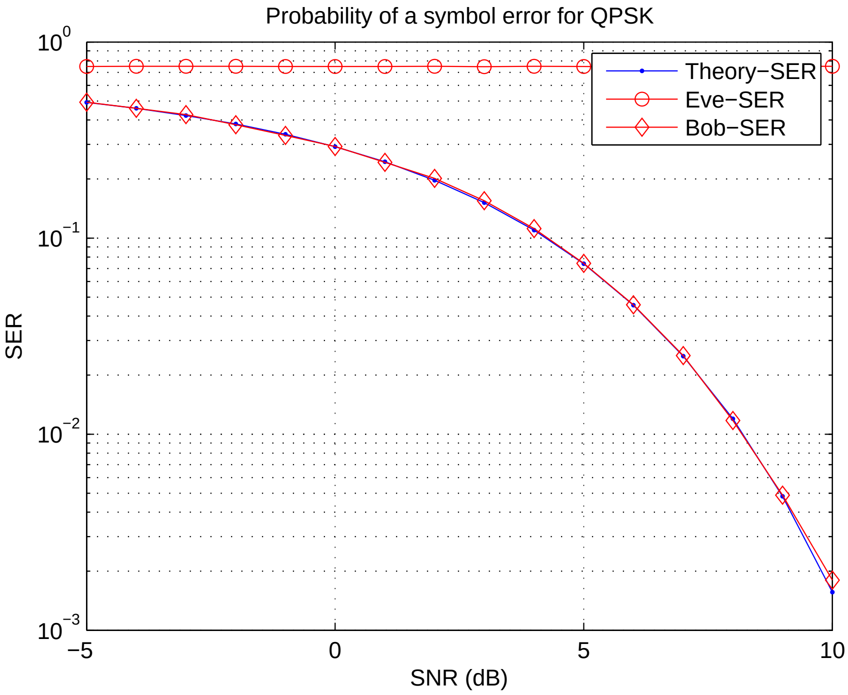

Figure 7.6. Plot of SER against SNR

The figure in fig. 7.7 shows the plot of Bit Error Rate (BER) and Frame Error Rate (FER) against SNR for both Bob and Eve (Eavesdropper) in an AWGN channel and the modulation scheme is the Binary Phase Shift Keying (BPSK). The plot shows how degraded the performance of Eve's communication channel as opposed to that of Bob's in cases where ldpc code was used at the receiver end of Eve. Eve does not possess the seed that is used to randomize the mapping in the modulation scheme. The circle points represent the plot for Bob which shows that there was no degradation in the performance. The BER reduces as the SNR increases from 0 dB to 5 dB. The square points is the plot of the FER for Bob and shows same characteristic as that of the BER except that it has higher dB values. The star and the diamond points is the plot of the BER/FER for Eve and it is constant as the SNR is increased from 0 dB to 3 dB. The black plot (BER/SNR) shows the case where the eavesdropper does not use an error correcting code (LDPC). The yellow plot is that of FER/SNR and an LDPC code is not employed. The red plot is that of BER/SNR and in this case Eve uses the LDPC decoder while the green plot is that of FER/SNR for the eavesdropper and the lDPC decode is also employed. The Error rate remains constant irrespective of the value of SNR. The eavesdropper does not employ ldpc This proves that our scheme is secure and it will be very difficult for EVE to recover message because of degradation of her channel. This also illustrates the security of our scheme irrespective of the SNR.

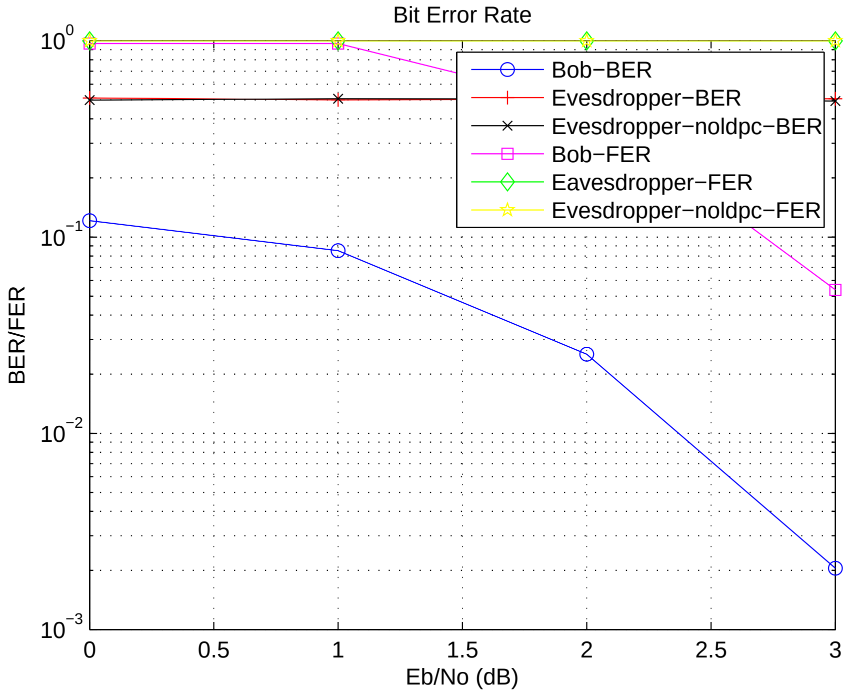

Figure 7.7. Plot showing BER/FER against SNR for both Bob and Eavesdropper

CHAPTER 8

CONCLUSION AND FUTURE RESEARCH

In this research, we have presented the motivation for combining error correction and security. We pointed out that the major challenges facing resource constraint wireless devices are error resiliance, security, and speed. In order to address this challenges, we have presented several physical layer encryption schemes that are capable of providing data reliability, secrecy and integrity. As a result, the main purpose of this research is to construct joint error correction and encryption/decryption schemes to facilitate secure, reliable and efficient data transmission. The schemes presented are all able to provide error correction and security.

The schemes presented in this research could be broadly divided into two groups, error correction based and cipher based. We presented two Error Correction Based Cipher (ECBC). The ECBC scheme 1 which is based on block chaining method. We take advantage of the properties of LDPC codes in this scheme where the normal cipher is replaced with this code. It is too computationally intensive to quess the H matrix of the LDPC code as already shown. The number of randomly chosen element in the H matrix is very high. We also analyzed the security of this cipher to Known-plaintext and chosen-plaintext attack. Randomization test was also carried out on this scheme which is an important test of a cipher. It is important to note that decoding error in one block requires the retransmission of all the ciphertext chained together.

Error Correction Based Cipher Scheme 2 was also presented. The scheme is a private key algbraic based cryptosystem based on non-linear code called Nordstrom Robinson Code. We take advantage of the fact that Nordstrom Robinson code is a binary image of an octacode. As a result, the eavesdropper is forced to decode the cipher using the non-linear code while the intended receiver can use a seed to decode the cipher linearly using the octacode. We also analyze the code against Known and Chosen-plaintext attack.

The AMH cipher is also presented in this research. The AMH is a hybrid of AES and McEliece ciphers. We employ the LDPC code in this scheme and we were able to reduce the number of rounds from 10 rounds of the AES to 5 rounds. This shows the advantage of having a joint scheme. We also analyzed the hybrid cipher to linear and differential cryptanalysis. The cipher was also subjected to randomization test.

The architecture and the implementation of joint schemes was also presented. In the schemes presented, the error correcting capability of the codes were fully preserved because the error deliberately introduced at the sender end can be removed at receiver because of synchronization. There were no trade off between reliability and security because errors introduced at the transmitter are removed at the receiver. Hence the schemes presented could utilize their full capacity to correct channel errors. The schemes are also secure against some conventional attacks. The result of implementations are presented. These joint schemes could be easily be adapted to existing protocols such as in CC2420 - Single-Chip 2.4 GHz IEEE 802.15.4 Compliant and ZigBee Ready RF Transceiver where AES is already implemented or CCMP.

8.1. Future Research

Although we have presented some crypanalytical attacks against the schemes presented in this research, other crypanalytical attack could be investigated to attack the cryptosystems presented in this research. New protocols could also be developed for digital signature and authentication using these schemes. The key generation and management could also be investigated for this scheme.

BIBLIOGRAPHY

[1] A. Biri A. Ahmad and H. Afifi, *Study of a new physical layer encryption concept*, 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, 2008, pp. 860–865.

[2] O. B. Adamo, S. P. Mohanty, E. Kougianos, and M. Varanasi, *Architecture for encryption and watermarking units towards the making of a secure digital camera*, IEEE International SOC Conference (SOCC), 2006, pp. 141–144.

[3] A. A. Sobhi Afshar, T. Eghlidos, and M. R. Aref, *Efficient secure channel coding based on quasi-cyclic low-density parity-check codes*, IET Communications, vol. 3, 2009, p. 279292.

[4] M. Baldi and F. Chiaraluce, *Cryptanalysis of a new instance of mceliece cryptosystem based on qc-ldpc codes*, IEEE International Syposium of Information Theory (2007), 25912595.

[5] ______, *Cryptanalysis of a new instance of mceliece cryptosystem based on qc-ldpc codes*, IEEE ISIT, vol. 2011, 2011, p. 25912595.

[6] M. Baldi, F. Chiaraluce, R. Garello, and F. Mininni, *Quasi-cyclic low-density parity-check codes in the mceliece cryptosystem*, IEEE ICC (2007), 951956.

[7] E.R. Berlekamp, R.J. McEliece, and H.C.A. Van Tilborg, *On the inherent intractability of certain coding problems*, IEEE Trans. on Information Theory 24 (1978), 384–386.

[8] T. A. Berson, *Failure of the mceliece public-key cryptosystem under message-resend and related-message attack*, In Advances in cryptology -EUROCRYPT97, Lecture Notes in Computer Science, 1997.

[9] C.C.Wang, T. Truong, H. Shao, L. Deutsch, J. Omura, and I. Reed, *Vlsi architectures for computing multiplications and inverses in gf(2^m)*, IEEE Transactions on Computers, 1985, pp. 709–717.

[10] K Narayan C.N. Mathur and K. P. Subbalakshmi, *On the design of error-correcting ciphers*, EURASIP Journal on Wireless Communications and Networking, vol. 2006, 2006, pp. 860–865.

[11] I. Csiszar and J. Korner, *Broadcast channels with confidential messages*, IEEE Trans. on Info. Theory 24 (1978), no. 3, 339–348.

[12] J. Daemen, L. Knudsen, and V. Rijmen, *The block cipher square*, 4th International Workshop on Fast Software Encryption (FSE'97, Springer-Verlag, 1997, pp. 149–165.

[13] J. Daemen and V. Rijmen, *The design of rijndael: Aes - the advanced encryption standard*, Springler-Verlag 2002 (2002).

[14] D. E. Denning, *Cryptography and data security*, Addison Wesley, 1982.

[15] G. D. J Forney, N. J. A. Sloane, and M. D. Trott, *The nordstrom-robinson code is the binary image of the octacode*, Coding and Quantization, vol. 14, 1992, pp. 19–26.

[16] D. Gligoroski, S. Knapskog, and S. Andova, *Cryptcoding - encryption and error-correction coding in a single step*, Security and Management Conference, 2006, pp. 145–151.

[17] W. Godoy and D. Periera, *A proposal of a cryptography algorithm with techniques of error correction*, Computer Communications, vol. 20, 1997, pp. 1374–1380.

[18] R. P. Grimaldi, *Discrete and combinatorial mathematics: An applied introduction*, Addison Wesley, 2003.

[19] A. R. Hammons, P. V. Kumar, A. R. Calderbank, N. J. A. Sloane, and P. Sole, *The z_4 linearity of kerdock, preparata, goethals and related codes*, IEEE Transaction of Information Theory, vol. 40, 1994, pp. 301–319.

[20] H.Brunner, A. Curiger, and M. Hofstetter, *On computing multiplicative inverses in gf(2^m)*, IEEE Transactions on Computers, 1993, pp. 1010–1015.

[21] M. Healy, T. Newe, and E. Lewis, *Analysis of hardware encryption versus software encryption on wireless sensor network motes*, Smart Sensors and Sensing Technology, Lecture Notes in Electrical Engineering, vol. 20, Springer Berlin Heidelberg, 2008, pp. 3–14.

[22] T. Hwang and T. R. N. Rao, *Secret error-correcting codes (secc)*, Proc. of Crpto'88, 1988, pp. 540–563.

[23] NIST Federal information processing standards publication 197:, *Advanced encryption standard*, http://csrc.nist.gov/publications/fips/fips197/fips197.pdf (2001).

[24] J. P. Jordan, *A variant of a public key cryptosystem based on goppa codes*, SIGACT News, 1983, pp. 61–66.

[25] S. C. Kak, *Joint encryption and error correction coding*, IEEE Conference on Security and Privacy, April 1983, pp. 55–60.

[26] R. Klein, M. Varanasi, and L. Dunning, *A systematic (16,8) code for correcting double errors, and detecting random tripple error*, IEEE SOUTHEASTCON'96, 1996.

[27] P. L'Ecuyer and R. Simard, *Testu01: A c library for empirical testing of random number generators*, ACM Transactions on Mathematical Software 33, 4 (2007), no. 22, 470–474.

[28] W. Leng, L. Sang C. Xu, and X. Zhang, *Applications of modulation in a mceliece-like symmetric-key scheme*, IEEE 71st Vehicular Technology Conference (VTC 2010-Spring, Lecture Notes in Computer Science, Springer-Verlag, 2010, pp. 1–4.

[29] S. K. Leung-Yan-Cheong and M. E. Hellman, *The gaussian wire-tap channel*, IEEE Transactions on Information Theory, vol. 24, 1978, p. 451456.

[30] M. C. Lin and H. L. Fu, *Information rate of mceliece's public-key cryptosystem*, Electronics Letter, vol. 26, 1990, pp. 16–18.

[31] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes. i and ii*, North-Holland Publishing Co., Amsterdam, North-Holland Mathematical Library, vol. 16, 1977.

[32] C. N. Mathur, *A mathematical framework for combining error correction and encryption*, Phd dissertation, Stevens Institute of Technology, NJ, 2007.

[33] R. J. Mceliece, *A public-key cryptosystem based on algebraic coding theory*, Tech. report, DSN Progress Rep., Jet Propulsion Laboratory, Pasadena,CA, 1978.

[34] C. H. Meyer and S. M. Matyas, *Cryptography: A new dimension in computer data security*, John Wiley and Sons, Inc, 1982.

[35] A. W. Nordstrom and J. P. Robinson, *An optimum nonlinear code*, Information Control, 1967, pp. 284–287.

[36] K. Nyberg, *Differentially uniform mappings for cryptography*, In Advances in cryptology -EUROCRYPT93, Lecture Notes in Computer Science 765, vol. 765, T. Helleseth, Ed., Springer-Verlag, 1994, p. 5564.

[37] L. Ozarow and A. D. Wyner, *Wire-tap channel ii*, ATT Bell Laboratories technical journal 63 (1984), 21352157.

[38] C. S. Park, *Improving code rate of mceliece's public-key cryptosystem*, Electronics Letter, 1989, pp. 1466–1467.

[39] C.S. Park, *Improving code rate of mcelieces public-key cryptosystem*, Electron Letter, vol. 25, 1989, pp. 1466–1467.

[40] W. W Peterson and E. J. Weldon, *Error correcting codes*, MIT Press, 1972.

[41] T. Pionteck, T. Staake, T. Stiefmeier, L. Kabulepa, and M. Glesner, *Design of a reconfigurable aes encryption/decryption engine for mobile terminals*, Proc. of the IEEE International Symposium on Circuits and Systems, vol. 2, May 2004, pp. 545–548.

[42] J. Proakis, *Digital communications*, McGraw-Hill, New York, aNY, 1995.

[43] T. R. N. Rao and K. H. Nam, *A private-key algebraic-coded cryptosystem*, IEEE Trans. on Information Theory 35 (1989), 829 –833.

[44] T.R.N. Rao, *Joint encryption and error correction schemes*, International Symposium on Computer Architecture (ISCA), 1984, pp. 240 –241.

[45] J. Costello S. Lin, *Error control coding*, Prentice Hall, New Jersey, 1995.

[46] B. Schneier, *Applied cryptography*, John Wiley and Sons, 1996.

[47] ______, *Applied cryptography*, John Wiley and Sons, 1996.

[48] N. Sendrier, *Efficient generation of binary words of given weight*, Fifth IMA Conference on Cryptography and Coding, 1995, pp. 146–150.

[49] C. E. Shannon, *Communication theory of secrecy systems*, Bell System Technical Journal 29 (1949), 656–715.

[50] R. Struik and J. van Tilburg, *The rao-nam scheme is insecure against a chosen-plaintext attack*, Proc. Crypto'87, 1987, pp. 445–457.

[51] Q. Su and Y. Xiao, *Design of ldpc-based error correcting cipher*, Proc. of 2nd International Conference on Wireless Mobile and Multimedia Networks (ICWMMN), 2008, pp. 470–474.

[52] H. M. Sun, *Private-key cryptosystem based on burst-error-correcting codes*, Electronics Letters, vol. 33, November 1997, pp. 2035–1036.

[53] ______, *Ehancing the security of the mceliece public-key cryptosystem*, Journal of Information Science and Engineering, vol. 16, November 2000, pp. 769–812.

[54] H. Van Tilborg, *Coding theory at work in cryptology and vice versa*, Handbook of Coding Theory, 1998, pp. 1195–1227.

[55] I. Verbauwhede, P. Schaumont, and H. Kuo, *Design and performance testing of a 2.29-gb/s rijndael processor*, IEEE Journal of Solid-State Circuits, 2003, pp. 569–572.

[56] B. Weeks, M. Bean, T. Rozylowicz, and C. Ficke, *Hardware performance simulations of round 2 advanced encryption standard algorithms*, In AES Candidate Conference, 2000, pp. 286–304.

[57] A. D. Wyner, *The wire-tap channel*, Bell System Technical Journal 54 (1975), 1355–1387.

[58] K. Zeng, C. H. Yang, and T. R. N. Rao, *Cryptanalysis of the hwang-rao secret error correcting code schemes*, Third International Conference in Information and Communications Security, ICICS 2001, Lecture Notes in Computer Science, Springer-Verlag, vol. 2229, 2001.

[59] X. Zhang and K. Parhi, *High-speed vlsi architectures for the aes algorithm*, IEEE Trans. on very large scale integration (VLSI) systems 12 (2004), no. 9, 957–967.

[60] A. Zuquete and J. Barros, *Physical-layer encryption with stream ciphers*, IEEE International Symposium on Information Theory, 2008, pp. 106–110.