[UCI]I804:24011-200000263906

August 2013

Master's Degree Thesis

# Design of New Public Key Encryption Scheme Based on the Polar Coding

Graduate School of Chosun University

Department of Information and Communication Engineering

Sujan Raj Shrestha

August 2013

Master's Degree Thesis

Design of New Public Key Encryption Scheme
Based on the Polar Coding

Sujan Raj Shrestha

# Design of New Public Key Encryption Scheme Based on the Polar Coding

극부호에 기반한 새로운 공개키 암호화 방식 설계

August 23, 2013

Graduate School of Chosun University

Department of Information and Communication Engineering

Sujan Raj Shrestha

# Design of New Public Key Encryption Scheme Based on the Polar Coding

Advisor: Prof. Young-Sik Kim

This thesis is submitted to the graduate school of Chosun University in partial fulfillment of the requirements for a Master's degree in engineering

April, 2013

Graduate School of Chosun University

Department of Information and Communication Engineering

Sujan Raj Shrestha

This is to certify that the master's thesis of

Sujan Raj Shrestha

has been approved by examining committee for the thesis requirement for the Master's degree in engineering.

Committee Chairperson ____________

Prof. Seung-Jo Han

Committee Member ____________ May 30

Prof. Goo-Rak Kwon

Committee Member ____________

Prof. Young-Sik Kim

May, 2013

Graduate School of Chosun University

## 슈레스타 수잔 라흐의 석사학위 논문을 인준함

위원장 조선대학교 교수 한 승 조 印

위 원 조선대학교 교수 권 구 락 印

위 원 조선대학교 교수 김 영 식 印

2013년 5월

조선대학교 대학원

# Table of Contents

# List of Tables

# List of Figures

# Acronyms

| | | |
|---|---|---|
| AWGNC | : | Additive White Gaussian Noise Channel |
| BEC | : | Binary Erasure Channel |
| BSC | : | Binary Symmetric Channel |
| B-DMC | : | Binary-input Discrete Memoryless Channel |
| DEC | : | Decryption |
| ENC | : | Encryption |
| ECC | : | Error-Correcting Code |
| FS | : | Frozen Set |
| GE | : | Gaussian Elimination |
| IS | : | Information Set |
| ISD | : | Information Set Decoding |
| LLR | : | Likelihood ratio |
| PKC | : | Public Key Cryptography |
| RM code | : | Reed Muller code |
| RSA | : | Rivest, Shamir and Adleman Cryptosystem |
| SC | : | Successive Cancellation |

# 초록

## 극부호에 기반한 새로운 공개키 암호화 방식 설계

Sujan Raj Shrestha
Advisor: Prof. Young-Sik Kim, Ph.D
Department of Information and Communications Engineering
Graduate School of Chosun University

이 논문에서는 극부호 기반의 공개키 암호 시스템을 제안한다. 제안된 시스템은 오류 정정 부호에 기반을 둔 McEliece 암호 시스템으로, 랜덤한 선형 부호의 복호의 어려움에 기반을 둔 시스템이다. McEliece 암호는 가장 오래된 공개키 암호 시스템 주 하나로서 30년 이상의 암호 해독 시도에도 불구하고 적절한 파라미터를 선택하면 McEliece가 초기에 제안한 시스템은 여전히 안전한 것으로 평가받고 있다. 그러나 공개키와 비공개키의 크기가 너무 크기 때문에 RSA와 타원 곡선 암호 시스템에 비해서 실용성이 떨어지는 것으로 평가받았다.

따라서 다른 오류 정정 부호를 사용해서 공개키 크기를 줄이기 위한 많은 연구들이 진행되었다. Sidelnikov는 이진 Reed Muller (RM) 부호를 이용해서 또 다른 McEliece 암호 시스템을 제안하였고 이것은 Sidelnikov 암호 시스템으로 알려져 있다. RM 부호 기반의 암호 시스템은 McEliece의 초기 제안에 비해서 더 작은 크기의 파라미터를 갖고 더 효율적인 복호 알고리즘이 존재하지만, Minder와 Skrollahi의 의해서 RM 부호의 대수적 구조를 사용한 공격에 의해 2006년에 해독되었다.

극부호는 Arikan이 제안한 새로운 형태의 오류 정정 부호로서 정보이론적으로 주어진 채널 용량을 점근적으로 달성할 수 있는 것으로 증명된 부호로 RM 부호와 유사한 구조

를 갖는다. 기존의 오류 정정 부호와는 달리 극부호는 주어진 채널에 의존적이며 채널 합성과 분리 과정을 통해서 유도되는 채널 양극화 현상을 이용한다. 채널 양극화 현상의 결과로서 어떤 채널들은 매우 좋은 채널이 되고 다른 채널은 매우 나쁜 채널이 된다. 따라서 나쁜 채널로는 고정된 비트가 전송되고 좋은 채널로는 사용자 데이터를 전송하게 된다.

이 논문에서는 극부호에 기반을 둔 새로운 공개키 암호시스템을 제안한다. 극부호를 사용해서 평문을 어떻게 부호화 하며 오류를 추가하여 데이터를 숨기는 방식을 보일 것이다. 또한 수학적 분석을 통해 보안 수준을 평가하고 시뮬레이션을 통해서 비밀키를 알지 못할 경우에 어떠한 데이터도 복구할 수 없음을 보일 것이다.

# Abstract

## Design of New Public Key Encryption Scheme Based on the Polar Coding

Sujan Raj Shrestha
Advisor: Prof. Young-Sik Kim, Ph.D
Department of Information and Communications Engineering
Graduate School of Chosun University

In this thesis, cryptosystem is proposed where information is transmitted over the insecure channel after encoding plain-text using the public key which is generated using the polar code. That is, the proposed system is a class of the McEliece cryptosystem, the error correcting code based cryptography, whose security is based on the difficulty of decoding of random linear codes. The McEliece cryptography is one of the oldest public key cryptography scheme and McEliece's original proposal is known to be secure with adequate selections of related parameters even with the intensive efforts of cryptanalyses on that for over 30 years. However, since the secure sizes of public and private keys are too large, it has been considered as less practical suggestion than the other public key cryptographic schemes such as RSA or the elliptic curve cryptography, shortly ECC. Therefore, many researches have been carried to reduce public key size by using other error-correcting codes.

Later, Sidelnikov proposed another McEliece cryptosystem which uses the binary Reed Muller (RM) code, also called as Sidelnikov cryptosystem. Although RM code based cryptosystem has smaller size of parameters and more efficient decoding algorithm than McEliece's original scheme, it was broken using the algebraic structure of the RM code by Minder and Skrollahi.

The polar code is a new kind of error correcting code which is information theoretically proved to asymptotically achieve the capacity of the given channels by Arikan and has similar structure to the RM codes. Contrary to the previous error correcting codes, the polar code depends on the given channel and utilizes the channel polarization which can be induced by using channel combining and splitting process. As a result of the channel polarization, some channels turn into good channels and the others into bad channels. While fixed bits are sent through the bad channels, user information is transmitted through good channels.

In this thesis, the polar code based public key cryptography is proposed and shown how polar code can be used to encode plaintext message and hide it by adding errors. Decoding method is shown to obtain plaintext from ciphertext using private keys and decoding.

# I. Introduction

Public-Key cryptosystem is widely used to encrypt and decrypt message for secure transmission which uses different keys for encrypting and decrypting messages, respectively. Among the 2 distinct keys which are known as public and private keys, the first is open to public and is shared via public channel while the last should be kept secret. That is why public key cryptosystem is called the 'asymmetric key encryption'. When a plaintext is encrypted using the receiver's public key, the decryption of ciphertext is computationally infeasible without the knowledge of the private key of the legitimate receive. Therefore, eavesdroppers cannot recover plaintext from the ciphertext.

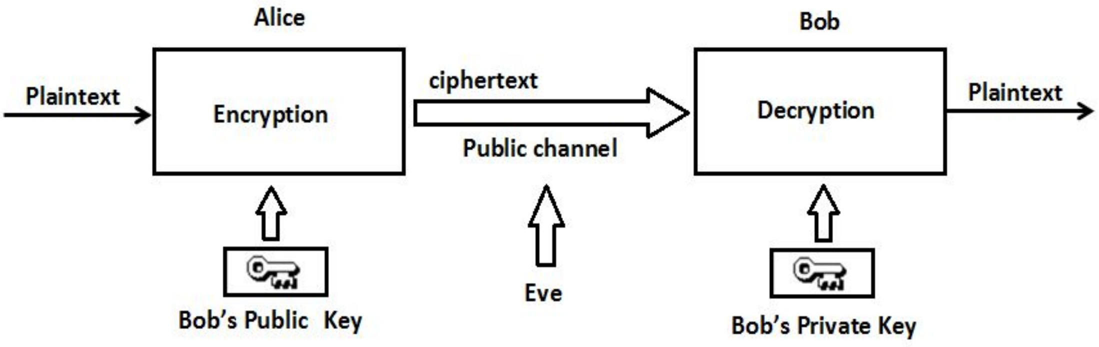


Fig. 1.1 Public Key Cryptosystem Block Diagram

The public key cryptography is depicted in Fig. 1.1. In Fig. 1.1, the sender (Alice) wants to send a secret message to the receiver (Bob) through public channel. The eavesdropper, Eve, tries to obtain the information sent by Alice in an unauthorized manner. For the public key encryption, Bob produces his public and private keys. Then, Bob makes his public key known to every one, but he keeps the private key in secret. There is mathematical relation between the public key and the private key. Hence if the message is encrypted with public key, then it can be decrypted by using only a particular private key.

Alice encrypts a confidential information using the Bob's public key and transmits the ciphertext through the public channel. After receiving the ciphertext, Bob decrypts it using his private key. Eve also has access to ciphertext transmitted by Alice. However, without the knowledge of the Bob's private key, she cannot decrypt the ciphertext.

RSA algorithm was developed by Rivest, Shamir and Adleman [26] in 1977. RSA algorithm relies in difficulty of factoring large integer numbers. In RSA cryptosystem, public key is $(n, e)$ where $n$ is large integer n whose factors are 2 prime numbers $p$ and $q$, and integer $e$ is co-prime to Euler's totient function of $n$, $[\varphi(n)]$. Private key is $(n, d)$ where integer $d$ is multiplicative inverse of $e$ in modulus of $\varphi(n)$. Numbers $d$ and $e$ are related as:

$$d.e \equiv 1(\mathrm{mod}\,\varphi(n)) \ . \tag{1.1}$$

Alice encrypts plaintext $m$ into ciphertext as:

$$c \equiv m^e(\mathrm{mod} n) \tag{1.2}$$

When Bob receives, ciphertext, he deciphers plaintext from ciphertext in following way:

$$m \equiv c^d(\mathrm{mod} n) \tag{1.3}$$

For adequate security, the size of key in RSA is 1024 to 2048 bits.

Elliptic Curve Cryptography (ECC) is based on algebraic structure of elliptic curve over finite fields. It was suggested independently by Koblitz [33] and Miller [34] in 1985. Hardness assumption of elliptic curve cryptography is based on difficulty of finding discrete logarithm of random elliptic curve element. The key size given by elliptic curve cryptography is smaller than RSA for same security level. ECC with 256 bit key length provides same level of security as 3072 bit RSA cryptosystem. Currently, the elliptic curve being used consists of points satisfying equation

$$y^2 = x^3 + ax + b \ . \tag{1.4}$$

The private and public keys are made from points that lie in the curve which is agreed upon by both Alice and Bob.

However, it was proved that the Shor's algorithm in a quantum computer can solve the mathematical problems such as the integer factoring and discrete log problem (DLP) and elliptic curve discrete log problem (ECDLP) in polynomial time [27]. Fortunately, quantum computer has not yet been realized up to manipulating practical parameters. However, considering the significant efforts to build a practical quantum computers, it is considered that within 15 years, RSA and ECC cannot be used for the public key cryptography any more. Therefore, there are studies to find alternative public key cryptography which is still safe with the quantum computations. One of the candidates after the realization of quantum computers is the code-based cryptography, also known as McEliece cryptosystem [2].

## A. Thesis Motivation and Overview

Various error correcting code based cryptosystems has been proposed. In his original proposition [2], McEliece constructed a cryptosystem using the binary Goppa codes multiplied by a random permutation matrix and a non-singular matrix. Even with the intensive efforts of the cryptanalysis [6], the McEliece's original proposition is known to be secure with the appropriate security parameter sizes. However, the sizes of the security parameters of the McEliece's cryptosystems are too large when comparing with other conventional public key cryptosystems such as RSA and elliptic curve cryptography. Therefore, many researches are carried out to reduce the size of public and private keys [2],

[7], [11], [21], [29], [30].

Niederreiter constructed a variation of McEliece cryptography using the generalized Reed Solomon (RS) codes and its parity check matrix instead of the generator matrix in order to reduce public key size. Although the original Niederereiter cryptography was broken [19], the binary Goppa code based Niederereiter scheme is known to be still secure and was shown that security of Niederreiter cryptosystem is equivalent to McEliece cryptosystem [28].

Later, Sidelnikov proposed another McEliece type cryptosystem using the binary Reed-Muller (RM) codes with the shorter security parameters and efficient decoding method [7]. However, the Sidelnikov cryptosystem also was broken by Minder and Skrollahi [4] by using algebraic structure of the RM codes.

## B. Research Objectives

The primary objective of this thesis is to construct a McEliece-type encryption scheme based the polar code. The ordinary generator matrix is scrambled and permuted using a non-singular matrix and permutation matrix of suitable size.

It is shown how plain-text message is encoded using randomized polar code and how it is deciphered. In addition, it is shown that decoding without the private key cannot disclose the original plaintext by using numerical simulations. Investigate for the amount of operations required to reveal secrecy by the brute force attacks is done. Furthermore, by comparing the structure of the polar code with that of the RM code, it is shown that the cryptanalysis for the Sidelnikov cryptosystem cannot be applied to the proposed system.

## C. Thesis Contribution

In this thesis, a new cryptosystem which is based on the polar code is presented. Different generator matrix which is scrambled and permuted is used as the public key and non-singular matrix and permutation matrix is used as private keys. Thus, when the message vector is encoded with generator matrix and random error is added due to polarization, then we get randomized data out of which the eavesdropper cannot draw out any meaningful information. The correct decoding can only be performed when the correct private keys are supplied. The main contributions of this thesis are as follows:

**New cryptosystem:** Cryptosystem is constructed for the encryption and decryption of the message vector of size $(1\times k)$. Decoding method and parameter for the correct decoding of the randomized message is presented.

**Design Procedure:** Design procedure is given in order to find the important parameters of this algorithm. Construction of generator matrix and decoding is carried out by method provided in [1]. Method for finding non-singular matrix is provided using Gaussian Elimination. For a given application with its specifications and requirements, an engineer can follow the steps in this procedure to make cryptosystem using polar code.

**Simulation:** Simulation code was written in MATLAB 2010 to implement the cryptosystem and test it without using private keys when decoding.

## D. Thesis Organization

This thesis is organized as follows: In Section II, brief introduction to McEliece cryptosystem is presented with its hardness assumptions and existing McEliece cryptanalysis. Major cryptanalysis which is improvements over Information set decoding are described in short. This is followed by introduction of Reed-Muller code and Sidelnikov cryptography. Then, Sidelnikov cryptanalysis is explained in brief. In later part of Section II, it is discussed in detail about the polar code and aspects of polar coding like its properties and construction. In Section III proposed cryptosystem is constructed. Security Analysis is presented and discussed in Section IV. Based on the security analysis, appropriate size of public and private key is proposed. Section V concludes this thesis.

# II. Background

## A. McEliece Cryptosystem

In this section, original McEliece cryptosystem [2] which uses the binary Goppa codes is explained.

### 1. Basic idea

McEliece public key cryptosystem is the first public key cryptosystem that uses the error-correcting codes for encryption and decryption of plaintext into ciphertext and ciphertext into plaintext respectively. Public and private key parameter and encoding and decoding procedures are discussed in short below.

#### Public Key

The public key is generator matrix $G_{pub} = SGP$ of dimension $(k \times N)$ and the integer 't' which is the number of error the Goppa code C can correct. Here $t = (d_{\min} - 1)/2$ . $d_{\min}$ is minimum distance of binary Goppa code $C$. $S$ is $(k \times k)$ dimension non-singular scrambling matrix, $G$ is the $(k \times N)$ generator matrix of $(k, N)$ the goppa code $C$ used for encoding and $P$ is the $(N \times N)$ permutation matrix.

#### Private key

The first private key is non-singular matrix $S$ of dimension $(k \times k)$, and second private key is permutation matrix, $P$ having dimension $(N \times N)$. Next private key is a secret decoding algorithm for binary Goppa code.

### Encryption

$C$ denotes binary goppa which is used in McEliece cryptosystem[2]. Alice encodes message using Bob's public key which is scrambled and permuted generator matrix of Goppa code.  After plaintext message has been encoded, it is converted into ciphertext by addition of error vector of weight at most $t$ at random positions. The cryptosystem is designed to correct at most $t$ errors from the code to recover plaintext. If message $m$ is encrypted then the ciphertext is shown in equation (2.1).

$$Y = m\,G_{Pub} + e(t) \tag{2.1}$$

$$Y = m(SGP) + e(t) \tag{2.2}$$

### Decryption

Decryption in McEliece cryptosystem is done in following way : Bob first reverses the permutation so that before applying the secret decoding algorithm, he can obtain the pseudo-message in correct order. Reversing of the permutation is done by applying the matrix which is the inverse of the private key $P$ that was used in the encryption process. This gives expression in equation (2.3).

$$YP^{-1} = m(SG) + eP^{-1} \tag{2.3}$$

Bob then uses his secret decoding algorithm to exclude out the random errors from ciphertext. So after decoding he obtains $m' = mS$.

Bob uses his next private key that is non-singular matrix $S$. From $S$, it's inverse $S^{-1}$ can be easily obtained. Multiplying $m'$ with $S^{-1}$ removes the scrambles from the message. Finally what remains is the original plaintext itself. Above performed operation is $m = (mS)S^{-1}$ .

One thing that we have to note here is that the decoding algorithm must be efficient and able to correct $t$ errors from the ciphertext. In the absence of efficient decoding algorithm, the correct message cannot be recovered from ciphertext.

### Scrambling matrix

Scrambling matrix is a randomly generated $(k \times k)$ non-singular matrix. For McEliece-type cryptosystem, scrambling must always be done by non-singular matrix. By non-singularity it means that $S$ must have an inverse $S^{-1}$ and product of $S$ and $S^{-1}$ must always be equal to identity matrix. If $C$ is the error-correcting code, then multiplying scrambling matrix with generator matrix of code $C$ produces another code $C'$ which is equivalent to code $C$.

### Permutation matrix

Permutation matrix is the matrix that changes the columns of the equivalent code of the $C$. This adds further redundancy to the structure of the generator of code $C$ and when the message produces the codeword, the code bits are not in the original systematic form. Permutation matrix is always constructed by random permutation of identity matrix of given size. Hence, in every rows and columns, there is always single ‘1’. Any matrix that doesn’t satisfy this condition is not considered as permutation matrix. The inverse of permutation matrix $P^{-1}$ is constructed by taking the transpose of the matrix $P$. And product of $P$ and $P^{-1}$ must always be identity matrix.

## 2. Hardness Assumption

There are two hardness assumptions of McEliece Cryptosystem. They are:-

1. It is hard to determine exact position of $t$ errors that are added to the codeword. Error is added at random positions every time ciphertext is sent. Error addition in fixed position allows Eve to determine position of error by some analysis but random error in large block length makes this process infeasible.

2. Given the public key matrix $G_{Pub}$, attacker cannot efficiently compute private key elements. This kind of attack is termed as **Structural Attack**. For McEliece PKC, apart from exception of some weak keys, structural attacks are ineffective due to large cardinality of possible permutation, generator and scrambling matrix. Also the number of effort required to exactly compute invertible scramble $S$ and permutation matrix $P$ surges to astronomically high values due to their large dimensions.

## 3. Parameters

The parameters of the McEliece cryptosystem are $N = 1024, k \geq 524, t = 50$ , $d_{\min} = 101$ . With these parameters, the size of the public key is 67,072 bytes. The transmission rate is 0.512.

## B. Existing McEliece Crypto-analysis and Variants

Various cryptanalysis methods for the McEliece and McEliece-type cryptosystems have been attempted. For McEliece cryptosystem, finding error is described in [6], [8], [9], [20], [31], [32]. In these algorithm, given the encrypted word $\left[ Y= mG_{Pub}+ e \right]$ , error vector which is minimum weight word is determined. This decoding method is known as decoding attack because it is equivalent to decoding a linear code. In this method and its other variants, basic principle is to append ciphertext into the public generator matrix. The generator matrix is then written as

$$\begin{bmatrix} G_{pub} \\ mG_{pub}+ e \end{bmatrix} .$$

Here, if $G_{pub}$ is the generator matrix of McEliece cryptosystem, then the new generator matrix shown above is represented as $C+ \{0, y\}$ . Due to the above transformation, minimum weight codeword is the error word $e$ having weight at most $t$ and this is the only word of such weight. The error-word is represented as

$$e= (m, 1)\begin{pmatrix} G_{pub} \\ mG_{pub}+ e \end{pmatrix} .$$

Here an algorithm known as **Information Set Decoding** is discussed. Here, the assumption is that we are working in the $[N, k]$ code $C$. $k$ columns out of $N$ columns from the generator matrix is randomly selected and generator matrix is diagonalized in these $k$ positions. If the selected rows cannot be converted into unit matrix, then we have to choose another set of k information set and repeat the process.

Upon conversion, the row containing the word of weight $t$ is found out. This is the minimum weight word. In this, the maximum weight of the rows is $(N- k+ 1)$ . The probability of finding word of weight $t$ is given in following way

$$\frac{\binom{k}{1}\binom{N-k}{t-1}}{\binom{N}{t}}.$$

### Lee and Brickell’ s Method

Lee and Brickell’s algorithm [20] works in following ways: Originally McEliece proposed finding codeword by applying Gaussian elimination of generator matrix $G$. Then the corresponding permuted permuted ciphertext is divided into two parts. If the first part of the ciphertext has no error and second has error $t$ then the vector $e$ is minimum weight word. Instead of finding the weight 0 codeword in first part, partial error of weight less than or equal to $p$ is found in the first part and another part contains the remaining weight of $e$. This reduces the time of finding minimum weight error.

### Leon’ s Method

Leon’s algorithm [31] has two parameters σ and p. The method is as follows- Leon chooses random selection of S consisting of $(k+\sigma)$ columns and places it on the right end. Then he applies Gaussian elimination so that resulting matrix has three sub-matrix $B(N-k-\sigma, e)$ , $Z(k+\sigma-e, e)$ , $D(N-k-\sigma, k-e)$ and $I_e$ for some $e \leq (k+\sigma)$ . Then he looks for linear combination in such a way that codewords $m$ has weight less than $p$ in $S$. This is achieved by considering the single matrix $Z$. In the case this weight is less than $p$, he computes the corresponding $N$ bit word and verifies its weight is less than the weight of the previously obtained shortest word. If $e$ is not equal to zero, this test must also be performed for the codewords that include D-codewords. For this method, the parameters used are $p=2$ and $\sigma=2$ .

### Stern’s Method

Stern's algorithm [32] is method using the parity-check matrix so it is slightly different from the above two methods. According to Stern, first the set $I$ consisting of $(N-k)$ columns is chosen from the parity check matrix $H$. This sub-matrix is shifted to is diagonalized in the form as $H' = (Q \mid I_{N-k})$. Columns of sub-matrix $Q$ is divided into two parts $X$ and $Y$ so that each columns are allowed to freely join either $X$ or $Y$ independently. This gives the form $H' = (X \mid Y \mid I_{N-k})$. Then $l$ rows are chosen to from set $Z$. Then Stern computes for each p column subset A from group $X$ the sum of $l$ rows to form $l$ bit vector $\sigma_l(P_X)$ and for each $p$ column subset B from group $Y$ the sum of $l$ rows to form $l$ bit vector $\sigma_l(P_Y)$. Then he checks if the two sums are equal. If $\sigma_l(P_X) = \sigma_l(P_Y)$, then he selects the $2p$ columns $(A \cup B)$ and computes the sum of these columns from $(N-k)$ rows. If the sum has weight $(w-2p)$, then it is possible to create codeword of weight $w$. The column $\sigma_l(P_X) + \sigma_l(P_Y)$ is returned.

If this fails, then entire step has to be repeated by selecting next set of $(N-k)$ columns.

## C. Reed-Muller Code And RM Cryptosystem

This Section discusses Reed-Muller code and Sidelnikov cryptosystem [7] that uses RM code for encoding and decoding plaintext. In first part, Reed-Muller is briefly presented and in second part Sidelnikov cryptosystem is presented.

## 1.Reed-Muller Code

Reed Muller code is denoted by $R(r,m)$, where length of the code is $N=2^m$ and $r$ is the degree of the Reed-Muller code. The dimension of Reed-Muller code is represented by $k$ which is written as

$$k=1+\binom{m}{1}+\binom{m}{2}+....+\binom{m}{r} \tag{2.4}$$

Reed-Muller code is defined by boolean functions of degree at most m and consists of variables of degree 1 from $v_1$ to $v_m$. For every increment in $r$, $\binom{m}{r}$ rows are added to the previous dimension $(r-1)$. Hence from this property, it can be said that RM code holds the relationship $R(0,m)\subset R(1,m)\subset .....\subset R(m,m)$ . Each word of Reed Muller code is defined by boolean functions. The words in $R(1,m)$ consists of m boolean functions $f_1,f_2,.....,f_m$. For the degree $r$, where $r>1$, boolean function $f$ is product of any $r$ boolean functions from $v_1$ to $v_m$ and is denoted as

$$f=f_1.f_2.....f_r \tag{2.5}$$

In RM code $R(r,m)$, minimum weight word is always of degree $r$. And the minimum weight word of degree $r$ has weight $2^{m-r}$. Symbol $(.)$ is dot operator. $f$ is element by element product of $r$ variables. There are total of $\binom{m}{r}$ minimum weight words in the generator matrix of Reed-Muller code of degree $r$. From construction of Reed-Muller code, it is clear that value of $r$ is always less than or equal to m.

In Fig. 2.1, structure of $R(3,3)$ code is shown. In this figure, the variables $x_1, x_2, x_3$ are boolean functions of degree 1. Variable $x_1x_2$, $x_1x_3$, $x_2x_3$ are made up of boolean function of degree 2. Variable $x_1 x_2 x_3$ is made of boolean function of degree 3. In general, when $r$ is given, lowest weight words are given by equation (2.5).

Reed-Muller code can also be constructed using the generator matrix in a way whose construction is different from the one above. Here, we denote the generator matrix of $R(r,m)$ by $G(r,m)$.

$$G(r,m) = \begin{bmatrix} G(r,m-1) & G(r,m-1) \\ 0 & G(r-1,m-1) \end{bmatrix} \tag{2.6}$$

$G(r,m-1)$ is generator matrix of $R(r,m-1)$ and $G(r-1,m-1)$ is generator matrix of $R(r-1,m-1)$.

$$R(3,3) = \left( \begin{array}{ccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x_1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ x_2 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ x_3 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ x_1x_2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_1x_3 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ x_2x_3 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ x_1x_2x_3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right)$$

Fig. 2.1 Generator matrix of R(3, 3) Reed-Muller code

## 2. Sidelnikov Cryptosystem

Sidelnikov cryptosystem [7] uses RM code for encoding and decoding plaintext. And unlike McEliece cryptosystem, with knowledge of length and dimension of public generator matrix, the basis of generator matrix can be constructed.

### Public Key

The public Generator matrix is $G_{pub}= SGP$. where $G$ is generator matrix of polar code having dimension $(k\times N)$, $S$ is $(k\times k)$ invertible matrix, and $P$ is $(N\times N)$ permutation matrix. With knowledge of public key alone, it is not possible to construct private key which is matrix $S$ and $P$.

### Private Key

Private key of the receiver is $(k\times k)$ matrix $S$ and permutation matrix $P$ with dimension $(N\times N)$.

### Encryption

Encryption is done using the same technique as in the McEliece cryptosystem. Message has to be of length $k$. This message is encoded using randomized RM code $G_{Pub}$ and error of pre-determined weight is added.

Size of error is determined by the minimum distance $d_{\min}$. Number of error that RM code can correct is at most $t= (d_{\min} - 1)/2$. But in [10] Sidelnikov proposed new decoding algorithm that can almost always correctly decode error whose weight is $t> (d_{\min} -1)/2$. Hence if the decoding algorithm proposed by Sidelnikov [10] is used, then more error bits can be added to increase security. Therefore the ciphertext $[b= m\,G_{pub}+ e]$ is sent to the receiver through unsecured channel with added security.

### Decryption

In decryption, the Bob receives the message $b$. Bob has complete knowledge of private keys so from the permutation matrix, its inverse permutation $P^{-1}$ is easily computed by transposing matrix $P$. This matrix removes the permutation that was applied in the encoding. Removal of permutation is done by multiplying $b$ with $P^{-1}$. From this, Bob obtains $b' = bP^{-1}$. In the next step, error-correcting algorithm that almost always correctly decodes error is applied. In [7], the decoding algorithm in [10] is applied. This is because the algorithm in [10] is able to decode higher number of errors than the $(d_{\min} - 1)/2$.

From the above step, only scrambled message $m' = mS$ is obtained. For the extraction of actual plaintext the randomized scrambler also has to be removed. This is done by inverse of matrix S which is matrix $S^{-1}$. When matrix $S^{-1}$ multiplies $m'$, the real information is extracted. Obtaining message $m$ is written mathematically as $m = m' S^{-1}$.

## 3. Attack Against Sidelnikov Cryptosystem

Sidelnikov Crytosystem was broken by Minder and Skrollahi  in [3] and [4]. They have shown in [3] and [4] that there is structural weakness in Reed-Muller code that makes Sidelnikov cryptosystem unsuitable for cryptographic purpose. In [7], Sidelnikov proposed using $R(3,10)$ as the encoding matrix. This is a very low rate code. Altogether, there is total of only 176 rows. This feature also makes ideal situation for applying low weight word finding algorithms to decode minimum weight word.

Another structural weakness of RM code is that the higher order basis is built from low order word which is already mentioned in the description of Reed-Muller code. This property is utilized in breaking Sidelnikov cryptosystem. Below, description of attack against Sidelnikov cryptosystem is discussed in detail.

When attacking Sidelnikov cryptosystem, main goal is to find a permutation $q$ and when this permutation is applied to the permuted and scrambled matrix $R_{SP}(r,m)$, resulting matrix has to be the generator of $R(r,m)$ code. The method of attack can be summarized below:

1. Find minimum weight words from random basis of given permuted $r^{th}$ order Reed-Muller code using low weight word finding algorithm. For this purpose, the algorithm used in [3] and [4] is Canteaut and Chabaud algorithm from [9]. Total number of codewords of degree $r$ is $2^{mr-r(r-1)}$. Hence this operation has to be done $2^{mr-r(r-1)}$ times to find all such words.

2. Find a codeword in $R_{SP}(r,m)$ that belongs to the subcode $R_{SP}(r-1,m)$. When this is found out, other codewords belonging to this subcode also has to be found out. There are $\binom{m}{r-1}$ such words that has to be found out.

3. When all codewords of $\binom{m}{r-1}$ are found, same process is repeated again by decreasing the value of $r$ by 1 and finding codewords that belong to the subcode of that lower dimension. This must be repeated until $(r=1)$ is reached. All codewords words of $R_{SP}(1,m)$ must be found out.

4. Then find permutation $q$. When permutation $q$ permutes columns of $R_{SP}(1,m)$, it should be equal to $R(1,m)$.

Finding factors of minimum weight words is the hardest part of this algorithm. It is known that the RM code of high degree are created by multiplication with low degree variables. Hence the words can be split into its factors. Once the factors are found out, it is almost done. Here, $w$ is the minimum weight word. So $w = v_1 \cdot \ldots \cdot v_r$ . For this $C_{Supp(x)}$ must be determined. $C_{Supp(x)}$ is the subcode of $R(r,m)$ that contains only those words which are zero on support of $x$ and these positions are punctured later. For supposition, $f$ is the codeword in the shortened code $C_{Supp(x)}$. Then $f$ can be written as :

$$f(v_1,...,v_r,\bar{v}) = \sum_{I \in 2^{1,\ldots,r}} f_I(\bar{v}).\prod_{i \in I} v_i, \tag{2.7}$$

For $I \subseteq \{1,\ldots.,r\}$ , $F_I$ is the codeword of degree $r - |I|$ , and variable $v_{r+1},\ldots,v_m$. From the previous assumption, the condition for this word $f$ to be on the support of minimum weight word is

$$0 = \sum_{I \in 2^{1,\ldots,r}} f_I(\bar{v})) \tag{2.8}$$

The shortened code is now the concatenated code and this shortened code satisfies condition $C_{Supp(x)} \subseteq R(r-1,m-r) \times \ldots. \times R(r-1,m-r)$ . The inner codes or block decomposition of the support of shortened codes is found out using the Sendrier's Algorithm which is described in [17]. The product $R(r-1,m-r)$ continues $(2^r - 1)$ times. Since Lorenz's method shortens on the set $\{V_1 = V_2 = \ldots.. = V_R = 1\}$ , there are $(2^r - 1)$ such sets and each has length $2^{m-r}$. To find the sets, the description is provided in [3] that uses Sendrier's algorithm [17]. In [4], Lorenz has used different technique. He used technique that works on the code itself.

The inner codewords that are recovered have the property as follows: each of the codewords lie in the set $\{V_1 = V_2 = \ldots\ldots = V_R = 1\}$ or the support of minimum weight word and the set S which is the block decomposition of concatenated codes. An example from [4] is presented. Let one of such set $S$ be $\{V_1 = 0, V_2 = 0, \ldots\ldots = V_l = 0, V_{l+1} = 1, V_{l+2} = 1, \ldots\ldots, V_r = 1\}$. The codeword that satisfies this condition is

$$y = (1 + V_1 + V_2)(1 + V_2 + V_3)\ldots(1 + V_{l-i} + V_l).V_{l+1}\ldots.V_r .$$

Here $\deg(y) = (r-1)$. So in fact, a word of degree $(r-1)$ has been constructed.

After finding permuted codewords of $R(1,m)$, next step is to find permutation $q$ which rearranges the columns of matrix into orderly form. This is shown as

$$(R(1,m)^p)^q = R(1,m)$$

For this, simple procedure described in [3] is followed. Codeword containing all 1 is discarded and only m variables are taken. When $m$ variables are in order, then by construction of $m$ variables, reading from column 1, each column $i$ is $(i-1)$ in binary equivalent. Permuted columns of $m$ boolean variables are rearranged so that they can be in the above form.

## D. Polar Code

### 1. Overview

Polar code discovered by Arikan [1] is the first code that is proven to achieve capacity of the binary-memoryless symmetric channel (B-DMC). If $N$ channels are polarized, then as a result there are a fraction of channels whose capacity tends to 1 and are reliable channels. And there is another set of channels whose capacity tends to 0. Those channels whose capacity tends to 1 is near to $I(W)$ while those that has capacity close 0 tends to $1-I(W)$. Performance of these codes increases with the increase in the number of channels. Hence for achieving near perfect polarization, greater length is always favored.

If $I(W)$ is the channel capacity of binary symmetric channel, then assuming that 1 and 0 occur with equal probability, channel capacity can be written in terms of transition probability of $x$ and $y$ as :

$$I(W)=\Sigma\Sigma\frac{1}{2}W(y|x)\frac{W(y|x)}{\frac{1}{2}W(y|0)+\frac{1}{2}W(y|1)} \tag{2.9}$$

Here, the term $W(y|x)$ is conditional probability of $y$ given $x$. Reliability is given by Bhattacharya parameter and is :

$$Z(W)=\sum_{y\in Y}\sqrt{W(y|0)\,W(y|1)} \tag{2.10}$$

The summation is carried out over all the possible values of output symbols

$y$. Higher value of $Z(W)$ indicates that there is high chance the sent symbols is likely to be corrupted and therefore capacity of that channel is bound to be low. On the other hand, lower value of $Z(W)$ indicates that sent symbol is less likely to be corrupted when received. Similarly, it is shown in [16] that rate can also be similarly manipulated. But the reason why channel capacity is more favored is because rate can be created but channel capacity cannot be created.

## 2. Channel Polarization

Channel polarization is the process of transforming $N$ number of channel in such a way that after transformation, each of them has different capacity and reliability parameter when viewed independently. Therefore, we can view channel polarization as a step that manipulates identical channels to create another set of channels, some of which has higher capacity than the original channel while others has capacity that is degraded than the original channels. Since the capacity of the channels are improved as well as degraded, uncoded bits can be sent through the good channels and none of the bits can be sent through the bad channels. And the method of achieving this is called polar coding.

In the following section, description of steps necessary for creating the channel polarization phenomenon is given.

### Channel Combination

In channel combination, channel is combined with another identical channel. But here, both the channels must possess equal capacity. When this is done for large number of channels, this process becomes recursive. When the two channels are combined, the transition probability of combined channel is equal to expression shown in equation (2.11).

$$W_2(y_1, y_2|u_1, u_2) = W(y_1|u_1 \oplus u_2)\, W(y_2|u_2) \qquad (2.11)$$

The above expression is for the combination of 2 independent channel. The operator $(\otimes)$ is called as Kronecker product which is used for calculating the kronecker product of any two matrices.

Equation (2.11) can be generalized to any number N which is a power of 2. Generalized form is expressed as shown in equation (2.12)

$$W_N(y_1^N | u_1^N) = W^N(y_1^N | u_1^N G_N) \qquad (2.12)$$

The $G_N$ is called as Generator Matrix.

## Channel Splitting

In channel splitting the combined channels shown above is split into different channels. Channel splitting is necessary because without splitting the channels, we cannot get the independent transformed $W_N$ channels that were combined in the previous Section. These split $N$ channels now have different capacity from the original channels. After splitting, transition probability of any $i^{th}$ channel expressed in [1] is given by equation (2.13) as

$$W_N^{(i)}(y_1^N, u_1^{i-1}|u_i) = \sum_{u_{i+1}^N} \frac{1}{2^{N-1}} W_N(y_1^N | u_1^N) \qquad (2.13)$$

## 3. Rate And Reliability Analysis

In this section, we see how channel capacity and the Bhattacharya parameter transforms under the channel polarization. The measure of rate is channel capacity and the measure of reliability is the Bhattacharya parameter. In case of BEC, the rate and reliability parameters can be computed recursively. For the rate, the expression from [1] are given by equations (2.14) and (2.15) as:

$$I(W_N^{(2i-1)}) = I(W_{N/2}^{(i)})^2 \tag{2.14}$$

$$I(W_N^{(2i)}) = 2I(W_{N/2}^{(i)}) - I(W_{N/2}^{(i)})^2 \tag{2.15}$$

Here, the recursive process stops when we reach at $I\left(W_1^{(1)}\right)$ which is the capacity of the untransformed channels. Its calculation varies depending upon the kind of channel that we are considering. For BEC, $I(W_1^{(1)}) = (1-\epsilon)$. Similarly, the reliability transformation happens in the following way as shown in equations (2.16) and (2.17) from [1] :

$$Z(W_{2N}^{(2i-1)}) \le 2Z(W_N^{(i)}) - Z(W_N^{(i)})^2 \tag{2.16}$$

$$Z(W_N^{(2i)}) \le Z(W_N^{(i)})^2 \tag{2.17}$$

Fig. 2.2 below shows the reliability of polarized channels for $N = 2^{10}, N = 2^{15}, N = 2^{20}$ . Fig. 2.3 (a) and 2.3 (b) shows channel capacities of individual channel after they are polarized. This graph is for $N$ = 256 and $N$ = 1024 channels. From this we can make conclusion that as the block length increases, more channels tend to polarize perfectly.

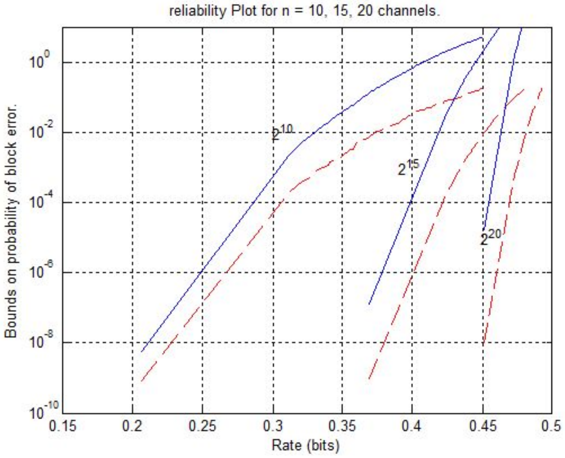
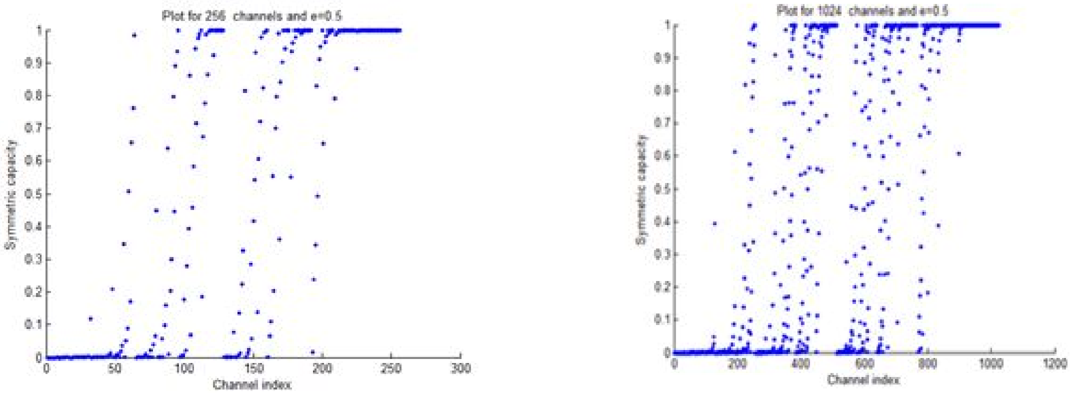


Fig. 2.2 Rate and reliability for block length 10, 15, 20



(a)

(b)

Fig. 2.3 Channel capacities for block length (a) 256 and (b) 1024

# 4. Polar Coding Method

In this section, we see how the polar encoding is done in order to achieve the channel polarization as it was originally shown by Arikan [1]. These frozen vectors are known beforehand to the decoder and will be used in the decoding procedure. But we have must know one thing that the choice of the frozen bits does not affect the decoding procedure. Hence, we are free to choose any random vector as frozen vector. In polar encoding, information bits and frozen bits are encoded by different set of generator matrix. In the next section some of the important terminologies for the polar codes are described in short.

### Information Set

Information set consists of those indices of the polarized channels that have capacity greater than or equal to the threshold value that are chosen for sending the information bits. These constitute reliable channels. When we send information from them, probability that the bit is corrupted is very small. The generator matrix that encodes the information bit is constituted by selecting the rows from information set.

### Information Vector

Information vectors represent the bits that we have to transmit. It is represented as $u_A$. These are sent through the reliable channels via information set. Therefore, they have to be encoded using the sub-matrix of the generator matrix whose rows are selected in certain fashion that is described below.

### Frozen Vectors

Frozen vector are the pre-determined values or the bits that are sent through the unreliable channels which are not part of the information set. It is represented as $u_{A^C}$.

### Generator matrix

Generator matrix encodes information bits and the frozen bits separately. The generator matrix in polar codes can be divided into two types :

1. Generator matrix for the Information bits

This is represented as $G_N(A)$. This is the sub-matrix of the generator matrix $G_N$ and it encodes the information bits.

2. Generator matrix for the Frozen vectors

This is represented as $G_N(A^c)$. This contains the row of $G_N$ which are not present in $G_N(A)$. This encodes the frozen vectors.

### Likelihood Ratio (LLR)

Likelihood ratio is the ratio of the probability of the bit being zero to the probability of the bit being one. The state of the current bit is estimated depending upon the value of the LLR. If LLR is ∞, then the bit is 0. If the LLR is 0, then the bit is 1. When LLR is 1, state is determined by fair coin flip.

Likelihood ratio for $i^{th}$ bit is given in equation (2.18)

$$L_N^i\left(y_1^N, \widehat{u_1^{i-1}}\right)= \frac{W_N^{(i)}\left(y_1^N, \widehat{u_1^{i-1}}|0\right)}{W_N^{(i)}\left(y_1^N, \widehat{u_1^{i-1}}|1\right)} \tag{2.18}$$

## a. Encoding operation

Encoding is done in the way shown below:

$$x_1^N = u_A G_N(A) + u_{A^c} G_N(A^c) \tag{2.19}$$

The equation (2.19) is equal to encoding the information bits by generator matrix consisting of good channels and encoding frozen bits with part of generator matrix consisting of bad channels. By performing this operation, we can justify that the information is transmitted from channels having capacity close to symmetric capacity of channel.

## b. Successive Cancellation decoding

Successive cancellation decoding or SC decoding in short relies on the calculation of LLRs for estimating the information bits. Assuming that calculation of i$^{th}$ LLR is already done, the decision of the bit is made as :

$$\hat{u}_i = \begin{cases} 0, & \text{if } L_N^i(y_i^N, \hat{u}_1^{i-1}) \geq 1 \\ 1 & otherwise \end{cases}$$

This decision is taken only for the information bits. Frozen bits are set to frozen value regardless of the value of the obtained LLRs.

### Calculation of LLR:

Equation (2.18) shows LLR expression. In this Section it is shown how the LLR value is calculated recursively as it was done originally in [1]. Recursion is in a sense that first the LLR value in level two is computed given LLR of level 1. LLR of level 2 is used to find the LLR value at level 3 and so on until $\log_2(N)$ level.

As in the case of channel splitting, there are two expressions of LLR. One is for even index and other is for the odd index. Here we show only the expression. For the proof, the reader is advised to read [1]. Equation (2.20) is for calculating LLR of odd index channel and equation (2.21) is for calculating LLR of even index channel.

$$L_N^{(2i-1)}(y_1^N, u_1^{\widehat{2i-2}}) = \frac{L_{N/2}^{i}(y_1^{N/2}, u_{1,o}^{\widehat{2i-2}} \oplus u_{1,e}^{\widehat{2i-2}}) L_{N/2}^{i}(y_{N/2+1}^{N}, u_{1,e}^{\widehat{2i-2}}) + 1}{L_{N/2}^{i}(y_1^{N/2}, u_{1,o}^{\widehat{2i-2}} \oplus u_{1,e}^{\widehat{2i-2}}) + L_{N/2}^{i}(y_{N/2+1}^{N}, u_{1,e}^{\widehat{2i-2}})} \tag{2.20}$$

$$L_N^{(2i)}(y_1^N, u_1^{\widehat{2i-1}}) = [L_{N/2}^{(i)}(y_1^{N/2}, u_{1,o}^{\widehat{2i-2}} \oplus u_{1,e}^{\widehat{2i-2}})]^{1-2\widehat{u_{2i-1}}} \bullet L_{N/2}^{(i)}(y_{N/2+1}^{N}, u_{1,e}^{\widehat{2i-2}}) \tag{2.21}$$

When the calculation reaches to block length 1 i.e at level 0, the LLR is calculated as:

$$L_1^{(1)}(y_i) = \frac{W(y_i|0)}{W(y_i|1)} \tag{2.22}$$

The above LLR calculation at the block 0 level is simply the ratio of conditional probability of $y_i$ when $u_i$ is 0 to the conditional probability of $y_i$ when $u_i$ is 1.

## 5. Generator Matrix Construction

In this section, description for constructing generator matrix is shown. First, we have to keep one thing in mind that this generator matrix is responsible for combining the channels with equal channel probabilities. Therefore, its construction is also coherent with this underlying principle. The generator matrix $G_N$ is represented as

$$G_N = B_N F^{\otimes N} \tag{2.23}$$

For combining 2 channels only, matrix used is

$$F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

This generator matrix combines the two channels as shown in the Fig. 2.4. When constructing Channel $W_2$, we use two independent unpolarized channels. However, after the construction of $W_2$ we have to be careful while combining the next level or $W_4$ because only similar channels can be combined. For this, there has to be criss-crossing as shown in Fig. 2.5.

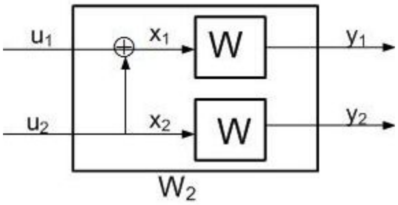


Fig. 2.4 Channel $W_2$

This criss-crossing is performed by the matrix $B_N$. Criss-crossing to add similar channels has to take place as we proceed to every next level of channel combination. Construction of $F^4$ is done using kronecker product. The kronecker product $F \otimes F$ is written as shown in equation 2.24

$$F \otimes F = \begin{bmatrix} 1 \times F & 0 \times F \\ 1 \times F & 1 \times F \end{bmatrix} \tag{2.24}$$

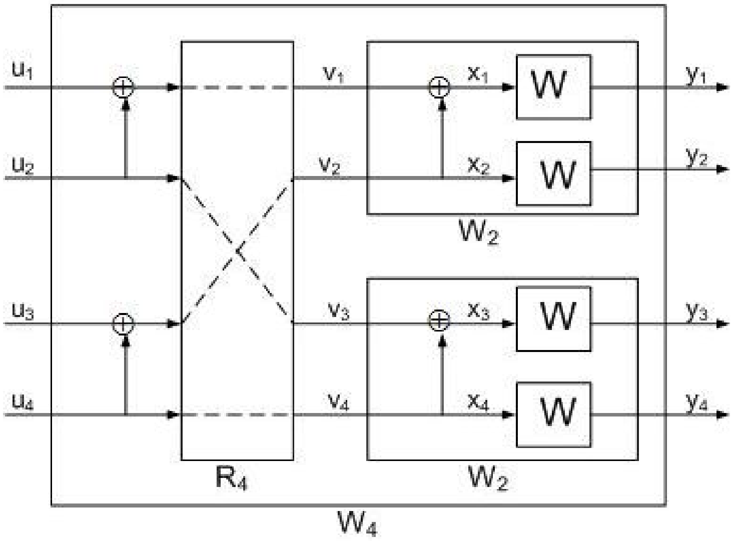


Fig. 2.5 Channel $W_4$ and Bit Index Reversal by $B_4$

The $n^{th}$ power of F is the kronecker product of F with itself taken $n$ times. So the $n^{th}$ kronecker power of F is mathematically written as $F \otimes F^{\otimes(n-1)}$. $B_N$ is called Bit Reversal Permutation matrix that rearranges the rows of $F^{\otimes N}$ matrix. This matrix can be constructed using equation (2.25).

$$B_N = R_N(I_2 \otimes B_{N/2}) \tag{2.25}$$

This iteration continues until $N=4$ because at this point $B_{N/2} = B_2 = I_2$. The matrix $R_N$ is the reverse shuffling matrix. This matrix is constructed by using the identity matrix of dimension $(N \times N)$. After that, the first $N/2$ rows are replaced by the rows with odd indices. And the last $N/2$ rows are filled by the even index rows.

If $V_1^N = \{v_1, v_2, \ldots.., v_{N-1}, v_N\}$ is vector consisting of $N$ elements, then

$V_1^N . R_N$ gives output as $V_1^N . R_N = (v_1, v_3, \cdots\cdot, v_{N-1}, v_2, v_4, \cdots, v_N)$ .

## 6. Inverse of matrix using Gaussian Elimination

Here, we will discuss how inverse of a square matrix is found out. The method we are using is called as 'Gaussian Elimination Method'. This is done in two iteration. They are forward and backward elimination. It is done in following way.

1. In the forward elimination, an identity matrix of size $(N \times N)$ is cascaded to the right size of the square matrix $A$ whose matrix inverse has to be found out. Then the new matrix $A'$ has the size $(N \times 2N)$ .

$$A' = [A_N \mid Id_N]$$

2. In the next step, matrix $A$ has to be diagonalized using elementary row operation. This means that using the elementary row operation on matrix $A'$, matrix $A$ must be converted into identity matrix. For this, in the first column of first row, we have to look if there is 1. If there is one, then second row can be proceeded. If there is no 1, nearest row that contains 1 has to be found and swapped with pivot row.

3. Then the first row is added to all the successive rows which contains 1 in the first column. This way all the row except first row has 0 in first column.

4. In the $i^{th}$ row, 1 is searched in the $i^{th}$ column. In case there is no 1, any row below $i$ that has 1 is searched and add the entire row to $i^{th}$ row. Then $i^{th}$ row is added to all the successive rows which contains 1 in $i^{th}$

column. This way all the rows below $i^{th}$ row has 0 in $i^{th}$ column. This step is done for all the $N$ rows of matrix $A'$.

5. If there is any row in the matrix consisting of only 0's then matrix $A$ does not have inverse. If this is not the case, step 6 is followed.

6. Then for the backward elimination, same step is done but this time at right bottom. Last row is taken and added to any row that has 1 in last column.

7. Above process is repeated from $N-1$ row to second row.

8. Then the matrix will transform as
$$A_T' = [Id_N \mid B_N].$$

9. If first $(N \times N)$ submatrix is an identity matrix, then matrix $B_N$ is inverse of $A$. If first $(N \times N)$ submatrix of $A_T'$ is not an identity matrix, then $A$ does not have an inverse.

# III. Proposed System

This section describes the proposed cryptosystem based on the polar code.

## A. Construction

The polar code is mathematically proven to achieve channel capacity in [1]. By repeatedly combining $N$ independent channels, next set of channels is obtained out of which some channels have capacity close to 1 with probability $I(W)$ and some channels have capacity close to 0 with probability $1-I(W)$. But, there are few channels which are not perfectly polarized and their capacity lie in between 0 and 1. If the length of polar code is very high, then these intermediate channels also vanish asymptotically. From [1], it is known that probability of block error when using successive cancellation decoding method has complexity of $O(N^{-1/4})$.

This gives idea that polar code with large block length can be used for cryptosystem. So, taking advantage of this feature parameter $N=2048$ is used. Case that is being considered is binary erasure channel (BEC). In [1], Arikan gave analysis of rate in terms of channel capacity and reliability in terms of Bhattacharya parameter for binary erasure channel. Analysis of the Bhattacharya parameter and channel capacity by recursive method is much easier for the BEC. Considering the Binary Symmetric Channel (BSC) same method cannot be applied because number of calculations required to calculate channel capacity for BSC grows by large extent as number of channels is increased. Capacity of BSC is a function of base 2 logarithm of error probability $e$. Due to these arguments, channel capacity cannot be calculated from equations (2.14) and (2.15) for BSC and equations (2.16) and (2.17) also cannot be used for calculating Z-parameter. However equation (2.16) gives

Z-parameter upper bound for BSC. In [18] and [25] a different approach is used for polarizing BSC.

Generator matrix $G_N(A)$ of size $(k \times N)$ is created by following the description of Section II-D and then selecting $k$ out of $N$ channels that have highest channel capacity. There can be many other constructions of generator matrix. Matrix $F$ which is described in [1] is used. In [15], Korada, and Sosaglu has given explanation on the class of matrix that can polarize channels. For polarization, according to [15] the generator matrix must be constructed from lower triangular matrix. The matrix $F$ from [1] satisfies this condition.

After generator matrix is created, $(k \times k)$ non-singular matrix and $(N \times N)$ permutation matrix must be constructed. Non-singular matrix and permutation matrix are generated randomly and multiplied to $G_N(A)$. Then $G_{pub} = SG_N(A)P$ can be constructed. When this randomized generator matrix encodes plaintext message, straightforward decoding becomes useless. Only the correct set of private keys, $S$ and $P$, can recover plaintext message from the ciphertext.

## B. Private key

Private keys are $(k \times k)$ matrix $S$ and $(N \times N)$ permutation matrix $P$. For the construction of invertible matrix $S$, random number of dimension $(k \times k)$ is generated in binary field $(F_2)$. In order to check its invertibility, Gaussian Elimination method is applied that has already been discussed in Section II-D.6. If this matrix can be reduced into identity matrix using elementary row operation, then its inverse exist else its inverse does not exist. In case inverse does not exist, whole process has to be repeated.

As for $(N \times N)$ permutation matrix it is created by random column shuffling of $(N \times N)$ identity matrix. When the receiver receives the encrypted message, first he has to inverse the permutation in the message so that they are correctly decodable. To obtain the inverse of permutation matrix, it is not necessary to perform Gaussian Elimination. By taking transpose of matrix $P$, its inverse can be obtained.

## C. Public key

Public key is the matrix $G_{pub} = SG_N(A)P$. Alice has knowledge of $G_{Pub}$ of Bob who is receiver, but does not have any knowledge about the factors $S$ and $P$ of Bob's public key $G_{Pub}$.

## D. Encryption

Message $m$ of $k$ bits is encoded by following process:

$$y = mSG_N(A)P + e \tag{3.1}$$

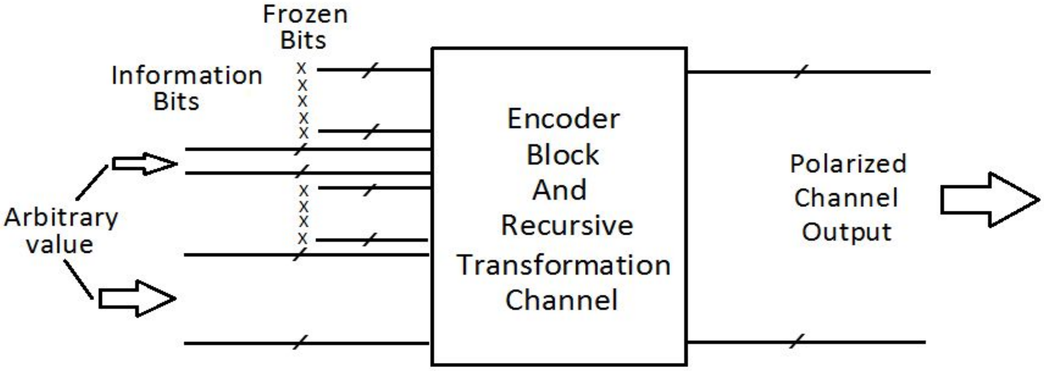


Fig. 3.1 Encoding in Polar code

Alphabet $e$ is error which occurs mostly in bad channels due to channel transformation. Fig. 3.1 shows encoding method. This figure shows how bits are encoded according to polar code. Fig. 3.1 does not represent encryption of proposed system.

## E. Decryption

In this section two decryption methods are described which is correct decryption and brute force decryption method.

### Correct Decryption

Here decryption by the true recipient (Bob) is discussed. In later section, description of decryption method that Eve is likely to follow is presented. In order to extract hidden information, decryption is done in the method described below.

First, private key $P$ is used to find inverse permutation $P^{-1}$. This matrix multiplies received ciphertext $Y$ and data which is shown in equation (3.2) is obtained.

$$Y' = YP^{-1} \tag{3.2}$$

Having removed permutation, bits of $Y$' can be decoded in an orderly manner using successive cancellation decoding. By calculating the values of likelihood ratios recursively using equations (2.20) and (2.21), likelihood ratio of any bit $i$ can be obtained if previous $i-1$ bits are known. Then by setting the value of bit to either 0 or 1 by using the decision rule of bits based on likelihood ratio, the estimate of $i^{th}$ bit is obtained. Then this bit along with all the other previously decoded bit is used to decode future bits until this process is completed for entire block length. During decoding process, decision has to be made only on the information set. In frozen set pre-fixed bits are inserted. In this case, pre-fixed bits are set as $0$ vector. After decoding, only those bits belonging to information set are chosen. The block diagram of successive cancellation decoding method is shown in Fig. 3.2. Bits represented by short arrows are discarded frozen bits and the bits represented by long arrows are estimation of information bits. All the calculations and decision making mechanisms takes place in the box labelled as "Successive Cancellation Decoder". Even after the bits from information set are obtained, correct message is not yet obtained yet. The obvious reason for this is that scrambles has to be removed from this information. The information that is obtained from successive cancellation decoder is message $m'$ shown in equation (3.3)

$$m' = mS \tag{3.3}$$

In order to obtain correct information $m$, another private key $S$ must be used. From $S$, its inverse $S^{-1}$ is calculated. Matrix $S^{-1}$ multiplies $m'$ and from this, finally we get our desired message $m$ which is shown in equation (3.4).

$$m = m'S^{-1} = (mS)S^{-1} \tag{3.4}$$

In Fig. 3.3 correct sequence of decoding to extract information from ciphertext $Y$ is shown. In this figure, it is shown in which order, the private key namely $S^{-1}$, $P^{-1}$ and successive cancellation decoding algorithm should be applied. In the encoding block, message of size $(1 \times k)$ is multiplied by public key $SG_N(A)P$. Multiplication is done in block which is labelled as "Encoder".

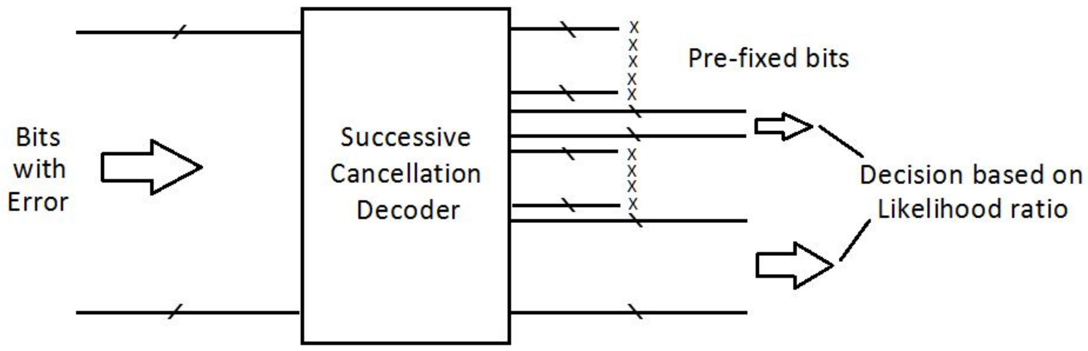

Bits with Error
Successive Cancellation Decoder
Pre-fixed bits
Decision based on Likelihood ratio

Fig. 3.2 Decoding Operation using pre-fixed bits

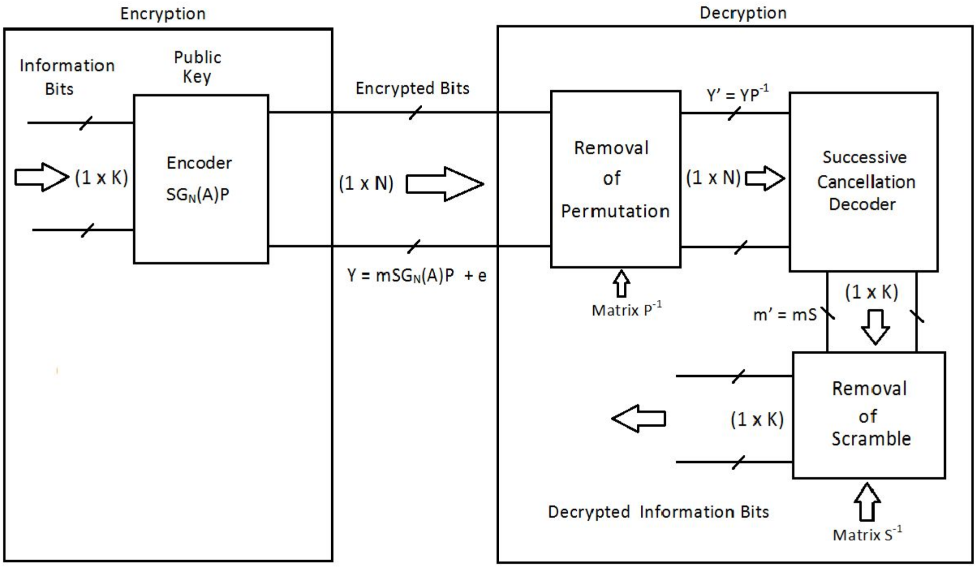


Fig. 3.3 Encoding and Decoding Operation in proposed method

In decryption block, first random permutation is removed by multiplying $P^{-1}$ with ciphertext. This operation is done in the block labelled “Removal of Permutation”. After removing permutation, size of data is still $(1 \times N)$. In the next step this data is decoded by successive cancellation method. Choosing the bits from information set, scrambled message of size $(1\times k)$ is obtained. To remove scrambles, it is multiplied with inverse of $S$. Scrambling matrix is multiplied with decoder output in block labelled as “Removal of Scrambles” in Fig. 3.3. In this figure, block diagram for determining the inverse matrices $S^{-1}$ and $P^{-1}$ is not shown because they are secondary. After this is complete, plaintext sent by Alice is obtained.

## Brute Force Decoding

This section describes another situation where Eve (intruder) tries to decode plaintext without having any knowledge of private keys. Fig. 3.4 shows the decoding process that Eve uses. In this scenario, Eve has knowledge of correct information set so she assumes that she can remove all the errors from ciphertext $Y$ by using successive cancellation decoding method and correctly decrypt the information bits without removing scramble $S$ and permutation $P$.

Due to the permutation applied in order to garble the encoded bits, positions of frozen bits and information bits are mixed. Hence, it is common for some frozen bits to lie in the position of information bits and some information bits to lie in place of frozen bits. Eve decodes message directly without applying inverse permutation to encoded message $Y$ into the successive cancellation decoder. Decoding them without pre-processing causes incorrect decoding. Error is propagated to other bits also so Eve cannot decode bits with knowledge of only correct information set of polar code.

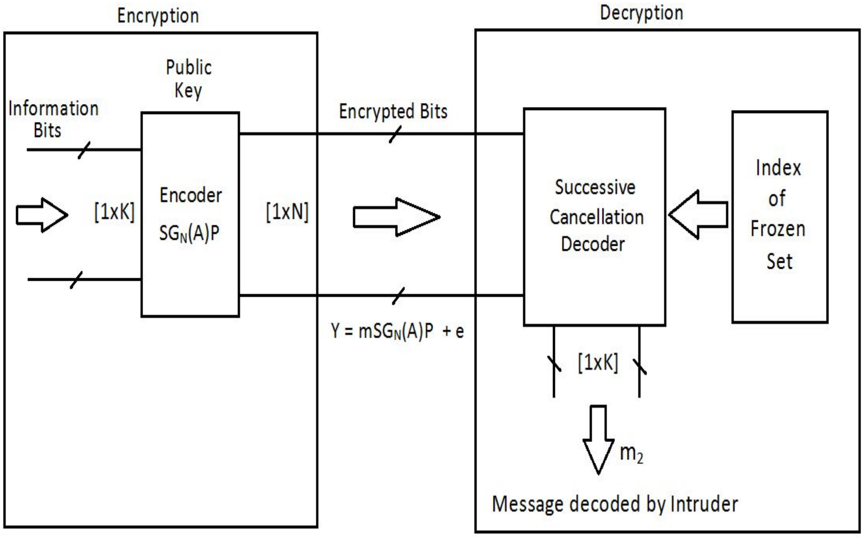


Fig. 3.4 Brute Force Decoding by Eve

# IV. Security Analysis

In this section, security analysis of the proposed model is presented.

## A. Brute Force Decoding Result

In this Section, brute force decryption method described in Section III-E is analysed. If message bits are decoded in exact manner as they were or bits are decoded with acceptably low error rate, then proposed encryption method cannot be used for concealing messages from unauthorized access. In this Section, three different types of scenarios are described when attacker tries to decode under Eve's assumption. They are - very low error rate which is near to $0\%$, error rate which is approximately $50\%$ and very high error rate $90\%-100\%$ .

Below, results of attacking method that attacker would perform in order to decode message bits is presented. First, data is encoded by using scrambled and permuted generator matrix or the public key $G_{pub}$. Then successive cancellation decoding method is used. Simulation was performed on block length $N$ = 1024, 2048, and 4096 where $N = 2^n$ and $n$ = 10, 11, and 12 with code rates starting from 0.3 to 0.8 and with erasure probabilities 0.2 to 0.7. Result of simulation is shown in Table. 3.1, 3.2, and 3.3.

Table. 3.1 Simulation result for decoding without private keys for length 1024

| rate | e=0.2 | e=0.3 | e=0.4 | e=0.5 | e=0.6 | e=0.7 |
|---|---|---|---|---|---|---|
| 0.3 | 154.3 | 153.5 | 153 | 154.1 | 152 | 152 |
| 0.5 | 257 | 256.4 | 256.5 | 244 | | |
| 0.6 | 308.3 | 308 | 304.8 | | | |
| 0.7 | 359 | 357.5 | | | | |
| 0.8 | 412.1 | | | | | |

Table. 3.2 Simulation result for decoding without private keys for length 2048

| rate | e=0.2 | e=0.3 | e=0.4 | e=0.5 | e=0.6 | e=0.7 |
|---|---|---|---|---|---|---|
| 0.3 | 303.8 | 311 | 306.8 | 305.4 | 306 | 303 |
| 0.5 | 508 | 511.7 | 507.9 | 508.3 | | |
| 0.6 | 613.1 | 617 | 614 | | | |
| 0.7 | 714 | 714.2 | | | | |
| 0.8 | 821 | | | | | |

Table. 3.3 Simulation result for decoding without private keys for length 4096

| rate | e=0.2 | e=0.3 | e=0.4 | e=0.5 | e=0.6 | e=0.7 |
|---|---|---|---|---|---|---|
| 0.3 | 612.3 | 614.9 | 613.4 | 610.9 | 614.1 | 619.5 |
| 0.5 | 1017 | 1014 | 1007 | 1035 | | |
| 0.6 | 1176 | 1219 | 1225 | | | |

The table was obtained by repeating the process many times and then averaging their values. In the simulations, the position of error was observed. When comparing the position of error during simulation, it was found that the position of error varied with each iterations.

## Analysis of obtained data

Here, results from the simulation is analyzed for each of the block lengths and data rate. Looking at number of error bits for each different block lengths in different data rates, error is approximately $50\%$.

When the error rate is $50\%$ or near to $50\%$, it implies that there is $50\%$ chance that the decoded bits are correct and $50\%$ chance that decoded bits are not correct. When we analyze this result, it is in favor of Alice because error

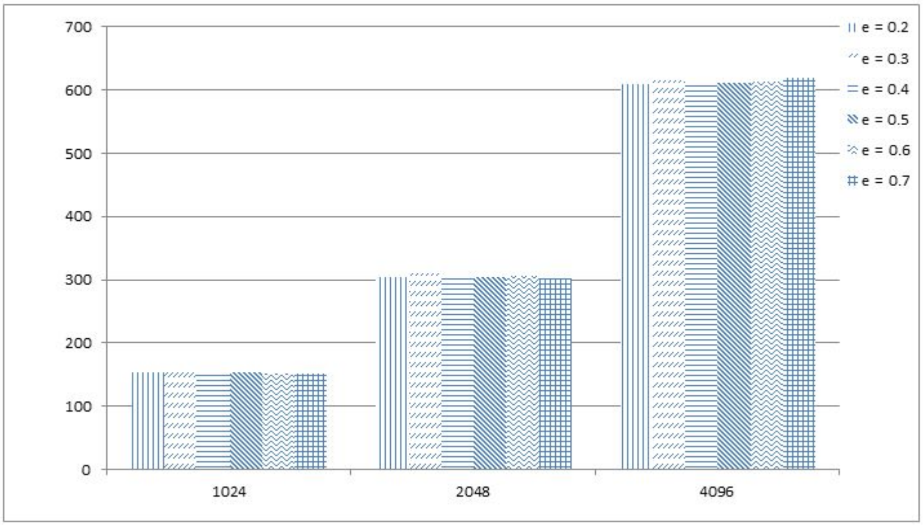


Fig. 4.1 Number of errors for code rate 0.3

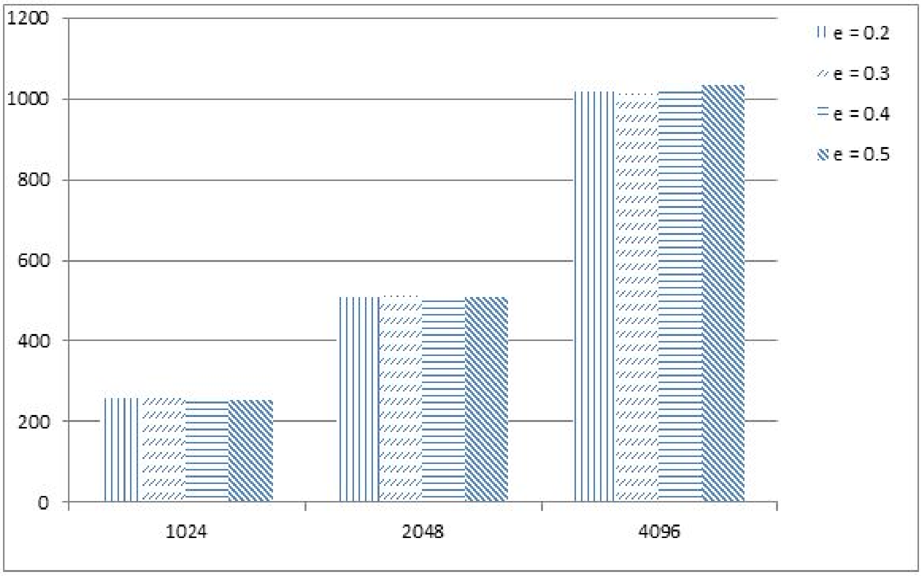


Fig. 4.2 Number of errors for code rate 0.5

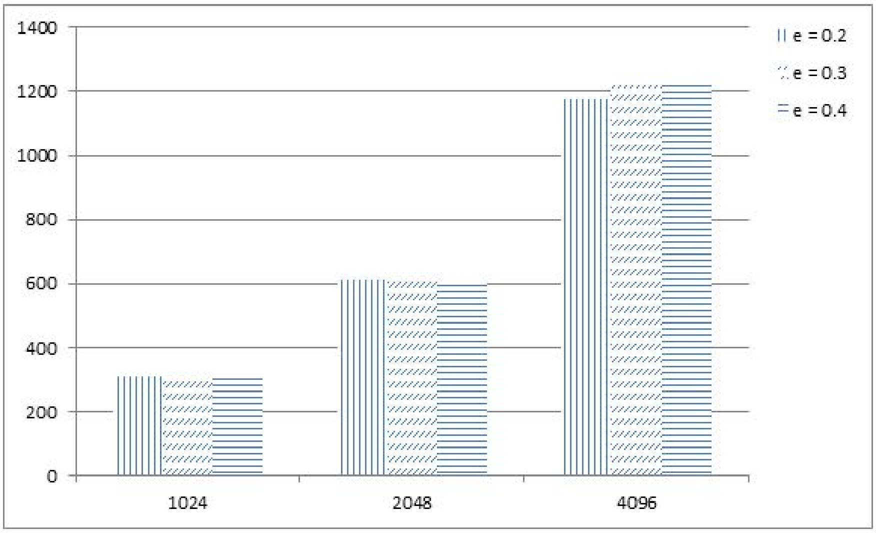


Fig. 4.3 Number of errors for code rate 0.6

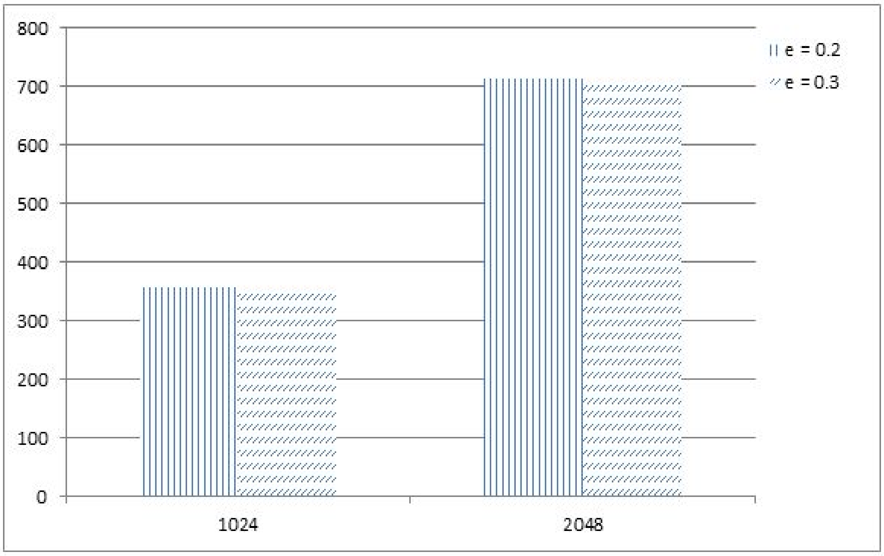


Fig. 4.4 Number of errors for code rate 0.7

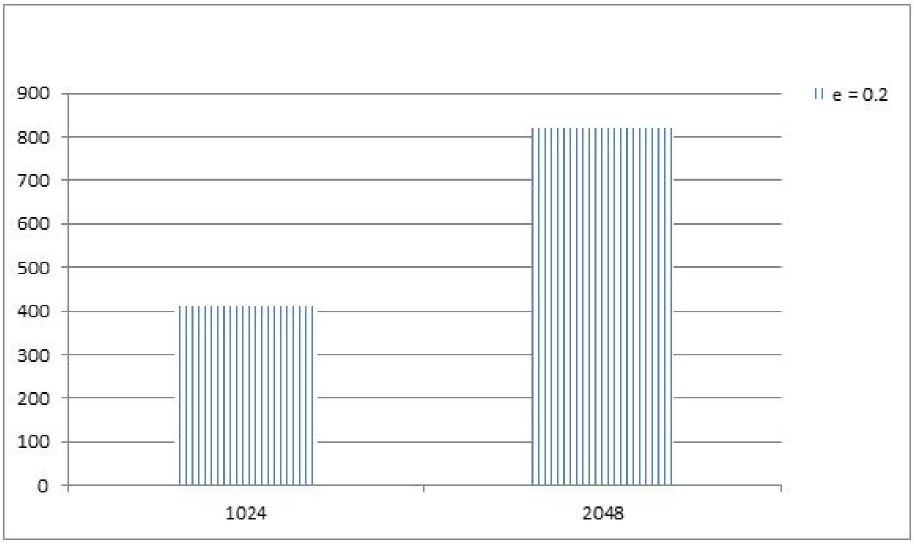


Fig. 4.5 Number of errors for code rate 0.8

is not biased towards high error rate or low error rate. Hence Eve cannot infer with full confidence whether any of the decoded bits are correct or not.

If the number of incorrect bits is much less than $50\%$ and almost near to $0\%$, then it means that even without knowing private keys, the decoder can obtain almost all of the information correctly. Due to the low error rate, attacker can be quite sure that the data she has decoded is almost correct. If error occurs on same positions, then it makes intruder easy to decode information because by repeatedly decoding, she can find out the position of error and so for later decoding procedures, she can guess the error location and obtain almost error-free plaintext.

If the number of error is much higher being at around $90\%$ or closer to $100\%$ then this condition also pose a serious disadvantage to our cryptosystem. If all of the decoded bits has error when compared to originally sent message then it implies that those bits error are just the opposite of message bits and therefore by flipping the bit states that is obtained from decoding, attacker can recover the true message bits with simple logic. In case all the decoded bits does not contain error but most of the decoded bits contains errors at around $90\%$, then by flipping the state of bits, the attacker can recover most if not all of the bits correctly and by observing the nature of information, attacker can guess the errors.

From Figs. 4.1, 4.2, 4.3, 4.4, and 4.5, it can be seen that error rate is approximately 50% hence it can be concluded that the proposed system is safe from the above discussed insecurities. The position of error also varied at each iterations.

In Fig. 4.1 to Fig. 4.5, decreasing bar indicates less erasure probability for increasing rates. In Fig. 4.1, simulation was carried out for code rate 0.3 and erasure probability of 0.2 to 0.7. In Fig. 4.5, simulation is carried out for only erasure probability 0.2 and code rate 0.1. This is because data rate must always be less than capacity of channel $I(W)$. In erasure channel, if erasure probability is denoted by alphabet $e$, then capacity of that channel is $1-e$. Hence for code rate 0.3, erasure probability is 0.7. So simulation is started by setting rate to 0.3. In Fig. 4.2, code rate is 0.5 and erasure probability varies from 0.2 to 0.5. This implies channel capacity varies from 0.8 to 0.5. In this case, if erasure probability was greater than 0.5, then channel capacity would have to be less than 0.5. And simulation for rate less than 0.5 has already been performed. In Fig. 4.3, code rate is 0.6. Hence, here erasure probability has to be from 0.2 and not greater than 0.4. Likewise in Fig. 4.4, simulations were performed for erasure rates 0.2 and 0.3. Under these erasure rates, minimum channel capacity is 0.7. And the code rate is 0.7. For code rate 0.8, simulation was carried out for erasure probabiity 0.2.

## B. Brute Force Attack

This section analyses security of public key $G_{Pub}$. For known $G$ and $SGP$, direct computation of matrix $S$ and $P$ is not feasible. For non-singular matrix $S$

of dimension $(k \times k)$, cardinality of $S$ is given as

$$|S| = (2^k - 1)(2^k - 2)(2^k - 2^2).....(2^k - 2^{k-1}) \qquad (4.1)$$

On simplifying equation (4.1) and substituting values of $k$, for block length $N = 1024$, and $k = 128$, number of possible invertible matrices are

$$|S| = 2^{127}.2^{126}.2^{125}....2.(127)\frac{3}{2}.2^{\binom{128}{2}} \quad .$$

For block length $N = 2048$, and $k = 256$, number of possible invertible matrices are

$$|S| = 2^{255}.2^{254}.2^{253}....2.(255)\frac{3}{2}.2^{\binom{256}{2}} \quad .$$

Cardinality of permutation matrix for block length 1024 is (1024!). Cardinality of permutation matrix for block length 2048 is $(2048!)$. With this many number of possible invertible matrix $S$ and total of $N!$ number of permutation matrix, it is impossible to find out correct $S$ and $P$ by hit and trial method as number of trials needed would be on average $(|S|)(N!)$ .

## C. Information Set Decoding Analysis

In this section, analysis of information set decoding is presented. Information set in this context is different from information set in the polar code. In this context, information set refers to the bits that are randomly selected under assumption that they are error free. Then from these bits, the plaintext is estimated by solving the public generator matrix $G_{Pub}$ consisting only of those columns which correspond with the information set.

In previous McElice-type public key cryptosystem number of errors is limited by the minimum distance of the code. But when polar code is used, number of error is not constrained by minimum distance but it depends on the capacity of the particular channel. Due to this reason, analysis by information set decoding may not be as efficient as in other McEliece type cryptosystems.

The analysis presented is the worst case for security with assumption that there is error only in worst channels (channels whose capacity is less than 0.1). In general there may be many more errors than that. For block length 2048 and erasure rate 0.5, number of such channels is 907. Therefore there are 907 error in unknown positions. For block length 1024 and erasure rate 0.5, number of such channels is 445. Error is denoted as $t$. Picking $k$ out of $N$ channels assuming that there is no error, probability of obtaining error free data is

$$\left(1-\frac{t}{N}\right)^{k}$$

Then from this set using the $k$ columns of public generator matrix, sent message has to be estimated. Equation (4.2) has to be solved

$$\begin{bmatrix} x_1\, x_2\, x_3\, x_4 \cdots x_2\, x_k \end{bmatrix} \begin{bmatrix} g_{11} & g_{12} & \cdot & \cdot & \cdot & g_{1k} \\ g_{21} & g_{22} & \cdot & \cdot & \cdot & g_{2k} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{k1} & g_{k2} & \cdot & \cdot & \cdot & g_{kk} \end{bmatrix} = \begin{bmatrix} y_1\, y_2\, y_3\, y_4 \cdots y_2\, y_k \end{bmatrix} \quad (4.2)$$

This gives $k$ equations which is then solved using Gaussian Elimination. Number of operations required for solving $k$ equations is $\left[\frac{2k^3}{3}+\frac{9k^2}{6}-\frac{13k}{6}\right]$.

For $N$ = 1024, and $k$ = 128, number of operation required is $2^{20.4399}$. For $N$ = 2048, and $k$ = 256, number of operation required is $2^{23.4276}$.

Therefore, using this information set, average operations required before finding appropriate data in worst case is

$$\left(\frac{2k^3}{3}+\frac{9k^2}{6}-\frac{13k}{6}\right)\left(1-\frac{t}{N}\right)^{-k}$$

For $N=1024$, and $k=128$, number of operation required is $2^{125.4399}$. For $N=2048$, and $k=256$, number of operation required is $2^{239.4703}$.

For McEliece cryptosystem with binary goppa code with $N=1024$. $k=524$ and $t=50$, number of operations is $2^{65}$.

For Sidelnikov cryptosystem with $N=1024$, $k=176$ and $t=200$, number of operations required is $2^{83.6}$.

## D. Sidelnikov Attack

Structure of Polar code and Reed-Muller code look alike because they both are constructed by choosing words from $F^{\otimes m}$ which is shown in Fig. 2.1 for $(m=3)$. But there is significant difference between them. In this Section, difference between Reed-Muller code and Polar code is discussed and subsequently shown that cryptographic attack against Reed-Muller codes cryptosystem does not apply to the case of Polar code cryptosystem.

It is known that in Reed-Muller code $R(r,m)$, only those basis having maximum weight are selected from $F^{\otimes m}$ for generator matrix. The number of words with equal weight is

$$c_i = \binom{m}{i}, \text{ i = 0 to m.}$$

Adding all $C_i$ is equal to number of rows in $R(r,m)$. Hamming weight of codes is equal to $2^{m-i}$.

Degree 2 codewords of $R(r,m)$ is equal to product of 2 codewords of $R(1,m)$. This is true for all $\binom{m}{2}$ combinations. And this also applies to any degree $r$. So all functions generating words in $R(r-1,m)$ is also present in $R(r,m)$. Equation (4.3) holds true for RM code.

$$R(0,m) \subset R(1,m) \subset \text{........} \subset R(m-1,m) \subset R(m,m) \tag{4.3}$$

Attack against Sidelnikov cryptosystem use equations (4.4) and (2.5) as basis to find permutation $P$ of the public generator matrix $G_{pub}$. Equation (2.5) makes factoring possible. Factoring minimum weight words of $R(r,m)$ gives $R(r-1,m)$. This process continues with decreasing $r$ in every iteration until enough factors are found to finally construct basis of permuted $R(1,m)$.

For given $r$ and $m$, generator matrix of Reed-Muller code consists of only those rows having high hamming weight. This is not the case in polar code. Rows are selected according polar code rule set by Arikan in [1]. Due to this all the functions generating high weight word are not present. This is one of the difference in comparison to RM code. For smaller code size, the generator matrix for polar and Reed-Muller code may be same but for larger dimension, the generator matrix varies. Generator matrix constructed in this way are in some sense not systematic and do not follow equation (3.2) also.

In $R(3,11)$, minimum weight of codeword is 256 but for (2048, 256) polar code, minimum weight is 64. Therefore, for polar code equation (4.4) holds true.

$$f \neq f_1 \cdot f_2 \cdot \cdots \cdot f_{r-1} \cdot f_r \tag{4.4}$$

In (2048, 256) polar code, minimum weight is 64. Number of such words is 14. From these words, only 35 words with weight 128 were found out. Remaining words of total 113 words could not be found.

All the words of weight 128 were not found from the low weight words of $(2048, 256)$. This shows that Sidelnikov attack cannot be used against the proposed method.

## E. Key-size and Rate Analysis

In this table, number of operation required for finding the error word and the key-size of different polar code rate is shown for block length 1024 and 2048 for erasure rate 0.5..

Table. 3.4 Number of operation and key-size for length 1024

| k | No.of operation | Key-size (Bytes) |
|---|---|---|
| 128 | $2^{125.730}$ | 16,384 |
| 192 | $2^{180.1221}$ | 24,576 |
| 256 | $2^{234.008}$ | 32,768 |
| 384 | $2^{341.049}$ | 49,152 |
| 512 | $2^{447.582}$ | 65,536 |

Table. 3.5 Number of operation and key-size for length 2048

| k | No. of operation | Key-size (Bytes) |
|---|---|---|
| 128 | $2^{128.461}$ | 32,768 |
| 192 | $2^{184.218}$ | 49,152 |
| 256 | $2^{239.470}$ | 65,536 |
| 384 | $2^{349.242}$ | 98,304 |
| 512 | $2^{458.506}$ | 131,072 |
| 768 | $2^{676.302}$ | 196,608 |
| 1024 | $2^{893.589}$ | 262,144 |

The figures in number of operation is accurate upto 3 decimal digits.

## F. Comparison Table

Here security and key-size of polar code cryptosystem is compared with McEliece and Sidelnikov cryptosystem for block length 1024 and 2048. Based on the comparison, proposed parameter for polar code cryptosystem is $N=2048$ and $k=256$ .

Table. 3.6 Comparison table for length 1024

| Cryptosystem | Security | Key-size (Bytes) |
|---|---|---|
| (1024, 524, 50) McEliece | $2^{65}$ | 67,072 |
| (1024, 176, 200) Sidelnikov | $2^{83.6}$ | 22,528 |
| (1024, 128) Polar | $2^{125.730}$ | 16,384 |

Table. 3.7 Comparison table for length 2048

| Cryptosystem | Security | Key-size (Bytes) |
|---|---|---|
| (1024, 1751, 34) McEliece | $2^{80}$ | 65,006 (CCA2-variant) |
| (2048, 232, 200) Sidelnikov | $2^{105.7750}$ | 59,392 |
| (2048, 256) Polar | $2^{239.4703}$ | 65,536 |

# V. Conclusion

In this thesis it is shown how polar code can be used for public-key encryption. By using properties of polar code for encoding and decoding, it is show that using polar codes can be one of the alternative for McEliece-type cryptosystem. Method for constructing public generator matrix from polar code is described. Then method which is used to encrypt plaintext by making it random using non-singular matrix and permutation matrix is described. And method of decrypting plaintext from them correctly is also described.

Apart from this, analysis of security by considering decoding without using private keys for different data rates is presented. And brute force method to break it is also analyzed. It is shown that polar code is different from Reed-Muller code. And it is this very dissimilarity from where advantages lies. From this, it is shown that Sidelnikov cryptanalysis cannot be used against the polar code.

In this thesis, polar code for BEC is implemented while analysis for BSC still remains an area of further research and it is a limitation of this thesis. Nevertheless, we can predict that implementation over BSC also shows similar behaviour as polarization is a general phenomenon and is not restricted to any specific channel. In polar code, the number of errors that can be added is much higher than for goppa code or RM code. And by mathematical analysis it is shown that McEliece cryptosystem information set decoding is also not feasible for polar code based PKC. Number of errors is always greater than lowest weight word of polar code and this feature makes low weight word finding attacks useless against polar code based PKC.

Hence in this thesis a new method for encryption and decryption of message using polar code is presented.

# References

[1] E. Arikan, “Channel Polarization: A Method For Constructing Capacity - Achieving Codes for Symmetric Binary-Input MemoryLess Channels,” *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051-3073, July 2009.

[2] R.J. McEliece, “Public Key Cryptosystem Based On Algebraic Coding Theory,” DSN Progress Report, pp. 114-116, 1978.

[3] L. Minder, “CryptoGraphy based On Error Correcting Codes,” Ph.D. Dissertation, 2007

[4] L. Minder, Amin Skrollahi, “Cryptanalysis of Sidelnikov Cryptosystem,” in *Proc. EUROCRYPT 2007, LNCS*, vol. 4515, pp. 347-360, 2007.

[5] S.B. Korada, “Polar Codes For Channel And Source Coding,” Ph.D. Dissertation, EPFL, 2009.

[6] D.J. Bernstein, “Attacking and defending McEliece Cryptosystem,” *Proc. 2nd International Workshop on Post-Quantum Cryptography*, pp. 31-46, 2008

[7] V.M. SidelNikov, “A Public Key Cryptosystem Based On Binary Reed-Muller codes,” *Discrete Math. Appl.*, vol 4, no. 3, pp. 191-207, 1994.

[8] A. Canteaut, F. Chabaud, “A New Algorithm For Finding Minimum-Weight Words In a Linear Code: Application To primitive Narrow-Sense BCH Code of Length 511,” *IEEE ITW*, vol. 44, pp. 367-378, October 1995.

[9] A. Canteaut, F. Chabaud, “Improvements Of The Attacks On Cryptosystems Based On Error-Correcting Codes,” *Rapport interne du Department mathematiques et Informatique, LIENS-95-21*, July 1995.

[10] V.M. Sidelnikov, A.V Pershakov, “Decoding of Reed Muller codes in case of large amount of errors.” *Probl. Inform. Transmission*, no.3, pp. 80-94, 1992.

[11] H. Niederreiter, “Knapsack-type Cryptosystem and Algebraic Coding

Theory," *Probl. Control. Inform. Theory.*, pp. 19-34, 1986.

[12] S. Zhao, P. Shi, B. Wang,"Design of Bhattacharya Parameter in Construction Of Polar Code," in *Proc. IEEE ICWC, Networking And Mobile Computing*, September 2011.

[13] A. Eslami, H.P. Nik, "A Practical Approach To Polar Codes," in *Proc. IEEE ISIT*, 2011.

[14] E. Abbe, "Polarization And Randomness Extraction," *IEEE ISIT 2011*, pp. 184-188, July 2011.

[15] S.B. Korada, E. Sasoglu, "A Class of Transformations that Polarize Symmetric Binary-Input Memoryless Channels,", in *Proc. IEEE ISIT 2009*, pp. 1478-1482, July 2009.

[16] E. Arikan, "Channel Combining and Splitting for Cutoff Rate Improvement," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, February 2006.

[17] N. Sendrier, "On The Structure Of A Randomly Permuted Concatenated Code," EUROCODE 94, October 1994.

[18] Ryuhei Mori, "Properties and Construction of Polar Codes," Dissertation, Kyoto University, February 1, 2010.

[19] V.M. Sidelnikov, S.O. Shestakov, "On the Insecurity of Cryptosystems based on Generalized Reed-Solomon codes," *Discrete Math. Appl.*, vol. 2, no.4, pp. 439-444, 1992.

[20] P.J. Lee, E.F. Brickell, "An Observation on the Security of McEliece's Public-Key Cryptosystem," *Advances in Cryptology, Eurocrypt'88, LNCS*, vol. 330 , pp. 275-280, 1988.

[21] C. Wieschebrink, "Cryptanalysis of the Niederreiter Public Key Scheme based on GRS Subcodes," *Post Quantum Crypto, LNCS*, vol. 6061, pp. 61-72, 2010.

[22] E. Arikan, "On the Rate of Channel Polarization," *Proc. IEEE ISIT*, vol.3, pp. 1493-1495, Korea, 2009.

[23] E. Arikan, "A Performance Comparison of Polar Codes and Reed-Muller Codes," *IEEE. Commun. Lett.*, vol. 12, no. 6, June 2008.

[24] E. Arikan, “A Survey of Reed-Muller Cod es from Polar Coding Perspective,” *IEEE ITW 2010*, pp. 1-5, January 2010

[25] R. Mori, “Performance of Polar code with construction using Density Evolution,” *IEEE Commun. Lett.*, vol. 13, no. 7, pp. 519-521, July 2009.

[26] R.L. Rivest, A. Shamir, L.M. Adleman, “A Method for Obtaining Digital Signature and Public Key Cryptosystems,” *Commun. ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[27] P.W. Shor, “Algorithms for Quantum Computation: Discrete Logarithms and Factoring,” *Proc. IEEE ASFC, Computer Society Press*, pp. 124-134, 1994.

[28] Y.X. Li, R.H. Deng, X,M, Wang, “On the Equivalence of McEliece’s and Niederreiter’s Public-Key Cryptosystems,” *IEEE ITW*, vol. 40, January, 1994.

[29] T.P. Berger, P. Loindreau, “How to mask the structure of codes for a cryptographic use,” *Designs, Codes and Cryptography*, vol. 35, pp. 63-70, 2005.

[30] T.P. Berger, P.L. Cayrel, P. Gaborit, A. Otmani, “Reducing key-length of McEliece Cryptosystem,” *AfricaCrypt 2009, 2nd International Conference on Cryptology*, *LNCS 5580*, pp. 77-97, June 2009.

[31] J.S. Leon, “A probabilistic algorithm for computing minimum weights of large error-correcting codes,” *IEEE Trans. Inf. Theory*, vol. 34, no. 5, pp. 1354-1359, 1988.

[32] J. Stern, “A method for finding codewords of small weight,” *Coding Theory. Applic, LNCS*, pp.106-113, 1989.

[33] N. Koblitz, “Elliptic curve cryptosystems,” Mathematics of Computations, Vol.28, pp. 203-209, 1987.

[34] V.S. Miller, “Use of Elliptic Curves in Cryptography,” *Proc. Advances in Cryptology,* Vol 85, pp. 417-426, 1985.

# List of Publications

**S.R Shrestha**, Y-S. Kim, “New Design of McEliece Type Public Key Crypto-System,” in *Proc JCCI 2013*, May 2013 (Kyoungju, Korea).

Y-S. Kim, **S.R Shrestha**, J-W. Jang, “Refined Algorithm for Prime Number Generation in Embedded Security System,” in *Proc IEEE, Asia-Pacific Services Computing Conference*, December 2011, (Jeju, Korea).

Y-S. Kim, S-J. Han, **S.R. Shrestha**, “Hadamard post Processing for True Random Number Generator,” in *Proc CISC S12*, June 2012 (Cheonan, Korea).

Y-S. Kim, **S.R. Shrestha**, S-J. Han, “Generalized Secure Network Coding Scheme based on All Or Nothing Transform,” in *Proc CISC W12* Dec 2012 (Seoul, Korea).

# Acknowledgement

This thesis is culmination of invaluable instruction and advice from many grateful individuals who has guided me and helped me achieve this goal. I am therefore indebted to all people whose constant guidance and encouragement.

Firstly, I would like to express my most sincere gratitude to my advisor, Prof. Young-Sik Kim for his constant supervision and valuable guidance throughout my 2 years graduate program. Through his relentless support and valuable advice, I learnt many aspects of research, developed knowledge about problem identification and ability to present solutions in comprehensive manner. Under his supervision, I also developed deep interest in Information Theory and Security and many other topics relating to Information and Communication that I was unaware of.

I would also like to thank committee chairperson, Prof. Seung-Jo Han and committee member Prof. Goo-Rak Kwon for their detail review, constructive criticism and insightful feedback during the preparation of this thesis.

During these years, I have collaborated with many people. Therefore I am most sincerely thankful to all the lab-mates of Information Theory and Security Lab and all other friends for their kind co-operation and support during my MS course.

I am as ever, indebted to my family for their love, support and encouragement in every step of my life. They have always been source of my inspiration.

At last but not the least, I greatly acknowledge Global IT program and Chosun University for financial assistance and academic scholarship during my 2 years program without which I would not be able to realize this endeavor.