

Asymmetric Encryption for Wiretap Channels

Salah Yousif Radhi Al-Hassan

Ph.D

October 20, 2015

Copyright ©2015 Salah Al-Hassan

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the author's prior consent.

This thesis is dedicated to my Father who
gave me the strength and unlimited support

**RESEARCH
DEGREES
WITH
PLYMOUTH
UNIVERSITY**

**Asymmetric Encryption for Wiretap
Channels**

by

SALAH YOUSIF Al-HASSAN

**A thesis submitted to Plymouth University
in partial fulfilment for the degree of**

DOCTOR OF PHILOSOPHY

**School of Computing and Mathematics
Faculty of Science and Technology
Plymouth University, UK**

October 20, 2015

Acknowledgements

First and foremost, I am grateful to Allah for His blessings and the good health that were necessary to accomplish my research.

I offer my sincerest gratitude to my first supervisor, Associate Professor Mohammed Zaki Ahmed, for his immense information in my topic, excellent guidance, patience, and providing me an excellent atmosphere for doing research. This dissertation would not be possible without his guidance and persistent help.

I would like to express the deepest appreciation to my second supervisor, Professor Martin Tomlinson, who gave me his care, support, advises, valuable comments, suggestions, and provisions that helped me in the completion and success of this research. I am so proud to accomplish this research under your supervision.

I am also grateful to the Ministry of Higher Education and Scientific Research in Kurdistan Regional Government for the financial support.

I take this opportunity to express gratitude to all faculty members of the Electronic Department in the Sulaymaniyah Technical Institute for their help and support, especially to my dear friends, Shwan Mohammed and Polla Ahmed, for their unlimited support throughout my research period.

I would like to thank all Iraqi friends and colleagues at Plymouth University especially Saif, Nadia, Abass, Ayad, Bashar, Saffa, Ali Hadi, Ali Jabbar and Sahib for their support which made me I do not feel loneliness and alienation.

I would also like to thank all my colleagues in Smeaton 208 for their help and support throughout my research work especially Ali Alfayly and Is-Haka Mkwawa. Thanks a lot for my colleague Mark Schofield for his time and effort in proof-reading my thesis.

I would like to thank my parents, sisters. They were always supporting me and encouraging me with their best wishes.

Finally, I would like to express my deepest love and appreciation to my wife, Zainab Al Shakarchi, for her moral support. She was always give me strength and patience during the good times and bad.

Author's Declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other university award without prior agreement of the Graduate Sub-Committee.

Work submitted for this research degree at the Plymouth University has not formed part of any other degree either at Plymouth University or at another establishment.

This study was financed by Ministry of Higher Education and Scientific Research in Kurdistan Regional Government(KRG)-Iraq under the Human Capacity Development Program (HCDP) for "Communication Engineering".

A programme of advanced study was undertaken, which included the extensive reading of literature relevant to the research project and attendance of international conferences on coding and communications.

The author has presented papers in the following international conferences:

1. IEEE GIIS 2013, The 5th Global Information Infrastructure and Networking Symposium, Trento, Italy, October 2013.
2. IEEE GIIS 2014, Global Information Infrastructure and Networking Symposium, Montreal, Canada, September 2014.
3. IEEE ICTC 2014, International Conference on Information and Communication Technology Convergence, Busan, South Korea, October 2014.
4. IEEE International Conference on Communication, IEEE ICC 2015 - Workshop on Wireless Physical Layer Security, London, UK, June 2015.

Word count of main body of thesis: 59536

Signed

Date

Asymmetric Encryption for Wiretap Channels

Salah Yousif Radhi Al-Hassan

Abstract

Since the definition of the wiretap channel by Wyner in 1975, there has been much research to investigate the communication security of this channel. This thesis presents some further investigations into the wiretap channel which improve the reliability of the communication security. The main results include the construction of best known equivocation codes which leads to an increase in the ambiguity of the wiretap channel by using different techniques based on syndrome coding.

Best known codes (*BKC*) have been investigated, and two new design models which includes an inner code and outer code have been implemented. It is shown that best results are obtained when the outer code employs a syndrome coding scheme based on the (23, 12, 7) binary Golay code and the inner code employs the McEliece cryptosystem technique based on *BKC's*.

Three techniques of construction of best known equivocation codes (BE_qC) for syndrome coding scheme are presented. Firstly, a code design technique to produce new (BE_qC) codes which have better secrecy than the best error correcting codes is presented. Code examples (some 50 codes) are given for the case where the number of parity bits of the code is equal to 15. Secondly, a new code design technique is presented, which is based on the production of a new (BE_qC) by adding two best columns to the parity check matrix(H) of a good (BE_qC), $[n, k]$ code.

The highest minimum Hamming distance of a linear code is an important parameter which indicates the capability of detecting and correcting errors by the code. In general, (BE_qC) have a respectable minimum Hamming distance, but are sometimes not as good as the best known codes with the same code parameters. This interesting point led to the production of a new code design technique which produces a (BE_qC) code with the highest minimum Hamming distance for syndrome coding which has better secrecy than the corresponding (*BKC*). As many as 207 new best known equivocation codes which have the highest minimum distance have been found so far using this design technique.

Contents

Acknowledgements	i
Author's Declaration	ii
Abstract	iii
List of Figures	viii
List of Tables	x
Abbreviations	xiii
1 Introduction	1
1.1 Thesis Aims and Organisation	3
1.2 Contribution to Knowledge	5
2 Syndrome Coding Scheme and Best Known Linear Codes	9
2.1 Software Platform and Tools Used	9
2.1.1 NTL Library	9
2.1.2 Magma Software Suite	10
2.2 Definition	11
2.2.1 Binary symmetric channel(<i>BSC</i>)	11
2.2.2 Definition of Information Theory	11
2.2.3 Definition of Coding System	13
2.3 The Binary Symmetric Channel(<i>BSC</i>)	20

2.3.1	Worked example of 2-bit <i>BSC</i>	24
2.4	Wiretap Channel	29
2.5	Syndrome Decoding	31
2.6	Syndrome Coding Scheme for the Wiretap Channel	33
2.6.1	Encoder	33
2.6.2	Legitimate Receiver's Decoder	34
2.6.3	Eavesdropper's Decoder	34
2.7	The (23, 12, 7) Binary Golay Code	35
2.7.1	Perfect Binary Golay Code	35
2.7.2	Generator matrix and Parity check matrix of Golay Code	36
2.7.3	Weight distribution of Golay Code	39
2.7.4	The Codewords and Algebraic Decoder of Golay Code	41
2.7.5	The Encoding and Decoding Algorithms of Golay Code	42
2.8	Best Known Linear Code (33, 23, 5)	47
2.8.1	Construction of Binary Linear Codes of minimum distance five	47
2.8.2	Construction of Parity Check Matrix $H[10, 33]$ of (33, 23, 5) code	48
2.8.3	Generation of Parity Check Matrix using Magma Software	52
2.9	Literature Review	56
3	Implementation of Secrecy Coding for the Wiretap Channel using BKLC	62
3.1	Introduction	62
3.2	McEliece Public Key Cryptosystem	64
3.2.1	Definitions	64
3.2.2	Key Generation	65
3.2.3	Encryption Algorithm	66
3.2.4	Decryption Algorithm	66
3.3	Proposed Coding scheme for the Wiretap Channel [Model-1]	67
3.3.1	Encoding Algorithm	68
3.3.2	Legitimate Receiver's Decoder	70
3.3.3	Eavesdropper's Decoder	71

3.4	Proposed Coding scheme for the Wiretap Channel [Model-2]	72
3.4.1	Encoding Algorithm	73
3.4.2	Legitimate Receiver's Decoder	75
3.4.3	Eavesdropper's Decoder	76
3.5	Computation Results of the Two Models	77
3.5.1	Joint Entropy	78
3.5.2	Equivocation	79
3.5.3	Channel Capacity	80
3.5.4	Normalised Equivocation Difference	81
3.5.5	Equivocation Gain	82
3.6	Summary	87
4	Implementation and Construction of Best Known Equivocation Codes for Syndrome Coding	88
4.1	Introduction	88
4.2	The systematic packed integer form of parity check matrix	90
4.3	Syndrome Coding Scheme	91
4.4	Evaluation of the equivocation rate achieved by syndrome coding	95
4.5	Exhaustive evaluation of new best known equivocation codes for syndrome coding with the example case of 15 parity bits	96
4.5.1	Recursive Evaluation of the syndrome probability distribution	96
4.5.2	Code design technique	107
4.5.3	Results	108
4.6	Implementation and Construction of best known equivocation codes with highest minimum Hamming distance for syndrome coding	110
4.6.1	Calculation of the minimum distance of a linear code	111
4.6.2	Code design technique	113
4.6.3	Results	114
4.7	Construction of best known equivocation codes from shorter best equivocation codes by adding two columns to the parity check matrix	118

4.7.1	Evaluation the probability mass function of the extended $[n+2, k+2]$ code	118
4.7.2	Code design technique	119
4.7.3	Results	121
4.8	Summary	122
5	Conclusions and Future Work	123
5.1	Conclusions	123
5.2	Future Work	128
Papers Published		130
Appendix A	Best Known Equivocation Codes with 15 parity bits	156
Appendix B	Best Known Equivocation Codes of the Highest Minimum Distance	164
Bibliography		183

List of Figures

1.1	Traditional digital communication system	1
1.2	General wiretap channel	2
2.1	Discrete Memoryless Channel	14
2.2	Binary Symmetric Channel (BSC)	20
2.3	Special case of the eavesdropper channel	22
2.4	Channel Capacity of 1-bit <i>BSC</i>	23
2.5	Secrecy Capacity (Equivocation) of 1-bit <i>BSC</i>	23
2.6	2-bit <i>BSC</i> Example	24
2.7	Channel Capacity for 2-bit <i>BSC</i>	27
2.8	Secrecy Capacity (Equivocation) of 2-bit <i>BSC</i>	28
2.9	Normalised Channel Capacity for 1-bit and 2-bit <i>BSC</i>	28
2.10	Normalised Equivocation for 1-bit and 2-bit <i>BSC</i>	29
2.11	wiretap channel Model (special case)	29
2.12	The syndrome coding scheme for the wiretap channel	33
2.13	Block Diagram of the Encoder	34
2.14	Block Diagram of the BSC channel and Decoder of Eavesdropper	35
3.1	Block Diagram of Proposed Coding scheme for the Wiretap Channel	63
3.2	Block Diagram of McEliece Cryptosystem	65
3.3	Block Diagram of Proposed Coding scheme for the Wiretap Channel[Model-1]	67
3.4	Block diagram of syndrome coding scheme based on (23, 12, 7) Golay code	68

3.5	Block Diagram of McEliece cryptosystem technique based on $BKLC(33, 23, 5)$	69
3.6	Block Diagram of the BSC channel and Decoder of Eavesdropper	71
3.7	Block Diagram of Proposed Coding scheme for the Wiretap Channel[Model- 2]	73
3.8	Block Diagram of McEliece cryptosystem technique based on $BKLC(58, 46, 5)$	74
3.9	Block Diagram of the BSC channel and Decoder of Eavesdropper	76
3.10	The Joint Entropy $H(M, \hat{M})$ vs. p_e	78
3.11	The Normalised Equivocation $H(M \hat{M})$ vs. p_e	79
3.12	The Channel Capacity $I(M; \hat{M})$ vs. p_e	80
3.13	Normalised Equivocation for an insecure system on BSC and secure sys- tem on BSC vs. p_e	81
3.14	Normalised Equivocation Difference between the secure system on BSC and insecure system on BSC vs. p_e	82
3.15	Equivocation gain of the secure system on BSC and insecure system on BSC vs. p_e	83
3.16	Normalised Equivocation vs. p_e	84
3.17	Channel Capacity vs. p_e	84
3.18	Normalised Equivocation Difference vs. p_e	85
3.19	Equivocation Gain vs. p_e	85
4.1	Wiretap channel Model	89
4.2	Block diagram of syndrome coding scheme for $[n, k, d]$ linear block code .	92
4.3	Block Diagram of the BSC channel and Eavesdropper's Decoder	94
4.4	Equivocation rate $Eqv.rate$ vs. p_e of best known equivocation (BE_qC) and best known (BKC) codes for $n = 82$	109

List of Tables

2.1	Transmitter Entropy $H(x)$ for $p_e = 0.01$	24
2.2	Receiver Entropy $H(y)$ for $p_e = 0.01$	24
2.3	Joint Entropy $H(x,y)$ for $p_e = 0.01$	25
2.4	Transmitter Entropy $H(x)$ for $p_e = 0.5$	25
2.5	Receiver Entropy $H(y)$ for $p_e = 0.5$	26
2.6	Joint Entropy $H(x,y)$ for $p_e = 0.5$	26
2.7	Channel Capacity and Equivocation values	27
2.8	The standard array of example	32
2.9	The Syndrome look up table of example	32
2.10	The number of error and syndrome patterns of $(23, 12, 7)$ Golay code . .	36
2.11	The Weight distribution of $(23, 12, 7)$ Golay Code	40
2.12	Galois Field of $GF(2^{10})$, primitive polynomial = $x^{10}+x^3+1$	49
2.13	The number of codewords of weight $(n-w)$ is equal to number of codeword of weight w	55
3.1	Normalised equivocation and Information Leakage for all schemes	86
4.1	Equivocation rate of the $[7, 4, 3]$ Hamming code	106
4.2	<i>Equivocation rate and minimum distance in syndrome coding for $p_e = 0.05$, $m = 7$</i>	115
4.3	<i>Equivocation rate and minimum distance in syndrome coding for $p_e = 0.05$, $m = 11$</i>	116

4.4 <i>Equivocation rate and minimum distance in syndrome coding for $p_e = 0.05$, $m = 12$</i>	117
A.1 <i>Best Known Equivocation Codes that achieve at least 70% secrecy in syndrome coding for $p_e = 0.05$</i>	157
A.1 <i>Continued from previous page</i>	158
A.1 <i>Continued from previous page</i>	159
A.1 <i>Continued from previous page</i>	160
A.1 <i>Continued from previous page</i>	161
A.1 <i>Continued from previous page</i>	162
A.1 <i>Continued from previous page</i>	163
B.1 <i>BE_qC and BKC(in parentheses) Table</i>	164
B.1 <i>Continued from previous page</i>	165
B.1 <i>Continued from previous page</i>	166
B.1 <i>Continued from previous page</i>	167
B.1 <i>Continued from previous page</i>	168
B.1 <i>Continued from previous page</i>	169
B.1 <i>Continued from previous page</i>	170
B.1 <i>Continued from previous page</i>	171
B.1 <i>Continued from previous page</i>	172
B.1 <i>Continued from previous page</i>	173
B.1 <i>Continued from previous page</i>	174
B.1 <i>Continued from previous page</i>	175
B.1 <i>Continued from previous page</i>	176
B.1 <i>Continued from previous page</i>	177
B.1 <i>Continued from previous page</i>	178
B.1 <i>Continued from previous page</i>	179
B.1 <i>Continued from previous page</i>	180
B.1 <i>Continued from previous page</i>	181

B.1 <i>Continued from previous page</i>	182
---	-----

Abbreviations

ARPA Advanced Research Projects Agency

AVC Arbitrarily Varying Channel

AWGN Additive White Gaussian Noise

BEC Binary Erasure Channel

BEqC Best Known Equivocation Codes

BKC Best Known Codes

BKLC Best Known Linear Codes

BSC Binary Symmetric Channel

DMC Discrete Memoryless Channel

GMP GNU Multi-Precision library

LDPC Low-Density Parity Check

pmf probability mass function

VoIP Voice over Internet Protocol

Chapter 1

Introduction

A traditional digital communication system consists of three basic elements: transmitter, channel, and receiver, as depicted in Fig. 1.1. In this model, digital data $\{0, 1\}$ from a transmitter are encoded for transmission over an unreliable channel. The transmitted signal propagates along the channel, the noise and interfering signals are added to the channel output, therefore the received signal is a corrupted version of the original transmitted data. The role of receiver is to decode the received data and try to recover the original data.

The internet began in 1969 when the US Department of Defense decided to establish an Advanced Research Projects Agency (*ARPA*) meant to protect the communications network during war time and as a result the *ARPA* network appeared and evolved during the eighties rapidly. In 1983, *ARPA* network split into two different networks: the *ARPA* network which was earmarked for civilian use and the *mil* network which was earmarked for military use.

Nowadays, the internet has become the universal communication between countries and is used in many different fields, such as information retrieval, commercial dealing and voice communication tools such as Skype, Viber and other VoIP technologies, including

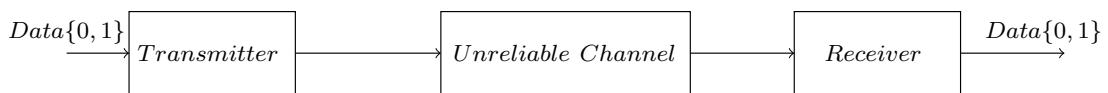


Figure 1.1: Traditional digital communication system

diplomatic contact among nations and military contact due to the ease and flexibility in operation of these systems and the freedom of roaming in and out of the workplace. In addition to all these advantages, there is also the cost-effective provision of the most important services provided by communication systems via the internet. It follows that the security of that information has become very important.

Many users of wireless communication can use network services such as email and internet applications. In addition, they have flexibility to move their smart devices from one place to another and to allow users to share data with network systems and other compatible devices. On the other hand, wireless networks transmit data through radio frequencies and are open to eavesdroppers unless protected. Eavesdroppers have exploited this openness to access systems, steal data and launch attacks that limit network bandwidth and reject service to authorised users.

For the purpose of maintaining the confidentiality and privacy of information, it is necessary to employ modern techniques to deal with such information. This could be done by using renewed methods to prevent intrusion on secret data in order to ensure the transfer of information securely through the channels of communication by taking into consideration the existence of another channel in addition to the main channel between the transmitter and the receiver, a so-called eavesdropping channel. Therefore, in 1975, Wyner [1] suggested a new type of communications channels dubbed the Wiretap channel, as shown in Fig. 1.2.

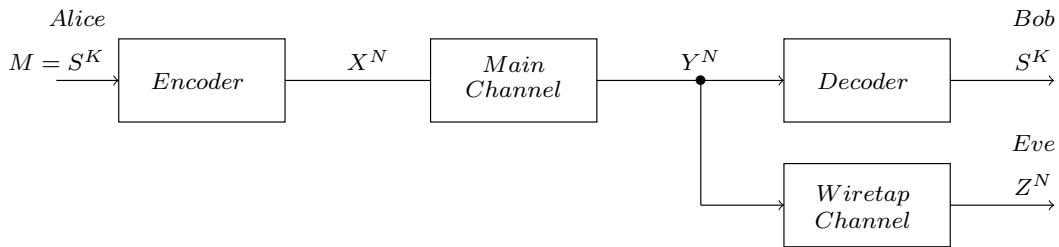


Figure 1.2: General wiretap channel

The wiretap channel contains one transmitter(*Alice*) and two receivers, one of which is the legitimate receiver(*Bob*) and the second is the eavesdropper(*Eve*). This model attempts to maximise the transmission rate of information between the transmitter and

legitimate receiver and to minimise the information leakage (i.e. maximise the equivocation, a mathematical description of secrecy) to the eavesdropper. We will describe more deeply this channel in Section 2.4.

1.1 Thesis Aims and Organisation

This thesis investigates the communication security for the wiretap channel when the main channel is an error free channel and the eavesdropper channel is a binary symmetric channel(*BSC*). The main aim of this thesis is to improve and guarantee the reliability of the communication security by increasing the equivocation rate in the eavesdropper side by using different techniques based on the syndrome coding scheme.

To investigate the issue of the communication security, firstly the impact of the best known codes¹ (*BKC*) are investigated on the wiretap channel by using a combination of the McEliece cryptosystem technique based on *BKC* coupled with syndrome coding scheme based on the (23, 12, 7) binary Golay code. A code design technique is proposed to construct new best known equivocation codes (BE_qC) for syndrome coding scheme.

Notably new codes with 15 parity bits are presented. In addition, we present a code design technique to extend the binary linear $[n, k]$ code to $[n + 2, k + 2]$ code to produce best known equivocation codes. Finally, we present a new code design technique which produces best known equivocation codes (BE_qC) with highest minimum distance (d_{min}) for the syndrome coding scheme which have better secrecy than the best known codes (*BKC*).

This thesis is organised into five chapters as follows:

Chapter 2 contains four main parts. The first part presents the software platform and tools used in this research, which includes the NTL Library and Magma software suite. The second part presents the syndrome coding scheme for the wiretap channel, in which all algorithms which includes the encoder, legitimate receiver's decoder and eavesdroppers decoder are described. Also, the encoding and decoding algorithms of the (23, 12, 7) binary Golay code which can correct all error patterns up to 3 bit error have been pre-

¹These are the highest performing error correcting codes that have so far been discovered.

sented as an example.

The third part of chapter 2 describes in detail the Best known linear code (33, 23, 5). The construction of the parity check matrix of this (33, 23, 5) code has been done using two methods, the first method based on the evaluation of the primitive root of the irreducible polynomial which required to calculate the values of all columns of the parity check matrix. In the second method, the parity check matrix of the (33, 23, 5) code can be obtained directly by using Magma software.

In the last part of chapter 2, a literature review for wiretap channel, syndrome coding scheme and code design techniques have been presented.

In chapter 3, two models are proposed to improve the security of the communication channel by maximise the equivocation rate in the eavesdropper channel. The secrecy coding for the wiretap channel using a Best Known Linear Code (*BKLC*) has been implemented. The designed models includes an inner code and an outer code, the outer code employs a syndrome coding scheme based on the (23, 12, 7) binary Golay code and the inner code employs the McEliece cryptosystem technique based on *BKLC*.

Chapter 4 presents many techniques to construct Best Known Equivocation Codes(BE_qC) for syndrome coding in the wiretap channel. Techniques are presented based on a recursive method for evaluation the equivocation rate of any linear, binary code when used in syndrome coding for the Binary Symmetric Channel(*BSC*).

This chapter can be divided into three main parts. The first part presents a code design technique to extend a set of good equivocation codes of length $[n, k]$ to $[n + 1, k + 1]$ code which have better equivocation rate for syndrome coding scheme. Also, all the best known equivocation codes(BE_qC) have been enumerated for the case where the number of parity bits of the code is equal to 15.

The second part presents a code design technique for producing the best known equivocation codes of highest minimum distance for the syndrome coding scheme by using a combination of the best known equivocation code design technique coupled with technique of the determination of the highest minimum distance. Also, best known equivo-

cation codes are listed in exhaustive tables for various values of parity bits of the code. In the last part of chapter 4, a code design technique has been proposed to produce best known equivocation codes by extending a set of good equivocation codes of length $[n, k]$ to $[n + 2, k + 2]$ code.

In chapter 5, the work presented in this thesis is concluded and some suggestions for future work in this area are given.

1.2 Contribution to Knowledge

The following list summarises the main contributions of the dissertation.

- **An implementation of secrecy coding for a special case of wiretap channel using a best known linear code.** A special case of wiretap channel is analysed when the main channel is an error free channel and the eavesdropper channel is a binary symmetric channel. New two models are proposed to increase the equivocation rate in the eavesdropper side and as a result the communication security is improved. The design of two models are based on the combination of the technique of the McEliece cryptosystem using best known linear code(*BKLC*) coupled with syndrome coding. The implementation results show that the performance of the proposed models considerably reduce the information leakage to the eavesdropper compared to previously published schemes.

This contribution was presented by the author and published in the following conference:

S. Al-Hassan, M. Ahmed, and M. Tomlinson,“Secrecy coding for the wiretap channel using best known linear codes”, IEEE Conference, The 5th Global Information Infrastructure and Networking Symposium , Trento, Italy, October 2013.

- **Modifying the syndrome coding scheme.** In the traditional syndrome coding, it is necessary to generate a look up table for error patterns and syndromes, but for long codes a syndrome table is impractical. Therefore we propose an encoding algorithm that shows this look up table is unnecessary and that the parity check matrix H of the code is sufficient.
- **A recursive method to calculate the equivocation rate of any linear code of syndrome coding scheme.** The traditional method for evaluating the equivocation rate works well for short codes, but for long codes it is impracticable. Therefore we propose an efficient recursive method to calculate the equivocation rate for the binary symmetric channel(BSC) and any linear binary code. This method is based on the evaluation of the probability mass function of the syndromes of a code which depends only on the columns of the parity check matrix and the probability of error of the binary symmetric channel.
- **Enumeration of new best known equivocation codes for syndrome coding with 15 parity bits.** A code design technique to extend the binary linear $[n, k]$ code to $[n+1, k+1]$ code to produce best known equivocation codes($BEqC$) for syndrome coding in wiretap channel is presented. The code design technique is based on the realisation that the syndrome probability mass function(pmf) of a new extended code is a function of the original code and good equivocation codes produce good extended codes. The design results for a given number of parity bits of the code ($m = 15$) show that the equivocation rate of these new BE_qCs have been increased by a large margin compared to all previously published best error correcting codes, the best known codes(BKC) compiled by Grassl [2].

This contribution was presented by the author and published in the following conference:

S. Al-Hassan, M. Ahmed, and M. Tomlinson,“New Best Equivocation Codes for Syndrome Coding”, IEEE Conference, International Conference on Information and Communication Technology Convergence (ICTC), Busan, South Korea, October 2014.

- **Construction of best known equivocation codes by adding two best columns to the parity check matrix of $[n, k]$ code.** The code design technique that produce the best known equivocation codes (BE_qC) for syndrome coding has been presented by extending the binary linear $[n, k]$ code to $[n + 2, k + 2]$ code. This technique is based on the extension of the parity check matrix of a good $BE_qC, [n, k]$ code by selecting the two best columns that extend the length of the code. The results obtained show that the BE_qC that are produced by adding two columns gives better equivocation rate compared to those codes that are generated by the addition of one column in two phases.

This contribution was presented by the author and published in the following conference:

S. Al-Hassan, M. Ahmed, and M. Tomlinson, "Extension of the Parity Check Matrix to Construct the Best Equivocation Codes for Syndrome Coding", IEEE Conference, Global Information Infrastructure and Networking Symposium (GIIS), Montreal, QC, Canada, September 2014.

- **Construction of new best known equivocation codes with highest minimum Hamming distance for syndrome coding.** To date, all the best known equivocation codes from recent research have a respectable minimum Hamming distance, but are sometimes not good as the best known code with the same parameters. Therefore, the most interesting goal in coding theory is to find new best known equivocation codes such that the minimum distance is maximal. The minimum distance of a linear code is an important parameter that lead to the mechanism of detection on correction of transmission errors by the code.

The new best known equivocation codes of highest minimum distance have been determined by using a combination of a code design technique based on extension of the parity check matrix from an optimal set of good equivocation codes coupled with the technique of the determination of the highest minimum distance of these codes.

Code examples have been presented for a given number of parity bits of the code ($m = 7, 11, 12$) and the design results show that the equivocation rate of these best known equivocation codes of highest minimum distance exceeds by a large margin the best known error correcting codes having the same parameters.

This contribution was presented by the author and published in the following conference:

S. Al-Hassan, M. Ahmed, and M. Tomlinson,“Construction of Best Equivocation Codes with Highest Minimum Distance for Syndrome Coding”, IEEE International Conference on Communication, IEEE ICC 2015 - Workshop on Wireless Physical Layer Security, London, UK, June 2015.

Chapter 2

Syndrome Coding Scheme and Best Known Linear Codes

2.1 Software Platform and Tools Used

2.1.1 NTL Library

NTL library was first introduced by Victor Shoup in 1990. It is a high-performance, portable C++ library providing data structures and algorithms for arbitrary length integers; for vectors, matrices, and polynomials over the integers and over finite fields; and for arbitrary precision floating point arithmetic [3].

NTL provides tools and high quality implementations of algorithms for:

- arbitrary length integer arithmetic and arbitrary precision floating point arithmetic.
- Polynomial arithmetic over the integers and finite fields including polynomial factorisation, irreducibility testing and computation of minimal polynomials.
- Basic linear algebra over the integer, finite fields, and arbitrary precision floating point numbers.

NTL can be easily installed on any platform, including virtually any 32- or 64-bit machine running any version of *Ubuntu*, as well as PCs running *Windows* or *NT*, and

Macintoshes. NTL achieves this portability by avoiding esoteric C++ features, and by avoiding assembly code; it should therefore remain usable for years to come with little or no maintenance, even as processors and operating systems continue to change and evolve. However, NTL can be used in conjunction with *GMP* (the *GNU Multi-Precision library*) for enhanced performance. NTL can also be used in conjunction with the *gf2x* library for faster arithmetic of large degree polynomials over GF(2).

There are many available modules in NTL, the most important modules that have been used in this work are:

- GF2 class : deals with integers mod 2.
- GF2X class : deals with Polynomials over GF(2).
- mat_GF2 class : deals with matrices over GF(2); includes basic matrix arithmetic operations, determinant calculation, matrix inversion, solving non-singular systems of linear equations and Gaussian elimination.
- vec_GF2 class : deals with vectors over GF(2), with arithmetic.

2.1.2 Magma Software Suite

The Magma Software was introduced by the Computational Algebra Group at the University of Sydney. The first release of Magma V1.30 was proposed in 1996.

Magma is a computer algebra system designed to provide a software environment to solve problems in algebra, number theory, algebraic geometry and combinatorics that may contain hard computations. Also, it enables the users to compute with structures such as groups, rings, fields, modules, schemes, curves, graphs, codes designs and many others. The main platforms that are supported by Magma:

- Linux(i386/PC, AMD64/Intel64, IBM PowerPC64, Intel IA64).
- Mac OSX(64-bit Intel).
- Solaris(AMD64/Intel64, Sparc 64-bit).

- Windows(32-bit).

Magma Software has databases for Best Known Linear Code(*BKLC*) over $GF(q) = 2, 3, 4, 5, 6, 7, 8, 9$. The database for codes over $GF(2)$ contains constructions of best codes of length up to $n_{max}=256$. By giving any two parameters of the code: length(n), dimension(k) and minimum distance(d), then Magma will return the code with the best value of the omitted parameter [4]. The construction of the *BKLC* tables was achieved by Grassl [2].

2.2 Definition

2.2.1 Binary symmetric channel(*BSC*)

This is realised in code by using the NTL Library:

```
NTL:: Vec_GF_2 BSC(NTL:: Vec_GF_2 C, float pe)
{
    for (int i=0; i<C.length(); i++)
        if (drand48() < pe)
            C[i] += 1;
    return C;
}
```

2.2.2 Definition of Information Theory

Entropy

The Entropy(H) of a discrete random variable X is a measure of the amount of uncertainty(or information) related with the value of X [5,6]. Assume $p(x)$ is the probability of some $x \in X$, then the entropy(H) of X is defined mathematically in Eq.(2.1):

$$H(X) = - \sum_{x \in X} p(x) \log_2 p(x) \quad (\text{bits}) \quad (2.1)$$

The special case of information entropy for a binary random variable which takes two values with probabilities p and $1 - p$ is the binary entropy function:

$$H(p) = -p \log_2 p - (1 - p) \log_2(1 - p) \quad (2.2)$$

Joint Entropy

The Joint Entropy $H(X, Y)$ of a pair of discrete random variables (X, Y) is a measure of the amount of information in X and Y [5, 6]. Assume $p(x, y)$ is the joint probability of some $x \in X$ and $y \in Y$, then the joint entropy $H(X, Y)$ is defined mathematically as follows:

$$H(X, Y) = - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 p(x, y) \quad (2.3)$$

Equivocation

The Equivocation (or conditional entropy) quantifies the remaining uncertainty of a random variable Y given that the value of another random variable X is known. Assuming $p(x, y)$ and $p(y | x)$ are expressions of the joint and conditional probabilities, then the equivocation is defined mathematically in Eq.(2.4) [7, 8]:

$$H(Y | X) = - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 p(y | x) = \sum_{x \in X} p(x) \log_2 H(Y | X = x) \quad (2.4)$$

the Equivocation can be written as:

$$H(Y | X) = H(X, Y) - H(X) \text{ or } H(X | Y) = H(X, Y) - H(Y)$$

Channel Capacity

Channel Capacity (or Mutual Information) $I(X; Y)$ is the common information between X and Y . It can be expressed as [6]:

$$I(X; Y) = H(X) - H(X | Y)$$

$$I(X; Y) = H(Y) - H(Y | X)$$

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

Shannon's Capacity Theorem

In 1948, Shannon [9] published his paper which includes an interesting theorem in information theory that give us an idea about the information rate that can be transmitted with low probability of error. Shannon's Capacity Theorem (or the Shannon-Hartley Theorem) states that the channel capacity of a channel C of bandwidth B , perturbed by additive white Gaussian noise (*AWGN*) is given by:

$$C = B \log_2\left(1 + \frac{S}{N}\right) \text{ bits/second} \quad (2.5)$$

where S is the average transmitted signal power and N is the average noise power. Shannon showed that the transmission over a noisy channel with low probability of error is possible as long as the information rate R does not exceed the channel capacity C (i.e. $R \leq C$).

2.2.3 Definition of Coding System

Discrete Memoryless Channel(*DMC*)

A discrete memoryless channel (*DMC*) can be described in terms of an input alphabet $X = \{x_1, x_2, \dots, x_M\}$, an output alphabet $Y = \{y_1, y_2, \dots, y_M\}$, and a likelihood function (probability transition matrix) $p(y | x)$ as shown in Fig. 2.1. The channel is said to be memoryless if the output depends only on the input at that time and is statistically independent of the outputs or inputs at other times [10].

All transition probabilities from x_i to y_i are placed in a transition matrix of *DMC* as follows:

$$p(y_i | x_i) = \begin{bmatrix} p(y_1 | x_1) & p(y_2 | x_1) & \cdots & p(y_M | x_1) \\ p(y_1 | x_2) & p(y_2 | x_2) & \cdots & p(y_M | x_2) \\ \vdots & \vdots & \cdots & \vdots \\ p(y_1 | x_M) & p(y_2 | x_M) & \cdots & p(y_M | x_M) \end{bmatrix}$$



Figure 2.1: Discrete Memoryless Channel

Linear code

An $[n, k, d]$ code C over a field F_q of q symbols and q^k codewords is said to be a q -ary linear $[n, k, d]$ code of length n and dimension k if C is a k -dimensional subset of F_q^n . Each vector in the k -dimensional subset of F_q^n is called a codeword and can be expressed as $c = (c_0, c_1, \dots, c_{n-1})$. The symbol d represents the minimum Hamming distance of the code. The rate of a linear $[n, k, d]$ code is $R = k/n$. For a linear code, any linear combination of codewords is a codeword and this property led to the use of the term ‘linear’ for such a code [11].

Binary Linear Code

A binary code C of length n and dimension k is said to be linear if for all codewords(2^k) $c_1, c_2 \in C$, $c_1 + c_2 \in C$. A linear code C must contain the zero codeword. For example, the following code $C_1 = \{000, 001, 010, 011\}$ is a linear code, because all the sums are in C_1

$$000 + 001 = 001$$

$$001 + 010 = 011$$

$$001 + 011 = 010$$

$$010 + 011 = 001$$

And so on. But $C_2 = \{000, 001, 101\}$ is not a linear code, since 001 and 101 are in C_2 but $(001+101=100)$ is not in C_2 [12].

Dual Code

For any code $C \subset F^n$, the dual code $C^\perp \subset F^n$ can be defined by

$$C^\perp = \{x \in F^n \mid \langle x, y \rangle = 0 \ \forall y \in C\} \quad (2.6)$$

i.e., C^\perp consists of all vectors in F^n which are orthogonal to all the vectors in C [11].

Generator Matrix (G)

A generator matrix G for a linear code C is a binary matrix whose rows are the codewords belonging to some basis for the code [11, 13]. A generator matrix G for $[n, k, d]$ code has dimension $k \times n$, which contains k linearly independent codewords of C , and every codeword c in C can be represented as a linear combination of these codewords.

Let $w = [w_0 \ w_1 \ \cdots \ w_{k-1}]$, then $c = wG$, and every codeword $c \in C$ has such representation for some vector w . The generator matrix G can be transformed into systematic form by elementary row operations. The systematic form for a generator matrix is

$$G = [I_k \mid P]$$

Where the first k coordinates of G are an identity matrix I_k of dimension $k \times k$ and P represents the remaining $n - k$ coordinates of G .

Parity Check Matrix (H)

A Parity Check Matrix H of a linear code C is a generator matrix of the dual code. It can be derived from the generator matrix G of the code (and vice-versa) [11]. If the generator matrix for an $[n, k, d]$ code is in systematic form

$$G = [I_k \mid P]$$

Then the parity check matrix H of dimensions $(n - k) \times n$ is given by

$$H = [-P^T \mid I_{n-k}]$$

because $G \cdot H^T = 0$.

Syndrome

Let C be a linear $[n, k, d]$ code over $F = GF(q)$ and H be a parity check matrix of C .

Then, for a given vector $c \in F_q^n$, the vector $s \in F_q^{n-k}$ defined by

$$s = cH^T \quad (2.7)$$

is the syndrome of a code whose parity check matrix is H . If the vector $c \in C$, then $s = 0$. That is, the codewords of C are strictly the vectors of F_q^n whose syndromes are 0 [13].

Hamming weight

The Hamming weight $wt(c)$ of a codeword $c = (c_0, c_1, \dots, c_{n-1})$ is the number of non zero components in c . The minimum weight w_{min} of a code C is the smallest Hamming weight of any nonzero codeword [11, 13]. That is

$$w_{min} = \min_{c \in C, c \neq 0} wt(c) \quad (2.8)$$

Minimum Hamming Distance

If the code C has at least two distinct codewords (c_1, c_2) , the minimum Hamming distance d of C is the smallest distance between c_1 and c_2 [5]. We can represent d mathematically as

$$d = d(C) = \min\{d(c_1, c_2) \mid c_1, c_2 \in C, c_1 \neq c_2\} \quad (2.9)$$

Any linear code C of length n , with dimension k , and with minimum distance d can be represented in the form of $[n, k, d]$ code. The minimum distance of any code must be at

least 1, because any two distinct codewords in C are not equal. A code C of minimum distance d is capable of correcting all error patterns of weight $t \leq \lfloor (d-1)/2 \rfloor$.

The Hamming Bound

For any linear code C with the given values of length n and distance ($d = 2t + 1$), the Hamming bound is [Eq.(2.10)]:

$$|C| \leq \frac{2^n}{\binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{t}} \quad (2.10)$$

The Hamming bound is an upper bound for the size or dimension k of a linear code C .

A code C of distance d will correct all error patterns of weight $t \leq \lfloor (d-1)/2 \rfloor$ [12].

The Hamming bound is useful to calculate the dimension of the linear code C that has length n and distance d . Also, it is used to investigate the perfect code that attains Eq.(2.10).

Perfect Codes

Any code C has length n and odd distance $d = 2t + 1$ is called a perfect code if C satisfies the Hamming bound, i.e.,

$$|C| = \frac{2^n}{\binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{t}} \quad (2.11)$$

The main condition of binary perfect codes is that the denominator of $|C|$ must be a power of 2 [12]. Perfect codes are very useful in the coding systems that deal with error correcting codes. An example of a perfect code is the (23,12,7) binary Golay code because it has ability to correct all error patterns t up to and including 3 (where $t \leq \lfloor (d-1)/2 \rfloor$).

Coset of linear code

Assume C be an (n, k) linear code over $\text{GF}(q)$ and x be any vector of length n . Then the subset

$$x + C = \{x + y \mid y \in C\}$$

is called a coset of C . Each coset $x + C$ of a linear code has q^k elements and the number of different cosets is q^{n-k} [13].

Coset leader

A word of a coset that has a minimum weight in that coset is called a coset leader.

Best Known Linear Code (*BKLC*)

If the $[n, k]$ linear code C has a highest minimum distance d among all known $[n, k]$ linear codes, the code C will be called a Best Known Linear Code(*BKLC*) [4].

It is clear from the definition that *BKLC* has a largest value of d among the other codes. Since d is associated with the number of error patterns t (where $t \leq \lfloor (d-1)/2 \rfloor$), this means that the use of *BKLC* will give us the ability to investigate and correct the largest possible number of errors in the linear code.

Cyclotomic Coset

Assume p and $p^m - 1$ are coprime. The cyclotomic coset of p modulo $p^m - 1$ containing s for any s with $0 \leq s < p^m - 1$ is defined by:

$$C_s = \{(sp^j \bmod (p^m - 1)) : 0 \leq j < m\} \quad (2.12)$$

The cyclotomic coset of s is denoted by:

$$C_s = \{s, ps, p^2s, \dots, p^{m_s-1}s\} (\bmod (p^m - 1)) \quad (2.13)$$

where m_s is the smallest positive integer such that $p^{m_s} s \equiv s \pmod{p^m - 1}$.

The smallest element in C_s is s and called the coset representative [12]. For example, the cyclotomic cosets mod 15 (with $p = 2$) can be calculated as follows:

$p^m - 1 = 2^4 - 1$, this means that the maximum number of elements for each cosets is equal to 4. For $(0 \leq s < 15)$, $C_s = \{(sp^j \bmod(15)) : 0 \leq j < 4\}$, then the cyclotomic cosets are as following:

$$\begin{aligned} C_0 &= \{0\}, \\ C_1 &= \{1, 2, 4, 8\}, \\ C_3 &= \{3, 6, 9, 12\}, \\ C_5 &= \{5, 10\}, \\ C_7 &= \{7, 11, 13, 14\}, \end{aligned}$$

Also, the coset representative is 0,1,3,5,7.

Quadratic Residue Code

An integer Q is said to be quadratic residue modulo n if the congruence $x^2 \equiv Q \pmod{n}$ has a solution. Otherwise, Q is a quadratic non-residue modulo n [14]. If n is an odd prime, then n has $(\frac{Q-1}{2})$ quadratic residue and $(\frac{Q-1}{2})$ quadratic non-residue. The quadratic residue are congruent modulo n to the integers $\{1^2, 2^2, \dots, (\frac{Q-1}{2})^2\}$; that is

$$Q_n = \{j^2 \bmod(n) : j = 1, 2, \dots, (\frac{n-1}{2})\} \quad (2.14)$$

The knowledge of quadratic residues of a number is very useful to find its factors. For example, the set of quadratic residues modulo 17 can be calculated as follows :

$$\begin{aligned} Q_{17} &= \{j^2 \bmod(17) : j = 1, 2, \dots, (\frac{17-1}{2})\} \\ Q_{17} &= \{1, 2, 4, 8, 9, 13, 15, 16\} \end{aligned}$$

Also, let N_{17} be the set of quadratic non-residues, then

$$N_{17} = \{3, 5, 6, 7, 10, 11, 12, 14\}$$

The Standard Array of the Linear Code

A standard array of a linear $[n, k, d]$ code is a $(q^{n-k} \times q^k)$ array that lists all elements of a particular vector space F_q^n . It plays an important role in the decoding the linear codes, thereby helping us to find the corresponding codeword for any received vector.

The standard array is configured as follows [15]:

1. The first row lists all codewords starting with the 0 codeword on the left.
2. Each row is a coset with the coset leader in the first column.
3. Each codeword in the array is the sum of the codeword at the top of its column and the coset leader at the far left of its row.

2.3 The Binary Symmetric Channel(*BSC*)

The binary symmetric channel(*BSC*) is one of the simplified communication channels model that are used in coding systems. The *BSC* deals with binary symbols; so, it can be transmit only one bit 0 or 1 per unit of time with a probability of error $p_e < \frac{1}{2}$. The simplified form of *BSC* is shown in Fig. 2.2 [16]. Assume a transmitter, Alice sends a message M_i to a receiver, Bob through the *BSC*. The probability of receiving an

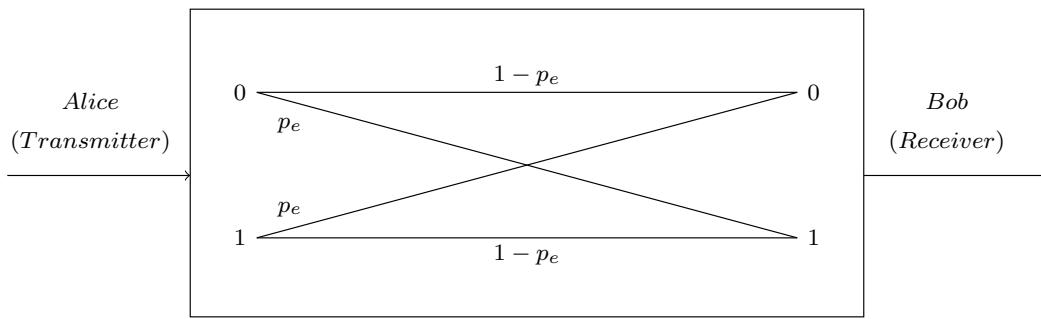


Figure 2.2: Binary Symmetric Channel (BSC)

incoming message M_i without any errors is $(1-p_e)$, and received in error with probability p_e . The *BSC* adds errors to the input sequence before transmission, which are generated as a random error sequence of zeros and ones. The transmitted message C_i that can be

seen by Bob is:

$$C_i = M_i + E_i \quad (2.15)$$

Where $C_i \in \{0, 1\}$ are the output bits, $M_i \in \{0, 1\}$ are the input bits, $E_i \in \{0, 1\}$ are the possible error bits. If $E_i = 1$, that mean an error occurs on bit i [17].

For example, assume Alice sends a message $M_i = 01101101$ and the *BSC* adds randomly an error sequence $E_i = 00100101$. The transmitted message C_i to Bob can be calculated by using modulo 2 addition as follows:

$$C_i = M_i + E_i$$

$$C_i = (01101101) + (00100101)$$

$$C_i = (01001000)$$

As mentioned above, a *BSC* is a channel has probability of error equal to p_e . Assume that M (input) is sent through a *BSC* and C (output) is received, then the conditional probabilities of *BSC* can be characterised as :

$$P_r(C = 0 | M = 0) = 1 - p_e$$

$$P_r(C = 0 | M = 1) = p_e$$

$$P_r(C = 1 | M = 0) = p_e$$

$$P_r(C = 1 | M = 1) = 1 - p_e$$

Where $0 \leq p_e \leq \frac{1}{2}$.

Therefore, the *BSC* matrix is

$$\begin{bmatrix} 1 - p_e & p_e \\ p_e & 1 - p_e \end{bmatrix}$$

The capacity of *BSC* can be expressed in terms of the binary entropy function $H(p_e)$:

$$\text{Capacity} = 1 - H(p_e) \quad (2.16)$$

Where

$$H(p_e) = -p_e \log p_e - (1 - p_e) \log(1 - p_e) \quad (2.17)$$

The special case of the eavesdropper channel model is shown in Fig. 2.3.

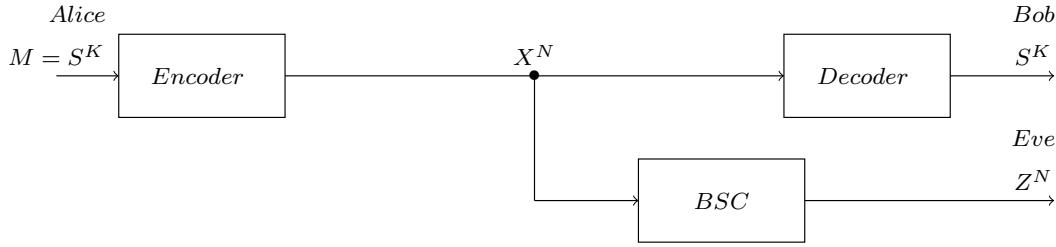
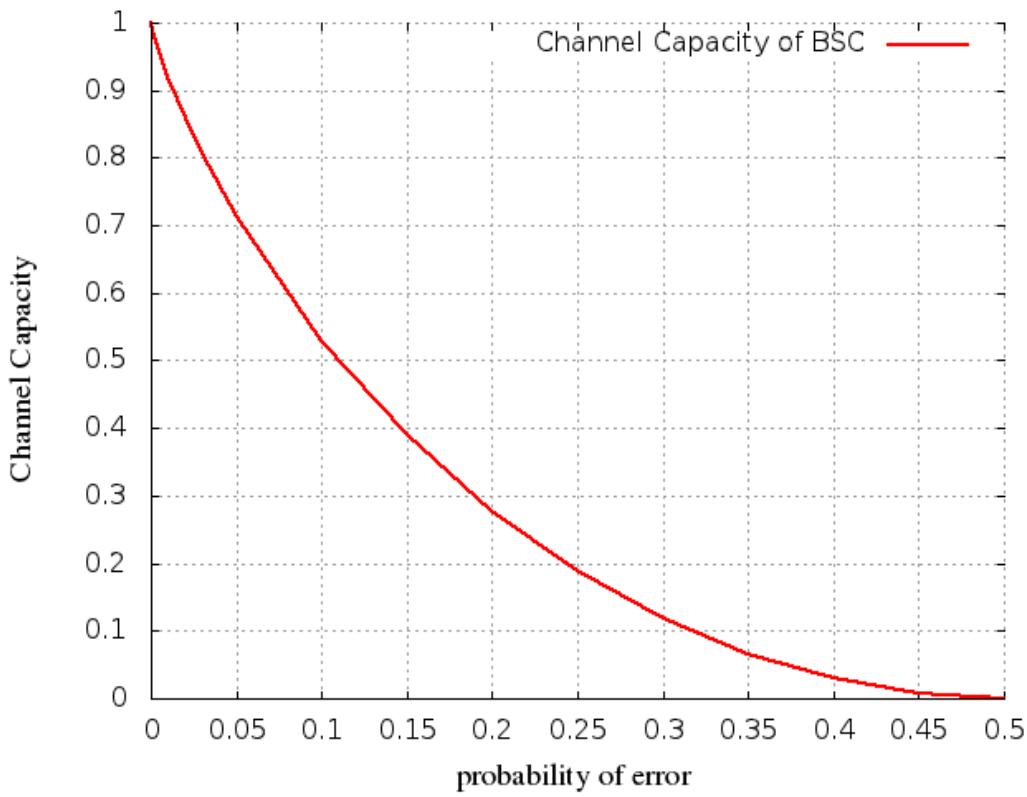
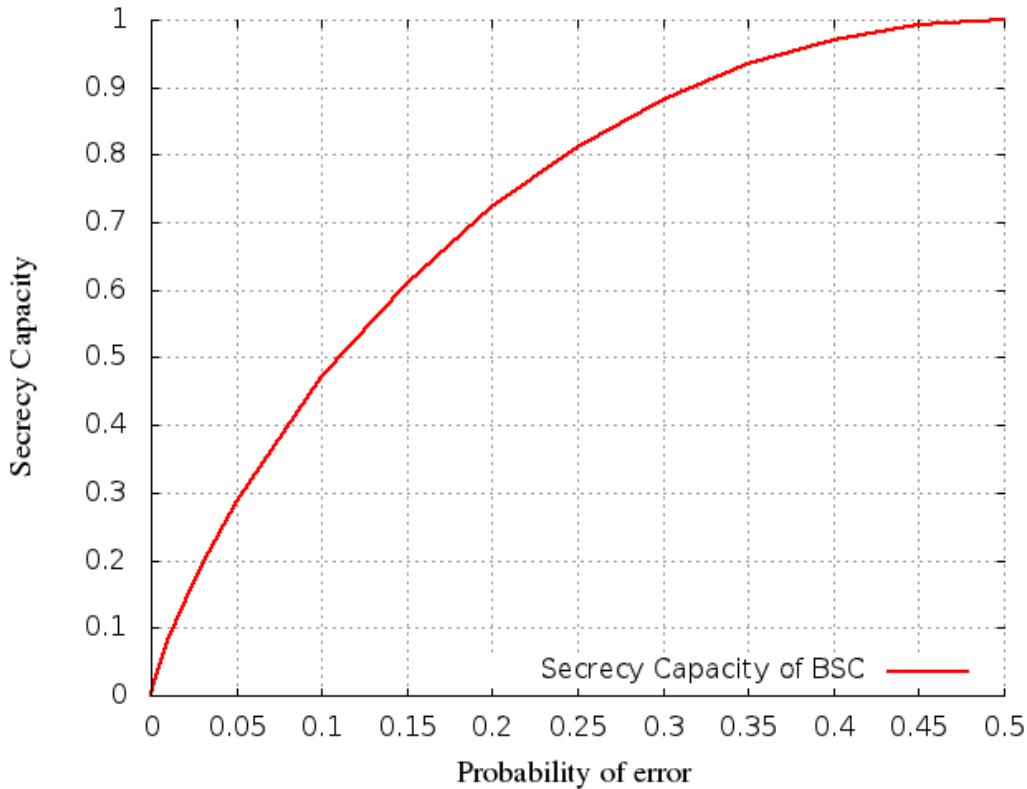


Figure 2.3: Special case of the eavesdropper channel

The model assumes the main channel(between Alice and Bob) is an error-free channel and the eavesdropper chanle is a binary symmetric channel (*BSC*) with probability of error p_e . The secrecy capacity of *BSC* represents the maximum equivocation rate at the eavesdropper when the maximum transmission rate has been achieved [18]. Therefore, the secrecy capacity S_c of a *BSC* can be calculated from the following equation:

$$S_c = 1 - \text{Capacity} \quad (2.18)$$

Fig. 2.4 and Fig. 2.5 shows the channel capacity and secrecy capacity(Equivocation) of *BSC* as a function of probability of error p_e .

Figure 2.4: Channel Capacity of 1-bit *BSC*Figure 2.5: Secrecy Capacity (Equivocation) of 1-bit *BSC*

2.3.1 Worked example of 2-bit BSC

For the Fig. 2.6 , draw the curves of capacity and equivocation of the 2-bit BSC in the case of 1-error occurs, for the following probability of errors:

$$p_e = 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5$$

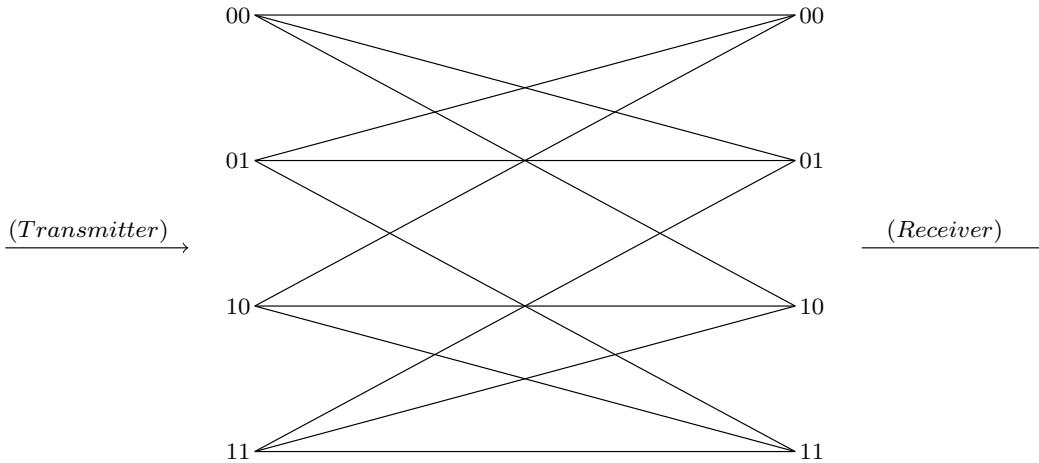


Figure 2.6: 2-bit BSC Example

1) For $p_e = 0.01$, from Table 2.1, $H(x) = \sum Pr \times I = (0.25 \times 2) \times 4 = 2$

Tx	Probability(Pr)	Information(I)
00	0.25	$-\log_2 Pr = 2$
01	0.25	2
10	0.25	2
11	0.25	2

Table 2.1: Transmitter Entropy $H(x)$ for $p_e = 0.01$

From Table 2.2, $H(y) = \sum Pr \times I = (0.25 \times 2) \times 4 = 2$

Rx	Probability(Pr)	Information(I)
00	0.25	$-\log_2 Pr = 2$
01	0.25	2
10	0.25	2
11	0.25	2

Table 2.2: Receiver Entropy $H(y)$ for $p_e = 0.01$
 $I(x,y) = H(x)-H(y)-H(x,y)$

From Table 2.3,

$$\begin{aligned} H(x,y) &= \sum Pr \times I = [(0.245025 \times 2.029)(0.002475 \times 8.658) \times 2(2.5 \times 10^{-5} \times 15.2877)] \times 4 \\ &= 2.1615 \end{aligned}$$

$$\text{Capacity} = I(x,y) = H(x) + H(y) - H(x,y) = 2 + 2 - 2.1615 = 1.8384$$

$$\text{Equivocation} = H(x,y) - H(y) = 2.1615 - 2 = 0.1615$$

Tx	Rx	Probability(Pr)	Information(I)
00	00	$0.25 \times 0.99 \times 0.99 = 0.245025$	2.029
	01	$0.25 \times 0.99 \times 0.01 = 0.002475$	8.658
	10	$0.25 \times 0.01 \times 0.99 = 0.002475$	8.658
	11	$0.25 \times 0.01 \times 0.01 = 2.5 \times 10^{-5}$	15.2877
01	00	$0.25 \times 0.99 \times 0.01 = 0.002475$	8.658
	01	$0.25 \times 0.99 \times 0.99 = 0.245025$	2.029
	10	$0.25 \times 0.01 \times 0.01 = 2.5 \times 10^{-5}$	15.2877
	11	$0.25 \times 0.01 \times 0.99 = 0.002475$	8.658
10	00	$0.25 \times 0.01 \times 0.99 = 0.002475$	8.658
	01	$0.25 \times 0.01 \times 0.01 = 2.5 \times 10^{-5}$	15.2877
	10	$0.25 \times 0.99 \times 0.99 = 0.245025$	2.029
	11	$0.25 \times 0.99 \times 0.01 = 0.002475$	8.658
11	00	$0.25 \times 0.01 \times 0.01 = 2.5 \times 10^{-5}$	15.2877
	01	$0.25 \times 0.01 \times 0.99 = 0.002475$	8.658
	10	$0.25 \times 0.99 \times 0.01 = 0.002475$	8.658
	11	$0.25 \times 0.99 \times 0.99 = 0.245025$	2.029

Table 2.3: Joint Entropy $H(x,y)$ for $p_e = 0.01$

$$2) \text{ For } p_e = 0.5, \text{ from Table 2.4, } H(x) = \sum Pr \times I = (0.25 \times 2) \times 4 = 2$$

Tx	Probability(Pr)	Information(I)
00	0.25	$-\log_2 Pr = 2$
01	0.25	2
10	0.25	2
11	0.25	2

Table 2.4: Transmitter Entropy $H(x)$ for $p_e = 0.5$

$$\text{From Table 2.5, } H(y) = \sum Pr \times I = (0.25 \times 2) \times 2 = 2$$

$$\text{From Table 2.6, } H(x,y) = \sum Pr \times I = [(0.0625 \times 4)] \times 16 = 4$$

$$\text{Capacity} = I(x,y) = H(x) + H(y) - H(x,y) = 2 + 2 - 4 = 0$$

$$\text{Equivocation} = H(x,y) - H(y) = 4 - 2 = 2$$

Rx	Probability(Pr)	Information(I)
00	0.25	$-\log_2 Pr = 2$
01	0.25	2
10	0.25	2
11	0.25	2

Table 2.5: Receiver Entropy $H(y)$ for $p_e = 0.5$

Tx	Rx	Probability(Pr)	Information(I)
00	00	$0.25 \times 0.5 \times 0.5 = 0.0625$	4
00	01	$0.25 \times 0.5 \times 0.5 = 0.0625$	4
00	10	$0.25 \times 0.5 \times 0.5 = 0.0625$	4
00	11	$0.25 \times 0.5 \times 0.5 = 0.0625$	4
01	00	$0.25 \times 0.5 \times 0.5 = 0.0625$	4
01	01	$0.25 \times 0.5 \times 0.5 = 0.0625$	4
01	10	$0.25 \times 0.5 \times 0.5 = 0.0625$	4
01	11	$0.25 \times 0.5 \times 0.5 = 0.0625$	4
10	00	$0.25 \times 0.5 \times 0.5 = 0.0625$	4
10	01	$0.25 \times 0.5 \times 0.5 = 0.0625$	4
10	10	$0.25 \times 0.5 \times 0.5 = 0.0625$	4
10	11	$0.25 \times 0.5 \times 0.5 = 0.0625$	4
11	00	$0.25 \times 0.5 \times 0.5 = 0.0625$	4
11	01	$0.25 \times 0.5 \times 0.5 = 0.0625$	4
11	10	$0.25 \times 0.5 \times 0.5 = 0.0625$	4
11	11	$0.25 \times 0.5 \times 0.5 = 0.0625$	4

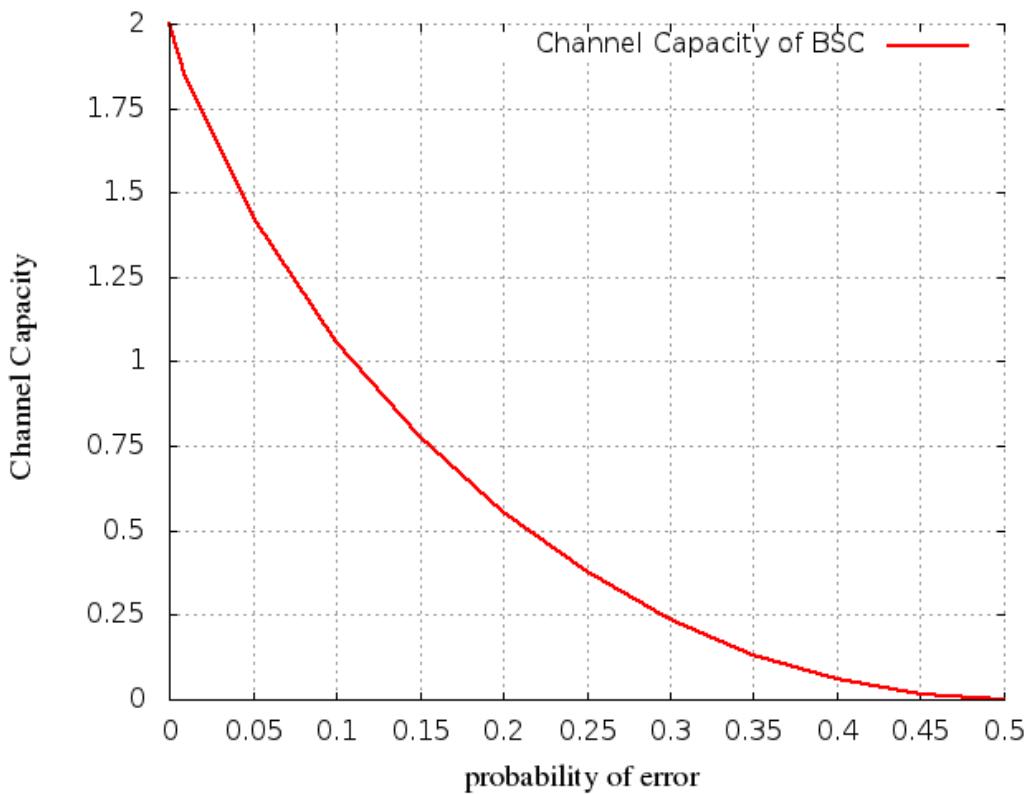
Table 2.6: Joint Entropy $H(x,y)$ for $p_e = 0.5$

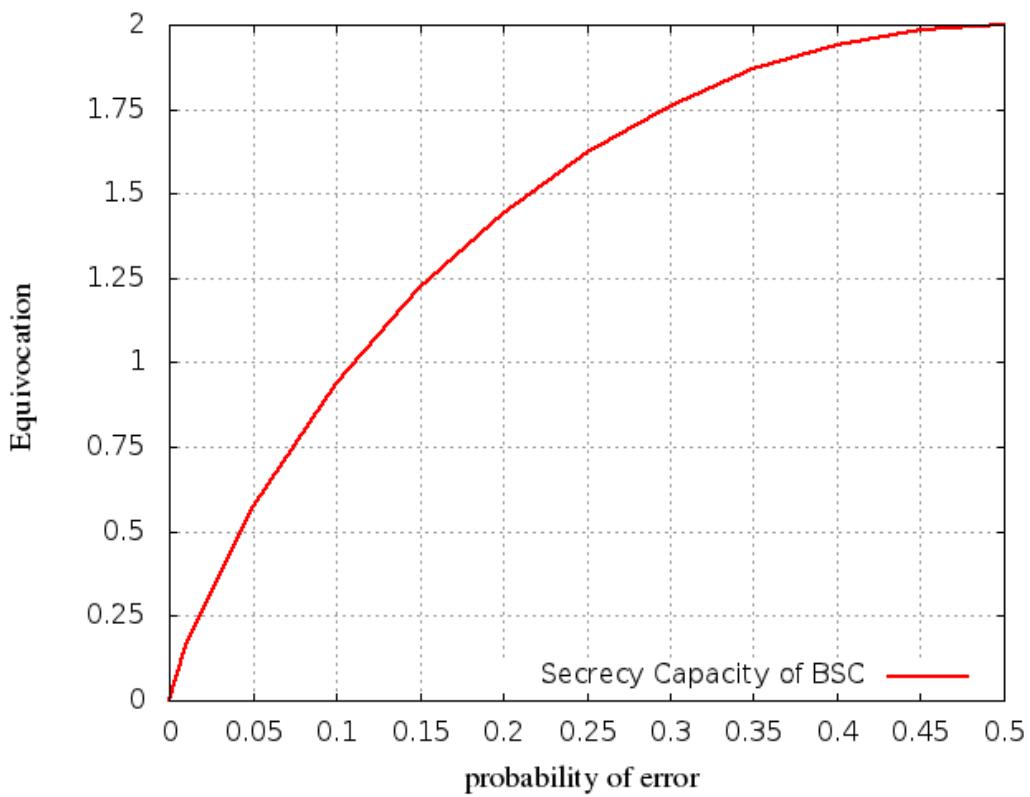
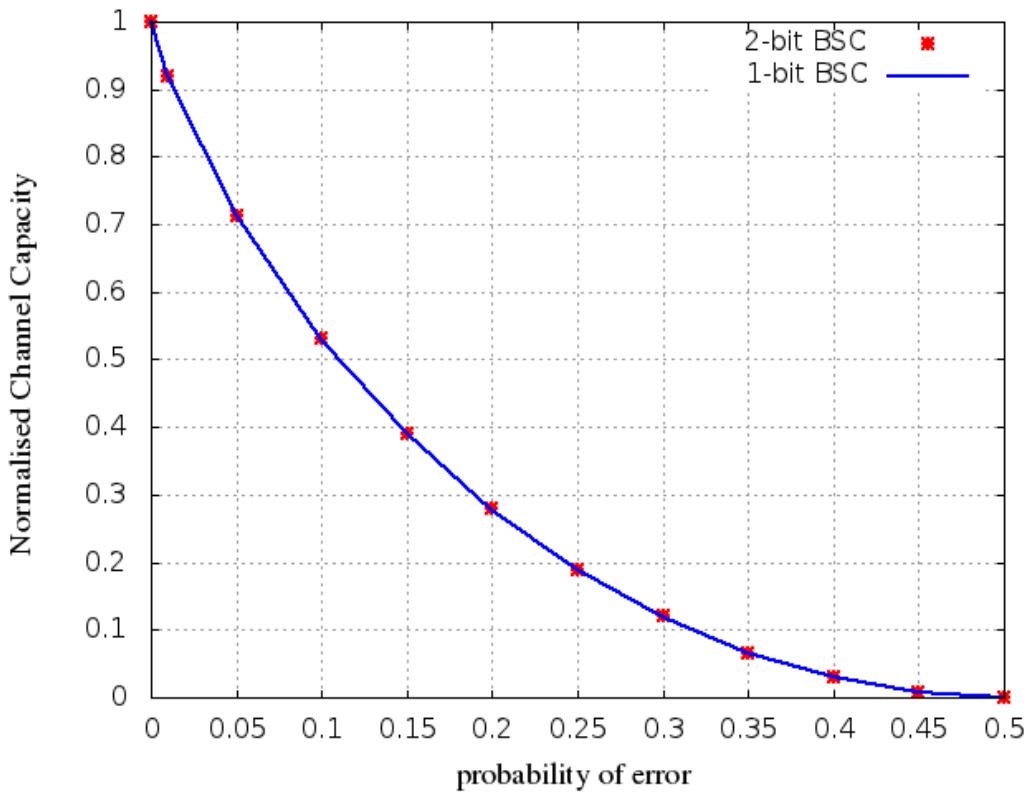
The remaining values can be calculated in the same manner. Table 2.7 show all values of channel capacity and equivocation for the example.

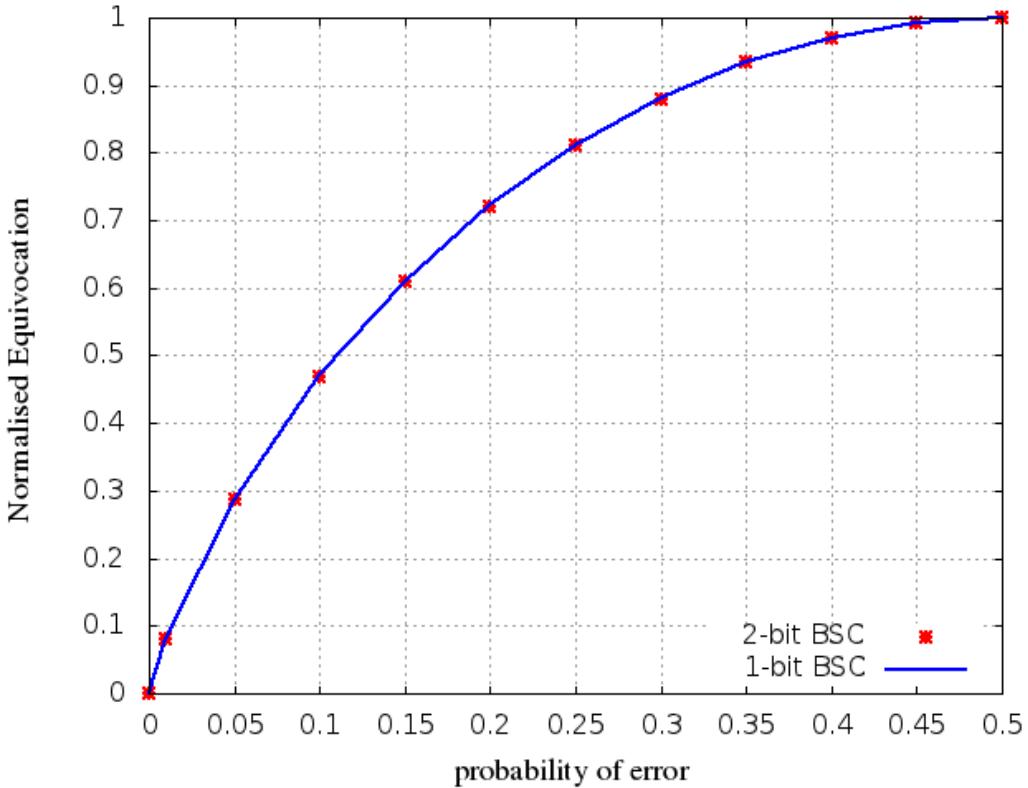
Fig. 2.7 and Fig. 2.8 shows the channel capacity and equivocation as a function of probability of error p_e . Fig. 2.9 and Fig. 2.10 shows the normalised channel capacity and normalised equivocation for 1-bit and 2-bit *BSC* together as a function of probability of error p_e . Clearly, the normalised values of equivocation and channel capacity are identical for 1-bit and 2-bit *BSC*.

p_e	Channel Capacity	Equivocation
0.01	1.8384	0.1616
0.05	1.4272	0.5728
0.10	1.0620	0.9380
0.15	0.7805	1.2195
0.20	0.5566	1.4434
0.25	0.3774	1.6226
0.30	0.2380	1.7620
0.35	0.1320	1.8680
0.40	0.0580	1.9420
0.45	0.0144	1.9856
0.50	0	2

Table 2.7: Channel Capacity and Equivocation values

Figure 2.7: Channel Capacity for 2-bit *BSC*

Figure 2.8: Secrecy Capacity (Equivocation) of 2-bit *BSC*Figure 2.9: Normalised Channel Capacity for 1-bit and 2-bit *BSC*

Figure 2.10: Normalised Equivocation for 1-bit and 2-bit *BSC*

2.4 Wiretap Channel

The wiretap channel proposed by Wyner in 1975 [1] is one of the channels that takes the security of transmitted information into consideration. The wiretap channel contains one sender and two receivers, one of them is the legitimate user and the second is the eavesdropper. The special case of the wiretap channel model is shown in Fig. 2.11.

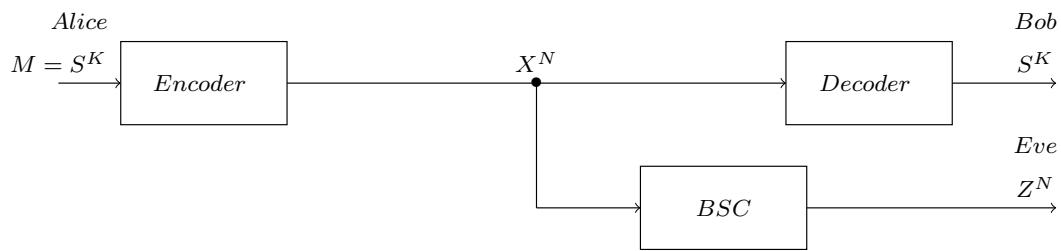


Figure 2.11: wiretap channel Model (special case)

In this model, Alice(transmitter) wants to send a secret message M to Bob(legitimate receiver) in the presence of an eavesdropper,Eve. The model assumes the main channel between Alice and Bob is an error-free channel and the eavesdropper channel is a Binary

Symmetric Channel(*BSC*) with probability of error p_e ($0 < p_e \leq \frac{1}{2}$).

In practice the link between Alice and Bob would be a link with high communication margin such as a line of sight digital microwave with no significant error rate. The eavesdroppers some distance away with a non line of sight link with errors similar to that of a *BSC*.

Assume the message consists random binary K-bits $M = S^K = \{S_1, S_2, \dots, S_K\}$, where $P_r\{S = 0\} = P_r\{S = 1\} = \frac{1}{2}$. Now, Alice encodes S^K as a vector X^N of length N where $X^N = (X_1, \dots, X_N)$ and transmits X^N . Bob and Eve receive the encoding message via their respective channels.

Suppose that Bob has an idea about the encryption system that was used by Alice and is therefore able to retrieve the original message. Assuming that S^K is uniformly distributed, the information rate to the Bob is [1]

$$R = \frac{K}{N} \quad (2.19)$$

The eavesdropper, Eve, receives the encoded message X^N via a *BSC* with probability of error p_e . The corresponding output received by Eve is $Z^N = (Z_1, \dots, Z_N)$. The equivocation(*Eq*) of the eavesdropper is defined as

$$Eq = \frac{H(S^K | Z^N)}{H(S^K)} = \frac{H(S^K | Z^N)}{K} \quad (2.20)$$

The equivocation is a measure of the uncertainty of Eve about the message S^K after observing Z^N .

The main objective of this design by Wyner is to maximise the rate of reliable communication from the transmitter to the legitimate receiver and to minimise the information(i.e. maximise the equivocation) that can be gained by the eavesdropper.

2.5 Syndrome Decoding

The syndrome can be used as an error detection scheme. Syndrome decoding is one of the high efficiency methods in the decoding a linear code in channels that add errors. Let C be a linear $[n, k, d]$ code, and H be a parity check matrix for C . Assume M is the transmitted message, r the received message and e the error vector, then

$$r = M + e$$

When r is received, the decoder calculates the following:

$$s = rH^T$$

s is called the syndrome of r .

Note that $s = rH^T = (M + e)H^T = MH^T + eH^T = eH^T$

i.e., The syndrome depends only on the error. The following procedure shows the steps for syndrome decoding [13, 15]:

1. List the cosets and select coset leaders for the code.
2. Calculates the syndromes for each coset leader and construct the syndrome look up table.
3. Calculate the syndrome of the received messages and from the table determine which coset leader (this is the error e) corresponding to this syndrome.
4. Calculate $M = r - e$.

For the following example, suppose $C = \{00000, 11100, 00111, 11011\}$, a code with parity-check matrix:

$$H = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

The standard array for the code C is shown in Table 2.8 and the syndrome look up table is shown in Table 2.9. If the string $r = 00110$ is received, the syndrome equation is applied as follows: $s = rH^T = (00110) \times H^T = 001$. Using the look up table, we

find that the corresponding coset leader is $e = 00001$. Therefore, we correct the received message as $M = r - e = 00110 - 00001 = 00111$.

Coset leader + C	Coset
C	00000 11100 00111 11011
10000 + C	10000 01100 10111 01011
01000 + C	01000 10100 01111 10011
00100 + C	00100 11000 00011 11111
00010 + C	00010 11110 00101 11001
00001 + C	00001 11101 00110 11010
01001 + C	01001 10101 01110 10010
10001 + C	10001 01101 10110 01010

Table 2.8: The standard array of example

Coset leader(e)	Syndrome(s) = Coset Leader $\times H^T$
00000	000
10000	110
01000	100
00100	010
00010	011
00001	001
01001	101
10001	111

Table 2.9: The Syndrome look up table of example

2.6 Syndrome Coding Scheme for the Wiretap Channel

Traditionally, the syndrome coding scheme uses a $[n, k, 2t + 1]$ code capable of correcting t errors, defined either by a $k \times n$ generator matrix G or by a $(n - k) \times n$ parity check matrix H [19]. The syndrome coding scheme for the wiretap channel introduced by Wyner [1] is shown in Fig. 2.12, We consider a special case of the wiretap channel when the main channel is an error-free channel and the eavesdropper channel is a *BSC* with a probability of error (p_e). The encoding and decoding process of syndrome coding scheme can be described using three algorithms as shown below.

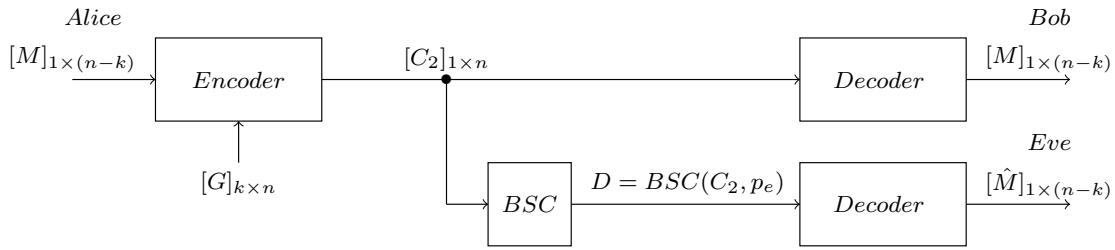


Figure 2.12: The syndrome coding scheme for the wiretap channel

2.6.1 Encoder

The block diagram of the Encoder is shown in Fig. 2.13, and the following algorithm shows how Alice starts the encoding process in order to generates the n -bit transmitted vector C_2 from the $m = (n - k)$ -bit message M .

Algorithm 1 Encoding Algorithm of $[n, k, 2t + 1]$ code

Require:	$[G]_{k \times n}$	▷ The Generator Matrix of $[n, k, 2t + 1]$ code
Require:	$[H]^T_{n \times (n-k)}$	▷ The Parity Check transpose Matrix of $[n, k, 2t + 1]$ code
Require:	$[DR]_{1 \times k}$	▷ Generate random data vector
1: Generate	$[DR]_{1 \times k}$	
2: $[C_1]_{1 \times n} \leftarrow [DR]_{1 \times k} \times [G]_{k \times n}$		
3: $[E]_{1 \times n} \leftarrow [M]_{1 \times (n-k)}$		▷ Generate E from M ¹
4: $[C_2]_{1 \times n} \leftarrow [C_1]_{1 \times n} + [E]_{1 \times n}$		
5: return	$[C_2]_{1 \times n}$	

¹All 2^n error patterns, E , are partitioned into 2^{n-k} cosets. There are 2^k error patterns $\in E$, which have same syndrome.

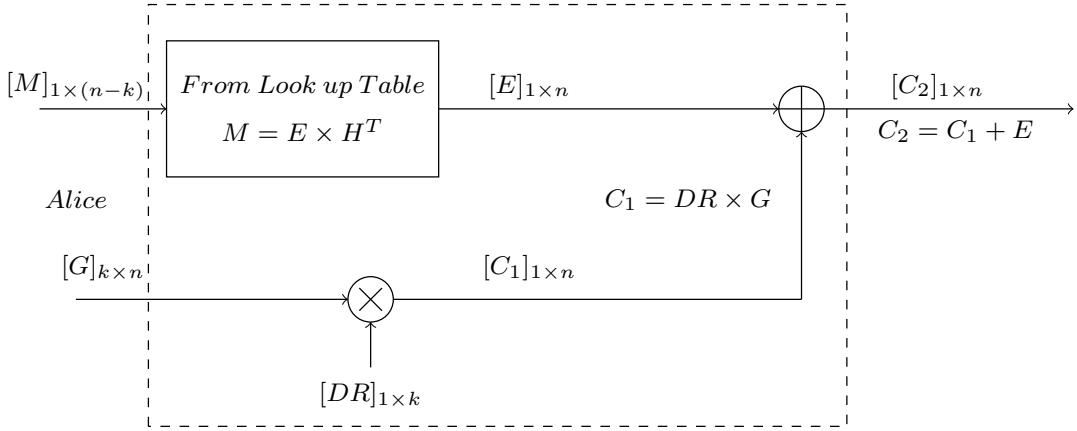


Figure 2.13: Block Diagram of the Encoder

To carry the m -bit message M , Alice looks up the error pattern E corresponding to the syndrome s which is set to be M .

2.6.2 Legitimate Receiver's Decoder

Since the main channel is the error-free channel, Bob receives the transmitted vector C_2 , and he recovers the original message M as shown in Algorithm 2.

Algorithm 2 Decoding Algorithm of $[n, k, 2t + 1]$ code

Require: $[H]_{(n-k) \times n}$ ▷ The Parity Check Matrix of $[n, k, 2t + 1]$ code
Require: $[C_2]_{1 \times n}$ ▷ The transmitted vector from Alice
1: **Generate** $[H]_{n \times (n-k)}^T$ ▷ The Parity Check transpose Matrix
2: $[s]_{1 \times (n-k)} \leftarrow [C_2]_{1 \times n} \times [H]_{n \times (n-k)}^T$
3: $M = s$
4: **return** M

2.6.3 Eavesdropper's Decoder

The block diagram of the *BSC* channel and eavesdropper's decoder is shown in Fig. 2.14.

Eve receives a corrupted vector D instead of the transmitted vector $[C_2]_{1 \times n}$ as a result of passing through the *BSC* which adds additional errors $[E_{BSC}]_{1 \times n}$ as follows:

$[D]_{1 \times n} = [C_2]_{1 \times n} + [E_{BSC}]_{1 \times n}$, Where $[E_{BSC}]_{1 \times n}$ is a random binary error vector which depends on the crossover probability p_e of *BSC*. Assuming that Eve uses the same type of decoder that has been used by Bob, the following steps explain how she gets the estimated message \hat{M} from the corrupted vector D :

Algorithm 3 Eavesdropper's Decoder

Require: $[H]_{(n-k) \times n}$ ▷ The Parity Check Matrix of $[n, k, 2t + 1]$ code
Require: $[D]_{1 \times n}$ ▷ The corrupted vector

- 1: **Generate** $[H]_{n \times (n-k)}^T$ ▷ The Parity Check transpose Matrix
- 2: $[s_c]_{1 \times (n-k)} \leftarrow [D]_{1 \times n} \times [H]_{n \times (n-k)}^T$
- 3: $\hat{M} = s_c = s + s_e = M + s_e, M \neq \hat{M}$ ▷ Eve recovers an estimate of the message \hat{M}
- 4: **return** \hat{M}

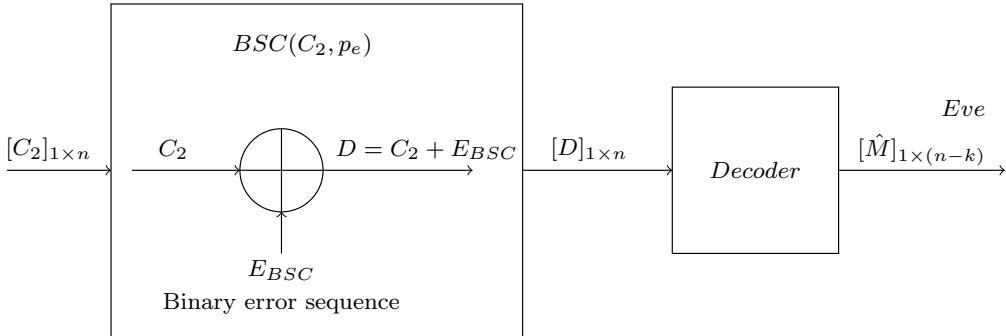


Figure 2.14: Block Diagram of the BSC channel and Decoder of Eavesdropper

2.7 The $(23, 12, 7)$ Binary Golay Code

2.7.1 Perfect Binary Golay Code

The $(23, 12, 7)$ binary Golay code is a perfect linear error-correcting code, that is, the number of correctable error patterns equal to the number of syndromes. It was found by M. J. Golay in 1949, is one of the most important binary quadratic residue(QR) codes. The parameters of Perfect binary Golay code are:

$$\text{Block length}(n)=23 \quad \text{Dimension}(k)=12$$

$$\text{Information Rate}(R)=12/23=0.522 \quad \text{Distance}(d)=7$$

The minimum distance is $d = 7$, therefore the $(23, 12, 7)$ Golay code can be correct all error patterns up to 3 (where $t \leq \lfloor (d - 1)/2 \rfloor$) [20, 21]. The total number of these error patterns are listed in Table 2.10. The total no. of error patterns = $2048 = 2^{11}$ = no. of syndromes. i.e. the Golay code satisfies the condition of the complete code.

$$\sum_{i=0}^t \binom{n}{i} = \binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{t} = 2^{n-k} \quad (2.21)$$

No. of Errors	No. of Syndrome patterns
0	1
1	23
2	253
3	1771
total	2048

Table 2.10: The number of error and syndrome patterns of $(23, 12, 7)$ Golay code

$$\sum_{i=0}^3 \binom{23}{i} = \binom{23}{0} + \binom{23}{1} + \binom{23}{2} + \binom{23}{3} = 2^{11} = 2048$$

According to the property above, the $(23, 12, 7)$ Binary Golay code can be encoded and decoded simply by using look-up tables.

2.7.2 Generator matrix and Parity check matrix of Golay Code

There are two Golay code generator polynomials [20, 22]:

$$g_1(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$$

or

$$g_2(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$$

Both of them $g_1(x)$ and $g_2(x)$ are factors of $x^{23}-1$ in $GF(2)$; in fact, we have:

$$x^{23} - 1 = (x - 1)g_1(x)g_2(x)$$

By using $g(x) = 1 + x + x^5 + x^6 + x^7 + x^9 + x^{11}$ as the generator polynomial of Golay code, the Generator matrix G of the (12×23) elements are:

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Now, we should systematise the generator matrix G by using row operation in order to protect the information bits. Consequently, the Generator matrix G will change into the form $[Q_{12 \times 11}, I_{12 \times 12}]$ that shown as follows:

The parity check polynomial of Golay code is:

$$h(x) = (x^{23} - 1)/g(x) = x^{12} + x^{10} + x^7 + x^4 + x^3 + x^2 + x + 1$$

and the parity check matrix H of the (11×23) elements are:

By using row operation, or directly by transform systematic G of $[Q_{12 \times 11}, I_{12 \times 12}]$ into $[I_{11 \times 11}, Q_{11 \times 12}^T]$. The systematic parity check matrix is shown as follows:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

2.7.3 Weight distribution of Golay Code

To verify that the minimum distance of Golay code is 7, the weight distribution needs to be calculated. Table 2.11 shows the weight distribution of the (23, 12, 7) Golay code has been found by Magma Software [4].

```
> C := BKLC(GF(2), 23, 12);
> WeightDistribution(C);
[ <0, 1>, <7, 253>, <8, 506>, <11, 1288>, <12, 1288>,
  <15, 506>, <16, 253>, <23, 1> ]
```

The total number of codewords is 2^{12} and the minimum Hamming distance of this code is indeed 7, as shown by Table 2.11.

Codeword Weight	No. of Codeword
0	1
1	0
2	0
3	0
4	0
5	0
6	0
7	253
8	506
9	0
10	0
11	1288
12	1288
13	0
14	0
15	506
16	253
17	0
18	0
19	0
20	0
21	0
22	0
23	1
The total No. of Codewords	4096

Table 2.11: The Weight distribution of (23, 12, 7) Golay Code

2.7.4 The Codewords and Algebraic Decoder of Golay Code

The codewords of Golay code over GF(2) can be expressed as [21, 23] :

$$c(x) = \sum_{i=0}^{22} c_i x^i \quad (2.22)$$

$$\text{Assume } i(x) = \sum_{i=0}^{11} c_i x^i \quad \text{and} \quad p(x) = \sum_{i=12}^{22} c_i x^i$$

are the information and parity-check polynomials of $c(x)$.

Let $e(x) = \sum_{i=0}^{22} e_i x^i$ be an error polynomial. Then, the received word can be represented by:

$$r(x) = \sum_{i=0}^{22} r_i x^i = c(x) + e(x) \quad (2.23)$$

Assume α is a primitive 23rd root of unity in GF(2¹¹). Now, we need to find an element of order 23.

Since $2^{11}-1 = 2047 = 23 \times 89$. Therefore; the element $\alpha^{89} \in \text{GF}(2^{11})$ has order 23.

The cyclotomic coset with respect to 23 can be calculated by assuming Q_{23} to be the set of quadratic residues modulo 23:

$$Q_{23} = j^2 \pmod{23}; (j = 1, 2, \dots, (23-1)/2) \quad (2.24)$$

This gives $Q_{23} = \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\}$.

Note that $g(\alpha^j) = 0$ for all $j \in Q_{23}$, the syndromes S_j can be obtained by:

$$S_j = r(\alpha^j) = c(\alpha^j) + e(\alpha^j) = e(\alpha^j) \quad (2.25)$$

As the Golay code has a cyclotomic set $Q_{23} = \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\}$. Consequently, $g(x)$ has three roots ($\alpha^1, \alpha^3, \alpha^9$). Hence the three syndromes (S_1, S_3, S_9) can be calculated by: $S_1 = r(\alpha^1)$, $S_3 = r(\alpha^3)$, and $S_9 = r(\alpha^9)$. The process of finding the errors and its locations has been explained by Elia [24]. To illustrate this method, the

error-locator polynomial $L(z)$ is defined as

$$L(z) = \prod_{i=1}^3 (z - z_i) = z^3 + \sum_{j=1}^3 \sigma_j z^{3-j} \quad (2.26)$$

Where $\sigma_1 = z_1 + z_2 + z_3$, $\sigma_2 = z_1 z_2 + z_2 z_3 + z_3 z_1$ and $\sigma_3 = z_1 z_2 z_3$.

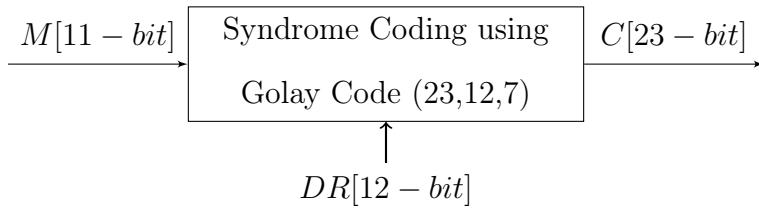
Here, z represents the root of error locator polynomial, then $z = \alpha^j$, where j locates the position of error to be corrected. The all error cases that have been obtained is summarised as follows:

- 1) If no error occurs, then $S_1 = S_3 = S_9 = 0$ and $L(z) = 0$.
- 2) If one error occurs, then $S_1^3 = S_3$, $S_3^3 = S_9$ and $L(z) = z + S_1$.
- 3) If two error occurs, then $L(z) = z^2 + S_1 z + (S_1^2 + \frac{S_3}{S_1})$, if $S_1 D^{1/3} = S_3$.
- 4) If three error occurs, then $L(z) = z^3 + S_1 z^2 + (S_1^2 + D^{1/3})z + (S_3 + S_1 D^{1/3})$, if $S_1 D^{1/3} \neq S_3$.

Where $D = S_1^6 + S_3^2 + (S_1^9 + S_9)/(S_1^3 + S_3)$, and the cube root $D^{1/3}$ is in $\text{GF}(2^{11})$, and it can be calculated as a power of exponent 1365, i.e., $D^{1/3} = D^{1365}$. Finally, the error locations can be found from $L(z)$.

2.7.5 The Encoding and Decoding Algorithms of Golay Code

Encoding Algorithm



To illustrate the procedure of encrypting the message $[M]_{1 \times 11}$ into cipher text vector $[C]_{1 \times 23}$, the following four steps must be achieved :

Step 1 : Select the Generator Matrix $[G]_{12 \times 23}$ of Golay Code.

Step 2 : Generate [12-bit] random data vector $[DR]_{1 \times 12}$ randomly.

Step 3 : Calculate

$$[C_1]_{1 \times 23} = [DR]_{1 \times 12} \times [G]_{12 \times 23} \quad (2.27)$$

Step 4 : Calculate

$$[C]_{1 \times 23} = [C_1]_{1 \times 23} + [E_1]_{1 \times 23} \quad (2.28)$$

Where E_1 is a [23-bit] error pattern of weight between 0 and 3. The error vector E_1 is related to the Message $[M]_{1 \times 11}$. So, for each error pattern a syndrome must exist because the Golay code is a perfect code. Therefore; the error vector can be calculated by setting $S_i = M_i$.

E_1 can be calculated from

$$S = C \times H^T = M \quad (2.29)$$

Where H^T is the transpose matrix of H

$$M = C \times H^T = [C_1 + E_1] \times H^T$$

$$M = [DR \times G + E_1] \times H^T = DR \times G \times H^T + E_1 \times H^T$$

Since G and H are orthogonal ($G \times H^T = 0$)

$$M - E_1 \times H^T = 0 \quad (2.30)$$

From the last equation, M and H^T are known. Therefore; the error vector E_1 can be computed.

Decoding Algorithm

Assuming that the main channel is error-free channel, the legitimate receiver receives the cipher text C . Now, to recover the original message, the legitimate receiver calculates the syndrome of C for Golay code $S = C \times H^T$, which equates to the message M .

Worked Example

The generator matrix $[G]_{12 \times 23}$ and parity check matrix $[H]_{11 \times 23}$ of the $(23, 12, 7)$ Golay code can be obtained directly by using Magma software as shown below:

```
> C:=BKLC(GF(2),23,12);
> C;
[23, 12, 7] "Quadratic Residue code" Linear Code over GF(2)
```

Generator matrix:

```
[1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 1 0]
[0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 1]
[0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 0 1 0 0]
[0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 0 1 0]
[0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 0 1]
[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 0 1]
[0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 1 0 0 1 1 0 0]
[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 1 1 0 0 1 0]
[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 1 0 1 0 0 1 1]
[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 1 1 0 0 0 1]
[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 1 1 0 0 0 1]
[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 1 1 0 0 0 1 1]
```

```
> ParityCheckMatrix(C);
```

```
[1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 1 1 1]
[0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 0 1 0 0 0]
[0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 0 1 0 0]
[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 0 1 0]
[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 0 1 0]
[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 0 1]
[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 1 1 0 0]
[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 0 0 1 1]
[0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 1 1 0 0 0 1 0 0 1 1]
[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1 0 1 0 1 1 1 0 0]
[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1 1 1 0 0 0 1 1]
[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0]
```

Assuming that Alice wishes to send the message $[M]_{1 \times 11} = [01111000011]$ to Bob, she first generates a random data vector, say $[DR]_{1 \times 12} = [000110101000]$ and computes:

$$[C_1]_{1 \times 23} = [DR]_{1 \times 12} \times [G]_{12 \times 23} = [00011010100011110110010]$$

Now we illustrate how to calculate the error pattern $[E_1]_{1 \times 23}$ of weight between 0 and 3. The error vector E_1 is related to the message $[M]_{1 \times 11}$. So the mapping between each error pattern and syndromes is one-to-one. The error vector can be calculated by setting $S_i = M_i (i = 0, 1, \dots, 2047)$

$$S_1 = E_1 \times H^T = M$$

where H^T represent the transpose of parity check matrix. As there are only 2048 syndromes, it is straightforward to generate a look up table linking S_1 to E_1 .

$$[E_1]_{1 \times 23} = [00000000010001000000001]$$

$$[C]_{1 \times 23} = [C_1]_{1 \times 23} + [E_1]_{1 \times 23}$$

$$[C]_{1 \times 23} = [00011010100011110110010] + [00000000010001000000001]$$

$$[C]_{1 \times 23} = [00011010110010110110011]$$

which she then sends to Bob.

Upon receiving $[C]_{1 \times 23}$, Bob decodes C by calculating the syndrome:

$$S = C \times H^T = [01111000011]$$

which equals the original message M .

$$H^T = \left[\begin{array}{cccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{array} \right]$$

3) P.16 , Modify Table 1.7

2.8 Best Known Linear Code (33, 23, 5)

2.8.1 Construction of Binary Linear Codes of minimum distance five

There are many techniques for constructing binary linear codes of minimum distance 5 as shown in [16, 25, 26]. Any codeword of length n , number of information bits k and minimum distance d is called an $[n, k, d]$ linear code, which can be defined by a $(n-k) \times n$ parity check matrix H . In one of these methods, as shown in [27], two binary matrices B and Q were assumed with $(n - k)$ rows and n_1, n_2 columns, respectively. The following parity check matrix is used to define a binary code C of length $n = n_1 + n_2$ with $(n - k)$ check bits as follows:

$$H = \begin{bmatrix} 000 & \cdots & 0 & 111 & \cdots & 1 \\ & B & & & Q & \end{bmatrix}$$

In this method, two matrices B and Q should be constructed. In addition, the minimum distance d of C is 5 if all combinations of any 4 columns of H are linearly independent. Assume α be a primitive element in $F = GF(2^m)$. The element of F has been constructed as a binary m -tuple using $(1, \alpha, \alpha^2, \dots, \alpha^{m-1})$ as a basis of F [16, 25, 26]. The B matrix was constructed with $2^m - 1$ columns as follows:

$$B = \begin{bmatrix} 1 & \alpha & \cdots & \alpha^i & \cdots & \alpha^{(2^m-2)} \\ 1 & \alpha^3 & \cdots & \alpha^{3i} & \cdots & \alpha^{3(2^m-2)} \end{bmatrix}$$

In order to calculate Q , assume $L(x)$ to be a linear map from F to F . So, the $i'th$ column q_i of Q has been constructed as follow:

$$q_i = \begin{bmatrix} y_i \\ y_i^3 + L(y_i) \end{bmatrix}$$

where $y_i (i = 1, 2, \dots, n_2)$ are distinct elements of F .

In particular, several (33, 23, 5) codes are constructed by using another approach, these

codes have one more information bit than the best known code $(32, 22, 5)$ [28]. One of these codes is described in section 2.8.2.

2.8.2 Construction of Parity Check Matrix $H[10, 33]$ of $(33, 23, 5)$ code

Generates a field with 1024 elements

Assume the irreducible polynomial $f = 1 + x^3 + x^{10}$ and α be a primitive root of the polynomial. Therefore, $\alpha^{10} = 1 + \alpha^3$.

Thus, the following Table 2.12 of field $GF(2^{10})$ is obtained. This table shows the vectors of coefficients with respect to the basis $(1, \alpha, \alpha^2, \dots, \alpha^9)$.

α^9	α^8	α^7	α^6	α^5	α^4	α^3	α^2	α^1	α^0	Polynomial	α
0	0	0	0	0	0	0	0	0	0	0	-
0	0	0	0	0	0	0	0	0	1	1	α^0
0	0	0	0	0	0	0	0	1	0	x	α
0	0	0	0	0	0	0	1	0	0	x^2	α^2
0	0	0	0	0	0	0	1	0	0	x^3	α^3
0	0	0	0	0	0	1	0	0	0	x^4	α^4
0	0	0	0	0	1	0	0	0	0	x^5	α^5
0	0	0	1	0	0	0	0	0	0	x^6	α^6
0	0	1	0	0	0	0	0	0	0	x^7	α^7
0	1	0	0	0	0	0	0	0	0	x^8	α^8
1	0	0	0	0	0	0	0	0	0	x^9	α^9
0	0	0	0	0	0	1	0	0	1	x^3+1	α^{10}
0	0	0	0	0	1	0	0	1	0	x^4+x	α^{11}
0	0	0	0	1	0	0	1	0	0	x^5+x^2	α^{12}
0	0	0	1	0	0	1	0	0	0	x^6+x^3	α^{13}
0	0	1	0	0	1	0	0	0	0	x^7+x^4	α^{14}
0	1	0	0	1	0	0	0	0	0	x^8+x^5	α^{15}
1	0	0	1	0	0	0	0	0	0	x^9+x^6	α^{16}
0	0	1	0	0	0	1	0	0	1	$x^{10}+x^7=x^7+x^3+1$	α^{17}
0	1	0	0	0	1	0	0	1	0	x^8+x^4+x	α^{18}
1	0	0	0	1	0	0	1	0	0	$x^9+x^5+x^2$	α^{19}
0	0	0	1	0	0	0	0	0	1	$x^{10}+x^6+x^3=x^6+x^3+1=x^6+1$	α^{20}
0	0	1	0	0	0	0	0	1	0	x^7+x	α^{21}
0	1	0	0	0	0	0	1	0	0	x^8+x^2	α^{22}
1	0	0	0	0	0	1	0	0	0	x^9+x^3	α^{23}
0	0	0	0	0	1	1	0	0	1	$x^{10}+x^4=x^4+x^3+1$	α^{24}
0	0	0	0	1	1	0	0	1	0	x^5+x^4+x	α^{25}
0	0	0	1	1	0	0	1	0	0	$x^6+x^5+x^2$	α^{26}
0	0	1	1	0	0	1	0	0	0	$x^7+x^6+x^3$	α^{27}
0	1	1	0	0	1	0	0	0	0	$x^8+x^7+x^4$	α^{28}
1	1	0	0	1	0	0	0	0	0	$x^9+x^8+x^5$	α^{29}
1	0	0	1	0	0	1	0	0	1	$x^{10}+x^9+x^6=x^9+x^6+x^3+1$	α^{30}
.
.
0	0	0	1	1	0	0	1	1	1	x^9+x^2	α^{1022}

Table 2.12: Galois Field of GF(2^{10}), primitive polynomial = $x^{10}+x^3+1$

Construction $H[10, 33]$ for $(33, 23, 5)$ code

H is divided into two partitions, the first one contains 22 columns and the other has 11 columns. Assume $\beta = \alpha^{31}$ and let B the matrix that contains the 22 elements β^i , $i = 1, 2, 4, 8, 16, 32, 31, 29, 25, 17, 3, 6, 12, 24, 15, 30, 27, 21, 9, 18, 11$, and 22. The values of β^i are calculated as shown below [27]:

$$\beta = \alpha^{31}$$

$$\alpha^{10} = 1 + \alpha^3$$

$$\alpha^{20} = 1 + \alpha^6$$

$$\alpha^{30} = \alpha^{10} \times \alpha^{20} = (1 + \alpha^3) \times (1 + \alpha^6) = 1 + \alpha^3 + \alpha^6 + \alpha^9$$

$$\alpha^{31} = \alpha \times (1 + \alpha^3 + \alpha^6 + \alpha^9) = \alpha + \alpha^4 + \alpha^7 + \alpha^{10} = 1 + \alpha + \alpha^3 + \alpha^4 + \alpha^7 = \beta$$

$$\beta = 1 + \alpha + \alpha^3 + \alpha^4 + \alpha^7$$

$$\beta^2 = \beta \times \beta = 1 + \alpha^2 + \alpha^4 + \alpha^6 + \alpha^7 + \alpha^8$$

$$\beta^4 = \beta^2 \times \beta^2 = 1 + \alpha^2 + \alpha^5 + \alpha^6 + \alpha^7 + \alpha^8 + \alpha^9$$

$$\beta^8 = \beta^4 \times \beta^4 = \alpha + \alpha^2 + \alpha^3 + \alpha^4 + \alpha^5 + \alpha^6 + \alpha^7 + \alpha^8 + \alpha^9$$

$$\beta^{16} = \beta^8 \times \beta^8 = 1 + \alpha + \alpha^3 + \alpha^4 + \alpha^5 + \alpha^7 + \alpha^9$$

$$\beta^{32} = \beta^{16} \times \beta^{16} = \alpha + \alpha^2 + \alpha^3 + \alpha^6 + \alpha^7$$

$$\beta^{31} = \beta \times \beta^2 \times \beta^4 \times \beta^8 \times \beta^{16} = \alpha^5 + \alpha^6 + \alpha^7$$

$$\beta^{29} = \beta \times \beta^4 \times \beta^8 \times \beta^{16} = 1 + \alpha^2 + \alpha^3 + \alpha^4 + \alpha^5 + \alpha^7$$

$$\beta^{25} = \beta \times \beta^8 \times \beta^{16} = \alpha^3 + \alpha^6 + \alpha^7 + \alpha^8$$

$$\beta^{17} = \beta \times \beta^{16} = \alpha^2 + \alpha^4 + \alpha^5 + \alpha^7 + \alpha^9$$

$$\beta^3 = \beta \times \beta^2 = 1 + \alpha^3 + \alpha^6 + \alpha^7 + \alpha^9$$

$$\beta^6 = \beta^2 \times \beta^4 = 1 + \alpha + \alpha^2 + \alpha^5 + \alpha^6 + \alpha^7 + \alpha^8$$

$$\beta^{12} = \beta \times \beta^{11} = \alpha^3 + \alpha^5 + \alpha^6 + \alpha^7 + \alpha^9$$

$$\beta^{24} = \beta^8 \times \beta^{16} = 1 + \alpha + \alpha^2 + \alpha^3 + \alpha^5 + \alpha^6 + \alpha^7 + \alpha^8$$

$$\beta^{15} = \beta^3 \times \beta^{12} = \alpha^3 + \alpha^5 + \alpha^7 + \alpha^9$$

$$\beta^{30} = \beta^{15} \times \beta^{15} = 1 + \alpha + \alpha^3 + \alpha^6 + \alpha^7 + \alpha^8$$

$$\beta^{27} = \beta^3 \times \beta^{24} = 1 + \alpha^4 + \alpha^5 + \alpha^7 + \alpha^9$$

$$\beta^{21} = \beta^6 \times \beta^{15} = \alpha + \alpha^3 + \alpha^7$$

$$\beta^9 = \beta^3 \times \beta^6 = \alpha^2 + \alpha^4 + \alpha^6 + \alpha^7$$

$$\beta^{18} = \beta^9 \times \beta^9 = \alpha^2 + \alpha^5 + \alpha^7 + \alpha^8$$

$$\beta^{11} = \beta^2 \times \beta^9 = 1 + \alpha^2 + \alpha^3 + \alpha^5 + \alpha^6 + \alpha^7$$

$$\beta^{22} = \beta^{11} \times \beta^{11} = \alpha^2 + \alpha^3 + \alpha^5 + \alpha^6 + \alpha^7$$

The next step is constructing the remaining 11 columns of B in order to get H for $(33, 23, 5)$. These columns can be calculated through the computer search as a set of α^j , $j=1, 16, 106, 195, 281, 460, 609, 786, 891, 941$, and 979.

The values of α^j is calculated as shown below [27]:

$$\alpha^{10} = 1 + \alpha^3; \alpha^{20} = 1 + \alpha^6; \alpha^{30} = 1 + \alpha^3 + \alpha^6 + \alpha^9$$

$$\alpha^{16} = \alpha^6 \times \alpha^{10} = \alpha^6 + \alpha^9$$

$$\alpha^{25} = \alpha^5 \times \alpha^{20} = \alpha + \alpha^4 + \alpha^5; \alpha^{50} = \alpha^{25} \times \alpha^{25} = 1 + \alpha^2 + \alpha^3 + \alpha^8$$

$$\alpha^{100} = \alpha^{50} \times \alpha^{50} = 1 + \alpha^4 + \alpha^9$$

$$\alpha^{106} = \alpha^6 \times \alpha^{100} = 1 + \alpha^3 + \alpha^5 + \alpha^6 + \alpha^8$$

$$\alpha^{24} = 1 + \alpha^3 + \alpha^4; \alpha^{48} = 1 + \alpha^6 + \alpha^8; \alpha^{96} = 1 + \alpha^2 + \alpha^5 + \alpha^6 + \alpha^9$$

$$\alpha^{192} = \alpha + \alpha^2 + \alpha^3 + \alpha^5 + \alpha^8$$

$$\alpha^{195} = \alpha^3 \times \alpha^{192} = \alpha + \alpha^5 + \alpha^6 + \alpha^8$$

$$\alpha^{34} = 1 + \alpha^4 + \alpha^6 + \alpha^7; \alpha^{68} = 1 + \alpha^2 + \alpha^4 + \alpha^5 + \alpha^7 + \alpha^8$$

$$\alpha^{136} = \alpha^3 + \alpha^6 + \alpha^7 + \alpha^8 + \alpha^9; \alpha^{272} = \alpha + \alpha^2 + \alpha^5 + \alpha^7 + \alpha^8 + \alpha^9$$

$$\alpha^{281} = \alpha^9 \times \alpha^{272} = \alpha^4 + \alpha^6 + \alpha^8 + \alpha^9$$

$$\alpha^{28} = \alpha^4 + \alpha^7 + \alpha^8; \alpha^{56} = \alpha^4 + \alpha^6 + \alpha^7 + \alpha^8 + \alpha^9$$

$$\alpha^{112} = \alpha + \alpha^2 + \alpha^5 + \alpha^6 + \alpha^7 + \alpha^9; \alpha^{224} = 1 + \alpha + \alpha^3 + \alpha^4 + \alpha^5 + \alpha^7 + \alpha^8$$

$$\alpha^{448} = \alpha^2 + \alpha^3 + \alpha^4 + \alpha^7 + \alpha^8 + \alpha^9$$

$$\alpha^{460} = \alpha^{12} \times \alpha^{448} = 1 + \alpha + \alpha^2 + \alpha^4 + \alpha^8$$

$$\alpha^{19} = \alpha^2 + \alpha^5 + \alpha^9; \alpha^{38} = 1 + \alpha + \alpha^3 + \alpha^8; \alpha^{76} = 1 + \alpha^2 + \alpha^9$$

$$\alpha^{152} = 1 + \alpha + \alpha^8; \alpha^{304} = 1 + \alpha^2 + \alpha^6 + \alpha^9; \alpha^{608} = 1 + \alpha^2 + \alpha^4 + \alpha^5 + \alpha^8$$

$$\alpha^{609} = \alpha \times \alpha^{608} = \alpha + \alpha^2 + \alpha^3 + \alpha^6 + \alpha^9$$

$$\alpha^{192} = \alpha + \alpha^2 + \alpha^3 + \alpha^5 + \alpha^8; \alpha^{384} = 1 + \alpha^2 + \alpha^3 + \alpha^4 + \alpha^9; \alpha^{768} = 1 + \alpha + \alpha^6$$

$$\alpha^{786} = \alpha^{18} \times \alpha^{768} = 1 + \alpha + \alpha^2 + \alpha^3 + \alpha^5 + \alpha^8 + \alpha^9$$

$$\alpha^{27} = \alpha^3 + \alpha^6 + \alpha^7; \alpha^{54} = \alpha^2 + \alpha^4 + \alpha^5 + \alpha^6 + \alpha^7; \alpha^{108} = 1 + \alpha^2 + \alpha^3 + \alpha^5 + \alpha^7 + \alpha^8$$

$$\alpha^{216} = \alpha^3 + \alpha^7 + \alpha^9; \alpha^{432} = \alpha + \alpha^6 + \alpha^7 + \alpha^8; \alpha^{864} = \alpha^4 + \alpha^5 + \alpha^6 + \alpha^7 + \alpha^9$$

$$\alpha^{891} = \alpha^{27} \times \alpha^{864} = \alpha^2 + \alpha^4 + \alpha^6$$

$$\alpha^{29} = \alpha^5 + \alpha^8 + \alpha^9 ; \alpha^{58} = 1 + \alpha + \alpha^3 + \alpha^4 + \alpha^6 + \alpha^8 + \alpha^9$$

$$\alpha^{116} = 1 + \alpha + \alpha^4 + \alpha^5 + \alpha^9 ; \alpha^{232} = \alpha + \alpha^2 + \alpha^3 + \alpha^4$$

$$\alpha^{464} = \alpha^2 + \alpha^4 + \alpha^6 + \alpha^8 ; \alpha^{928} = \alpha^2 + \alpha^4 + \alpha^5 + \alpha^6 + \alpha^8 + \alpha^9$$

$$\alpha^{941} = \alpha^{13} \times \alpha^{928} = 1 + \alpha^3 + \alpha^4 + \alpha^8 + \alpha^9$$

$$\alpha^{30} = 1 + \alpha^3 + \alpha^6 + \alpha^9 ; \alpha^{60} = 1 + \alpha + \alpha^2 + \alpha^4 + \alpha^5 + \alpha^6 + \alpha^8$$

$$\alpha^{120} = \alpha^3 + \alpha^4 + \alpha^5 + \alpha^6 + \alpha^8 + \alpha^9 ; \alpha^{240} = 1 + \alpha + \alpha^2 + \alpha^3 + \alpha^4 + \alpha^5 + \alpha^9$$

$$\alpha^{480} = \alpha + \alpha^2 + \alpha^3 + \alpha^6 ; \alpha^{960} = \alpha^4 + \alpha^5 + \alpha^6$$

$$\alpha^{979} = \alpha^{19} \times \alpha^{960} = 1 + \alpha + \alpha^5 + \alpha^9$$

Now, the $H[10, 33]$ for $(33, 23, 5)$ code is constructed as shown below:

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

2.8.3 Generation of Parity Check Matrix using Magma Software

The parity check matrix H of the $(33, 23, 5)$ code has been obtained directly by using Magma software as shown below:

```
> C := BKLC(GF(2), 33, 23);
> C;
[33, 23, 5] Linear Code over GF(2)
Generator matrix:
```



```
> ParityCheckMatrix (C);

[1 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 1 1 1 1 0 1 0 1 0 1 0 1 1 0 1 1]

[0 1 0 0 0 0 0 0 0 1 1 1 1 0 1 0 0 1 1 0 1 0 0 0 1 0 0 1 1 1 0 1]

[0 0 1 0 0 0 0 0 0 1 1 1 1 1 0 1 0 0 1 1 0 1 1 1 1 0 0 1 1 0 0]

[0 0 0 1 0 0 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 1 0 0 1 0 1 1 1 1]

[0 0 0 0 1 0 0 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 1 0 0 1 0 1 1 1]

[0 0 0 0 0 1 0 0 0 0 0 1 1 1 1 1 0 1 0 0 1 1 0 0 0 1 0 1 1 0 0]

[0 0 0 0 0 1 0 0 0 0 0 1 1 1 1 1 1 0 1 0 1 0 0 1 1 1 0 1 1 0 1]

[0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 1 1 1 0 1 0 1 0 0 1 1 1 1 0 0 1 0]

[0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1 1 0 1 0 1 0 0 1 1 1 1 0 1 1]

[0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 0 1]
```

The Weight Distribution of the (33,23,5) Code.

To verify that the minimum distance of this code is 5, the weight distribution of the code needs to be found. Table 2.13 shows the weight distribution of the (33,23,5) code that has been obtained by Magma Software:

```
> C := BKLC(GF(2),33,23);

> WeightDistribution(C);

[<0,1>, <5,275>, <6,1287>, <7,4037>, <8,13090>, <9,37840>,
 <10,90937>, <11,189027>, <12,346247>, <13,559350>, <14,799590>,
 <15,1013298>, <16,1139325>, <17,1139325>, <18,1013298>, <19,799590>,
 <20,559350>, <21,346247>, <22,189027>, <23,90937>, <24,37840>,
 <25,13090>, <26,4037>, <27,1287>, <28,275>, <33,1> ]
```

The total number of codewords is 2^{23} and the minimum Hamming distance of this code is indeed 5, as shown by Table 2.13.

Codeword Weight	No. of Codeword
0	1
1	0
2	0
3	0
4	0
5	275
6	1287
7	4037
8	13090
9	37840
10	90937
11	189027
12	346247
13	559350
14	799590
15	1013298
16	1139325
17	1139325
18	1013298
19	799590
20	559350
21	346247
22	189027
23	90937
24	37840
25	13090
26	4037
27	1287
28	275
29	0
30	0
31	0
32	0
33	1
The total No. of Codewords	
8,388,608	

Table 2.13: The number of codewords of weight $(n-w)$ is equal to number of codeword of weight w

2.9 Literature Review

The wiretap channel has been investigated by several researcher. In [29] Leung showed that the secrecy capacity is equal to the difference between the capacities of the main channel and eavesdropper channel for symmetric wiretap channels. In [30], Leung and Hellman extends Wyner's results for discrete memoryless wiretap channels to the Gaussian wiretap channel.

Wyner's results have been generalised by Csiszar and Korner [31], who considered a discrete memoryless channels with a common input and two receivers. In this model, the main results have been obtained as the set of triples (R_1, R_e, R_o) , such that the secret message is transmitted to receiver1 (legitimate receiver) at rate R_1 and common messages to both receivers at rate R_o , while maximising the equivocation rate R_e of receiver2 (the eavesdropper channel) as much as possible. Further, if the eavesdropper channel is a degraded version of that used by a legitimate receiver and no common message is sent, Wyner's results will be generalised. Another generalisation of Wyner's coding scheme [1] has been done by Cohen and Zemor [32], which assumes that the main and eavesdropper channels are noisy. In addition, they proved that the generalised scheme achieves the Shannon capacity of the system.

Studies of the syndrome coding scheme, whose basic idea is to transmit information in the syndromes of a code so as to increase the communication security has been done by several researchers. For example, Cohen and Zemor [33] analysed the information leakage of syndrome coding for the wiretap channel and proposed a method to select a syndrome function in order to minimise both the length of the transmitted vector and the information leakage to the eavesdropper. Rouayheb and Soljanin [34] analysed the problem of securing a multicast network against an eavesdropper who has the ability to observe a limited number of network links of his choice. They showed that the problem can be solved by generalisation of the wiretap channel type II (which was analysed by Ozarow and Wyner [35]). Also, they proved that network security can be achieved by using syndrome coding as an additional layer to a network code.

Thangaraj *et al.* [36] focused on the problem of developing coding schemes for wiretap channels in order to increase communication security by providing an alternative view of the proof for the secrecy capacity of wiretap channels. Several cases of main and eavesdropper channels have been analysed, for the case where the main channel is noiseless and the eavesdropper channel is *BSC*, they showed that codes with good error detecting properties provide security. Also, they presented code designs for codes achieving secrecy on the wiretap channel when the main channel is noiseless and the eavesdropper channel is a binary erasure channel(*BEC*). In addition, they showed that the secrecy can be achieved by constructing linear-time decodable secrecy codes based on low-density parity-check(*LDPC*) codes.

In [37], Reddy *et al.* presented the video coding system with syndrome coding by using low-density parity-check(*LDPC*) codes.

Dai *et al.* [38] studied a new wiretap combination model which includes Wyner's wiretap channel and wiretap channel of type II by assuming that an eavesdropper can not only catch the main channel output via an eavesdropper channel, but also get some transmitted information from the encoder, Alice. They provided the reliable transmission rate R of the main channel, the equivocation rate of eavesdropper and the ratio α of the leaked transmitted symbols of the new model, and they found that the perfect secrecy can be achieved when $0 \leq \alpha \leq 1 - R$.

Liang *et al.* [39] studied the compound wiretap channel, which can be interpreted as the multicast channel with multiple eavesdroppers. They have obtained the lower and upper bounds on the secrecy capacity for the general compound wiretap channel. The main objective of this design was to generalise Wyner's wiretap channel to permit both the legitimate and the eavesdropper channels to take a number of possible states, ensuring that the transmitter sends the information to all legitimate receivers and protects it from all eavesdroppers.

Chen and Vinck [40] also investigated the binary symmetric wiretap channel and they showed that the secrecy capacity can be obtained by using random linear codes with

syndrome coding. Suresh *et al.* [41] showed that duals of *LDPC* codes have achieved strong secrecy on the binary erasure wiretap channel when used in a syndrome coding scheme.

Recently, researchers have directed their studies on how to increase the equivocation rate of the eavesdropper in the wiretap channel to obtain perfect secrecy. Bafghi *et al.* [42] introduced a new technique of Gaussian wiretap channel with side information, this technique is based on sending the information to the legitimate receiver and eavesdropper through the different channels with different channel states. They have assumed that the state of the main channel is known for the transmitter and uses this knowledge to randomize its information and to disable the eavesdropper in decoding. This leads to an increase the security. In addition, the equivocation rate of the eavesdropper will be increased due to the complexity of extracting the main channel's state for the eavesdropper.

For the traditional wiretap channel, Zhang *et al.* [43] analysed the equivocation rate of the McEliece cryptosystem under full-code attack and Brickell's attack and showed that there is a significant amount of information leaked to the eavesdropper. Therefore, Zhang *et al.* proposed a modified system based on the McEliece system to reduce the information leakage under Brickell's attack, which employs two encoding stages to improve the communication security. The original message is first processed using a syndrome coding scheme based on the (23,12,7) Golay code and then by the traditional McEliece cryptosystem with the following parameters(1024,524,101). The results obtained showed that the eavesdropper cannot recover any useful information under the Brickell's attack but the price for the higher security is a lower information rate. The negative aspects of this design have not succeeded with full-code attack. The results showed that the equivocation rate is zero under full-code attack, meaning that the eavesdropper can recover the original information under this attack.

The security performance of (n, k) random linear binary codes in syndrome coding scheme has been analysed by Zhang *et al.* [44]. The average equivocation of random codes has been calculated by proposed theoretical analysis based on a putative (n, k) code having

the same distribution of syndrome probabilities as the ensemble of all (n, k) random codes. Also, Zhang *et al.* showed that a higher value of average equivocation has been obtained for syndrome coding using a randomly chosen code of length greater than 150 bits. In addition, the theoretical results have been compared with the simulation results obtained from Monte Carlo analysis. The results reveal that there is little difference between the theoretical and Monte Carlo analysis when the random codes have a value of $m = 20$ (m is the number of parity bits), but with increasing values of m , the Monte Carlo analysis becomes impractical.

New constructions for the wiretap channel with security and error-correction guarantees have been proposed by Cassuto and Bandic [45]. They have discussed the state of wire-tap II channel in two directions: in the first case of error-free main channels, two families of codes were constructed with optimal encoding and decoding complexities for their wiretap security to ensure access to perfect security against an eavesdropper. For the second case of main channels with errors, two concatenation types were investigated for wiretap and error correcting codes in order to obtain an optimal construction with security and error-correction guarantees.

Recently, Zhang *et al.* [46] proposed a chain-based syndrome coding scheme in a wiretap channel, where the main channel is noiseless and the eavesdropper channel is the binary symmetric channel. The main difference between the conventional syndrome coding scheme and the chain-based syndrome coding scheme is the chain depth, where the encoding and decoding processes are memoryless in the conventional syndrome coding scheme. This is a special case of the chain-based syndrome coding scheme where the chain depth is equal to zero. In contrast, the encoder and decoder exhibit memory in the chain based syndrome coding scheme. The simulation results showed that the equivocation rate has been increased significantly in the proposed model compared with the conventional syndrome coding scheme and they found from analytic results that the chain technique helps to increase the equivocation of the eavesdropper as long as the chain depth increases.

Code design for error correcting codes is an important and long standing topic in coding theory. Numerous contributions have been made in concatenating, extending or shortening the codes to obtain good codes [16]. Traditionally, these codes are designed for an error correcting system to eliminate the noisy transmission channels and secure the reliability of communications. The essential and fundamental parameter of an error correcting code that is responsible for achieving reliable communication is the minimum Hamming distance, d , which provides a measure of the number of errors that can be corrected.

An error correcting code can be used in syndrome coding, but this does not necessarily mean getting the best performance and better codes. Generally code performance in syndrome coding can be measured by the equivocation rate of the codes when used in an unreliable channel and the best codes have the highest value of equivocation rate.

It is worth mentioning that all the codes used in this thesis are restricted to binary, linear codes. The information rate of a syndrome coding scheme using an $[n, k, d]$ linear code is $(\frac{n-k}{n})$ and all binary vectors of length n may be transmitted, while the information rate of error correcting coding applications is $(\frac{k}{n})$ but only codewords are transmitted.

Determining the generator matrix G or the parity check matrix H is the critical objective of code design. Traditional code design methods are either based on the design of the generator matrix or based on the design of the parity check matrix. For example, good codes may be derived by extending good short codes in length or by shortening good long codes [16]. In [47], Bouyukliev introduces two algorithms to extend the length of codes, by modifying the generator matrix of the code. In deriving the best known codes (*BKC*) table, methods have been used, such as code extension [48–50] and code shortening [16]. The entire purpose of these methods is to maximise the minimum Hamming distance d of the codes. Construction X and its variations increase the length of a code in order to enlarge the minimum weight, which use two or more codes of the same length where one code contains the other codes which have a higher minimum weight [49, 50].

Construction Y_1 is a construction method which decreases both the length and the number of parity bits m ($m = n - k$) without compromising the minimum Hamming distance

d of the shortened code [16].

Recently, Zhang *et al.* [51] proposed a code design technique which produces some best known equivocation codes² for the syndrome coding scheme. The best known equivocation codes have been determined for a given number of parity bits m , of the code as follows: $m = 4, 5, 6, 7, 8, 9, 10, 11, 12, 12, 14, 16, 18, 20, 22, 24, 26$.

The main objective of this design is to produce a best known equivocation codes of higher equivocation rate for the syndrome coding scheme compared to all best known error correcting codes. The codes are listed in an on-line database [52] in packed integer format. These codes have a respectable minimum Hamming distance d , but are sometimes not as good as the best known code with the same parameters.

²For given code parameters n and m the best known equivocation code has the highest equivocation rate for *BSC* compared to all other known codes with the same parameters.

Chapter 3

Implementation of Secrecy Coding for the Wiretap Channel using BKLC

3.1 Introduction

In this chapter, two models are proposed for a special case of wiretap channel when the main channel is an error free channel and the eavesdropper channel is a binary symmetric channel. The desired goal in this phase is to maximise the equivocation rate in the eavesdropper side with the result that the communication security is improved. The model design in Fig. 3.1 includes an inner code and an outer code. Best results are obtained when the outer code employs a syndrome coding scheme based on the $(23, 12, 7)$ binary Golay code and the inner code employs the McEliece cryptosystem technique based on Best Known Linear codes(*BKLC*).

In the first model, referred to as Model-1, the transmitted message is first encoded by a syndrome coding scheme based on the $(23, 12, 7)$ binary Golay code and secondly using McEliece technique based on the *BKLC* $(33, 23, 5)$.

In the second model, referred to as Model-2, the first stage employs a syndrome coding scheme based on the $(23, 12, 7)$ binary Golay code and the second stage employs the McEliece technique based on the *BKLC* $(58, 46, 5)$, by concatenating two Golay coded vectors and using this as the input to the second stage.

Analysis shows that the second model increases the equivocation rate of the eavesdropper compared to the first model. In addition, the results obtained show that the arrangement reduces the information leakage by a large margin compared to previously published schemes.

Parts of this chapter are published in the following conference:

S. Al-Hassan, M. Ahmed, and M. Tomlinson,“**Secrecy coding for the wiretap channel using best known linear codes**”, IEEE Conference, **The 5th Global Information Infrastructure and Networking Symposium , Trento, Italy, October 2013.**

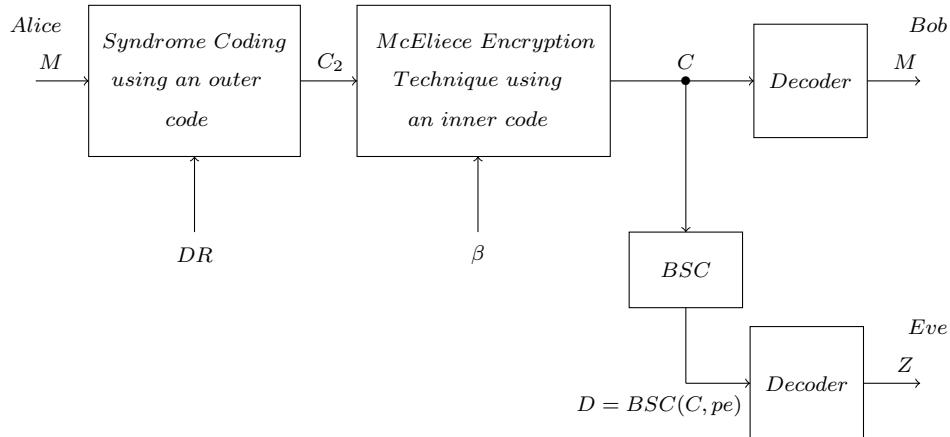


Figure 3.1: Block Diagram of Proposed Coding scheme for the Wiretap Channel

3.2 McEliece Public Key Cryptosystem

3.2.1 Definitions

Irreducible polynomial

A polynomial that cannot be written as the product of two polynomials is called an irreducible polynomial.

Binary Goppa Code

Let $G(x)$ is an irreducible polynomial of degree t over $\text{GF}(2^m)$ and $L=(\alpha_1, \dots, \alpha_n)$ subset of $\text{GF}(2^m)$ such that $G(\alpha_i) \neq 0$. The binary Goppa code $\Gamma(L, G)$ is defined by the rational function [16] as follows:

$$R_a(z) = \sum_{i=1}^n \frac{a_i}{z - \alpha_i} = 0 \bmod G(z) \quad (3.1)$$

The parameters of Goppa code can be described as [53]:

- The length of $\Gamma(L, G)$ is $n = 2^m$.
- The dimension of $\Gamma(L, G)$ is $k \geq n - tm$.
- The minimum distance of $\Gamma(L, G)$ is $d \geq 2t + 1$.

McEliece Cryptosystem

The McEliece cryptosystem is an asymmetric encryption algorithm that was proposed by Robert McEliece in 1978. Based on algebraic coding theory [54], the original algorithm uses binary Goppa codes; these codes are linear and has a fast decoding algorithm. Any linear code with a fast decoding algorithm can be used in McEliece cryptosystem [55], but the general problem of finding a codeword of a given weight in a linear binary code is NP-complete [56].

Although the McEliece cryptosystem was one of the first public key algorithms and very efficient, it has not received wide acceptance in the cryptographic community.

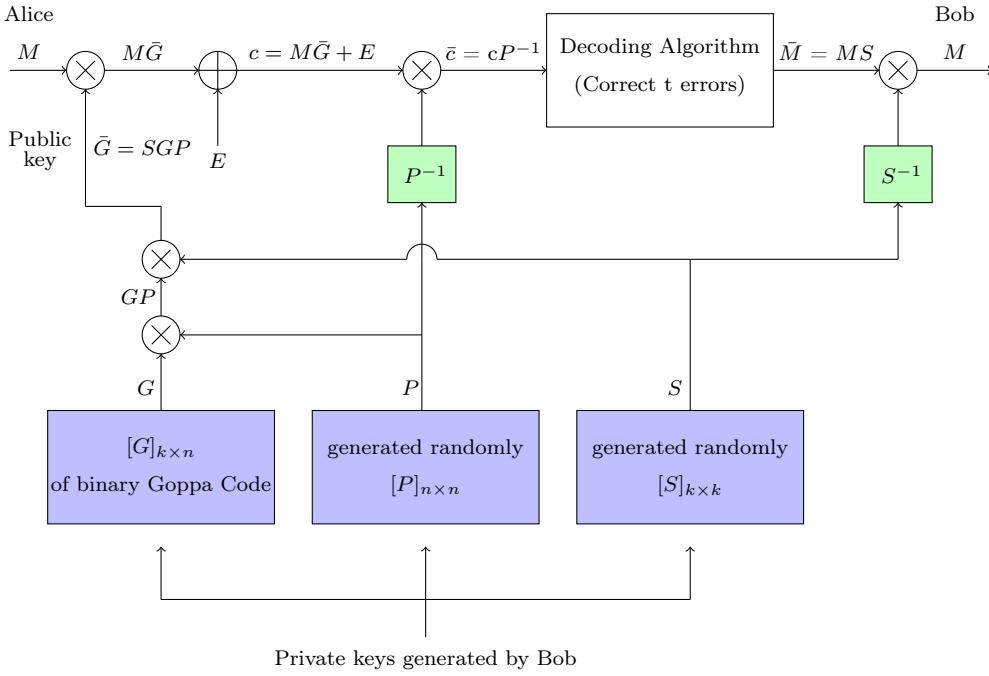


Figure 3.2: Block Diagram of McEliece Cryptosystem

The McEliece cryptosystem have some advantages compared with other public key cryptosystems, like RSA [57]. Firstly, the speed of the encryption and decryption process are faster. Secondly, it employs the probabilistic encryption which is better than deterministic encryption in preventing information loss against any attacks [58]. On the other hand, a drawback of the system is that the length of the public key is large and the length of the ciphertext is much longer than the original message. The block diagram of McEliece cryptosystem is shown in Fig. 3.2.

3.2.2 Key Generation

In general, the parameters of the McEliece cryptosystem are the parameters of the selected Goppa code . Bob selects a $(n, k, 2t + 1)$ binary Goppa code and the error correcting capability of the code t . The process of generating the public and private keys are summarized in Algorithm 4 [59–61].

Algorithm 4 Key Generation for McEliece Cryptosystem

Require: m, t ▷ Security parameters of binary Goppa code
 1: $n \leftarrow 2^m$, $k \leftarrow n - m \times t$ ▷ Bob select a $(n, k, 2t + 1)$ binary Goppa code
Require: $[G]_{k \times n}$ ▷ Select a Generator matrix for the Goppa code
Require: $[S]_{k \times k}$ ▷ Select randomly a Scrambler(binary non-singular) matrix
Require: $[P]_{n \times n}$ ▷ Select randomly a Permutation matrix
 2: $[\bar{G}]_{k \times n} \leftarrow S \times G \times P$ ▷ Compute the public Generator matrix
 3: **return** Public key(\bar{G}, t); Private key(G, S, P)

3.2.3 Encryption Algorithm

Assume Alice wants to send a message M to Bob. Firstly, Alice must obtain the public key of Bob (\bar{G}, t). Algorithm 5, shows the encryption process that is performed by Alice [59–61].

Algorithm 5 Message Encryption for McEliece Cryptosystem

Require: (\bar{G}, t) ▷ Obtain public key of Bob
Require: $[M]_{1 \times k}$ ▷ Represent the message M as a blocks of binary string of k -bit
Require: $[E]_{1 \times n}$ ▷ Generate a random error vector E of length n -bit and weight t
 1: $c \leftarrow M \times \bar{G} + E$ ▷ Compute the ciphertext c
 2: **return** c ▷ Alice sends the ciphertext c to Bob

3.2.4 Decryption Algorithm

Bob receives the ciphertext c from Alice. Bob should do the following steps to recover the original message M [59–61]:

Algorithm 6 Message Decryption for McEliece Cryptosystem

Require: (c, G, P)
Require: $[P]^{-1}, [S]^{-1}$ ▷ Compute the inverse of permutation and scrambler matrix
 1: $\bar{c} \leftarrow c \times P^{-1}$
 2: $\bar{M} \leftarrow \bar{c} \times S$ ▷ Use the decoding algorithm for the Goppa code to decrypt \bar{c} to \bar{M}
 3: $M \leftarrow \bar{M} \times S^{-1}$
 4: **return** M ▷ Bob recovers the plaintext M

From Algorithm 6(step 2), we can prove that $\bar{M} = M \times S$ as follows:

$$\bar{c} = c \times P^{-1}$$

$$\bar{c} = (M \times \bar{G} + E) \times P^{-1}$$

$$\bar{c} = (M \times S \times G \times P + E) \times P^{-1}$$

$$\bar{c} = (M \times S) \times G + E \times P^{-1}$$

$$\bar{c} = \bar{M} \times G + \bar{E}$$

3.3 Proposed Coding scheme for the Wiretap Channel [Model-1]

The model design in Fig. 3.3 includes two Best Known Linear Codes to increase the security of the transmitted message to the legitimate receiver and to maximise the equivocation rate of the eavesdropper. The first stage employs a syndrome coding scheme based on the $(23, 12, 7)$ binary Golay code and the second stage employs the McEliece cryptosystem technique based on $BKLC(33, 23, 5)$.

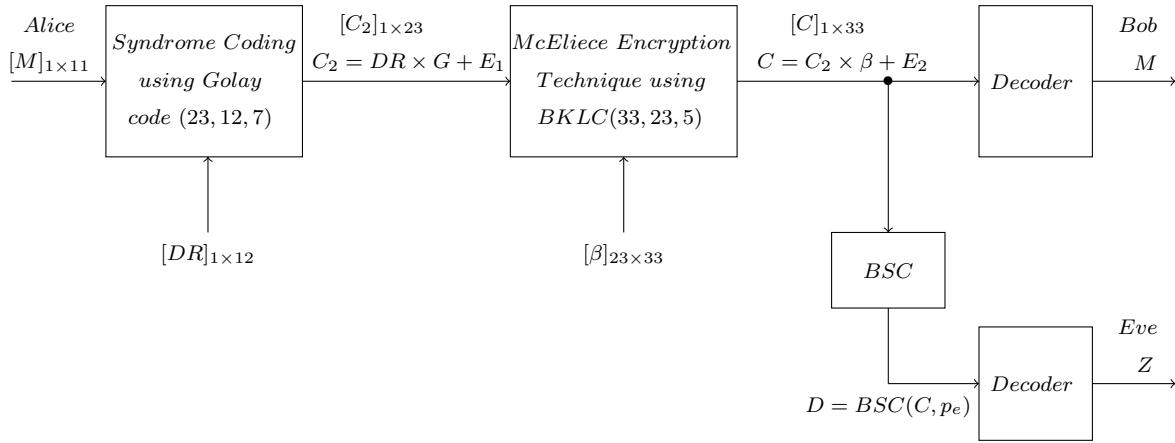


Figure 3.3: Block Diagram of Proposed Coding scheme for the Wiretap Channel[Model-1]

The $(23, 12, 7)$ binary Golay code is used because it is a perfect linear error-correcting code. In addition, the $BKLC(33, 23, 5)$ was selected as the second stage because the output vector from the first stage of Golay code has a length of 23-bits. Therefore, the length of the public key of second stage must be equal to 23.

The model will be analysed through three parts. The first part shows the encoding operation(from Alice), which includes two stages. The second part will explain the decoding operation in the legitimate receiver (Bob). Finally, the information received by the eavesdropper via the binary symmetric channel will be explained.

The modified coding scheme has been implemented using C++ with the support of the NTL library and Magma software.

3.3.1 Encoding Algorithm

First Stage [Syndrome coding scheme based on (23, 12, 7) binary Golay code]

The (23, 12, 7) binary Golay code has the following parameters shown below:

$$\text{Length}(n) = 23 \quad \text{Dimension}(k) = 12 \quad \text{Message}(M) = n - k = 11$$

$$\text{No. of syndromes} = 2^{11} = 2048 \quad \text{Minimum distance}(d) = 7$$

$$\text{No. of correctable Errors}(t \leq \lfloor \frac{d-1}{2} \rfloor) = (0 \rightarrow 3) \text{ errors}$$

The block diagram of syndrome coding scheme based on the (23, 12, 7) binary Golay code is shown in Fig. 3.4 and Algorithm 7 shows how Alice started the encryption process in order to generate the 23-bit vector C_2 .

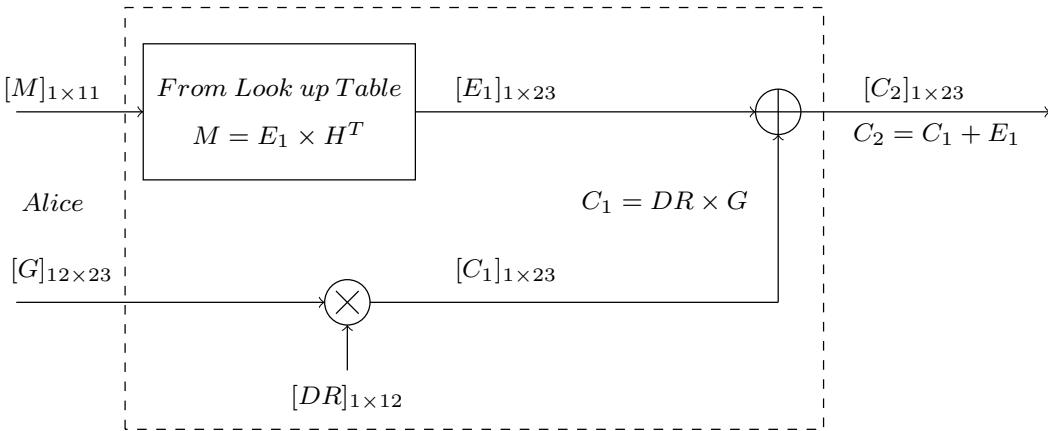


Figure 3.4: Block diagram of syndrome coding scheme based on (23, 12, 7) Golay code

Algorithm 7 Encoding Algorithm of (23, 12, 7) Golay code

Require: $[G]_{12 \times 23}$ ▷ The Generator matrix of Golay code
Require: $[H]_{23 \times 11}^T$ ▷ The parity check transpose matrix of Golay code
Require: $[DR]_{1 \times 12}$ ▷ Generate random data vector

- 1: **Generate** $[DR]_{1 \times 12}$
- 2: $[C_1]_{1 \times 23} \leftarrow [DR]_{1 \times 12} \times [G]_{12 \times 23}$
- 3: $[E_1]_{1 \times 23} \leftarrow [M]_{1 \times 11}$ ▷ Generate E_1 from M
- 4: $[C_2]_{1 \times 23} \leftarrow [C_1]_{1 \times 23} + [E_1]_{1 \times 23}$
- 5: **return** $[C_2]_{1 \times 23}$

Now, we illustrate how to calculate the error pattern $[E_1]_{1 \times 23}$ of weight between 0 and

3. The error vector E_1 is related to the Message $[M]_{1 \times 11}$. So, the mapping between each error pattern and syndromes is one-to-one because the Golay code is a perfect code. Therefore the error vector can be calculated by setting $S_i = M_i (i = 0, 1, \dots, 2047)$,

E_1 can be calculated from $S_1 = E_1 H^T = M$, where H^T represents the transpose of the parity check matrix. As there are only 2048 syndromes, it is straightforward to generate a look up table linking S_1 to E_1 . Bob calculates \hat{M} the estimate of the message as follows:

$$\hat{M} = C_2 \times H^T = [C_1 + E_1] \times H^T = [DR \times G + E_1] \times H^T$$

$$\hat{M} = DR \times G \times H^T + E_1 \times H^T$$

Since G and H are orthogonal ($G \times H^T = 0$), then $\hat{M} = E_1 \times H^T = S_1 = M$.

From the last equation, M and H^T are known. Therefore the error vector E_1 can be computed.

Second Stage [McEliece cryptosystem technique based on $BKLC(33, 23, 5)$]

The block diagram of McEliece cryptosystem technique based on $BKLC(33, 23, 5)$ is shown in Fig. 3.5. Key generation and message encryption are described below:

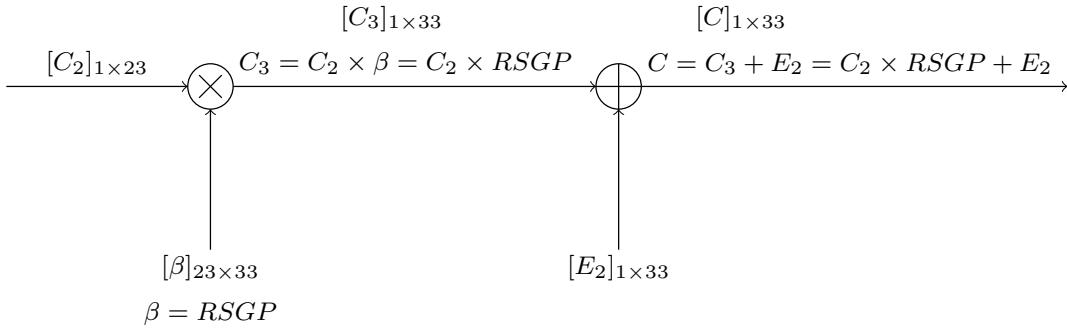


Figure 3.5: Block Diagram of McEliece cryptosystem technique based on $BKLC(33, 23, 5)$

1) Key generation: Bob selects a $BKLC(33, 23, 5)$ which can correct 2 errors. The process of generating the public and private keys are summarized in Algorithm 8. From Algorithm 8, R can be produced by row additions and column swaps of α that led to obtain a reduced echelon form of α , i.e. $R.\alpha = [I_k \mid Q]$ where I_k is the (23×23) identity matrix and Q is $(k \times (n - k))$ parity check matrix.

Algorithm 8 Key Generation of $BKLC(33, 23, 5)$

Require: $[G]_{23 \times 33}$ ▷ Generator matrix of $BKLC(33, 23, 5)$
Require: $[S]_{23 \times 23}$ ▷ Select randomly a Scrambler matrix
Require: $[P]_{33 \times 33}$ ▷ Select randomly a Permutation matrix
1: **Generate** $[S]_{23 \times 23}$ ▷ Generate non-singular matrix
Ensure: $|S| \neq 0$ ▷ The determinant of S not equal 0
2: **Generate** $[P]_{33 \times 33}$
3: $[\alpha]_{23 \times 33} \leftarrow SGP$
4: $[R][\alpha] \leftarrow [I_{23} | Q]$ ▷ Determine $[R]_{23 \times 23}$
Ensure: $[R] \neq [S]^{-1}$
5: $[\beta]_{23 \times 33} \leftarrow RSGP$
6: **return** $[\beta]_{23 \times 33}$

The public key, which will be known to everyone, is $[\beta]_{23 \times 33} = RSGP$. While S, G, P and R form the private key kept by Bob. It is assumed in the following that Eve has knowledge of the private key.

2) Message encryption: The following steps show completion of the encryption process by Alice, where the output of the first stage $[C_2]_{1 \times 23}$ is used as input to the second stage:

Algorithm 9 Message Encryption of $BKLC(33, 23, 5)$

Require: C_2, β
1: $[C_3]_{1 \times 33} \leftarrow [C_2]_{1 \times 23} \times [\beta]_{23 \times 33}$
2: $[C]_{1 \times 33} \leftarrow [C_3]_{1 \times 33} + [E_2]_{1 \times 33}$ ▷ Alice adds error vector E_2 of weight 2
3: **return** $[C]_{1 \times 33}$

3.3.2 Legitimate Receiver's Decoder

Bob receives the transmitted vector $[C]_{1 \times 33}$ via the main channel that is error-free. He recovers the original message M as shown in Algorithm 10.

Algorithm 10 Legitimate Receiver's Decoder

Require: C, P^{-1}, H^T ▷ Bob using P^{-1} and H^T of $BKLC(33, 23, 5)$
1: $S_1 \leftarrow C \times P^{-1} \times H^T$ ▷ Bob computes $[S_1]_{1 \times 33}$
2: $S_1 \leftarrow E_2 \times P^{-1} \times H^T$ ▷ Bob computes E_2
3: $C_3 \leftarrow C + E_2$ ▷ Bob corrects the error E_2 in C to get C_3 of 33-bit
4: $C_2 \leftarrow C_3$ ▷ Bob obtain C_2 from C_3 by get only the first 23-bit
5: $S_2 \leftarrow C_2 \times H^T$ ▷ H^T of $(23, 12, 7)$ Golay code
6: $\hat{M} \leftarrow S_2, M = \hat{M}$ ▷ Bob recovers an estimate of the message \hat{M}
7: **return** \hat{M}

From Algorithm 10(step 3), we can prove that $S_1 = E_2 \times P^{-1} \times H^T$ as follows:

$$S_1 = C \times P^{-1} \times H^T = (C_3 + E_2) \times P^{-1} \times H^T$$

$$S_1 = (C_2 \times RSGP) \times P^{-1} \times H^T + E_2 \times P^{-1} \times H^T$$

$$S_1 = C_2 \times RSG \times H^T + E_2 \times P^{-1} \times H^T$$

But $G \times H^T = 0$.

$$\text{So, } S_1 = E_2 \times P^{-1} \times H^T.$$

Also, we can give an explanation about the mechanism that is used by Bob to obtain $[C_2]_{1 \times 23}$ from $[C_3]_{1 \times 33}$ by getting the first 23-bit only as follows:

$$[C_3]_{1 \times 33} = [C_2]_{1 \times 23} \times [\beta]_{23 \times 33}$$

$$[C_3]_{1 \times 33} = [C_2]_{1 \times 23} \times [I_{23 \times 23} \mid Q_{23 \times 10}]$$

This means the first 23-bit of vector C_3 represents C_2 exactly.

3.3.3 Eavesdropper's Decoder

The block diagram of the *BSC* channel and decoder of the eavesdropper, Eve is shown in Fig. 3.6.

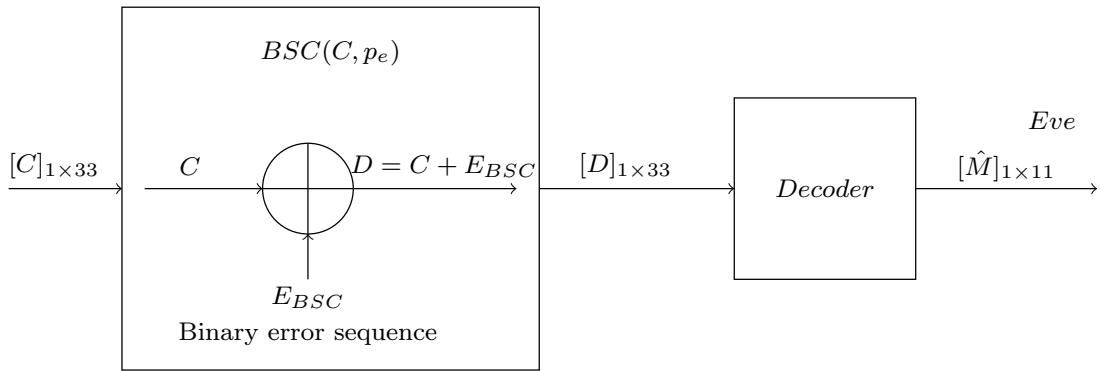


Figure 3.6: Block Diagram of the *BSC* channel and Decoder of Eavesdropper

Eve receives a corrupted vector D instead of the transmitted vector $[C]_{1 \times 33}$ as a result of passing through the *BSC* which adds additional errors $[E_{BSC}]_{1 \times 33}$ as follows:

$$[D]_{1 \times 33} = [C]_{1 \times 33} + [E_{BSC}]_{1 \times 33}.$$

Where $[E_{BSC}]_{1 \times 33}$ is a random binary error vector which depends on the probability of error p_e of *BSC*. Assume Eve uses the same type of decoder that has been used by Bob, Algorithm 11 explains how gets the estimated message \hat{M} from the corrupted vector D .

Algorithm 11 Eavesdropper's Decoder

Require: D, P^{-1}, H^T

- 1: $S_3 \leftarrow D \times P^{-1} \times H^T$ \triangleright Eve using P^{-1} and H^T of $BKLC(33, 23, 5)$
- 2: $S_3 \leftarrow \hat{E} \times P^{-1} \times H^T$ \triangleright Eve computes $[S_3]_{1 \times 33}$
- 3: $\hat{C}_3 \leftarrow D + \hat{E}, \hat{C}_3 \neq C_3$ \triangleright Eve computes \hat{E}
- 4: $\hat{C}_2 \leftarrow \hat{C}_3$ \triangleright Eve corrects the error \hat{E} to get \hat{C}_3 of 33-bit
- 5: $S_4 \leftarrow \hat{C}_2 \times H^T$ \triangleright Eve obtain \hat{C}_2 from \hat{C}_3 by get only the first 23-bit
- 6: $\hat{M} \leftarrow S_4, M \neq \hat{M}$ \triangleright H^T of $(23, 12, 7)$ Golay code
- 7: **return** \hat{M} \triangleright Eve recovers an estimate of the message \hat{M}

From Algorithm 11(step 3), we can prove that $S_3 = \hat{E} \times P^{-1} \times H^T$ as follows:

$$S_3 = D \times P^{-1} \times H^T = (C + E_{BSC}) \times P^{-1} \times H^T$$

$$S_3 = (C_3 + E_2 + E_{BSC}) \times P^{-1} \times H^T$$

$$S_3 = (C_2 \times RSGP) \times P^{-1} \times H^T + E_2 \times P^{-1} \times H^T + E_{BSC} \times P^{-1} \times H^T$$

$$S_3 = C_2 \times RSG \times H^T + E_2 \times P^{-1} \times H^T + E_{BSC} \times P^{-1} \times H^T$$

But $G \times H^T = 0$.

$$\text{So, } S_3 = E_2 \times P^{-1} \times H^T + E_{BSC} \times P^{-1} \times H^T$$

$$S_3 = (E_{BSC} + E_2) \times P^{-1} \times H^T = \hat{E} \times P^{-1} \times H^T.$$

\hat{M} represents the estimated message of 11-bit length and it is equal to $\hat{M} = M + E$.

The error signal E which represents the difference between the original message M and the estimated message \hat{M} , which gives us imagine that Eve will receive an equivocation, that means that the percentage of the original information leakage will be less.

3.4 Proposed Coding scheme for the Wiretap Channel

[Model-2]

The first model has been improved in order to increase the equivocation rate of the eavesdropper. The model design in Fig. 3.7 includes two Best Known Linear Codes. The first stage employs a syndrome coding scheme based on the $(23, 12, 7)$ binary Golay code and the second stage employs the McEliece cryptosystem technique based on $BKLC(58, 46, 5)$.

In this model, the transmitted message is encoded firstly by a syndrome coding scheme based on the $(23, 12, 7)$ binary Golay code. The total numbers of syndromes generated

from first stage is equal to $2^{11} = 2048$ syndromes, each syndrome has a length of 23-bits. Then each two syndromes are concatenated together to create 1024 vectors; each one has a length 46-bit.

The aim of concatenating two syndromes that are produced from the first stage is because the second stage has employed $BKLC(58, 46, 5)$, so the length of the public key must be equal to 46.

The second model has also been implemented using C++ with the support of NTL library and Magma software.

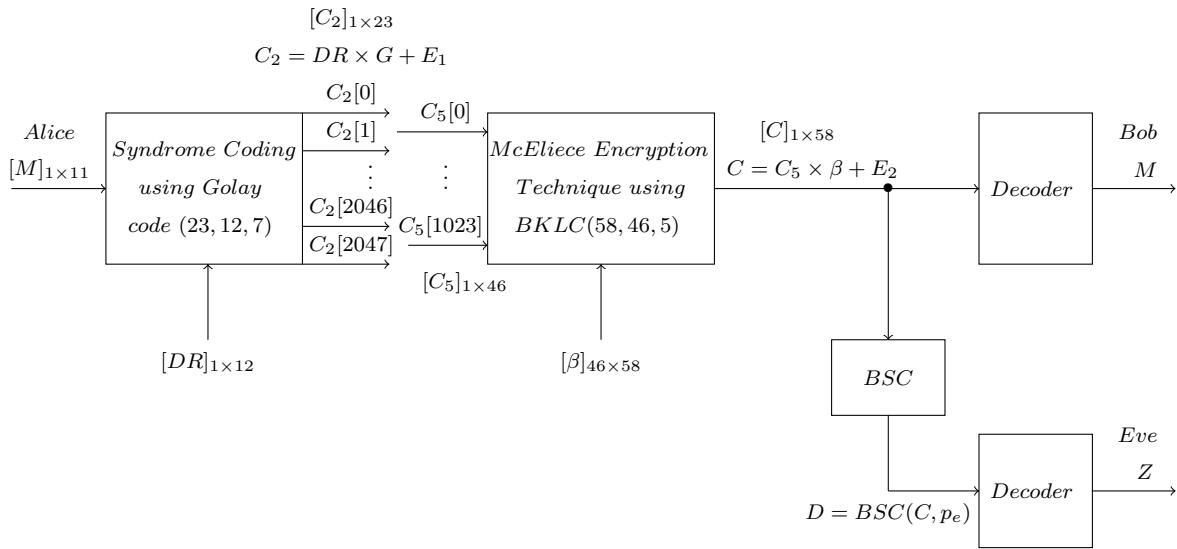


Figure 3.7: Block Diagram of Proposed Coding scheme for the Wiretap Channel[Model-2]

3.4.1 Encoding Algorithm

First Stage [Syndrome coding scheme based on $(23, 12, 7)$ binary Golay code]

Alice will start the encryption process in order to generate 2048 vectors, C_2 , each vector having a length of 23-bit. The encryption process of the first stage in the second model is identical to that in the first model as shown in section 3.3.1 (First Stage).

Second Stage [McEliece cryptosystem technique based on $BKLC(58, 46, 5)$]

The Block Diagram of McEliece cryptosystem technique based on $BKLC(58, 46, 5)$ is shown in Fig. 3.8. The key generation and message encryption are described below:

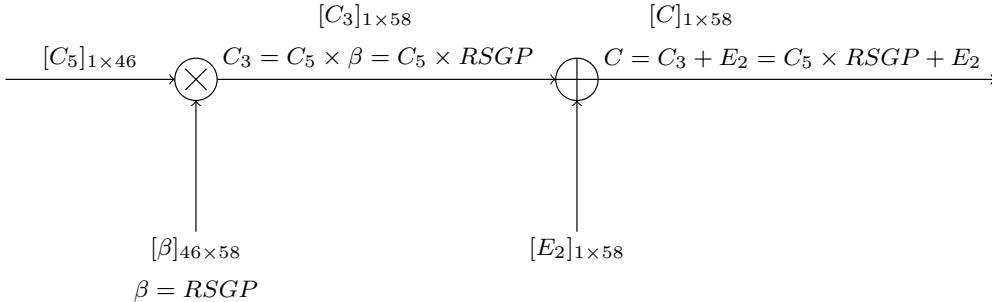


Figure 3.8: Block Diagram of McEliece cryptosystem technique based on $BKLC(58, 46, 5)$

1) Key generation: Bob selects a $BKLC(58, 46, 5)$ which can correct 2 errors. The process of generating the public and private keys are summarized in Algorithm 12.

Algorithm 12 Key Generation of $BKLC(58, 46, 5)$

Require: $[G]_{46 \times 58}$	▷ Generator matrix of $BKLC(58, 46, 5)$
Require: $[S]_{46 \times 46}$	▷ Select randomly a Scrambler matrix
Require: $[P]_{58 \times 58}$	▷ Select randomly a Permutation matrix
1: Generate $[S]_{46 \times 46}$	▷ Generate non-singular matrix
Ensure: $ S \neq 0$	▷ The determinant of S not equal 0
2: Generate $[P]_{58 \times 58}$	
3: $[\alpha]_{46 \times 58} \leftarrow SGP$	
4: $[R][\alpha] \leftarrow [I_{46} \mid Q]$	▷ Determine $[R]_{46 \times 46}$
Ensure: $[R] \neq [S]^{-1}$	
5: $[\beta]_{46 \times 58} \leftarrow RSGP$	
6: return $[\beta]_{46 \times 58}$	

From Algorithm 12, R can be produced by row additions and column swap of α that led to obtain a reduced echelon form of α , i.e. $R.\alpha = [I_k \mid Q]$ where I_k is the (46×46) identity matrix and Q is $(k \times (n - k))$ parity check matrix.

The public key, which will be known to everyone, is $[\beta]_{46 \times 46} = RSGP$. While S , G , P and R form the private key kept by Bob. It is assumed in the following that Eve has knowledge of the private key.

2) Message encryption: The following steps show completion of the encryption process by Alice, which takes the output of the first stage that contains 2048 vectors, $[C_2]_{1 \times 23}$. Then each two vectors are concatenated in order to generate 1024 new vectors, $[C_5]_{1 \times 46}$, and they are sent as the input of the second stage:

Algorithm 13 Message Encryption of $BKLC(58, 46, 5)$

Require: C_5, β

- 1: $[C_3]_{1 \times 58} \leftarrow [C_5]_{1 \times 46} \times [\beta]_{46 \times 58}$
 - 2: $[C]_{1 \times 58} \leftarrow [C_3]_{1 \times 58} + [E_2]_{1 \times 58}$ ▷ Alice adds error vector E_2 of weight 2
 - 3: **return** $[C]_{1 \times 58}$
-

3.4.2 Legitimate Receiver's Decoder

Bob receives the transmitted vector $[C]_{1 \times 58}$ via the main channel that is error-free. He recovers the original message M as shown in Algorithm 14.

Algorithm 14 Legitimate Receiver's Decoder

Require: C, P^{-1}, H^T

▷ Bob using P^{-1} and H^T of $BKLC(58, 46, 5)$

- 1: $S_1 \leftarrow C \times P^{-1} \times H^T$ ▷ Bob computes $[S_1]_{1 \times 58}$
 - 2: $S_1 \leftarrow E_2 \times P^{-1} \times H^T$ ▷ Bob computes E_2
 - 3: $C_3 \leftarrow C + E_2$ ▷ Bob corrects the error E_2 in C to get C_3 of 58-bit
 - 4: $C_5 \leftarrow C_3$ ▷ Bob obtain C_5 from C_3 by get only the first 46-bit
 - 5: $C_2 \leftarrow C_5$ ▷ Bob separates $[C_5]_{1 \times 46}$ into two vectors $[C_2]_{1 \times 23}$
 - 6: $S_2 \leftarrow C_2 \times H^T$ ▷ H^T of (23, 12, 7)Golay code
 - 7: $\hat{M} \leftarrow S_2, M = \hat{M}$ ▷ Bob recovers an estimate of the message \hat{M}
 - 8: **return** \hat{M}
-

From Algorithm 14(step 3), we can prove that $S_1 = E_2 \times P^{-1} \times H^T$ as follows:

$$S_1 = C \times P^{-1} \times H^T = (C_3 + E_2) \times P^{-1} \times H^T$$

$$S_1 = (C_5 \times RSGP) \times P^{-1} \times H^T + E_2 \times P^{-1} \times H^T$$

$$S_1 = C_5 \times RSG \times H^T + E_2 \times P^{-1} \times H^T$$

But $G \times H^T = 0$. So, $S_1 = E_2 \times P^{-1} \times H^T$.

Also, we can give an explanation about the mechanism that used by Bob to obtain $[C_5]_{1 \times 46}$ from $[C_3]_{1 \times 58}$ by getting the first 46-bit only as follows:

$$[C_3]_{1 \times 58} = [C_5]_{1 \times 46} \times [\beta]_{46 \times 58}$$

$$[C_3]_{1 \times 58} = [C_5]_{1 \times 46} \times [I_{46 \times 46} \mid Q_{46 \times 12}]$$

This means, the first 46-bit of vector C_3 represents C_5 exactly.

3.4.3 Eavesdropper's Decoder

The block diagram of the *BSC* channel and decoder of the eavesdropper, Eve is shown in Fig. 3.9.

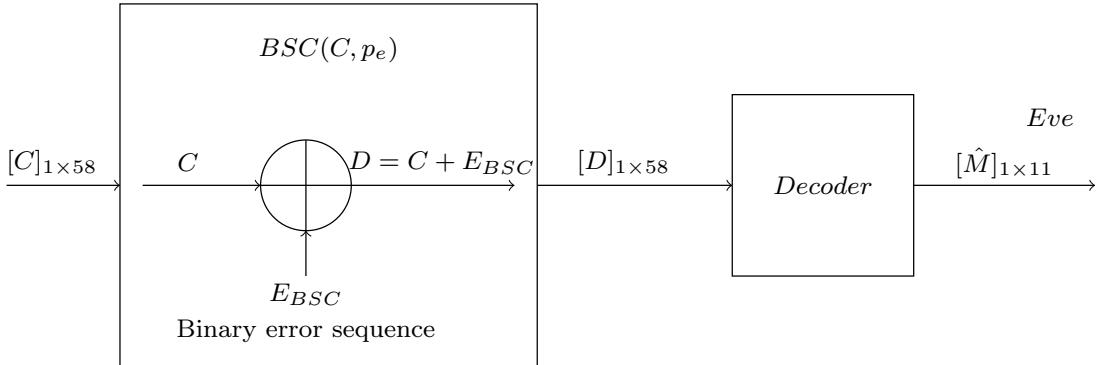


Figure 3.9: Block Diagram of the *BSC* channel and Decoder of Eavesdropper

Eve receives a corrupted vector D instead of the transmitted vector $[C]_{1 \times 58}$ as a result of passing through the *BSC* which adds additional errors $[E_{BSC}]_{1 \times 58}$ as follows:

$$[D]_{1 \times 58} = [C]_{1 \times 58} + [E_{BSC}]_{1 \times 58}.$$

Where $[E_{BSC}]_{1 \times 58}$ is a random binary error vector which depends on the probability of error p_e of *BSC*. Assuming Eve uses the same type of decoder that has been used by Bob, Algorithm 15 explains how to get the estimated message \hat{M} from the corrupted vector D .

Algorithm 15 Eavesdropper's Decoder

Require: D, P^{-1}, H^T	▷ Eve using P^{-1} and H^T of $BKLC(58, 46, 5)$
1: $S_3 \leftarrow D \times P^{-1} \times H^T$	▷ Eve computes $[S_3]_{1 \times 58}$
2: $S_3 \leftarrow \hat{E} \times P^{-1} \times H^T$	▷ Eve computes \hat{E}
3: $\hat{C}_3 \leftarrow D + \hat{E}, \hat{C}_3 \neq C_3$	▷ Eve corrects the error \hat{E} to get \hat{C}_3 of 58-bit
4: $\hat{C}_5 \leftarrow \hat{C}_3$	▷ Eve obtain \hat{C}_5 from \hat{C}_3 by get only the first 46-bit
5: $\hat{C}_2 \leftarrow \hat{C}_5$	▷ Bob separates $[\hat{C}_5]_{1 \times 46}$ into two vectors $[\hat{C}_2]_{1 \times 23}$
6: $S_4 \leftarrow \hat{C}_2 \times H^T$	▷ H^T of $(23, 12, 7)$ Golay code
7: $\hat{M} \leftarrow S_4, M \neq \hat{M}$	▷ Eve recovers an estimate of the message \hat{M}
8: return \hat{M}	

From Algorithm 15(step 3), we can prove that $S_3 = \hat{E} \times P^{-1} \times H^T$ as follows:

$$S_3 = D \times P^{-1} \times H^T = (C + E_{BSC}) \times P^{-1} \times H^T$$

$$S_3 = (C_3 + E_2 + E_{BSC}) \times P^{-1} \times H^T$$

$$S_3 = (C_5 \times RSGP) \times P^{-1} \times H^T + E_2 \times P^{-1} \times H^T + E_{BSC} \times P^{-1} \times H^T$$

$$S_3 = C_5 \times RSG \times H^T + E_2 \times P^{-1} \times H^T + E_{BSC} \times P^{-1} \times H^T$$

But $G \times H^T = 0$.

$$\text{So, } S_3 = E_2 \times P^{-1} \times H^T + E_{BSC} \times P^{-1} \times H^T$$

$$S_3 = (E_{BSC} + E_2) \times P^{-1} \times H^T = \hat{E} \times P^{-1} \times H^T.$$

\hat{M} represents the estimated message of 11-bit length and it is equal to $\hat{M} = M + E$.

The error signal E which represents the difference between the original message M and the estimated message \hat{M} , which gives us imagine that Eve will receive an equivocation, which means that the percentage of the original information leakage will be less.

3.5 Computation Results of the Two Models

The equivocation rate and the information leakage have been calculated for the proposed coding scheme for two models(Model-1 and Model-2). Both models employ two encoding stages; the first stage of models employs a syndrome coding scheme based on the (23, 12, 7) binary Golay code while the second stage of models employs the McEliece cryptosystem technique based on $BKLC_s$. Model-1 employs $BKLC(33, 23, 5)$ while Model-2 employs $BKLC(58, 46, 5)$.

In addition, the equivocation rates and the information leakage of the proposed models have been compared with the results obtained by Zhang *et al.* [62]. Zhang *et al.* calculated the equivocation rates for various codes, such as the (23, 12, 7) Golay code, the extended (24, 12, 7) Golay code and a bunch of the BCH codes in the specific wiretap channel for the syndrome coding scheme. The results have shown that the (23, 12, 7) Golay code had the best performance. Therefore, the results obtained from two schemes in [62] that are based on (23, 12, 7) Golay code have been compared with the results obtained from the proposed models (Model-1 and Model-2).

The first scheme used the conventional syndrome-coding scheme based on the (23, 12, 7) Golay code, referred to as **Golay scheme-1**. In [62], Zhang *et al.* proposed a modified syndrome-coding scheme, which impels the probability distribution of error (E) to be closer to a uniform distribution, so the information leakage to the eavesdropper will

be less. The second scheme used the modified syndrome-coding scheme based on the (23, 12, 7)Golay code, referred to as **Golay scheme-2**.

The results of this chapter have been published in [63].

3.5.1 Joint Entropy

The amount of information in the original message M and estimated message \hat{M} has been calculated as follows:

$$H(M, \hat{M}) = - \sum_{allM} \sum_{all\hat{M}} p(M, \hat{M}) \times \log_2 p(M, \hat{M}) \quad (3.2)$$

where $p(M, \hat{M})$ is the joint probability of M and \hat{M} . Fig. 3.10 shows the joint entropy $H(M, \hat{M})$ vs. p_e .

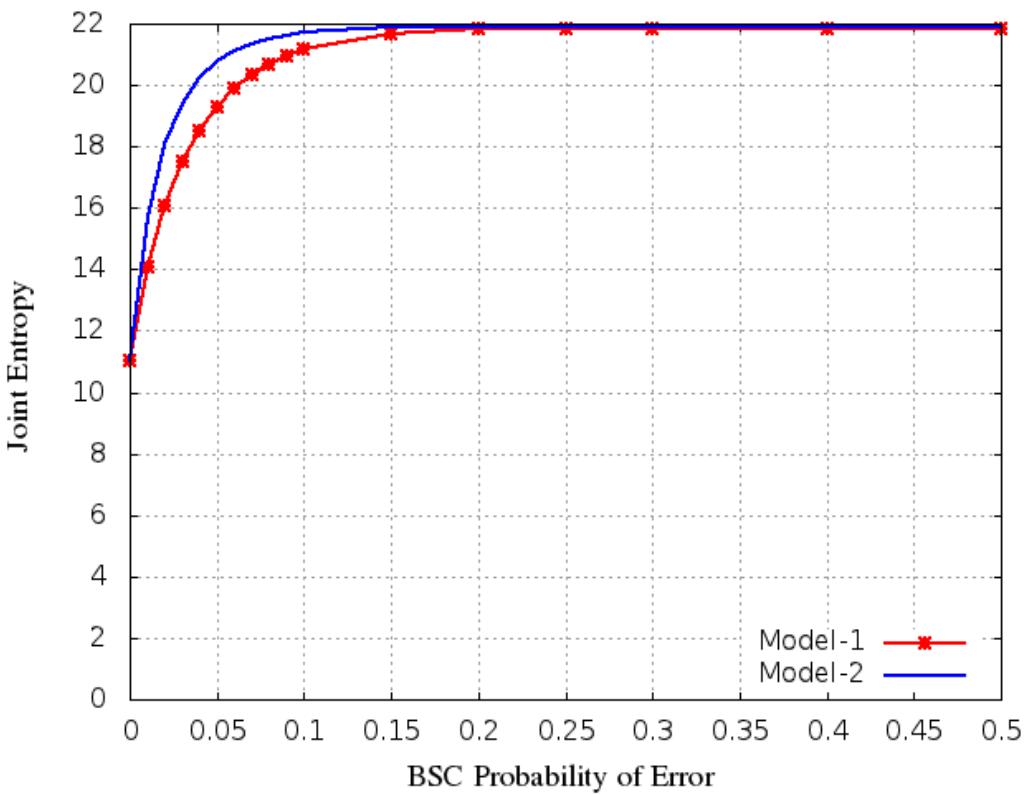


Figure 3.10: The Joint Entropy $H(M, \hat{M})$ vs. p_e

3.5.2 Equivocation

The amount of information lost in the channel during the transition process from Alice to Eve(i.e. the uncertainty of Eve) has been calculated as follows [7]:

$$H(M | \hat{M}) = H(M, \hat{M}) - H(\hat{M}) \quad (3.3)$$

where $H(M, \hat{M})$ is the joint Entropy of M and \hat{M} . Fig. 4.4 shows the Normalised equivocation $\frac{H(M|\hat{M})}{m}$ as a function of probability of error p_e , ($m = n - k$) represents the number of parity bits of the code. This figure shows that the equivocation rate of the Model-2 has increased significantly compared with the Model-1 in the critical values of the probability of error ($0 < p_e \leq 0.1$).

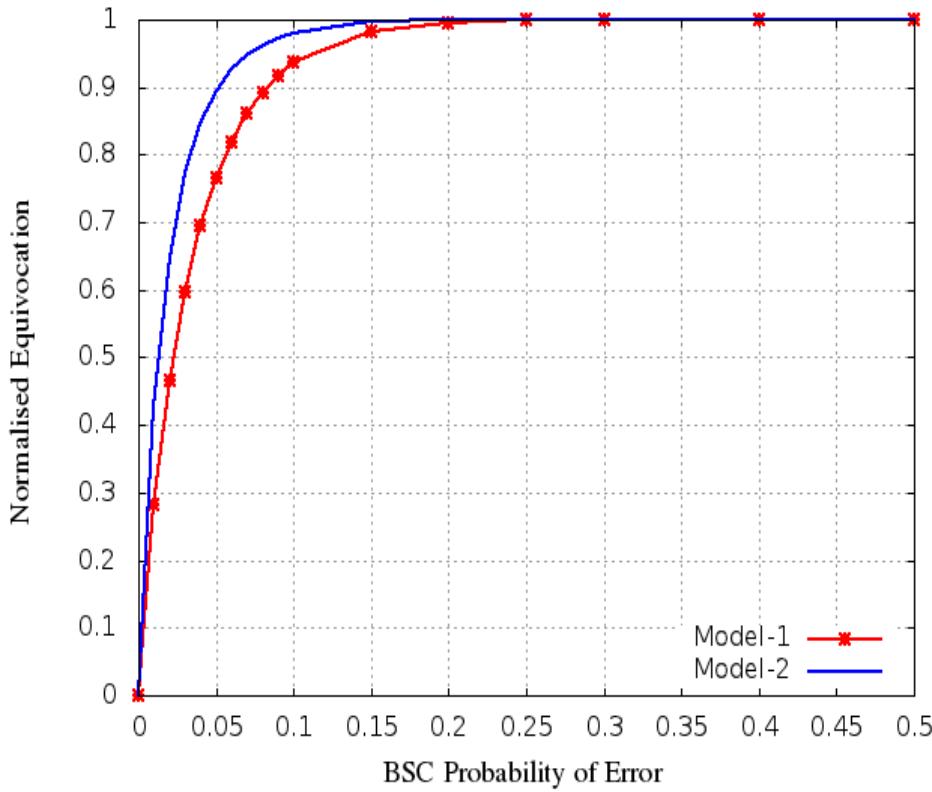


Figure 3.11: The Normalised Equivocation $H(M | \hat{M})$ vs. p_e

3.5.3 Channel Capacity

The amount of information that can be received by Eve from Alice via the channel has been calculated as follows [6]:

$$I(M; \hat{M}) = H(M) + H(\hat{M}) - H(M, \hat{M}) \quad (3.4)$$

where $H(M, \hat{M})$ is the joint Entropy of M and \hat{M} . Fig. 3.12 shows the Normalised information $I(M; \hat{M})$ vs. p_e . This graph gives us the impression of the amount of reduction in the information leakage to the eavesdropper that obtained from Model-2 compared to Model-1.

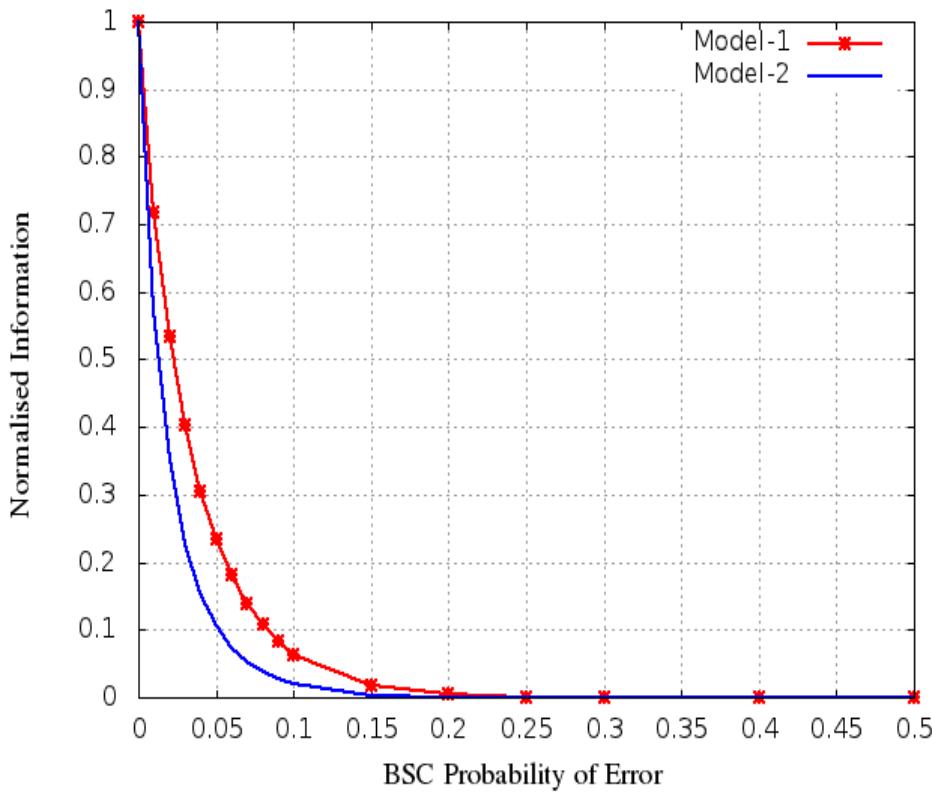


Figure 3.12: The Channel Capacity $I(M; \hat{M})$ vs. p_e

Fig. 3.13 shows the Normalised equivocation for an insecure system on BSC (only binary symmetric channel) and a secure system on BSC of the design models as a function of probability of error p_e . It is clear that the equivocation rate of the secure system is better than insecure system and it can be seen that the difference level in equivocation

rate reaches a maximum when $(0 < p_e \leq 0.1)$.

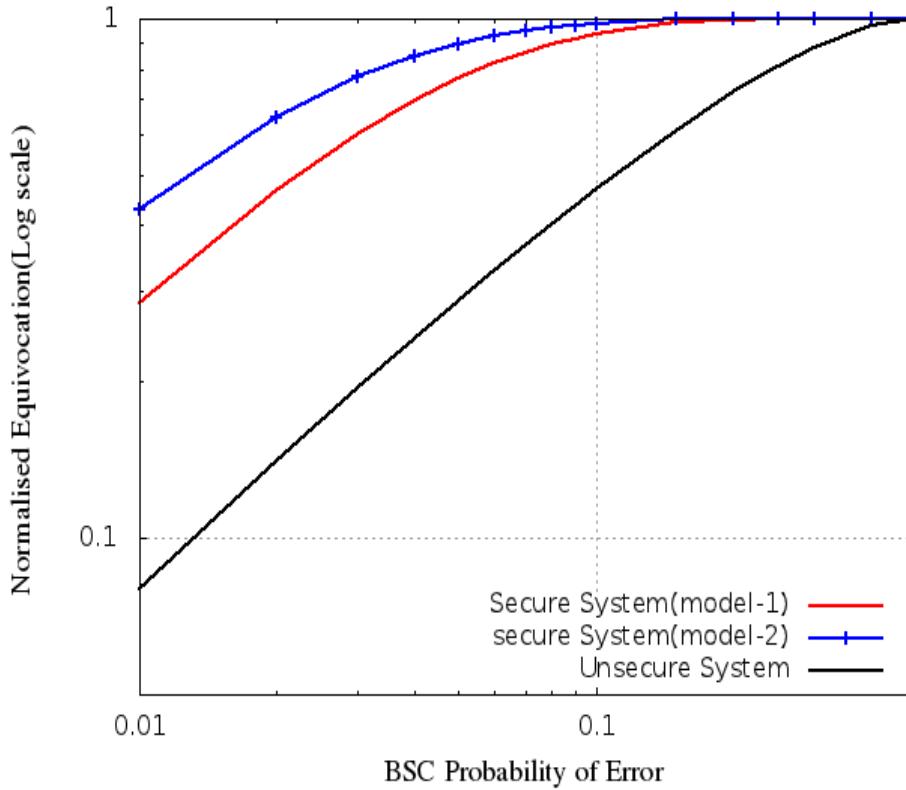


Figure 3.13: Normalised Equivocation for an insecure system on BSC and secure system on BSC vs. p_e

3.5.4 Normalised Equivocation Difference

The Normalised Equivocation Difference between the secure system on BSC of the design models and insecure system on BSC as a function of probability of error p_e can be expressed as:

$$\text{Normalised Equivocation Difference} = Eq_{(secure\ system)} - Eq_{(insecure\ system)} \quad (3.5)$$

This value gives us an idea about the increase occurring in equivocation rate in the secure system on BSC compared to insecure system on BSC . Fig. 3.14 shows that the Model-2 increases the normalised equivocation difference value compared to Model-1.

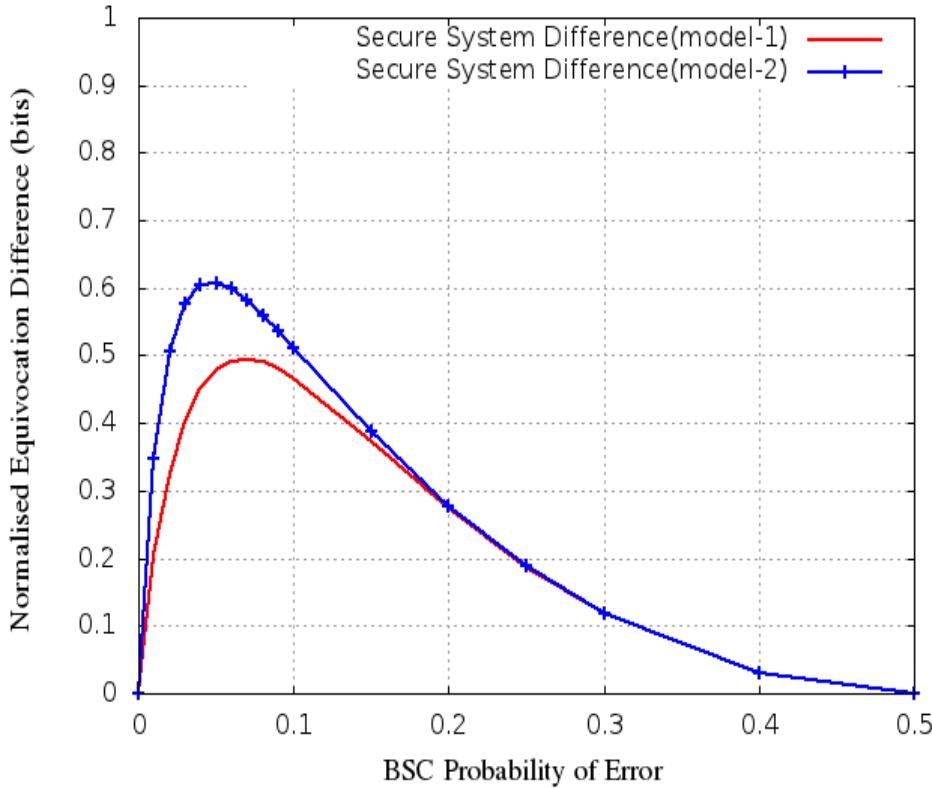


Figure 3.14: Normalised Equivocation Difference between the secure system on BSC and insecure system on BSC vs. p_e

3.5.5 Equivocation Gain

The equivocation gain of the secure system on BSC of the design models and insecure system on BSC as a function of probability of error p_e can be expressed as:

$$\text{Equivocation gain} = \frac{Eq_{(\text{secure system})}}{Eq_{(\text{insecure system})}} \quad (3.6)$$

A huge improvement in equivocation gain has been achieved between secure and insecure system when the probability of error ($0 < p_e \leq 0.1$) as shown in Fig. 3.15.

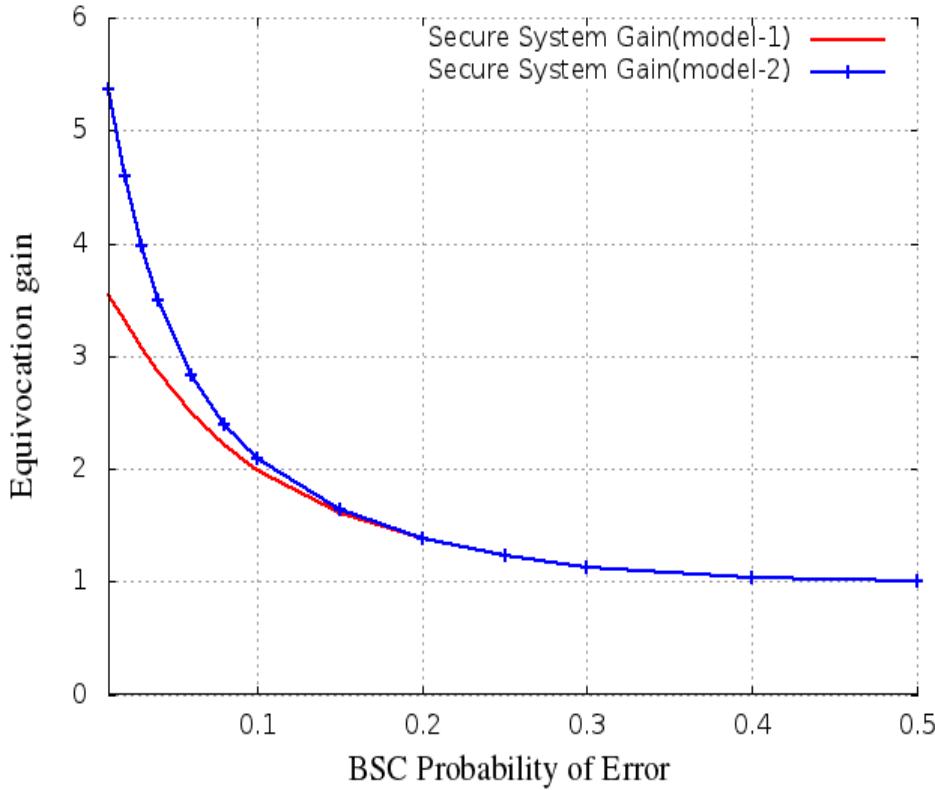
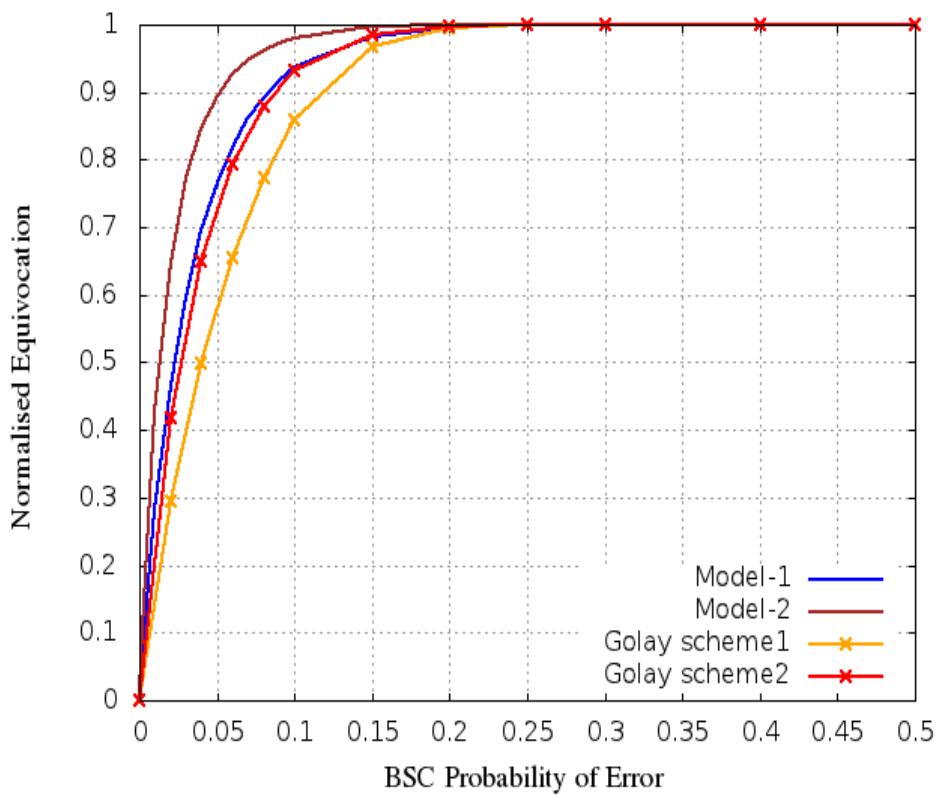
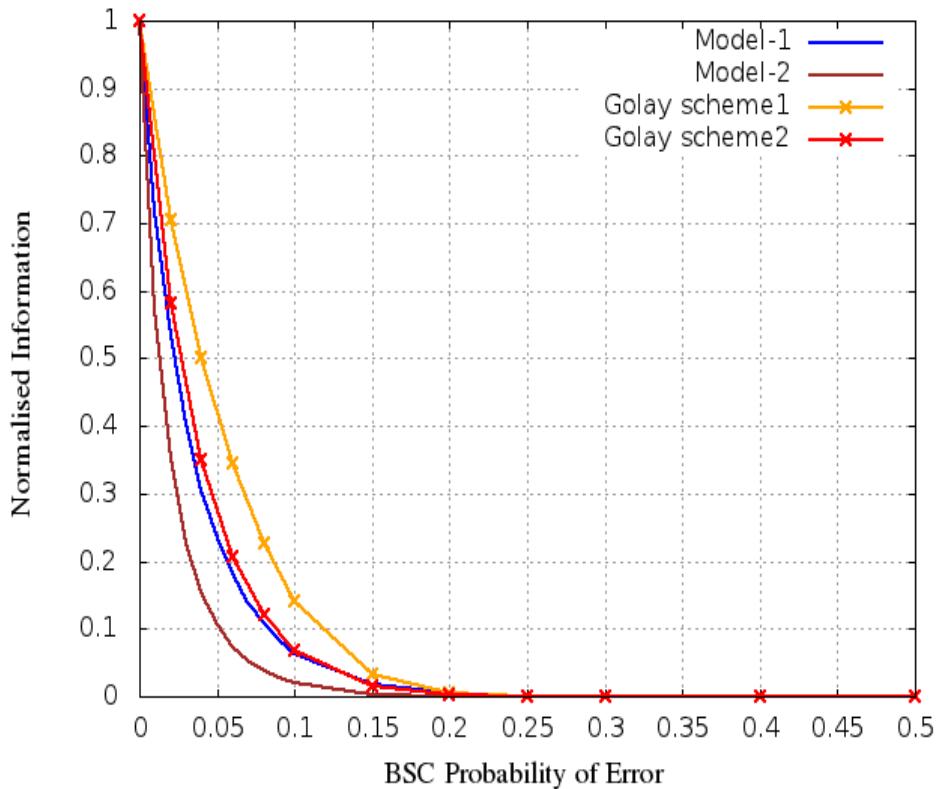


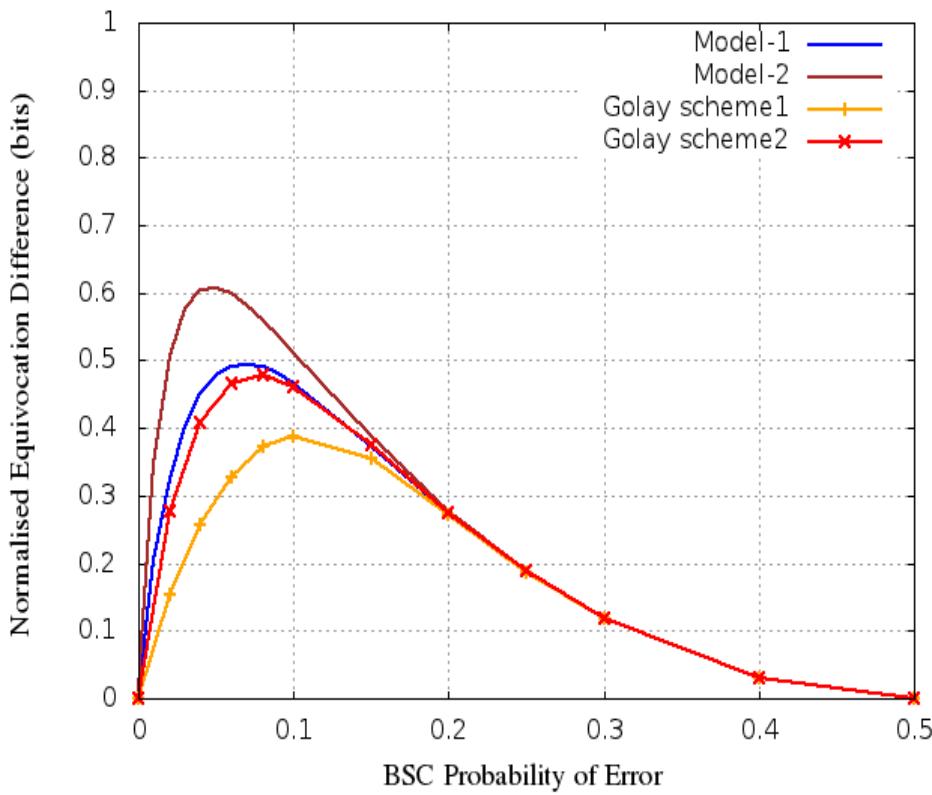
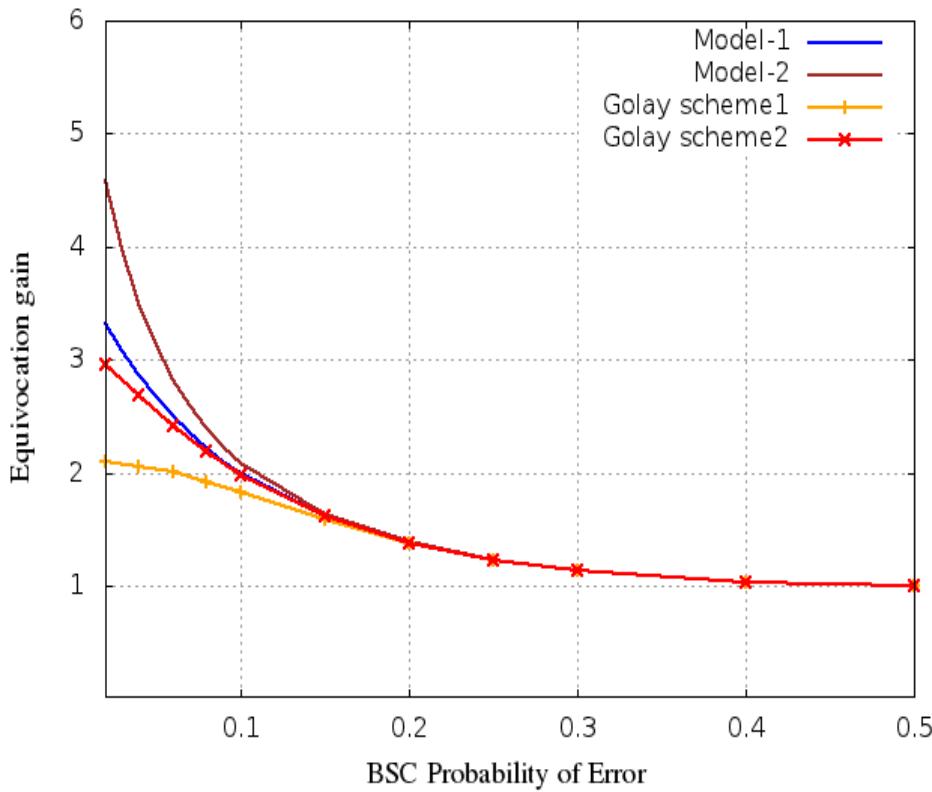
Figure 3.15: Equivocation gain of the secure system on BSC and insecure system on BSC vs. p_e

Figures 4.16, 4.17, 4.18 and 4.19 shows the Normalised equivocation, Channel capacity, normalised equivocation difference and the equivocation gain for the four schemes, two of the new models are proposed in sections 3.3 and 3.4 (Model-1 and Model-2) and two for the models designed by Zhang *et al.* [62].

Analysis shows that the new models (Model-1 and Model-2) increases the equivocation rate hugely compared to other schemes. In addition, the results obtained from the new models (Model-1 and Model-2) showed that the information leakage to the eavesdropper has been reduced by a large margin compared to previously published schemes.

Table 3.1 shows the Normalised equivocation and Information Leakage for the four schemes at some specific values of p_e .

Figure 3.16: Normalised Equivocation vs. p_e Figure 3.17: Channel Capacity vs. p_e

Figure 3.18: Normalised Equivocation Difference vs. p_e Figure 3.19: Equivocation Gain vs. p_e

p_e	Schemes	Normalised Equivocation	Information Leakage
0.02	Golay scheme-1	0.2950	0.7050
	Golay scheme-2	0.4175	0.5825
	Model-1	0.4671	0.5329
	Model-2	0.6474	0.3526
0.04	Golay scheme-1	0.4987	0.5013
	Golay scheme-2	0.6509	0.3491
	Model-1	0.6940	0.3060
	Model-2	0.8475	0.1525
0.06	Golay scheme-1	0.6558	0.3442
	Golay scheme-2	0.7929	0.2071
	Model-1	0.8195	0.1805
	Model-2	0.9264	0.0736
0.08	Golay scheme-1	0.7739	0.2261
	Golay scheme-2	0.8795	0.1205
	Model-1	0.8923	0.1077
	Model-2	0.9625	0.0375
0.1	Golay scheme-1	0.8587	0.1413
	Golay scheme-2	0.9317	0.0683
	Model-1	0.9358	0.0642
	Model-2	0.9806	0.0194

Table 3.1: Normalised equivocation and Information Leakage for all schemes

3.6 Summary

This chapter can be summarized as follows:

- A McEliece cryptosystem algorithms (key generation, Encryption algorithm and Decryption algorithm) have been presented.
- A new coding scheme [Model-1] which including two encoding stages for the wiretap channel has been presented by using $BKLC(33, 23, 5)$ code.
- A new coding scheme [Model-2] which including two encoding stages for the wiretap channel has been presented by using $BKLC(58, 46, 5)$ code which based on the concatenation two syndromes that are produced from the first stage together.
- The results show that the performance of [Model-2] is significantly better than [Model-1] which is attributable to the longer code used in the second model.
- It has been found from the results that both new models (Model-1 and Model-2) considerably reduce the information leakage to the eavesdropper compared to previously published schemes.
- The results of this chapter have been published in [63].

Chapter 4

Implementation and Construction of Best Known Equivocation Codes for Syndrome Coding

4.1 Introduction

This chapter describes and implements three schemes for constructing the best known equivocation codes for syndrome coding in the wiretap channel. These schemes are based on a recursive method for evaluation of the probability mass function of the syndromes of a code which depends only on the columns of the parity check matrix and the probability of error of the binary symmetric channel. Therefore, the construction technique of the best known equivocation codes($BEqC$) for syndrome coding depends on the parity check matrix only without the need for a syndrome look up table.

The wiretap channel used in the syndrome coding scheme is shown in Fig. 4.1, where the main channel (between Alice and Bob) is an error-free channel and the eavesdropper channel is a Binary Symmetric Channel(BSC) with a probability of error (p_e).

The first scheme generates new best known equivocation(BE_qC) codes with 15 parity bits. The design results for $m = 15$ show that these new best known equivocation codes(BE_qC) have better equivocation rate compared to all previously published best

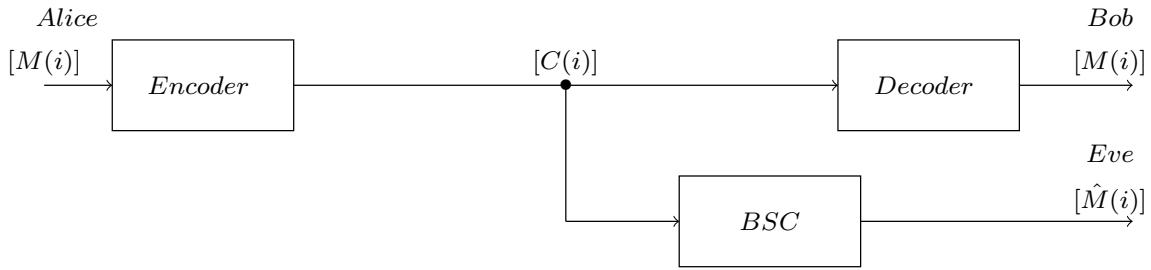


Figure 4.1: Wiretap channel Model

error correcting codes, the best known codes(BKC) compiled by Grassl [2].

In the second scheme, a new code design technique which produces best known equivocation codes(BE_qC) with highest minimum distance for syndrome coding has been presented. Code examples for a given number of parity bits of the code ($m = 7, 11, 12$) also are presented. The design results show that these new best known equivocation codes(BE_qC) have better equivocation rate compared to all previously published best error correcting codes having the same parameters.

Finally, a code design technique has been presented to produce best known equivocation codes by extending the binary linear $[n, k]$ code to a $[n + 2, k + 2]$ code. Analysis shows that the best known equivocation codes produced by adding two columns to the parity check matrix give better equivocation rates compared to those that are produced by the addition of one column in two phases.

Parts of this chapter are published in the following conferences:

- S. Al-Hassan, M. Ahmed, and M. Tomlinson, “New Best Equivocation Codes for Syndrome Coding”, IEEE Conference, International Conference on Information and Communication Technology Convergence (ICTC), Busan, South Korea, October 2014.
- S. Al-Hassan, M. Ahmed, and M. Tomlinson, “Construction of Best Equivocation Codes with Highest Minimum Distance for Syndrome Coding”, IEEE International Conference on Communication, IEEE ICC 2015 - Workshop on Wireless Physical Layer Security, London, UK, June 2015.

- S. Al-Hassan, M. Ahmed, and M. Tomlinson, “Extension of the Parity Check Matrix to Construct the Best Equivocation Codes for Syndrome Coding”, IEEE Conference, Global Information Infrastructure and Networking Symposium (GIIS), Montreal, QC, Canada, September 2014.

4.2 The systematic packed integer form of parity check matrix

Any binary linear $[n, k, d]$ code is defined by its $(k \times n)$ generator matrix G or by its $(m \times n)$ parity check matrix H . The best equivocation codes are constructed by representing the parity check matrix of the code in the systematic packed integer form. H can be defined by representing each column of H by an integer, b_i , in the range 0 to $(2^{n-k} - 1)$. The parity check matrix of a code of length n is defined by n integers, referred to as packed integers and can be placed in any order, so all the corresponding codes will be equivalent. The parity check matrix produces a reduced echelon form by row additions and column swaps, the first m columns of H is an identity matrix and the remaining $n - m$ columns depend on code design as shown below:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & \dots & 0 & a_{m0} & \dots & a_{(n-1)0} \\ 0 & 1 & \dots & 0 & a_{m1} & \dots & a_{(n-1)1} \\ \vdots & \vdots & \dots & \vdots & \vdots & a_{ij} & \vdots \\ 0 & 0 & \dots & 1 & a_{m(m-1)} & \dots & a_{(n-1)(m-1)} \end{bmatrix}$$

in which $0 \leq j \leq m - 1$, $m \leq i \leq n - 1$ and a_{ij} takes a value of 0 or 1. Each column can be represented as a packed integer defined as $b_i = \sum_{j=0}^{m-1} a_{ij} \times 2^j$.

The packed integers correspond directly to syndrome values in that a single bit error in a transmitted codeword result in a syndrome equal to the packed integer of column corresponding to the bit error position. As the codes are linear any combination of bit error produces a syndrome equal to modulo 2 sum of the packed integers corresponding to each column of the parity check matrix (H) in which a bit error occurred.

Then the systematic packed integer form of parity check matrix can be represented as:

$$H = [1, 2, \dots, 2^{m-1}, b_m, \dots, b_{n-1}] \quad (4.1)$$

where the first m integers represent the identity matrix and the other integers have values between 3 and $2^m - 1$. Usually no integers are repeated ensuring $d \geq 3$ and higher values of d are ensured by the constraint that no integer is a modulo 2 sum of any other $d - 2$, or smaller, number of integers.

4.3 Syndrome Coding Scheme

Wyner showed that the secrecy capacity of the wiretap channel [1] is :

$$C_s = -p_e \times \log_2(p_e) - (1 - p_e) \times \log_2(1 - p_e) \quad (4.2)$$

which is the highest transmission rate that can be obtained while maintaining perfect secrecy. In this model, Alice (transmitter) wants to transmit a sequence of independent and uniformly distributed m -bit binary messages to Bob (legitimate receiver), $M[1], \dots, M[r]$. This sequence of messages is encoded into n -bit words $C[1], \dots, C[r]$. Bob receives the same sequence of n -bit words $C[1], \dots, C[r]$ and Eve receives the sequence of n -bit words $D[1], \dots, D[r]$ where

$$D(i) = C(i) + E_{BSC}(i), \quad i = 1, \dots, r$$

and $E_{BSC}(i)$ represents a n -bit error vector generated by the binary symmetric channel and r is the block length.

The syndrome coding scheme uses a $[n, k, d]$ linear block code which guarantees to correct all error patterns of weight t , where $t = \lfloor (d - 1)/2 \rfloor$. All 2^m syndromes are used to send messages where $m = n - k$ and not just the syndromes corresponding to weight t or less error patterns. For any linear block code there exist 2^m distinct minimum weight error patterns, the coset leaders, in which each pattern produces a distinct syndrome of the total 2^m syndromes. Therefore, these error patterns can be represented in a table of

2^m syndromes. In the traditional syndrome coding, the look up table for error patterns and syndromes is known by Alice, Bob and Eve.

For long codes a syndrome table is impractical, but it is shown below that this look up table is unnecessary and that the parity check matrix H of the code is sufficient taking into consideration the structure of H in systematic format. A block diagram for syndrome coding for a $[n, k, d]$ linear block code is shown in Fig. 4.2.

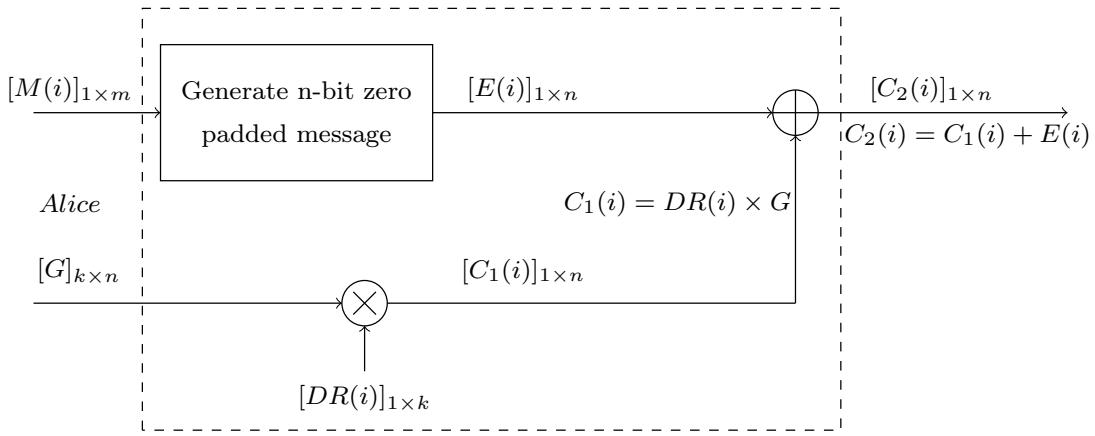


Figure 4.2: Block diagram of syndrome coding scheme for $[n, k, d]$ linear block code

Encoding Algorithm

Alice starts the encryption process in order to generate a n -bit vector $C_2(i)$ from each m -bit message $M(i)$ at time i such that $C_2(i) \times H^T = M(i)$ as shown in Algorithm 16.

Algorithm 16 Encoding Algorithm of $[n, k, d]$ linear code

Require: $[G]_{k \times n}$ Require: $[DR(i)]_{1 \times k}$	\triangleright The Generator matrix of $[n, k, d]$ code \triangleright Generate random uniformly distributed vector
1: Generate $[DR(i)]_{1 \times k}$ 2: $[C_1(i)]_{1 \times n} \leftarrow [DR(i)]_{1 \times k} \times [G]_{k \times n}$ 3: $[E(i)]_{1 \times n} \leftarrow [M(i) 0\dots0]$ \triangleright Generate n -bit zero padded message 4: $[C_2(i)]_{1 \times n} \leftarrow [C_1(i)]_{1 \times n} + [E(i)]_{1 \times n}$ 5: return $[C_2(i)]_{1 \times n}$	

It should be noted that these encoding algorithm differs from the classical method in that a look up table is not required. The message, $n - k$ bits is directly added to the $n - k$ parity bits of the codeword.

Now, we show how to calculate the error pattern $[E(i)]_{1 \times n}$. Since the syndrome of any codeword is zero, any codeword added to an error pattern will produce the same syndrome. Hence Alice may produce the required syndrome by generating an n -bit zero padded message vector $E(i)$, which consists of the original message $M(i)$ which is m -bits long followed by k 0's.

Decoding Algorithm

1. Legitimate Receiver's Decoder

Bob receives the transmitted vector $[C_2(i)]_{1 \times n}$ via the main channel that is error-free. He recovers the original message $M(i)$ by using the parity check matrix of the code as shown in Algorithm 17.

Algorithm 17 Legitimate Receiver's Decoder

Require: $[C_2(i)]_{1 \times n}$ ▷ The transmitted vector from Alice
Require: $[H]_{(n-k) \times n}$ ▷ The parity check matrix of $[n, k, d]$ code
1: **Generate** $[H]_{n \times (n-k)}^T$ ▷ The parity check transpose matrix of $[n, k, d]$ code
2: $S(i) \leftarrow C_2(i) \times H^T$ ▷ Bob computes $[S(i)]_{1 \times m}$
3: $\hat{M}(i) \leftarrow S(i)$, $[M(i)] = [\hat{M}(i)]$ ▷ Bob recovers the original message $[M(i)]_{1 \times m}$
4: **return** $[M(i)]_{1 \times m}$

From Algorithm 17, we can prove that $C_2(i) \times H^T = M(i)$ as follows:

$$S(i) = C_2(i) \times H^T \longrightarrow S(i) = (C_1(i) + E(i)) \times H^T$$

$$S(i) = C_1(i) \times H^T + E(i) \times H^T$$

$$S(i) = DR(i) \times G \times H^T + E(i) \times H^T$$

Since G and H are orthogonal ($G \times H^T = 0$), then

$$S(i) = E(i) \times H^T = M(i).$$

The syndrome formed from $E(i) \times H^T$, because of the k leading zeros of $E(i)$ is simply $M(i)$ multiplied by the identity sub-matrix of H^T which produces $M(i)$. This reduces the complexity by 91% (for example using the (23, 12, 7) binary Golay code) compared with the traditional syndrome coding which requires an error pattern-syndrome look up table to be stored. The complexity of achieving the syndrome coding scheme (Encoding and Decoding Algorithms) can be measured

as follows:

$$\text{Complexity} = \frac{T_1 - T_2}{T_1} \quad (4.3)$$

Where T_1 represents the time required for execution of the traditional syndrome coding(requires look up table) and T_2 represents the time required for execution of the new syndrome coding(look up table is unnecessary).

All simulations have been done using a PC machine using Ubuntu 12.04 LTS operating system:

Intel [®] Core TM i5-2500 CPU @ 3.30 GHz ×4 , RAM: 32GB.

2. Eavesdropper's Decoder

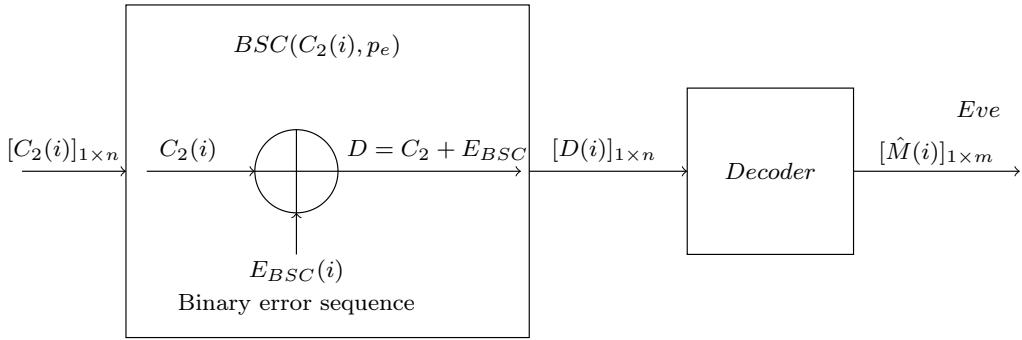


Figure 4.3: Block Diagram of the *BSC* channel and Eavesdropper's Decoder

The block diagram of the *BSC* channel and eavesdropper's decoder is shown in Fig. 4.3. Eve receives a corrupted vector $D(i)$ instead of the transmitted vector $C_2(i)$ as a result of passing through the *BSC* which adds additional errors $[E_{BSC}]_{1 \times n}$: $[D(i)]_{1 \times n} = [C_2(i)]_{1 \times n} + [E_{BSC}(i)]_{1 \times n}$,

Where $[E_{BSC}(i)]_{1 \times n}$ is a random binary error vector which depends on the crossover probability p_e of *BSC*. Assuming Eve uses the same type of decoder that has been used by Bob, the following steps explains how she gets the estimated message $\hat{M}(i)$ from the corrupted vector $D(i)$:

Algorithm 18 Eavesdropper's Decoder

Require: $D(i), H^T$	▷ Eve using H^T of $[n, k, d]$ code
1: $S_{Eve}(i) \leftarrow D(i) \times H^T$	▷ Eve computes $[S_{Eve}(i)]_{1 \times m}$
2: $\hat{M}(i) \leftarrow S_{Eve}(i), M(i) \neq \hat{M}(i)$	▷ Eve recovers an estimate of the message $\hat{M}(i)$
3: return $[\hat{M}(i)]_{1 \times m}$	

From Algorithm 18, Eve estimates $\hat{M}(i)$ as follows:

$$S_{Eve}(i) = D(i) \times H^T$$

$$S_{Eve}(i) = [C_2(i) + E_{BSC}(i)] \times H^T$$

$$S_{Eve}(i) = C_2(i) \times H^T + E_{BSC}(i) \times H^T$$

$$S_{Eve}(i) = [C_1(i) + E(i)] \times H^T + E_{BSC}(i) \times H^T$$

$$S_{Eve}(i) = E(i) \times H^T + E_{BSC}(i) \times H^T = \hat{M}(i)$$

$$\hat{M}(i) = S_{Eve}(i) = M(i) + S_e(i)$$

4.4 Evaluation of the equivocation rate achieved by syndrome coding

The secrecy realised by syndrome coding is measured by the eavesdropper decoder output equivocation, $H(M(i)|\hat{M}(i))$:

$$\begin{aligned} H(M(i)|\hat{M}(i)) &= H(M(i), \hat{M}(i)) - H(\hat{M}(i)) \\ &= H(M(i)) + H(\hat{M}(i)|M(i)) - H(\hat{M}(i)) \\ &= H(M(i)) - H(M(i) + S_e(i)) + H(M(i) + S_e(i)|M(i)) \\ &= m - m + 0 + H(S_e(i)|M(i)) \end{aligned} \tag{4.4}$$

$$H(M(i)|\hat{M}(i)) = H(S_e(i)) \tag{4.5}$$

$$H(M(i)|\hat{M}(i)) = - \sum_{i=0}^{2^m-1} p(S_e(i)) \times \log_2 p(S_e(i)) \tag{4.6}$$

where $H(S_e(i))$ is the entropy of $S_e(i)$. The simplifications in equations 4.4 and 4.5 are due to $M(i)$ being uniformly distributed and independent of $S_e(i)$. The equivocation is calculated after deriving the probability mass function of the syndromes due to errors from the *BSC*, $p(S_e(i))$ and is a function of the code being used through the parity check matrix of the code.

4.5 Exhaustive evaluation of new best known equivocation codes for syndrome coding with the example case of 15 parity bits

A code design technique to extend the binary linear $[n, k]$ code to a $[n + 1, k + 1]$ code to produce best known equivocation codes(BE_qC) for syndrome coding in wiretap channel is presented, which have better secrecy than the best error correcting codes. Code examples are given for the case where the number of parity bits of the code is equal to 15($m = 15$) where $m = n - k$.

The code construction method for obtaining good equivocation codes is based on extensions of the parity check matrix of a set of good equivocation codes of shorter length. It is also shown that syndrome coding can be implemented without the traditional syndrome look up table, enabling any length codes to be used. An efficient recursive method to calculate the equivocation rate of any linear, binary code when used in syndrome coding for the binary symmetric channel(BSC) is also presented.

The design results show that the best known equivocation codes(BE_qC) that are produced have better equivocation rates for the syndrome coding compared to all previously published codes, including the best known codes(BKC) compiled and published by Grassl [2].

4.5.1 Recursive Evaluation of the syndrome probability distribution

The code construction technique that produces codes with a good equivocation rate is based on the realisation that the syndrome probability mass function (*pmf*) of a new extended code is a function of the probability mass function of the original code and good equivocation codes produce good extended codes. For Eve, there are 2^n possible error patterns, $e(i)$, occur for each transmitted n -bit vector. These error patterns occur

with probability:

$$p(e(i)) = p_e^{w(i)} \times (1 - p_e)^{n-w(i)} \quad (4.7)$$

where $w(i)$ is the weight of $e(i)$. Each error pattern results in one of the 2^m syndromes being produced.

$$S_e(i) = e(i) \times H^T \quad (4.8)$$

As the code is linear, for each syndrome, S_j , there are 2^k error patterns that produce the same syndrome and the probability of each syndrome due to all possible error patterns is given by:

$$p(S_j) = \sum_{i=0}^{2^n-1} p(e(i)) \times \delta(S_e(i) - S_j) \quad (4.9)$$

where $\delta()$ is the Dirac function and

$$H(S_e(i)) = - \sum_{j=0}^{2^m-1} p(S_j) \times \log_2[p(S_j)] \quad (4.10)$$

This method for evaluating the equivocation works well for short codes ($n < 40$), but for the long codes it is impracticable because it involves the evaluation of 2^n error patterns. Due to the limitation of this method, the probability distribution of the syndromes may be determined recursively, this method enables the reduction of the number of terms from 2^n to 2^m as shown in the following theorem. A worked example of this procedure is given in page 100.

Theorem 1. *The probability mass function (pmf) of S_j for $j=0$ to 2^m-1 can be defined as $p(S_j) = \beta(j)$ where $\beta(j)$ are coefficients of the probability generating function using the Z transform, denoted as $p_z(S)$ and $p_z(S)$ only depends on the columns of the parity check matrix H and the probability error of the binary symmetric channel p_e .*

$$p_z(S) = \sum_{j=0}^{2^m-1} \beta(j) Z^j = \prod_{i=0}^{n-1} ((1 - p_e) + p_e \times Z^{b_i}) \quad (4.11)$$

where b_i are the integers representations of the columns of H and exponent sums of powers of Z are added modulo 2.

Proof. Any error pattern may be represented as a sum of single bit error events:

$$e(i) = [e_1 \ e_2 \ \dots \ e_n]$$

$$e(i) = [e_1 \ 0 \ \dots \ 0] + [0 \ e_2 \ \dots \ 0] + \dots + [0 \ 0 \ \dots \ e_n]$$

where $e_i=1$ with probability p_e and $e_i=0$ with probability $1 - p_e$. The linearity of the syndrome coding scheme means that the syndrome resulting from any error pattern is the linear sum of the syndromes for each bit error position:

$$S_e(i) = e(i) \times H^T = [e_1 \ e_2 \ \dots \ e_n] \times H^T \quad (4.12)$$

$$S_e(i) = b_1\delta(e_1 - 1) \oplus b_2\delta(e_2 - 1) \dots \oplus b_n\delta(e_n - 1) \quad (4.13)$$

where \oplus denotes the modulo 2 sum. Since the probabilities of e_1, e_2, \dots, e_n are independent, the probability of $S_e(i)$ is the product of the probabilities of n separate error events. By adding the coefficients of the same powers of Z results in the coefficients, β_j , the number of terms can be reduced from 2^n to 2^m . A classical result in statistics is that the *pmf* of a sum of random variables is given by the convolution of the *pmfs* of each variable. The Z transform carries out this convolution. \square

If the columns of H of the shortened code of length r are taken from $i = 0$ to $r - 1$ and the *pmf* generating function of the shortened code is represented as $p_z(S_r)$ then

$$p_z(S_r) = \prod_{i=0}^{r-1} [(1 - p_e) + p_e \times Z^{b_i}] \quad (4.14)$$

$$p_z(S_r) = [(1 - p_e) + p_e \times Z^{b_0}] \times [(1 - p_e) + p_e \times Z^{b_1}] \dots \times [(1 - p_e) + p_e \times Z^{b_{r-1}}]$$

$$p_z(S_r) = (1 - p_e)^2 + p_e(1 - p_e) \times Z^{b_1} + p_e(1 - p_e) \times Z^{b_0} + p_e^2 \times Z^{b_0 \oplus b_1} + \dots$$

Now, we can extend the length of the original code from r to $r + 1$ by adding one column to its parity check matrix H , so the *pmf* generating function of the extended code $r + 1$ can be represented as

$$p_z(S_{r+1}) = \prod_{i=0}^r [(1 - p_e) + p_e \times Z^{b_i}] = p_z(S_r)[(1 - p_e) + p_e \times Z^{b_r}] \quad (4.15)$$

Denoting the $\beta(j)$ coefficients of the original code of length r , as $\beta_r(j)$ then equation 4.14 can be re-written as follows:

$$p_z(S_r) = \sum_{j=0}^{2^m-1} \beta_r(j) Z^j \quad (4.16)$$

and for extended code $r + 1$

$$p_z(S_{r+1}) = (1 - p_e) \sum_{j=0}^{2^m-1} \beta_r(j) Z^j + p_e \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_r} \quad (4.17)$$

which simplifies to

$$p_z(S_{r+1}) = (1 - p_e)p_z(S_r) + p_e \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_r} \quad (4.18)$$

Adding together the coefficients of the same powers of Z in the coefficients, $\beta_r(j)$ to obtain $\beta_{r+1}(j)$, simplifies the above equation to

$$p_z(S_{r+1}) = \sum_{j=0}^{2^m-1} \beta_{r+1}(j) Z^j \quad (4.19)$$

From equation 4.18, it is clear that the syndrome pmf of the new code of length $r + 1$ is equal to the syndrome pmf of the original code of length r weighted by $1 - p_e$ plus a permuted syndrome pmf of the original code of length r , weighted by p_e . The permutation arises from the results of the modulo 2 additions $j \oplus b_r$. Therefore the syndrome pmf of the code can be calculated recursively, starting with the generating function $p_z(S_1)$, determining $p_z(S_2)$ then $p_z(S_3)$ through to $p_z(S_n)$.

The syndrome pmf of each $[n, k, d]$ code of length r is stored and the syndrome pmf for each extended code of length $r + 1$ is determined using the equation 4.18 which makes for a fast algorithm. This leads to the conclusion that codes with good equivocation will produce good equivocation codes when extended in length.

Worked example

The parity check matrix of the [7, 4, 3] Hamming code is:

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Which in systematic packed integer representation is: $\mathbf{H}=[1,2,4,3,5,6,7]$

$$n = 7, k = 4, m = n - k = 3$$

The probability mass function (*pmf*) of the syndrome (S_j) for $j = 0$ to $2^m - 1$ of the [7, 4, 3] Hamming code can be obtained recursively, starting with the generating function $p_z(S_0)$ determining $p_z(S_1)$ then $p_z(S_2)$ through to $p_z(S_6)$ as follows:

$$1) p_z(S_0) = (1 - p_e) + p_e \times Z^{b_0} = (1 - p_e) + p_e \times Z^1$$

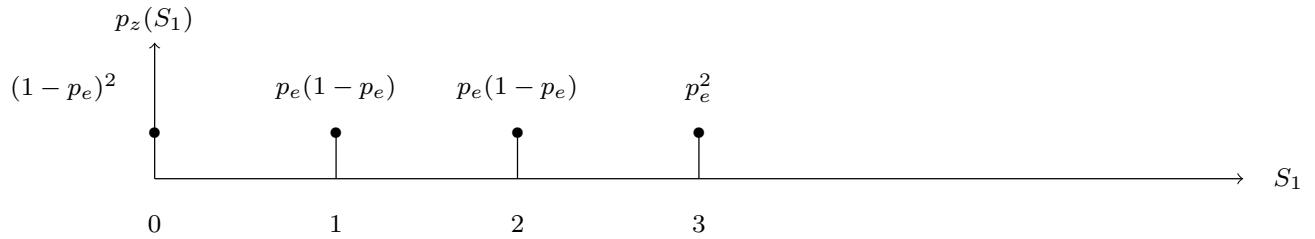
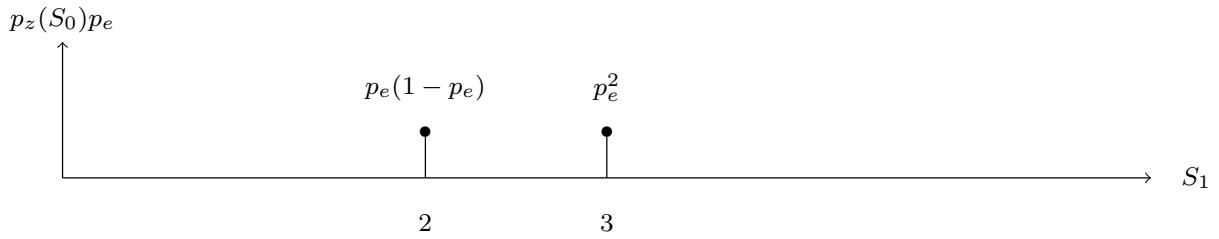


$$2) p_z(S_1) = p_z(S_0) \times [(1 - p_e) + p_e \times Z^{b_1}] = p_z(S_0) \times [(1 - p_e) + p_e \times Z^2]$$

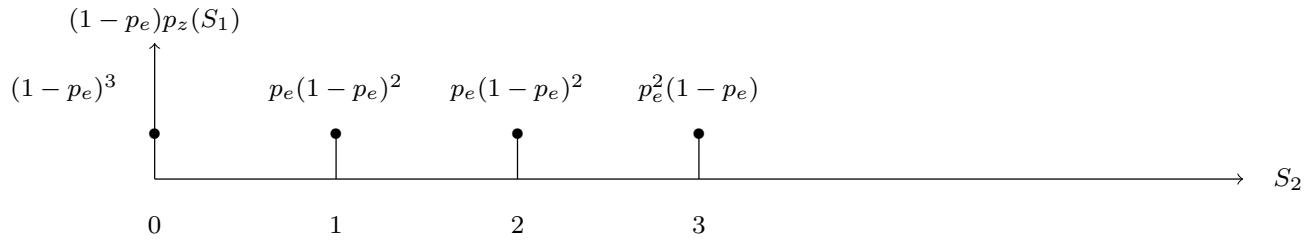


As previously mentioned, the permutation arises from the results of the modulo 2 additions $j \oplus b_i$.

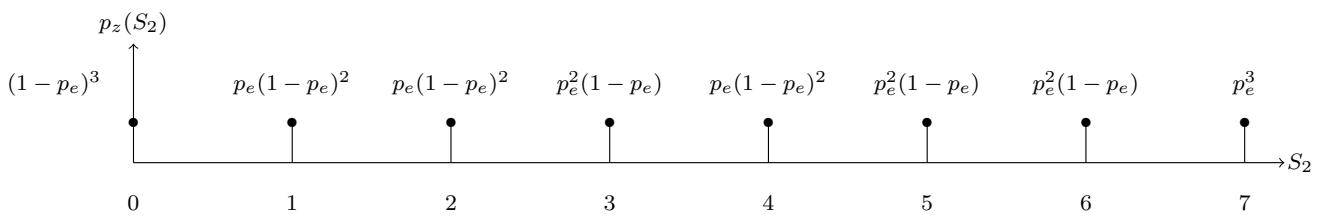
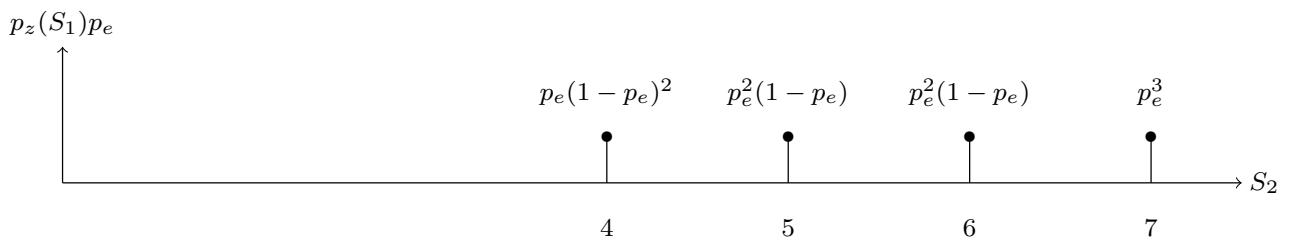
$$j \oplus b_1 = j \oplus 2, 0 \oplus 2 = 2, 1 \oplus 2 = 3$$



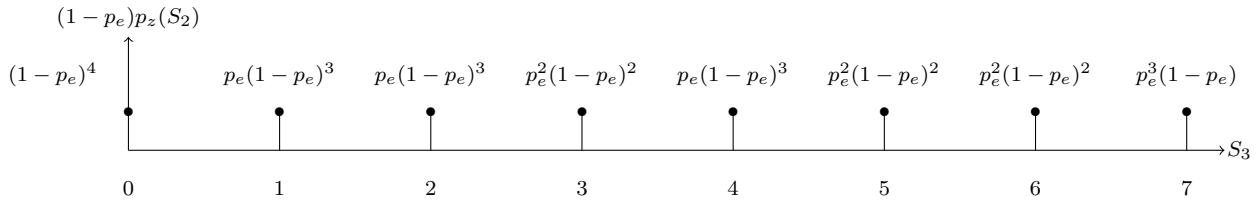
$$3) p_z(S_2) = p_z(S_1) \times [(1 - p_e) + p_e \times Z^{b_2}] = p_z(S_1) \times [(1 - p_e) + p_e \times Z^4]$$



$$j \oplus b_2 = j \oplus 4, 0 \oplus 4 = 4, 1 \oplus 4 = 5, 2 \oplus 4 = 6, 3 \oplus 4 = 7$$

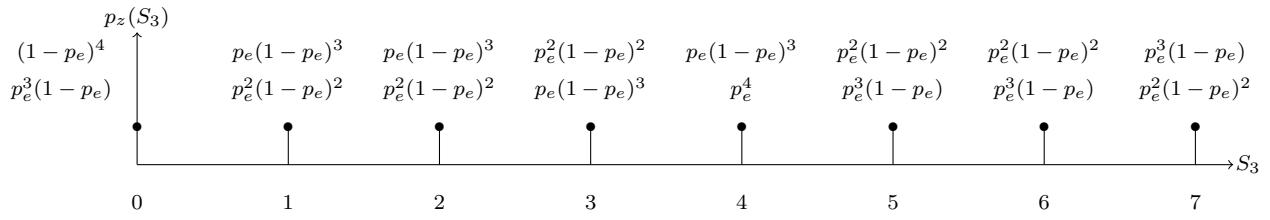
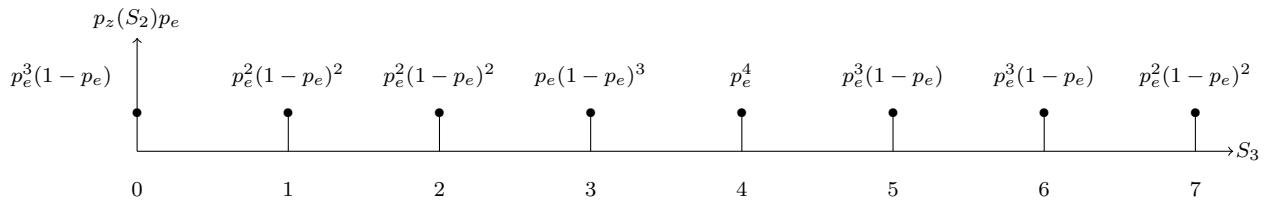


$$4) p_z(S_3) = p_z(S_2) \times [(1 - p_e) + p_e \times Z^{b_3}] = p_z(S_2) \times [(1 - p_e) + p_e \times Z^3]$$

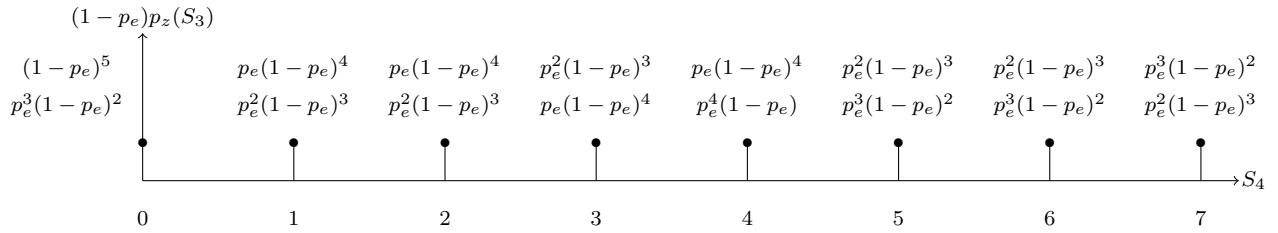


$$j \oplus b_3 = j \oplus 3, 0 \oplus 3 = 3, 1 \oplus 3 = 2, 2 \oplus 3 = 1, 3 \oplus 3 = 0, 4 \oplus 3 = 7, 5 \oplus 3 = 6,$$

$$6 \oplus 3 = 5, 7 \oplus 3 = 4$$

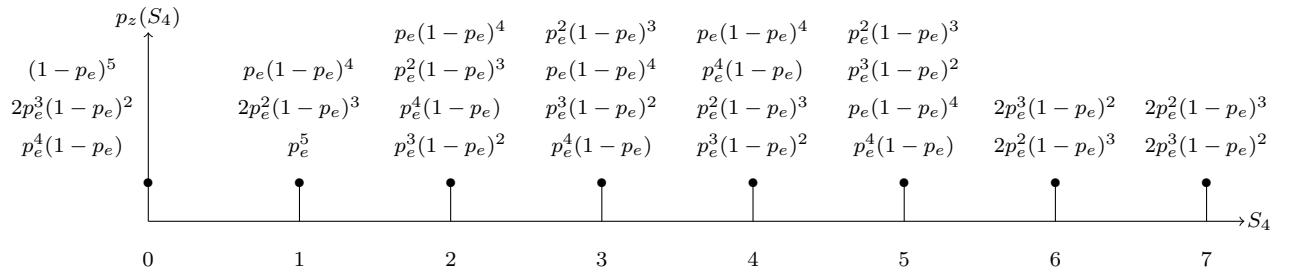
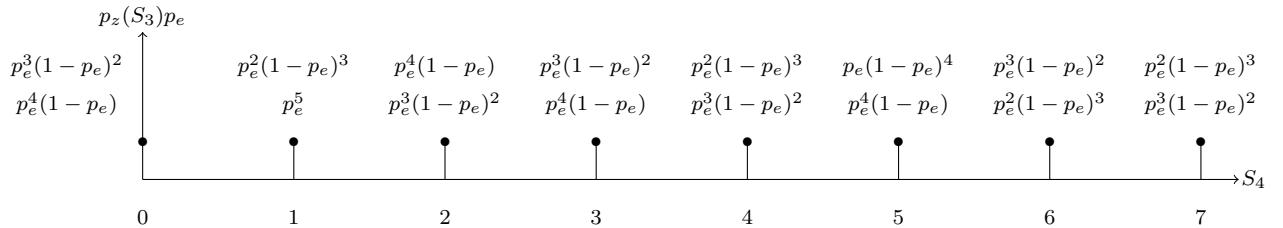


$$5) p_z(S_4) = p_z(S_3) \times [(1 - p_e) + p_e \times Z^{b_4}] = p_z(S_3) \times [(1 - p_e) + p_e \times Z^5]$$

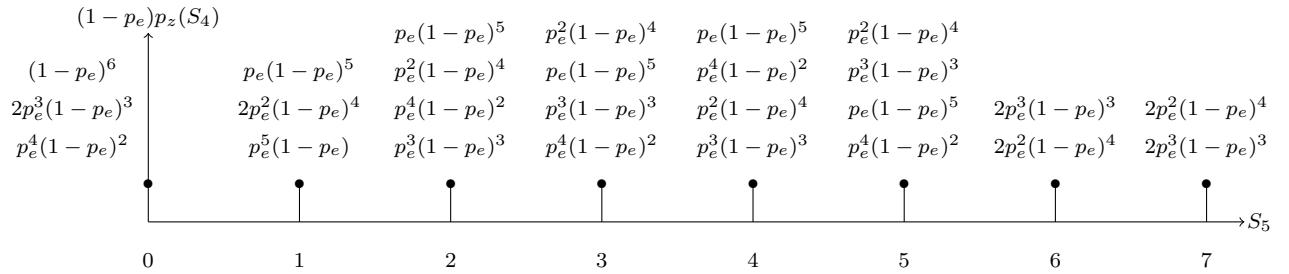


$$j \oplus b_4 = j \oplus 5, 0 \oplus 5 = 5, 1 \oplus 5 = 4, 2 \oplus 5 = 7, 3 \oplus 5 = 6, 4 \oplus 5 = 1, 5 \oplus 5 = 0,$$

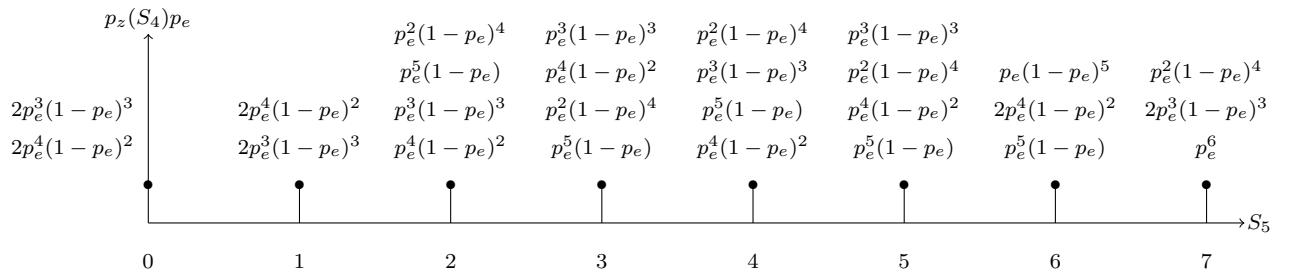
$$6 \oplus 5 = 3, 7 \oplus 5 = 2$$

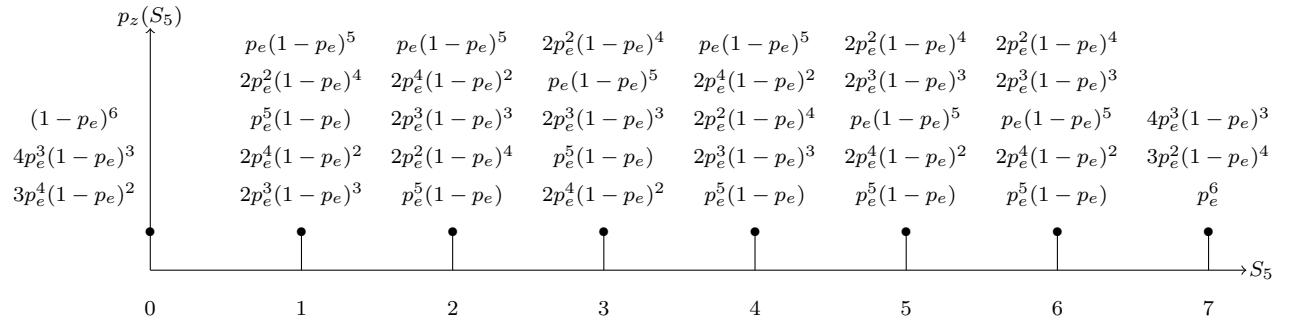


$$6) p_z(S_5) = p_z(S_4) \times [(1-p_e) + p_e \times Z^{b_5}] = p_z(S_4) \times [(1-p_e) + p_e \times Z^6] j \oplus b_5 = j \oplus 6$$

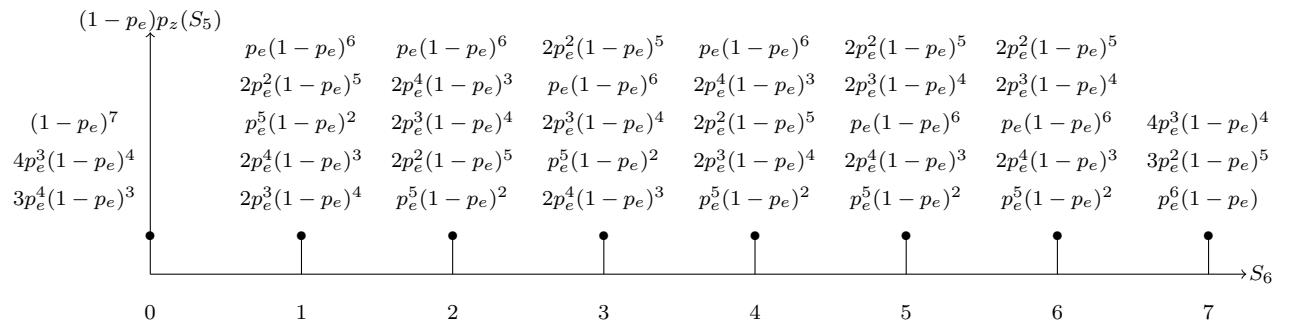


$$, 0 \oplus 6 = 6 , 1 \oplus 6 = 7 , 2 \oplus 6 = 4 , 3 \oplus 6 = 5 , 4 \oplus 6 = 2 , 5 \oplus 6 = 3 , 6 \oplus 6 = 0 , 7 \oplus 6 = 1$$



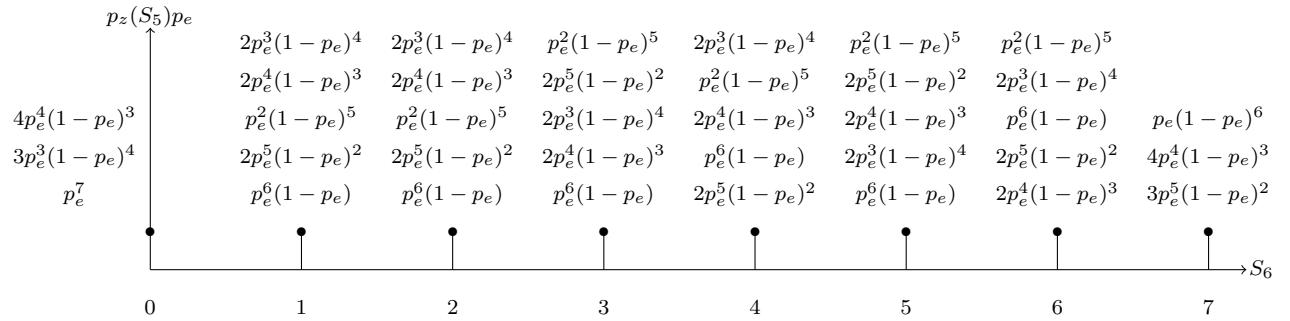


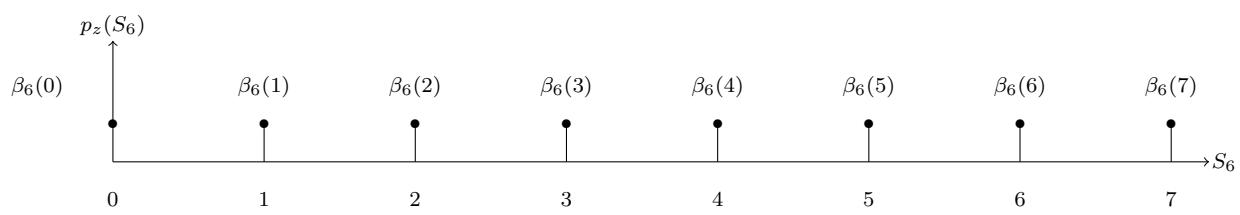
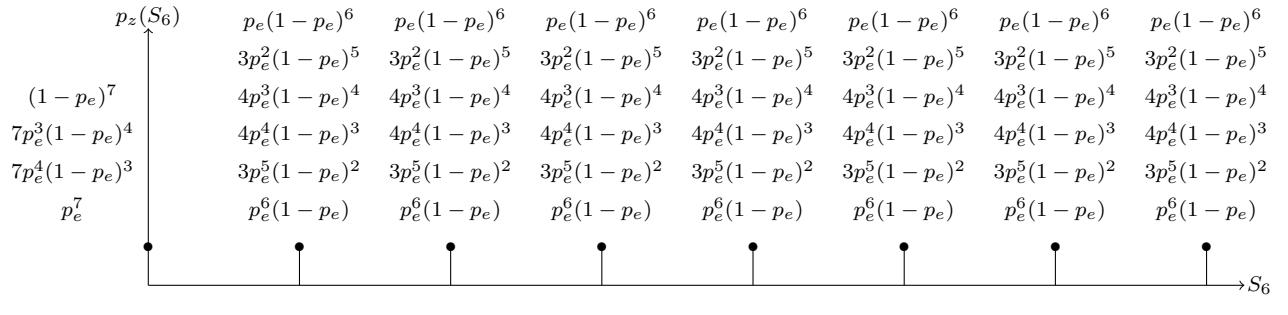
$$7) p_z(S_6) = p_z(S_5) \times [(1 - p_e) + p_e \times Z^{b_6}] = p_z(S_5) \times [(1 - p_e) + p_e \times Z^7]$$



$$j \oplus b_6 = j \oplus 7, 0 \oplus 7 = 7, 1 \oplus 7 = 6, 2 \oplus 7 = 5, 3 \oplus 7 = 4, 4 \oplus 7 = 3, 5 \oplus 7 = 2,$$

$$6 \oplus 7 = 1, 7 \oplus 7 = 0$$





From equation 4.16:

$$p_z(S_r) = \sum_{j=0}^{2^m-1} \beta_r(j) Z^j$$

$$p_z(S_6) = \sum_{j=0}^7 \beta_6(j) Z^j = \beta_6(0) + \beta_6(1)Z^1 + \beta_6(2)Z^2 + \beta_6(3)Z^3 + \beta_6(4)Z^4 + \beta_6(5)Z^5 + \beta_6(6)Z^6 + \beta_6(7)Z^7$$

$$\text{Where: } \beta_6(0) = (1-p_e)^7 + 7p_e^3(1-p_e)^4 + 7p_e^4(1-p_e)^3 + p_e^7$$

$$\beta_6(1) = p_e(1-p_e)^6 + 3p_e^2(1-p_e)^5 + 4p_e^3(1-p_e)^4 + 4p_e^4(1-p_e)^3 + 3p_e^5(1-p_e)^2 + p_e^6(1-p_e)$$

$$\beta_6(7) = \beta_6(6) = \beta_6(5) = \beta_6(4) = \beta_6(3) = \beta_6(2) = \beta_6(1)$$

The normalised equivocation rate (*Eqv. rate*) of the [7, 4, 3] Hamming code can be calculated for the binary symmetric eavesdropper channel for a given error probability (p_e) as follows:

$$\text{Eqv. rate} = \frac{- \sum_{j=0}^{2^m-1} \beta_r(j) \times \log_2 \beta_r(j)}{m} \quad (4.20)$$

$$\begin{aligned} \text{Eqv. rate} &= - \sum_{j=0}^7 \beta_6(j) \times \log_2 \beta_6(j)/3 \\ &= -[(\beta_6(0) \times \frac{\log \beta_6(0)}{\log 2}) + (\beta_6(1) \times \frac{\log \beta_6(1)}{\log 2}) + \dots + (\beta_6(7) \times \frac{\log \beta_6(7)}{\log 2})]/3 \end{aligned}$$

$$\text{Eqv. rate} = -[(\beta_6(0) \times \frac{\log \beta_6(0)}{\log 2}) + 7 \times (\beta_6(1) \times \frac{\log \beta_6(1)}{\log 2})]/3$$

Table 4.1 shows the normalised equivocation rate(*Eqv. rate*) of the [7, 4, 3] Hamming code at some specific values of p_e .

p_e	<i>Eqv. rate</i>
0.02	0.310828
0.04	0.501668
0.05	0.575724
0.06	0.63921
0.08	0.740937
0.1	0.816495
0.15	0.928895

Table 4.1: Equivocation rate of the [7, 4, 3] Hamming code

4.5.2 Code design technique

To produce a best known equivocation code the *pmf* of the syndromes should be as uniform as possible. Since the eavesdropper channel is a binary symmetric channel, for low values of p_e the equivocation is dominated by error patterns of low weight. To produce best known equivocation codes, we must take into account the following observations:

1. If the error pattern has low weight, then the probability of the error events is high. If each error pattern produces different syndrome sums, then this makes the pmf of the syndromes become more uniform.
2. By using the systematic format of the parity check matrix H , the packed integers of any information bit cannot have a weight less than $d - 1$, where d is the minimum Hamming distance of the code. Otherwise the codeword formed from that information bit alone will have weight less than d .
3. If any column of the parity check matrix H is repeated, a weight 2 error event will produce a zero syndrome, that leads to a non uniform pmf of the syndrome.

Algorithm 19 shows how to extend an $[n, k]$ code into $[n + 1, k + 1]$ code by adding the best column to the original parity check matrix H of the $[n, k]$ code.

The steps of the algorithm can be simplified as follows:

1. Calculate the syndrome *pmf* of the original code $[n, k]$ from equation 4.16.
2. Represent the parity check matrix H of the $[n, k]$ code in the systematic packed integer form: $H = [1, 2, 4, \dots, 2^{m-1}, b_m, \dots, b_{n-1}]$
3. Extend H with one integer (b_r) by generating randomly all possible integers between 3 and $2^m - 1$ with the constraint that there are no repeated integers included in the original H . This ensures that the minimum Hamming distance of each extended code is at least 3.
4. Eliminate all equivalent codes and evaluate the equivocation rate for the binary symmetric eavesdropper channel for a given p_e for each remaining code.

5. Rank the inequivalent codes by their equivocation rate in descending order, and select a best codes subset. These codes are used as the initial input for the next extension round.

Algorithm 19 Code Design Technique 1

```

Require:  $p_z(S_r)$                                 ▷ Syndrome  $pmf$  of  $[n, k]$  code
Require:  $H$                                     ▷ Systematic format of  $H$  of  $[n, k]$  code
Require:  $b[i]$                                 ▷ Integer sequence(columns) of  $H$ 
Require:  $(n, k, m, p_e)$                       ▷ Code parameters and error probability of  $BSC$ 
Require:  $C_{in}$                                 ▷ Initial inequivalent codes of the highest equivocation rate
1: Generate  $(b_r)$                                 ▷ Generating randomly integers between 3 and  $2^m - 1$ 
Ensure:  $(b_r) \neq b[i]$                                 ▷ Ensure no repeated columns
2:  $p_{z1}(S_r) \leftarrow \frac{(1 - p_e)p_z(S_r)}{2^{m-1}}$ 
3:  $p_{z2}(S_r) \leftarrow p_e \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_r}$ 
4:  $p_z(S_{r+1}) \leftarrow p_{z1}(S_r) + p_{z2}(S_r)$           ▷ Apply equations 4.18 and 4.19
5:  $Eqv. \leftarrow - \sum_{j=0}^{2^m-1} \beta_{r+1}(j) \times \log_2(\beta_{r+1}(j))$     ▷ Calculate the equivocation rate of
   [n + 1, k + 1] code
6:  $Eqv. rate \leftarrow Eqv./m$                       ▷ Calculate the normalised equivocation rate
7: return  $(b_r), Eqv. rate, C_{out}$       ▷ Extended inequivalent codes, which are ranked by
   equivocation rate in descending order

```

4.5.3 Results

By using the code design technique above, the best known equivocation codes have been determined for $m = 15$. The codes are listed in Appendix A in the packed integer format, which provide at least 70% secrecy. The minimum Hamming distance (d) and the equivocation rate($Eqv. rate$) for a BSC error probability of $p_e = 0.05$ is given for each code. The equivocation rates of the corresponding best error correcting codes previously published, the (BKC) codes compiled by Grassl [2] with the same n and m are also given in in Appendix A (in parentheses).

The results show that significant improvements have been achieved on the equivocation rate for the best known equivocation codes compared with best known codes.

Fig. 4.4 shows the equivocation rate($Eqv. rate$) as a function of probability of error p_e of best equivocation and best known codes for $n = 82$ at different values of p_e . It shows

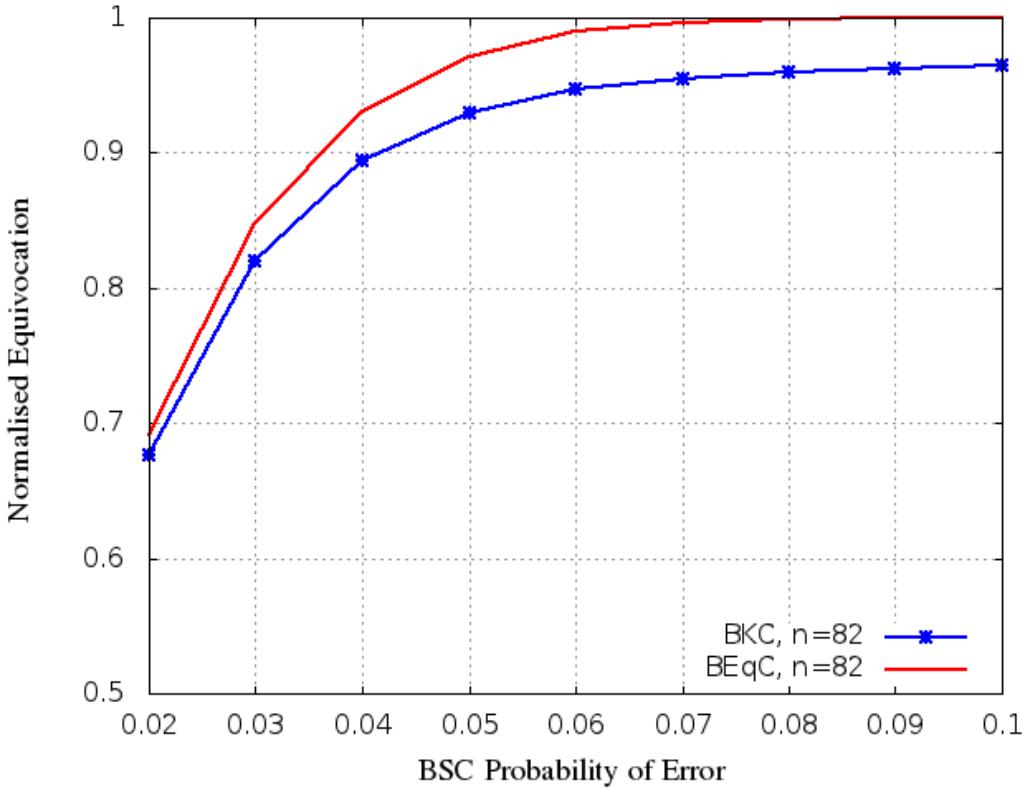


Figure 4.4: Equivocation rate $Eqv.rate$ vs. p_e of best known equivocation (BE_qC) and best known (BKC) codes for $n = 82$

that the equivocation rate of (BE_qCs) has been increased by a large margin compared with ($BKCs$) not only for $p_e = 0.05$ but also for other values of p_e . These results have been published in [64].

4.6 Implementation and Construction of best known equivocation codes with highest minimum Hamming distance for syndrome coding

Several researchers have analysed the syndrome coding scheme as a function of the code used in order to increase the communication security, Zhang *et al.* [51] produced some best known equivocation codes for the syndrome coding scheme. The best known equivocation codes have been determined for a given number of parity bits m , of the code as follows:

$$(m = 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 18, 20, 22, 24, 26).$$

Al-Hassan *et al.* [64] also determined the new best known equivocation code for syndrome coding when the parity bit of the code is equal ($m = 15$). Usually, the best known equivocation codes obtained in [51, 64] have a respectable minimum Hamming distance, but are sometimes not as good as the best known code with the same parameters. An $[n, k]$ linear code C is said to be a best known code (BKC) if C has the highest minimum distance among all known $[n, k]$ linear codes, the tables of ($BKCs$) have been published by Grassl [2] in the form of tables of lower and upper bounds of (d_{min}).

In this section, a new code design technique which produces best known equivocation codes(BE_qC) with highest minimum Hamming distance(d_{min_H}) for syndrome coding is presented. The producing codes have better secrecy than the best known error correcting codes(BKC).

The best known equivocation codes of highest minimum Hamming distance have been determined by using a combination of a code design technique based on extensions of the parity check matrix from an optimal set of good equivocation codes of highest minimum distance coupled with a technique of determining the highest minimum distance (d_{min_H}) of these extended codes. The equivocation rate for the binary symmetric channel(BSC) and any linear code has been calculated by using the recursive evaluation of the syndrome probability distribution which are described in the section 4.5.1.

Candidate codes are constructed with a given d_{min_H} by constraining the dependencies of columns of the codes parity check matrix. The candidate codes are then ranked according to equivocation rate.

Code examples are presented for a given number of parity bits of the code ($m = 7, 11, 12$), that demonstrate the equivocation rate of these best known equivocation codes (BE_qC) exceeds by a large margin the equivocation rate of the equivalent best known error correcting codes(BKC), published by Grassl [2].

4.6.1 Calculation of the minimum distance of a linear code

The Minimum distance of a linear code is an important parameter that provides the capability of detecting and correcting errors by the code. In the past, many researchers have studied the minimum distance (d_{min}) of linear codes. Some results on calculating distance were presented in [65, 66].

If C is an $[n, k, d]$ linear code with parity check matrix H , then the minimum distance (d_{min}) of C is equal to the smallest number of columns of H which are linearly dependent. That is, all combinations of $(d_{min} - 1)$ columns are linearly independent, so there is some set of (d_{min}) columns which are linearly dependent [13, 17].

There are three cases of d_{min} :

1. If H has a column of all zeros, then $d_{min} = 1$.
2. If H has two identical columns, then $d_{min} \leq 2$.
3. For binary codes, if all columns are distinct and non-zero, then $d_{min} \geq 3$.

Worked example

The parity check matrix of the [7, 4, 3] Hamming code is:

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Which in systematic packed integer representation is: $\mathbf{H}=[1,2,4,3,5,6,7]$

All columns of the parity check matrix are distinct and non-zero, therefore it is expected that $d_{min} \geq 3$.

Now, the process starts to check whether all three columns of H are linearly dependent or not by adding any three columns within H . If the result of the addition process of any three columns is equal to zero then $d_{min} = 3$. If the result of addition process not equal to zero, the process starts again to check all four columns of H are linearly dependent or not and this process will continue until to be obtained the smallest number of columns of H which are linearly dependent.

It is clear from the form of the parity check matrix that the first, second and forth columns are linearly dependent, so $d_{min} = 3$.

4.6.2 Code design technique

To produce a best known equivocation code the *pmf* of the syndromes should be as uniform as possible. The following code design algorithm shows how to extend an $[n, k]$ code into $[n+1, k+1]$ by adding the best column to the original parity check matrix H of the $[n, k]$ code. We must take into consideration that the value of the minimum distance (d_{min}) of the $[n+1, k+1]$ code should be equal to the minimum distance (d_{min_H}) of the $[n+1, k+1]$ *BKC* code because the best known $[n, k]$ code (*BKC*) has the highest minimum distance among all known $[n, k]$ linear codes.

Algorithm 20 Code Design Technique 2

Require: $p_z(S_r)$ ▷ Syndrome pmf of $[n, k]$ code
Require: H ▷ Systematic format of H of $[n, k]$ code
Require: $dmin_H$ ▷ Highest $dmin$ of $[n + 1, k + 1]$ BKC code
Require: $b[i]$ ▷ Integer sequence(columns) of H
Require: (n, k, m, p_e) ▷ Code parameters and error probability of BSC
Require: C_{in} ▷ Initial inequivalent codes of the highest equivocation rate

1: **Generate** (b_r) ▷ Generating randomly integers between 3 and $2^m - 1$
Ensure: $(b_r) \neq b[i]$ ▷ Ensure no repeated columns
Require: $dmin$ ▷ $dmin$ of $[n + 1, k + 1]$ code
Ensure: $dmin = dmin_H$
2: $p_{z1}(S_r) \leftarrow \sum_{j=0}^{2^m-1} (1 - p_e)p_z(S_r)$
3: $p_{z2}(S_r) \leftarrow p_e \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_r}$
4: $p_z(S_{r+1}) \leftarrow p_{z1}(S_r) + p_{z2}(S_r)$ ▷ Apply equations 4.18 and 4.19
5: $Eqv. \leftarrow - \sum_{j=0}^{2^m-1} \beta_{r+1}(j) \times \log_2(\beta_{r+1}(j))$ ▷ Calculate the equivocation rate of
 $[n + 1, k + 1]$ code
6: $Eqv. rate \leftarrow Eqv./m$ ▷ Calculate the normalised equivocation rate
7: **return** $(b_r, dmin, Eqv. rate, C_{out})$ ▷ Extended inequivalent codes of highest $dmin$, which are ranked by equivocation rate in descending order

The steps of the algorithm can be simplified as follows:

1. Calculate the syndrome pmf of the original code $[n, k]$ from equation 4.16.
 2. Represent the parity check matrix H of the $[n, k]$ code in the systematic packed integer form: $H = [1, 2, 4, \dots, 2^{m-1}, b_m, \dots, b_{n-1}]$
 3. Extend H with one integer (b_r) by generating randomly all possible integers between 3 and $2^m - 1$ with the constraint that there are no repeated integers included

in the original H . This ensures that the minimum Hamming distance of each extended code is at least 3.

4. Calculate the highest minimum distance(d_{min_H}) of the $[n + 1, k + 1]$ BKC code from Magma Software.
5. Calculate the minimum distance (d_{min}) of the extended code $[n + 1, k + 1]$ and ensure that $d_{min} = d_{min_H}$.
6. Eliminate all equivalent codes and evaluate the equivocation rate for the binary symmetric eavesdropper channel for a given p_e for each remaining code.
7. Rank the inequivalent codes by their equivocation rate in descending order, and select a best codes subset. These codes are used as the initial input for the next extension round.

4.6.3 Results

Following the Algorithm above, the best known equivocation codes(BE_qC) of the highest minimum distance have been determined in the form of the equivocation rate for various values of n and for a given number of parity bits of the code $m = 7, 11, 12$.

The codes are listed in Appendix B in the packed integer format. The highest minimum Hamming distance (d) and the normalised equivocation rate ($Eqv.rate$) for a BSC error probability of $p_e = 0.05$ is given for each code. The equivocation rates of the corresponding best error correcting codes previously published, the (BKC) codes compiled by Grassl [2] with the same n , m and d are also given in Appendix B (in parentheses). The results show that significant improvements have been achieved on the equivocation rate for the best known equivocation codes(BE_qC) compared with best known codes(BKC). Tables 4.2, 4.3 and 4.4 show the comparison in the normalised equivocation rate ($Eqv.rate$) and minimum Hamming distance (d) between the best known equivocation codes (BE_qC) (**Scheme-3**) determined by the **Algorithm 20** described in this section with the best known correcting codes BKC (**Scheme-1**) compiled by Grassl [2] and the best known

equivocation codes (**Scheme-2**) listed in an on-line database by Zhang [52] for some representative codes. It shows that significant improvements in equivocation rate are obtained for (**scheme-3**) compared with (**scheme-1**) and (**scheme-2**) for all codes having the same parameters (m, n, d) . This results has been published in [67].

Table 4.2: Equivocation rate and minimum distance in syndrome coding for $p_e = 0.05$, $m = 7$

m	n	Eqv.rate(Scheme-1)	d_1	Eqv.rate(Scheme-2)	d_2	Eqv.rate(Scheme-3)	d_3
7	35	0.872739	4	0.907273	3	0.904253	4
7	36	0.888646	4	0.914603	3	0.912049	4
7	37	0.901769	4	0.921343	3	0.919097	4
7	38	0.912693	4	0.927635	3	0.925583	4
7	39	0.921923	4	0.933472	3	0.931601	4
7	40	0.929789	4	0.938810	3	0.937142	4
7	41	0.936808	4	0.943771	3	0.94223	4
7	42	0.942873	4	0.947890	3	0.946878	4
7	43	0.948166	4	0.951763	3	0.951182	4
7	44	0.952803	4	0.955391	3	0.955172	4
7	45	0.956930	4	0.958694	3	0.958863	4
7	46	0.960586	4	0.961848	3	0.962228	4
7	47	0.963848	4	0.964899	3	0.965345	4
7	48	0.966766	4	0.967706	3	0.968205	4
7	49	0.969876	4	0.970311	3	0.970856	4
7	50	0.972635	4	0.972711	3	0.973272	4
7	51	0.975095	4	0.974942	3	0.975516	4
7	52	0.977298	4	0.976991	3	0.977571	4
7	53	0.979291	4	0.979291	4	0.979454	4
7	54	0.981087	4	0.981087	4	0.981189	4
7	55	0.982712	4	0.982713	4	0.982787	4
7	56	0.984185	4	0.984186	4	0.984253	4
7	57	0.985569	4	0.985569	4	0.985599	4
7	58	0.986824	4	0.986824	4	0.986835	4
7	59	0.987968	4	0.987968	4	0.98972	4

Table 4.3: Equivocation rate and minimum distance in syndrome coding for $p_e = 0.05$, $m = 11$

m	n	Eqv.rate(Scheme-1)	d_1	Eqv.rate(Scheme-2)	d_2	Eqv.rate(Scheme-3)	d_3
11	28	0.675605	6	0.676537	5	0.676027	6
11	29	0.693319	6	0.693783	5	0.693663	6
11	30	0.710225	6	0.710315	5	0.710469	6
11	37	0.803255	5	0.806696	4	0.807675	5
11	38	0.815439	5	0.817966	4	0.818806	5
11	39	0.826902	5	0.828624	4	0.829343	5
11	40	0.837627	5	0.838712	4	0.839390	5
11	41	0.847670	5	0.848307	4	0.848951	5
11	48	0.710488	4	0.902358	4	0.903225	4
11	49	0.713286	4	0.908427	4	0.909219	4
11	50	0.715762	4	0.914199	4	0.914902	4
11	51	0.717951	4	0.919611	4	0.920284	4
11	52	0.719886	4	0.924695	4	0.925399	4
11	53	0.721705	4	0.929476	4	0.930200	4
11	54	0.723311	4	0.933982	4	0.934726	4
11	55	0.724755	4	0.938224	4	0.938998	4
11	56	0.726067	4	0.942226	4	0.943023	4
11	57	0.727336	4	0.945978	4	0.946818	4
11	58	0.728485	4	0.949509	4	0.950376	4
11	59	0.729528	4	0.952823	4	0.953600	4
11	60	0.730463	4	0.955931	4	0.956635	4
11	61	0.731332	4	0.958863	4	0.959491	4
11	62	0.732129	4	0.961617	4	0.962176	4
11	63	0.732861	4	0.964190	4	0.964696	4
11	64	0.733520	4	0.966606	4	0.967081	4
11	65	0.734121	4	0.968862	4	0.969326	4
11	66	0.749506	4	0.970987	4	0.971429	4
11	67	0.760622	4	0.972972	4	0.973399	4
11	68	0.761144	4	0.974832	4	0.975245	4
11	69	0.769587	4	0.976573	4	0.976930	4
11	70	0.770032	4	0.978196	4	0.978503	4
11	71	0.770448	4	0.979715	4	0.979978	4
11	72	0.776960	4	0.981130	4	0.981360	4
11	73	0.782101	4	0.982457	4	0.982653	4
11	74	0.786203	4	0.983696	4	0.983861	4
11	75	0.789478	4	0.984852	4	0.984993	4
11	76	0.792140	4	0.985929	4	0.986053	4
11	77	0.794298	4	0.986936	4	0.987039	4
11	78	0.796060	4	0.987877	4	0.987964	4
11	79	0.797503	4	0.988754	4	0.988826	4

Table 4.4: Equivocation rate and minimum distance in syndrome coding for $p_e = 0.05$, $m = 12$

m	n	Eqv.rate(Scheme-1)	d_1	Eqv.rate(Scheme-2)	d_2	Eqv.rate(Scheme-3)	d_3
12	35	0.743336	6	0.747104	5	0.746769	6
12	36	0.758019	6	0.760427	5	0.759141	6
12	37	0.771753	6	0.773184	5	0.772606	6
12	38	0.784715	6	0.785441	5	0.785354	6
12	39	0.796947	6	0.797169	5	0.797463	6
12	40	0.808546	6	0.808547	6	0.808919	6
12	41	0.819566	6	0.819566	6	0.819852	6
12	42	0.830044	6	0.830043	6	0.830230	6
12	43	0.839988	6	0.839985	6	0.840081	6
12	44	0.849432	6	0.849432	6	0.849494	6
12	45	0.858420	6	0.858419	6	0.858427	6
12	55	0.922141	5	0.922450	4	0.922271	5
12	56	0.927042	5	0.927198	4	0.927142	5
12	57	0.931666	5	0.931678	4	0.931739	5
12	58	0.936021	5	0.936020	5	0.936072	5
12	59	0.940113	5	0.940112	5	0.940148	5
12	60	0.943958	5	0.943961	5	0.943986	5
12	61	0.947587	5	0.947587	5	0.947601	5
12	62	0.950994	5	0.950995	5	0.951003	5
12	63	0.954203	5	0.954205	5	0.954207	5
12	64	0.957222	5	0.957220	5	0.957222	5
12	65	0.960062	5	0.960062	5	0.960062	5
12	66	0.710374	4	0.962073	4	0.962563	4
12	67	0.720589	4	0.964528	4	0.964938	4
12	68	0.721090	4	0.966833	4	0.967183	4
12	69	0.728861	4	0.969001	4	0.969306	4
12	70	0.729329	4	0.971041	4	0.971306	4
12	71	0.729730	4	0.972963	4	0.973186	4
12	72	0.730124	4	0.974773	4	0.974954	4
12	73	0.736062	4	0.976457	4	0.976619	4
12	74	0.740797	4	0.978034	4	0.978181	4
12	75	0.744526	4	0.979512	4	0.979648	4
12	76	0.747576	4	0.980897	4	0.981018	4
12	77	0.749986	4	0.982193	4	0.982305	4
12	78	0.751983	4	0.983409	4	0.983510	4
12	79	0.753583	4	0.984547	4	0.984637	4
12	80	0.754920	4	0.985609	4	0.985695	4

4.7 Construction of best known equivocation codes from shorter best equivocation codes by adding two columns to the parity check matrix

Section 4.5 presented a method of extending the parity check matrix of the linear $[n, k]$ code by one best column. In this section, two best columns have been added to the parity check matrix of the linear $[n, k]$ code. The results obtained show that the equivocation rate has been increased when the original best codes have been extended by adding two columns compared to those codes that are produced by the addition of one column in two phases.

4.7.1 Evaluation the probability mass function of the extended $[n + 2, k + 2]$ code

According to section 4.5.1, the length of the original code can be extended from r to $r + 2$ by adding two columns on its parity check matrix. The equation 4.15 was extended to calculate the *pmf* generating function of the extended code $r + 2$ as follows:

$$p_z(S_{r+2}) = \prod_{i=0}^{r+1} [(1 - p_e) + p_e \times Z^{b_i}] \quad (4.21)$$

$$\begin{aligned} p_z(S_{r+2}) &= p_z(s_r)[(1 - p_e) + p_e \times Z^{b_r}][(1 - p_e) + p_e \times Z^{b_{r+1}}] \\ &= (1 - p_e)^2 p_z(S_r) + p_e(1 - p_e) \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_{r+1}} p_z(S_r) + p_e(1 - p_e) Z^{b_r} p_z(S_r) + p_e^2 Z^{(b_r \oplus b_{r+1})} p_z(S_r) \\ p_z(S_{r+2}) &= (1 - p_e)^2 p_z(S_r) + p_e(1 - p_e) \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_{r+1}} + p_e(1 - p_e) \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_r} + \\ &\quad p_e^2 \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus (b_r \oplus b_{r+1})} \end{aligned}$$

Adding together the coefficients of the same powers of Z in the coefficients, $\beta_r(j)$ to obtain $\beta_{r+2}(j)$, simplifies the above equation to

$$p_z(S_{r+2}) = \sum_{j=0}^{2^m-1} \beta_{r+2}(j) Z^j \quad (4.22)$$

This leads to the conclusion that the syndrome p_{zr} of the code can be obtained recursively, starting with the generating function $p_z(S_1)$, determining $p_z(S_2)$ then $p_z(S_3)$ through to $p_z(S_n)$. The syndrome p_{zr} of each $[n, k, d]$ code of length r is stored and the syndrome p_{zr} for each extended code of length $r + 2$ is determined using the last equation which makes for a fast algorithm. It is also apparent that good equivocation codes will also produce good equivocation codes when extended in length.

4.7.2 Code design technique

The goal of this technique is to increase the equivocation rate of the eavesdropper as much as possible, i.e. the p_{zr} of the syndromes should be as uniform as possible.

Algorithm 21 shows how to extend an $[n, k]$ code into $[n + 2, k + 2]$ by adding two best columns to the original parity check matrix H of the $[n, k]$ code.

Algorithm 21 Code Design Technique 3

Require: $p_z(S_r)$ ▷ Syndrome p_{zr} of $[n, k]$ code
Require: H ▷ Systematic format of H of $[n, k]$ code
Require: $b[i]$ ▷ Integer sequence(columns) of H
Require: (n, k, m, p_e) ▷ Code parameters and error probability of BSC
Require: C_{in} ▷ Initial inequivalent codes of the highest equivocation rate
1: **Generate** (b_r, b_{r+1}) ▷ Generating randomly integers between 3 and $2^m - 1$
Ensure: $(b_r, b_{r+1}) \neq b[i]$ ▷ Ensure no repeated columns
2: $p_{z1}(S_r) \leftarrow (1 - p_e)^2 p_z(S_r)$
3: $p_{z2}(S_r) \leftarrow p_e (1 - p_e) \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_{r+1}}$
4: $p_{z3}(S_r) \leftarrow p_e (1 - p_e) \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_r}$
5: $p_{z4}(S_r) \leftarrow p_e^2 \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus (b_r \oplus b_{r+1})}$
6: $p_z(S_{r+2}) \leftarrow p_{z1}(S_r) + p_{z2}(S_r) + p_{z3}(S_r) + p_{z4}(S_r)$ ▷ Apply equations 4.21 and 4.22
7: $Eqv. \leftarrow - \sum_{j=0}^{2^m-1} \beta_{r+2}(j) \times \log_2(\beta_{r+2}(j))$ ▷ Calculate the equivocation rate of
 $[n + 2, k + 2]$ code
8: $Eqv. rate \leftarrow Eqv./m$ ▷ Calculate the normalised equivocation rate
9: **return** $(b_r, b_{r+1}), Eqv. rate, C_{out}$ ▷ Extended inequivalent codes, which are ranked by equivocation rate in descending order

The steps of the algorithm can be simplified as follow:

1. Calculate the syndrome pmf of the original code $[n, k]$ from equation 4.16.
2. Represent the parity check matrix H of the $[n, k]$ code in the systematic packed integer form: $H = [1, 2, 4, \dots, 2^{m-1}, b_m, \dots, b_{n-1}]$
3. Store the integer sequence of H in $b[i]$, where $0 \leq i \leq n$.
4. Extend H with two integers (b_r, b_{r+1}) by generating randomly all possible integers between 3 and $2^m - 1$ with the constraint that there are no repeated integers included in the original H . This ensures that the minimum Hamming distance of each extended code is at least 3.
5. Select a fixed value of b_r , vary the value of b_{r+1} for all possible values, eliminate all equivalent codes and evaluate the equivocation rate for the binary symmetric eavesdropper channel for a given p_e for each remaining code.
6. Rank the inequivalent codes by their equivocation rate in descending order, and select a best codes subset. These codes are used as the initial input for the next extension round.

4.7.3 Results

By using the code design technique above, the best known equivocation codes have been determined. As example, the $(33, 23, 5)$ code has been extended to $(35, 25, 4)$ code by adding two columns to the parity check matrix of the original code. The H of $(33, 23, 5)$ code can be represented by the following integer sequence:

$$[H] = [1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 77, 191, 232, 341, 382, 411, 495, 505, 607, 629, 643, 664, 682, 763, 764, 815, 822, 860, 919, 971, 977, 997, 1010]$$

By following the code design technique, the maximum equivocation rate has been obtained when adding the following columns (217,928). At $p_e = 0.05$, the equivocation rate for $(33, 23, 5)$ code is equal to $Eqv. rate = 0.792695$ while for $(35, 25, 4)$ is $Eqv. rate = 0.816794$.

We have also extended the code by adding two columns but in two phases. For the same example, firstly the $(33, 23, 5)$ code has been extended to $(34, 24, 4)$ code by adding one column. At $p_e = 0.05$, the maximum equivocation rate has been obtained when adding (89) to the original $(33, 23, 5)$ code and is equal to ($Eqv. rate = 0.805024$). After that, the second column has been added to extend $(34, 24, 4)$ code to $(35, 25, 4)$ code. The maximum equivocation rate has been obtained when adding(150) to the $(34, 24, 4)$ code and the equivocation rate for $(35, 25, 4)$ code becomes ($Eqv. rate = 0.816769$). This leads to say that the process of adding two columns give better results than adding one column in two phases.

By comparing the equivocation values obtained in this example with the corresponding best known codes (BKC) obtained by Grassl [2], we can note that the results obtained by the code design technique described in this section are better. For $(34, 24, 4)$ code, we obtained $Eqv. rate = 0.805024$ while $Eqv. rate = 0.667461$ for Grassl and for $(35, 25, 4)$ code, we obtained $Eqv. rate = 0.816794$ while $Eqv. rate = 0.681773$ for Grassl.

This results has been published in [68].

4.8 Summary

This chapter can be summarized as follows:

- The code design technique for obtaining best known equivocation codes(BE_qC) by extending the binary linear $[n, k]$ code to a $[n + 1, k + 1]$ and $[n + 2, k + 2]$ codes for syndrome coding have been presented.
- A method of implementing syndrome coding without the need for a syndrome look up table has been presented.
- An efficient recursive method for the evaluation of the probability mass function of the syndromes of a code which depends only on the columns of the parity check matrix and the probability of the binary symmetric channel also has been presented.
- It has been shown that the best known equivocation codes with highest minimum distance can be determined by using a combination of the code design technique based on extension of the parity check matrix of a set of good equivocation codes coupled with technique of determining the highest minimum distance of these codes.
- The best known equivocation codes(BE_qC) of the highest minimum distance for the syndrome coding have been determined for a given number of parity bits of the code ($m = 7, 11, 12$) and they are listed in Appendix .
- The results of this chapter have been published in [64], [67] and [68].

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The aim of this dissertation was to investigate the communication security of the wiretap channel when the main channel is a noiseless channel and the eavesdropper channel is a binary symmetric channel (*BSC*). In this channel, the equivocation rate of the eavesdropper has been calculated when the legitimate receivers employ the syndrome coding scheme.

The first stage of the investigation focused on the secrecy coding for the wiretap channel using best known codes (*BKC*). The equivocation and channel capacity in the *BSC* wiretap channel have been investigated using two models with different *BKC's* which were shown to increase the equivocation rate to the eavesdropper. Two models using *BKC* have been proposed using computer implementations, with software written using C++ with the support of NTL library and each model uses two encoding stages. In the first model (Model-1), the first stage employs a syndrome coding scheme based on the (23, 12, 7) binary Golay code and the second stage employs the McEliece cryptosystem technique based on the *BKC*(33, 23, 5) code. In the second model (Model-2), the first stage employs a syndrome coding scheme based on the (23, 12, 7) binary Golay code and the second stage employs the McEliece cryptosystem technique based on the *BKC*(58, 46, 5) code.

The results show that the performance of Model-2 is significantly better than Model-1 which is attributable to the longer code used in the second model. In addition, the results show that both proposed models considerably reduce the information leakage to the eavesdropper compared to previously published schemes. Also, the normalised equivocation difference and the equivocation gain were calculated with reference to an uncoded system. Significant improvements were obtained from Model-2 compared with Model-1. In addition, the performance of the proposed models is better than the performance of previously published schemes in terms of the normalised equivocation difference and the equivocation gain.

The second stage of the investigation focused on the construction of best equivocation codes (BE_qC) for syndrome coding scheme. These have been divided into three paths of investigation:

1. Exhaustive evaluation of new best equivocation codes for syndrome coding with the example case of 15 parity bits.
2. Implementation and construction of best equivocation codes with highest minimum Hamming distance for syndrome coding.
3. Construction of best equivocation codes from shorter best equivocation codes by adding two columns to the parity check matrix.

Codes designed for error correction can be used in syndrome coding but the optimum performance can be realised with codes precisely designed for syndrome coding. The information rate of a syndrome coding scheme using an (n, k, d) linear code is $(\frac{n-k}{n})$ and all possible binary vectors of length n may be transmitted, whilst the information rate of an error correcting coding is $(\frac{k}{n})$ but only codewords are transmitted.

One main objective has been to present a design technique for producing the best known equivocation codes (BE_qC) for the syndrome coding scheme as measured by an information theoretic secrecy metric, the equivocation rate. For this purpose, three schemes for constructing BE_qC codes for syndrome coding in the wiretap channel have been implemented which are based on an efficient recursive method for determining the probability

mass function of the syndromes of a code from the parity check matrix of the code with columns represented as packed integers. This method depends only on the columns of the parity check matrix of the code and the probability of error (p_e) of the binary symmetric channel without the need for a syndrome look-up table.

In the first path of the investigation (first scheme), a code design technique has been presented to produce a new BE_qC codes with 15 parity bits($m = 15$). The best known equivocation codes for the syndrome coding scheme that achieve at least 70% secrecy to an eavesdropper using the BSC with an error probability of 0.05 are presented in Appendix A. The results obtained show that the equivocation rate of the new best known equivocation codes exceeds by a large margin the equivocation rate of the equivalent best error correcting codes (BKC), compiled and published by Grassl [2]. The most surprising result is that, in general, the best known codes rarely coincide with the best known equivocation codes. Usually, the best known equivocation codes have a respectable minimum Hamming distance (d_{min}), but are sometimes not as good as the best known codes with the same parameters. In addition to constructing new best known equivocation codes which have maximum equivocation rate [51, 64], it is useful to constrain the minimum Hamming distance of the codes since this is a key parameter which indicates the code capability in detection and correction errors. Therefore, it is a surprising result in coding theory that there are new best known equivocation codes (BE_qC) in which the minimum distance (d_{min}) is maximal.

This leads to the second path of the investigation (second scheme), where it is shown that the best known equivocation codes with highest minimum distance can be determined by using a combination of the code design technique based on extension of the parity check matrix of a set of good equivocation codes coupled with technique of determining the highest minimum distance of these codes. The best known equivocation codes for the syndrome coding scheme for a given number of parity bits of the code ($m = 7, 11, 12$) are listed in Appendix B in a packed integer format. The highest minimum Hamming distance (d) and the normalised equivocation rate ($Eqv.rate$) for a BSC error probability of $p_e=0.05$ is given for each code. Also the equivocation rates of the corresponding

best error correcting codes previously published, the (*BKC*) codes listed by Grassl [2] with the same n, m and d are given in Appendix B (in parentheses).

The results obtained show that the equivocation rate of the new best known equivocation codes is significantly better than all previously published codes, including the best known codes (*BKC*).

Tables 4.2, 4.3 and 4.4 shows the comparison between three previously published results [2, 51, 67]. In these tables, all cases are presented which have different minimum distances and some representative codes which have the same minimum distances for the following three schemes:

1. The best known correcting codes (*BKC*) (**Scheme-1**) listed by Grassl [2].
2. The best known equivocation codes (**Scheme-2**) listed by Zhang *et al.* [51, 52].
3. The best known equivocation codes (BE_qC) (**Scheme-3**) determined by the research described in this thesis [67].

From Table 4.2, it can be observed that the equivocation rate of **Scheme-2** is better than **Scheme-1** and **Scheme-3** for $n = 35$ up to $n = 44$ as a result of the small minimum distance ($d_2 = 3$) while the highest minimum distance obtained from other schemes is ($d_1 = d_3 = 4$). It is also interesting to note that the equivocation rate of **Scheme-3** is better than **Scheme-1** and **Scheme-2** for $n = 45$ up to $n = 52$ for which the minimum distance of **Scheme-3** is higher than the minimum distance of **Scheme-2**. Also it shown that the equivocation rate of **Scheme-3** is better than other schemes for all other cases at which ($d_1 = d_2 = d_3$).

From Table 4.3, the equivocation rate of **Scheme-2** is better than **Scheme-1** and **Scheme-3** only for ($n = 28, 29$) as a result of the small minimum distance ($d_2 = 5$) while the highest minimum distance obtained from other schemes is ($d_1 = d_3 = 6$).

The most surprising result is that the equivocation rate of **Scheme-3** is better than **Scheme-1** and **Scheme-2** for $n = 30$ up to $n = 89$ including the cases at which the minimum distance of **Scheme-3** is higher than the minimum distance of **Scheme-2**.

Finally, it was shown also from Table 4.4 there are some codes of **Scheme-2** that have

equivocation rate better than **Scheme-1** and **Scheme-3**, possibly because the minimum Hamming distance is less dominant.

In general, Tables 4.2, 4.3 and 4.4 show that significant improvements in equivocation rate are obtained for **Scheme-3** compared with **Scheme-1** for all codes having the same parameters (m, n, d) . Also, it is shown that **Scheme-3** has better equivocation rates compared with **Scheme-2** for all codes having the same parameters (m, n, d) and most codes for which the minimum distance of **Scheme-3** is higher than the minimum distance of **Scheme-2**.

In the final path of the investigation (third scheme), a code design technique has been presented which is based on the extension of the parity check matrix of a good (BE_qC) , $[n, k]$ code by selecting the two best columns that extend the binary linear code $[n, k]$ to $[n + 2, k + 2]$ code and which produce best known equivocation codes. The presented results show that the best equivocation codes obtained by adding two columns gives better performance compared with those codes that are obtained by adding one column in two phases but the price for this improvement is that powerful computer resources are required.

In summary, a total of **207** new best known equivocation codes with the highest minimum Hamming distance have been determined and presented.

5.2 Future Work

- It is proposed that the model design presented in Fig. 3.1 be modified in the following directions to investigate the secrecy coding for the wiretap channel:
 1. The second stage of the model employs the McEliece cryptosystem technique using *BKLC*. It will be interesting to investigate the effect of another Public key cryptosystem such as RSA and NTRU cryptosystems in the model. A comparison of the equivocation rates of these different cryptosystems would form a useful benchmark.
 2. The model includes an inner code and outer code, the first stage employs a syndrome coding scheme based on the outer code ((23, 12, 7) binary Golay code) and the second stage employs the McEliece cryptosystem technique based on the inner code (*BKLC*). In chapter 4, the best equivocation codes(BE_qC) with highest minimum distance have been listed in Appendix A. It would also be interesting to investigate the equivocation and channel capacity in the *BSC* wiretap channel by selecting any two code combinations of (BE_qC). It will be necessary to select the appropriate combinations of (BE_qC) codes in the proposed model so that the length(n) of the outer code is equal to the dimension(k) of the inner code.
- In this thesis, the best known equivocation codes(BE_qC) for the syndrome coding scheme have been calculated using a code design technique of extending the parity check matrix of the linear $[n, k]$ code by one best column as shown in section 4.5 and by adding two best columns as shown in section 4.7. It will be useful to generalise the code design technique to include the addition of best columns for greater than 3 columns coupled with parallel computer processing with appropriate sophisticated software.
- Investigation of the other channel models instead of the binary symmetric channel (*BSC*) for the eavesdropper channel and then to determine the best known equiv-

ocation codes for these new channels. There are several channels that could be used:

1. Additive White Gaussian Noise(*AWGN*) channel.
 2. Binary Erasure Channel(*BEC*).
 3. Arbitrarily Varying Channel(*AVC*).
- It would be useful to investigate and analyse the effect of using circulant codes on the equivocation performance of the eavesdropper. Information is encoded using an circulant and transmitted. The eavesdropper and the legitimate user have to decode using the inverse circulant, which well amplify any channel errors. The equivocation rate may be investigated as a function of the circulant polynomial. In the case of double circulant codes, this method permits correction to full error-correcting capacity in the legitimate receiver [69, 70].

Papers Published

IEEE GIIS 2013, The 5th Global Information Infrastructure and Networking Symposium Conference Paper

Secrecy Coding for the Wiretap Channel Using Best Known Linear Codes

This paper was presented by the author at IEEE GIIS 2013.

Secrecy Coding for the Wiretap Channel Using Best Known Linear Codes

Salah Al-Hassan
 University of Plymouth
 United Kingdom
 salah.al-hassan@plymouth.ac.uk

Mohammed Zaki Ahmed
 University of Plymouth
 United Kingdom
 M.Ahmed@plymouth.ac.uk

Martin Tomlinson
 University of Plymouth
 United Kingdom
 M.Tomlinson@plymouth.ac.uk

Abstract—A special case of wiretap channel is studied and analysed when the main channel is an error free channel and the eavesdropper channel is a binary symmetric channel. The goal of this work is to maximise the equivocation on the eavesdropper side by using a combination of the technique of the McEliece cryptosystem using Best Known Linear Codes(BKLC) coupled with syndrome coding. It is shown that as a result the communication security is improved. In this paper, two Best known linear codes are analysed which increase the equivocation on the eavesdropper side. Two encoding stages are employed. The first stage employs a syndrome coding scheme based on the (23,12,7) binary Golay code and the second stage employs the McEliece cryptosystem technique using BKLC. Analysis shows that the arrangement reduces the information leakage to the eavesdropper compared to previously published schemes.

I. INTRODUCTION

The wiretap channel was proposed by Wyner [1], is one of the channels that take the security of transmitted information into account. The wiretap channel contains one sender and two receivers, one of them is the legitimate user and the second is the eavesdropper. The special case of wiretap channel model is shown in Fig. 1. In this model, Alice(transmitter) wants to send a secret message M to Bob(legitimate receiver) in the presence of an eavesdropper(Eve). The model assumes the main channel between Alice and Bob is an error-free channel and the eavesdropper channel is a Binary Symmetric Channel(BSC) with crossover probability p ($0 < p \leq \frac{1}{2}$).

The goal is to maximise the equivocation on the the eavesdropper side. Wyner shows that in order to increase the equivocation of the eavesdropper as much as possible, the transmission rate of the channel should be reduced. Cohen and Zemor [2] have been clarified that to transmit the information through a wiretap channel, there is a method to select a syndrome function in order to minimise both the length of the transmitted vector and the information leakage to the eavesdropper when using the syndrome coding scheme for the wiretap channel. In her yet to be published PhD thesis [3], Zhang analysed the equivocation of the syndrome-coding scheme in the BSC wiretap channel, and has proposed a modified syndrome-coding scheme to reduce the information leakage in the channel when it has a small probability of error. Zhang showed that the equivocation rate has been increased for many codes such as (23,12,7) Golay code, the

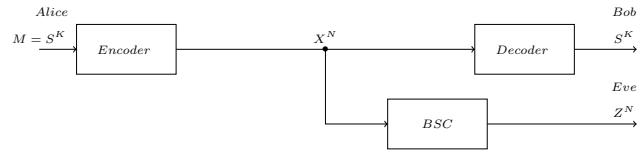


Fig. 1: Wiretap channel Model

extended (24,12,7) Golay code and a number of different BCH codes when compared with the conventional syndrome coding scheme.

In this paper, two models using Best Known Linear Codes (BKLC) are proposed to increase the equivocation in the eavesdropper side and as a result the communication security is improved. In the first model, referred to as Model-1 the transmitted message is first encoded by a syndrome coding scheme based on the (23,12,7) binary Golay code and secondly using the McEliece technique based on the BKLC(33,23,5) code. In the second model, referred to as Model-2 the first stage employs a syndrome coding method based on the (23,12,7) binary Golay code and the second stage employs the McEliece cryptosystem technique based on the BKLC(58,46,5), by concatenating two Golay coded vectors and using this as the input to the second stage. Analysis shows that the second model increases the equivocation rate of the eavesdropper compared to the first model. In addition, the results obtained show that the information leakage has been reduced by a large margin compared to previously published schemes.

II. PROPOSED CODING SCHEME FOR THE WIRETAP CHANNEL

The model design in Fig. 2 includes an inner code and an outer code. Best results are obtained when the outer code employs a syndrome coding scheme based on the (23,12,7) binary Golay code and the the inner code employs the McEliece cryptosystem technique based on Best Known Linear Codes.

A. Encoding Algorithm

1) First Stage[Syndrome Coding using Golay code (23,12,7)]: The Block Diagram of Syndrome Coding using the Golay code (23,12,7) is shown in Fig. 3, and the following

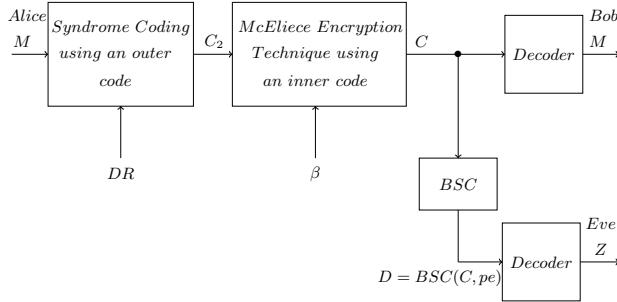


Fig. 2: Block Diagram of Proposed Coding scheme for the Wiretap Channel

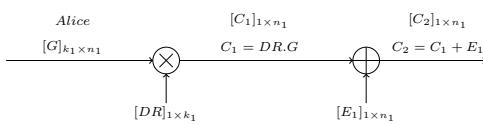


Fig. 3: Block Diagram of Syndrome Coding using Golay code (23,12,7)

procedure shows how Alice starts the encryption process in order to generate a 23-bit vector $[C_2]_{1 \times n_1}$:

Algorithm 1 Encoding Algorithm of Golay code (23,12,7)

Require: $[G]_{k_1 \times n_1}$ \triangleright The Generator Matrix of Golay Code
Require: $[DR]_{1 \times k_1}$ \triangleright Generate random data vector
1: **Generate** $[DR]_{1 \times k_1}$
2: $[C_1]_{1 \times n_1} \leftarrow [DR]_{1 \times k_1} \cdot [G]_{k_1 \times n_1}$
3: $[E_1]_{1 \times n_1} \leftarrow [M]_{1 \times (n_1 - k_1)}$ \triangleright Generate E_1 from M
4: $[C_2]_{1 \times n_1} \leftarrow [C_1]_{1 \times n_1} + [E_1]_{1 \times n_1}$
5: **return** $[C_2]_{1 \times n_1}$

Now, we illustrate how to calculate the error pattern $[E_1]_{1 \times 23}$ of weight between 0 and 3. The error vector E_1 is related to the Message $[M]_{1 \times 11}$. So, the mapping between each error pattern and syndromes is one-to-one because the Golay code is a perfect code. Therefore; the error vector can be calculated by setting $S_i = M_i (i = 0, 1, \dots, 2047)$, E_1 can be calculated from $S_1 = E_1 H^T = M$, where H^T represent the transpose of parity check matrix. As there are only 2048 syndromes, it is straightforward to generate a look up table linking S_1 to E_1 . Bob calculates \hat{M} the estimate of the message as follows:

$$\begin{aligned}\hat{M} &= C_2 H^T = [C_1 + E_1] H^T = [DR.G + E_1] H^T \\ \hat{M} &= DR.GH^T + E_1 H^T\end{aligned}$$

Since G and H are orthogonal ($G.H^T = 0$), then

$$\hat{M} = E_1 H^T = S_1 = M.$$

2) *Second Stage[McEliece cryptosystem technique based on BKLC(n, k, 2t + 1)]*: The Block Diagram of McEliece cryptosystem technique based on $BKLC(n, k, 2t + 1)$ is shown in Fig. 4. The key generation and message encryption are described below:

1) **Key generation**: Bob select a $BKLC(n, k, 2t + 1)$ which can correct t errors. The process of generating the public and

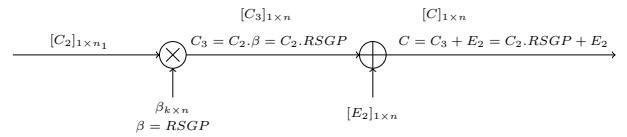


Fig. 4: Block Diagram of McEliece cryptosystem technique based on $BKLC(n, k, 2t + 1)$

private keys are summarized as follows:

Algorithm 2 Key Generation of BKLC(n,k,2t+1)

Require: $[G]_{k \times n} \triangleright$ Generator Matrix of $BKLC(n, k, 2t + 1)$
Require: $[S]_{k \times k}$ \triangleright Select Randomly a Scrambler Matrix
Require: $[P]_{n \times n}$ \triangleright Select Randomly a Permutation Matrix
1: **Generate** $[S]_{k \times k}$ \triangleright Generate non-singular Matrix
Ensure: $|S| \neq 0$ \triangleright The determinant of S not equal 0
2: **Generate** $[P]_{n \times n}$
3: $[\alpha]_{k \times n} \leftarrow SGP$
4: $[R][\alpha] \leftarrow [I_k \mid Q]$ \triangleright Determine $[R]_{k \times k}$
Ensure: $[R] \neq [S]^{-1}$
5: $[\beta]_{k \times n} \leftarrow RSGP$
6: **return** $[\beta]_{k \times n}$

From Algorithm 2, R can be produced by row additions and column swap of α that led to obtain a reduced echelon form of α , i.e. $R.\alpha = [I_k \mid Q]$ where I_k is the $(k \times k)$ identity matrix and Q is $(k \times (n - k))$ parity check matrix. The public key, which will be known to everyone, is $[\beta]_{k \times n} = RSGP$. While S, G, P and R form the private key kept by Bob [4]. It is assumed in the following that Eve has knowledge of the private key.

2) **Message encryption**: The following steps show completion of the encryption process by Alice, where the output of the first stage $[C_2]_{1 \times n_1}$ is used as input to the second stage:

Algorithm 3 Message Encryption of BKLC(n,k,2t+1)

Require: C_2, β
1: $[C_3]_{1 \times n} \leftarrow [C_2]_{1 \times n_1} \cdot [\beta]_{k \times n}$
2: $[C]_{1 \times n} \leftarrow [C_3]_{1 \times n} + [E_2]_{1 \times n}$ \triangleright Alice adds error vector E_2 of weight t
3: **return** $[C]_{1 \times n}$

B. Legitimate Receiver's Decoder

Bob receives the transmitted vector $[C]_{1 \times n}$ via the main channel that is error-free. He recovers the original message M as shown in Algorithm 4. From Algorithm 4(step 3), we can prove that $S_1 = E_2 P^{-1} H^T$ as follows:

$$\begin{aligned}S_1 &= CP^{-1} H^T = (C_3 + E_2) P^{-1} H^T \\ S_1 &= (C_2 \cdot RSGP) P^{-1} H^T + E_2 P^{-1} H^T \\ S_1 &= C_2 \cdot RSGH^T + E_2 P^{-1} H^T \\ \text{But } G \cdot H^T &= 0. \text{ So, } S_1 = E_2 P^{-1} H^T.\end{aligned}$$

Algorithm 4 Legitimate Receiver's Decoder

Require: C, P^{-1}, H^T \triangleright Bob using P^{-1} and H^T of $BKLC(n, k, 2t + 1)$

- 1: $S_1 \leftarrow CP^{-1}H^T$ \triangleright Bob computes $[S_1]_{1 \times n}$
- 2: $S_1 \leftarrow E_2P^{-1}H^T$ \triangleright Bob computes E_2
- 3: $C_3 \leftarrow C + E_2$ \triangleright Bob corrects the error E_2 in C to get C_3 of n -bit
- 4: $C_2 \leftarrow C_3$ \triangleright Bob obtain C_2 from C_3 by get only the first n_1 -bit
- 5: $S_2 \leftarrow C_2H^T$ \triangleright H^T of Golay(23,12,7)code
- 6: $\hat{M} \leftarrow S_2, M = \hat{M}$ \triangleright Bob recovers an estimate of the message \hat{M}
- 7: **return** \hat{M}

C. Eavesdropper's Decoder

The block diagram of the BSC channel and Decoder of the eavesdropper, Eve is shown in Fig. 5. Eve receives a corrupted vector D instead of the transmitted vector $[C]_{1 \times n}$ as a result of passing through the BSC which adds additional errors $[E_{BSC}]_{1 \times n}$ as follows: $[D]_{1 \times n} = [C]_{1 \times n} + [E_{BSC}]_{1 \times n}$, Where $[E_{BSC}]_{1 \times n}$ is a random binary error vector which depends on the crossover probability pe of BSC. Assume Eve uses the same type of decoder that has been used by Bob, the following steps explains how gets the estimated message \hat{M} from the corrupted vector D :

Algorithm 5 Eavesdropper's Decoder

Require: D, P^{-1}, H^T \triangleright Eve using P^{-1} and H^T of $BKLC(n, k, 2t + 1)$

- 1: $S_3 \leftarrow DP^{-1}H^T$ \triangleright Eve computes $[S_3]_{1 \times n}$
- 2: $S_3 \leftarrow \hat{E}P^{-1}H^T$ \triangleright Eve computes \hat{E}
- 3: $\hat{C}_3 \leftarrow C + \hat{E}, \hat{C}_3 \neq C_3$ \triangleright Eve corrects the error \hat{E} to get \hat{C}_3 of n -bit
- 4: $\hat{C}_2 \leftarrow \hat{C}_3$ \triangleright Eve obtain \hat{C}_2 from \hat{C}_3 by get only the first n_1 -bit
- 5: $S_4 \leftarrow \hat{C}_2H^T$ \triangleright H^T of Golay(23,12,7)code
- 6: $\hat{M} \leftarrow S_4, M \neq \hat{M}$ \triangleright Eve recovers an estimate of the message \hat{M}
- 7: **return** \hat{M}

From Algorithm 5(step 3), we can prove that

$$S_3 = \hat{E}P^{-1}H^T \text{ as follows:}$$

$$S_3 = DP^{-1}H^T = (C + E_{BSC})P^{-1}H^T$$

$$S_3 = (C_3 + E_2 + E_{BSC})P^{-1}H^T$$

$$S_3 = (C_2.RSGP)P^{-1}H^T + E_2P^{-1}H^T + E_{BSC}P^{-1}H^T$$

$$S_3 = C_2.RSGH^T + E_2P^{-1}H^T + E_{BSC}P^{-1}H^T$$

$$\text{But } G.H^T = 0. \text{ So, } S_3 = E_2P^{-1}H^T + E_{BSC}P^{-1}H^T$$

$$S_3 = (E_{BSC} + E_2)P^{-1}H^T = \hat{E}P^{-1}H^T.$$

\hat{M} represents the estimated message of 11-bit length and it is equal to $\hat{M} = M + E$. The error signal E which represents the difference between the original message M and the estimated message \hat{M} , which gives us imagine that Eve will receive an equivocation, that mean the percentage of the original information leakage will be less.

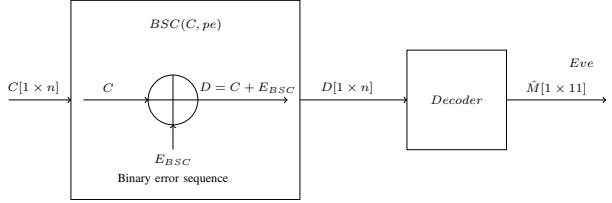


Fig. 5: Block Diagram of the BSC channel and Decoder of Eavesdropper

III. CALCULATION RESULTS OF THE TWO MODELS

The Equivocation rate and the information leakage have been calculated for the proposed coding scheme for two models (Model-1 and Model-2). Both models employs two encoding stages, the first stage of models employs a syndrome coding scheme based on the (23,12,7) binary Golay code. The second stage of Models employs the McEliece cryptosystem technique based on BKLC's. Model-1 employs BKLC(33,23,5) while Model-2 employs BKLC(58,46,5).

A. Channel Capacity

The amount of information that can be received by Eve from Alice via the channel has been calculated as follows [5]: $I(M; \hat{M}) = H(M) + H(\hat{M}) - H(M, \hat{M})$ where $H(M, \hat{M})$ is the joint Entropy of M and \hat{M} . Fig. 6 shows the Normalised information $I(M; \hat{M})$ vs. pe .

B. Equivocation

The amount of information lost in the channel during the transition process from Alice to Eve(i.e. the uncertainty of Eve) has been calculated as follows [6]:

$$H(M | \hat{M}) = H(M, \hat{M}) - H(\hat{M})$$

where $H(M, \hat{M})$ is the joint Entropy of M and \hat{M} . Fig. 7 shows the Normalised equivocation $\frac{H(M|\hat{M})}{M}$ as a function of probability of error pe . Fig. 8 shows the Normalised equivocation for unsecure system on BSC and secure system on BSC of the design models as a function of probability of error pe . Fig. 9 shows the Normalised equivocation Difference between the secure system on BSC of the design models and unsecure system on BSC as a function of probability of error pe .

$$\text{Norm. Eq. Diff.(bits)} = Eq_{(\text{secure system})} - Eq_{(\text{unsecure system})}$$

$$\text{Where } Eq_{(\text{secure system})} = H(M, \hat{M}) \text{ and}$$

$$Eq_{(\text{unsecure system})} = -pe \log_2 pe - (1 - pe) \log_2 (1 - pe)$$

Fig. 10 shows the equivocation gain of the secure system on BSC of the design models and unsecure system on BSC as a function of probability of error pe . A huge improvement in equivocation gain has been achieved between the secure and unsecure system.

$$\text{Equivocation gain(bits)} = \frac{Eq_{(\text{secure system})}}{Eq_{(\text{unsecure system})}}$$

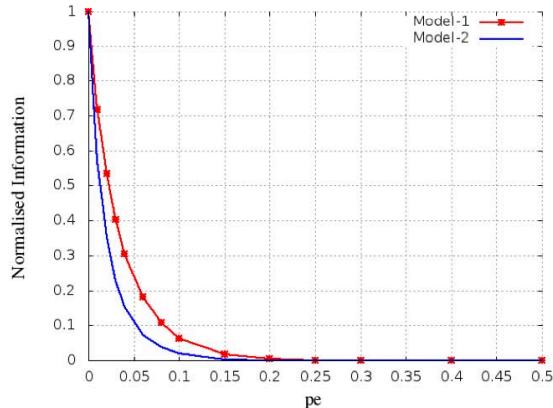


Fig. 6: The Channel Capacity $I(M; \hat{M})$ vs. pe

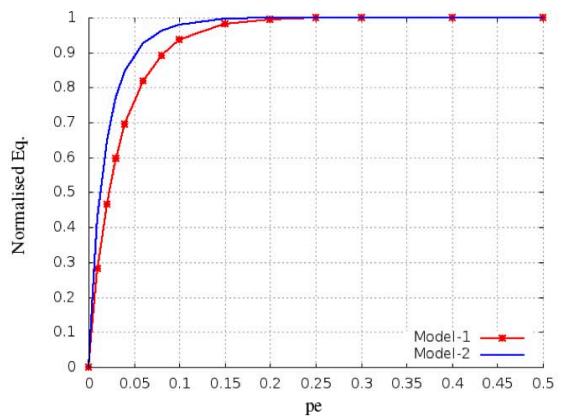


Fig. 7: The Normalised Equivocation $H(M | \hat{M})$ vs. pe

Figures 14, 15, 16 and 17 shows the Normalised equivocation, Channel capacity, the normalised equivocation difference and the equivocation gain for the four schemes, two of the models that have been designed in this paper and two for the models designed by Zhang in [3]. Zhang calculated the equivocation rates for various codes. The results have shown that the (23,12,7) Golay code had the best performance. Therefore, the results obtained from two shcemes in [3] that based on (23,12,7) Golay code have been compared with the results obtained from the proposed models in this paper. The first scheme used the conventional syndrome-coding scheme based on the (23,12,7)Golay code, reffered to as Golay scheme1. The second scheme used the modified syndrome-coding scheme based on the (23,12,7)Golay code, reffered to as Golay scheme2. Table I shows the Normalised equivocation and Information Leakage for the four schemes at some specific values of pe .

IV. CONCLUSION

In this paper, the equivocation and channel capacity in the BSC wiretap channel have been investigated using two models with different BKLC's which were shown to increase the equivocation to the eavesdropper. The results show that the performance of the second model is significantly better than the first model which is attributable to the longer code used in the second model. In addition, the results show that both proposed models considerably reduce the information leakage to the eavesdropper compared to previously published schemes. Also, the normalised equivocation difference and the equivocation gain were calculated with reference to an uncoded system. Significant improvements were obtained from Model-2 compared with Model-1. In addition, the performance of the proposed models is better than the performance of previously published schemes in terms of the normalised equivocation difference and the equivocation gain.

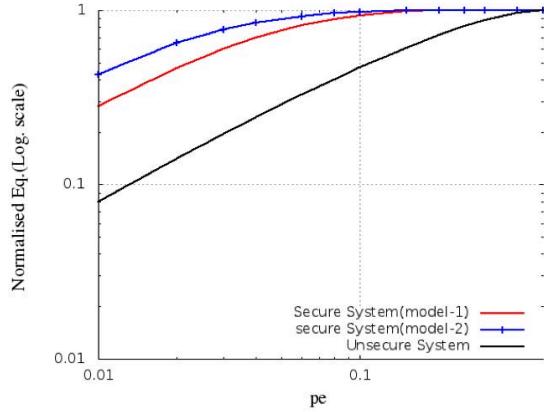


Fig. 8: Normalised Equivocation(Log. Scale) for unsecure system on BSC and secure system on BSC vs. pe

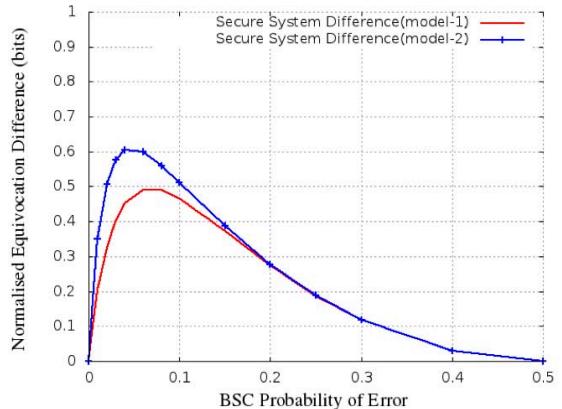


Fig. 9: Normalised Equivocation Difference between the secure system on BSC and unsecure system on BSC vs. pe

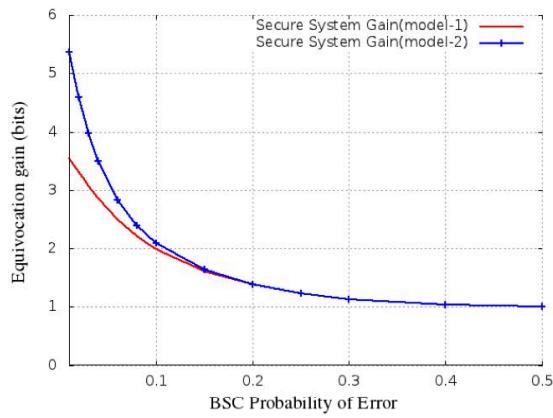


Fig. 10: Equivocation gain of the secure system on *BSC* and unsecure system on *BSC* vs. *pe*

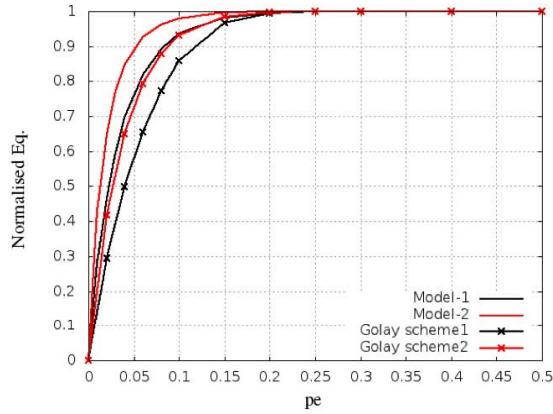


Fig. 11: Normalised Equivocation vs. *pe*

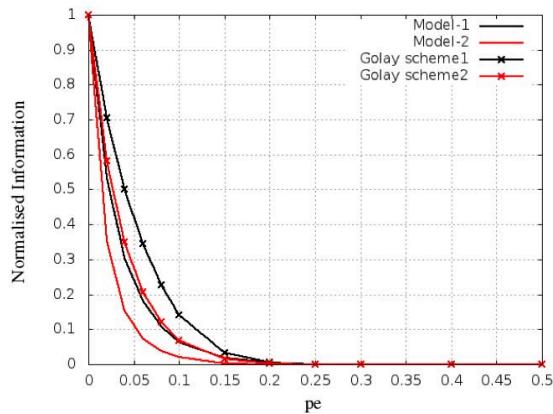


Fig. 12: Channel Capacity(Normalised Information) vs. *pe*

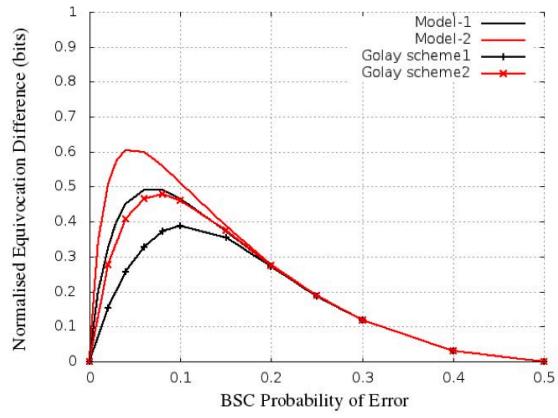


Fig. 13: Normalised Equivocation Difference vs. *pe*

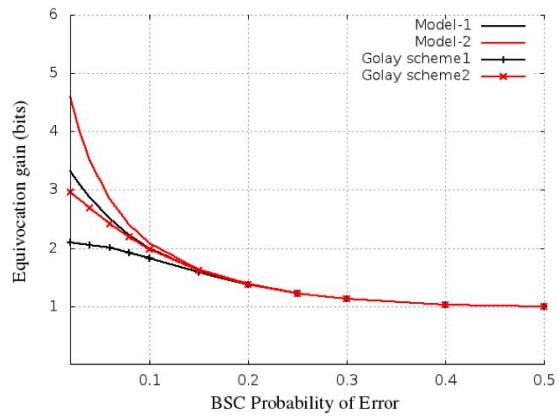


Fig. 14: Equivocation Gain vs. *pe*

<i>pe</i>	Schemes	Normalised Eq.	Information Leakage
0.02	Golay scheme-1	0.2950	0.7050
	Golay scheme-2	0.4175	0.5825
	Model-1	0.4671	0.5329
	Model-2	0.6474	0.3526
0.04	Golay scheme-1	0.4987	0.5013
	Golay scheme-2	0.6509	0.3491
	Model-1	0.6940	0.3060
	Model-2	0.8475	0.1525
0.06	Golay scheme-1	0.6558	0.3442
	Golay scheme-2	0.7929	0.2071
	Model-1	0.8195	0.1805
	Model-2	0.9264	0.0736
0.08	Golay scheme-1	0.7739	0.2261
	Golay scheme-2	0.8795	0.1205
	Model-1	0.8923	0.1077
	Model-2	0.9625	0.0375
0.1	Golay scheme-1	0.8587	0.1413
	Golay scheme-2	0.9317	0.0683
	Model-1	0.9358	0.0642
	Model-2	0.9806	0.0194

TABLE I: Normalised equivocation and Information Leakage for the four schemes

REFERENCES

- [1] A. D. Wyner, "The Wire-Tap Channel," *The Bell System Technical Journal*, vol. 54, pp. 1355–1367, May 1975.
- [2] G. Cohen and G. Zemor, "Syndrome-coding for the wiretap channel revisited," in *Information Theory Workshop, 2006. ITW '06 Chengdu*. IEEE, 2006, pp. 33–36.
- [3] K. Zhang, "A study of syndrome-coding," PhD Thesis(yet to be published), University of Porto.
- [4] K. Zhang, M. Tomlinson, and M. Z. Ahmed, "A Modified McEliece Public Key Encryption System with a Higher Security Level," in *Proc IEEE International Conference on Information Science and Technology*, vol. 1, China, March 2013, pp. 1–5.
- [5] R. W. Hamming, *Coding and Information Theory*. USA: Prentice-Hall, 1997.
- [6] C. Arndt, *Information Measures : information and its description in science and engineering*. Berlin New York: Springer, 2001.

IEEE ICTC 2014, International Conference on Information and Communication Technology Convergence Conference Paper

New Best Equivocation Codes for Syndrome Coding

This paper was presented by the author at IEEE ICTC 2014.

New Best Equivocation Codes for Syndrome Coding

Salah Al-Hassan, Mohammed Zaki Ahmed and Martin Tomlinson
School of Computing and Mathematics
University of Plymouth
United Kingdom
Email: salah.al-hassan, M.Ahmed, M.Tomlinson @plymouth.ac.uk

Abstract—In this paper we present a code design technique which produces codes for syndrome coding which have better secrecy than the best error correcting codes. Code examples are given for the case where the number of parity bits of the code is equal to 15. The code design technique presented is based on extensions of the parity check matrix of a set of good equivocation codes of shorter length. It is also shown that syndrome coding can be implemented without the traditional syndrome look up table, enabling any length codes to be used. An efficient recursive method to calculate the equivocation rate for the binary symmetric channel (BSC) and any linear binary code is also presented. The design results show that the best equivocation codes (BEC) that are produced have better equivocation rate for the syndrome coding scheme compared to all previously published codes, including the best known codes (BKC).

I. INTRODUCTION

The wiretap channel was proposed by Wyner [1], and is a physical layer model that takes the security of transmitted information into account. In this model, Alice (transmitter), wishes to send a secret message M to Bob (legitimate receiver) in the presence of an eavesdropper (Eve). The wiretap channel model is shown in Fig. 1, where the main channel (between Alice and Bob) is an error-free channel and the eavesdropper channel is a Binary Symmetric Channel (BSC) with a probability of error (p_e) [2]. Secrecy is measured by the equivocation rate.

Wyner showed that the equivocation rate approaches unity to the eavesdropper if codes are used that have length extending to infinity, if the syndrome space is chosen to be smaller than the Shannon entropy of the binary symmetric channel (BSC). Chen and Vinck [3] also investigated the binary symmetric wiretap channel, and they showed that the secrecy capacity can be obtained by using random linear codes with syndrome coding.

The syndrome coding scheme, whose basic idea is to convey information in the syndromes of a code so as to increase the communication security has been studied by several researchers. For example, Rouayheb and Soljanin [4] showed that network security can be achieved by using syndrome coding as an additional layer to a network code. Al-Hassan, Ahmed and Tomlinson [5] showed that the equivocation rate can be maximised on the eavesdropper side by using a combination of the technique of the McEliece cryptosystem using Best Known Codes (BKC) coupled with syndrome coding. Cohen and Zemor [6] analysed the information leakage of syndrome coding for the wiretap channel and proposed a

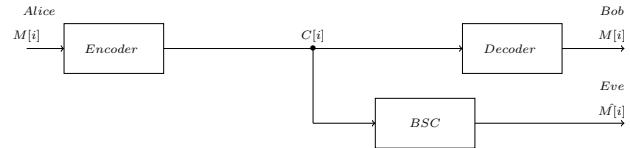


Fig. 1: Wiretap channel Model

method to select a syndrome function in order to minimise both the length of the transmitted vector and the information leakage to the eavesdropper. Code design for error correcting codes is an important and long standing topic in coding theory. Good codes can be designed by extending the parity check matrix of good codes as shown in [7], [8].

In this paper, we present an efficient recursive method for evaluating the equivocation rate of any linear, binary code when used in syndrome coding for the Binary Symmetric Channel (BSC). In addition, we present a code design technique to extend the binary linear $[n, k]$ code to a $[n + 1, k + 1]$ code to produce best equivocation codes. We present examples for the code where the number of parity bits of the code is equal to 15 ($m = 15$) where $m = n - k$. The code construction method for obtaining good equivocation codes is based on the observation that the syndrome probability mass function of a code extended in length is a function of the probability mass function of the original code, and good equivocation codes produce good extended codes. The design results for $m = 15$ show that these new best equivocation codes (BEC) have better equivocation rate compared to all previously published best error correcting codes, the best known codes (BKC) listed by Grassl [9].

II. SYNDROME CODING SCHEME

Wyner showed that the secrecy capacity of the wiretap channel [1] is :

$$C_s = -p_e \cdot \log_2(p_e) - (1 - p_e) \cdot \log_2(1 - p_e) \quad (1)$$

which is the highest transmission rate that can be obtained while maintaining perfect secrecy. In this model, Alice (transmitter) wants to transmit a sequence of independent and uniformly distributed m -bit binary messages to Bob (legitimate receiver), $M[1], \dots, M[r]$. This sequence of messages is encoded into n -bit words $C[1], \dots, C[r]$. Bob receives the same sequence of n -bit words $C[1], \dots, C[r]$ and Eve receives the sequence of n -bit words $D[1], \dots, D[r]$ where

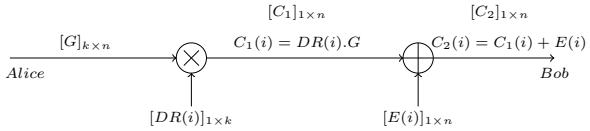


Fig. 2: Block Diagram of Syndrome Coding scheme for (n, k, d) linear block code

$$D(i) = C(i) + E_{BSC}(i), \quad i = 1, \dots, r$$

and $E_{BSC}(i)$ represents a n -bit error vector generated by the binary symmetric channel, r is the block length. The syndrome coding scheme uses a (n, k, d) linear block code which guarantees to correct all error patterns of weight t , where $t = \lfloor (d-1)/2 \rfloor$. All 2^m syndromes are used to send messages where $m = n - k$ and not just the syndromes corresponding to weight t or less error patterns. For any linear block code there exist 2^m distinct minimum weight error patterns, the coset leaders, in which each pattern produces a distinct syndrome of the total 2^m syndromes. Therefore, these error patterns can be represented in a table of 2^m syndromes. In the traditional syndrome coding, the look up table for error patterns and syndromes is known by Alice, Bob and Eve.

For long codes a syndrome table is impractical, but it is shown below that this look up table is unnecessary, and that the parity check matrix H of the code is sufficient taking into consideration the structure of H in systematic format. A block diagram for syndrome coding for a (n, k, d) linear block code is shown in Fig. 2.

A. Encoding Algorithm

Alice starts the encryption process in order to generate a n -bit vector $C_2(i)$ from each m -bit message $M(i)$ at time i such that $C_2(i) \times H^T = M(i)$ as shown in **Algorithm 1**.

Algorithm 1 Encoding Algorithm

Require: $[G]_{k \times n}$ \triangleright The Generator Matrix of (n, k, d) Code
Require: $[DR(i)]_{1 \times k}$ \triangleright random, uniformly distributed vector
1: **Generate** $[DR(i)]_{1 \times k}$
2: $[C_1(i)]_{1 \times n} \leftarrow [DR(i)]_{1 \times k} \cdot [G]_{k \times n}$
3: $[E(i)]_{1 \times n} \leftarrow [M(i)|0\dots0]$ \triangleright Generate n -bit zero padded message
4: $[C_2(i)]_{1 \times n} \leftarrow [C_1(i)]_{1 \times n} + [E(i)]_{1 \times n}$
5: **return** $[C_2(i)]_{1 \times n}$

Now, we show how to calculate the error pattern $[E(i)]_{1 \times n}$. Since the syndrome of any codeword is zero, any codeword added to an error pattern will produce the same syndrome. Hence Alice may produce the required syndrome by generating an n -bit zero padded message vector $E(i)$, which consists of the original message $M(i)$ which is m -bits long followed by k 0's where $m = n - k$.

B. Decoding Algorithm

1) *Legitimate Receiver's Decoder:* Bob receives the transmitted vector $[C_2(i)]_{1 \times n}$ via the main channel that is error-

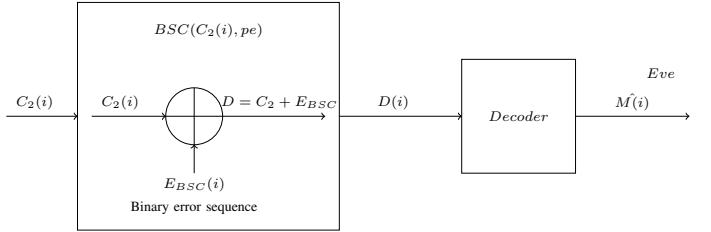


Fig. 3: Block Diagram of the BSC channel and Eavesdropper's Decoder

free. He recovers the original message $M(i)$ by using the parity check matrix of the code as shown in **Algorithm 2**.

Algorithm 2 Legitimate Receiver's Decoder

Require: $C_2(i), H^T$ \triangleright Bob using H^T of (n, k, d) code
1: $S(i) \leftarrow C_2(i) \cdot H^T$ \triangleright Bob computes $[S(i)]_{1 \times n}$
2: $M(i) \leftarrow S(i), M(i) = \hat{M}(i)$ \triangleright Bob recovers the original message $M(i)$
3: **return** $M(i)$

From **Algorithm 2**, we can prove that $C_2(i) \times H^T = M(i)$ as follows:

$$\begin{aligned} S(i) &= C_2(i) \times H^T \longrightarrow S(i) = (C_1(i) + E(i)) \times H^T \\ S(i) &= C_1(i) \times H^T + E(i) \times H^T \\ S(i) &= DR(i) \times G \times H^T + E(i) \times H^T, \text{ but } G \cdot H^T = 0. \\ \text{So, } S(i) &= E(i) \times H^T = M(i). \end{aligned}$$

The syndrome formed from $E(i) \times H^T$, because of the k leading zeros of $E(i)$ is simply $M(i)$ multiplied by the identity sub-matrix of H^T which produces $M(i)$.

2) *Eavesdropper's Decoder:* The block diagram of the BSC channel and eavesdropper's decoder is shown in Fig. 3. Eve receives a corrupted vector $D(i)$ instead of the transmitted vector $[C_2(i)]$ as a result of passing through the BSC which adds additional errors $[E_{BSC}]_{1 \times n}$. Where $[E_{BSC}]_{1 \times n}$ is a random binary error vector which depends on the crossover probability pe of BSC. Assuming Eve uses the same type of decoder that has been used by Bob, the following steps explains how she gets the estimated message $\hat{M}(i)$ from the corrupted vector $D(i)$:

Algorithm 3 Eavesdropper's Decoder

Require: $D(i), H^T$ \triangleright Eve using H^T of (n, k, d) code
1: $S_{Eve}(i) \leftarrow D(i) \cdot H^T$ \triangleright Eve computes $[S_{Eve}(i)]_{1 \times n}$
2: $\hat{M}(i) \leftarrow S_{Eve}(i), M(i) \neq \hat{M}(i)$ \triangleright Eve recovers an estimate of the message $M(i)$
3: **return** $\hat{M}(i)$

From **Algorithm 3**, Eve estimates $\hat{M}(i)$ as follows:

$$S_{Eve}(i) = D(i) \times H^T$$

$$S_{Eve}(i) = [C_2(i) + E_{BSC}(i)] \times H^T$$

$$S_{Eve}(i) = C_2(i) \times H^T + E_{BSC}(i) \times H^T$$

$$S_{Eve}(i) = [C_1(i) + E(i)] \times H^T + E_{BSC}(i) \times H^T$$

$$S_{Eve}(i) = E(i) \times H^T + E_{BSC}(i) \times H^T = \hat{M}(i)$$

$$\hat{M}(i) = S_{Eve}(i) = M(i) + S_e(i)$$

III. CALCULATION OF THE SECRECY ACHIEVED BY SYNDROME CODING

The secrecy realised by syndrome coding is measured by the eavesdropper decoder output equivocation, $H(M(i)|\hat{M}(i))$:

$$\begin{aligned} H(M(i)|\hat{M}(i)) &= H(\hat{M}(i), \hat{M}(i)) - H(\hat{M}(i)) \\ &= H(M(i)) - H(\hat{M}(i)) + H(\hat{M}(i)|M(i)) \\ &= H(M(i)) - H(M(i) + S_e(i)) + \\ &\quad H(M(i) + S_e(i)|M(i)) \\ &= m - m + 0 + H(S_e(i)|M(i)) \end{aligned} \quad (2)$$

$$= H(S_e(i)) \quad (3)$$

$$H(M(i)|\hat{M}(i)) = - \sum_{i=0}^{2^m-1} p(S_e(i)). \log_2 p(S_e(i)) \quad (4)$$

where $H(S_e(i))$ is the entropy of $S_e(i)$. The simplifications in equations (2) and (3) are due to $M(i)$ being uniformly distributed and independent of $S_e(i)$. The equivocation is calculated after deriving the probability mass function of the syndromes due to errors from the BSC, $p(S_e(i))$ and is a function of the code being used through the parity check matrix of the code.

A. Code Representation

Any binary linear (n, k, d) code is defined by its $(k \times n)$ generator matrix G or by its $(m \times n)$ parity check matrix H . The best equivocation codes are constructed by designing the parity check matrix of the code. H can be defined by representing each column of H by an integer, b_i , in the range 0 to $(2^{n-k} - 1)$. The binary parity check matrix of a code of length n is defined by $n.k$ binary integers, the first m columns of H is an identity matrix if H is in systematic form, as shown below:

$$H = \begin{bmatrix} 1 & 0 & \dots & 0 & a_{m0} & \dots & a_{(n-1)0} \\ 0 & 1 & \dots & 0 & a_{m1} & \dots & a_{(n-1)1} \\ \vdots & \vdots & \dots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 & a_{m(m-1)} & \dots & a_{(n-1)(m-1)} \end{bmatrix}$$

in which $0 \leq j \leq m-1$, $m \leq i \leq n-1$ and $a_{i,j}$ takes a value of 0 or 1. Each column can be represented as a packed integer defined as $b_i = \sum_{j=0}^{m-1} a_{ij} \cdot 2^j$. Then the systematic format of H can be represented as following:

$$H = [1, 2, \dots, 2^{m-1}, b_m, \dots, b_{n-1}]$$

where the first m integers represent the identity matrix and the other integers have values between 3 and $2^m - 1$. Usually no integers are repeated ensuring $d \geq 3$ and higher values of d are ensured by constraining no integer is a modulo 2 sum of any other $d-2$, or smaller, number of integers.

B. Evaluation of the syndrome probability distribution

The code construction technique that produces codes with good equivocation is based on the realisation that the syndrome probability mass function (pmf) of a new extended code is a function of the probability mass function of the original code and good equivocation codes produce good extended codes. For Eve, there are 2^n possible error patterns, $e(i)$, occur for each transmitted vector $C(i)_{1 \times n}$. These error patterns occur with probability:

$$p(e(i)) = pe^{w(i)}.(1-pe)^{n-w(i)} \quad (5)$$

where $w(i)$ is the weight of $e(i)$. Each error pattern results in one of the 2^m syndromes being produced.

$$S_e(i) = e(i) \times H^T \quad (6)$$

As the code is linear, for each syndrome there are 2^k error patterns that produce the same syndrome and the probability of each syndrome due to all possible error patterns is given by:

$$p(S_j) = \sum_{i=0}^{2^m-1} p(e(i)).\delta(S_e(i) - S_j) \quad (7)$$

where $\delta()$ is the Dirac function and

$$H(S_e(i)) = - \sum_{j=0}^{2^m-1} p(S_j).Log_2[p(S_j)] \quad (8)$$

This method for evaluating the equivocation works well for short codes ($n < 40$), but for the long codes it is impracticable because it involves the evaluation of 2^n error patterns. Due to the limitation of this method, the probability distribution of the syndromes may be determined recursively, this method leads to reduce the number of terms from 2^n to 2^m as shown in the following theorem.

Theorem : The probability mass function (pmf) of S_j for $j=0$ to 2^m-1 can be defined as $p(S_j) = \beta(j)$ where $\beta(j)$ are coefficients of the probability generating function using the Z transform, denoted as $p_z(S)$ and $p_z(S)$ only depends on the columns of the parity check matrix H and the probability error of the binary symmetric channel p_e .

$$p_z(S) = \sum_{j=0}^{2^m-1} \beta(j)Z^j = \prod_{i=0}^{n-1} ((1-p_e) + p_e.Z^{b_i}) \quad (9)$$

where b_i are the integers representations of the columns of H and exponent sums of powers of Z are added modulo 2.

Proof: Any error pattern may be represented as a sum of single bit error events: $e(i) = [e_1 \ e_2 \ \dots \ e_n]$

$$e(i) = [e_1 \ 0 \ \dots \ 0] + [0 \ e_2 \ \dots \ 0] + \dots + [0 \ 0 \ \dots \ e_n]$$

where $e_i=1$ with probability p_e and $e_i=0$ with probability $1-p_e$. The linearity of the syndrome coding scheme means that the syndrome resulting from any error pattern is the linear sum of the syndromes for each bit error position:

$$S_e(i) = e(i) \times H^T = [e_1 \ e_2 \ \dots \ e_n] \times H^T \quad (10)$$

$$S_e(i) = b_1\delta(e_1 - 1) \oplus b_2\delta(e_2 - 1) \dots \oplus b_n\delta(e_n - 1) \quad (11)$$

Since the probabilities of e_1, e_2, \dots, e_n are independent, the probability of $S_e(i)$ is the product of the probabilities of n separate error events. By adding the coefficients of the same powers of Z results in the coefficients, β_j , the number of terms to be reduced from 2^n to 2^m . If the columns of H of the shortened code of length r are taken from $i = 0$ to $r - 1$ and the pmf generating function of the shortened code is represented as $p_z(S_r)$ then

$$p_z(S_r) = \prod_{i=0}^{r-1} [(1 - pe) + pe \cdot Z^{b_i}] \quad (12)$$

$$\begin{aligned} p_z(S_r) &= [(1 - pe) + pe \cdot Z^{b_0}][(1 - pe) + pe \cdot Z^{b_1}] \dots \\ &\quad \cdot [(1 - pe) + pe \cdot Z^{b_{r-1}}] \\ &= (1 - pe)^2 + pe(1 - pe) \cdot Z^{b_1} + pe(1 - pe) \cdot Z^{b_0} + \\ &\quad pe^2 \cdot Z^{b_0 \oplus b_1} + \dots \end{aligned}$$

Now, we can extend the length of the original code from r to $r + 1$ by adding one column to its parity check matrix H , the pmf generating function of the extended code $r + 1$ is given by

$$p_z(S_{r+1}) = \prod_{i=0}^r [(1 - pe) + pe \cdot Z^{b_i}] = p_z(S_r)[(1 - pe) + pe \cdot Z^{b_r}] \quad (13)$$

Denoting the $\beta(j)$ coefficients of the original code of length r , as $\beta_r(j)$ then

$$p_z(S_r) = \sum_{j=0}^{2^m-1} \beta_r(j) Z^j \quad (14)$$

and for extended code $r + 1$

$$p_z(S_{r+1}) = (1 - pe) \sum_{j=0}^{2^m-1} \beta_r(j) Z^j + pe \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_r} \quad (15)$$

which simplifies to

$$p_z(S_{r+1}) = (1 - pe)p_z(S_r) + pe \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_r} \quad (16)$$

Adding together the coefficients of the same powers of Z in the coefficients, $\beta_r(j)$ to obtain $\beta_{r+1}(j)$, simplifies the above equation to

$$p_z(S_{r+1}) = \sum_{j=0}^{2^m-1} \beta_{r+1}(j) Z^j \quad (17)$$

From equation (16), it is clear that the syndrome pmf of the new code of length $r + 1$ is equal to the syndrome pmf of the original code of length r weighted by $1 - pe$ plus a permuted syndrome pmf of the original code of length r , weighted by pe . The permutation arises from the results of the modulo 2 additions $j \oplus b_r$. This leads to the conclusion that the syndrome pmf of the code can be obtained recursively, starting with the generating function $p_z(S_1)$, determining $p_z(S_2)$ then $p_z(S_3)$ through to $p_z(S_n)$. The syndrome pmf of each (n, k, d) code of length r is stored and the syndrome pmf for each extended code of length $r + 1$ is determined using the equation (16)

which makes for a fast algorithm. It is also apparent that good equivocation codes will also produce good equivocation codes when extended in length.

IV. CODE DESIGN TECHNIQUE

To produce a best equivocation codes the pmf of the syndromes should be as uniform as possible. Since the eavesdropper channel is a binary symmetric channel, for low values of p_e the equivocation is dominated by error patterns of low weight. To produce best equivocation codes, we must take into account the following observations:

- 1) If the error pattern has low weight, then the probability of the error events is high. If each error pattern produces different syndrome sums, then this makes the pmf of the syndromes become more uniform.
- 2) By using the systematic format of the parity check matrix H , the packed integers of any information bit cannot have a weight less than $d - 1$, where d is the minimum Hamming distance of the code. Otherwise the codeword formed from that information bit alone will have weight less than d .
- 3) If any column of the parity check matrix H is repeated, a weight 2 error event will produce a zero syndrome, that leads to a non uniform pmf of the syndrome.

The following code design algorithm shows how to extend an $[n, k]$ code into $[n + 1, k + 1]$ by adding the best column to the original parity check matrix H of the $[n, k]$ code.

Algorithm 4 Code Design Technique

```

Require:  $p_z(S_r)$             $\triangleright$  syndrome pmf of  $(n, k)$  code
Require:  $H$                   $\triangleright$  systematic format of  $H$  of  $(n, k)$  code
Require:  $b[i]$                $\triangleright$  integer sequence(columns) of  $H$ 
Require:  $(n, k, m, pe)$        $\triangleright$  code parameters and error
                           probability of BSC
Require:  $C_{in}$               $\triangleright$  initial inequivalent codes of the highest
                           equivocation rate
1: Generate  $(b_r)$             $\triangleright$  generating between 3 and  $2^m - 1$ 
Ensure:  $(b_r) \neq b[i]$          $\triangleright$  ensure no repeated columns
2:  $p_{z1}(S_r) \leftarrow (1 - pe).p_z(S_r)$ 
3:  $p_{z2}(S_r) \leftarrow pe \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_r}$ 
4:  $p_z(S_{r+1}) \leftarrow p_{z1}(S_r) + p_{z2}(S_r)$      $\triangleright$  apply equations (16)
   and (17)
5:  $Eq \leftarrow - \sum_{j=0}^{2^m-1} \beta_{r+1}(j).Log_2(\beta_{r+1}(j))$      $\triangleright$  calculate the
   equivocation of  $[n + 1, k + 1]$  code
6:  $EqN \leftarrow Eq/m$            $\triangleright$  calculate the Normalised equivocation
7: return  $(b_r), EqN, C_{out}$    $\triangleright$  extended inequivalent codes,
   which are ranked by equivocation in descending order

```

The steps of the algorithm can be simplified as follows:

- 1) Calculate the syndrome pmf of the original code (n, k) from equation(14).
- 2) Represent the parity check matrix H of the (n, k) code in the systematic format:

$$H = [1, 2, 4, \dots, 2^{m-1}, b_m, \dots, b_{n-1}]$$
- 3) Extend H with one integer (b_r) by generating randomly all possible integers between 3 and $2^m - 1$ with the constraint that there are no repeated integers included in the original H . This ensures that the minimum Hamming distance of each extended code is at least 3.
- 4) Eliminate all equivalent codes and evaluate the equivocation rate for each remaining code by using equation(4).
- 5) Rank the inequivalent codes by their equivocation rate in descending order, and select a best codes subset. These codes are used as the initial input for the next extension round.

V. RESULTS

By using the code design technique above, the best equivocation codes have been determined for $m = 15$. As a result of the large number of codes we only present here in Table 1 codes which provide at least 80% secrecy. The minimum Hamming distance (d) and the equivocation rate ($Eq.$) for a BSC error probability of $p_e = 0.05$ is given for each code. The equivocation rates of the corresponding best error correcting codes previously published, the (BKC) codes listed by Grassl [9] with the same n and m are also given in Table 1 (in parentheses). The results show that significant improvements have been achieved on the equivocation rate for the best equivocation codes compared with best known codes.

Fig. 4 shows the equivocation rate $Eq.$ as a function of probability of error p_e of best equivocation and best known codes for $n = 82$ at different values of p_e . It shows that the equivocation rate of BECs has been increased by a large margin compared with BKC not only for $p_e = 0.05$ but also for other values of p_e .

VI. CONCLUSIONS

In this paper, we presented a code design technique for obtaining best equivocation codes and also presented a method of implementing syndrome coding without the need for a syndrome look up table. The best equivocation codes for the syndrome coding scheme that achieve at least 80% secrecy to an eavesdropper using the BSC with an error probability of 0.05 are presented in Table 1. In addition, a recursive method for the evaluation of the probability mass function of the syndromes of a code which depends only on the columns of the parity check matrix and the probability of error of the binary symmetric channel has been presented. The results obtained show that the equivocation rate of the new best equivocation codes exceeds by a large margin the equivocation rates of the best error correcting codes, previously published.

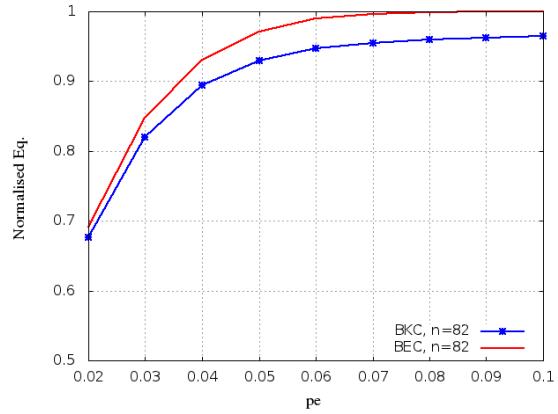


Fig. 4: Equivocation rate $Eq.$ vs. p_e of best equivocation (BEC) and best known (BKC) codes for $n = 82$

REFERENCES

- [1] A. D. Wyner, "The Wire-Tap Channel," *The Bell System Technical Journal*, vol. 54, pp. 1355–1367, May 1975.
- [2] L. H. Ozarow and A. D. Wyner, "Wire-tap channel ii," *The Bell System Technical Journal*, vol. 63, no. 10, pp. 2135–2157, 1984.
- [3] Y. Chen and A. J. Vinck, "On the binary symmetric wiretap channel," *Int. Zurich Seminar on Communications(IZS)*, pp. 17–20, 3-5 March 2010.
- [4] S. Y. E. Rouayheb and E. Soljanin, "On wiretap networks ii," *ISIT*, pp. 551–555, 24–29 June 2007.
- [5] S. Al-Hassan, M. Ahmed, and M. Tomlinson, "Secrecy coding for the wiretap channel using best known linear codes," in *Global Information Infrastructure Symposium, 2013*, Oct 2013, pp. 1–6.
- [6] G. Cohen and G. Zemor, "Syndrome-coding for the wiretap channel revisited," in *Information Theory Workshop, 2006. ITW '06 Chengdu*. IEEE, 2006, pp. 33–36.
- [7] W. Alltop, "A method for extending binary linear codes (corresp.)," *Information Theory, IEEE Transactions on*, vol. 30, no. 6, pp. 871–872, Nov 1984.
- [8] Y. Edel and J. Bierbrauer, "Inverting construction y1," *Information Theory, IEEE Transactions on*, vol. 44, no. 5, pp. 1993–, Sep 1998.
- [9] M. Grassle, "Bounds on the minimum distance of linear codes and quantum codes," 2007,online, Available:<http://www.codetables.de>.

TABLE I: Best Equivocation Codes that achieve at least 80% secrecy in syndrome coding for $p_e = 0.05$

m	n	d	Eq.	Packed integer parity check matrix
15	48	5	0.804141 (0.740435)	1 2 4 8 16 32 64 126 128 256 512 826 1024 2048 3879 4096 7163 7913 8192 9215 9632 10552 16384 16975 17378 17779 18843 19664 21136 21973 22578 23393 24092 24495 25144 26321 26409 26640 26663 27411 28092 28622 29302 29977 31397 31871 32395 32607
15	49	5	0.813668 (0.772691)	1 2 4 8 16 32 64 128 256 512 1024 2048 3879 4096 5541 7031 7160 7913 8192 9215 13987 14289 16384 16975 17378 17579 18413 18843 18960 19350 19955 21973 23259
15	50	5	0.822836 (0.780014)	1 2 4 8 16 32 64 128 256 512 1024 204 3879 4096 5541 7031 7160 7913 8192 9215 9683 10365 13987 16384 16975 17378 17579 18843 18960 19350 19955 21973 23259 23393 24092 24495 25609 25698 26321 26409 27411 28091 28622 28816 31397 31675 31871 32153
15	51	5	0.831624 (0.787048)	1 2 4 8 16 32 64 128 256 512 1024 1914 2048 3879 4030 4096 4956 6925 7913 8192 8508 9215 12422 12617 14876 16384 16975 17378 17797 18843 19400 20477 20900 21021 21530 21973 22767 23393 24092 25048 26321 26409 27411 28091 28622 29302 30283 31397 31651 31871 32754
15	52	5	0.840045 (0.793374)	1 2 4 8 16 32 64 126 128 151 251 512 1024 1349 2048 3879 4098 4938 7913 8192 9215 12183 15202 16207 16384 16975 17378 17957 18843 19400 20477 20900 21021 21530 21973 22767 23393 24092 24495 25609 25698 26321 26409 27411 28091 28622 29302 30283 31397 31651 31871 32754
15	53	5	0.848118 (0.800219)	1 2 4 8 16 32 64 128 256 512 1024 204 3879 4096 4797 5541 7031 7160 7913 8192 9215 13987 14289 14371 16384 16975 17378 17579 18180 18293 18860 19350 19955 21973 22357 23259 23393 24092 24495 25609 25698 26321 26409 27411 28092 28622 28816 31397 31675 31871 32153
15	54	5	0.855849 (0.806378)	1 2 4 8 16 32 64 128 256 512 1024 2048 3879 406 5541 6315 7031 7160 7489 7913 8192 9215 13987 14289 16384 16963 16975 17378 17579 18843 18960 19350 21973 23259 23393 24092 24495 25609 25698 26321 26409 27411 28092 28622 28816 30171 31397 31675 31871 32153
15	55	5	0.863253 (0.812037)	1 2 4 8 16 32 64 126 128 235 256 512 1024 1914 2048 3879 4030 4096 4956 6743 6925 7913 8192 8508 9215 12422 12617 13288 14876 15371 16384 16975 17378 17799 18164 18843 20207 20569 21973 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	56	5	0.870337 (0.817515)	1 2 4 8 16 32 64 126 128 235 256 512 1024 1914 2048 3879 4030 4096 4956 6743 6925 7913 8192 8508 9215 12422 12617 13288 13677 14876 15371 16384 16975 17378 17799 18164 18843 20207 20569 21973 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	57	5	0.877108 (0.822674)	1 2 4 8 16 32 64 128 235 256 512 1024 1914 2048 3879 4030 4096 4956 6743 6925 7913 8192 8508 9215 12422 12617 13288 13677 14876 15371 16384 16975 17378 17799 18164 18858 18884 20207 20569 21973 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	58	5	0.8833542 (0.827573)	1 2 4 8 16 32 64 128 235 256 512 1024 1914 2048 3879 4030 4096 4956 6743 6925 7913 8192 8508 9215 12422 12617 13288 13677 14876 15371 16384 16975 17378 17799 18164 18858 18884 20207 20569 21973 23393 23724 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	59	5	0.889691 (0.832243)	1 2 4 8 16 32 64 128 235 512 1024 2048 3879 406 4964 4797 5541 6703 7031 7160 7870 7913 8192 9215 13987 14289 14371 16384 16975 17378 17579 18180 18293 18843 18960 19350 19955 21973 22357 23259 23393 24092 24495 25609 25698 26321 26409 26993 27411 27570 28092 28313 28622 28816 31397 31675 31871 32484
15	60	5	0.895581 (0.836649)	1 2 4 8 16 32 64 128 256 512 1024 2048 3879 4096 4664 4797 5541 6674 7031 7160 7870 7913 8192 9215 10420 13987 14289 14371 16384 16975 17378 17579 18180 18293 18993 18960 19350 19955 21973 22357 23259 23393 24092 25048 26321 26409 26993 27411 27570 28092 28313 28622 28816 31397 31675 31871 32484
15	61	5	0.901181 (0.84071)	1 2 4 8 16 32 64 128 256 512 1024 2048 3879 4096 4664 4797 5541 6674 7031 7160 7870 7913 8192 9215 10420 13987 14289 14371 16384 16975 17378 17795 18180 18293 18843 18960 19350 19955 21973 22357 23259 23393 24092 24495 25609 25698 26321 26409 26993 27411 27570 28092 28313 28622 28816 31397 31675 31871 32153
15	62	5	0.906529 (0.84471)	1 2 4 8 16 32 64 126 128 235 256 512 1024 1914 2048 3879 4030 4096 4956 6743 6925 7913 8192 8508 9215 10638 12422 12617 13288 13767 14876 15371 16384 16975 17378 17738 17799 18164 18843 20207 20569 21973 23393 23724 24093 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	63	5	0.911620 (0.848502)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 4030 4096 4956 5368 6256 6454 6925 7913 8192 8508 9215 10627 11106 12422 12617 14876 15254 16384 16964 16975 17378 17799 18164 18324 18700 18843 20207 20569 21973 22743 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	64	5	0.916471 (0.852014)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3940 4030 4096 4956 5359 5368 6256 6454 6925 7913 8192 8508 9215 10627 11106 12422 12617 14876 15254 16384 16964 16975 17378 17799 18164 18324 18843 20207 20569 21973 22743 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	65	5	0.921080 (0.853543)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3940 4030 4096 4956 5368 6256 6454 6925 7913 8192 8508 9215 10627 11106 12422 12617 13267 13766 14876 15254 16384 16964 16975 17378 17799 18164 18324 18700 18843 20207 20569 21973 22743 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	66	5	0.925470 (0.858469)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3940 4030 4096 4956 5368 6256 6454 6925 7913 8192 8508 9215 10627 11106 12422 12617 14876 15254 16384 16964 16975 17378 17799 18164 18324 18843 20207 20569 21973 22743 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	67	5	0.929650 (0.861375)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3940 4030 4096 4956 5368 6256 6454 6925 7913 8192 8508 9215 10627 11106 12422 12617 13267 13766 14876 15254 16384 16964 16975 17378 17799 18164 18324 18843 20207 20569 21522 21973 22743 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	68	5	0.933609 (0.873716)	1 2 4 8 16 32 64 126 128 235 256 512 1024 1914 2048 3145 3879 3940 4030 4096 4956 5368 6256 6454 6925 7913 8192 8508 9215 10627 11106 12422 12617 14876 15254 16384 16964 16975 17378 17799 18164 18324 18700 18843 20207 20569 21522 21973 22743 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	69	5	0.937373 (0.883069)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3940 4030 4096 4956 5368 6256 6454 6925 7913 8192 8508 9215 9264 10627 11106 12422 12617 14876 15254 16384 16964 16975 17378 17799 18164 18324 18843 20207 20569 21522 21973 22743 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	70	5	0.940950 (0.890594)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3940 4030 4096 4956 5359 5368 6256 6454 6925 7913 8192 8508 9215 9264 10627 11106 12422 12617 14876 15254 16384 16964 16975 17378 17799 18164 18324 18843 20207 20569 21522 21973 22743 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	71	5	0.944327 (0.896977)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3940 4030 4096 4956 5359 5368 6256 6454 6925 7913 8192 8508 9215 10627 11106 12422 12617 14876 15254 16384 16964 16975 17378 17799 18164 18324 18700 18843 20207 20569 21973 22743 23393 24011 24092 24326 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	72	5	0.947533 (0.902252)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3940 4030 4096 4956 5359 5368 6256 6454 6925 7913 8192 8508 9215 9264 10627 11106 12422 12617 14876 15254 16384 16964 16975 17378 17799 18164 18324 18843 20207 20569 21522 21973 22743 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	73	5	0.950577 (0.906807)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3974 4030 4096 4956 5359 5368 6256 6454 6925 7913 8192 8508 9215 9265 10627 11106 12422 12617 14876 15254 16384 16964 16975 17378 17799 18164 18324 18843 20207 20569 21522 21973 22743 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	74	5	0.953458 (0.910715)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3974 4030 4096 4956 5359 5368 6256 6454 6925 7913 8192 8508 9215 9265 10627 11106 12422 12617 14876 15254 16384 16964 16975 17378 17799 18164 18324 18843 20207 20569 21522 21973 22743 23393 24011 24092 24326 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	75	5	0.956184 (0.914168)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3974 4030 4096 4956 5359 5368 6256 6454 6925 7913 8192 8508 9215 9266 10627 11106 12422 12617 14876 15254 16384 16964 16975 17378 17799 18164 18324 18843 20207 20569 21522 21973 22743 23393 24011 24092 24326 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	76	5	0.958767 (0.916242)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3974 4030 4096 4956 5359 5368 6256 6454 6925 7913 8192 8508 9215 10627 10636 11106 12422 12617 14876 15254 16384 16964 16975 17378 17799 18164 18324 18843 20207 20569 21522 21973 22743 23393 24011 24092 24326 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	77	4	0.961205 (0.919134)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3974 4030 4096 4956 5359 5368 6256 6454 6925 7913 8192 8508 9215 10627 11106 12422 12617 14876 15254 16384 16964 16975 17378 17799 18164 18324 18843 20207 20569 21522 21973 22743 23393 24011 24092 24326 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	78	5	0.963517 (0.92183)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3974 4030 4096 4956 5359 5368 6256 6454 6925 7913 8192 8508 9215 10627 11106 12422 12617 14876 15254 16384 16964 16975 17378 17799 18164 18324 18843 20207 20569 21522 21973 22743 23393 24011 24092 24326 25048 26321

**IEEE GIIS 2014, Global Information Infrastructure and
Networking Symposium Conference Paper**

**Extension of the Parity Check Matrix to Construct the
Best Equivocation Codes for Syndrome Coding**

This paper was presented by the author at IEEE GIIS 2014.

Extension of the Parity Check Matrix to Construct the Best Equivocation Codes for Syndrome Coding

Salah Al-Hassan, Mohammed Zaki Ahmed and Martin Tomlinson

School of Computing and Mathematics
University of Plymouth
United Kingdom

Email: salah.al-hassan, M.Ahmed, M.Tomlinson @plymouth.ac.uk

Abstract—In this paper, we present a code design technique that produces the best equivocation codes(BEqC) for the syndrome coding scheme. This technique is based on the extension of the parity check matrix of a good (BEqC), $[n, k]$ code by selecting the two best columns that extend the length of the code. The goal of this construction is to improve the communication security by increasing the equivocation rate. An efficient recursive method which derives the syndrome probability distribution for a particular code is presented to determine the equivocation rate for the Binary Symmetric Channel(BSC). Analysis show that the (BEqC) that is produced by adding two columns gives better equivocation rate compared to those codes that are produced by the addition of one column in two phases.

I. INTRODUCTION

The wiretap channel was proposed by Wyner [1], is a physical layer model that take the security of transmitted information into account. The special case of wiretap channel model is shown in Fig. 1, when the main channel is an error-free channel and the eavesdropper channel is a Binary Symmetric Channel(BSC). In this model, Alice(transmitter), wishes to send a secret message M to Bob(legitimate receiver) in the presence of an eavesdropper(Eve). Chen and Vinck [2] investigated the binary symmetric wiretap channel, and they show that the secrecy capacity can be obtained by using random linear codes. Salah, Zaki and Martin [3] showed that the equivocation rate has been maximised on the eavesdropper side by using a combination of the technique of the McEliece cryptosystem using Best Known Linear Codes(BKLC) coupled with syndrome coding. Zhang, Tomlinson and Ahmed [4] showed that the best equivocation codes for syndrome coding scheme can be obtained by extending the parity check matrix of the code with one integer. Good codes can be designed by extending the parity check matrix, as shown in [5], [6].

In this paper, we present an efficient recursive method for evaluating the equivocation rate of any linear, binary code when used in syndrome coding for the Binary Symmetric Channel(BSC). In addition, we present a code design technique to extend the binary linear $[n, k]$ code to $[n + 2, k + 2]$ code to produce best equivocation codes. The code construction method for obtaining good equivocation codes is based on the observing that the syndrome probability mass function of a code extended in length is a function of the probability mass function of the original code, and good equivocation codes produce good extended codes. The results obtained

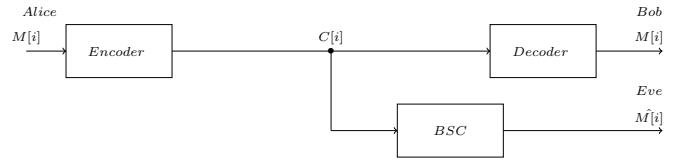


Fig. 1: Wiretap channel Model

show that the equivocation rate has been increased by using this technique compared to previously published scheme that obtained by Grassl [7].

II. CODE REPRESENTATION

Any binary linear (n, k, d) code is defined by its $(k \times n)$ generator matrix G or by its $(m \times n)$ parity check matrix H . In this work, the best equivocation codes are constructed from the parity check matrix of the code. The systematic format of H can be obtained by represented each column of H by an integer, b_i , in the range 0 to $(2^{n-k} - 1)$. The parity check matrix of a code of length n is defined by n integers, the first m columns of H is an identity matrix as shown below:

$$H = \begin{bmatrix} 1 & 0 & \dots & 0 & a_{m0} & \dots & a_{(n-1)0} \\ 0 & 1 & \dots & 0 & a_{m1} & \dots & a_{(n-1)1} \\ \vdots & \vdots & \dots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 & a_{m(m-1)} & \dots & a_{(n-1)(m-1)} \end{bmatrix}$$

in which $0 \leq j \leq m-1$, $m \leq i \leq n-1$ and $a_{i,j}$ takes a value of 0 or 1. Each column can be represented as a packed integer defined as $b_i = \sum_{j=0}^{m-1} a_{ij} \cdot 2^j$. Then the systematic format of H can be represented as following:

$$H = [1, 2, \dots, 2^{m-1}, b_m, \dots, b_{n-1}]$$

where the first m integers represent the identity matrix and the other integers are generated randomly between 3 and $2^m - 1$ while not to be repeated any integers that is contained within H .

III. EVALUATION OF THE SYNDROME PROBABILITY DISTRIBUTION

The code construction technique for obtaining codes with good equivocation is based on the monitoring that the syndrome probability mass function(pmfv) of a new extended code

is a function of the probability mass function of the original code and good equivocation codes produce good extended codes. There are 2^n possible error patterns, $e(i)$, occur in each transmitted vector $C(i)_{1 \times n}$. These error patterns occurs with probability:

$$p(e(i)) = pe^{w(i)} \cdot (1 - pe)^{n-w(i)} \quad (1)$$

where $w(i)$ is the weight of $e(i)$. Each error pattern results in one of the 2^m syndromes being produced.

$$S_e(i) = e(i) \times H^T \quad (2)$$

As the code is linear, for each syndrome there are 2^k error patterns that produce the same syndrome and the probability of each syndrome due to all possible error patterns is given by:

$$p(S_j) = \sum_{i=0}^{2^n-1} p(e(i)) \cdot \delta(S_e(i) - S_j) \quad (3)$$

where $\delta()$ is the Dirac function and

$$H(S_e(i)) = - \sum_{j=0}^{2^m-1} p(S_j) \cdot \log_2[p(S_j)] \quad (4)$$

This method for evaluating the equivocation works well for short codes ($n < 40$), but for the long codes it is impracticable because it involves the evaluation of 2^n error patterns. And due to the limitation of this method, the probability distribution of the syndromes may be determined recursively, this method leads to reduce the number of terms from 2^n to 2^m as shown in the following theorem.

Theorem : The probability mass function (pmf) of S_j for $j=0$ to 2^m-1 can be defined as $p(S_j) = \beta(j)$ where $\beta(j)$ are coefficients of the probability generating function using the Z transform, denoted as $p_z(S)$ and $p_z(S)$ only depends on the columns of the parity check matrix H and the probability error of the binary symmetric channel pe .

$$p_z(S) = \sum_{j=0}^{2^m-1} \beta(j) Z^j = \prod_{i=0}^{n-1} ((1 - pe) + pe \cdot Z^{b_i}) \quad (5)$$

where b_i are the integers representations of the columns of H and exponent sums of powers of Z are added modulo 2. If the columns of H of the shortened code of length r are taken from $i = 0$ to $r-1$ and the pmf generating function of the shortened code is represented as $p_z(S_r)$ then

$$p_z(S_r) = \prod_{i=0}^{r-1} [(1 - pe) + pe \cdot Z^{b_i}] \quad (6)$$

Now, we can extend the length of the original code from r to $r+1$ by adding one column on its parity check matrix H , the pmf generating function of the extended code $r+1$ is given by

$$p_z(S_{r+1}) = \prod_{i=0}^r [(1 - pe) + pe \cdot Z^{b_i}] = p_z(S_r) [(1 - pe) + pe \cdot Z^{b_r}] \quad (7)$$

Denoting the $\beta(j)$ coefficients of the original code of length r , as $\beta_r(j)$ then

$$p_z(S_r) = \sum_{j=0}^{2^m-1} \beta_r(j) Z^j \quad (8)$$

and for extended code $r+1$

$$p_z(S_{r+1}) = (1 - pe) \sum_{j=0}^{2^m-1} \beta_r(j) Z^j + pe \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_r} \quad (9)$$

which simplifies to

$$p_z(S_{r+1}) = (1 - pe)p(S_r) + pe \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_r} \quad (10)$$

In this paper, we extend the length of the original code from r to $r+2$ by adding two columns on its parity check matrix H . We can evaluate the pmf generating function of the extended code $r+2$ as follows:

$$p_z(S_{r+2}) = \prod_{i=0}^{r+1} [(1 - pe) + pe \cdot Z^{b_i}] \quad (11)$$

$$p_z(S_{r+2}) = (1 - pe)^2 p(S_r) + pe(1 - pe) \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_{r+1}} \\ + pe(1 - pe) \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_r} + pe^2 \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus (b_r \oplus b_{r+1})}$$

Adding together the coefficients of the same powers of Z in the coefficients, $\beta_r(j)$ to obtain $\beta_{r+2}(j)$, can simplify the above equation to

$$p_z(S_{r+2}) = \sum_{j=0}^{2^m-1} \beta_{r+2}(j) Z^j \quad (12)$$

This leads to the conclusion that the syndrome pmf of the code can be obtained recursively, starting with the generating function $p_z(S_1)$, determining $p_z(S_2)$ then $p_z(S_3)$ through to $p_z(S_n)$. The syndrome pmf of each (n,k,d) code of length r is stored and the syndrome pmf for each extended code of length $r+2$ is determined using the last equation which makes for a fast algorithm. It is also apparent that good equivocation codes will also produce good equivocation codes when extended in length.

IV. CODE DESIGN TECHNIQUE

The goal of this work is to increase the equivocation of the eavesdropper as much as possible, i.e. the pmf of the syndromes should be as uniform as possible. The following steps shows how to extend an $[n, k]$ into $[n+2, k+2]$ code:

- 1) Calculate the syndrome pmf of the original code (n, k) from equation(8).
- 2) Represent the parity check matrix H of the (n, k) code in the systematic format:

$$H = [1, 2, 4, \dots, 2^{m-1}, b_m, \dots, b_{n-1}]$$
- 3) Store the integer sequence of H in $b[i]$, where $0 \leq i \leq n$.

- 4) Extend H with two integers (b_r, b_{r+1}) by generating randomly all possible integers between 3 and $2^m - 1$ with the constraint that there are no repeated integers included in the original H . This ensures that the minimum Hamming distance of each extended code is at least 3.
- 5) Calculate the syndrome pmf of the extended code $(n + 2, k + 2)$ from equation(12).
- 6) Select a fixed value of b_r and varying the value of b_{r+1} for all possible values, and for each case evaluate the equivocation of the code by using the following equation:

$$Eq. = - \sum_{i=0}^{2^m-1} p(S_e(i)) \cdot \log_2 p(S_e(i)) \quad (13)$$

The equivocation is calculated after deriving the probability mass function of the syndromes due to errors from *BSC*, $p(S_e(i))$ and is a function of the code being used through the parity check matrix of the code. Repeat this process for all possible values of (b_r, b_{r+1}) .

- 7) Choose the values of (b_r, b_{r+1}) that give the maximum equivocation.

V. RESULTS

By using the code design technique above, the best equivocation codes have been determined. The minimum Hamming distance d_{min} and equivocation rate Eq for a *BSC* error probability pe is given for each code. As example, the $(33,23,5)$ code has been extended to $(35,25,4)$ code by adding two columns to the parity check matrix of the original code. The H matrix of $(33,23,5)$ code can be represented by the following integer sequence:

$$[H] = [1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 77, 191, 232, 341, 382, 411, 495, 505, 607, 629, 643, 664, 682, 763, 764, 815, 822, 860, 919, 971, 977, 997, 1010]$$

By following the code design technique, the maximum equivocation rate has been obtained when adding the following columns(217,928). At $pe = 0.05$, the equivocation rate for $(33, 23, 5)$ code is equal to 0.792695 while for $(35, 25, 4)$ is 0.816794. We have also extended the code by adding two columns but in two phases. For the same example, firstly the $(33,23,5)$ code has been extended to $(34,24,4)$ code by adding one column. The maximum equivocation rate has been obtained when adding (89) to the original($33,23,5$)code and at $pe = 0.05$, the equivocation rate for $(34, 24, 5)$ code is 0.805024. After that, the second column has been added to extend $(34, 24, 4)$ code to $(35, 25, 4)$ code, The maximum equivocation rate has been obtained when adding(150) to the $(34, 24, 4)$ code and at $pe = 0.05$, the equivocation rate for $(35, 25, 4)$ code is 0.816769. This leads us to say that the process of adding two columns give better results than adding one column in two phases. By comparing the equivocation values obtained in this example with the corresponding best known codes obtained by Grassl [7], we can note that the results obtained in our technique are better. For $(34, 24, 4)$ code, we obtained $Eq. = 0.805024$ while $Eq. = 0.667461$ for

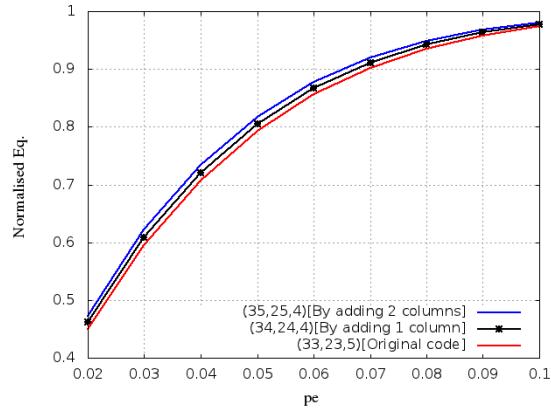


Fig. 2: The Normalised Equivocation $H(M | \hat{M})$ vs. pe

Grassl and for $(35, 25, 4)$ code, we obtained $Eq. = 0.816794$ while $Eq. = 0.681773$ for Grassl. Another example, the $(41, 26, 5)$ code has been extended to $(43, 28, 5)$ code at $pe = 0.05$. For $(41, 26, 5)$ code, we obtained $Eq. = 0.726213$ while $Eq. = 0.685428$ for Grassl, for $(42, 27, 5)$ code, we obtained $Eq. = 0.738604$ while $Eq. = 0.695063$ for Grassl and for $(43, 28, 5)$ code, we obtained $Eq. = 0.750549$ while $Eq. = 0.704144$ for Grassl.

Fig. 2 shows the Normalised equivocation $\frac{H(M(i)|\hat{M}(i))}{m}$ as a function of probability of error pe for the $(33, 23, 5), (34, 24, 4)$ and $(35, 25, 4)$ codes.

VI. CONCLUSION

In this paper, we present a new code design technique for obtaining best equivocation codes. Also, a recursive method for evaluation the probability mass function of the syndromes of a code which depends only on the columns of the parity check matrix and the probability of error of the binary symmetric channel has been presented. In addition, we show that the best equivocation codes obtained by adding two columns gives better equivocation rate compared with those codes that are obtained by adding one column in two phases. The results obtained show that the equivocation rate has been increased by a large margin compared to previously published schemes.

REFERENCES

- [1] A. D. Wyner, "The Wire-Tap Channel," *The Bell System Technical Journal*, vol. 54, pp. 1355–1367, May 1975.
- [2] Y. Chen and A. J. Vinck, "On the binary symmetric wiretap channel," *Int. Zurich Seminar on Communications(IZS)*, pp. 17–20, 3-5 March 2010.
- [3] S. Al-Hassan, M. Ahmed, and M. Tomlinson, "Secrecy coding for the wiretap channel using best known linear codes," in *Global Information Infrastructure Symposium, 2013*, Oct 2013, pp. 1–6.
- [4] K. Zhang, M. Tomlinson, M. Ahmed, M. Ambrose, and M. Rodrigues, "Best binary equivocation code construction for syndrome coding," *IET Communications*, vol. 8, no. 10, pp. 1696–1704, 3 July 2014.
- [5] W. Alltop, "A method for extending binary linear codes (corresp.)," *Information Theory, IEEE Transactions on*, vol. 30, no. 6, pp. 871–872, Nov 1984.
- [6] Y. Edel and J. Bierbrauer, "Inverting construction y1," *Information Theory, IEEE Transactions on*, vol. 44, no. 5, pp. 1993–, Sep 1998.
- [7] M. Grassle, "Bounds on the minimum distance of linear codes and quantum codes," 2007,online, Available:<http://www.codetables.de>.

**IEEE International Conference on Communication, IEEE
ICC 2015 - Workshop on Wireless Physical Layer Secu-
rity Conference Paper**

**Construction of Best Equivocation Codes with Highest
Minimum Distance for Syndrome Coding**

This paper was presented by the author at IEEE ICC 2015.

Construction of Best Equivocation Codes with Highest Minimum Distance for Syndrome Coding

Salah Al-Hassan, Mohammed Zaki Ahmed and Martin Tomlinson

School of Computing and Mathematics
University of Plymouth
United Kingdom

Email: salah.al-hassan, M.Ahmed, M.Tomlinson @plymouth.ac.uk

Abstract—In this paper we present a new code design technique which produces best equivocation codes (BE_qC) with highest minimum distance for syndrome coding which have better secrecy than the best known error correcting codes (BKC). The best equivocation codes of highest minimum distance have been determined by using a combination of a code design technique based on extensions of the parity check matrix from an optimal set of good equivocation codes of highest minimum distance coupled with a technique of determining the highest minimum distance (d_{min}) of these extended codes. Code examples are presented that demonstrate that the equivocation rate of these best equivocation codes (BE_qC) exceeds by a large margin the best known error correcting codes having the same parameters.

I. INTRODUCTION

The concept of the wiretap channel was first introduced by Wyner [1], and is a basic physical layer model that captures the essence of communication security. In this model, Alice (transmitter), wishes to send confidential information to Bob (legitimate receiver) in the presence of an eavesdropper (Eve). Wyner also proposed a syndrome coding scheme for the case [2], shown in Fig. 1, where the main channel is noiseless and the eavesdropper channel is a Binary Symmetric Channel (BSC) with a given probability of error (p_e).

Thangaraj *et al.* [3] showed how capacity achieving codes can be used to achieve the secrecy capacity for any wiretap channel, he also considered the binary erasure channel and the binary symmetric channel as special cases for the wiretap channel. Chen and Vinck [4] also investigated the binary symmetric wiretap channel, and they showed that the secrecy capacity can be obtained by using random linear codes with syndrome coding.

Several researchers have analysed the syndrome coding scheme as a function of the code used in order to increase the communication security, Zhang *et al.* [5] produced some best equivocation codes for the syndrome coding scheme. The best equivocation codes have been determined for a given number of parity bits m , of the code as follows:

($m = 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 18, 20, 22, 24, 26$).

Al-Hassan *et al.* [6] also determined the new best equivocation code for syndrome coding when the parity bit of the code is equal ($m = 15$). In general, the best equivocation codes obtained in [5], [6] have a respectable minimum Hamming distance, but are sometimes not as good as the best known code with the same parameters. An $[n, k]$ linear code C is

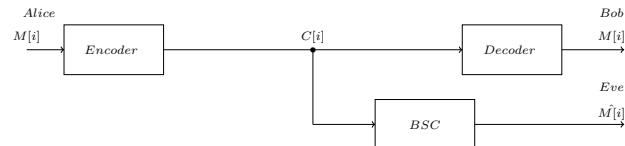


Figure 1: Wiretap channel Model

said to be a best known code (BKC) if C has the highest minimum distance among all known $[n, k]$ linear codes, the tables of (BKC) have been published by Grassl [7] in the form of tables of lower and upper bounds of (d_{min}). Rouayheb and Soljanin [8] showed that network security can be achieved by using syndrome coding as an additional layer to a network code. Al-Hassan *et al.* [9] showed that the equivocation rate can be maximised on the eavesdropper side by using a combination of the technique of the McEliece cryptosystem using Best Known Codes (BKC) coupled with syndrome coding. The Minimum distance of a linear code is an important parameter that provides the capability of detecting and correcting errors by the code. In the past, many researchers have studied the minimum distance (d_{min}) of linear codes. Some results on calculating distance were presented in [10], [11].

In this paper, an efficient recursive method to calculate the equivocation rate for the binary symmetric channel (BSC) and any linear binary code is presented. We also present a code design technique which is used to extend the binary linear $[n, k]$ code of highest minimum distance (d_{min}) to a $[n + 1, k + 1]$ code in order to produce best equivocation codes (BE_qC) with highest minimum distance. The main objective of this paper is to present a design technique for producing the best equivocation codes (BE_qC) of highest minimum distance for the syndrome coding scheme by using a combination of the best equivocation code design technique coupled with the technique of the determination of the highest minimum distance (d_{min}). We also present examples for a given number of parity bits of the code ($m = 7, 11, 12$). The design results show that these new best equivocation codes (BE_qC) have better equivocation rate compared to all previously published best error correcting codes, namely the best known codes (BKC) listed by Grassl [7].

II. CODE REPRESENTATION

Any binary linear (n, k, d) code is defined by its $(k \times n)$ generator matrix G or by its $(m \times n)$ parity check matrix H . The best equivocation codes are constructed by designing the parity check matrix of the code. H can be defined by representing each column of H by an integer, b_i , in the range 0 to $(2^{n-k} - 1)$. The binary parity check matrix of a code of length n is defined by $n.k$ binary integers, the first m columns of H is an identity matrix if H is in systematic form, as shown below:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & \dots & 0 & a_{m0} & \dots & a_{(n-1)0} \\ 0 & 1 & \dots & 0 & a_{m1} & \dots & a_{(n-1)1} \\ \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 & a_{m(m-1)} & \dots & a_{(n-1)(m-1)} \end{bmatrix}$$

in which $0 \leq j \leq m-1$, $m \leq i \leq n-1$ and $a_{i,j}$ takes a value of 0 or 1. Each column can be represented as a packed integer defined as $b_i = \sum_{j=0}^{m-1} a_{ij} \times 2^j$. Then the systematic format of H can be represented as following:

$$H = [1, 2, \dots, 2^{m-1}, b_m, \dots, b_{n-1}]$$

where the first m integers represent the identity matrix and the other integers have values between 3 and $2^m - 1$. Usually no integers are repeated ensuring $d_{min} \geq 3$ and higher values of (d_{min}) are ensured by constraining no integer is a modulo 2 sum of any other ($d_{min} - 2$), or smaller, number of integers.

III. CALCULATION OF THE MINIMUM DISTANCE AND THE EQUIVOCATION RATE ACHIEVED BY SYNDROME CODING

A. Finding minimum distance from parity check matrix

Theorem 1: If C is an (n, k, d) linear code with parity check matrix H , then the minimum distance (d_{min}) of C is equal to the smallest number of columns of H which are linearly dependent. That is, all combinations of $(d_{min} - 1)$ columns are linearly independent, so there is some set of (d_{min}) columns which are linearly dependent.

Proof: Let the columns of the parity check matrix H be denoted by b_0, b_1, \dots, b_{n-1} . A vector $c = [c_0 \ c_1 \ \dots \ c_{n-1}]$ is a codeword iff $c \times H^T = 0$

$$[c_0 \ c_1 \ \dots \ c_{n-1}] \times H^T = c_0 b_0 + c_1 b_1 + \dots + c_{n-1} b_{n-1} = 0 \quad (1)$$

If $c = [c_0 \ c_1 \ \dots \ c_{n-1}]$ is a codeword of smallest weight $d_{min} = w(C)$, then Equation (1) holds, and we deduce that the d_{min} columns of H corresponding to the d_{min} nonzero elements c_i are linearly dependent. ■

There are three cases of d_{min} :

- 1) If H has a column of all zeros, then $d_{min} = 1$.
- 2) If H has two identical columns, then $d_{min} \leq 2$.
- 3) For binary codes, if all columns are distinct and non-zero, then $d_{min} \geq 3$.

B. Evaluation of the equivocation rate achieved by syndrome coding

The secrecy realised by syndrome coding is measured by the eavesdropper decoder output equivocation, $H(M(i)|\hat{M}(i))$:

$$\begin{aligned} H(M(i)|\hat{M}(i)) &= H(M(i), \hat{M}(i)) - H(\hat{M}(i)) \\ &= H(M(i)) - H(\hat{M}(i)) + H(\hat{M}(i)|M(i)) \\ &= H(M(i)) - H(M(i) + S_e(i)) + \\ &\quad H(M(i) + S_e(i)|M(i)) \\ &= m - m + 0 + H(S_e(i)|M(i)) \end{aligned} \quad (2)$$

$$= H(S_e(i)) \quad (3)$$

$$H(M(i)|\hat{M}(i)) = - \sum_{i=0}^{2^m-1} p(S_e(i)) \times \log_2 p(S_e(i)) \quad (4)$$

where $H(S_e(i))$ is the entropy of $S_e(i)$. The simplifications in equations (2) and (3) are due to $M(i)$ being uniformly distributed and independent of $S_e(i)$. The equivocation is calculated after deriving the probability mass function of the syndromes due to errors from the BSC, $p(S_e(i))$ and is a function of the code being used through the parity check matrix of the code.

IV. RECURSIVE EVALUATION OF THE SYNDROME PROBABILITY DISTRIBUTION

The code construction technique that produces codes with good equivocation is based on the realisation that the syndrome probability mass function (pmf) of a new extended code is a function of the probability mass function of the original code and good equivocation codes produce good extended codes.

Theorem 2: The probability mass function (pmf) of S_j for $j=0$ to 2^m-1 can be defined as $p(S_j) = \beta(j)$ where $\beta(j)$ are coefficients of the probability generating function using the Z transform, denoted as $p_z(S)$ and $p_z(S)$ only depends on the columns of the parity check matrix H and the probability error of the binary symmetric channel p_e .

$$p_z(S) = \sum_{j=0}^{2^m-1} \beta(j) Z^j = \prod_{i=0}^{n-1} ((1 - p_e) + p_e \times Z^{b_i}) \quad (5)$$

where b_i are the integers representations of the columns of H and exponent sums of powers of Z are added modulo 2.

Proof: Any error pattern may be represented as a sum of single bit error events: $e(i) = [e_1 \ e_2 \ \dots \ e_n]$ $e(i) = [e_1 \ 0 \ \dots \ 0] + [0 \ e_2 \ \dots \ 0] + \dots + [0 \ 0 \ \dots \ e_n]$ where $e_i=1$ with probability p_e and $e_i=0$ with probability $1 - p_e$. The linearity of the syndrome coding scheme means that the syndrome resulting from any error pattern is the linear sum of the syndromes for each bit error position:

$$S_e(i) = e(i) \times H^T = [e_1 \ e_2 \ \dots \ e_n] \times H^T \quad (6)$$

$$S_e(i) = b_1 \delta(e_1 - 1) \oplus b_2 \delta(e_2 - 1) \dots \oplus b_n \delta(e_n - 1) \quad (7)$$

Since the probabilities of e_1, e_2, \dots, e_n are independent, the probability of $S_e(i)$ is the product of the probabilities of n separate error events. By adding the coefficients of the same

powers of Z results in the coefficients, β_j , the number of terms to be reduced from 2^n to 2^m . ■

If the columns of H of the shortened code of length r are taken from $i = 0$ to $r - 1$ and the pmf generating function of the shortened code is represented as $p_z(S_r)$ then

$$p_z(S_r) = \prod_{i=0}^{r-1} [(1-p_e) + p_e \times Z^{b_i}] \quad (8)$$

$$\begin{aligned} p_z(S_r) &= [(1-p_e) + p_e \times Z^{b_0}] \times [(1-p_e) + p_e \times Z^{b_1}] \dots \\ &\quad \times [(1-p_e) + p_e \times Z^{b_{r-1}}] \\ &= (1-p_e)^2 + p_e(1-p_e) \times Z^{b_1} + p_e(1-p_e) \times Z^{b_0} + \\ &\quad p_e^2 \times Z^{b_0 \oplus b_1} + \dots \end{aligned}$$

Now, we can extend the length of the original code from r to $r+1$ by adding one column to its parity check matrix H , the pmf generating function of the extended code $r+1$ is given by

$$p_z(S_{r+1}) = \prod_{i=0}^r [(1-p_e) + p_e \times Z^{b_i}] = p_z(S_r)[(1-p_e) + p_e \times Z^{b_r}] \quad (9)$$

Denoting the $\beta_r(j)$ coefficients of the original code of length r , as $\beta_r(j)$ then

$$p_z(S_r) = \sum_{j=0}^{2^m-1} \beta_r(j) Z^j \quad (10)$$

and for extended code $r+1$

$$p_z(S_{r+1}) = (1-p_e) \sum_{j=0}^{2^m-1} \beta_r(j) Z^j + p_e \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_r} \quad (11)$$

which simplifies to

$$p_z(S_{r+1}) = (1-p_e)p_z(S_r) + p_e \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_r} \quad (12)$$

Adding together the coefficients of the same powers of Z in the coefficients, $\beta_r(j)$ to obtain $\beta_{r+1}(j)$, simplifies the above equation to

$$p_z(S_{r+1}) = \sum_{j=0}^{2^m-1} \beta_{r+1}(j) Z^j \quad (13)$$

From equation (12), it is clear that the syndrome pmf of the new code of length $r+1$ is equal to the syndrome pmf of the original code of length r weighted by $1-p_e$ plus a permuted syndrome pmf of the original code of length r , weighted by p_e . The permutation arises from the results of the modulo 2 additions $j \oplus b_r$. This leads to the conclusion that the syndrome pmf of the code can be obtained recursively, starting with the generating function $p_z(S_1)$, determining $p_z(S_2)$ then $p_z(S_3)$ through to $p_z(S_n)$. The syndrome pmf of each (n,k,d) code of length r is stored and the syndrome pmf for each extended code of length $r+1$ is determined using the equation (12) which makes for a fast algorithm. It is also apparent that good equivocation codes will also produce good equivocation codes when extended in length.

V. CODE DESIGN TECHNIQUE

To produce a best equivocation codes the pmf of the syndromes should be as uniform as possible. The following code design algorithm shows how to extend an $[n, k]$ code into $[n+1, k+1]$ by adding the best column to the original parity check matrix H of the $[n, k]$ code. We must take into consideration that the value of the minimum distance ($dmin$) of the $[n+1, k+1]$ code should be equal to the minimum distance ($dminH$) of the $[n+1, k+1]$ BKC code because the best known $[n, k]$ code (BKC) has the highest minimum distance among all known $[n, k]$ linear codes.

Algorithm 1 Code Design Technique

```

Require:  $p_z(S_r)$            ▷ syndrome pmf of  $(n, k)$  code
Require:  $H$                  ▷ systematic format of  $H$  of  $(n, k)$  code
Require:  $dminH$             ▷ highest  $dmin$  of  $(n+1, k+1)$  BKC code
Require:  $b[i]$               ▷ integer sequence(columns) of  $H$ 
Require:  $(n, k, m, p_e)$       ▷ code parameters and error probability of BSC
Require:  $C_{in}$               ▷ initial inequivalent codes of the highest equivocation rate
1: Generate  $(b_r)$            ▷ generating between 3 and  $2^m - 1$ 
Ensure:  $(b_r) \neq b[i]$         ▷ ensure no repeated columns
Require:  $dmin$              ▷  $dmin$  of  $(n+1, k+1)$  code
Ensure:  $dmin = dminH$ 
2:  $p_{z1}(S_r) \leftarrow (1-p_e) \times p_z(S_r)$ 
3:  $p_{z2}(S_r) \leftarrow p_e \sum_{j=0}^{2^m-1} \beta_r(j) Z^{j \oplus b_r}$ 
4:  $p_z(S_{r+1}) \leftarrow p_{z1}(S_r) + p_{z2}(S_r)$     ▷ apply equations (12) and (13)
5:  $Eq \leftarrow - \sum_{j=0}^{2^m-1} \beta_{r+1}(j) \times \log_2(\beta_{r+1}(j))$  ▷ calculate the equivocation of  $[n+1, k+1]$  code
6:  $EqN \leftarrow Eq/m$  ▷ calculate the Normalised equivocation
7: return  $(b_r), dmin, EqN, C_{out}$           ▷ extended inequivalent codes of highest  $dmin$ , which are ranked by equivocation in descending order

```

The steps of the algorithm can be simplified as follows:

- 1) Calculate the syndrome pmf of the original code (n, k) from equation(10).
- 2) Represent the parity check matrix H of the (n, k) code in the systematic format:

$$H = [1, 2, 4, \dots, 2^{m-1}, b_m, \dots, b_{n-1}]$$
- 3) Extend H with one integer (b_r) by generating randomly all possible integers between 3 and $2^m - 1$ with the constraint that there are no repeated integers included in the original H . This ensures that the minimum Hamming distance of each extended code is at least 3.
- 4) Calculate the highest minimum distance ($dminH$) of the $(n+1, k+1)$ BKC code from Magma Software.
- 5) Calculate the minimum distance ($dmin$) of the extended code $(n+1, k+1)$ and ensure that $dmin = dminH$.

- 6) Eliminate all equivalent codes and evaluate the equivocation rate for each remaining code by using equation(4).
- 7) Rank the inequivalent codes by their equivocation rate in descending order, and select a best codes subset. These codes are used as the initial input for the next extension round.

VI. RESULTS

In this section, we present the best equivocation codes of the highest minimum distance, which are obtained by the code design technique, in the form of the equivocation rate for various values of n and for a given number of parity bits of the code $m = 7, 11, 12$. In Table 1, we have compare the equivocation rate of our results (scheme-3) with the best known correcting codes *BKC* (scheme-1) listed by Grassl [7] and the best equivocation codes (scheme-2) listed by Zhang [5] for some representative codes. It shows that significant improvements in equivocation rate were obtained from our results(scheme-3) compared with Grassl (scheme-1) and Zhang (scheme-2) results. As a result of the large number of codes we only present here in Table 2 (for $m = 7$) and Table 4 (for $m = 11$) codes which provide at least 90% secrecy and in Table 3 (for $m = 12$) codes which provide at least 96% secrecy. The highest minimum Hamming distance (d_{min}) and the equivocation rate ($Eq.$) for a BSC error probability of $p_e = 0.05$ is given for each code. The equivocation rates of the corresponding best error correcting codes previously published, the (BKC) codes listed by Grassl [7] with the same n and m are also given in Tables (2,3,4) (in parentheses). The results show that significant improvements have been achieved on the equivocation rate for the best equivocation codes compared with best known codes.

VII. CONCLUSIONS

In this paper, we have shown that the best equivocation codes with highest minimum distance can be determined by using a combination of the code design technique based on extension of the parity check matrix of a set of good equivocation codes coupled with technique of determining the highest minimum distance of these codes. The best equivocation codes for the syndrome coding scheme for a given number of parity bits of the code ($m = 7, 11, 12$) that achieve at least 90% and 96% secrecy to an eavesdropper using the BSC with an error probability of 0.05 are presented in Tables (2,3,4). In addition, it has been shown that syndrome coding can be implemented without the traditional syndrome look up table by using a recursive method for the evaluation of the probability mass function of the syndromes of a code which depends only on the columns of the parity check matrix and the probability of error of the binary symmetric channel. The results obtained show that the equivocation rate of the new best equivocation codes is significantly better than all previously published codes, including the best known codes (*BKC*).

REFERENCES

- [1] A. D. Wyner, "The Wire-Tap Channel," *The Bell System Technical Journal*, vol. 54, pp. 1355–1367, May 1975.

Table I: Equivocation rate and minimum distance for three schemes in syndrome coding for $p_e = 0.05$

m	n	$Eq.(Scheme-1)$	d_{min1}	$Eq.(Scheme-2)$	d_{min2}	$Eq.(Scheme-3)$	d_{min3}
7	45	0.956930	4	0.958694	3	0.958863	4
7	46	0.960586	4	0.961848	3	0.962228	4
7	47	0.963848	4	0.964899	3	0.965345	4
7	48	0.966766	4	0.967706	3	0.968205	4
7	49	0.969876	4	0.970311	3	0.970856	4
7	50	0.972635	4	0.972711	3	0.973272	4
7	51	0.975095	4	0.974942	3	0.975516	4
7	52	0.977298	4	0.976991	3	0.977571	4
7	53	0.979291	4	0.979291	4	0.979454	4
7	54	0.981087	4	0.981087	4	0.981189	4
7	55	0.982712	4	0.982713	4	0.982787	4
7	56	0.984185	4	0.984186	4	0.984253	4
7	57	0.985569	4	0.985569	4	0.985599	4
7	58	0.986824	4	0.986824	4	0.986835	4
7	59	0.987968	4	0.987968	4	0.98972	4
11	30	0.710225	6	0.710315	5	0.710469	6
11	37	0.803255	5	0.806696	4	0.807675	5
11	38	0.815439	5	0.817966	4	0.818806	5
11	39	0.826902	5	0.828624	4	0.829343	5
11	40	0.837627	5	0.838712	4	0.839390	5
11	41	0.847670	5	0.848307	4	0.848951	5
11	48	0.710488	4	0.902358	4	0.903225	4
11	49	0.713286	4	0.908427	4	0.909219	4
11	50	0.715762	4	0.914199	4	0.914902	4
11	51	0.717951	4	0.919611	4	0.920284	4
11	52	0.719886	4	0.924695	4	0.925399	4
11	53	0.721705	4	0.929476	4	0.930200	4
11	54	0.723311	4	0.933982	4	0.934726	4
11	60	0.730463	4	0.955931	4	0.956635	4
11	65	0.734121	4	0.968862	4	0.969326	4
11	70	0.770032	4	0.978196	4	0.978503	4
11	75	0.789478	4	0.984852	4	0.984993	4
11	79	0.797503	4	0.988754	4	0.988826	4
12	39	0.796947	6	0.797169	5	0.797463	6
12	40	0.808546	6	0.808547	6	0.808919	6
12	57	0.931666	5	0.931678	4	0.931739	5
12	58	0.936021	5	0.936020	5	0.936072	5
12	59	0.940113	5	0.940112	5	0.940148	5
12	60	0.943958	5	0.943961	5	0.943986	5
12	70	0.729329	4	0.971041	4	0.971306	4
12	71	0.729730	4	0.972963	4	0.973186	4
12	72	0.730124	4	0.974773	4	0.974954	4
12	73	0.736062	4	0.976457	4	0.976619	4
12	74	0.740797	4	0.978034	4	0.978181	4
12	75	0.744526	4	0.979512	4	0.979648	4
12	76	0.747576	4	0.980897	4	0.981018	4
12	77	0.749986	4	0.982193	4	0.982305	4
12	78	0.751983	4	0.983409	4	0.983510	4
12	79	0.753583	4	0.984547	4	0.984637	4
12	80	0.754920	4	0.985609	4	0.985695	4

- [2] L. H. Ozarow and A. D. Wyner, "Wire-tap channel ii," *The Bell System Technical Journal*, vol. 63, no. 10, pp. 2135–2157, 1984.
- [3] A. Thangaraj, S. Dihidar, A. Calderbank, S. McLaughlin, and J.-M. Merolla, "Applications of ldpc codes to the wiretap channel," *Information Theory, IEEE Transactions on*, vol. 53, no. 8, pp. 2933–2945, Aug 2007.
- [4] Y. Chen and A. J. Vinck, "On the binary symmetric wiretap channel," *Int. Zurich Seminar on Communications(IZS)*, pp. 17–20, 3-5 March 2010.
- [5] K. Zhang, M. Tomlinson, M. Ahmed, M. Ambrose, and M. Rodrigues, "Best binary equivocation code construction for syndrome coding," *Communications, IET*, vol. 8, no. 10, pp. 1696–1704, July 2014.
- [6] S. Al-Hassan, M. Z. Ahmed, and M. Tomlinson, "New best equivocation codes for syndrome coding," in *Information and Communication Technology Convergence (ICTC), 2014 International Conference on*, Oct 2014, pp. 669–674.
- [7] M. Grassle, "Bounds on the minimum distance of linear codes and quantum codes," 2007,online, Available:<http://www.codetables.de>.
- [8] S. Y. E. Rouayheb and E. Soljanin, "On wiretap networks ii," *ISIT*, pp. 551–555, 24–29 June 2007.
- [9] S. Al-Hassan, M. Ahmed, and M. Tomlinson, "Secrecy coding for the wiretap channel using best known linear codes," in *Global Information Infrastructure Symposium, 2013*, Oct 2013, pp. 1–6.
- [10] R. Bhattar, K. Ramakrishnan, and K. Dasgupta, "On computation of minimum distance of linear block codes above 1/2 rate coding," in *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, June 2010, pp. 280–284.
- [11] M. Askali, S. Nough, and M. Belkasmi, "An efficient method to find the minimum distance of linear block codes," in *Multimedia Computing and Systems (ICMCS), 2012 International Conference on*, May 2012, pp. 318–324.

Table II: Best Equivocation Codes that achieve at least 90% secrecy in syndrome coding for $p_e = 0.05$

m	n	dmin	Eq.	Packed integer parity check matrix
7	35	4	0.904253 (0.872739)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 74 82 84 88 98 100 103 115 122 127
7	36	4	0.912049 (0.888646)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 67 74 82 84 88 98 100 103 115 122 127
7	37	4	0.919097 (0.901769)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 67 74 76 82 84 88 98 100 103 115 122 127
7	38	4	0.925583 (0.912693)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 67 74 76 82 84 88 97 98 100 103 115 122 127
7	39	4	0.931601 (0.921923)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 67 74 76 82 84 88 91 97 98 100 103 115 122 127
7	40	4	0.937142 (0.929789)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 67 74 76 82 84 88 91 97 98 100 103 115 118 122 127
7	41	4	0.942230 (0.936808)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 59 64 67 74 76 82 84 88 91 97 98 100 103 115 118 122 127
7	42	4	0.946878 (0.942873)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 59 64 67 74 76 79 82 84 88 91 97 98 100 103 115 118 122 127
7	43	4	0.951182 (0.948166)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 59 64 67 70 74 76 82 84 88 91 97 98 100 103 110 115 118 122 127
7	44	4	0.955172 (0.952803)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 59 64 67 74 76 79 82 84 88 91 97 98 100 103 104 115 117 118 122 127
7	45	4	0.958863 (0.956930)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 59 62 64 67 74 76 79 82 84 88 91 97 98 100 103 104 115 117 118 122 127
7	46	4	0.962228 (0.960586)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 59 62 64 67 74 76 79 82 84 88 91 97 98 100 103 104 115 117 118 122 127
7	47	4	0.965345 (0.963848)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 59 62 64 67 70 76 79 84 87 91 93 94 98 103 107 109 110 115 117 121 122 124 127
7	48	4	0.968205 (0.966766)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 52 55 56 64 67 70 73 76 79 82 87 88 93 94 97 98 100 104 110 112 115 121 122 124
7	49	4	0.970856 (0.969876)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 59 61 62 64 67 70 73 76 79 84 87 91 93 94 98 103 107 109 110 115 117 121 122 124
7	50	4	0.973277 (0.972635)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 59 62 64 67 70 74 76 79 82 84 87 88 91 94 97 98 100 103 104 115 117 118 121 122 127
7	51	4	0.975516 (0.975095)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 56 59 62 64 67 70 74 76 79 82 84 87 88 91 94 97 98 100 103 104 115 117 118 121 122 127
7	52	4	0.977571 (0.977298)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 52 59 62 64 67 73 74 76 79 82 84 87 88 91 94 97 98 100 103 104 107 115 117 118 121 122 127
7	53	4	0.979454 (0.979291)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 52 59 62 64 67 73 74 76 79 82 84 87 88 91 94 97 98 100 103 104 107 115 117 118 121 122 127
7	54	4	0.981189 (0.981087)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 50 52 56 59 62 64 67 69 70 74 76 79 82 84 87 88 91 94 97 98 100 103 104 115 117 118 121 122 127
7	55	4	0.982787 (0.982712)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 50 52 56 59 62 64 67 69 70 74 76 79 82 84 87 88 91 94 97 98 100 103 104 115 117 118 121 122 124
7	56	4	0.984253 (0.984185)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 50 52 55 56 59 62 64 67 69 73 74 76 81 82 84 87 88 91 93 94 97 98 100 103 104 110 112 115 117 122
7	57	4	0.985599 (0.985569)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 50 52 55 56 59 61 62 64 67 69 70 73 74 76 81 82 84 88 91 93 97 98 100 103 104 110 112 115 117
7	58	4	0.986835 (0.986824)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 50 52 55 56 59 62 64 67 69 70 74 76 79 81 82 84 87 88 91 93 94 97 98 100 103 104 115 117 118 121
7	59	4	0.987972 (0.987968)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 50 52 55 56 59 61 62 64 67 69 70 74 76 79 81 82 84 87 88 91 93 94 97 98 100 103 104 115 117 118

Table III: Best Equivocation Codes that achieve at least 96% secrecy in syndrome coding for $p_e = 0.05$

m	n	dmin	Eq.	Packed integer parity check matrix
12	66	4	0.962563 (0.710374)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 549 618 657 687 690 717 783 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1885 1897 1926 2054 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3306 3721 3770 3794 3852 3877 3924
12	67	4	0.964938 (0.720589)	1 2 4 8 16 32 64 128 256 512 1024 2048 292 309 345 549 618 657 687 690 717 783 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1885 1897 1926 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3306 3721 3770 3794 3852 3877 3924
12	68	4	0.967183 (0.721090)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 549 618 657 687 690 717 783 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1885 1897 1926 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3306 3721 3770 3794 3852 3877 3924
12	69	4	0.969306 (0.728861)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 549 618 657 687 690 717 783 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1885 1897 1926 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3306 3721 3770 3794 3852 3877 3924
12	70	4	0.971306 (0.729329)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 687 690 717 783 963 981 1037 1098 1168 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1885 1897 1926 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3305 3726 3770 3794 3852 3877 3924
12	71	4	0.973186 (0.729730)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 687 690 717 783 963 981 1037 1098 1168 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1885 1897 1926 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3305 3726 3770 3794 3852 3877 3924
12	72	4	0.974954 (0.730124)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 687 690 717 783 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3306 3726 3772 3794 3852 3877 3924
12	73	4	0.976619 (0.736062)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 687 690 717 783 882 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3306 3726 3772 3794 3852 3877 3924
12	74	4	0.978181 (0.740479)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 687 690 717 783 882 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1718 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2659 2748 2760 2790 2809 2818 2868 2990 3132 3306 3724 3852 3877 3924
12	75	4	0.979648 (0.744526)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 687 690 717 783 818 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2659 2748 2760 2790 2809 2818 2868 2990 3132 3306 3724 3852 3877 3924
12	76	4	0.981018 (0.747576)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 687 690 717 783 818 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2659 2748 2760 2790 2809 2818 2868 2990 3132 3306 3724 3852 3877 3924
12	77	4	0.982305 (0.749986)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 687 690 717 783 875 882 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2659 2748 2760 2790 2809 2818 2868 2990 3132 3306 3727 3770 3794 3852 3877 3924
12	78	4	0.983510 (0.751983)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 687 690 717 783 875 882 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2659 2748 2760 2790 2809 2818 2868 2990 3132 3306 3727 3770 3794 3852 3877 3924
12	79	4	0.984637 (0.753583)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 687 690 717 783 875 882 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2659 2748 2760 2790 2809 2818 2868 2990 3132 3306 3727 3770 3794 3852 3877 3924
12	80	4	0.985695 (0.754920)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 687 690 717 783 875 882 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1766 1653 1718 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2659 2748 2760 2790 2809 2818 2868 2990 3132 3306 3721 3727 3770 3794 3852 3877 3924
12	81	4	0.986682 (0.756032)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 687 690 717 783 875 882 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1766 1653 1718 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2659 2748 2760 2790 2809 2818 2868 2990 3132 3306 3764 3721 3727 3770 3794 3852 3877 3924
12	82	4	0.987605 (0.756923)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 687 690 717 783 875

Table IV: Best Equivocation Codes that achieve at least 90% secrecy in syndrome coding for $p_e = 0.05$

Appendix A

Best Known Equivocation Codes with 15 parity bits

Table A.1: Best Known Equivocation Codes that achieve at least 70% secrecy in syndrome coding for $p_e = 0.05$

m	n	d	Equiv. rate	Packed integer parity check matrix
15	39	5	0.700212 (0.664859)	1 2 4 8 16 32 64 126 128 256 512 858 1024 2048 3879 4096 4875 7913 8192 9215 10691 11365 15737 16384 16975 17378 18843 21973 23393 24092 26321 26409 27411 28092 28622 29302 29873 31397 31871
15	40	5	0.713428 (0.675327)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3879 4030 4096 7913 8192 9215 12617 14876 16384 16975 17378 17799 18843 21973 23393 24092 25048 26321 26409 27411 28092 28622 29302 30283 31397 31871
15	41	5	0.726213 (0.685428)	1 2 4 8 16 32 64 126 128 256 512 1024 2048 3879 4096 5197 7913 8192 8757 9215 13729 16384 16647 16975 17378 18843 20506 21973 22973 23393 24092 26321 26409 27411 28092 28533 28622 29302 29872 31397 31871
15	42	5	0.738604 (0.695063)	1 2 4 8 16 32 64 126 128 256 512 1024 1252 2048 3879 4096 7163 7913 8192 9215 14246 16384 16975 17378 17779 18843 21136 21973 23393 24092 24495 26321 26409 26640 27411 28092 28622 29302 29977 31397 31871 32395
15	43	5	0.750549 (0.704144)	1 2 4 8 16 32 64 126 128 256 512 1024 2048 3879 4096 7099 7571 7913 8192 9215 15602 16384 16975 17378 18843 21138 21973 22445 22735 23393 24092 26321 26409 26706 26773 27396 27411 28092 28622 29302 31397 31871 32457
15	44	5	0.762076 (0.712837)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3879 4030 4096 7913 8192 8508 9215 12422 12617 13147 14876 16384 16975 17378 17799 18843 21973 23393 24092 25048 26321 26409 27411 27683 28092 28622 29302 30283 31397 31871
15	45	5	0.773181 (0.721299)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3879 4030 4096 7913 8192 8508 9215 12422 12617 13147 14876 16384 16975 17378 17799 18843 21973 23393 24092 25048 26321 26409 27411 27683 28092 28622 29302 30283 31397 31871 32758
15	46	5	0.783918 (0.729129)	1 2 4 8 16 32 64 126 128 256 512 1024 2048 3879 4096 7099 7571 7913 8192 9215 13177 15602 16384 16975 17378 18843 20121 21138 21973 22445 22735 23393 24092 25144 26321 26409 27411 27673 27396 27411 28092 28622 29302 31397 31871 32457
15	47	5	0.794221 (0.736830)	1 2 4 8 16 32 64 126 128 151 256 512 1024 1349 2048 3879 4096 7913 8192 9215 12183 16207 16384 16975 17378 18843 19400 20477 21021 21530 21973 22762 23393 24092 25438 24092 26321 26409 27411 28092 28622 29302 29873 31199 31397 31871 32620

(Continues on next page)

Table A.1: *Continued from previous page*

m	n	d	Eqv. rate	Packed integer parity check matrix
15	48	5	0.804141 (0.744035)	1 2 4 8 16 32 64 126 128 256 512 826 1024 2048 3879 4096 7163 7913 8192 9215 9632 10552 16384 16975 17378 17779 18843 19664 21136 21973 22578 23393 24092 24495 25144 26321 26409 26640 26663 27411 28092 28622 29302 29977 31397 31871 32395 32607
15	49	5	0.813668 (0.772691)	1 2 4 8 16 32 64 128 256 512 1024 2048 3879 4096 5541 7031 7160 7913 8192 9215 13987 14289 16384 16975 17378 17579 18413 18843 18960 19350 19955 21973 23259 23393 24092 24495 25609 25698 26321 26409 27411 28092 28133 28622 28816 31397 31675 31871 32153
15	50	5	0.822836 (0.780014)	1 2 4 8 16 32 64 128 256 512 1024 2048 3879 4096 5541 7031 7160 7913 8192 9215 9683 10365 13987 16384 16975 17378 17579 18843 18960 19350 19955 21973 22294 23259 23393 24092 24495 25609 25698 25850 26321 26409 27411 28092 28622 28816 31397 31675 31871 32153
15	51	5	0.831624 (0.787048)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3879 4030 4096 4956 6925 7913 8192 8508 9215 12422 12617 14876 16384 16975 17378 17799 18164 18843 20207 20569 21973 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	52	5	0.840045 (0.793774)	1 2 4 8 16 32 64 126 128 151 256 512 1024 1349 2048 3879 4096 4938 7913 8192 9215 12183 15202 16207 16384 16975 17378 17579 18843 19400 20477 20900 21021 21530 21973 22762 23393 24092 24260 25438 26077 26321 26409 27411 28092 28622 29302 29873 31199 31397 31871 32620
15	53	5	0.848118 (0.800219)	1 2 4 8 16 32 64 126 128 151 256 512 1024 2048 3879 4096 4797 5541 7031 7160 7913 8192 9215 13987 14289 14371 16384 16975 17378 17579 18180 18293 18843 19400 20477 20900 21021 21530 21973 22537 23259 23393 24092 24495 25609 25698 26321 26409 27411 28092 28133 28622 28816 31397 31871 32153
15	54	5	0.855849 (0.806378)	1 2 4 8 16 32 64 126 128 256 512 1024 2048 3691 3879 4096 5541 6315 7031 7160 7489 7913 8192 9215 13987 14289 16384 16653 16975 17378 17579 18843 19400 20477 20900 21021 21530 21973 23393 24092 24495 25609 25698 26321 26409 27411 27517 28092 28133 28622 28816 30717 31397 31675 31871 32153
15	55	5	0.863253 (0.812037)	1 2 4 8 16 32 64 126 128 235 256 512 1024 1914 2048 3879 4030 4096 4956 6743 6925 7913 8192 8508 9215 12422 12617 13288 14876 15371 16384 16975 17378 17799 18164 18843 20207 20569 21973 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754
15	56	5	0.870337 (0.817515)	1 2 4 8 16 32 64 126 128 235 256 512 1024 1914 2048 3879 4030 4096 4956 6743 6925 7913 8192 8508 9215 12422 12617 13288 13767 14876 15371 16384 16975 17378 17799 18164 18843 20207 20569 21973 23393 24092 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754

(Continues on next page)

Table A.1: *Continued from previous page*

m	n	d	Eqv.	rate	Packed integer parity check matrix														
15	57	5	0.877108 (0.822674)	1 2 4 8 16 32 64 126 128 235 256 512 1024 1914 2048 3879 4030 4096 4956 6743 6925 7913 8192 8508 9215 12422 12617 13288 13767 14876 15371 16384 16975 17378 17799 18164 18358 18843 20207 20569 21973 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754															
15	58	5	0.883542 (0.827531)	1 2 4 8 16 32 64 126 128 235 256 512 1024 1914 2048 3879 4030 4096 4956 6743 6925 7913 8192 8508 9215 12422 12617 13288 13767 14876 15371 16384 16975 17378 17799 18164 18358 18843 20207 20569 21973 23393 23724 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 31397 31651 31871 32754															
15	59	5	0.889691 (0.832243)	1 2 4 8 16 32 64 126 256 512 1024 2048 3879 4096 4664 4797 5541 6674 7031 7160 7870 7913 8192 9215 13987 14289 14371 16384 16975 17378 17579 18180 18293 18843 18960 19350 21973 22537 23259 23393 24092 24495 25609 25698 26321 26409 26993 27411 27570 28092 28133 28622 28816 31397 31675 31871 32153 32484															
15	60	5	0.895581 (0.836659)	1 2 4 8 16 32 64 126 256 512 1024 2048 3879 4096 4664 4797 5541 6674 7031 7160 7870 7913 8192 9215 10420 13987 14289 14371 16384 16975 17378 17579 18180 18293 18843 18960 19350 19955 21973 22537 23259 23393 24092 24495 25609 25698 26321 26409 26993 27411 27570 28092 28133 28622 28816 31397 31675 31871 32153 32484															
15	61	5	0.901181 (0.84071)	1 2 4 8 16 32 64 126 256 512 1024 2048 3879 4096 4664 4797 5541 6674 7031 7160 7870 7913 8192 9215 10420 13987 14289 14371 16384 16975 17378 17579 17795 18180 18293 18843 18960 19350 19955 21973 22537 23259 23393 24092 24495 25609 25698 26321 26409 26993 27411 27570 28092 28133 28622 28816 31397 31675 31871 32153 32484															
15	62	5	0.906529 (0.84471)	1 2 4 8 16 32 64 126 128 235 256 512 1024 1914 2048 3879 4030 4096 4956 6743 6925 7913 8192 8508 9215 10638 12422 12617 13288 13767 14876 15371 16384 16975 17378 17798 18164 18358 18843 20207 20569 21973 23393 23724 23855 24092 25048 26321 26409 27258 27411 27683 28092 28622 29163 29302 30283 31397 31651 31871 32754															
15	63	5	0.911620 (0.848502)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3879 4030 4096 4956 6743 6925 7913 8192 8508 9215 8192 8508 9215 10627 11106 12422 12617 14876 15371 16384 16975 17378 17799 18164 18358 18843 20207 18843 20207 20569 21973 22743 23393 24092 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 30961 31397 31651 31871 32754															

(Continues on next page)

Table A.1: *Continued from previous page*

m	n	d	Eqv.	rate	Packed integer parity check matrix
15	64	5 (0.852014)	0.916471 (0.855343)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3940 4030 4096 4956 5359 5368 6256 6454 6925 7913 8192 8508 9215 10627 11106 12422 12617 14876 15254 16384 16604 16975 17378 17799 18164 18324 18843 20207 20569 21973 22743 23393 24092 25048 26321 26409 27258 27411 27683 28092 29302	
15	65	5	0.921080 (0.855343)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3940 4030 4096 4956 5359 5368 6256 6454 6925 7913 8192 8508 9215 10627 11106 12422 12617 14876 15254 16384 16604 16975 17378 17799 18164 18324 18700 18843 20207 20569 21973 22743 23393 24092 25048 26321 26409 27258 27411 27683 28092	
15	66	5	0.925470 (0.858469)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3940 4030 4096 4956 5368 6256 6454 6925 7913 8192 8508 9215 9264 10627 11106 12422 12617 14876 15254 16384 16604 16975 17378 17799 18164 18324 18843 20207 20569 21973 22743 23393 24092 25048 25148 26321 26409 27258 27411 27683 28092	
15	67	5	0.929650 (0.861375)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3940 4030 4096 4956 5368 6256 6454 6925 7913 8192 8508 9215 9264 10627 11106 12422 12617 14876 15254 16384 16604 16975 17378 17799 18164 18324 18843 20207 20569 21522 21973 22743 23393 24092 25048 25148 26321 26409 27258 27411 27683	
15	68	5	0.933609 (0.873716)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3940 4030 4096 4956 5368 6256 6454 6925 7913 8192 8508 9215 9264 10574 10627 11106 12422 12617 14876 15254 16384 16604 16975 17378 17799 18164 18324 18843 20207 20569 21554 21973 22743 23393 24092 25048 25148 26321 26409 27258 27411	
15	69	5	0.937373 (0.883069)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3940 4030 4096 4956 5368 6256 6454 6925 7913 8192 8508 9215 9264 10627 11106 12422 12617 14876 15254 16384 16604 16975 17378 17799 18164 18324 18843 20207 20569 21522 21973 22743 23393 24092 25048 25148 26321 26409 27211 27258	
15	70	5	0.940950 (0.890594)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 3940 4030 4096 4956 5359 5368 6256 6454 6861 6925 7520 7913 8192 8508 9215 10627 11106 12422 12617 14876 15254 16082 16384 16604 16975 17378 17758 17799 18164 18324 18843 20207 20569 21973 22743 23393 24092 25048 25148 26321 26409 27211 27258	
15				27411 27683 28092 28622 29302 30283 30961 31397 31651 31871 32754	

(Continues on next page)

Table A.1: *Continued from previous page*

m	n	d	Eqv.	rate	Packed integer parity check matrix
15	71	5	0.944327 (0.896977)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 4030 4096 4956 5359 5368 6256 6454 6861 6925 7520 7913 8192 8508 9215 10627 11106 12422 12617 13924 14876 15254 16082 16384 16604 16975 17378	
15	72	5	0.947533 (0.902252)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 2697 3145 3879 4030 4096 4956 5359 5368 6256 6454 6861 6925 7520 7913 8192 8508 9215 10627 11106 12422 12617 14876 15254 16082 16384 16604 16975 17378 17758 17799 18164 18324 18700 18843 20207 20569 21973 22743 23393 24011 24092 24326 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 30961 31397 31651 31871 32754	
15	73	5	0.950577 (0.906807)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 4030 4096 4956 5359 5368 6256 6454 6861 6925 7520 7913 8192 8508 9215 10627 11106 11606 12422 12617 14876 15254 16082 16384 16604 16975 17378 17758 17799 18164 18324 18700 18843 20207 20569 21973 22743 23393 24011 24092 24326 25048 25968 26239 26321 26409 27258 27411 27683 28092 28622 29302 30283 30961 31397 31651 31871 32754	
15	74	5	0.953458 (0.910715)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 4030 4096 4956 5359 5368 6256 6454 6564 6861 6925 7520 7913 8192 8508 9215 10627 11106 11606 12422 12617 14876 15254 16082 16384 16604 16975 17378 17758 17799 18164 18324 18700 18843 20207 20569 21973 22743 23393 24011 24092 24326 25048 25968 26239 26321 26409 27258 27411 27683 28092 28622 29302 30283 30961 31397 31651 31871 32754	
15	75	5	0.956184 (0.914168)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 4030 4096 4956 5359 5368 6256 6454 6564 6861 6925 7520 7913 8192 8508 9215 10627 11106 11606 12422 12617 14876 15254 16082 16384 16604 16975 17378 17758 17799 18164 18324 18700 18843 20207 20569 21973 22743 23393 24011 24092 24326 25048 25968 26239 26321 26409 27258 27411 27683 28092 28622 29302 30283 30961 31397 31651 31871 32133 32754	
15	76	5	0.958767 (0.91624)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 2697 3145 3726 3879 4030 4096 4109 4956 5359 5368 6256 6454 6861 6925 7520 7913 8192 8508 9215 10627 1106 12422 12617 14876 15254 16082 16384 16604 16975 17378 17758 18164 18324 18700 18843 18164 18324 18700 18843 18889 20207 20569 21973 22743 23393 24011 24092 24326 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 30961 31397 31651 31871 32007 32754	

(Continues on next page)

Table A.1: *Continued from previous page*

m	n	d	Eqv. rate	Packed integer parity check matrix
15	77	4	0.961205 (0.919134)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 4030 4096 4956 5359 5368 6256 6454 6564 6861 6925 7520 7913 8192 8508 9215 10627 11106 11606 12422 12617 13285 14876 15254 16082 16384 16604 16975 17378 17758 1799 18164 18324 18700 18843 20207 20569 21973 22743 23393 24011 24092
				24326 25048 25968 26239 26321 26409 27258 27411 27683 28092 28622 29302 30283 30961 31397 31651 31871 32133 32754
15	78	5	0.963517 (0.92183)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 4030 4096 4956 5359 5368 6256 6454 6564 6861 6925 7520 7913 8192 8508 9215 10627 11106 11606 12422 12617 13285 13685 13924 14876 15254 16082 16384 16604 16975 17378 17758 1799 18164 18324 18700 18843 20207 20569 21973 22743 23393 24011 24092 24326 25048 25968 26239 26321 26409 27258 27411 27683 28092 28622 29302 30283 30961 31397 31651 31871 32133 32754
15	79	5	0.963696 (0.924199)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 2697 3145 3726 3879 4030 4096 4956 5359 5368 6256 6454 6256 6454 6861 6925 7254 7520 7913 8192 8508 9215 10627 10636 11106 12422 12617 14876 15254 16082 16384 16578 16604 16975 17378 17758 1799 18164 18324 18700 18843 18889 20207 20569 21973 22743 23393 24011 24092 24326 25048 26321 26409 27258 27411 27683 28092 28622 29302 30283 30961 31397 31651 31871 32007 32616 32754
15	80	5	0.967762 (0.926307)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 4030 4096 4956 5052 5359 5368 6256 6454 6564 6861 6925 7520 7913 8192 8508 9215 10627 11106 11606 12422 12617 13285 13685 14876 15254 15540 16082 16384 16604 16975 17378 17758 1799 18164 18324 18700 18843 20207 20569 21973 22743 23393 24011 24092 24326 25048 25358 25968 26239 26321 26409 27258 27411 27683 28092 28622 29302 30283 30961 31397 31651 31871 32133 32754
15	81	5	0.9699716 (0.928241)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3879 4030 4096 4956 5359 5368 6256 6454 6861 6925 7520 7913 8192 8508 8797 9215 10627 11106 11606 12422 12617 14876 15254 15644 16082 16384 16604 16975 17378 17758 1799 18164 18267 18324 18700 18843 18955 20207 20569 21824 21973 22743 23393 24011 24092 24326 25048 25968 26321 26409 27258 27411 27683 28092 28622 29302 30283 30961 31397 31651 31703 31871 32754
15	82	5	0.971556 (0.92999)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3429 3879 4030 4096 4956 5359 5368 6256 6454 6861 6925 7520 7913 8192 8508 8797 9215 10627 11106 11606 12422 12617 14876 15254 15644 16082 16384 16604 16975 17378 17758 1799 18164 18267 18324 18700 18843 18955 20207 20569 21824 21973 22743 23393 24011 24092 24326 25048 25968 26321 26409 27258 27411 27683 28092 28622 29302

(Continues on next page)

Table A.1: *Continued from previous page*

m	n	d	Eqv. rate	Packed integer parity check matrix
15	83	5	0.973296 (0.973189)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3429 3879 4030 4096 4956 5359 5368 6256 6454 6861 6925 7520 7913 8192 8508 8797 9215 10627 11106 11606 12422 12617 14876 15254 15644 16082 16384
				16604 16975 17378 17758 17799 18164 18267 18324 18700 18843 18955 20207 20569 21824 21973 22743
				23393 24011 24092 24326 25048 25968 26321 26409 26764 27258 27411 27683 27992 28092 28176 28622
15	84	5	0.974936 (0.974850)	29302 30283 30961 31397 31651 31703 31871 32709 32754
15	85	4	0.976482 (0.976415)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3429 3879 4030 4096 4956 5359 5368 5751 6256 6454 6861 6925 7520 7913 8192 8508 8797 9215 10627 11106 11606 12422 12617 14876 15254 15644 16082
				16384 16604 16975 17378 17758 17799 18164 18267 18324 18700 18843 18955 20207 20569 21824 21973
				22743 23393 24011 24092 24326 25048 25968 26321 26409 26764 27258 27411 27683 27992 28092 28622
15	86	5	0.977941 (0.977893)	29302 30042 30283 30961 31397 31651 31703 31871 32105 32709 32754
15	87	5	0.979315 (0.979288)	1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 2820 3145 3429 3879 4030 4096 4956 5359 5368 6256 6454 6454 6861 6925 7520 7913 8192 8508 8797 9215 10627 11106 11606 11707 12016 12422 12617 14876
				15254 15644 16082 16384 16604 16975 17378 17758 17799 18164 18267 18324 18700 18843 20207 20569
				21824 21973 22743 23393 23960 24011 24092 24326 25048 25968 26321 26409 26764 26993 27258 27411
				27683 27992 28092 28622 29302 30283 30961 31397 31651 31703 31871 32105 32754
15	88	5	0.980606 (0.980601)	27992 28092 28622 29302 29669 30283 30961 31397 31651 31703 31871 32105 32754
				1 2 4 8 16 32 64 126 128 256 512 1024 1914 2048 3145 3429 3879 4030 4096 4956 5359 5368 6256 6454 6861 6925 7520 7913 8192 8508 8797 9215 10194 10627 11106 11606 11707 12016 12422 12617 14876 15254
				15644 16082 16384 16604 16975 17378 17758 17799 18164 18267 18324 18700 18843 20207 20569 21824
				21973 22743 23393 23960 24011 24092 24326 25048 25968 26321 26409 26764 26993 27258 27411 27683
				27683 27992 28092 28622 29302 29669 30283 30961 31397 31651 31703 31871 32105 32754

Appendix B

Best Known Equivocation Codes of the Highest Minimum Distance

Table B.1: BE_qC and BKC (in parentheses) Table

m	n	d	Eqv.rate	Packed integer parity check matrix
7	8	8	0.327237 (0.327237)	1 2 4 8 16 32 64 127
7	9	6	0.367001 (0.367000)	1 2 4 8 16 32 64 115 124
7	10	5	0.405261 (0.404808)	1 2 4 8 16 32 63 64 115 124
7	11	5	0.443011 (0.443011)	1 2 4 8 16 32 63 64 85 115 124
7	12	4	0.478897 (0.461720)	1 2 4 8 16 26 32 63 64 85 115 124
7	13	4	0.512617 (0.491069)	1 2 4 8 16 32 63 64 79 85 86 115 124
7	14	4	0.545891 (0.518182)	1 2 4 8 16 32 63 64 79 83 90 107 118 125
7	15	4	0.575360 (0.543464)	1 2 4 7 8 16 32 63 64 79 83 90 107 118 125
7	16	4	0.604499 (0.566880)	1 2 4 8 16 21 26 32 41 63 64 79 83 99 100 125
7	17	4	0.631806 (0.588469)	1 2 4 7 8 16 32 42 51 63 64 79 83 90 107 118 125
7	18	4	0.658043 (0.618943)	1 2 4 7 8 16 30 32 42 51 63 64 79 83 90 107 118 125
7	19	4	0.682022 (0.644628)	1 2 4 8 16 27 28 32 41 46 63 64 74 87 89 109 115 122 124
				(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
7	20	4	0.704918 (0.662960)	1 2 4 8 16 27 28 32 41 46 63 64 74 79 87 89 109 115 122 124
7	21	4	0.725116 (0.684509)	1 2 4 8 14 16 21 32 35 41 58 63 64 74 79 83 88 100 107 118 125
7	22	4	0.745065 (0.703436)	1 2 4 7 8 11 16 26 32 41 53 63 64 74 83 93 94 101 111 112 121 124
7	23	4	0.763451 (0.720320)	1 2 4 7 8 11 16 26 32 41 53 63 64 74 83 93 94 101 111 112 118 121 124
7	24	4	0.781028 (0.735442)	1 2 4 8 11 14 16 26 29 32 47 51 57 62 64 74 83 85 92 103 104 112 118 127
7	25	4	0.797310 (0.749785)	1 2 4 7 8 16 30 32 46 51 53 58 63 64 70 75 83 93 97 100 106 111 112 118 123
7	26	4	0.812459 (0.762700)	1 2 4 8 11 14 16 26 29 32 47 51 57 62 64 74 77 83 85 92 100 103 104 112 118 127
7	27	4	0.826914 (0.774429)	1 2 4 8 11 14 16 26 29 32 47 51 57 62 64 74 77 83 85 92 97 100 103 104 112 118 127
7	28	4	0.840506 (0.785083)	1 2 4 8 11 14 16 26 29 32 38 47 51 57 62 64 74 77 83 85 92 97 100 103 104 112 118 127
7	29	4	0.851506 (0.794959)	1 2 4 8 11 14 16 26 29 32 38 47 51 57 62 64 74 77 83 85 92 97 100 103 104 110 112 118 127
7	30	4	0.854176 (0.803969)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 73 79 119 123 125
7	31	4	0.867182 (0.812237)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 69 73 79 119 123 125
7	32	4	0.879097 (0.819818)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 69 73 79 114 119 123 125
7	33	4	0.886455 (0.826760)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 74 82 88 98 100 103 115 127
7	34	4	0.895741 (0.852821)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 74 82 84 88 98 100 115 122 127
7	35	4	0.904253 (0.872739)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 74 82 84 88 98 100 103 115 122 127
7	36	4	0.912049 (0.888646)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 67 74 82 84 88 98 100 103 115 122 127
7	37	4	0.919097 (0.901769)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 67 74 76 82 84 88 98 100 103 115 122 127
7	38	4	0.925583 (0.912693)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 67 74 76 82 84 88 97 98 100 103 115 122 127
7	39	4	0.931601 (0.921923)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 67 74 76 82 84 88 91 97 98 100 103 115 122 127
7	40	4	0.937142 (0.929789)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 64 67 74 76 82 84 88 91 97 98 100 103 115 118 122 127
7	41	4	0.942230 (0.936808)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 59 64 67 74 76 82 84 88 91 97 98 100 103 115 118 122 127

(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
7	42	4	0.946878 (0.942873)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 59 64 67 74 76 79 82 84 88 91 97 98 100 103 115 118 122 127
7	43	4	0.951182 (0.948166)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 59 64 67 70 74 76 82 84 88 91 97 98 100 103 110 115 118 122 127
7	44	4	0.955172 (0.952803)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 59 64 67 74 76 79 82 84 88 91 97 98 100 103 104 115 117 118 122 127
7	45	4	0.958863 (0.956930)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 59 62 64 67 74 76 79 82 84 88 91 97 98 100 103 104 115 117 118 122 127
7	46	4	0.962228 (0.960586)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 59 62 64 67 74 76 79 82 84 87 88 91 97 98 100 103 104 115 117 118 122 127
7	47	4	0.965345 (0.963848)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 59 62 64 67 70 76 79 84 87 91 93 94 98 103 107 109 110 115 117 121 122 124 127
7	48	4	0.968205 (0.966766)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 52 55 56 64 67 70 73 76 79 82 87 88 93 94 97 98 100 104 110 112 115 121 122 124
7	49	4	0.970856 (0.969876)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 42 44 47 49 59 61 62 64 67 70 73 76 79 84 87 91 93 94 98 103 107 109 110 115 117 121 122 124 127
7	50	4	0.973272 (0.972635)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 59 62 64 67 70 74 76 79 82 84 87 88 91 94 97 98 100 103 104 115 117 118 121 122 127
7	51	4	0.975516 (0.975095)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 56 59 62 64 67 70 74 76 79 82 84 87 88 91 94 97 98 100 103 104 115 117 118 121 122 127
7	52	4	0.977571 (0.977298)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 52 59 62 64 67 73 74 76 79 82 84 87 88 91 94 97 98 100 103 104 107 115 117 118 121 122 127
7	53	4	0.979454 (0.979291)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 52 59 62 64 67 73 74 76 79 82 84 87 88 91 94 97 98 100 103 104 107 115 117 118 121 122 124 127
7	54	4	0.981189 (0.981087)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 50 52 56 59 62 64 67 69 70 74 76 79 82 84 87 88 91 94 97 98 100 103 104 115 117 118 121 122 127
7	55	4	0.982787 (0.982712)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 50 52 56 59 62 64 67 69 70 74 76 79 82 84 87 88 91 94 97 98 100 103 104 115 117 118 121 122 124 127
7	56	4	0.984253 (0.984185)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 50 52 55 56 59 62 64 67 69 73 74 76 81 82 84 87 88 91 93 97 98 100 103 104 110 112 115 117 122 124 127
				(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
7	57	4	0.985599 (0.985569)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 50 52 55 56 59 61 62 64 67 69 70 73 74 76 81 82 84 88 91 93 97 98 100 103 104 110 112 115 117 122 124 127
7	58	4	0.986835 (0.986824)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 50 52 55 56 59 62 64 67 69 70 74 76 79 81 82 84 87 88 91 93 94 97 98 100 103 104 115 117 118 121 122 124 127
7	59	4	0.987972 (0.987968)	1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38 41 42 44 47 49 50 52 55 56 59 61 62 64 67 69 70 74 76 79 81 82 84 87 88 91 93 94 97 98 100 103 104 115 117 118 121 122 124 127
11	12	12	0.312421 (0.312421)	1 2 4 8 16 32 64 128 256 512 1024 2047
11	13	8	0.338382 (0.338381)	1 2 4 8 16 32 64 128 256 415 512 1024 1786
11	14	8	0.364203 (0.364202)	1 2 4 8 16 32 64 128 256 415 512 875 1024 1710
11	15	8	0.389947 (0.389947)	1 2 4 8 16 32 64 128 256 415 512 875 1024 1143 1710
11	16	8	0.415516 (0.415515)	1 2 4 8 16 32 64 128 256 415 512 875 1024 1143 1710 2002
11	17	7	0.440378 (0.440234)	1 2 4 8 16 32 64 128 237 256 415 512 875 1024 1143 1710 2002
11	18	7	0.465075 (0.465038)	1 2 4 8 16 32 64 128 237 256 415 512 606 875 1024 1143 1710 2002
11	19	7	0.489476 (0.489476)	1 2 4 8 16 32 64 126 128 256 415 512 875 972 1024 1325 1629 1710 2039
11	20	7	0.513477 (0.513476)	1 2 4 8 16 32 64 126 128 256 415 512 875 1024 1227 1325 1528 1629 1710 2039
11	21	7	0.537014 (0.537014)	1 2 4 8 16 32 64 126 128 256 415 512 697 875 1024 1227 1325 1528 1629 1710 2039
11	22	7	0.560019 (0.560019)	1 2 4 8 16 32 64 126 128 256 415 512 697 875 972 1024 1227 1325 1528 1629 1710 2039
11	23	7	0.582470 (0.582470)	1 2 4 8 16 32 64 126 128 256 415 512 697 875 972 1024 1227 1325 1528 1629 1710 1818 2039
11	24	6	0.596254 (0.592625)	1 2 4 8 16 32 64 110 128 179 256 445 491 512 527 731 749 793 803 890 908 982 1024 2033
11	25	6	0.618532 (0.617354)	1 2 4 8 16 32 64 128 253 256 319 506 512 638 671 682 847 935 979 1001 1012 1024 1441 1645 1881
11	26	6	0.638561 (0.637822)	1 2 4 8 16 32 64 128 253 256 319 506 512 638 671 682 847 935 979 1001 1012 1024 1441 1645 1729 1881
11	27	6	0.657674 (0.657212)	1 2 4 8 16 32 64 128 253 256 319 506 512 638 671 682 847 935 979 1001 1012 1024 1130 1441 1595 1645 1881
11	28	6	0.676027 (0.675605)	1 2 4 8 16 32 64 128 253 256 319 506 512 638 671 682 847 935 979 1001 1012 1024 1130 1441 1595 1645 1832 1881
				(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
11	29	6	0.693663 (0.693319)	1 2 4 8 16 32 64 128 253 256 319 506 512 638 671 682 847 935 979 1001 1012 1024 1130 1441 1595 1645 1718 1832 1881
11	30	6	0.710469 (0.710225)	1 2 4 8 16 32 64 128 253 256 319 506 512 638 671 682 847 935 979 1001 1012 1024 1081 1390 1441 1645 1797 1832 1881 1979
11	31	6	0.726603 (0.726505)	1 2 4 8 16 32 64 128 253 256 319 341 506 512 638 671 682 847 935 979 1001 1012 1024 1123 1144 1243 1485 1718 1797 1881 1888
11	32	6	0.742144 (0.742073)	1 2 4 8 16 32 64 128 253 256 319 341 512 638 671 682 847 935 979 1001 1012 1024 1123 1144 1243 1322 1468 1485 1718 1881 1888 2030
11	33	6	0.757049 (0.757028)	1 2 4 8 16 32 64 128 253 256 319 341 512 638 671 682 847 935 979 1001 1012 1024 1123 1144 1243 1322 1468 1485 1718 1797 1881 1888 2030
11	34	6	0.771280 (0.771280)	1 2 4 8 16 32 64 128 253 256 319 341 506 512 638 671 682 847 935 979 1001 1012 1024 1123 1144 1243 1322 1468 1485 1718 1797 1881 1888 2030
11	35	5	0.783901 (0.775716)	1 2 4 8 16 32 64 128 253 256 319 341 506 512 638 671 682 847 935 979 1001 1012 1024 1100 1123 1144 1243 1322 1468 1485 1718 1797 1881 1888 2030
11	36	5	0.796047 (0.790051)	1 2 4 8 16 32 64 128 253 256 319 341 506 512 638 671 682 847 935 979 1001 1012 1024 1100 1123 1144 1243 1322 1468 1485 1512 1718 1797 1881 1888 2030
11	37	5	0.807675 (0.803255)	1 2 4 8 16 32 64 128 253 256 319 341 420 456 506 512 638 671 682 847 935 979 1001 1012 1024 1100 1123 1144 1243 1322 1468 1485 1718 1797 1881 1888 2030
11	38	5	0.818806 (0.815439)	1 2 4 8 16 32 64 128 253 256 319 341 420 506 512 638 671 682 711 847 935 979 1001 1012 1024 1100 1106 1123 1144 1243 1322 1468 1485 1718 1797 1881 1888 2030
11	39	5	0.829343 (0.826902)	1 2 4 8 16 32 64 128 253 256 319 341 402 420 506 512 638 671 682 822 847 935 979 1001 1012 1024 1100 1123 1144 1243 1322 1468 1485 1718 1797 1853 1881 1888 2030
11	40	5	0.839390 (0.837627)	1 2 4 8 16 32 64 128 253 256 319 341 420 506 512 638 671 682 822 847 935 945 979 1001 1012 1024 1100 1123 1144 1243 1322 1468 1485 1601 1718 1797 1853 1881 1888 2030
11	41	5	0.848951 (0.847670)	1 2 4 8 16 32 64 128 253 256 319 341 420 506 512 549 638 671 682 822 847 935 945 979 1001 1012 1024 1100 1123 1144 1243 1322 1468 1485 1601 1718 1797 1853 1881 1888 2030
11	42	5	0.857378 (0.857105)	1 2 4 8 16 32 64 128 209 237 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1111 1361 1382 1403 1480 1590 1837 1874 1898 1934 1952
11	43	5	0.866258 (0.866055)	1 2 4 8 16 32 64 128 209 237 247 256 381 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1073 1111 1295 1361 1382 1403 1475 1837 1874 1898 1934 1952
				(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
11	44	5	0.874565 (0.874446)	1 2 4 8 16 32 64 128 209 237 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1073 1111 1315 1361 1382 1403 1480 1590 1837 1874 1898 1934 1952
11	45	5	0.882433 (0.882366)	1 2 4 8 16 32 64 128 209 237 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1073 1111 1295 1315 1361 1382 1403 1480 1590 1837 1874 1898 1934 1952
11	46	5	0.889855 (0.889847)	1 2 4 8 16 32 64 128 209 247 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1073 1111 1295 1315 1361 1382 1403 1475 1480 1590 1837 1874 1898 1934 1952
11	47	5	0.896909 (0.896905)	1 2 4 8 16 32 64 128 209 237 247 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1073 1111 1295 1315 1361 1382 1403 1475 1480 1590 1837 1874 1898 1934 1952
11	48	4	0.903225 (0.710484)	1 2 4 8 16 32 64 128 209 237 247 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1073 1111 1295 1315 1361 1382 1403 1434 1475 1480 1590 1837 1874 1898 1934 1952
11	49	4	0.909219 (0.713286)	1 2 4 8 16 32 64 128 209 237 247 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1073 1111 1295 1315 1361 1382 1403 1475 1480 1590 1733 1808 1837 1874 1898 1934 1952
11	50	4	0.914902 (0.715762)	1 2 4 8 16 32 64 128 209 237 247 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1054 1073 1111 1295 1315 1361 1382 1403 1475 1480 1590 1710 1837 1874 1898 1934 1952 2045
11	51	4	0.920284 (0.717951)	1 2 4 8 16 32 64 128 209 237 247 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1073 1111 1150 1213 1295 1315 1361 1382 1403 1475 1480 1590 1733 1808 1837 1874 1898 1934 1952
11	52	4	0.925399 (0.719886)	1 2 4 8 16 32 64 128 209 237 247 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1073 1111 1128 1295 1315 1353 1361 1382 1403 1469 1475 1480 1590 1645 1837 1874 1898 1934 1952 2047
11	53	4	0.930200 (0.721705)	1 2 4 8 16 32 64 128 209 237 247 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1073 1111 1295 1312 1315 1361 1382 1403 1434 1475 1480 1590 1689 1733 1808 1837 1874 1898 1934 1952 2038
11	54	4	0.934726 (0.723311)	1 2 4 8 16 32 64 128 209 237 247 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1073 1111 1295 1306 1315 1361 1382 1403 1458 1475 1480 1561 1578 1590 1699 1796 1837 1874 1898 1934 1951 1952

(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
11	55	4	0.938998 (0.724755)	1 2 4 8 16 32 64 128 209 237 247 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1073 1111 1295 1306 1315 1361 1382 1403 1458 1475 1480 1561 1578 1590 1699 1796 1837 1874 1898 1934 1951 1952 2045
11	56	4	0.943023 (0.726067)	1 2 4 8 16 32 64 128 209 237 247 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1073 1111 1171 1295 1306 1315 1361 1382 1403 1458 1475 1480 1561 1578 1590 1699 1796 1837 1874 1898 1934 1951 1952 2045
11	57	4	0.946818 (0.727336)	1 2 4 8 16 32 64 128 209 237 247 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1073 1111 1141 1171 1295 1306 1315 1361 1382 1403 1458 1475 1480 1561 1578 1590 1699 1796 1837 1874 1898 1934 1951 1952 2045
11	58	4	0.950376 (0.728485)	1 2 4 8 16 32 64 128 209 237 247 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1029 1073 1111 1141 1171 1295 1306 1315 1361 1382 1403 1458 1475 1480 1561 1578 1590 1699 1796 1837 1874 1898 1934 1951 1952 2045
11	59	4	0.953600 (0.729528)	1 2 4 8 16 32 64 128 209 237 247 256 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1029 1073 1111 1141 1171 1295 1306 1315 1361 1382 1403 1458 1475 1480 1561 1578 1590 1635 1699 1796 1837 1874 1898 1934 1951 1952 2045
11	60	4	0.956635 (0.730463)	1 2 4 8 16 32 64 128 209 237 247 256 277 364 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 948 962 975 988 1024 1029 1073 1111 1141 1171 1295 1306 1315 1361 1382 1403 1458 1475 1480 1561 1578 1590 1699 1796 1837 1874 1898 1934 1951 1952 2045
11	61	4	0.959491 (0.731332)	1 2 4 8 16 32 64 128 209 237 247 256 277 364 381 389 418 443 474 481 494 512 595 669 762 778 836 873 886 899 948 962 975 988 1024 1029 1073 1111 1141 1171 1295 1306 1315 1361 1382 1403 1458 1475 1480 1561 1578 1590 1699 1796 1837 1874 1898 1934 1951 1952 2045
11	62	4	0.962176 (0.732129)	1 2 4 8 16 32 64 128 209 237 247 256 296 381 389 418 443 474 481 494 512 595 669 762 765 778 836 873 886 948 962 969 975 988 1024 1073 1111 1150 1213 1295 1301 1312 1315 1334 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1808 1817 1837 1874 1898 1934 1952 2038
11	63	4	0.964696 (0.732861)	1 2 4 8 16 32 64 128 209 237 247 251 256 381 389 418 443 444 474 481 494 512 595 669 683 762 778 818 836 862 873 886 948 962 975 988 1024 1029 1073 1111 1141 1171 1295 1306 1315 1361 1382 1403 1458 1475 1480 1561 1578 1590 1699 1796 1837 1874 1898 1934 1951 1952 2045

(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
11	64	4	0.967081 (0.733520)	1 2 4 8 16 32 64 128 209 222 237 247 251 256 269 381 389 418 443 474 481 494 512 538 595 669 762 778 836 873 886 899 948 962 975 988 1024 1029 1073 1111 1141 1171 1295 1306 1315 1361 1382 1403 1458 1475 1480 1561 1578 1590 1635 1699 1796 1837 1874 1898 1934 1951 1952 2045
11	65	4	0.969326 (0.734121)	1 2 4 8 16 32 64 128 209 222 237 247 251 256 381 389 418 443 444 474 481 494 512 538 595 669 683 762 778 818 836 862 873 886 948 962 975 988 1024 1029 1073 1111 1141 1171 1295 1306 1315 1361 1382 1403 1458 1475 1480 1561 1578 1590 1699 1796 1837 1874 1898 1934 1951 1952 2045
11	66	4	0.971429 (0.749506)	1 2 4 8 16 32 64 74 128 209 222 237 247 251 256 269 381 389 418 443 444 474 481 494 512 538 595 669 762 778 836 873 886 899 948 962 975 988 1024 1029 1073 1111 1141 1171 1295 1306 1315 1361 1382 1403 1458 1475 1480 1561 1578 1590 1635 1699 1796 1837 1874 1898 1934 1951 1952 2045
11	67	4	0.973399 (0.760622)	1 2 4 8 16 32 64 111 128 209 237 247 251 256 296 381 389 418 443 474 481 494 512 538 575 595 654 669 708 762 778 836 873 886 888 938 948 962 969 975 988 1024 1029 1073 1111 1150 1213 1295 1301 1312 1315 1334 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1808 1837 1874 1898 1934 1952 2038
11	68	4	0.975245 (0.761144)	1 2 4 8 16 32 64 74 111 128 209 222 237 247 251 256 381 389 418 443 444 474 481 494 512 538 595 669 683 762 778 818 836 862 873 886 899 948 962 975 988 1024 1029 1073 1111 1141 1171 1295 1306 1315 1361 1382 1403 1458 1475 1480 1561 1578 1590 1699 1796 1837 1874 1898 1934 1951 1952 2045
11	69	4	0.976930 (0.769587)	1 2 4 8 16 32 64 74 111 128 209 222 237 247 251 256 381 389 418 443 444 474 481 494 512 538 595 669 683 762 778 818 836 862 873 886 899 948 962 975 988 1024 1029 1073 1111 1141 1171 1221 1295 1306 1315 1361 1382 1403 1458 1475 1480 1561 1578 1590 1699 1796 1837 1874 1898 1934 1951 1952 2045
11	70	4	0.978503 (0.770032)	1 2 4 8 16 32 64 111 128 209 237 247 251 256 296 381 389 418 443 474 481 494 512 538 575 595 654 669 708 762 778 836 873 886 888 899 938 948 962 969 975 988 1024 1073 1111 1136 1150 1213 1295 1301 1312 1315 1334 1361 1382 1403 1434 1449 1475 1480 1590 1661 1689 1733 1808 1837 1874 1898 1934 1952 2038
11	71	4	0.979978 (0.770448)	1 2 4 8 16 32 64 74 111 128 209 222 237 247 251 256 381 389 418 443 444 474 481 494 512 538 595 669 683 762 778 818 836 862 873 886 899 948 962 975 988 1024 1029 1073 1111 1141 1171 1221 1295 1306 1315 1361 1382 1389 1403 1458 1475 1480 1561 1578 1590 1682 1699 1796 1837 1874 1898 1934 1951 1952 2045
				(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
11	72	4	0.981360 (0.776960)	1 2 4 8 16 32 64 111 128 209 237 247 256 296 381 389 418 443 474 481 494 512 538 575 595 654 669 708 762 765 778 836 861 873 886 899 948 962 969 975 988 1016 1024 1067 1073 1111 1150 1213 1295 1301 1312 1315 1334 1361 1382 1403 1434 1449 1475 1480 1542 1590 1663 1689 1733 1808 1837 1874 1898 1934 1952 2038
11	73	4	0.982653 (0.782101)	1 2 4 8 16 32 64 111 128 209 237 247 256 296 381 389 418 443 474 481 494 512 538 575 591 595 654 669 708 762 765 778 836 861 873 886 899 948 962 969 975 988 1016 1024 1067 1073 1111 1150 1213 1295 1301 1312 1315 1334 1361 1382 1403 1434 1449 1475 1480 1542 1590 1663 1689 1733 1808 1837 1874 1898 1934 1952 2038
11	74	4	0.983861 (0.786203)	1 2 4 8 16 32 64 111 128 209 237 247 256 296 381 389 418 443 474 481 494 512 538 575 591 595 654 669 708 762 765 778 836 861 873 886 899 948 962 969 975 988 1016 1024 1067 1073 1111 1150 1213 1295 1301 1312 1315 1334 1361 1382 1403 1434 1449 1475 1480 1542 1590 1626 1663 1689 1733 1808 1837 1874 1898 1934 1952 2038
11	75	4	0.984993 (0.789478)	1 2 4 8 16 32 64 111 128 209 237 247 256 296 381 389 418 443 474 481 494 512 538 575 595 654 669 708 762 765 778 836 861 873 886 899 948 962 969 975 988 1016 1024 1061 1067 1073 1111 1150 1213 1223 1295 1301 1312 1315 1334 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1798 1808 1837 1874 1883 1898 1934 1952 1990 2038
11	76	4	0.986053 (0.792140)	1 2 4 8 16 32 64 111 128 209 237 247 256 296 381 389 418 443 474 481 494 512 538 575 595 654 669 708 762 765 778 836 861 873 886 899 948 962 969 975 988 1016 1024 1061 1067 1073 1111 1116 1150 1213 1223 1295 1301 1312 1315 1334 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1798 1808 1837 1874 1883 1898 1934 1952 1990 2038
11	77	4	0.987039 (0.794298)	1 2 4 8 16 32 64 111 128 209 237 247 256 296 381 389 418 443 474 481 494 512 538 575 595 654 669 708 762 765 778 786 836 861 873 886 899 948 962 969 975 988 1016 1024 1061 1067 1073 1111 1116 1150 1213 1223 1295 1301 1312 1315 1334 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1798 1808 1837 1874 1883 1898 1934 1952 1990 2038
11	78	4	0.987964 (0.796060)	1 2 4 8 16 32 64 111 128 209 237 247 256 296 381 389 418 443 474 481 494 512 538 575 578 595 654 669 708 762 765 778 836 861 873 886 899 948 962 969 975 988 1016 1024 1061 1067 1073 1111 1116 1150 1213 1223 1269 1295 1301 1312 1315 1334 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1798 1808 1837 1874 1883 1898 1934 1952 1990 2038
				(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
11	79	4	0.988826 (0.797503)	1 2 4 8 16 32 64 111 128 209 237 247 256 296 381 389 418 439 443 474 481 494 512 538 575 578 595 654 669 708 762 765 778 836 861 873 886 899 948 962 969 975 988 1016 1024 1061 1067 1073 1111 1116 1150 1213 1223 1269 1295 1301 1312 1315 1334 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1798 1808 1837 1874 1883 1898 1934 1952 1990 2038
11	80	4	0.989632 (0.798687)	1 2 4 8 16 32 64 101 111 128 209 237 247 256 296 381 389 418 443 474 481 494 512 538 575 578 595 654 669 708 762 765 775 778 836 861 873 886 899 948 962 969 975 988 1016 1024 1061 1067 1073 1111 1116 1150 1213 1223 1295 1301 1304 1312 1315 1334 1353 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1798 1808 1837 1874 1898 1934 1952 1990 2038
11	81	4	0.990381 (0.799679)	1 2 4 8 16 32 64 101 111 128 209 237 247 256 296 381 389 418 443 474 481 494 512 538 575 578 595 596 654 669 708 762 765 775 778 836 861 873 886 899 948 962 969 975 988 1016 1024 1061 1067 1073 1111 1116 1150 1213 1223 1295 1301 1304 1312 1315 1334 1353 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1798 1808 1837 1874 1898 1934 1952 1990 2038
11	82	4	0.991082 (0.800500)	1 2 4 8 16 32 64 111 128 209 237 247 256 296 355 381 389 418 439 443 474 481 494 512 538 575 578 595 596 654 669 708 762 765 775 778 836 846 861 873 886 899 948 962 969 975 988 1016 1024 1061 1067 1073 1111 1116 1150 1213 1223 1295 1301 1312 1315 1334 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1798 1803 1808 1837 1874 1898 1934 1952 1990 2038
11	83	4	0.991733 (0.801182)	1 2 4 8 16 32 64 111 128 209 237 247 256 296 355 381 389 418 439 443 474 481 494 512 538 575 578 595 596 654 669 708 762 765 775 778 836 846 861 873 886 899 948 962 969 975 988 1016 1024 1031 1061 1067 1073 1111 1116 1150 1213 1223 1295 1301 1312 1315 1334 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1798 1803 1808 1837 1874 1898 1934 1952 1990 2038
11	84	4	0.992339 (0.801750)	1 2 4 8 16 32 64 111 128 209 237 247 256 296 355 381 389 418 439 443 474 481 494 512 538 575 578 595 596 654 669 694 708 762 765 775 778 786 836 846 861 873 886 899 948 962 969 975 988 1016 1024 1061 1067 1073 1111 1116 1150 1213 1223 1295 1301 1312 1315 1334 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1798 1803 1808 1837 1874 1898 1934 1952 1990 2038
				(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
11	85	4	0.992904 (0.802230)	1 2 4 8 16 32 64 111 128 209 237 247 256 296 355 381 389 418 439 443 474 481 494 512 538 575 578 595 596 654 669 694 708 762 765 775 778 786 836 846 861 873 886 899 948 962 969 975 988 1016 1024 1031 1061 1067 1073 1111 1116 1150 1213 1223 1295 1301 1312 1315 1334 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1798 1803 1808 1837 1874 1898 1934 1952 1990 2038
11	86	4	0.993428 (0.802631)	1 2 4 8 16 32 64 111 128 209 237 247 256 296 355 381 389 418 439 443 474 481 494 512 538 575 578 595 596 654 669 694 708 762 765 775 778 786 836 846 861 873 886 899 948 962 969 975 988 1016 1024 1031 1034 1061 1067 1073 1111 1116 1150 1213 1223 1295 1301 1312 1315 1334 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1798 1803 1808 1837 1874 1898 1934 1952 1990 2038
11	87	4	0.993915 (0.802969)	1 2 4 8 16 32 64 111 128 209 237 247 256 296 355 381 389 418 439 443 474 481 494 512 538 575 578 595 596 654 669 694 708 762 765 775 778 786 836 846 861 873 886 899 948 962 969 975 988 1016 1024 1031 1061 1067 1073 1111 1116 1150 1213 1223 1295 1301 1312 1315 1334 1357 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1766 1798 1803 1808 1837 1874 1898 1934 1952 1990 2038
11	88	4	0.994369 (0.803261)	1 2 4 8 16 32 64 101 111 128 209 212 216 237 247 256 296 302 381 389 418 439 443 474 481 494 512 538 575 578 595 596 654 669 694 708 762 765 775 778 836 846 861 873 886 899 948 962 969 975 988 1016 1024 1031 1061 1067 1073 1111 1116 1150 1213 1223 1295 1301 1312 1315 1334 1353 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1787 1798 1808 1837 1874 1898 1934 1952 1990 2038
11	89	4	0.994791 (0.803510)	1 2 4 8 16 32 64 101 111 128 209 212 216 237 247 249 256 296 302 381 389 418 439 443 474 481 494 512 538 575 578 595 596 654 669 708 762 765 775 778 836 846 861 873 886 899 948 962 969 975 988 1016 1024 1031 1061 1067 1073 1111 1116 1150 1213 1223 1295 1301 1304 1312 1315 1334 1353 1361 1382 1403 1434 1449 1475 1480 1590 1689 1733 1787 1798 1808 1837 1874 1898 1934 1952 1990 2038
				(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
12	13	13	0.310250 (0.310250)	1 2 4 8 16 32 64 128 256 512 1024 2048 4095
12	14	9	0.334077 (0.334074)	1 2 4 8 16 32 64 128 256 512 1024 2048 1533 2907
12	15	8	0.357807 (0.357728)	1 2 4 8 16 32 64 128 256 512 1024 2048 951 1757 2159
12	16	8	0.381489 (0.381317)	1 2 4 8 16 32 64 128 256 512 1024 2048 1723 1996 2492 2907
12	17	8	0.405042 (0.404863)	1 2 4 8 16 32 64 128 256 512 1024 2048 415 743 1513 2414 3891
12	18	8	0.428189 (0.429189)	1 2 4 8 16 32 64 128 256 512 1024 2048 446 847 3032 3434 3839 3974
12	19	8	0.451301 (0.451295)	1 2 4 8 16 32 64 128 256 512 1024 2048 415 878 1259 1629 1976 3731 3812
12	20	8	0.474157 (0.474152)	1 2 4 8 16 32 64 128 256 512 1024 2048 415 878 1259 1629 1976 3399 3731 3812
12	21	8	0.496720 (0.496717)	1 2 4 8 16 32 64 128 256 512 1024 2048 127 1778 1815 2715 2929 3285 3418 3644 4079
12	22	8	0.518970 (0.518970)	1 2 4 8 16 32 64 128 256 512 1024 2048 127 1778 1815 2486 2715 2929 3285 3418 3644 4079
12	23	8	0.540878 (0.540878)	1 2 4 8 16 32 64 128 256 512 1024 2048 127 988 1778 1815 2486 2715 2929 3285 3418 3644 4079
12	24	8	0.562346 (0.562346)	1 2 4 8 16 32 64 128 256 512 1024 2048 127 988 1465 1778 1815 2486 2715 2929 3285 3418 3644 4079
12	25	6	0.574914 (0.571447)	1 2 4 8 16 32 64 128 256 512 1024 2048 253 319 506 671 682 847 935 979 1001 1012 2168 2905 3900
12	26	6	0.595741 (0.590549)	1 2 4 8 16 32 64 128 256 512 1024 2048 253 319 341 506 671 682 935 979 1001 1012 1662 2168 2905 3484
12	27	6	0.615682 (0.608983)	1 2 4 8 16 32 64 128 256 512 1024 2048 253 319 341 506 671 682 847 979 1001 1012 1959 2168 2905 3486 3910
12	28	6	0.634616 (0.626344)	1 2 4 8 16 32 64 128 256 512 1024 2048 253 319 341 506 671 847 935 979 1001 1012 1421 2168 2905 3486 3892 4064
12	29	6	0.652638 (0.643046)	1 2 4 8 16 32 64 128 256 512 1024 2048 253 319 506 671 682 847 935 979 1001 1012 1624 1646 2168 2905 3597 3769 3900
12	30	6	0.669963 (0.659305)	1 2 4 8 16 32 64 128 256 512 1024 2048 253 319 506 671 682 847 935 979 1001 1012 1624 1646 2168 2507 2905 3597 3769 3900
12	31	6	0.686494 (0.674852)	1 2 4 8 16 32 64 128 256 512 1024 2048 253 319 341 506 671 847 935 979 1001 1012 1662 1706 2168 2315 2905 3183 3486 3766 3858
12	32	6	0.701519 (0.689781)	1 2 4 8 16 32 64 128 256 512 1024 2048 253 319 506 638 671 682 847 935 979 1001 1012 1468 1797 2105 2543 2708 2849 2874 3596 3929
				(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
12	33	6	0.717395 (0.704112)	1 2 4 8 16 32 64 128 256 512 1024 2048 253 319 506 638 671 682 847 935 979 1001 1012 1365 2105 2543 2708 2849 2874 3164 3255 3813 3929
12	34	6	0.732451 (0.717837)	1 2 4 8 16 32 64 128 256 512 1024 2048 253 319 506 638 671 682 847 935 979 1001 1012 1322 1468 1797 2030 2105 2543 2708 2849 2874 3596 3929
12	35	6	0.746769 (0.743336)	1 2 4 8 16 32 64 128 256 512 1024 2048 253 319 506 638 671 682 847 935 979 1001 1012 1365 2105 2543 2708 2849 2874 3164 3255 3699 3813 3929 3970
12	36	6	0.759141 (0.758019)	1 2 4 8 16 32 64 128 256 512 1024 2048 247 381 443 481 494 595 762 886 962 988 1069 1123 1190 1245 1524 1715 1739 1924 1976 2385 2760 2893 2922 2976
12	37	6	0.772606 (0.771753)	1 2 4 8 16 32 64 128 256 512 1024 2048 247 381 443 481 494 595 762 886 988 1069 1123 1190 1245 1524 1715 1739 1772 1924 1976 2159 2385 2922 2976 3053 3268
12	38	6	0.785354 (0.784715)	1 2 4 8 16 32 64 128 256 512 1024 2048 247 381 443 481 494 595 762 886 962 988 1069 1123 1190 1245 1524 1715 1772 1924 1976 2385 2799 2893 2922 2976 3338 3719
12	39	6	0.797463 (0.796947)	1 2 4 8 16 32 64 128 256 512 1024 2048 247 381 443 481 494 595 762 886 962 988 1069 1123 1190 1524 1715 1739 1772 1924 1976 2385 2437 2922 2976 3265 3338 3695 3957
12	40	6	0.808919 (0.808546)	1 2 4 8 16 32 64 128 256 512 1024 2048 247 381 443 481 494 595 762 886 988 1069 1123 1190 1245 1524 1715 1739 1772 1924 1976 2159 2385 2648 2754 2922 2976 3053 3190 3268
12	41	6	0.819852 (0.819566)	1 2 4 8 16 32 64 128 256 512 1024 2048 247 381 443 481 494 595 762 886 962 988 1069 1123 1190 1524 1715 1739 1772 1924 1976 2385 2759 2922 2976 3193 3265 3498 3695 3934 3957
12	42	6	0.830230 (0.830044)	1 2 4 8 16 32 64 128 256 512 1024 2048 247 381 443 481 494 595 762 886 988 1069 1123 1190 1245 1524 1715 1739 1772 1924 1976 2159 2385 2504 2648 2754 2922 2976 3053 3190 3268 3602
12	43	6	0.840081 (0.839988)	1 2 4 8 16 32 64 128 256 512 1024 2048 247 381 443 481 494 595 762 886 962 988 1069 1123 1190 1245 1524 1739 1772 1924 1976 2339 2385 2861 2922 2958 2976 3131 3481 3599 3602 3754 3902
12	44	6	0.849494 (0.849432)	1 2 4 8 16 32 64 128 256 512 1024 2048 247 381 443 481 494 595 762 886 962 988 1069 1123 1190 1245 1524 1739 1772 1924 1976 2339 2385 2861 2922 2958 2976 3131 3481 3599 3602 3686 3754 3902
12	45	6	0.858427 (0.858420)	1 2 4 8 16 32 64 128 256 512 1024 2048 247 381 443 481 494 595 762 886 962 988 1069 1123 1190 1524 1715 1739 1772 1897 1924 1976 2385 2504 2861 2922 2958 2976 3131 3599 3602 3686 3754 3902 3928
				(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
12	46	6	0.866940 (0.866940)	1 2 4 8 16 32 64 128 256 512 1024 2048 247 381 443 481 494 595 762 886 962 988 1069 1123 1190 1245 1524 1715 1739 1772 1924 1976 2339 2385 2504 2861 2922 2958 2976 3131 3423 3599 3602 3754 3902 3928
12	47	6	0.875017 (0.875016)	1 2 4 8 16 32 64 128 256 512 1024 2048 247 381 443 481 494 595 762 886 962 988 1069 1123 1190 1524 1715 1739 1772 1897 1924 1976 2339 2385 2504 2861 2922 2958 2976 3131 3423 3599 3602 3686 3754 3902 3928
12	48	6	0.882680 (0.882680)	1 2 4 8 16 32 64 128 256 512 1024 2048 247 381 443 481 494 595 762 886 962 988 1069 1123 1190 1245 1524 1715 1739 1772 1897 1924 1976 2339 2385 2504 2861 2922 2958 2976 3131 3423 3599 3602 3686 3754 3902 3928
12	49	5	0.886172 (0.885947)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 618 657 690 717 783 981 1037 1207 1236 1314 1375 1380 1395 1409 1434 1566 1897 1962 2074 2327 2414 2472 2549 2601 2628 2760 2790 2809 2818 2868 3132 3770 3794 3877 3924
12	50	5	0.893084 (0.892827)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 618 657 690 717 783 981 1037 1207 1236 1314 1375 1380 1395 1409 1434 1566 1897 1962 2074 2327 2414 2472 2549 2601 2628 2760 2790 2809 2818 2868 3132 3721 3770 3794 3877 3924
12	51	5	0.899623 (0.899395)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 618 657 690 717 783 981 1037 1207 1236 1314 1375 1380 1395 1409 1434 1566 1897 1962 2074 2327 2414 2472 2549 2601 2628 2760 2790 2809 2818 2868 3132 3721 3770 3794 3852 3877 3924
12	52	5	0.905840 (0.905576)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 618 657 690 749 783 963 981 1037 1207 1236 1314 1380 1395 1434 1495 1566 1653 1885 1897 1962 2074 2327 2414 2472 2549 2601 2628 2748 2760 2790 2809 2818 2868 3132 3770 3794 3877 3924
12	53	5	0.911634 (0.911416)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 618 657 690 749 783 963 981 1037 1207 1236 1314 1380 1395 1434 1495 1566 1653 1885 1897 1926 1962 2074 2327 2414 2472 2549 2601 2628 2748 2760 2790 2809 2818 2868 3132 3770 3794 3877 3924
12	54	5	0.917092 (0.916930)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 618 657 690 717 783 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1566 1885 1897 1962 2074 2196 2327 2414 2472 2549 2601 2628 2639 2790 2809 2818 2868 3132 3721 3770 3794 3852 3877 3924
12	55	5	0.922271 (0.922141)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 618 657 687 690 717 783 981 1037 1098 1207 1236 1314 1380 1395 1409 1434 1566 1885 1897 1926 1962 2074 2327 2414 2472 2549 2601 2628 2790 2809 2818 2868 2990 3132 3306 3721 3770 3794 3852 3877 3924
				(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
12	56	5	0.927142 (0.927042)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 549 618 657 687 690 717 783 981 1037 1098 1207 1236 1314 1395 1409 1434 1566 1885 1897 1926 1962 2074 2327 2414 2472 2549 2601 2628 2760 2790 2809 2818 2868 2990 3132 3306 3721 3770 3794 3852 3877 3924
12	57	5	0.931739 (0.931666)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 618 657 687 690 717 783 981 1037 1098 1207 1236 1314 1380 1395 1409 1434 1566 1885 1897 1926 1962 2074 2327 2373 2414 2472 2549 2601 2628 2639 2790 2809 2818 2868 2990 3132 3306 3721 3770 3794 3852 3877 3924
12	58	5	0.936072 (0.936021)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 549 618 657 687 690 717 783 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1885 1897 1926 1962 2074 2196 2327 2414 2472 2549 2601 2628 2639 2760 2790 2809 2818 2868 3132 3306 3721 3794 3852 3877 3924
12	59	5	0.940148 (0.940113)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 618 657 687 690 717 783 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1566 1885 1897 1926 1962 2074 2327 2373 2414 2472 2549 2601 2628 2639 2748 2790 2809 2818 2868 2990 3132 3306 3721 3770 3794 3852 3877 3924
12	60	5	0.943986 (0.943958)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 618 657 687 690 717 783 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1566 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2790 2809 2818 2868 2990 3132 3306 3721 3770 3794 3852 3877 3924
12	61	5	0.947601 (0.947387)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 549 618 657 687 690 717 783 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2760 2790 2809 2818 2868 2990 3132 3306 3721 3794 3852 3877 3924
12	62	5	0.951003 (0.950994)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 549 618 657 687 690 717 783 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2760 2790 2809 2818 2868 2990 3132 3306 3721 3794 3852 3877 3924
12	63	5	0.954207 (0.954203)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 549 618 657 687 690 717 783 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2760 2790 2809 2818 2809 2818 2868 2990 3132 3306 3721 3794 3852 3877 3924
12	64	5	0.957222 (0.957222)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 549 618 657 687 690 717 783 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1885 1897 1926 1962 2074 2196 2327 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3306 3721 3770 3794 3852 3877 3924
				(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
12	65	5	0.960062 (0.960062)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 549 618 657 687 690 717 783 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3306 3721 3770 3794 3852 3877 3924
12	66	4	0.962563 (0.710374)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 549 618 657 687 690 717 783 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1885 1897 1926 1962 2054 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3306 3721 3770 3794 3852 3877 3924
12	67	4	0.964938 (0.720589)	1 2 4 8 16 32 64 128 256 512 1024 2048 292 309 345 549 618 657 687 690 717 783 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3306 3721 3770 3794 3804 3852 3877 3924
12	68	4	0.967183 (0.721090)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 549 618 657 687 690 717 783 933 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3306 3442 3721 3770 3794 3852 3877 3924 4086
12	69	4	0.969306 (0.728861)	1 2 4 8 16 32 64 128 256 512 1024 2048 309 345 549 618 657 687 690 717 783 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1885 1897 1926 1962 2001 2074 2196 2327 2373 2414 2472 2549 2585 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3306 3721 3746 3770 3794 3799 3852 3877 3924
12	70	4	0.971306 (0.729329)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 687 690 717 783 963 981 1037 1098 1168 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3305 3306 3442 3721 3727 3770 3794 3852 3877 3924
12	71	4	0.973186 (0.729730)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 687 690 717 783 963 981 1037 1098 1168 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3305 3306 3442 3607 3721 3727 3770 3794 3852 3877 3924
				(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
12	72	4	0.974954 (0.730124)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 671 687 690 717 783 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3306 3442 3540 3674 3721 3727 3770 3794 3852 3877 3924
12	73	4	0.976619 (0.736062)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 671 687 690 717 783 882 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2748 2760 2790 2809 2818 2868 2990 3132 3306 3442 3540 3674 3721 3727 3770 3794 3852 3877 3924
12	74	4	0.978181 (0.740797)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 671 687 690 717 783 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2659 2748 2760 2790 2809 2818 2868 2990 3132 3306 3442 3540 3674 3721 3727 3770 3794 3852 3877 3924
12	75	4	0.979648 (0.744526)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 671 687 690 717 783 818 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2659 2748 2760 2790 2809 2818 2868 2990 3132 3306 3442 3540 3674 3721 3727 3770 3794 3852 3877 3924
12	76	4	0.981018 (0.747576)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 671 687 690 717 783 818 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1873 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2659 2748 2760 2790 2809 2818 2868 2990 3132 3306 3442 3540 3674 3721 3727 3770 3794 3852 3877 3924
12	77	4	0.982305 (0.749986)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 671 687 690 717 783 818 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1873 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2659 2722 2748 2760 2790 2809 2818 2868 2990 3132 3306 3442 3540 3674 3721 3727 3770 3794 3852 3877 3924
12	78	4	0.983510 (0.751983)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 671 687 690 717 783 875 882 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1873 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2659 2722 2748 2760 2790 2809 2818 2868 2990 3132 3306 3442 3540 3674 3721 3727 3770 3794 3852 3877 3924
				(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
12	79	4	0.984637 (0.753583)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 549 618 657 671 687 690 717 783 875 882 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1873 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2659 2722 2748 2760 2790 2809 2818 2868 2990 3132 3306 3442 3540 3674 3721 3727 3770 3794 3852 3877 3882 3924
12	80	4	0.985695 (0.754920)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 399 549 618 657 671 687 690 717 783 875 882 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1873 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2601 2628 2639 2659 2722 2748 2760 2790 2809 2818 2868 2990 3132 3306 3442 3540 3674 3721 3727 3770 3794 3852 3877 3882 3924
12	81	4	0.986682 (0.756032)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 399 549 618 657 671 687 690 717 783 875 882 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1654 1718 1873 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2582 2601 2628 2639 2659 2748 2760 2790 2809 2818 2868 2990 3132 3306 3442 3540 3674 3721 3727 3770 3794 3852 3877 3882 3924
12	82	4	0.987605 (0.756923)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 399 549 618 657 671 687 690 717 783 875 882 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1654 1718 1873 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2582 2601 2628 2639 2659 2748 2760 2790 2803 2809 2818 2868 2990 3132 3306 3442 3540 3674 3721 3727 3770 3794 3852 3877 3882 3924
12	83	4	0.988468 (0.757672)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 399 549 618 657 671 687 690 717 783 875 882 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1543 1566 1653 1654 1718 1873 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2549 2582 2601 2628 2639 2659 2722 2748 2760 2790 2809 2818 2868 2990 3132 3306 3442 3540 3674 3721 3727 3770 3794 3852 3877 3882 3924
				(Continues on next page)

Table B.1: *Continued from previous page*

m	n	d	Eqv.rate	Packed integer parity check matrix
12	84	4	0.989272 (0.758296)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 399 549 618 657 671 687 690 717 772 783 875 882 953 963 981 1037 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1873 1885 1897 1926 1962 2074 2196 2327 2373 2414 2472 2487 2549 2582 2601 2628 2639 2659 2722 2748 2760 2790 2809 2818 2868 2990 3132 3306 3417 3442 3540 3674 3721 3727 3770 3794 3852 3877 3882 3924
12	85	4	0.990025 (0.758807)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 399 411 549 618 657 671 687 690 717 783 875 882 953 963 981 1037 1064 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1873 1885 1897 1926 1932 1962 2074 2196 2327 2373 2414 2472 2487 2549 2601 2628 2639 2659 2722 2748 2760 2790 2809 2818 2868 2990 3132 3306 3417 3442 3463 3540 3674 3721 3727 3770 3794 3852 3877 3924
12	86	4	0.990727 (0.759250)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 399 411 549 618 657 671 687 690 717 783 875 882 953 963 981 1037 1064 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1873 1885 1897 1926 1932 1962 2074 2196 2327 2373 2414 2472 2487 2549 2601 2628 2639 2659 2722 2748 2760 2790 2809 2818 2868 2990 3132 3205 3306 3417 3442 3463 3540 3674 3721 3727 3770 3794 3852 3877 3924
12	87	4	0.991389 (0.759621)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 399 411 449 549 618 657 671 687 690 717 783 875 882 953 963 981 1037 1064 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1653 1718 1873 1885 1897 1926 1932 1962 2074 2196 2327 2373 2414 2472 2487 2549 2601 2628 2639 2659 2722 2748 2760 2790 2809 2818 2868 2990 3132 3144 3306 3417 3442 3463 3540 3674 3721 3727 3770 3794 3852 3877 3924
12	88	4	0.992002 (0.759921)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 309 345 399 411 449 549 618 657 671 687 690 717 783 875 882 953 963 981 1037 1064 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1599 1653 1718 1873 1885 1897 1926 1932 1962 2074 2196 2327 2373 2414 2472 2487 2549 2601 2628 2639 2659 2722 2748 2760 2790 2809 2818 2868 2990 3132 3144 3306 3417 3442 3463 3540 3674 3721 3727 3770 3794 3852 3877 3924
12	89	4	0.992574 (0.760178)	1 2 4 8 16 32 64 128 256 512 1024 2048 73 119 309 345 399 411 549 618 657 671 687 690 717 783 875 882 953 963 981 1037 1064 1098 1207 1236 1314 1374 1380 1395 1409 1434 1495 1566 1599 1653 1718 1873 1885 1897 1926 1932 1962 2074 2196 2327 2373 2414 2472 2487 2549 2601 2628 2639 2659 2722 2748 2760 2790 2809 2818 2868 2990 3132 3144 3306 3417 3442 3463 3540 3674 3677 3721 3727 3770 3794 3852 3877 3924

Bibliography

- [1] A. D. Wyner, “The Wire-Tap Channel,” *The Bell System Technical Journal*, vol. 54, pp. 1355–1367, May 1975.
- [2] M. Grassl, “Bounds on the minimum distance of linear codes and quantum codes,” 2007,online, Available:<http://www.codetables.de>.
- [3] NTL. A tour of NTL @ONLINE. [Online]. Available: <http://www.shoup.net/ntl/>
- [4] C. F. J. Cannon, W. Bosma and A. Steel, *HANDBOOK OF MAGMA FUNCTIONS*. Sydney: Sydney University, 2011.
- [5] G. A. Jones and J. M. Jones, *Information and Coding Theory*. UK: Springer-Verlag, 2002.
- [6] R. W. Hamming, *Coding and Information Theory*. USA: Prentice-Hall, 1997.
- [7] C. Arndt, *Information Measures : information and its description in science and engineering*. Berlin New York: Springer, 2001.
- [8] T. M. Korn, *Mathematical Handbook for Scientists and Engineers*. New York: Dover Publications.
- [9] C. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal, The*, vol. 27, no. 3, pp. 379–423, July 1948.
- [10] S. Haykin, *Communication systems*. John Wiley and Sons Inc., 2001.
- [11] R. M. Roth, *Introduction to Coding Theory*. USA: Cambridge University Press, 2006.

-
- [12] C. L. K. P. D.G. Hoffman, D.A. Leonard and J. Wall, *Coding Theory*. USA: Press MARCEL DEKKER, INC., 1991.
 - [13] J. Bierbrauer, *Introduction to Coding Theory*. USA: Chapman and Hall/CRC, 2005.
 - [14] H. F. A. K. A. Betten, M. Braun and A. Kohnert, *Error-Correcting Linear Codes*. Berlin Heidelberg: Springer-Verlag, 2006.
 - [15] S. Roman, *Introduction to Coding and Information Theory*. USA: Springer, 1997.
 - [16] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam: The Netherlands: North-Holland, 1977.
 - [17] T. K. Moon, *Error correction coding*. USA: John Wiley and Sons Inc., 2005.
 - [18] J. Barros and M. R. D. Rodrigues, “Secrecy Capacity of Wireless channels,” in *Information Theory, 2006 IEEE International Symposium on*, July, pp. 356–360.
 - [19] S. Lin and D. Costello, *Error Control Coding(second edition)*. Pearson Education, 2004.
 - [20] H.-C. Chang, H.-P. Lee, T. Lin, and T. K. Truong, “A weight method of decoding the (23, 12, 7) Golay code using reduced table lookup,” in *Communications, Circuits and Systems, 2008. ICCCCAS 2008. International Conference on*, May, pp. 1–5.
 - [21] L. H.-P. Chu, S.-I. and H.-C. Chang, “Fast decoding of the (23, 12, 7) Golay code with four-error-correcting capability,” *Eur. Trans. Telecomm.*, vol. 22, p. 388–395, july 2011.
 - [22] R. J. McEliece, *The Theory of Information and Coding*. UK: Cambridge University Press, 2004.
 - [23] C.-L. Chr, S.-L. Su, and S.-W. Wu, “Decoding the (23, 12, 7) binary Golay code,” in *Communications, 2005 Asia-Pacific Conference on*, Oct., pp. 478–480.

-
- [24] M. Elia, “Algebraic decoding of the (23,12,7) Golay code (corresp.),” *Information Theory, IEEE Transactions on*, vol. 33, no. 1, pp. 150–151, 1987.
 - [25] E. R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw Hill, 1968.
 - [26] W. W. Peterson and E. J. Weldon, *Error-Correcting Codes, second ed.* MA: MIT Press, 1972.
 - [27] C. Chen, “Construction of some binary linear codes of minimum distance five,” *Information Theory, IEEE Transactions on*, vol. 37, no. 5, pp. 1429–1432, Sep 1991.
 - [28] T. Verhoeff, “An updated table of minimum-distance bounds for binary linear codes,” *Information Theory, IEEE Transactions on*, vol. 33, no. 5, pp. 665 – 680, sep 1987.
 - [29] S. Leung-Yan-Cheong, “On a special class of wiretap channels (corresp.),” *Information Theory, IEEE Transactions on*, vol. 23, no. 5, pp. 625–627, Sep 1977.
 - [30] S. Leung-Yan-Cheong and M. Hellman, “The gaussian wire-tap channel,” *Information Theory, IEEE Transactions on*, vol. 24, no. 4, pp. 451–456, Jul 1978.
 - [31] I. Csiszar and J. Korner, “Broadcast channels with confidential messages,” *Information Theory, IEEE Transactions on*, vol. 24, no. 3, pp. 339–348, May 1978.
 - [32] G. Cohen and G. Zemor, “Generalized coset schemes for the wire-tap channel: application to biometrics,” in *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, June 2004, pp. 46–.
 - [33] G. Zemor and G. Cohen, “Syndrome-coding for the wiretap channel revisited,” in *Information Theory Workshop, 2006. ITW '06 Chengdu. IEEE*, Oct 2006, pp. 33–36.
 - [34] S. Y. E. Rouayheb and E. Soljanin, “On wiretap networks ii,” *ISIT*, pp. 551–555, 24-29 June 2007.
 - [35] L. Ozarow and A. Wyner, “Wire-tap channel ii,” *AT T Bell Laboratories Technical Journal*, vol. 63, no. 10, pp. 2135–2157, Dec 1984.

-
- [36] A. Thangaraj, S. Dihidar, A. Calderbank, S. McLaughlin, and J.-M. Merolla, “Applications of ldpc codes to the wiretap channel,” *Information Theory, IEEE Transactions on*, vol. 53, no. 8, pp. 2933–2945, Aug 2007.
 - [37] S. Reddy, P. Aparna, and S. David, “Syndrome coding of video with ldpc codes,” in *Signal Processing, 2008. ICSP 2008. 9th International Conference on*, Oct 2008, pp. 1985–1988.
 - [38] B. Dai, Y. Luo, and A. Vinck, “Wiretap channel with side information from part of encoder,” in *Network and Parallel Computing, 2008. NPC 2008. IFIP International Conference on*, Oct 2008, pp. 353–357.
 - [39] Y. Liang, G. Kramer, H. Poor, and S. Shamai, “Recent results on compound wire-tap channels,” in *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, Sept 2008, pp. 1–5.
 - [40] Y. Chen and A. J. Vinck, “On the binary symmetric wiretap channel,” *Int. Zurich Seminar on Communications(IZS)*, pp. 17–20, 3-5 March 2010.
 - [41] A. Suresh, A. Subramanian, A. Thangaraj, M. Bloch, and S. McLaughlin, “Strong secrecy for erasure wiretap channels,” in *Information Theory Workshop (ITW), 2010 IEEE*, Aug 2010, pp. 1–5.
 - [42] H. Bafghi, B. Seyfe, M. Mirmohseni, and M. Aref, “On the achievable rate region of a new gaussian wiretap channel with side information,” in *Information Theory Workshop (ITW), 2012 IEEE*, Sept 2012, pp. 657–661.
 - [43] K. Zhang, M. Tomlinsin, and M. Ahmed, “A modified mceliece public key encryption system with a higher security level,” in *Information Science and Technology (ICIST), 2013 International Conference on*, March 2013, pp. 991–996.
 - [44] K. Zhang, M. Tomlinson, and M. Ahmed, “The average equivocation of random linear binary codes in syndrome coding,” in *Telecommunications (ICT), 2014 21st International Conference on*, May 2014, pp. 47–51.

-
- [45] Y. Cassuto and Z. Bandic, “Low-complexity wire-tap codes with security and error-correction guarantees,” in *Information Theory Workshop (ITW), 2010 IEEE*, Aug 2010, pp. 1–5.
 - [46] K. Zhang, M. Tomlinson, and M. Ahmed, “A chain based syndrome coding scheme for secure communication in the wiretap channel,” in *Vehicular Technology Conference (VTC Spring), 2014 IEEE 79th*, May 2014, pp. 1–5.
 - [47] I. Bouyukliev and E. Jacobsson, “Results on binary linear codes with minimum distance 8 and 10,” *Information Theory, IEEE Transactions on*, vol. 57, no. 9, pp. 6089–6093, Sept 2011.
 - [48] Y. Edel and J. Bierbrauer, “Inverting construction y1,” *Information Theory, IEEE Transactions on*, vol. 44, no. 5, pp. 1993–, Sep 1998.
 - [49] W. Alltop, “A method for extending binary linear codes (corresp.),” *Information Theory, IEEE Transactions on*, vol. 30, no. 6, pp. 871–872, Nov 1984.
 - [50] N. Sloane, S. Reddy, and C.-L. Chen, “New binary codes,” *Information Theory, IEEE Transactions on*, vol. 18, no. 4, pp. 503–510, Jul 1972.
 - [51] K. Zhang, M. Tomlinson, M. Ahmed, M. Ambroze, and M. Rodrigues, “Best binary equivocation code construction for syndrome coding,” *Communications, IET*, vol. 8, no. 10, pp. 1696–1704, July 2014.
 - [52] K. Zhang, “Best equivocation rate codes,” 2013,online, Available:<http://www.it.pt/auto temp web page preview.asp?id=1219>.
 - [53] S. Bezzateev and N. Shekhunova, “Chain of separable binary Goppa codes and their minimal distance,” *Information Theory, IEEE Transactions on*, vol. 54, no. 12, pp. 5773–5778, Dec 2008.
 - [54] R. J. McEliece, “A public key cryptosystem based on algebraic coding theory,” *DSN Progress Report 42-44*, pp. 114–116, 1978.

-
- [55] D. J. Bernstein, T. Lange, and C. Peters, “Attacking and defending the McEliece cryptosystem,” Cryptology ePrint Archive, Report 2008/318, 2008, <http://eprint.iacr.org/>.
- [56] E. Berlekamp, R. McEliece, and H. Van Tilborg, “On the inherent intractability of certain coding problems (corresp.),” *Information Theory, IEEE Transactions on*, vol. 24, no. 3, pp. 384–386, May 1978.
- [57] R. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.
- [58] H.-M. Sun, “Enhancing the security of the mceliece public-key cryptosystem.” *Journal of Information Science and Engineering*, vol. 16, no. 6, pp. 799–812, 2000.
- [59] H. C. van Tilborg, *Fundamentals of Cryptology*. USA: Kluwer Academic Publishers, 2000.
- [60] P. J. Lee and E. F. Brickell, “An observation on the security of mceliece’s public-key cryptosystem,” in *Advances in Cryptology - EUROCRYPT ’88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings*, ser. Lecture Notes in Computer Science, vol. 330. Springer, 1988, pp. 275–280.
- [61] Y. X. Li, R. Deng, and X. M. Wang, “On the equivalence of mceliece’s and niederreiter’s public-key cryptosystems,” *Information Theory, IEEE Transactions on*, vol. 40, no. 1, pp. 271–273, Jan 1994.
- [62] K. Zhang, M. Ahmed, and M. Tomlinson, “A modified syndrome-coding scheme with a higher security level,” Unpublished.
- [63] S. Al-Hassan, M. Ahmed, and M. Tomlinson, “Secrecy coding for the wiretap channel using best known linear codes,” in *Global Information Infrastructure Symposium, 2013*, Oct 2013, pp. 1–6.

-
- [64] S. Al-Hassan, M. Tomlinson, and M. Ahmed, “New best equivocation codes for syndrome coding,” in *Information and Communication Technology Convergence (ICTC), 2014 International Conference on*, Oct 2014, pp. 669–674.
 - [65] R. Bhattacharjee, K. Ramakrishnan, and K. Dasgupta, “On computation of minimum distance of linear block codes above 1/2 rate coding,” in *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, June 2010, pp. 280–284.
 - [66] M. Askali, S. Nouh, and M. Belkasmi, “An efficient method to find the minimum distance of linear block codes,” in *Multimedia Computing and Systems (ICMCS), 2012 International Conference on*, May 2012, pp. 318–324.
 - [67] S. Al-Hassan, M. Ahmed, and M. Tomlinson, “Construction of best equivocation codes with highest minimum distance for syndrome coding,” in *Communication Workshop (ICCW), 2015 IEEE International Conference on*, June 2015, pp. 485–490.
 - [68] S. Al-Hassan, M. Z. Ahmed, and M. Tomlinson, “Extension of the parity check matrix to construct the best equivocation codes for syndrome coding,” in *Global Information Infrastructure and Networking Symposium (GIIS), 2014*, Sept 2014, pp. 1–3.
 - [69] M. Karlin, “New binary coding results by Circulants,” *Information Theory, IEEE Transactions on*, vol. 15, no. 1, pp. 81 – 92, jan 1969.
 - [70] M. Karlin., “Decoding of Circulant codes (corresp.),” *Information Theory, IEEE Transactions on*, vol. 16, no. 6, pp. 797 – 802, nov 1970.