National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file / Votre référence

Our file / Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

Error Rate Performance Analysis of Stream and Block Ciphers in a Digital Mobile Communication Channel

By

Guy J. M. Ferland, B.Eng.

A thesis submitted to the
School of Graduate Studies and Research
in partial fulfilment of the requirements
for the degree

Master of Applied Science

Ottawa-Carleton Institute for Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
University of Ottawa
October 1993

© 1993, Guy J.M. Ferland, Ottawa, Canada

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington,
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-89646-9

Canada

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

Abstract

This thesis analyses the effect of encrypting data on the bit error rate at the receiver. Block and stream cipher modes based on the Data Encryption Standard (DES) are examined. More specifically, Electronic Code Book (ECB), Cipher Block Chaining (CBC), Output FeedBack (OFB) and Cipher FeedBack (CFB) modes are considered. The ability of binary BCH codes and of bit interleaving to reduce the error rate over fading channels is examined in detail. Experimental results are derived from computer simulations where the digital mobile communication channel is represented by the Gilbert and Fritchman models. Simulations are also carried out over memoryless binary symmetric channels (BSC) to illustrate the difference on the error rate at the receiver (after decryption) as a result of the presence or absence of channel memory.

Acknowledgements

I would like to thank my thesis supervisor, Dr. Jean-Yves Chouinard, for his constant support and for his numerous suggestions and corrections which led to significant improvements to all aspects of this thesis.

I also wish to thank the Telecommunication Research Institute of Ontario (TRIO) and the Natural Sciences and Engineering Research Council of Canada (NSERC) for funding which made it possible to publish and present the results of this research work at various engineering conferences.

Last but not least, I wish to thank my supervisors at the Directorate of Information Systems and Delivery (DIS Del, Department of National Defence) for allowing me to work a very flexible schedule for over a year and a half. This made it possible for me to attend graduate courses at the university in order to fulfil the requirements for the degree of Master of Applied Science.

Table of Contents

List of abbreviations	8
List of symbols	9
Chapter 1: Introduction	12
1.1 Introduction	12
1.2 Thesis organization	13
Chapter 2: The mobile communication channel	16
2.1 Introduction	16
2.2 Mobile channel characteristics	17
2.2.1 Physical attributes of the channel	17
2.2.1.1 Near scatterers	17
2.2.1.2 Far scatterers	19
2.2.1.3 Street orientation	19
2.2.1.4 Doppler effects	21
2.2.2 Statistical models for the received signal	23
2.2.2.1 The Rayleigh distribution	23
2.2.2.2 The Rician distribution	25
2.2.2.3 The Nakagami distribution	26
2.2.2.4 The lognormal distribution	28
2.2.3 Additional parameters related to the mobile channel	28
2.2.3.1 Delay spread	28
2.2.3.2 Coherence bandwidth	29
2.2.3.3 Level crossing rate	29
2.2.3.4 Average fade duration	31
2.3 Digital mobile channel simulation models	33
2.3.1 Characterization of digital mobile channels	33
2.3.2 Renewal versus non-renewal processes	37

2.3.3	Binary symmetric channel	40
2.3.4	Gilbert channel model	44
2.3.5	Fritchman channel model	50
2.3.6	Other channel models	58
2.3.7	Channel capacity	59
2.3.7.1	Channel entropy as a function of memory	62
2.3.7.2	Capacity of a binary symmetric DMC .	63
2.3.7.3	Capacity of a Gilbert model channel	64
2.3.7.4	Capacity of a Fritchman model channel	66
2.4	Conclusion	69
Chapter 3:	Cryptography	70
3.1	Introduction	70
3.2	Theoretical considerations on cryptography	71
3.2.1	A generic encryption system	71
3.2.2	Unicity distance and equivocation	72
3.2.3	An encryption system with perfect secrecy .	73
3.2.4	Confusion, diffusion and mixing functions .	74
3.3	The Data Encryption Standard	77
3.4	Implementation of encryption systems based on the DES	83
3.4.1	Block ciphers	84
3.4.1.1	Electronic Code Book (ECB) mode . .	85
3.4.1.2	Cipher Block Chaining (CBC) mode . .	88
3.4.2	Stream ciphers	90
3.4.2.1	Output Feedback (OFB) mode	92
3.4.2.2	Cipher Feedback (CFB) mode	96
3.5	New trends in cryptography	100
3.6	Conclusion	101
Chapter 4:	Channel coding and interleaving	104
4.1	Introduction	104

4.2	Linear block codes	104
4.3	Binary BCH codes	106
4.3.1	Encoding procedure for binary BCH codes . .	107
4.3.2	Decoding procedure for binary BCH codes . .	108
4.4	Bit interleaving	111
4.5	Probability $P(m,n)$ of m errors in a block of n bits	113
4.5.1	$P(m,n)$ for a binary symmetric channel . .	113
4.5.2	$P(m,n)$ for the Gilbert's channel model . .	114
4.5.3	$P(m,n)$ for the Fritchman's channel model .	115
4.5.4	Comparison between analytical and experimental values of $P(m,n)$	116
4.6	Conclusion	117
Chapter 5: Analysis and simulation results		118
5.1	Introduction	118
5.2	Error propagation characteristics of DES in a binary symmetric channel	118
5.2.1	Error propagation characteristics without channel coding	119
5.2.1.1	Error propagation for Electronic Code Book mode	120
5.2.1.2	Error propagation for Cipher Block Chaining mode	122
5.2.1.3	Error propagation for Output Feedback mode	127
5.2.1.4	Error propagation for Cipher Feedback mode	127
5.2.2	Error propagation characteristics of DES in BSC with BCH channel coding	129
5.2.2.1	Error propagation: ECB mode with BCH coding	133
5.2.2.2	Error propagation: CBC mode with BCH coding	136
5.2.2.3	Error propagation: OFB mode with BCH	

	coding	139
5.2.2.4	Error propagation: CFB mode with BCH	
	coding	140
5.3	Analysis of simulation results over binary	
	symmetric channels	140
5.3.1	OFB mode simulation results	141
5.3.2	ECB mode simulation results	147
5.3.3	CBC mode simulation results	153
5.3.4	CFB mode simulation results	155
5.3.5	Comparison of <i>DBER</i> for all cipher modes . .	155
	5.3.5.1 Comparison of <i>DBER</i> of ECB, CBC, CFB	
	with OFB mode	159
	5.3.5.2 Comparison of <i>DBER</i> of CBC, CFB with	
	ECB mode	162
	5.3.5.3 Comparison of <i>DBER</i> of CFB with CBC	
	mode	165
5.3.6	Dependency of <i>DBER</i> on the data stream and	
	the DES key	166
5.4	Simulated fading channels	167
5.5	Error propagation for OFB mode over fading	
	channels with BCH coding	168
5.6	Simulation results for the Gilbert channel model .	169
5.6.1	Determination of the Gilbert model	
	parameters	169
5.6.2	Performance of binary BCH codes	174
5.6.3	Error propagation characteristics for all	
	modes	179
5.6.4	Avalanche effect of BCH codes	183
5.6.5	Relationship between <i>DBER</i> and BER	183
5.6.6	BCH code performance versus blocklength n .	186
5.6.7	BCH code performance versus code rate . . .	186
5.6.8	Comparison of <i>DBER</i> of ECB, CBC and CFB with	
	OFB mode	187
5.6.9	Comparison of <i>DBER</i> of CBC and CFB with ECB	
	mode	188

5.6.10	Comparison of <i>DBER</i> of CFB with CBC mode	190
5.6.11	Effect of a speed increase	190
5.7	Simulation results for the Fritchman channel model	191
5.7.1	Determination of the Fritchman model parameters	191
5.7.2	Fritchman's model versus Gilbert's model	192
5.7.3	Performance of binary BCH codes	196
5.7.4	Error propagation for the four DES modes	200
5.7.5	<i>DBER</i> versus BER for Fritchman's channels	201
5.7.6	Code performance versus blocklength n	204
5.7.7	Code performance versus code rate	204
5.7.8	Comparison of <i>DBER</i> of ECB, CBC and CFB with OFB mode	206
5.7.9	Comparison of <i>DBER</i> of CBC and CFB with ECB mode	207
5.7.10	Comparison of <i>DBER</i> of CFB with CBC mode	208
5.7.11	Effect of a speed increase	209
5.8	Bit interleaving in Fritchman's fading channels	209
5.8.1	Effect of bit interleaving on channel memory	211
5.8.2	Simulation results for bit interleaving	215
5.9	Conclusion	221
Chapter 6: Conclusions and recommendations		226
6.1	Thesis summary	226
6.2	Recommendations for further research	228
Annex A: Supplementary graphs		232
A.1	Introduction	232
A.2	Graphs for simulations over BSC	232
A.3	Graphs for simulations over Gilbert model channels	239
A.4	Graphs for simulations over Fritchman model channels	242

Annex B: Tables of simulation results	243
B.1 Simulation runs over binary symmetric channels . .	243
B.1.1 Output feedback mode simulation results . .	244
B.1.2 Electronic code Book mode simulation results	246
B.1.3 Cipher Block Chaining mode simulation results	248
B.1.4 Cipher Feedback mode simulation results . .	250
B.1.5 Relative performance of ECB, CBC, CFB modes versus OFB mode	251
B.1.6 Relative performance of CBC, CFB modes versus ECB mode	256
B.1.7 Relative performance of CFB mode versus CBC mode	260
B.1.8 Additional simulations over BSC for the four DES cipher modes	262
B.2 Simulation runs over fading channels (Gilbert model)	264
B.2.1 Output Feedback mode simulation results . .	264
B.2.2 Electronic code Book mode simulation results	266
B.2.3 Cipher Block Chaining simulation results .	267
B.2.4 Cipher Feedback mode simulation results . .	268
B.2.5 Relative performance of ECB, CBC and CFB modes versus OFB mode	269
B.2.6 Relative performance of CBC and CFB modes versus ECB mode	272
B.2.7 Relative performance of CFB mode versus CBC Mode	274
B.3 Simulation runs over fading channels (Fritchman model)	275
B.3.1 Output Feedback mode simulation results . .	275
B.3.2 Electronic code Book mode simulation results	277

B.3.3	Cipher Block Chaining mode simulation results	279
B.3.4	Cipher Feedback mode simulation results	280
B.3.5	Relative performance of ECB, CBC and CFB modes versus OFB mode	282
B.3.6	Relative performance of CBC and CFB modes versus ECB mode	286
B.3.7	Relative performance of CFB mode versus CBC mode	289
B.3.8	Bit interleaving for OFB mode (Fritchman model channel)	291
B.3.9	Bit interleaving for CBC mode (Fritchman model channel)	292
Annex C:	Description of simulated channels	294
C.1	Memoryless binary symmetric channels	294
C.2	Fading channels based on the Gilbert model	295
C.3	Fading channels based on the Fritchman model	299
	List of references	304

List of abbreviations

AFD	Average Fade Duration
ARQ	Automatic Repeat reQuest
BCH	Bose Chaudhuri Hocquenghem
BER	Bit Error Rate
bps	bits per second
BSC	Binary Symmetric Channel
CBC	Cipher Block Chaining
CDF	Cumulative Distribution Function
CDMA	Code Division Multiple Access
CFB	Cipher Feedback
CW	Continuous Wave
DMC	Discrete Memoryless Channel
DPSK	Differential Phase Shift Keying
EBD	Error Burst Distribution
ECB	Electronic Code Book
EGD	Error Gap Distribution
erf	error function
FEC	Forward Error Correction
kHz	kiloHertz
LCM	Least Common Multiple
LCR	Level Crossing Rate
MHz	MegaHertz
NC-FSK	Non Coherent Frequency Shift Keying
OFB	Output Feedback
pdf	probability density function
SNR	Signal to Noise Ratio

List of symbols

A_n	Event that one or more of the bits in the cipher block $C'_n = \{c'_{i,n}, c'_{i+1,n}, \dots, c'_{i+N-1,n}\}$ are received in error.
A_{n-1}	Event that one or more of the bits in the cipher block $C'_{n-1} = \{c'_{i-N}, c'_{i-N+1}, \dots, c'_{i-1}\}$ are received in error.
B	Event that a bit of the sequence $M'_m = \{m'_i, m'_{i+1}, \dots, m'_{i+N-1}\}$ or $\tilde{M}_m = \{\tilde{m}_i, \tilde{m}_{i+1}, \dots, \tilde{m}_{i+N-1}\}$ is decrypted in error.
C	Channel capacity in bits per symbol.
C_{MAX}	Channel capacity in bits per symbol for a memoryless source.
C_n	m -th encrypted block at the transmitter.
C'_n	m -th encrypted block at the receiver, after the channel pass.
$\tilde{C}_n$	m -th encrypted block at the receiver, after the channel pass and decoding.
$CBER$	Bit error rate at the receiver after decoding
d_H	Hamming distance in bits
d_{min}	Minimum code distance: $d_{min} = 2t+1$, where t is the error correcting capability of the (n, k, t) code.
D	Cryptographic degradation: $D = \frac{DBER}{BER}$
$DBER$	Bit error rate at the receiver after decoding (if applicable) and after decryption ($DBER = P_b$ or $DBER = P_{cb}$)
$\Delta DBER_{mode\ 1}$	The % difference in $DBER$ of one cipher mode relative to another one:

$$\Delta DBER_{mode\ 1} = \frac{DBER_{mode\ 1} - DBER_{mode\ 2}}{DBER_{mode\ 2}} \times 100, \text{ where } mode\ 1$$

and *mode 2* can be OFB, ECB, CBC or CFB.

$D_K(C'_m)$	Decryption of the m -th encrypted block at the receiver, using DES key K .
E_m	Event that one or more of the $\bar{N}_{BC}$ codewords (required to encode the encrypted block $C_m = \{c_1, c_{1+1}, \dots, c_{1+N-1}\}$) are decoded in error at the receiver.
E_{m-1}	Event that one or more of the $\bar{N}_{BC}$ codewords (required to encode the encrypted block $C_{m-1} = \{c_{1-N}, c_{1-N+1}, \dots, c_{1-1}\}$) are decoded in error at the receiver.
$E_K(M_m)$	Encryption of the m -th data block at the transmitter, using the DES key K .
K_m	m -th key of s bits for CFB mode, at the transmitter.
K'_m	m -th key of s bits for CFB mode, at the receiver (no FEC coding used).
$\tilde{K}_m$	m -th key of s bits for CFB mode, at the receiver (FEC coding used).
M_m	m -th data block at the transmitter, before encryption.
M'_m	m -th data block at the receiver, after the channel pass and decryption.
$\tilde{M}_m$	m -th data block at the receiver, after the channel pass, decoding and decryption.
$\bar{N}_{BC}$	Average number of codewords required to encode an encrypted block of $N = 64$ bits for ECB and CBC block cipher modes.

$\bar{N}_{sc}$	Average number of codewords required to encode an encrypted block of $N = 64$ bits for CFB stream cipher mode.
P_b	Bit error probability for the decrypted sequence $\{m'_1, m'_{1+1}, \dots, m'_{N-1}\}$ when no FEC coding is used.
P_c	Probability of a decoding failure for an (n, k, t) codeword: $p_c = \sum_{i=t+1}^n P(i, n)$
P_{cb}	Bit error probability for the decrypted sequence $\{\hat{m}_1, \hat{m}_{1+1}, \dots, \hat{m}_{N-1}\}$ when FEC coding is used.
P_{MAX}	Bit error rate for binary symmetric channels at which point $P_{cb} = P_b$ for an (n, k, t) binary BCH code.
$P(m, n)$	Probability that exactly m bits are in error in a block of n bits (assuming bit error independence).
$P_c(V_N, \bar{N}_{sc})$	Probability that exactly v_N codewords out of $[\bar{N}_{sc}]$ are decoded incorrectly.
$\bar{V}_n$	Average number of bits in error in a block of n bits.
$\bar{V}_N$	Average number of codewords in error in a block of $[\bar{N}_{sc}]$ codewords.

Chapter 1

Introduction

1.1 Introduction

If personal computers were the technological revolution of the 70's and 80's, mobile communication systems promise to be the revolution of the 90's and beyond. Rapid technological advances and significant reduction in prices have led to a dramatic growth in the sale of systems such as cellular telephones. Within a few short years, wireless, pocket telephone may replace conventional ones as more consumers demand (and receive) the convenience of continuous, uninterrupted access to communication networks. Additional developments such as road navigation systems for commuters will ensure that mobile communications will remain an area of strong, continuous growth for decades to come.

This surge in the popularity of such systems has also exposed the growing problem of unauthorized eavesdropping of conversations or data transfers over public airwaves. Recent political scandals involving intercepted conversations by senior public officials (e.g. the Charlottetown constitutional talks) highlight clearly how easy it is to intercept private conversations and how potentially damaging this can be to the parties involved.

Despite recent legislation introduced by the federal government which increases the penalties for eavesdroppers, the fact remains that it is nearly impossible to prevent such activities. This is true not only because eavesdropping is a passive activity that cannot be detected by outside sensors (i.e. no electromagnetic emissions are made by the receiver) but also because democracies such as our own recognize to a certain extent

the right of individuals to intercept transmissions over public airwaves (e.g. police broadcast).

Hence, the responsibility of protecting private conversations and sensitive data transfers falls on the shoulders of the people who use public airwaves for private transactions.

However, the use of cryptographic systems to deny access to unauthorized eavesdroppers almost invariably introduces technical problems and often affects the quality and reliability of communication links. The principal goal of this thesis is to determine the impact of cryptographic systems (based on the Data Encryption Standard) on the error rate performance of narrow-band digital mobile communication channels (fading channels). We are particularly interested in the performance of Forward Error Correction (FEC) block codes used in conjunction with such systems. To limit the scope of the thesis, only binary BCH codes are considered. These codes were selected for the simulations because they are available in a wide variety of lengths and code rates. They are also easy to encode or decode, and their performance is relatively easy to analyze. The ability of bit interleaving to enhance the performance of such codes over fading channels is also assessed.

The experimental results are based entirely on computer simulations, where the fading channels are represented by various models whose accuracy has been established by many researchers.

1.2 Thesis organization

The present chapter describes briefly the subject of the thesis and the scope of the work therein.

In chapter 2, a thorough analysis of mobile communication channels is carried out. The main physical factors which affect

this link are described in detail and the most important parameters used to characterize it are described. This is followed by a description of the digital channel models used in the simulations (i.e. binary symmetric channel model, the Gilbert model and the Fritchman model). The advantages and disadvantages of each model are analyzed in detail. The techniques used to determine the model parameters based on the characteristics of a real mobile communication channel are also discussed. This is followed by an analysis of the channel capacity of fading channels. To conclude the chapter, equations are developed to determine the theoretical capacity of fading channels from the parameters of the models used to represent them.

Chapter 3 is devoted entirely to cryptographic considerations. Basic attributes of any good encryption system are examined in detail. Furthermore, parameters which allow us to evaluate the cryptographic strength of a system are examined. Following this, the Data Encryption Standard (DES) is described. The description will demonstrate that this system possesses the required qualities of a robust encryption system, making the DES suitable for low to medium security digital applications. Subsequently, the block and stream cipher modes of operation of the DES are analyzed. These are:

- a) Electronic Code Book (ECB),
- b) Cipher Block Chaining (CBC),
- c) Output FeedBack mode (OFB), and
- d) Cipher FeedBack mode (CFB).

The advantages and disadvantages of each mode are covered in detail. The chapter ends with a brief section describing new trends in cryptography.

Chapter 4 examines linear block codes and the error control technique of bit interleaving. After a short description of linear

block codes in general, the class of binary BCH codes is examined in some detail. The encoding and decoding procedure for such codes is also described. Subsequently, the technique of bit interleaving is examined. The last part of the chapter is devoted to the development of the equations required to determine $P(m,n)$ (i.e. probability of having exactly m bit errors in a block of n bits). This parameter is essential when the performance of block codes has to be determined.

Chapter 5 is divided into two main parts. The first part provides a theoretical analysis of the error propagation characteristics of the four cipher modes used in this work. This leads to equations predicting the bit error rate of these modes after decoding and decryption. The second part focuses on the simulations results: the error rate performance of the four cipher modes is analyzed and compared for all three channel models used. The last section of this chapter examines the simulation results when bit interleaving is carried out over channels generated by the Fritchman model.

Chapter 6 highlights the major conclusions reached in this work and proposes further areas of study in this domain of research.

Annex A contains graphs which are based on the simulation results. In annex B, the simulation results are displayed in table form. Annex C provides the details of the simulated channels used in this work, such as model parameters, theoretical capacity, autocorrelation function, etc. The source code for the computer programs used to carry out the simulations and various data analysis tasks is listed in a separate document (annex D). The last section of the thesis provides a list of references.

Chapter 2

Mobile communication channel characteristics and simulation models

2.1 Introduction

A mobile communication network can take many forms. It may consist of a base station in communication with several mobile units travelling in different areas. Or, it may involve two mobile units communicating with each other directly or through a base station. A cellular telephone system is a good example of such a network. In such a channel, a mobile unit will sometimes be stopped and at other times will be travelling at different speeds, in different directions and in different areas (urban, suburban, tunnels, etc). These factors result in a wide variation in the propagation characteristics of the mobile link and make it one of the most difficult environments in which communications must take place. This chapter examines the characteristics of this difficult communication channel and explores various models which may be used to simulate it.

The first part of this chapter examines the physical aspects of the channel and describes various parameters and functions which characterize it.

The second part of this chapter describes how experimental data derived from tests in the real channel can be used to generate simulation models which can accurately reproduce the behaviour of the real channel. Subsequently, the three simulation models used in this research project are examined in detail (binary symmetric channel, Gilbert and Fritchman channel models). The advantages, disadvantages, accuracy and limitations of each model are then

examined. For each model, the parameters which characterize its behaviour are discussed and the channel capacity is calculated.

2.2 Mobile channel characteristics

In this section, we begin with a description of the attributes of the physical environment in which the mobile unit travels and analyze their impact on the propagation characteristics of the channel. This is followed by a description of the statistical models used to describe the signal at the receiver. Other relevant parameters used to characterize the digital mobile channel are then discussed briefly.

2.2.1 Physical attributes of the channel

As is the case for all channels, the mobile communication link must contend with the problems of attenuation and additive noise. However, it is afflicted with additional impediments which are unique to it. Multipath propagation, fading, refraction, diffraction and doppler frequency effects are the most important additional impairments which influence the mobile channel. These problems are related to factors that can be broken down into four main categories:

- a) near scatterers,
- b) far scatterers,
- c) road orientation and
- d) mobile unit speed.

2.2.1.1 Near scatterers

Near scatterers include all objects in the proximity of the mobile unit such as people, trees, other vehicles, houses and buildings. These obstacles attenuate the signal strength at the receiver by absorbing incident energy. But the degree of

attenuation may vary significantly with factors such as material composition, angle of incidence, weather (wet or dry surfaces) and even the season. For example, trees in summer cause 10 dB more attenuation than in winter because of foliage [56]. These obstacles also cause refraction and diffraction which diffuse the incident signal energy.

But more importantly, other vehicles and buildings which surround the mobile unit will cause reflections of the incident signal. As the signal bounces back and forth between the obstacles surrounding the receiver, multiple copies of the signal combine to create a complex standing wave pattern through which the mobile unit must travel. As it moves through this pattern, the receiver experiences rapid and large fluctuations in signal strength which are related to the peaks and troughs of the standing waves. Figure 2.1 illustrates this phenomenon in the simple case of one scatterer. As a result of the varying signal strength, the bit

Figure 2.1: Standing wave pattern [41].

error rate (BER) at the mobile unit becomes a function of position; it is low when the signal strength is high (corresponding to a peak in the standing wave pattern) and becomes high when the signal strength is low (corresponding to a fade notch). Thus, the multipath reflections create a fading channel with memory where long periods of almost error free reception are interrupted by short periods characterized by a high bit error rate.

The frequency of occurrence and duration of these error bursts is partly dependent on the speed of the mobile unit as it

travels through the signal pattern. As the unit speed increases, the fade notches are encountered more often (higher fade rate) but the fade duration is shorter for each fade. A fade notch will occur every half wavelength on average. Fading frequency increases as the carrier frequency increases and as the mobile unit speed increases:

$$f_d = \frac{2V_m}{\lambda} \quad (2-1)$$

where v_m is the mobile unit speed in meters per second and λ is the carrier wavelength in meters¹.

Furthermore, if the buildings around the mobile unit are high enough and properly oriented, they may obstruct completely the line of sight path between the mobile unit and the base antenna. This results in the receiver having to depend entirely on reflected waves to acquire the signal. Figure 2.2 [39] illustrates the fast fading behaviour for a real channel.

2.2.1.2 Far scatterers

Far scatterers include distant, large buildings, hills and other topological attributes. The impact of these obstacles on the propagation characteristics of the channel are usually more subdued than for near scatterers. The main influence of far scatterers is to modify the average SNR at the receiver, thereby creating a slow fading pattern which is superimposed on the fast fading pattern caused by near scatterers. These slow variations are easily discernible in Figure 2.2.

2.2.1.3 Street orientation

When the mobile unit is travelling on a road with a large

¹ If a square law detector is used at the receiver, the measured fading frequency will be $2f_d$. If a linear detector is used, the measured fading frequency will be f_d [41, pp. 30-31].

Signal amplitude (dB (1 mW)) and speed of mobile unit ((km/h) as function of time.

Amplitude (dB (1 mW)) of signal and its slow variation component as function of position (m).

Figure 2.2: Slow and fast fading of the signal envelope [39]

number of buildings on both sides, a significant level of attenuation occurs when the mobile unit is moving perpendicular to the direction of propagation from the base antenna. The attenuation has been found to be up to 20 dB higher in that case [55], relative to a vehicle at the same distance but travelling on a road parallel to the line of propagation from the base antenna. This is caused by a ducting effect, where the relatively obstacle-free road bounded by the buildings on both sides of it act as a waveguide to channel the signal received from the base antenna.

2.2.1.4 Doppler effects

The motion of the mobile unit through the fast fading pattern at various speeds results in a doppler effect which introduces additional noise at the receiver. When the mobile unit is in a fade notch, the phase of the received signal varies in a rapid, random fashion [41, pp. 219-224], [68, pp. 136]. These phase jumps introduce additional noise in the receiver which is referred to as random Frequency Modulation. Random FM is independent of the signal to noise ratio (SNR)². However, it is dependent on the speed of the fading frequency f_d (a function of the vehicle speed). Random FM noise increases as f_d increases. This is illustrated in Figure 2.3 where it is clear that a lower limit on the BER is imposed by the fading frequency. As a result, it can be seen that random FM introduces an irreducible BER.

² Thus, increasing the transmitted power will not eliminate it.

Figure 2.3: Random FM and the irreducible BER [68]

2.2.2 Statistical models for the received signal

In order to derive statistical models for the signal power at the receiver, it is necessary to first conduct experimental test runs in the real channel. These usually take the form of a base station transmitting a continuous wave (CW) signal to a mobile unit travelling at a fixed speed in a predetermined area. The signal received at the mobile unit is measured at regular intervals to determine its amplitude and phase. This leads to power level versus distance (or speed) graphs such as those of Figure 2.2 and similar ones for the phase of the signal. Typically, the vehicle moves randomly over an area of a few hundred meters under the assumption that the propagation characteristics are invariant over that area. Under the further assumptions of wide sense stationarity³ and ergodicity⁴, it is possible to extend the statistical properties derived from the experimental run to more general cases. This is usually true when the number of sample points of the experimental run is large enough to meet the restrictions of the weak law of large numbers [52, pp. 191-192].

Based on the experimental results described above, it is then possible to derive statistical models to represent the power and phase characteristics of both the slow and fast fading components of the signal. In this section we examine three models proposed for the fast fading properties of the received signal and one for its slow fading aspects [14,41,65,68].

2.2.2.1 The Rayleigh distribution

For a receiver in a dense urban environment, the incident

³ Invariance of first and second order statistics to the time origin.

⁴ Where it is assumed that the statistics derived from the specific realization of the process apply to the ensemble.

signal is a composite sum of reflected waves only. If we assume that a transmitted signal

$$s_o(t) = A_o \exp[j(\omega_c t + \phi)] \quad (2-2)$$

arrives at the receiver as a sum of reflected waves which are independent, identically distributed Gaussian random variables with zero mean, noise variance σ^2 and uniformly distributed random phase, we get

$$r(t) = \sum_{i=1}^N A_o A_i \exp[j(\omega_c t + \phi - \beta V_m t \cos \alpha_i + \phi_i)] \quad (2-3)$$

where A_i , ϕ_i and, α_i are the amplitude, phase and angle of incidence (in relation to the vehicle's direction of travel) respectively. v_m is the mobile unit speed and $\beta = \frac{2\pi}{\lambda}$. It can then be shown that the probability density function (pdf) is given by [41, pp. 169-174]:

$$\begin{aligned} p_{r\theta} &= \frac{r}{2\pi\sigma^2} e^{-\left(\frac{r^2}{2\sigma^2}\right)} \\ p_\theta &= \begin{cases} \frac{1}{2\pi}, & 0 \leq \theta \leq 2\pi \\ 0, & \text{otherwise} \end{cases} \\ p_r &= \frac{r}{\sigma^2} e^{-\left(\frac{r^2}{2\sigma^2}\right)}, \quad r \geq 0 \end{aligned} \quad (2-4)$$

The mean and standard deviation for the Rayleigh distribution are given by:

$$\begin{aligned} \eta_r &= E[r] = \sigma \sqrt{\frac{\pi}{2}} \\ \sigma_r &= \sigma \sqrt{2 - \frac{\pi}{2}} \end{aligned} \quad (2-5)$$

It is generally agreed that at least 7 multipath signals are required to satisfy the assumption of independent, identically distributed Gaussian random variables [41]. Note that the distribution of the signal phase is assumed to be uniform. Again, this assumption is reasonable when the number of reflected waves

incident on the receiver is high enough since the waves all propagate through paths of different lengths before reaching the antenna.

The Rayleigh distribution is considered to be a worst case scenario since many experimental results confirm that the received signal rarely consists of only reflected waves, even in dense urban areas [4,14,39,58,68]. The difference in signal strength between the peaks and fade notches of the interference pattern can be up to 40 dB [68, pp. 135] when the composite signal is a sum of reflections only. For the mobile receiver, this means that it is likely to experience fairly long error bursts of high BER density.

2.2.2.2 The Rician distribution

In most instances, the signal at the mobile unit receiver is a composite of multiple reflected waves and a direct path component of relatively stable amplitude [39]. In such a case, the fade notches are much shallower and the signal strength dynamic range is in the range of 10 to 20 dB (peak to notch ratio) as opposed to the 40 dB mentioned earlier for a pure Rayleigh fading channel⁵. As a result, a mobile receiver in such an environment is likely to experience error bursts of shorter duration and lower BER density than a Rayleigh channel of equivalent average SNR.

The Rician probability distribution is a function which represents well the short term fading of the signal power level at a mobile receiver when the composite signal consists of a stable direct path component (between the receiver and transmitter antennas) and a sum of multiple reflections:

$$p_r(u) = \frac{u}{\sigma^2} \cdot e^{-\frac{(u^2+A^2)}{2\sigma^2}} \cdot I_0\left(\frac{uA}{\sigma^2}\right), u \geq 0 \quad (2-6)$$

⁵ The dynamic range is inversely proportional to the magnitude of the line-of-sight component of the received signal.

where I_0 is the modified Bessel function of order zero given by:

$$I_0(x) = \int_0^{2\pi} e^{(x \cos \theta)} d\theta \quad (2-7)$$

and A is the value of the line of sight component. As A increases, the standard deviation of the signal envelope decreases and fade notches become shallower. The mean and standard deviation for the Rician are given by:

$$\begin{aligned} \eta_r &= e^{-\gamma} \sqrt{\frac{\pi}{2}} [(1+2\gamma) I_0(\gamma) + 2\gamma I_1(\gamma)] \\ \gamma &= \frac{A^2}{4\sigma^2} \\ \sigma_r &= \sqrt{A^2 + 2\sigma^2 - 1} \end{aligned} \quad (2-8)$$

The phase distribution for the Rician is uniform only when $A = 0$. As A increases, the probability density function of the phase looks more and more like a Gaussian distribution but is in fact given by [40, pp. 177]:

$$f_\psi(\psi) = \frac{1}{2\pi} e^{-\frac{A^2}{2\sigma^2}} + \frac{A \cos \psi}{2\sqrt{2\pi}\sigma^2} e^{-\frac{A^2 \sin^2 \psi}{2\sigma^2}} \left[1 + 2 \operatorname{erf} \left(\frac{A \cos \psi}{\sigma} \right) \right] \quad (2-9)$$

for $-\pi \leq \psi \leq \pi$ and where 'erf' is the error function.

2.2.2.3 The Nakagami distribution

The Rician distribution is more accurate than the Rayleigh model to represent most real channels because they usually have a direct line of sight component, even in fairly dense urban environments. Nevertheless, many researchers have found the Rician to be wanting in accuracy [4,68]. This prompted the development of another model to better represent the fast fading characteristics of the received signal. The Nakagami distribution [50] has been found to be the most accurate model for experimentally derived

short term fading statistics [4,14,65]. Its probability density function is given by:

$$p_r(r) = \frac{2}{\Gamma(m)} \left(\frac{m}{\Omega} \right)^m \cdot r^{2m-1} \cdot \exp \left[-\frac{mr^2}{\Omega} \right], \quad m \geq \frac{1}{2}, r \geq 0$$

$$\Omega = E[r^2] = \sigma^2, \quad (\text{if } m_r = E[r] = 0)$$
(2-10)

where $\Gamma(m)$ is the Gamma function given by:

$$\Gamma(m) = \int_0^{\infty} t^{m-1} e^{-t} dt, \quad m > 0$$

$$\Gamma(m) = \frac{\Gamma(m+1)}{m}, \quad m < 0$$
(2-11)

For $m = 0.5$, the Nakagami distribution is a one-sided Gaussian distribution. If $m = 1$, it becomes a Rayleigh distribution. For $m \neq 1$, it can approximate a Rician distribution. Finally, for $m \in [e^{\eta-\frac{1}{4}}, e^{\eta+\frac{1}{4}}]$, the Nakagami distribution approximates a $N(\eta, \sigma)$ lognormal distribution [14] ⁶.

The fundamental difference between the Nakagami and the Rayleigh distribution is that the former assumes that the composite received signal is a sum of reflected waves of random amplitude and phase while for the Rayleigh, it is assumed that the reflected waves all have the same amplitude (except for the noise variance) and random phase. The assumption of equal strength for the reflected waves is often unrealistic and leads to the significant inaccuracies reported by [4,14] and [65] for the Rayleigh distribution pdf. Thus, the Nakagami distribution is more flexible than the Rayleigh in that respect. It is also apparent that both the Rayleigh and Rician distributions are special cases of the Nakagami distribution.

⁶ $m = 0.5$ implies severe fading, $m = 1$ is Rayleigh fading and $m = \infty$ implies no fading [14].

2.2.2.4 The lognormal distribution

The lognormal distribution has been found to be appropriate for representing the long term fading characteristics of the received signal [4,41]. The probability density function of the lognormal distribution is shown below:

$$p_r(r) = \frac{1}{\sqrt{2\pi\sigma^2 r}} \cdot e^{-\frac{(\ln(r)-m_r)^2}{2\sigma^2}}, \quad r > 0 \quad (2-12)$$

In that equation, m_r is the average value of the signal strength. Figure 2.2 illustrates the slow fading component of the received signal strength as a separate curve under (or superimposed) on the composite signal. Incidentally, it should be noted that the slow variations of the signal cause much less problems than the fast fading. As a result, slow fading is seldom analyzed in as much detail or with as much care as fast fading.

2.2.3 Additional parameters related to the mobile channel

There are many other parameters and functions which describe various aspects of the mobile channel (see [41]). Of these, delay spread, coherence bandwidth, level crossing rate (LCR) and the average fade duration (AFD) are important enough to deserve a brief description.

2.2.3.1 Delay spread

Because of multipath scattering, a transmitted impulse will be significantly lengthened by the time it arrives at the receiver. The standard deviation of the average received pulse length is the delay spread [41, pp. 40-46], [54, pp. 703-710]. It imposes a limit on the signalling rate since the transmitter cannot send the next signal until enough time has elapsed to allow the magnitude of the preceding pulse at the receiver to fall below a given level

(otherwise, intersymbol interference would occur). Delay spread is independent of frequency above 30 MHz [41] and is typically 0.5 to 1.3 μ sec although it can go up to 12 μ sec in dense urban areas [41].

2.2.3.2 Coherence bandwidth

The coherence bandwidth [41, pp. 44], [54, pp. 708], [68, pp. 138]. is inversely proportional to the delay spread and is given by:

$$B_c = \frac{1}{4\pi\Delta} \quad (2-13)$$

for frequency modulation, where Δ is the delay spread. Its importance lies in the fact that fading is not frequency selective over that range of frequencies; i.e. for a given propagation path, the attenuation is identical for all frequencies of the transmitted signal. Channels whose bandwidth is smaller than the coherence bandwidth are characterized by non-frequency selective fading and are referred to as narrow-band channels. Throughout this research work, we consider only narrow band digital communication channels whose bandwidths are typically around 25 kHz or lower. For these communication links, traditional modulation techniques like NC-FSK (Non-Coherent Frequency Shift Keying) or DPSK (Differential Phase Shift Keying) are generally used. Wideband channels have a bandwidth higher than the coherence bandwidth and typically utilize spread-spectrum modulation techniques such as CDMA (Code Division Multiple Access). For wideband channels, the individual multipath signals can be resolved at the receiver (this is not true of narrow band channels) [54, pp. 715]. Such channels are not considered in this thesis in order to limit its scope.

2.2.3.3 Level crossing rate

The level crossing rate indicates the expected number of

crossings in one direction for a specified signal power level. This parameter can be important for developing simulation models because it determines the average frequency of occurrence of the fading notches as a function of distance or time. Since fade notches correspond to error bursts at the receiver (because of the low SNR and random FM), this parameter provides us with the average time interval between error clusters. This can then be used to find the average length of error-free runs.

For the Rayleigh distribution, the LCR as a function of time and distance are given by [58]:

$$\begin{aligned}\overline{LCR}(r = R_o) &= \frac{\beta V_m R_o}{2\sigma\sqrt{\pi}} \exp\left(-\frac{R_o^2}{2\sigma^2}\right), \quad \frac{\text{crossings}}{\text{second}} \\ \overline{LCR}(r = R_o) &= \frac{R_o\sqrt{\pi}}{\lambda\sigma} \exp\left(-\frac{R_o^2}{2\sigma^2}\right), \quad \frac{\text{crossings}}{\text{meter}} \\ \beta &= \frac{2\pi}{\lambda}\end{aligned} \quad (2-14)$$

where v_m is the vehicle speed in meters per second and λ is the carrier wavelength in meters. Typical values of LCR based on experimental results (derived by Roy [58]) for the Rayleigh case are shown below ($f = 910$ MHz):

$$\begin{aligned}\overline{LCR}(r = \eta + \sigma) &= 1.66 \frac{\text{crossings}}{\text{meter}} \\ \overline{LCR}(r = \eta) &= 3.07 \frac{\text{crossings}}{\text{meter}} \\ \overline{LCR}(r = \eta - \sigma) &= 2.69 \frac{\text{crossings}}{\text{meter}}\end{aligned} \quad (2-15)$$

where η is the average signal power and σ is the standard deviation.

For a Rician channel, the average LCR is given by [39]:

$$\overline{LCR}(r = R_o) = \frac{\sqrt{\pi}R_o}{\lambda\sigma} \exp\left(-\frac{(R_o^2 + A^2)}{2\sigma^2}\right) I_0\left(\frac{R_o A}{\sigma^2}\right) \quad (2-16)$$

where A is the line of sight component power, R_0 is the power level selected for the crossing threshold and λ is the carrier wavelength. However, it should be noted that some experimental results have revealed that the measured LCR is consistently higher than predicted by the above equation [39, pp. 210]. When the threshold was chosen to correspond to the average signal level, Lecours et al. [39] found that the theoretical LCR was roughly equal to the 3.0 crossings per meter ($\lambda = 0.33\text{m}$) using a linear detector at the receiver. The experimental LCR varied from 2.61 to 3.65 crossings per meter for their measurements.

The level crossing rate is related to the urban density. It increases as the urban density increases. For both Rician and Rayleigh distributions, the maximum LCR occurs when the threshold is in the range $\eta - \sigma$ to σ [58, pp. 48]. The presence of a line-of-sight signal component (e.g. Rician distribution) results in an LCR which is slightly lower than for a Rayleigh channel of similar average power [58, pp. 49-50]. Also, the LCR as a function of threshold for the Rician distributed channel is more symmetric about the signal mean than for the Rayleigh channel.

2.2.3.4 Average fade duration

The average fade duration (AFD) provides us with a measure of the average duration of a fade for a pre-determined power threshold. It is given by the general equation:

$$AFD(r = R_0) = \frac{P(r \leq R_0)}{LCR(r = R_0)} \quad (2-17)$$

where $P(r \leq R_0)$ is the probability that the signal level will remain below the threshold R_0 . The AFD is usually expressed as a function of time or distance.

For a Rayleigh distribution, the AFD is given by:

$$\overline{AFD}(r = R_o) = \frac{P(r \leq R_o)}{\overline{LCR}(r = R_o)} = \frac{1 - \exp\left(-\frac{R_o^2}{2\sigma^2}\right)}{\frac{\beta R_o V_m}{2\sigma\sqrt{\pi}} \exp\left(-\frac{R_o^2}{2\sigma^2}\right)}, \text{ in seconds} \quad (2-18)$$

For $\beta = \frac{2\pi}{\lambda}$, this simplifies to the following forms:

$$\begin{aligned} \overline{AFD}(r = R_o) &= \frac{c\sigma}{V_m f R_o \sqrt{\pi}} \left[\exp\left(\frac{R_o^2}{2\sigma^2}\right) - 1 \right], \text{ in seconds} \\ \overline{AFD}(r = R_o) &= \frac{\lambda\sigma}{R_o \sqrt{\pi}} \left[\exp\left(\frac{R_o^2}{2\sigma^2}\right) - 1 \right], \text{ in meters} \end{aligned} \quad (2-19)$$

For a frequency $f = 910$ MHz in a Rayleigh channel, Roy [58] found these typical values for the AFD:

$$\begin{aligned} \overline{AFD}(r = \eta + \sigma) &= 0.51 \text{ meter} \\ \overline{AFD}(r = \eta) &= 0.18 \text{ meter} \\ \overline{AFD}(r = \eta - \sigma) &= 0.061 \text{ meter} \end{aligned} \quad (2-20)$$

There is no simple mathematical expression for the AFD in a Rician channel. The determination of the AFD involves numerical integration of the following equation:

$$\begin{aligned} \overline{AFD}(r = R_o) &= \frac{P(r \leq R_o)}{\overline{LCR}(r = R_o)} \\ &= \frac{\int_0^{R_o} p_r(u) du}{\frac{V_m \sqrt{\pi} R_o}{\sigma \lambda} \exp\left(-\frac{(R_o^2 + A^2)}{2\sigma^2}\right) I_0\left(\frac{R_o A}{\sigma^2}\right)}, \text{ in seconds} \end{aligned} \quad (2-21)$$

where $p_r(u)$ is the probability density function of the Rician distribution given in equation (2-6) (page 25).

In the experiments conducted by Lecours et al. [39], the average fade duration was found to be about half a wavelength for

Rician channels and a power level equal to the mean (Rician channel, $f = 910$ MHz, urban environment).

2.3 Digital mobile channel simulation models

In this section, we examine how computer models may be constructed to accurately simulate a real digital mobile channel. First, we describe how relevant channel statistics may be derived to characterize the link to simulate. Following this, the concepts of renewal and non-renewal models are examined. The advantages and disadvantages of using one or the other are discussed in relation with the model simplicity and accuracy. This is followed by a detailed description of the three simulation models used in this work; the binary symmetric channel (BSC), the Gilbert and the Fritchman models. For each model, a general narrative is followed by a description of the equations giving the channel parameters. The methods used to derive these parameters from the experimental test results in the real channel are mentioned and graphs illustrating typical statistical distributions of error-free gaps, error bursts and autocorrelation are provided for each. The general algorithm used to generate the channel error sequence is briefly described for each model. Each sub-section concludes with a list of advantages and disadvantages for the model in question.

The last part of this chapter provides a development of the channel capacity for each model and contrasts the capacity of memoryless channels to those with memory.

2.3.1 Characterization of digital mobile channels

In order to be useful, models used to simulate channels must be simple and accurate. However, as is the case in many engineering problems, these two requirements are usually at odds with each other and compromises must be made between simplicity and accuracy. This should be kept in mind when reading the following discussion.

Elements such as mobile unit speed, urban density and modulation technique all contribute to characterize the channel in a complex manner. For example, as the mobile unit speed increases, random FM increases (thus, irreducible BER increases), the LCR increases but the AFD decreases. Similarly, factors such as the presence or absence of a line-of-sight component, or the frequency and modulation technique used (NC-FSK, DPSK) all affect the signal characteristics. Attempting to develop a mathematical model to represent this behaviour based on these parameters would be very complicated.

A much simpler strategy consists in acquiring a large sample of actual error patterns as they occur in the real channel and then use different techniques to create a computer model which can simulate these error patterns.

This involves the transmission of a predetermined sequence of binary digits $\vec{X} = \{X_1, X_2, \dots, X_n\}$ to a mobile unit travelling in the urban area to model⁷. The receiver would record the sequence of digits received $\vec{Y} = \{Y_1, Y_2, \dots, Y_n\}$. By later comparing the received sequence $\vec{Y}$ with the transmitted sequence $\vec{X}$, it is possible to reconstruct the error sequence $\vec{E} = \{E_1, E_2, \dots, E_n\}$ which represents the channel where $Y_i = X_i \oplus E_i$ (Figure 2.4). From this sequence, statistical expressions characterizing the channel may be calculated. In turn, these expressions can be used to determine the parameters for simulation models so that they will generate sequences of digits $\vec{E}_s$ whose statistics are identical to those of the real channel sequence $\vec{E}$ (the index s indicates that the sequence is simulated). Using these sequences $\vec{E}_s$, it is then possible to accurately predict the effect of digital communication

⁷ The indices refer to time. Also, $X_i \in \{0,1\}$ for a binary source.

techniques such as interleaving, forward error correction (FEC) codes and encryption on the word error rate in the real channel, based on the simulation run results.

For a fading channel, the sequence $\bar{E}$ will consist of long intervals of error-free digits followed by shorter intervals with many error digits. The statistical properties of this sequence provide information which is necessary to construct a channel model. These statistical functions are described below.

Figure 2.4: Channel with additive noise

The bit error rate (BER) is defined as:

$$BER = \frac{\text{Number of Bits in Error}}{\text{Total Number of Bits}} \quad (2-22)$$

and is a popular first order statistic in communications. Although it completely specifies a memoryless binary symmetric channel, it is not sufficient to characterize a channel with memory.

The error-free gap distribution (EGD) is given by $P(0^m|1) = P[G_i \geq m + 1]$ where G_i is the length of the i -th gap. This is the probability that m or more consecutive zeros (no errors) will occur given that a one (an error) has just occurred.

The error-burst distribution (EBD) is given by $P(1^m|0) = P[B_i \geq m + 1]$ where B_i is the i -th burst. This is the probability that m or more consecutive digits will be in error given that a zero (no error) has just occurred. The EGD and EBD are given by [37]:

$$\begin{aligned} P(0^m|1) &= \sum_{k=m}^{\infty} P(0^k 1|1) \\ P(1^m|0) &= \sum_{k=m}^{\infty} P(1^k 0|0) \end{aligned} \quad (2-23)$$

where $P(0^k1|1)$ is the probability of having exactly k zeros followed by a '1', given that a '1' has just occurred (similarly for $P(1^k0|0)$). These equations are related as follows:

$$\begin{aligned} P(0^i1|1) &= P(0^i|1) - P(0^{i+1}|1) \\ P(1^i0|0) &= P(1^i|0) - P(1^{i+1}|0) \end{aligned} \quad (2-24)$$

Note that by definition, $P(0^0|1) = P(1^0|0) = 1$.

Two other parameters which describe a channel with memory are the average gap length $E[G]$ and the average burst length $E[B]$. The general equations are given below [37]:

$$\begin{aligned} E[G] &= \sum_{i=0}^{\infty} (i+1) P[G = i] = \sum_{i=0}^{\infty} (i+1) P(0^i1|1) = \sum_{i=0}^{\infty} P(0^i|1) \\ E[B] &= \sum_{i=0}^{\infty} (i+1) P[B = i] = \sum_{i=0}^{\infty} (i+1) P(1^i0|0) = \sum_{i=1}^{\infty} P(1^i|0) \end{aligned} \quad (2-25)$$

Another parameter of interest is the error cluster length. An error cluster consists of a sequence of consecutive digits which starts with a '1' and ends with a '1' but which contains '0's in the sequence. The length of an error cluster is arbitrary and is determined in part by the minimum bit error rate (BER) allowed within the cluster. When the number of consecutive zeros following a one is such that the resultant BER falls below the chosen threshold, the cluster ends at the '1' preceding these zeros. A cluster also ends at the preceding '1' if the number of consecutive zeros exceeds a given threshold, regardless of the BER within the cluster when this occurs. Error clusters are separated by error-free runs, which consist of a string of '0's between two '1's. Note that error-free runs are different from error-free gaps since error clusters and error bursts are defined differently.

Finally, the (normalized) autocorrelation function defined as [51]:

$$\Phi(k) = \frac{\sum_{j=1}^{n-k} E_j E_{j+k}}{\sum_{j=1}^{n-k} E_j}, \quad k = 1, 2, 3, \dots, n \quad (2-26)$$

(where n is the number of symbols in the sequence) is a useful parameter. It provides a measure of the amount of memory present in the channel and gives an indication of the optimal interleaving degree for FEC codes (see chapter 4). The longer and denser error bursts are, the slower will be the rate of decrease of $\Phi(k)$ where k is in bits. For memoryless channels, the autocorrelation function should be an impulse (Dirac delta function) at $\Phi(0)$. The autocorrelation function is also useful to determine if bursts occur at regular (cyclic) intervals since secondary peaks would occur in that case (the peak sharpness and height being a function of the regularity of recurrence of the error bursts).

These statistics are generally sufficient to represent the digital mobile channel and are used in curve fitting techniques to generate a model to simulate the channel. However, second order statistics such as the multigap distribution⁸ [37] may be required to characterize the channel more accurately. But this is seldom done, for reasons which are explained in the following sub-section.

2.3.2 Renewal versus non-renewal processes

Consider a mobile platform in an urban environment, travelling at a constant speed through the fading interference pattern illustrated in Figure 2.2. Because the unit speed is constant, the duration of consecutive error-free gaps should be similar since the vehicle requires the same amount of time to travel from one fade

⁸ The pdf representing the correlation between successive error-free gaps.

notch to another. If the mobile unit increases its speed, the error-free gaps will now be shorter but consecutive gaps will have roughly the same duration at the new speed.

Thus, it is clear that there is a strong correlation between successive gap lengths (and burst lengths). Given that a gap of length s_1 bits has occurred, the probability that the next gap is also of average length s_1 is much higher than the probability of the next gap being of length s_2 where $s_2 < s_1$ or $s_2 > s_1$. Research work on this subject has indicated that the correlation between successive gaps is very high for long gaps but lower for short ones [37, pp. 728]. Because of this correlation between successive gap lengths, the real channel is said to be a non-renewal process.

By contrast, a renewal process is one where there is no correlation between successive gap lengths. In such processes, the probability of getting a gap of length s_1 is independent of the previous gap length. Thus, long and short gaps are randomly distributed over time in the sequence $\vec{E} = \{E_1, E_2, \dots\}$. Clearly, a channel model based on a renewal model does not accurately represent the behaviour of a real channel even though the statistical distributions EGD, EBD, the BER and the autocorrelation of the model sequence $\vec{E}_m$ may be identical to those of the real channel sequence $\vec{E}$.

In order to completely specify a non-renewal process, it is necessary to have not only the EGD and EBD distributions but also second order statistics such as the multigap distribution. However, such statistics are difficult to calculate and require a very large sample sequence $\vec{E}$ to be reliable (because of the weak law of large numbers). Furthermore, Markov chain models which are non-renewal are significantly more complicated to analyze than simple renewal models [37]. In contrast, a renewal channel model is completely specified by the error-free gap distribution [37], which simplifies

greatly the determination of the model parameters in order to fit the real channel statistics. Finally, renewal models are easier to analyze mathematically than non-renewal models [29,37].

As mentioned earlier, simplicity is an essential attribute of channel models in order that estimation of parameters, statistical calculations and simulation speed on a computer all be reasonably simple and quick to perform. For those reasons, most models used in practice are of the renewal type.

The impact of using a renewal model to represent a non-renewal process has been analyzed in some detail [37,51]. It has generally been found that the performance of error control codes predicted by the use of a renewal model is more optimistic than is the case in the real, non-renewal process channel [51]. This is thought to be due to the fact that multigap correlation in the real channel results in consecutive short error-free gaps. These shorter gaps between error bursts result in more errors in block codes than would be the case if short gaps were not followed or preceded by other short gaps. In a renewal process model, the lack of multigap correlation would result in short gaps being followed or preceded by either short or long gaps with equal probability. Thus, on average, multiple, consecutive short gaps are less likely to occur in the simulated sequence $\bar{E}_s$ and consequently block codes would be less likely to be overwhelmed by two or more error bursts in close proximity to each other. However, this is only apparent when the code length is long enough to span two or more gaps separated by one or more error bursts.

As a result of the relatively small inaccuracy introduced by the use of a renewal model, it is usually preferable to use such a model and put up with the inaccuracy rather than attempt to use a non-renewal model whose accuracy is usually outweighed by the additional model complexity. However, when a renewal model is used

to simulate a non-renewal channel, it is necessary to examine the implications of such a choice on the simulated results. It could be necessary to include a correction factor to compensate for the slightly optimistic figures which the simulation runs may yield.

2.3.3 Binary symmetric channel

The binary symmetric channel (BSC) is a special case of the discrete memoryless channel (DMC), where the input and output symbol sets consist of $\{0,1\}$ and where the channel transition probabilities are given by:

Figure 2.5: Binary symmetric channel.

$$\begin{aligned} P(0|1) &= P(1|0) = p \\ P(1|1) &= P(0|0) = 1-p \end{aligned} \quad (2-27)$$

$P(0|1)$ should be interpreted as the probability of receiving a '0' given that a '1' was transmitted. Thus, $P(0|1)$ and $P(1|0)$ represent the probability of a channel error occurring. The channel model is shown in Figure 2.5. In a DMC such as the BSC, errors are distributed uniformly over time and individual bit errors E_i occur independently of each other. Thus, the error burst and error-free gap densities are easily derived from Bayes' theorem, as shown below:

$$\begin{aligned} P(0^m|1) &= \frac{P(10^m)}{P(1)} = \frac{p(1-p)^m}{p} = (1-p)^m \\ P(1^m|0) &= \frac{P(01^m)}{P(0)} = \frac{(1-p)p^m}{(1-p)} = p^m \end{aligned} \quad (2-28)$$

The model parameter p is easily derived from the experimental test run in the real channel; it is simply the measured BER of the sequence $\bar{E}$. The BSC is a renewal model for which the average gap length and burst length are easily found from the infinite series sum:

$$\begin{aligned} E[G] &= \sum_{i=0}^{\infty} (i+1) P(0^i 1 | 1) = \left(p \sum_{i=1}^{\infty} i (1-p)^i \right) + 1 = \frac{1}{p} \\ E[B] &= \sum_{i=0}^{\infty} (i+1) P(1^i 0 | 0) = \left((1-p) \sum_{i=1}^{\infty} i p^i \right) = \frac{1+p}{1-p} \end{aligned} \quad (2-29)$$

where $\sum_{i=1}^{\infty} i a^i = \frac{a}{(1-a)^2}$, $-1 < a < 1$ [64, pp. 107].

The algorithm of the simulation program which generates the pseudo-random sequence $\bar{E}_s = \{E_1, E_2, \dots, E_n\}$ for the channel is given below:

```

1 --> GIVEN p and n (the number of bits to generate)
2 --> For i = 1 to n, DO
  a --> Generate random number Z in the range [0,1]
  b --> IF Z <= p, Ei = 1
      ELSE Ei = 0
3 --> Save the sequence {E1, E2, ..., En} in a file.

```

In order to simulate a channel pass, the file of data to be transmitted is EXCLUSIVE-OR'ed with the channel file and the result is saved in a file which then corresponds to the sequence $\bar{Y} = \{Y_1, Y_2, \dots, Y_n\}$ as described earlier.

Figure 2.6 illustrates the autocorrelation function of the BSC, Gilbert and Fritchman channel models. As expected, it is an impulse at $k = 0$ for the memoryless channel and decays more slowly as channel memory increases. The capacity of the BSC with the BER given earlier is 0.830705 bits per symbol.

Figure 2.7 illustrates the EGD for the BSC (measured BER of file is 0.0256) compared with the models for channels with memory (with a similar BER). Note that the curve of the EGD is represented by a single, weighted exponential for the BSC, as opposed to the other models. Furthermore, the probability of long gaps is lower for the memoryless channel than for the fading channels.

Figure 2.8 shows the EBD for the BSC and the other models. Note that the probability of long bursts is lower for the memoryless channel than for fading channels. As the channel memory increases, the slope of the EBD versus m curve becomes less negative. Furthermore, note that regardless of the channel model used, a single weighted exponential is sufficient to represent the error burst distribution.

Figure 2.6: Autocorrelation function for BSC, Gilbert and Fritchman channel models.

Figure 2.7: EGD For BSC, Gilbert and Fritchman channel models.

Figure 2.8: EBD for BSC, Gilbert and Fritchman channel models.

The main advantage of the BSC is its simplicity. It accurately represents memoryless channels but is totally inadequate for channels with memory. Nevertheless, it is used extensively in this research work because experimental results for a channel with memory are often compared to those of a memoryless channel. One reason for this is that the error control technique of interleaving operates by removing the channel memory to some extent, thereby transforming a fading channel into a memoryless BSC. Thus, a performance comparison between a fading and memoryless channel indicates the degree of success with which the channel memory has been removed. Note that the renewal nature of the model is not a disadvantage as long as the model is used to represent memoryless channels since these are renewal processes.

2.3.4 Gilbert channel model

Figure 2.9: Gilbert channel model

The Gilbert model [31] is a renewal process consisting of a two state Markov chain as shown in Figure 2.9. In the good state (state = 'G'), the channel is represented by a binary symmetric channel with probability of error $p = 0$. When the chain is in the bad state (state = 'B'), the channel is represented by a BSC with a non-zero error probability p . The

transition probability P_{gb} represents the probability of transition from the good to the bad state while P_{bg} is the converse. P_{gg} is given by $1 - P_{gb}$ and is the probability of remaining in the good state. Similarly, P_{bb} is the probability of remaining in the bad state and is given by $1 - P_{bg}$.

For the Gilbert model, the channel bit error rate (BER) is given by:

$$P(e) = p P_b \quad (2-30)$$

where

$$P_b = P_{gb} P_g + P_{bb} P_b \quad (2-31)$$

Given Bayes' theorem:

$$P(S=j) = \sum_i P_i P_{ij} \quad (2-32)$$

and $P_b + P_g = 1$, we obtain the following relationships:

$$P_b = \frac{P_{gb}}{P_{gb} + P_{bg}}, \quad P_g = \frac{P_{bg}}{P_{bg} + P_{gb}} \quad (2-33)$$

Thus, the channel model bit error rate is given by:

$$P(e) = p P_b = p \frac{P_{gb}}{P_{bg} + P_{gb}} \quad (2-34)$$

The error gap distribution function (EGD) is given by:

$$P(0^k|1) = AJ^k + (1-A)L^k \quad (2-35)$$

where J , L and A are given by [26]:

$$\begin{aligned} J &= \frac{P_{gg} + (1-p)P_{bb} + \sqrt{(P_{gg} + (1-p)P_{bb})^2 + 4(1-p)(1 - P_{bb} - P_{gg})}}{2} \\ L &= \frac{P_{gg} + (1-p)P_{bb} - \sqrt{(P_{gg} + (1-p)P_{bb})^2 + 4(1-p)(1 - P_{bb} - P_{gg})}}{2} \\ A &= \frac{2 - P_{gg} - (1+p)P_{bb} + \sqrt{(P_{gg} + (1-p)P_{bb})^2 + 4(1-p)(1 - P_{bb} - P_{gg})}}{2 \sqrt{(P_{gg} + (1-p)P_{bb})^2 + 4(1-p)(1 - P_{bb} - P_{gg})}} \end{aligned} \quad (2-36)$$

Generally, the values of P_{gg} and P_{bb} are close to one. This results in long error-free gaps followed by short error bursts. The selection of the parameters P_{gg} , P_{bb} and p completely specifies the renewal process model and determines the error burst and error-free

gap distributions.

The average gap length calculated from equation (2-25) is given by [37]: $E[G_i] = \frac{1}{BER}$. Similarly, the average burst length is $E[B_i] = \frac{1+BER}{1-BER}$, (the BER is given by equation (2-30)).

Gilbert developed recursive equations for the calculation of the EGD based on the model parameters [25,31]:

$$\begin{aligned} P(0^0|1) &= 1 \\ P(0^1|1) &= 1 - pP_{bb} \\ P(0^k|1) &= (P_{gg} + (1-p)P_{bb})P(0^{k-1}|1) - (1-p)(P_{gg} + P_{bb} - 1)P(0^{k-2}|1) \end{aligned} \quad (2-37)$$

The determination of the parameters P_{gg} , P_{bb} and p from experimental test runs can be accomplished by curve fitting techniques (e.g. least square error estimates) as described by Gilbert [31]. It is first necessary to analyze the test results in order to generate the experimental EGD. Then, a recursive algorithm is used to determine the model parameters which lead to the error-free gap distribution $P(0^m|1)$ with the smallest error with respect to the experimental results.

Once the model parameters are known, the algorithm to generate the pseudo-random error sequence $\vec{E}_n = \{E_1, E_2, \dots, E_n\}$ is as follows:

```

1 --> GIVEN  $P_{gg}$ ,  $P_{bb}$ ,  $p$  and  $n$  (number of bits to generate)
2 --> Determine the initial chain state as follows:
  a --> Calculate  $P_g$  and  $P_b$ 
  b --> Generate random number  $Z$  in  $[0,1]$ 
  c --> IF  $Z \leq P_g$ , initial state = 'G'
      ELSE initial state = 'B'
3 --> For  $i = 1$  to  $n$  do:
  a --> Generate random number  $Z$  in  $[0,1]$ 
  b --> IF  $Z \leq P_g$ , then stay in the same state (?? = gg or bb)
      ELSE move to the other state.
  c --> IF state = 'G' then  $E_i = 0$ 
      ELSE do
        generate random number  $Z$  in  $[0,1]$ 
        IF  $Z \leq p$  then  $E_i = 1$ 
        ELSE  $E_i = 0$ 
4 --> Save sequence  $\{E_1, E_2, \dots, E_n\}$  in a file.

```

A typical EGD distribution is shown in Figure 2.7 and is based on a model with $P_{gg} = 0.98$, $P_{bb} = 0.92$ and $p = 0.12425$ for a theoretical BER of 0.0249 (measured BER of channel file is 0.025286). The presence of memory is clearly visible as the EGD for large values of m is higher than for the memoryless channel. Furthermore, the curve for the EGD is a sum of two exponentials. Figure 2.8 illustrates the effect of memory on the EBD, which results in a higher probability for long bursts than a memoryless channel.

From Figure 2.6, we see that the presence of memory results in an autocorrelation function which decays more slowly than for a memoryless channel. Furthermore, Figure 2.10 shows that the bursts for the Gilbert model have virtually no tendency to occur regularly (cyclic behaviour), as opposed to the Fritchman model (note that artificial offsets of 0.05 and 0.1 were added to the autocorrelation curves of the Gilbert and Fritchman models respectively so that all three curves could be shown on the same graph without overlapping).

Finally, the theoretical capacity of the Gilbert channel model

is 0.846889 bits per symbol (versus 0.830705 for the BSC model seen

Figure 2.10: Autocorrelation and cyclic behaviour for the BSC, Gilbert and Fritchman model channels (artificial offsets of 0.05 and 0.1 were added to $R(X, X+K)$ for the Gilbert and Fritchman channel points respectively so that the three curves would not overlap).

earlier). As predicted by information theory, this is higher than for a memoryless channel with the same BER (but only slightly higher). This indicates that the Gilbert channel model with the given parameters has only a slight amount of memory, which is confirmed by the autocorrelation function as well as by the EGD and EBD graphs.

Since the Gilbert model is a renewal model, it is completely determined by the error-free gap distribution. This simplifies the process of determining the parameters P_{gg} , P_{bb} and p of the model since only the error-free gap distribution of the real channel needs to be calculated. Furthermore, the algorithm is simple to implement on a computer. The Gilbert model has been found to be

accurate for telephone channels by Gilbert [31] and is usually acceptable to represent digital mobile communication channels with low BER (below 10^{-3}) [37,60].

Figure 2.11: Gilbert model with 3 states [37]

One limitation of the model is that the Markov chain is non-unifilar [37]. This implies that the state of the chain at any given time cannot be deduced from an examination of the sequence $\vec{E}_s = \{E_1, E_2, \dots, E_n\}$ (i.e. when $E_i = 0$, it is not possible to determine whether the chain is in the good or bad state since E_i can take that value in either state). This complicates

the calculation of the channel capacity (covered later). However, the Gilbert model may be redrawn as shown in Figure 2.11 where the chain can be made unifilar by merging the state B_0 with the state G (no errors occur in state B_0). The chain state is then determined from the sequence $\vec{E}_s$ without ambiguity.

Another limitation of the Gilbert model is the underlying assumption that error-free gaps have a geometric distribution [37]. It has been shown that this assumption is not normally valid for channels with memory [6]. The limitation is due to the fact that only one state is used to represent the error-free gap distribution. Other models with more states usually yield an error-free gap distribution which is closer to reality. The model inaccuracy is less apparent when a low BER channel is simulated since the geometric distribution assumption is reasonably accurate for such channels. Finally, the renewal nature of the model introduces errors in the simulation results as mentioned earlier.

2.3.5 Fritchman channel model

The Fritchman model is a general case of the Gilbert model but is more flexible. It was proposed in 1967 by Fritchman [29] and is based on an N state Markov chain which is partitioned into two groups of states as shown in Figure 2.12. The states $s = 1$ to $s = k$ are error free states while the states $s = k + 1$ to $s = N$ are error states. Transitions are allowed between all states.

Figure 2.12: Partition of states in Fritchman's model

The general Fritchman model shown in Figure 2.12 is a non-renewal model but its mathematical analysis is very complicated and the determination of the parameters (transition probabilities) is equally difficult. In order to simplify those calculations, Fritchman has proposed a simplified model as shown in Figure 2.13. For this model with N

Figure 2.13: Fritchman's simplified model

states, the first $N - 1$ states are 'good' states while the N -th one is the bad state. A further restriction is added in that no transitions are allowed between states in the same partition (e.g. no transitions are allowed between the good states). The error digit E_i corresponds to the value of z_i . The simplified model can be shown to be a renewal model and is completely specified by the

error-free gap distribution.

The transition matrix for the simplified Fritchman model is given below:

$$\begin{bmatrix} P_{1,1} & 0 & \dots & \dots & P_{1,N} \\ 0 & P_{2,2} & \dots & \dots & P_{2,N} \\ 0 & 0 & P_{3,3} & \dots & P_{3,N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & P_{N-1,N-1} & P_{N-1,N} \\ P_{N,1} & P_{N,2} & P_{N,3} & \dots & P_{N,N} \end{bmatrix} \quad (2-38)$$

where P_{ij} is the probability of going from state i to state j . The error-free gap distribution of the model is given by [29]:

$$P(0^m|1) = \sum_{i=1}^{N-1} \frac{P_{Ni}}{P_{ii}} (P_{ii})^m, m \geq 1 \quad (2-39)$$

If we substitute parameters α and β where

$$\alpha_i = \frac{P_{Ni}}{P_{ii}}, \quad \beta_i = P_{ii} \quad (2-40)$$

the transition probability matrix becomes:

$$\begin{bmatrix} \beta_1 & 0 & \dots & \dots & 1-\beta_1 \\ 0 & \beta_2 & \dots & \dots & 1-\beta_2 \\ 0 & 0 & \beta_3 & \dots & 1-\beta_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \beta_{N-1} & 1-\beta_{N-1} \\ \alpha_1\beta_1 & \alpha_2\beta_2 & \alpha_3\beta_3 & \dots & 1-\sum_{i=1}^{N-1} \alpha_i\beta_i \end{bmatrix} \quad (2-41)$$

and the EGD and EBD become

$$\begin{aligned}
P(0^m|1) &= \sum_{i=1}^{N-1} \alpha_i (\beta_i)^m, m \geq 1 \\
P(1^m|0) &= \alpha_N (\beta_N)^m, m \geq 1 \\
\beta_N &= 1 - \sum_{i=1}^{N-1} \alpha_i \beta_i
\end{aligned} \tag{2-42}$$

It is clear from the last equations that the error-free gap distribution is a sum of k weighted exponential functions while the error burst distribution is represented by a single exponential function, weighted by a factor α_N . Each error-free state represents a gap whose average length is determined by the persistence of the state (P_{ii}) and whose probability of occurrence is determined partly by P_{Ni} . The length of error bursts is similarly determined by the persistence of the N -th state s_N given by the probability P_{NN} while the density of bit errors within the cluster is determined by P_{Ni} and P_{1N} ⁹.

It has been experimentally established that a single error state is usually sufficient to adequately represent the error burst distribution [16,51,60]. Two error free states are generally sufficient to represent a fading channel with a BER below 10^{-2} [16,60] while a three error-free state Fritchman model is sufficient when the BER is above 10^{-2} [60]. The use of a five-state Fritchman model (four error-free states and one error state) in a high BER channel has been found to yield no significant improvement in the accuracy of the model over a four-state model [51] and therefore the additional complexity is not warranted in most cases.

⁹ As a general rule, states are labelled in order of increasing P_{ii} and decreasing P_{Ni} . Thus, if state s_i has a low P_{ii} , and state s_N has a correspondingly high P_{Ni} , it is generally considered that they jointly determine the density of errors within a cluster. The cluster ends when the chain transits from state s_N to any other state except state s_i .

Thus, a four-state model can be considered to be fully adequate for all practical channels. For a four-state Fritchman model, the *a priori* state probabilities can be determined from:

$$\begin{aligned}
 P_i &= \sum_{j=1}^N P_j P_{ji}, \\
 \sum_{i=1}^N P_i &= 1 \\
 1 &\leq i \leq N
 \end{aligned}
 \tag{2-43}$$

where:

$$\begin{aligned}
 P_1 &= P_1 P_{11} + P_4 P_{41} \\
 P_2 &= P_2 P_{22} + P_4 P_{42} \\
 P_3 &= P_3 P_{33} + P_4 P_{43} \\
 P_4 &= P_1 P_{14} + P_2 P_{24} + P_3 P_{34} + P_4 P_{44} \\
 1 &= P_1 + P_2 + P_3 + P_4
 \end{aligned}
 \tag{2-44}$$

By solving these equations, we get

$$\begin{aligned}
 P_4 &= \frac{P_{14} P_{24} P_{34}}{P_{41} P_{24} P_{34} + P_{42} P_{14} P_{34} + P_{43} P_{14} P_{24} + P_{14} P_{24} P_{34}} \\
 P_1 &= \frac{P_4 P_{41}}{P_{14}} \\
 P_2 &= \frac{P_4 P_{42}}{P_{24}} \\
 P_3 &= \frac{P_4 P_{43}}{P_{34}}
 \end{aligned}
 \tag{2-45}$$

In terms of parameters α_i and β_i , the above equations become:

$$P_4 = \frac{(1-\beta_1)(1-\beta_2)(1-\beta_3)}{\alpha_1\beta_1(1-\beta_2)(1-\beta_3) + \alpha_2\beta_2(1-\beta_1)(1-\beta_3) + \alpha_3\beta_3(1-\beta_1)(1-\beta_2) + (1-\beta_1)(1-\beta_2)(1-\beta_3)}$$

$$P_1 = \frac{P_4\alpha_1\beta_1}{(1-\beta_1)}, \quad P_2 = \frac{P_4\alpha_2\beta_2}{(1-\beta_2)}, \quad P_3 = \frac{P_4\alpha_3\beta_3}{(1-\beta_3)} \quad (2-46)$$

$$P(0) = P_1 + P_2 + P_3, \quad P(1) = P_4 = BER$$

The last two expressions give the probability of a 0 or 1 occurring. From these the average gap length and the average burst length can be derived as follows:

$$E[G] = \sum_{i=0}^{\infty} (i+1) P(0^i 1 | 1) = \frac{1}{P(1)}$$

$$E[B] = \sum_{i=0}^{\infty} (i+1) P(1^i 0 | 0) = \frac{1+P(1)}{1-P(1)} = \frac{1+P(1)}{P(0)} \quad (2-47)$$

The fact that a single state is adequate to represent error bursts while up to three states may be required to model the error-free gap distribution is easily explained. It is possible to have very short error-free gaps when the channel is very noisy as a result of low SNR, a Rayleigh distributed received power (deep fade notches) and high mobile unit speed (high irreducible error rate due to random FM). Similarly, very long error-free gaps may occur in a channel with high SNR, a Rician distributed received power (strong line-of-sight component and shallow fade notches), low mobile unit speed or combinations of these. However, error bursts rarely extend beyond a few bits because even in the worst case scenario where no signal power is incident at the receiver (e.g. a mobile unit is in the middle of a tunnel or in an underground garage), the receiver still has about a 50% chance of 'guessing' the correctly transmitted binary digits. Consequently, an error-burst of n consecutive bits occurs only with maximum probability 2^{-n} when no signal power reaches the receiver.

This is well illustrated by experimental results obtained by

Semmar et al. [60] where a high BER channel (10^{-1}) in a dense urban area (Quebec city) resulted in the following distribution of error bursts:

33% of error bursts had length $n = 1$

33% of error bursts had length $n = 2$

32% of error bursts had length $3 \leq n \leq 6$

1% of error bursts had length $7 \leq n \leq 13$

For channels with a lower BER, the distribution of bursts would be weighted even more heavily towards the lower end of the scale.

A sample EGD is shown in Figure 2.7 (page 43) for a four state model with parameters :

$$\alpha_1 = 0.7998, \beta_1 = 0.5997$$

$$\alpha_2 = 0.1019, \beta_2 = 0.9964$$

$$\alpha_3 = 0.0031, \beta_3 = 0.9996$$

and theoretical BER = 0.026213 (BER measured from channel file is 0.026366). The model is based on a real channel representing the downtown Ottawa region (medium urban density, Rician signal power distribution) [27]. As can be seen in Figure 2.14, the EGD curve for the Fritchman model is a sum of three weighted exponentials.

The EBD is shown in Figure 2.8 (page 43) and it is apparent that a single weighted exponential is sufficient to accurately represent the error-burst distribution, as mentioned earlier. The higher values of EBD for large m indicate that this channel model has more memory than the other two.

The autocorrelation function of Figure 2.6 (page 42) confirms that this channel model has the most memory. Furthermore, Figure 2.10 (page 48) clearly illustrates that there is some regularity in the recurrence of error clusters in the channel. As error clusters tend to occur at more regular intervals, the secondary peaks of the autocorrelation function become sharper and higher. Note again that for Figure 2.10, artificial offsets of 0.05

and 0.1 were added to the autocorrelation curves of the Gilbert and Fritchman models respectively so that all three curves could be shown on the same graph without overlapping.

The capacity for the Fritchman model channel with the parameters stated earlier is 0.9070906 bits per symbol. The fact that this value is larger than for the Gilbert (0.846889 bits/symbol) and BSC (0.830705

bits/symbol) models with a similar BER confirms that the Fritchman channel model has more memory than the other ones.

Figure 2.14: EGD for Fritchman model channel.

The parameters α_i and β_i of the Fritchman model can be obtained by curve fitting techniques based on the calculated error free gap distribution derived from experimental transmission runs in the real channel. Least squares approximation techniques [51] or gradient search algorithms [16,17] can be used to match the model parameters to the channel.

As for the other models, the algorithm cycles the chain once for each transmitted symbol x_i in order to generate E_i as follows:

```

1 --> GIVEN  $\alpha_1$  to  $\alpha_4$ ,  $\beta_1$  to  $\beta_4$ , n
2 --> Determine the a priori state probabilities
3 --> Build the Cumulative Distribution Function (CDF) matrix
4 --> Generate random number Z
      IF Z  $\in$  [0,  $P_1$ ] THEN initial state = 1
      IF Z  $\in$  ( $P_1$ ,  $P_1 + P_2$ ] THEN initial state = 2
      IF Z  $\in$  ( $P_1 + P_2$ ,  $P_1 + P_2 + P_3$ ] THEN initial state = 3
      IF Z  $\in$  ( $P_1 + P_2 + P_3$ , 1] THEN initial state = 4
5 --> For i = 1 to n do:
  a --> Generate random number Z in the range [0,1]
  b --> Given the present chain state, determine in which
        partition of the CDF the variable Z belongs to.
  c --> set chain state to the corresponding state of the CDF
        matrix.
  d --> IF new state is an error-free state,  $E_i = 0$ 
        ELSE  $E_i = 1$ 
6 --> Save the sequence  $\{E_1, E_2, \dots, E_n\}$  in a file.

```

where the CDF matrix is given by:

$$\begin{bmatrix}
 \beta_1 & 0 & \dots & \dots & 1 \\
 0 & \beta_2 & \dots & \dots & 1 \\
 0 & 0 & \beta_3 & \dots & 1 \\
 \vdots & \vdots & \vdots & \vdots & \vdots \\
 0 & 0 & \dots & \beta_{N-1} & 1 \\
 \alpha_1\beta_1 & \alpha_1\beta_1 + \alpha_2\beta_2 & \alpha_1\beta_1 + \alpha_2\beta_2 + \alpha_3\beta_3 & \dots & 1
 \end{bmatrix} \quad (2-48)$$

The main advantage of the Fritchman model is that it is more accurate than the Gilbert model, especially in fading channels where the BER is high. Furthermore, it was shown that a four-state model is sufficient to accurately represent most fading channels with a BER $\leq 10^{-1}$ [16,60]. Since the renewal model is completely specified by the EGD, it is only necessary to calculate that distribution from the experimental results in the real channel to be able to derive the model parameters. Furthermore, the algorithms to obtain these parameters are systematic and lend themselves well to computer solutions.

Another advantage of that model is that, by grouping all the

good states into one, the model becomes unifilar and the sequence of states for the chain can be deduced from the generated sequence of channel symbols. This simplifies somewhat the calculation of the channel capacity.

The main disadvantage of the Fritchman is that it is substantially more complicated than the Gilbert model. Furthermore, the renewal nature of the simplified model introduces inaccuracies in the simulation results (as described earlier).

2.3.6 Other channel models

Over the years, many other models have been proposed to represent fading channels. Among these, the Elliott model [24] is noteworthy in that it is a generalization of the Gilbert model. Elliott's model also has two states which represent two binary symmetric channels but the good state has a non-zero error probability in contrast with Gilbert's model where it is zero. The error probability in the good state is usually much smaller than in the bad state and is used to account for random, independent bit errors which also occur in a fading channel (which Gilbert ignored in his model). Elliott has found that this model represents telephone channels more accurately than Gilbert's model.

McCullough [37,46] has proposed a model which is a generalization of the Elliott model which provides more flexibility in the control of error-free gap processes. The model has several error producing states with different error rates. Furthermore, transitions between states are allowed only immediately after an error, thereby ensuring that the number of digits between errors is determined by a single state. This gives the model more flexibility without a significant increase in its mathematical complexity. Finally, the model can either be renewal or non-renewal, depending on the parameters used.

Another model of interest was introduced by Aulin [4] in 1981. The model consists of a Markov chain where the transition probabilities are a function of the received signal power level. The range of received signal power is split into non-overlapping partitions and the error process in each partition is represented by a binary symmetric channel. A Markov chain state corresponds to each partition and transitions to other states occur only when the power level at a given instant exceeds the range of the present partition. Although Aulin has applied the model to a channel with success, it is seldom used because of the complexity of the model.

Non-renewal models have been proposed by many authors [29,34,49] but the complexity involved in the parameter determination or the simulation algorithm has limited significantly their use in practice.

2.3.7 Channel capacity

In this section, we discuss some theoretical aspects of information theory, leading to the definition of channel capacity. Following this, we examine the effect of memory on channel capacity. Then, we develop equations which will allow us to calculate the channel capacity¹⁰ for the three simulation models used in this research work.

The channel capacity is a measure of the maximum information transfer rate achievable with an arbitrarily low error probability over a channel. As such, it is an important parameter which can be used, among other things, to choose a FEC code of optimal rate for a given communication link.

The mutual information of a communication channel is defined as

¹⁰ based on the simulated sequence $\vec{E}_s = \{E_1, E_2, \dots, E_n\}$.

$$I(x_i; y_j) = \log_2 \frac{P(x_i|y_j)}{P(x_i)} \quad (2-49)$$

in bits per symbol where x_i and y_j are specific symbols from the alphabets $\{x_1, x_2, \dots, x_T\}$ and $\{y_1, y_2, \dots, y_U\}$ respectively. $P(x_i|y_j)$ is the probability that x_i was sent given that y_j was received. When the mutual information is averaged over all possible pairs of x_i and y_j , one obtains the average mutual information $I(X; Y)$:

$$I(X; Y) = \sum_{x_i} \sum_{y_j} P(x_i, y_j) \log_2 \frac{P(x_i|y_j)}{P(x_i)} \quad (2-50)$$

However, in practice we usually have only the channel transition probabilities $P(y_j|x_i)$ and the source symbol a priori probabilities $P(x_i)$. Thus, we rewrite the above equation as:

$$I(X; Y) = \sum_{x_i} \sum_{y_j} P(x_i) P(y_j|x_i) \log_2 \frac{P(y_j|x_i)}{\sum_{x_k} P(x_k) P(y_j|x_k)} \quad (2-51)$$

If y_n is the symbol received at time n where x_n was transmitted, it is clear that $y_n = x_n + E_n$ where E_n is the channel error symbol added to the transmitted symbol at time n . Note that

$$\begin{aligned} X_n &\in \{x_1, x_2, \dots, x_T\} \\ Y_n &\in \{y_1, y_2, \dots, y_U\} \\ E_n &\in \{e_1, e_2, \dots, e_v\} \end{aligned} \quad (2-52)$$

where T , U and v are the alphabet size of the corresponding symbol (all equal to 2 for binary channels). In the forthcoming developments, the indices to capital letter variables always refer to time while those of small letter variables indicate a specific symbol in an alphabet. Capital letter variables without an index are used whenever the time is irrelevant (i.e. stationary, stochastic processes).

The equation for the average mutual information can be decomposed as follows:

$$I(X; Y) = H(X) - H(X|Y) \quad (2-53)$$

where $H(X)$ is the source entropy and $H(X|Y)$ is the equivocation, defined as the remaining uncertainty about the transmitted symbol x given the observation of the received symbol y . But

$$Y = X \oplus E \quad (2-54)$$

Thus,

$$H(X|Y) = H(X|X \oplus E) = H(X|E) \quad (2-55)$$

However, the channel symbol E is independent of the source symbol x . Therefore,

$$H(X|Y) = H(E) \quad (2-56)$$

The average mutual information between a received symbol and a transmitted symbol is then given by:

$$I(X; Y) = H(X) - H(E) \quad (2-57)$$

This leads to the channel capacity, defined as

$$C = \underset{P(X_i)}{\text{Max}} I(X; Y) = \underset{P(X_i)}{\text{Max}} H(X) - H(E) \quad (2-58)$$

For a binary memoryless source $H_{\text{max}}(X) = 1$ bit per symbol when $P(1) = P(0) = 0.5$ (i.e. an equiprobable source). Thus, the maximum channel capacity is given by:

$$C_{\text{max}} = 1 - H(E) = 1 - H_c \quad (2-59)$$

where H_c is the channel entropy.

For a sequence of transmitted (and received) symbols, mutual information and channel capacity become:

$$\begin{aligned} I(X_1, X_2, \dots, X_n; Y_1, Y_2, \dots, Y_n) &= H(X_1, X_2, \dots, X_n) - H(E_1, E_2, \dots, E_n) \\ C &= \text{Max } I(X_1, X_2, \dots, X_n; Y_1, Y_2, \dots, Y_n) \\ &= \text{Max } [H(X_1, X_2, \dots, X_n) - H(E_1, E_2, \dots, E_n)] \end{aligned} \quad (2-60)$$

where the average entropy of a sequence can be defined as:

$$H(E_1, E_2, \dots, E_n) = \lim_{n \rightarrow \infty} \sum_{E_1=e_1}^{e_V} \dots \sum_{E_n=e_n}^{e_V} P(E_1, E_2, \dots, E_n) h(E_1, E_2, \dots, E_n) \quad (2-61)$$

and where

$$h(E_1, E_2, \dots, E_n) = - \sum_{E_{n+1}=0}^{e_V} P(E_{n+1}|E_1, \dots, E_n) \log_2 P(E_{n+1}|E_1, \dots, E_n) \quad (2-62)$$

This formulation will allow us to determine the channel capacity from the simulated sequence $\vec{E}_s = \{E_1, E_2, \dots, E_n\}$.

2.3.7.1 Channel entropy as a function of memory

From the above development, it is apparent that the channel capacity is determined entirely from the entropy of the channel (assuming a maximum entropy source). However, the channel entropy is a function of both the error probability and of the amount of memory (clustering of errors) in the channel.

It is known that the entropy of a discrete memoryless channel is maximum [37]. Let us define H_o as the entropy of a memoryless BSC. Thus,

$$H_o = H(E) = - \sum_{e_i=0}^1 P(e_i) \log_2 P(e_i) = -p \log_2(p) - (1-p) \log_2(1-p) \quad (2-63)$$

Hence, for an arbitrary channel, $H_c \leq H_o$ and that channel has memory only when $H_c < H_o$. Kanal [37] defines μ -memory as

$$\mu = \frac{H_o - H_c}{H_o} \quad (2-64)$$

If $\mu = 0$, the channel has no memory. For $0 < \mu \leq 1$, the channel exhibits persistent or cyclic memory. As μ increases, the channel memory increases and errors become more clustered (persistent memory) or appear more regularly (cyclic memory). Thus, if the channel capacity of a DMC is given by $C_o = 1 - H_o$, we may express the

channel capacity for an arbitrary channel as:

$$C = 1 - H_c = C_o + \mu H_o \quad (2-65)$$

It is therefore obvious that a channel with memory has a higher theoretical capacity than a memoryless channel. This is not surprising since there is less uncertainty about when errors occur in a fading channel because those errors occur in bursts. However, this higher capacity may only be achieved if the memory of the channel is exploited. This is difficult to do and requires a more complex communications system than when memory is not exploited.

For example, automatic repeat request (ARQ) systems exploit memory by taking advantage of error-free gaps to transmit data at a high throughput by using a high rate FEC code. The FEC code is used at the receiver to detect the occurrence of errors. When errors do occur, the receiver sends a request to the transmitter for a retransmission of the corrupted data. In this case, the added complexity takes the form of a feedback channel from the receiver to the transmitter. By contrast, the use of interleaving does not utilize channel memory and the capacity of a channel where it is used is suboptimal [37].

2.3.7.2 Capacity of a binary symmetric DMC

Consider the following definition of channel entropy for an ergodic process:

$$H_c = H(E_1, E_2, \dots, E_n) = \lim_{n \rightarrow \infty} \frac{1}{n} h(E_1, E_2, \dots, E_n) \quad (2-66)$$

But since the channel is memoryless, this becomes [9, pp. 151]:

$$H_c = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n h(E_i) = h(E_1) = h(E) \quad (2-67)$$

where $h(E)$ is simply

$$h(E) = - \sum_{e_i=0}^1 P(e_i) \log_2 P(e_i) = -p \log_2(p) - (1-p) \log_2(1-p) \quad (2-68)$$

as seen earlier. Thus, the channel entropy for the ergodic sequence E is $H_c = h(E)$. As a result, the maximum channel capacity is:

$C_{MAX} = 1 + p \log_2 p + (1-p) \log_2 (1-p) \quad (2-69)$
--

2.3.7.3 Capacity of a Gilbert model channel

One definition of entropy for an ergodic process is given by:

$$H(E_1, E_2, \dots, E_n) = \lim_{n \rightarrow \infty} \sum_{E_1=0}^1 \dots \sum_{E_n=0}^1 P(E_1, E_2, \dots, E_n) h(E_1, E_2, \dots, E_n) \quad (2-70)$$

where [31]:

$$h(E_1, E_2, \dots, E_n) = - \sum_{E_{n+1}=0}^1 P(E_{n+1} | E_1, \dots, E_n) \log_2 P(E_{n+1} | E_1, \dots, E_n) \quad (2-71)$$

From only the knowledge that $E_i = 0$, it is not possible to determine the chain state since Gilbert's model is non-unifilar. However, if $E_i = 1$, the chain state is certainly B . Thus,

$$P(E_{i+1} | E_1, \dots, E_{i-1}, 1) = P(E_{i+1} | 1) \quad (2-72)$$

because the model is a first order chain. Therefore,

$$\begin{aligned} P(E_{n+1} | E_1, \dots, E_{i-1}, 1, E_{i+1}, \dots, E_n) &= P(E_{n+1} | 1, E_{i+1}, \dots, E_n) \\ h(E_1, \dots, E_{i-1}, 1, E_{i+1}, \dots, E_n) &= h(1, E_{i+1}, \dots, E_n) \end{aligned} \quad (2-73)$$

and it becomes apparent that the entropy of an arbitrary sequence is completely determined by the number of consecutive zeros at the end of a block. Each of the 2^n terms $h(E_1, \dots, E_n)$ in equation (2-70) belong to the following set of possible realizations: $h(1), h(10), \dots, h(10^k), \dots, h(10^{n-1}), h(10^n)$. By using this simplification in equation (2-70), and performing the limit operation, we get:

$$H_c = \sum_{k=0}^{\infty} P(10^k) h(10^k) \quad (2-74)$$

But by Bayes' theorem,

$$P(10^k) = P(1) P(0^k|1) \quad (2-75)$$

where $P(1)$ is the same as $P(e)$ calculated previously in equation (2-34) (page 45). Furthermore, since

$$P(0|10^k) = \frac{P(0, 10^k)}{P(10^k)} = \frac{P(10^{k+1})}{P(10^k)} = \frac{P(1) P(0^{k+1}|1)}{P(1) P(0^k|1)} = \frac{P(0^{k+1}|1)}{P(0^k|1)} \quad (2-76)$$

one obtains:

$$h(10^k) = -\frac{P(0^{k+1}|1)}{P(0^k|1)} \log_2 \frac{P(0^{k+1}|1)}{P(0^k|1)} - \left[1 - \frac{P(0^{k+1}|1)}{P(0^k|1)}\right] \log_2 \left[1 - \frac{P(0^{k+1}|1)}{P(0^k|1)}\right] \quad (2-77)$$

Using the above equation in equation (2-74) and recalling that the maximum channel capacity is $C_{MAX} = 1 - H_c$ (for a maximum entropy source), Gilbert found [31] that:

$$\begin{aligned} C &= 1 + P(1) \sum_{k=0}^{\infty} [P(0^k|1) - P(0^{k+1}|1)] \log_2 [P(0^k|1) - P(0^{k+1}|1)] \\ C &= 1 + P(1) \sum_{k=0}^{\infty} P(0^{k+1}|1) \log_2 P(0^{k+1}|1) \end{aligned} \quad (2-78)$$

The sum converges slowly but Gilbert simplified it by demonstrating that for a large k ,

$$h_o = \lim_{K \rightarrow \infty} h(10^K) = -J \log_2 J - (1-J) \log_2 (1-J) \quad (2-79)$$

where J is given in equation (2-36) (page 45). This leads to the following simplification:

$$\begin{aligned} H_c &= \left(\sum_{k=0}^{K_o-1} P(10^k) h(10^k) \right) + \sum_{k=K_o}^{\infty} P(10^k) h_o \\ H_c &= \left(\sum_{k=0}^{K_o-1} P(10^k) h(10^k) \right) + h_o P(1) \sum_{k=K_o}^{\infty} P(0^k|1) \\ H_c &= \left(\sum_{k=0}^{K_o-1} P(10^k) h(10^k) \right) + h_o \left[1 - P(1) \sum_{k=0}^{K_o-1} P(0^k|1) \right] \end{aligned} \quad (2-80)$$

where K_0 is moderately large (e.g. from 10 to 15). Maximum channel capacity is then derived from $C_{MAX} = 1 - H_c$ and is

$$C_{MAX} = 1 - \left(\sum_{k=0}^{K_0-1} P(10^k) h(10^k) \right) - h_o \left[1 - P(1) \sum_{k=0}^{K_0-1} P(0^k|1) \right] \quad (2-81)$$

for the Gilbert model. When expressed in terms of the EGD only, this becomes:

$$\begin{aligned} C_{MAX} = 1 + P(1) \sum_{k=0}^{K_0-1} P(0^k|1) & \left(\frac{P(0^{k+1}|1)}{P(0^k|1)} \log_2 \frac{P(0^{k+1}|1)}{P(0^k|1)} \right. \\ & \left. + \left[1 - \frac{P(0^{k+1}|1)}{P(0^k|1)} \right] \log_2 \left[1 - \frac{P(0^{k+1}|1)}{P(0^k|1)} \right] \right) \\ & - h_o \left[1 - P(1) \sum_{k=0}^{K_0-1} P(0^k|1) \right] \end{aligned} \quad (2-82)$$

For example, given a Gilbert model with $p = 0.12425$, $P_{gg} = 0.98$, $P_{bb} = 0.92$ and a corresponding (theoretical) BER of 0.0249, the channel capacity is 0.846889 bits per symbol. Note that for a BSC memoryless channel of similar BER, it was seen earlier that $c = 0.830705$ bits/symbol. The BSC capacity is lower than that of the Gilbert channel with memory, in accordance with theory.

2.3.7.4 Capacity of a Fritchman model channel

Again, we proceed from the knowledge that $C_{MAX} = 1 - H_c$ under the assumption of a maximum entropy source for the sequence $\bar{X} = \{X_1, X_2, \dots, X_n\}$. The channel entropy can be defined as:

$$H_c = H(E_1, E_2, \dots, E_n) = \lim_{n \rightarrow \infty} \frac{1}{n} h(E_1, \dots, E_n) \quad (2-83)$$

where

$$\lim_{n \rightarrow \infty} \frac{1}{n} h(E_1, E_2, \dots, E_n) = \lim_{n \rightarrow \infty} h(E_n | E_1, E_2, \dots, E_{n-1}) \quad (2-84)$$

assuming that all sequences are equiprobable ($P(E_1, E_2, \dots, E_n) = \frac{1}{v^n}$,

where v is the alphabet size). We can further develop equation (2-84) as follows (for a binary channel):

$$\begin{aligned} h(E_n|E_1, \dots, E_{n-1}) &= - \sum_{E_1=0}^1 \dots \sum_{E_{n-1}=0}^1 P(E_1, \dots, E_n) \log_2 P(E_n|E_1, \dots, E_{n-1}) \\ &= - \sum_{E_1=0}^1 \dots \sum_{E_{n-1}=0}^1 P(E_1, \dots, E_{n-1}) \sum_{E_n=0}^1 P(E_n|E_1, \dots, E_{n-1}) \\ &\quad \times \log_2 P(E_n|E_1, \dots, E_{n-1}) \end{aligned} \quad (2-85)$$

Furthermore, for the simplified model with only one error state, we know that

$$P(E_n|E_1, \dots, E_i=1, \dots, E_{n-1}) = P(E_n|E_i=1, \dots, E_{n-1}) \quad (2-86)$$

Then, equation (2-85) can be rewritten as

$$\begin{aligned} h(E_n|E_1, \dots, E_{n-1}) &= - \sum_{m=0}^{n-2} P(10^m) \sum_{E_n=0}^1 P(E_n|10^m) \log_2 P(E_n|10^m) \\ &\quad - P(0^{n-1}) \sum_{E_n=0}^1 P(E_n|0^{n-1}) \log_2 P(E_n|0^{n-1}) \end{aligned} \quad (2-87)$$

But Bayes' theorem gives us the following relations

$$\begin{aligned} P(E_n|10^m) &= \frac{P(10^m E_n)}{P(10^m)} = \frac{P(0^m E_n|1)}{P(0^m|1)} \\ P(1|10^m) &= \frac{P(0^m 1|1)}{P(0^m|1)} = \frac{P(0^m|1) - P(0^{m+1}|1)}{P(0^m|1)} \\ P(0|10^m) &= \frac{P(0^{m+1}|1)}{P(0^m|1)} \end{aligned} \quad (2-88)$$

By using these relationships, Fritchman found that [29]:

$$\begin{aligned} h(E_n|E_1, \dots, E_{n-1}) &= \\ &= -P(1) \sum_{m=0}^{n-2} P(0^m|1) \left[\frac{P(0^{m+1}|1)}{P(0^m|1)} \log_2 \left(\frac{P(0^{m+1}|1)}{P(0^m|1)} \right) + \frac{P(0^m 1|1)}{P(0^m|1)} \log_2 \left(\frac{P(0^m 1|1)}{P(0^m|1)} \right) \right] \\ &\quad - P(0^{n-1}) \sum_{E_n=0}^1 P(E_n|0^{n-1}) \log_2 P(E_n|0^{n-1}) \end{aligned} \quad (2-89)$$

By taking the limit as n approaches infinity, the last term of the

previous equation approaches zero. The channel entropy then simplifies to [29]:

$$\begin{aligned}
 H_c &= -P(1) \sum_{m=0}^{\infty} [P(0^m|1) - P(0^{m+1}|1)] \log_2 [P(0^m|1) - P(0^{m+1}|1)] \\
 &\quad - P(0^{m+1}|1) \log_2 P(0^{m+1}|1) + P(0^m|1) \log_2 P(0^m|1) \\
 H_c &= -P(1) \sum_{m=0}^{\infty} P(0^{m+1}|1) \log_2 P(0^{m+1}|1) \\
 &\quad - P(0^{m+1}|1) \log_2 P(0^{m+1}|1) + P(0^m|1) \log_2 P(0^m|1)
 \end{aligned} \tag{2-90}$$

When expressed in terms of the EGD only, the maximum channel capacity for the Fritchman model becomes

$$\begin{aligned}
 C_{MAX} &= 1 + P(1) \sum_{m=0}^{\infty} [P(0^m|1) - P(0^{m+1}|1)] \log_2 [P(0^m|1) - P(0^{m+1}|1)] \\
 &\quad + P(0^{m+1}|1) \log_2 P(0^{m+1}|1) - P(0^m|1) \log_2 P(0^m|1)
 \end{aligned} \tag{2-91}$$

Unlike the equation for the capacity of the Gilbert model, the above sum is unbounded. However, the probability of long runs of zeros decreases as the BER increases [60, pp. 773] and a good approximation can be achieved with a reasonable upper limit on m when the BER is above 10^{-3} .

It was seen earlier that for a four state Fritchman channel with the parameters given on page 55, the capacity is $c = 0.907906$ bits per symbol (for a BER = 0.026213). By contrast, the capacity of the memoryless BSC channel of similar BER was seen to be 0.830705 bits per symbol. Again, it is apparent that the capacity is higher than for a memoryless channel with the same BER. The fact that the capacity is also higher than for the Gilbert model with the same BER (capacity = 0.846889) implies that the error clusters are more dense and better defined for the channel simulated by the Fritchman model than for the one represented by the Gilbert model (thus, lower uncertainty about the occurrence of errors).

2.4 Conclusion

In this chapter, the characteristics of the digital mobile communication channel were examined in detail. The effect of near and far scatterers, of vehicle speed, and of modulation techniques on the signal propagation characteristics in the channel were analyzed. More specifically, fast and slow fading were discussed and their impact on the channel memory was considered. Statistical models describing the signal power level were described and their advantages and disadvantages were pointed out. The irreducible BER caused by random FM was mentioned and its relationship to the mobile unit speed was established. Finally, it was seen that parameters such as delay spread, coherence bandwidth, level crossing rate and average fade duration yield useful additional information about the channel.

In the second part, the concepts of renewal and non-renewal processes were described in relation to real fading channels and their simulation models. The consequences of using renewal models to represent real channel processes (which are essentially non-renewal) were examined. The advantages of renewal models were also explained. Following this, three channel models were presented; the binary symmetric channel and the Markov chain models proposed by Gilbert and Fritchman. The equations for the statistical distributions of error-free gaps and error bursts were given for each model, as well as other relevant statistics. Methods of matching the model parameters to the real channel to simulate were mentioned and the algorithms used to generate the sequence of error digits $\bar{E}_n = \{E_1, E_2, \dots, E_n\}$ were described. Finally, the capacity of each channel model was developed in detail and it was shown that channels with memory have a higher capacity than a memoryless channel of equivalent bit error rate.

Chapter 3

Cryptography: theoretical considerations and practical implementation in a digital communication channel

3.1 Introduction

The unauthorized interception of conversations or data transfers over a mobile communications link has been discussed earlier in chapter one. With the rapid growth in popularity of such systems, it is safe to assume that the problem of eavesdropping will only get worse. Because such transmissions can easily be intercepted with relatively inexpensive receivers and since it is virtually impossible to prevent people from tuning in to these broadcasts from the safety of their homes, it is clear that the users of these services will have the burden of protecting their own transmissions to ensure their privacy.

In this chapter, we examine various encryption techniques which can provide adequate protection of conversations or data transfers over this medium. First, some basic theory of cryptography is studied in order to determine what constitutes a good encryption system. Then, we examine the Data Encryption Standard (DES) and demonstrate its suitability as a good encryption system which can provide low to medium security for digital applications over public airwaves. This will be followed by a description of practical implementations of the DES in stream and block cipher modes for digital transmission systems. Four specific modes will be analyzed:

- a) Electronic Code Book (ECB),
- b) Cipher Block Chaining (CBC),
- c) Output Feedback Mode (OFB), and

d) Cipher Feedback Mode (CFB).

The advantages and disadvantages of each mode and their performance relative to each other will be discussed in detail. Finally, a brief description of future trends in cryptography will be provided.

3.2 Theoretical considerations on cryptography

After a basic definition of a generic encryption system, this section will introduce the concepts of unicity distance and key equivocation. Subsequently, a system which provides complete theoretical security is examined. We will then examine how practical encryption systems, which are not theoretically and unconditionally secure, can be made safe enough for most applications. More specifically, we will examine the concepts of confusion, diffusion and mixing functions (defined by C. E. Shannon [62]).

3.2.1 A generic encryption system

All encryption systems serve two main purposes:

- a) preventing unauthorized eavesdroppers from determining the message sent, and
- b) allowing legitimate recipients to recover the message easily.

In order to accomplish this, the input message (plaintext) must be transformed into an unrecognizable format (cipher text). The transformation functions must be complex enough to prevent an unauthorized interceptor from working backwards from the cipher to recover the message. However, the transformations must be easily reversible so that an authorized recipient may retrieve the message

quickly and reliably.

Consider a basic encryption system using an algorithm $E_K(\cdot)$ which transforms an input message block $M_m = \{m_i, m_{i+1}, \dots, m_{i+N-1}\}$ of N bits into a cipher block $C_m = E_K(M_m)$ of N bits where $C_m = \{c_i, c_{i+1}, \dots, c_{i+N-1}\}$ and which uses the key $K = \{k_i, k_{i+1}, \dots, k_{i+J-1}\}$ (with J bits) as a basis for the transformation of the message. The advantage of using a key to determine the specific nature of the encryption algorithm lies in the fact that the safety of the encryption algorithm depends mostly on the key and not on the specific steps of the encrypting function. Thus, even if the interceptor knows the transformation function $E_K(\cdot)$, he should still be unable to recover the message. However, the authorized recipient knows the key used and can easily perform the deciphering operation in order to get $M_m = D_K(C_m)$. Of course, this implies that the transformation function is reversible.

3.2.2 Unicity distance and equivocation

Shannon defined the unicity distance U_d as [62]:

$$U_d = \frac{H(K)}{D} \quad (3-1)$$

where $H(K)$ is the key entropy, given by:

$$H(K) = -\sum_{i=1}^N P(k_i) \log_2 P(k_i) \quad (3-2)$$

and D is the redundancy of the input message in bits per symbol. The unicity distance represents the theoretical minimum number of cipher symbols required to uniquely determine the key used. Cryptographic security increases as the unicity distance increases. This can be accomplished by increasing the uncertainty about the key (i.e. using a longer key), or removing some of the redundancy in the input message (i.e. source compression [36]).

Equivocation is another concept which provides useful clues about the strength of an encryption algorithm. Key equivocation $H(K|C)$ is defined as:

$$H(K|C) = - \sum_{K_i=1}^m \sum_{C_j=1}^n P(K_i, C_j) \log_2 P(K_i|C_j) \quad (3-3)$$

where $P(K_i, C_j)$ is the joint probability that key K_i (with J bits) is used and cipher block C_j (with N bits) is received, while $P(K_i|C_j)$ is the conditional probability that key K_i was used given that cipher block C_j was received. The sum is made over all possible keys and all possible cipher blocks. Key equivocation is therefore the average amount of uncertainty about the key K , given that the cipher sequence C has been received. The message equivocation is defined as:

$$H(M|C) = - \sum_{M_i=1}^r \sum_{C_j=1}^n P(M_i, C_j) \log_2 P(M_i|C_j) \quad (3-4)$$

where $P(M_i, C_j)$ is the joint probability that the message block M_i (of N bits) was encrypted and that cipher block C_j (of N bits) was received, while $P(M_i|C_j)$ is the conditional probability that message block M_i was encrypted given that cipher block C_j was received. The message equivocation gives us the average amount of uncertainty about the original message sequence M , given that the cipher sequence C has been received. In real systems, the key and message equivocation always decrease to zero as the number of cipher symbols intercepted becomes large¹.

3.2.3 An encryption system with perfect secrecy

For a theoretical system with perfect secrecy, the key and

¹ It is important to note that neither the unicity distance nor the equivocation provide the cryptanalyst with any clue as to how the code may be broken; they merely provide a theoretical lower bound on the quantity of cipher text required to do so.

message equivocation remain constant and the unicity distance is infinite. For such a system, $H(M|C) = H(M)$, $H(K|C) = H(K)$. This implies that the cipher blocks intercepted provide absolutely no information about the message stream or the key. It can be shown that this only occurs when the key used to encrypt the message M is at least as long as the message M itself [5, pp. 146].

The one-time pad cipher [5, pp. 148-150] is an example of a perfectly secure system. It provides complete secrecy when the key is completely random, is at least as long as the message and is never repeated. Such systems are seldom used in practice because the key length becomes excessive when a large amount of data is to be transmitted.

3.2.4 Confusion, diffusion and mixing functions

In most applications, the key size must be limited and thus, most real encryption systems are inherently unsafe. However, it is possible to make a practical encryption system secure by ensuring that the relationship between the key, the plaintext and the cipher is complex enough to make it extremely difficult for a cryptanalyst to uncover that relationship. Shannon [62] studied the properties required of a good encryption system and developed the important notions of confusion, diffusion and mixing functions.

An encryption system is said to have a high degree of confusion when:

- a) the relationship between the output cipher and the key is very complex, and
- b) when the entire key is used to encrypt every message bit.

The first condition ensures that the amount of work (and time) required to develop equations mapping input symbols into the corresponding output symbols is too high for a cryptanalyst to

attempt it. The second condition is essential to ensure that the cryptanalyst will not be able to deduce the key symbol by symbol, but will have to solve a large number of simultaneous equations to determine its value. The following example should make this clear.

Consider a simple substitution algorithm which implements $c_i = m_i + k_i \text{ (modulo 26)}$, where m_i , c_i and k_i are letters of the alphabet and each letter is associated with a numerical value (i.e. $a = 0, b = 1, \dots, z = 25$)². Enciphering is accomplished by writing the plaintext above the key and performing the modulo 26 addition to get the cipher text:

```

PLAINTEXT --> cryptography...
KEY        --> DESDESDESDES...
CIPHER     --> FVQSXGJVSSLR...

```

where the key ($K = \{k_1, k_2, k_3\} = \{D, E, S\}$) is repeated as often as necessary to encrypt all of the message. Note that although the key is 3 symbols long, only one symbol is used to encrypt each plaintext symbol. As a result, the cryptanalyst can try to guess each key symbol individually with an a priori probability of 1/26 of finding the right one. However, if the encryption algorithm is changed as follows (for example),

$$c_i = \left(\left[(m_i + k_1) \times k_2 \right] - \left[k_3 \times (k_2 - k_1) \right] \right) \text{ modulo } 26 \quad (3-5)$$

each cipher symbol becomes a function of the entire key and the cryptanalyst's task is made more difficult. It is therefore apparent that confusion is a desirable feature of any good encryption system.

An encryption system is said to possess a high degree of diffusion when each cipher output symbol is equally dependent on

² This is a Vigenère cipher [5, pp. 32].

all symbols of the input plaintext. When diffusion is present, the statistical distribution of the input plaintext does not resurface in the cipher text. Ideally, the statistical distribution of cipher text symbols is uniform and has no obvious relationship to the input text, thereby depriving the cryptanalyst of important clues about the latter. An example will illustrate this.

Consider again the former two examples, where only one input symbol is encrypted at a time. As a result of this, the well known statistical distribution of letters and groups of letters of the English language [59, pp. 300-320] re-appears, unchanged, in the cipher text. Such encryption systems can easily be broken [59, pp. 61-76], regardless of the degree of confusion between the key and cipher text. This is accomplished by determining the probability density function of the cipher symbols and matching them with the pdf of plain English letters (assuming the original message was in English). Referring again to the previous example, the following modification to the encryption algorithm illustrates a simple way of adding diffusion:

$$c_i = \left[m_i (k_1 + k_2 - k_3) \times m_{i-1} (k_3 (k_2 + k_1)) - m_{i+1} (k_1 + k_2 + k_3) \right] \text{ modulo } 26 \quad (3-6)$$

Note that three input symbols are used to generate each output symbol and that the entire key is used with each symbol. Thus, there is both diffusion and confusion. The cryptanalyst now has to solve six equations with six unknowns to determine each symbol of the key or plaintext. Furthermore, the statistical distribution of each plaintext symbol is diffused with other symbols. As a result, the pdf of the cipher symbols will be different from that of the input message symbols.

The concept of mixing functions was introduced by Shannon in recognition of the fact that achieving thorough confusion and

diffusion all at once would involve excessively complex equations. This is evident in the last example provided, where a relatively low degree of confusion and diffusion requires a fairly involved equation. When one considers the need to achieve confusion and diffusion over dozens or hundreds of key and plaintext symbols, the task becomes nearly impossible. Rather than attempting this, Shannon [62] proposed using a relatively simple set of operations (permutations and substitutions) but repeating them recursively many times in order to eventually achieve a high degree of confusion and diffusion. The exact nature of the basic set of operations has been the subject of intensive research in the field of cryptography and is beyond the scope of this work. Nevertheless, the description of the DES in the next section and the references provided throughout this chapter will provide specific examples of the type of functions used. For now, it is sufficient to say that Shannon's idea of using mixing functions to achieve diffusion and confusion has been widely accepted by the cryptographic community.

It is worthwhile to note that the security of the encryption algorithm depends on both confusion and diffusion being achieved to a high degree. For example, it was seen that equation (3-5) provides confusion but is very easy to defeat because the algorithm does not diffuse the statistics of the input symbols over the output symbols. Furthermore, even a system with a high degree of both is not necessarily invulnerable. The key must be long enough to preclude the feasibility of breaking the code by trying out all key combinations. For the last example provided, the key is only 3 symbols long and a search through the entire key space would take only fractions of a second on a computer.

3.3 The Data Encryption Standard

The DES is an encryption system which incorporates Shannon's ideas described in the last section. The DES [5, pp. 267-290] was introduced in 1977 by the U.S. Department of Commerce. The

algorithm consists of 16 recursive rounds of encryption performed on a 64 bits message and using a key with 56 bits³. The sequence of events for one round of encryption is illustrated in Figure 3.1.

After an initial permutation (done once only), the 64 bit message is split in two blocks of 32 bits each. The right block is expanded to 48 bits and added modulo-2 with a 48 bit subkey based on the 56 bits key. The resultant block is reduced back to 32 bits by using a series of 6 to 4 bits substitution boxes. The 32 bits block is permuted and then added modulo-2 with the left block of 32 message bits. The right block of 32 message bits is then shifted into the left block and the block 'TEMP 32' (in the figure) is moved into the right block of 32 message bits. This is repeated 16 times, after which the left and right blocks are passed through the inverse of the initial permutation. The result is an output cipher block which is also 64 bits long.

For the DES, the mixing function as defined by Shannon is achieved by repeating the encryption algorithm 16 times. The 48 bit subkey is different at each round of encryption. Furthermore, the subkey permutations are selected in such a way that, after 16 rounds, each bit of the 56 bit key has made an equal contribution to the output cipher. In order to accomplish this, boxes 'C28' and 'D28' perform a left shift of one or two bits at each iteration (with the most significant bit being rotated into the least significant bit position). Permutation 'PC2' will then select 24 bits from each 28 bits register to create the sub-key for the present iteration. The left shifts ensure that the key bits occupy different positions in 'C28' and 'D28' at each iteration. After 16 rounds, all key bits have participated in the encryption process an equal number of times (on average) and have appeared in different positions of the sub-key (output of 'PC2') each time. Thus, a high level of confusion is achieved.

³ The key is really 64 bits long but eight of them are parity bits.

Data Encryption Standard (DES)

Figure 3.1: The Data Encryption Standard algorithm

Similarly, the expansion (32 bits to 48) and compression (48 bits to 32) sub-functions are at the heart of the diffusion of the message bit statistics. The repeated application of these steps results in a high level of diffusion throughout the 64 bits of the output cipher block.

The high level of confusion is easily demonstrated by an example. Consider a message $M = \text{AAAAAAAA AAAAAAAAAA}$ (hexadecimal) which yields the cipher vector $C = 0F36FAF1\ 1BB35D0C$ when the key $K = 4B02F19D\ E37AC5F6$ is used. The table below shows the cipher produced by trial keys for the above message (d_H is the Hamming distance between the trial key and the original key, or the trial cipher block and the original cipher block, as appropriate):

TRIAL KEY		d_H	CIPHER		d_H
-----			-----		
4B02F19D	E37AC5F4	1	5FAE9508	0CD12176	34
4B12F19D	E37AC5F6	1	4A1592BA	A0BAA329	31
4B02F19D	F37AC5F6	1	4DE2D03F	BA75E50C	25
2D9A0765	4FB13E79	39	57BBB347	CB163728	28

In all cases except one, the trial key differs from the real key by only one bit but the resultant cipher differs from the one produced by the real key by almost half the bits. Since two totally uncorrelated blocks of n bits should have a Hamming distance of about $N/2$ on average, it is clear that for the DES a small change in the key results in a large change in the resultant cipher text for a given message block⁴. It can easily be verified that all key combinations with a Hamming distance of one or two will produce cipher blocks with an average Hamming distance close to 32 bits.

⁴ This is sometimes referred to as the avalanche effect [1].

The last trial key of the table illustrates clearly that there is no correlation between how close a trial key is to the real one and the Hamming distance of the corresponding cipher blocks. An encryption system with this property is said to have a high degree of confusion. Note that a cryptanalyst trying to determine the actual key K by a known plaintext attack (i.e. by trying different keys to generate C from M) would receive no indication that a trial key is close to the real key by examining the Hamming distance of the cipher text. This is an essential property of any robust encryption algorithm.

The high degree of diffusion achieved by the DES can also be demonstrated by an example. In this case, the key K is kept constant but the message M is modified slightly. The table below illustrates clearly that changing only one bit in the 64 bit message results in about half the resultant cipher bits being different.

(K = 4B02F19D E37AC5F6)					
TRIAL MESSAGE		d _H	CIPHER		d _H

AAAAAAAA	AAAAAAB	1	FEAAEC82	E611FFCB	35
8AAAAAAAA	AAAAAAAA	1	EC6F0C94	17228D2B	31
AAAAAAB	AAAAAAAA	1	DD5E66CE	85F5A00E	33

This is another example of the avalanche effect and it clearly illustrates that each bit of the output cipher block is equally dependent on every single bit of the input message block. Thus, information about each input bit is dispersed throughout the 64 output bits and the encryption algorithm displays a high degree of diffusion. This makes it extremely difficult for a cryptanalyst to use statistical analysis to infer the message text from the cipher text.

It is interesting to note that if plain English text is encrypted, the unicity distance for the DES is only:

$$U_d = \frac{H(K)}{D} = \frac{56}{3.2} = 17.5 \text{ Characters} \quad (3-7)$$

where the language redundancy is assumed to be 3.2 characters per symbol [21, pp. 27]. Thus, in theory, only 18 characters of cipher text are required to uniquely define the key. However, the security of the DES is higher than the unicity distance implies since the latter only provides a theoretical lower bound on the number of cipher bits required to break the cipher; it does not provide any clues as to how the cipher can be broken. As a matter of fact, no one has yet been able to develop an algorithm to actually break the DES; most attacks proposed against it are variations of an exhaustive search technique [33,47]⁵.

Nevertheless, the feasibility of searching through the entire key space to break the cipher cannot be dismissed. Diffie and Hellman [22] estimated that a \$20 million dollar computer could search through the entire key space (2^{56} different keys) in one day. But that article was published in 1976. If we factor in the drop in prices and the increase in computing power of modern chips, it becomes apparent that the DES could not withstand a determined attempt to break it. Thus, it is clear that one of the most serious flaws of the DES lies in the fact that its key size is too small to provide a high level of security.

Another problem of DES related to the keys involves so-called groups of "weak keys", where two or more different keys yield the same output cipher [59, pp.337-339]. Other security issues have been raised about the DES [5, pp. 282-285], [53, pp.117-119]. These concern the permutation, substitution and expansion-compression

⁵ Differential cryptanalysis [7] has shown some success in correlating input bit patterns to cipher output bit patterns and can break DES in 2 minutes if only 8 rounds of encryption are used.

functions used in the DES.

Despite its apparent flaws, the DES is still more than adequate for applications which require low to medium security. The high cost in time and effort required to break the DES is likely to deter all but the most dedicated cryptanalyst. Furthermore, the eavesdropper's task can be made more difficult by the frequent change of keys. Therefore, the DES remains a suitable candidate for encryption of digital data transfers in a mobile communication channel.

3.4 Implementation of encryption systems based on the DES

In this section, we describe the different types of encipherment systems which can be used in a digital communication channel. Despite the fact that these systems are described with the DES as a component, it is important to note that the latter can be viewed as a black box inserted in the overall system. Thus, the encipherment systems are generic and the DES could be replaced by other encryption systems (e.g. REDOC II [20] or FEAL-n [48,63]) as required.

There are two basic modes of operations for cipher systems:

- a) Block ciphers, and
- b) Stream ciphers.

Block ciphers are further broken down into Electronic Code Book mode (ECB) and Cipher Block Chaining mode (CBC). Similarly, stream encipherment includes Output FeedBack (OFB) mode and Cipher FeedBack mode (CFB). The operation, advantages, and disadvantages of these different systems will now be discussed.

3.4.1 Block ciphers

For block ciphers, a message $M_m = \{m_i, m_{i+1}, \dots, m_{i+N-1}\}$ (where N is the word length in bits), is encrypted as a unit, using the entire key $K = \{k_1, k_2, \dots, k_J\}$ (J is the key length in bits), resulting in the cipher word $C_m = \{c_i, c_{i+1}, \dots, c_{i+N-1}\}$ being generated. For the case of the DES, the key K , the message M_m and the output cipher block C_m are all 64 bits long⁶. Successive message blocks $M_1, M_2, \dots, M_m, \dots$ are encrypted with the same key K . The ciphered blocks $C_m = E_K(M_m)$ are transmitted over the channel and the receiver performs the decipherment operation $M'_m = D_K(C'_m)$ to recover the message block. The 'prime' symbol emphasizes the fact that the received block (and decrypted block) may differ from the one at the transmitter because of errors introduced by the channel. The main advantage of block cipher systems is that they provide a good level of security because of the high degree of confusion and diffusion resulting from the encryption of a large block of data with the entire key. Another advantage to that approach is that it is impossible for an eavesdropper to modify an intercepted block and sending it to the intended receiver without detection. This was demonstrated earlier with the DES, where the modification of a single bit in the ciphered block would alter about half the bits of the message block after decryption (avalanche effect).

The main disadvantage of block ciphers is error propagation. If a single bit of a cipher block is transmitted in error in the channel, the entire message block becomes corrupted during decryption (about half the bits will be in error). Another problem with block encipherment is the extra delay introduced at the time of encryption and decryption. The transmitter must wait until it has 64 bits of message before a cipher block can be generated.

⁶ As mentioned before, the actual key size is only 56 bits but 8 parity bits have been added to make a 64 bits block.

Similarly, the receiver must wait until all 64 bits of a cipher block are received before beginning decryption.

3.4.1.1 Electronic Code Book (ECB) mode

The Electronic Code Book mode of operation is illustrated in Figure 3.2. In this mode of operation, a block M_m of $N = 64$ bits is encrypted with a pre-determined key K into a cipher block C_m which is then transmitted over the channel. The latter will occasionally introduce bit errors, as described in chapter two. As a result, the received cipher block C'_m may not be identical to the transmitted block C_m . Nevertheless, the receiver performs the decryption operation on C'_m by using key K and generates the message block M'_m . If $C'_m = C_m$, then $M'_m = M_m$. However, if C'_m differs from C_m by one or more bits, M'_m will be completely different from M_m because of the avalanche effect (on average, the two will differ by about 32 bits). The operations of encryption and decryption are shown below:

Figure 3.2: ECB block cipher mode.

$$\begin{aligned}
 C_m &= E_K(M_m) \\
 C'_m &= C_m \oplus E_m \\
 M'_m &= D_K(C'_m) = D_K(C_m \oplus E_m) \\
 M'_m &= M_m \text{ iff } E_m = \vec{0}
 \end{aligned}
 \tag{3-8}$$

where $E_m = \{e_1, e_{1,1}, \dots, e_{1,N-1}\}$ is the vector representing bit errors introduced by the channel. When no errors occur,

$e_j = 0$, ($i \leq j \leq i+N-1$). Individual bits from the transmitted cipher block $C_m = \{c_i, c_{i+1}, \dots, c_{i+N-1}\}$ are added (modulo 2) with the corresponding bits of the error vector E_m as follows

$$c'_i = c_i \oplus e_i \quad (3-9)$$

to generate the cipher block $C'_m = \{c'_i, c'_{i+1}, \dots, c'_{i+N-1}\}$ at the receiver.

The ECB mode offers a relatively high level of security because of the high degree of confusion and diffusion achieved by encrypting a large block of message bits. However, this type of block encryption also results in error propagation. As stated earlier, if one or more bits of the received block are in error, the entire message block will be corrupted and about half the bits (32 bits) will be in error after decryption. This problem is analyzed in more detail in chapter 5. For now, it is sufficient to note that this type of propagation of errors can significantly deteriorate the performance of a channel link. It should be noted however, that transmitted cipher blocks are independent of each other in this mode of operation. Thus, a transmission error in a given block C'_m will affect only that block and will not propagate to adjacent blocks.

Another important disadvantage of the ECB mode is that a given message block M_m always yields the same cipher block C_m for as long as the key K does not change. This can provide important clues to an eavesdropper who is trying to break the code. But more importantly, it makes it possible for an interceptor to record a transmitted cipher block and replay it at a later time, by pretending to be the legitimate transmitter. This vulnerability of ECB to replay is due to the fact that ciphered blocks are encrypted independently of each other. This independence also makes it possible for an intruder to add, delete or replace entire cipher blocks, possibly without detection.

In addition, the requirement to encrypt blocks of exactly 64 bits can lead to significant message expansion in some cases. In order to make up blocks of that length, "padding" bits may have to be added to the end of some data blocks. If the messages transmitted are usually short, these extra padding bits can significantly increase the size of a message and consequently reduce the actual data throughput over the channel.

Furthermore, since both the transmitter and receiver must collect bits until they have a 64 bit block before using it, there is a processing delay introduced. This delay may be unacceptable in some real time applications. For example, for a digital voice system with a rate of 4800 bits per second (e.g. vocoder), the total time required to collect 64 bits at the transmitter and receiver before processing adds an additional delay of 26.67 milliseconds.

As can be seen, the ECB mode of operation has few advantages and many problems. It suffers from significant error propagation, processing delay and is vulnerable to block replay, insertion, deletion and replacement. Furthermore, message expansion can be a problem if short messages are frequently transmitted. Nevertheless, it provides a relatively high level of cryptographic security because of the confusion and diffusion qualities of the DES. The ECB mode of operation would probably be an acceptable choice in a low noise communication link where a fairly high level of security is required but where the possibility of replay or block modification is not possible⁷. However, the processing delay may be significant enough to preclude the use of ECB in a digital voice system.

⁷ i.e. if the message itself contains a time stamp or a numbering sequence. In such a case, replay, deletions, insertions or replacement of blocks would be immediately detected.

3.4.1.2 Cipher Block Chaining (CBC) mode

This mode of operation is illustrated in Figure 3.3. As for the ECB, the CBC mode operates on blocks of $N = 64$ bits. Both the receiver and transmitter start out with the same initial vector in the DES input register and the same key. At the transmitter, the first message block M_1 is added (modulo 2) with the initial vector and the resultant block is encrypted by the DES, yielding c_1 . This cipher block is transmitted and is also stored in the register. The next message block is then encrypted as before, but this time it is added with the cipher block c_1 before encryption. As a result, the encryption of an arbitrary message block M_m is given by:

$$C_m = E_K(M_m \oplus C_{m-1}) \quad (3-10)$$

Figure 3.3: CBC block cipher mode.

Thus, consecutive cipher blocks are chained together by the encryption operation.

As before, the transmitted block c_m is passed through the channel whose effect on the block is represented by:

$$C'_m = C_m \oplus E_m \quad (3-11)$$

At the receiver, the first cipher block is decrypted and is then added with the initial vector, to yield the message block. Following this, the received cipher block c'_1 is moved into the register and replaces the initial vector. The decryption of subsequent blocks is similar and can be represented as follows:

$$\begin{aligned}
D_K(C'_m) \oplus C'_{m-1} &= D_K(C_m \oplus E_m) \oplus (C_{m-1} \oplus E_{m-1}) \\
\text{if } E_m &= \vec{0} \text{ and } E_{m-1} = \vec{0} \text{ then} \\
D_K(C'_m) \oplus C'_{m-1} &= D_K(C_m) \oplus C_{m-1} \\
&= D_K(E_K(M_m \oplus C_{m-1})) \oplus C_{m-1} \\
&= M_m \oplus C_{m-1} \oplus C_{m-1} \\
D_K(C'_m) \oplus C'_{m-1} &= M_m
\end{aligned} \tag{3-12}$$

However, if the received block C'_m has v_N bits in error in it ($1 \leq v_N \leq N$, where $N = 64$ when the DES is used), the decrypted message M'_m will be corrupted (i.e. about $\frac{N}{2} = 32$ bits in error) since

$$D_K(C'_m) \neq M_m \oplus C_{m-1} \tag{3-13}$$

and therefore

$$D_K(C'_m) \oplus C'_{m-1} \neq M_m \tag{3-14}$$

Furthermore, even if the next block C'_{m+1} is received without errors ($E_{m+1} = \vec{0}$), it will have v_N bits in error in it after decryption since

$$\begin{aligned}
D_K(C'_{m+1}) \oplus C'_m &= (M_{m+1} \oplus C_m) \oplus C'_m \\
&= M_{m+1} \oplus C_m \oplus (C_m \oplus E_m) \\
&= M_{m+1} \oplus E_m
\end{aligned} \tag{3-15}$$

But it should be noted that the next cipher block will be decrypted correctly (assuming it is received without errors) because

$$D_K(C_{m+2}) \oplus C'_{m+1} = (M_{m+2} \oplus C_{m+1}) \oplus C'_{m+1} = M_{m+2} \tag{3-16}$$

where $E_{m+1} = E_{m+2} = \vec{0}$. Thus, it is immediately apparent that CBC mode is even worse than ECB as far as error propagation is concerned since v_N bits in error in a received block will result, on average, in $\frac{N}{2} + v_N$ bits in error after decryption (i.e. $32 + v_N$ versus 32 bits on average for ECB). This error propagation behaviour for CBC

is also examined in chapter 5. As the ECB mode, the CBC mode also experiences processing delay and message expansion.

Nevertheless, the CBC has some important advantages over the ECB mode. For instance, identical message blocks M_m will not yield identical cipher blocks because of the chaining operation. This not only removes important clues from a cryptanalyst but results in enhanced diffusion and confusion as a result of combining two blocks in the encryption process. Furthermore, the dependence of blocks on each other makes CBC mode immune to block playback, insertion, deletion or replacement.

In short, the CBC mode is an improvement on the ECB mode in that it is secure against playback and block tampering as well as generating different cipher blocks for similar input blocks. However, it has a slightly higher level of error propagation than ECB and does not resolve the problems of processing delay or message expansion. As for the ECB, the CBC mode would probably be useful only in a digital link requiring enhanced security and characterized by a very low noise level. Furthermore, the processing delay makes CBC mode more suitable for data transfer than for digital voice applications.

3.4.2 Stream ciphers

In a stream cipher, a message stream $M = \{m_1, m_{1+1}, \dots\}$ is broken down into individual words of length s bits $M_m = \{m_1, m_{1+1}, \dots, m_{1+s-1}\}$ ($1 \leq s \leq N$) which are encrypted individually with the corresponding key words $K_m = \{k_1, k_{1+1}, \dots, k_{1+s-1}\}$ of the key stream $K = \{k_1, k_{1+1}, \dots\}$ by a *modulo 2* addition (for binary ciphers) of the corresponding bits of K_m and M_m . In general, the value of s is a fraction of N and is often chosen to be equal to 1 bit only. The resultant cipher word

Figure 3.4: Generic stream cipher.

$C_m = \{c_1, c_{1,1}, \dots, c_{1,s-1}\}$ is then transmitted over the channel⁸. A generic stream cipher system is shown in Figure 3.4. Stream ciphers are further divided into two categories; those where the key stream is independent of the data stream are called synchronous ciphers while those where the key stream generated is dependent on the data stream are called self-

synchronous or autokey ciphers. For synchronous ciphers, it is necessary to have a separate system to maintain the transmitter and receiver in bit synchronization. However, self-synchronous ciphers do not require any bit synchronization; when an autokey cipher loses synchronization it automatically regains it after a fixed number of bits have been transmitted.

It is interesting to note that stream ciphers attempt to simulate the one-time pad cipher; just like the one-time pad, stream ciphers generate a key whose length is as long as the message being sent. However, unlike the one-time pad, the key stream of the ciphers discussed in this section use a pseudo random key instead of a purely random one. Furthermore, the key is not always independent of the data stream (e.g. CFB mode). As a result, the cryptographic strength of practical stream ciphers depends on the complexity of the algorithm used to generate the pseudo random key stream.

The main advantage of stream ciphers over block ciphers is that there is less delay in the encryption of the data since words can be much shorter than n bits, as opposed to block ciphers which

⁸ Note that the key in this case is not the 64 bit key used by the DES but a key stream which is generated indirectly by the DES.

must process N bits at a time. As a result, stream ciphers are more suitable for digital voice systems or any other application where long delays cannot be tolerated. Furthermore, the specific implementations discussed in this chapter will be shown to have lower error propagation than the block cipher modes seen earlier⁹. Thus, stream ciphers are more suitable for noisy communication links where the BER is high.

3.4.2.1 Output Feedback (OFB) mode

One possible implementation of the OFB mode is shown in Figure 3.5. In this setup, both the transmitter and receiver use the same key for the DES and set the $N = 64$ bits input register (input to the DES) to the same initial vector. That input vector is then encrypted by the DES and s bits of the resultant cipher block are used as key word K_1 to encipher the data stream. M_1 and K_1 are added modulo 2 and the resultant cipher word C_1 is transmitted. Note that the 64 bit cipher block generated by the DES is fed back into the input register for the next round of encryption. At the receiver, the initial vector is also encrypted by the DES and K_1 is added to the received cipher word C'_1 to recover the message word M'_1 . Note again that $C'_m = C_m \oplus E_m$ where E_m is the s bits error vector added by the channel. The encryption and decryption operations are given by:

Figure 3.5: OFB stream cipher mode.

⁹ This may or may not be true for CFB mode, depending on the word length s in bits.

$$\begin{aligned}
 C_m &= M_m \oplus K_m \\
 M'_m &= C'_m \oplus K_m
 \end{aligned}
 \tag{3-17}$$

where $C'_m = C_m \oplus E_m$. Note that in this mode of operation, the DES is only used to generate a long pseudo random sequence of bits $k_1, k_2, \dots$ which is the key stream used to encrypt the data stream. Although the key stream is as long as the data stream¹⁰, the cryptographic strength of this mode of operation is not really higher than that of block cipher modes which use the DES directly to encrypt the data. For the OFB mode, the security of the system depends on the secrecy of the 64 bit DES key¹¹. In the implementation of Figure 3.5, the entire cipher block generated by the DES is fed back to its input for the next round of encryption.

The main advantage of the OFB mode is that it introduces no error propagation. As long as both the receiver and transmitter are in bit synchronization, they will always have the correct key stream since it is not dependent on the data stream and is generated locally. Thus, a corrupted cipher bit c'_i will result in a message bit m'_i being deciphered incorrectly but the error will have no effect on the cipher bits received subsequently. This is a significant advantage, especially in noisy links such as mobile communication channels.

Stream ciphers introduce lower encryption delays (when $s < N$) since they operate on shorter words, in contrast to block ciphers

¹⁰ The maximum period of the key stream $k_1, k_2, \dots$ is limited to 2^{56} but this is more than sufficient for most applications.

¹¹ For OFB mode, the initial vector can provide additional security, if it is kept secret and changed regularly, like the DES key itself. That is because the DES output sequence is completely determined by the key and the initial vector used. A cryptanalyst would have to know both of them to generate the key stream sequence.

which need n bits before encryption (or decryption) can start. As a result, the OFB mode is suitable for digital voice and other applications which require low delays. Furthermore, the fact that the OFB mode can operate on short words means that it will produce less message expansion than block ciphers do, or none at all when $s = 1$. Thus, it is well suited for systems where messages are generally short. Finally, since the key stream is independent of the data stream, repeated data sequences in the data stream do not result in similar cipher bit sequences since the key stream itself is always changing.

But because the key stream is independent of the data stream, the OFB mode is a synchronous cipher and therefore a separate system is required to maintain bit synchronization between the transmitter and receiver. For instance, if the transmitter is enciphering M_m with $K_m = \{k_{j,1}, k_{j,2}, \dots, k_{j,s-1}\}$ to generate word C_m but the receiver attempts to decipher C'_m with $K_m = \{k_{j,1}, k_{j,2}, \dots, k_{j,s}\}$ because it lost synchronization, the deciphered word M'_m and all subsequent words will be wrong (with a 0.5 probability) until bit synchronization is regained. The problem of maintaining synchronization is a complex one, especially in a difficult channel such as a mobile communications link. In general, maintaining synchronization involves the regular transmission of timing sequences over the channel. This results in a reduction of the information throughput since the data stream must be interrupted regularly to allow the transmission of the synchronization sequence. If synchronization is maintained through a different channel than the data channel, the bandwidth allocated to each link is increased because of the additional channel. In either case, the bandwidth utilization efficiency (data bits transmitted per Hertz per second) is reduced.

Another disadvantage of the OFB mode is that each use of the DES transformation encrypts only s data bits, in contrast with

block ciphers where one round of DES encryption encrypts N bits. This may cause delay problems in some instances, but this is rarely the case since DES chips can encrypt up to 100 Mbits/s. If only one bit is encrypted per round instead of 64, the throughput of the DES is 312,500 bits per second, which is substantially higher than the data throughput rate of most mobile communication channels. Thus, this disadvantage can be ignored in most practical situations.

A potential security problem of the OFB mode of operation is that the independence of the key and data stream make it possible for an intruder to replace individual bits in the stream without being detected. For example, an eavesdropper close to the receiving station can "listen" in to the channel and at appropriate times transmit false cipher bits to the legitimate receiver. If the intruder's false transmission can mask the real one, the receiver will accept the false cipher bit. Since errors do not propagate to other bits in the stream, there is a significant probability that the false data may go undetected.

Since the OFB cipher mode encrypts by a simple substitution algorithm, there is no confusion or diffusion in the strict sense. The key stream is independent of the data stream and thus there is no direct relationship between the data and the key. Similarly, independence of consecutive cipher bits implies that there is no diffusion of the data's statistics. The absence of confusion and diffusion in the strict sense does not necessarily imply that the OFB mode is significantly weaker than block modes since the generated key stream appears random enough. Nevertheless, the independence of adjacent cipher bits does give a cryptanalyst an advantage since he can analyze bits individually without having to worry about their inter-dependence.

Another important restriction of OFB mode is that once a part of the key stream has been used to encrypt, it cannot be used again to encrypt different data without seriously compromising the

security of the system since the plaintext can be recovered easily in such a case [21, pp.144]. This means that a key can only be used once.

In conclusion, the OFB mode of operation benefits from considerable advantages such as the absence of error propagation, less processing delay and message expansion (depending on the word length s) and the concealment of repeated data sequences (which produce different cipher sequences). These make the OFB suitable for noisy channels, digital voice or other low-delay applications and for systems where short messages are common. However, the independence of the data and key stream result in the requirement for a separate system to keep the transmitter and receiver in bit synchronization. Furthermore, the OFB is vulnerable to bit replacement and provides less cryptographic strength than block ciphers. Thus, OFB mode may not be suitable for communication links where a relatively high level of security is required.

3.4.2.2 Cipher Feedback (CFB) mode

This mode of operation is illustrated in Figure 3.6. In CFB

Figure 3.6: CFB stream cipher mode.

mode, the stream of message bits $M = \{m_1, m_2, \dots, m_l, \dots\}$ is partitioned into words $M_m = \{m_i, m_{i+1}, \dots, m_{i+s-1}\}$ of s bits where $1 \leq s \leq N$. For CFB mode, s is usually selected so that it can be divided into N without remainder (i.e. if $N = 64$, then $s \in \{1, 2, 4, 8, 16, 32, 64\}$). This is done to ensure that the memory buffer contains a non-fractional number of encrypted

words¹². For CFB mode, the words K_m and M_m are enciphered individually (like OFB) but not independently of each other (unlike OFB). As was the case for OFB, the DES is used to generate a pseudo-random key stream which is as long as the data stream. At the start of transmission, both the transmitter and receiver use the same DES key and set the 64 bit DES input register to the same initial vector¹³. The transmitter then generates a 64 bit output block from the DES and uses s bits (arbitrarily chosen from that output block) as key word K_1 . The cipher word C_1 is then generated as $C_1 = M_1 \oplus K_1$ and is transmitted over the channel. Following this, the s -bit cipher word C_1 is shifted into the least significant s bit positions of the 64 bit input register and all other bits are shifted left by s positions (the s most significant bits are discarded). The process is repeated for subsequent data words. In this mode, the key word K_m is a function of the $\frac{N}{s} = \frac{64}{s}$ previously generated cipher words (the N previous cipher bits) as follows:

$$\begin{aligned} K_m &= f(E_K(C_{m-\frac{N}{s}}, C_{m-\frac{N}{s}+1}, \dots, C_{m-2}, C_{m-1})) \\ &= f(E_K(C_{1-N}, C_{1-N+1}, \dots, C_{1-2}, C_{1-1})) \end{aligned} \quad (3-18)$$

where $N = 64$ for the setup illustrated in Figure 3.6. The function $f(\cdot)$ in the above equation is used to emphasize the fact that only s of the N bits generated by the encryption operation are used to create the key word. The value of N determines the amount of memory of the CFB mode and has an impact on the degree of error propagation. If N is smaller than 64, the input to the DES buffer

¹² This results in a lower degree of error propagation.

¹³ For CFB mode, the initial vector does not provide any additional security even if it is kept secret. That is because the initial vector is shifted out of the DES input buffer after N/s iterations and thus, subsequent cipher words are independent of its value [45, pp. 465].

would consist of the N bits from the memory register plus $64 - N$ additional bits which would be preset at both the transmitter and receiver (the DES input buffer must always be 64 bits long).

The receiver encrypts the initial vector to generate key word K_1 , which is then added to the received cipher word C'_1 to generate M'_1 . The cipher word C'_1 is then shifted into the DES input register for the decryption of the next cipher word to be received.

A cipher word C'_m received in error will not be decoded properly since $C'_m \oplus K'_m \neq C_m \oplus K_m$. Furthermore, the corrupted word is then shifted into the DES input register and will remain there for the next $\frac{N}{s}$ word periods. As a result, the receiver's DES input vector will be different from the transmitter's and the next $\frac{N}{s}$ DES cipher output blocks will also be different. Thus, the received key words $K'_{m+1} = f(E_K(C'_{m-\frac{N}{s}+1}, C'_{m-\frac{N}{s}+2}, \dots, C'_m))$ to $K'_{m+\frac{N}{s}} = f(E_K(C'_m, C'_{m+1}, \dots, C'_{m+\frac{N}{s}-1}))$ inclusive will be corrupted. On average, about half of the bits of each key word will be in error because of the diffusion and confusion characteristics of the DES. Assuming that no other cipher word is received in error during the next $\frac{N}{s}$ word periods, it is easy to see that one word received in error will, on average, result in $\frac{N}{s} + 1$ words being deciphered incorrectly. For example, if the implementation of Figure 3.6 is used with $s = 1$, one bit received in error would result in about 33 message bits being in error after decryption.

Based on this result, it would be tempting to conclude that the performance of the CFB mode of operation is identical to that of block cipher mode CBC, as far as error propagation is concerned (when N is the same value for both modes). However, this is not the

case, as will be shown in chapter 5. As a general rule, the lower the word length s is, the lower the error propagation is for CFB mode, compared to CBC mode¹⁴. Furthermore, for the CFB mode, the severity of error propagation is directly dependent on the length N of the memory register. For instance, if N was 32 instead of 64, the amount of error propagation would be significantly lower. For the implementations of ECB and CBC shown earlier, N is fixed at 64 bits because this is the output length of the encrypted word c_m regardless of the length of the data word M_m .

One advantage of making the key stream dependent of the cipher bits is that it makes the CFB system more secure against undetected bit replacements, in contrast to OFB mode. For the CFB system, if an erroneous bit is inserted in the cipher stream, the resultant error propagation will immediately signal the intrusion. Furthermore, the dependence of adjacent cipher bits on each other results in a higher degree of confusion and diffusion for the CFB mode than was the case for the OFB mode. This makes it harder for a cryptanalyst to break the code. In fact, the longer the memory of the CFB system is (i.e. the longer the value of N is), the higher is the degree of confusion. However, as N increases, so does the amount of error propagation and a compromise must be reached between the two.

Another advantage of the dependence between the key stream and the data stream is that it makes the CFB mode a self-synchronous cipher. As a result, it does not require a separate bit synchronization system. This is easy to see since a bit skipped or erroneously added by the receiver is equivalent to a cipher bit being received in error, as described earlier. When $\frac{N}{s}$ word periods have elapsed, the extra (or missing) bit is no longer part of the DES input register and the receiver automatically falls back into

¹⁴ However, the difference is marginal, even when $s = 1$ for CFB mode.

synchronization with the transmitter. This is a significant advantage over OFB mode since it was seen earlier that synchronization systems are complex, expensive and reduce the channel bandwidth utilization efficiency.

In addition, the fact that the key stream is indirectly dependent on the data stream (through the cipher stream) makes it impossible for the same key stream sequence to be used to encrypt two or more different data sequences, thereby compromising the security of the system (as was the case for OFB). It does, however, remain essential to change the DES key frequently to protect the system's security.

Finally, another advantage of the CFB mode is that it can operate on shorter words than ECB and CBC block cipher modes and therefore does not introduce as much additional processing delay or message expansion. However, since the DES is used to encrypt only s data bits at a time, as opposed to $N = 64$ bits for block ciphers, CFB is less efficient than the latter in that respect, but this rarely results in practical problems.

3.5 New trends in cryptography

In the interest of completeness, this short section provides a few references which will give some indication to the reader of the new developments and areas of interest in the field of cryptography.

In order to improve on the DES and extend its use in the 90's and beyond, many enhancements have been proposed. A list of all such proposals would be excessively long and thus only a few, recent ones are mentioned here. Brown et al. [11] proposed new design criteria to enhance the effectiveness of the permutation boxes for the DES. Connell [19] and Adams et al. [1] proposed changes to the crucial substitution boxes of the DES to improve the

diffusion characteristics of the algorithm. Carroll et al. [12] and Connell [19] are two among many who have proposed a modified DES algorithm using a larger key.

Many researchers have come up with entirely new encryption algorithms to replace the DES altogether. Among these, the REDOC-II system [20] and the FEAL-n family [48,63] have received a lot of attention. The final verdict is not out on these new proposals but independent researchers have found problems with both the REDOC-II [8] and the FEAL-n family [32,33,66]. Finally, Lai [38] proposed a new standard for data encryption which incorporates a large key (128 bits) with mixing functions to confuse and diffuse a 64 bits input block into a 64 bits output cipher block.

The cryptographic security of self-synchronizing stream ciphers continues to be studied in detail and new functions are being proposed to replace the simple input register apparatus of the CFB mode shown in Figure 3.6. Both Maurer [45] and Anderson [3] provide interesting ideas on that subject. Rueppel [57] gives a thorough analysis of the theoretical security of stream ciphers in general. Also on the subject of stream ciphers, there is a growing interest in the use of chaos theory and fractal functions as substitutes to the DES in order to generate pseudo random key streams. Matthews [44] and Carroll et al. [13] describe such systems. However, security flaws have been pointed out by Wheeler [69,70] and the suitability of such functions in cryptographic applications has not been firmly established yet. Finally, the techniques of combining data compression and encryption in one step (Boyd [10]) or combining encryption and error correction coding in one step (Hwang [35]) are generating a lot of interest in the cryptographic community.

3.6 Conclusion

In this chapter, basic concepts of cryptography were described

and the requirements of a strong cryptographic system were explained in some detail. It was made apparent that in real encryption systems the key size is necessarily limited, thereby making all encryption systems theoretically unsafe. Nevertheless, by using Shannon's concepts of confusion, diffusion and mixing functions, it was shown that it is possible to make the relationship between the plaintext and the corresponding cipher complex enough to make the task of breaking the system prohibitively long and expensive. It was also made clear that, regardless of the security level of the encryption system, the key must always be long enough to make an exhaustive search of the key space unfeasible.

Subsequently, it was argued that the Data Encryption Standard meets the requirements of a good cryptographic system and is therefore suitable for digital encryption applications where low to medium security is required. The four modes of operation of the DES in block and stream cipher systems were then analyzed in detail. It was shown that the CBC mode has the worst error propagation characteristic, followed by the CFB, ECB and OFB modes in that order¹⁵. Block cipher modes were seen to suffer from processing delay but possess better cryptographic strength than stream ciphers (which introduce less delay and message expansion, when the word length s is smaller than N). The CBC mode was shown to have the best diffusion and confusion qualities, followed by the CFB, ECB and OFB modes in that order. It was made clear that OFB mode requires accurate bit synchronization while block cipher modes need only block synchronization and CFB requires neither. It was also made apparent that CBC mode is the most secure against bit or block insertions, deletions or modifications. The CFB mode is second best in that respect, followed by ECB which provides good protection

¹⁵ This assumes that the value of N for CFB mode is identical to that of CBC and ECB modes. If the value of N is smaller for CFB mode, it could outperform both CBC and ECB modes.

against bit tampering but is vulnerable to block replay. The OFB mode was shown to be vulnerable to bit tampering.

Finally, the last section of this chapter provided a short description of new trends and directions in the area of cryptographic research.

Chapter 4

Error control coding and interleaving

4.1 Introduction

In this chapter, error control techniques are examined. More specifically, the relationship between the code rate, minimum distance and error correcting capability of linear block codes is discussed. This is followed by a description of the basic properties of Bose-Chaudhuri-Hocquenghem (BCH) codes. The encoding and decoding procedures for binary BCH codes are then reviewed. Subsequently, the technique of bit interleaving is examined and its advantages and disadvantages are presented. This chapter ends with a description of the equations required to calculate $P(m,n)$ (probability of m bits in error in a block of n bits) for memoryless and fading channels.

4.2 Linear block codes

Linear block codes provide an easy and efficient way of introducing a controlled amount of redundancy to a stream of data bits. For all such codes, a block of k data bits is mapped into a block of n data bits ($n > k$). The additional $n - k$ bits are sometimes referred to as parity bits. These redundant bits are a function of the k data bits themselves and allow the detection and (or) correction of bit errors introduced by the channel at the receiver. Such block codes are commonly labelled as (n,k,t) codes where t is the error correcting capability of the code; it is the maximum number of bit errors which can be corrected in a codeword of n bits. If the number of channel bit errors introduced v is such that $v \leq t$, the channel errors will be corrected and the k data bits will be recovered error-free at the receiver. If $v = t+1$ (for example),

a decoding failure will occur and (at most) $2t+1 = t+v$ of the k data bits will be decoded incorrectly at the receiver [54, pp. 405].

The minimum distance $d_{\min}$ of a code is the minimum number of bits by which any two codewords differ. For linear block codes, $d_{\min} = w_{\min}$, which is the minimum weight of the code (i.e. the number of '1's in the codeword with the smallest number of '1's, excluding the all-zero codeword). The error correcting capability of a code is directly related to the minimum distance; the larger the minimum distance is, the higher the error correcting capability of the code is. For linear block codes [43, pp. 65-67]

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor \quad (4-1)$$

Furthermore, all error patterns of $d_{\min} - 1$ bits or less (within a block of n bits) can be detected at the receiver but not corrected¹. For error patterns which can be detected but not corrected, the receiver has three options:

- a) ignore the codeword entirely (i.e. erasure),
- b) request a retransmission from the transmitter, or
- c) attempt to decode the codeword.

The second option is often referred to as automatic repeat request (ARQ) and requires a feedback channel from the receiver to the transmitter. In order to limit the scope of this thesis, ARQ is not considered. In all our simulations, the receiver implements the third option. As a result, the receiver will attempt to correct all codewords regardless of the number of bit errors they contain. As mentioned earlier, this will result in decoding failures (when

¹ It is possible to do error correction and detection simultaneously.

$v > t$) which will cause (at most) $t+v$ of the k data bits of each such codeword being in error at the receiver. When the decoder is unable to decode the received vector, the simulation algorithm selects the all-zero vector by default. In such a case, about half of the k data bits will be in error (on average).

Another important parameter for linear block codes is the code rate R , defined as:

$$R = \frac{k}{n} \quad (4-2)$$

The code rate indicates the 'real' throughput of information in the channel, in bits per symbol. As the code rate decreases, the throughput of information decreases. However, a decrease in R (for a fixed n) implies an increase in the number of parity bits ($n - k$) which usually results in an increase in the minimum distance $d_{\min}$. Therefore, as R decreases, the error correcting capability t of the code increases. A compromise between these factors has to be made; the code rate must be held as high as possible in order to utilize the channel bandwidth efficiently but it must be low enough to allow the code to correct most channel errors.

4.3 Binary BCH codes

The Bose-Chaudhuri-Hocquenghem codes [43, pp. 141-183] form a large and powerful class of linear, cyclic² block codes which includes the Reed-Solomon codes. In order to limit the scope of this thesis, we restrict our attention to binary BCH codes, which are designed to operate in memoryless channels. Although these

² A code is said to be cyclic when any codeword rotated left or right by any amount is also a codeword. For a rotation to the left, the most significant bit is moved to the least significant bit position (vice versa for a rotation to the right).

codes usually perform poorly in channels with memory, the use of interleaving (discussed later) can enhance their performance significantly in such links.

BCH codes obey the following set of conditions:

$$\begin{aligned} n &= 2^m - 1 \\ n - k &\leq mt \\ d_{\min} &\geq 2t + 1 \\ m &\geq 3 \end{aligned} \tag{4-3}$$

These provide the engineer with a wide variety of codes of different lengths, code rates and error correcting capabilities [43, appendix C].

4.3.1 Encoding procedure for binary BCH codes

An (n, k, t) binary BCH code can be described by a generator polynomial which has the form

$$g(X) = 1 + g_1X + g_2X^2 + \dots + g_{n-k-1}X^{n-k-1} + X^{n-k} \tag{4-4}$$

where $g_i \in \{0, 1\}$, $1 \leq i \leq n-k-1$.

$g(X)$ is defined in terms of the roots from the Galois Field $GF(2^m)$ and is the lowest degree polynomial over $GF(2)$ which has $\alpha, \alpha^2, \dots, \alpha^{2t}$ as its roots (i.e. $g(\alpha^i) = 0$, $1 \leq i \leq 2t$). α is a primitive element of $GF(2^m)$. The generator polynomial can be calculated from [43, pp. 142]:

$$g(X) = LCM\{\phi_1(X), \phi_3(X), \dots, \phi_{2t-1}(X)\} \tag{4-5}$$

where LCM stands for 'lowest common multiple' and $\phi_1(X)$ is the minimal polynomial of α , [43, pp. 35-38]. For example, the (15,5,3) code has this generator polynomial: $g(X) = 1 + X + X^2 + X^4 + X^5 + X^8 + X^{10}$

Encoding is performed by multiplying the message polynomial $u(X)$ by the generator polynomial $g(X)$, yielding the codeword polynomial $v(X)$, where

$$\begin{aligned} u(X) &= u_0 + u_1X + u_2X^2 + \dots + u_{k-1}X^{k-1} \\ u_i &\in [0,1], \quad 0 \leq i \leq k-1 \\ v(X) &= u(X) \times g(X) \\ &= v_0 + v_1X + v_2X^2 + \dots + v_{n-1}X^{n-1} \\ v_i &\in [0,1], \quad 0 \leq i \leq n-1 \end{aligned} \tag{4-6}$$

For example, data block 01101 corresponds to the message polynomial $u(X) = X + X^2 + X^4$. The codeword polynomial is then $v(X) = X + X^6 + X^7 + X^8 + X^{10} + X^{11} + X^{14}$ and the transmitted codeword is 010000111011001.

4.3.2 Decoding procedure for binary BCH codes

The codeword vector at the receiver is given by

$$r(X) = v(X) + e(X) \tag{4-7}$$

where

$$\begin{aligned} e(X) &= e_0 + e_1X + e_2X^2 + \dots + e_{n-1}X^{n-1} \\ e_i &\in [0,1], \quad 0 \leq i \leq n-1 \end{aligned} \tag{4-8}$$

is the error vector, representing the bit errors introduced by the

channel. Decoding involves the following steps:

1) Calculate the syndrome vector $\vec{S} = (S_1, S_2, \dots, S_{2t})$, where

$$S_i = r(\alpha^i) = r_0 + r_1 \alpha^i + \dots + r_{n-1} \alpha^{(n-1)i}$$

2) If $\vec{S} \neq \vec{0}$, proceed as follows:

a) Compute the error location polynomial $\sigma(X) = \sigma_0 + \sigma_1 X + \dots + \sigma_v X^v$ where v is the number of bit errors introduced by the channel,

b) find the v roots of $\sigma(X)$ (these are $\beta_1, \beta_2, \dots, \beta_v$),

c) calculate the inverse of the roots of $\sigma(X)$.

d) use $\beta_1^{-1}, \beta_2^{-1}, \dots, \beta_v^{-1}$ to construct the error polynomial $e(X) = e_0 + e_1 X + \dots + e_{n-1} X^{n-1}$,

e) correct the bit errors by adding $e(X)$ to $r(X)$, and

3) perform the division $u'(X) = \frac{r(X)}{g(X)}$ where $u'(X)$ is the receiver's estimate of the transmitted message $u(X)$.

Note that if $\vec{S} = \vec{0}$, the received vector is assumed to be error-free since all the elements S_i are equal zero only if $r(X)$ is a valid codeword from the code space. If one or more of the S_i elements are non-zero, then $r(X)$ is not a valid codeword and the bit(s) in error must be located in order to correct the vector.

In order to do this, the simplified Berlekamp-Massey algorithm [43, pp. 158-159] is used. The technique involves $t+1$ iterations which use the elements of $\vec{S}$ to determine the coefficients of the error location polynomial. The simplified algorithm can only be

used for binary BCH codes but it is faster than the general algorithm which involves $2t+1$ iterations. The steps involved in the algorithm are fairly involved and are described in detail in [43].

When $\sigma(x)$ is known, we must calculate $\sigma(\alpha^i)$, $1 \leq i \leq n-1$ to determine the roots $\beta_1, \beta_2, \dots, \beta_v$ of this polynomial. When the roots have been found, their inverse has to be determined. For example, if $\beta_s = \alpha^i$, then [43, pp. 31]

$$\beta_s^{-1} = \alpha^{2^n-i-1} \quad (4-9)$$

so that $\beta_s \beta_s^{-1} = \alpha^0 = 1$.

These inverses are referred to as the error location numbers. For example, if three bit errors occur in a block of n bits and $\beta_1^{-1} = \alpha^i$, $\beta_2^{-1} = \alpha^j$, $\beta_3^{-1} = \alpha^k$ where $i, j, k \in [0, n-1]$, $i \neq j \neq k$, the error polynomial becomes $e(x) = x^i + x^j + x^k$. By adding $e(x)$ to $r(x)$, these bits are corrected and the received vector can then be divided by the generator polynomial $g(x)$ to recover $u'(x)$.

It is important to note that if the number of errors introduced by the channel is $v > t$, some of the coefficients of the error location polynomial will be incorrect and a decoding failure will occur. The resultant message vector $u'(x)$ will differ from $u(x)$ in at most $t+v$ bit positions [54, pp. 405] when the decoder succeeds in decoding the vector. As a result of this, BCH codes may lead to some error propagation (i.e. where the decoded data vector contains more bit errors than were introduced in the codeword by the channel). Note that if the decoder fails to locate a plausible codeword, the simulation algorithm used in this work will select the all-zero data word by default. This will then result in about half of the data bits being erroneously decoded (on average).

4.4 Bit interleaving

The process of bit interleaving [43, pp. 271] transforms an (n, k, t) codeword into an (In, Ik, It) codeword, where I is the interleaving degree in bits.

Interleaving is accomplished by arranging I codewords (n, k, t) into I rows of a rectangular array and transmitting the columns from left to right and the bits from top to bottom in each column. For example, consider the three codewords given below and an interleaving degree $I = 3$;

$$\vec{C}_1 = \{c_{1,1}, c_{1,2}, \dots, c_{1,n}\}$$

$$\vec{C}_2 = \{c_{2,1}, c_{2,2}, \dots, c_{2,n}\}$$

$$\vec{C}_3 = \{c_{3,1}, c_{3,2}, \dots, c_{3,n}\}$$

The transmitted sequence will take this form

$$\vec{C} = \{c_{1,1}, c_{2,1}, c_{3,1}, c_{1,2}, c_{2,2}, c_{3,2}, \dots, c_{1,n}, c_{2,n}, c_{3,n}\}$$

and the received sequence at the receiver will have this form

$$\begin{aligned} \vec{C}' = \{ & c_{1,1} + e_1, c_{2,1} + e_2, c_{3,1} + e_3, \\ & c_{1,2} + e_4, c_{2,2} + e_5, c_{3,2} + e_6, \dots, \\ & c_{1,n} + e_{(n-1)I+1}, c_{2,n} + e_{(n-1)I+2}, c_{3,n} + e_{nI} \} \end{aligned}$$

where $\vec{e} = \{e_1, e_2, e_3, \dots, e_{nI}\}$ is the error vector representing the channel bit errors introduced for each consecutive bit interval. Hence, an error burst of length $v \leq I$ will affect at most one bit of

each codeword. If $t \geq 1$ for the code used, the error burst will be corrected.

Note that the equivalent channel error sequence, as perceived by the interleaved codewords, has the form

$$\tilde{e}' = \{e_1, e_4, \dots, e_{(n-1)I+1}, e_2, e_5, \dots, e_{(n-1)I+2}, e_3, e_6, \dots, e_{nI}\}$$

where consecutive bit errors in the channel appear separated by n bit intervals. Hence, interleaving can be viewed as a process where channel memory is being reduced. However, interleaving is a suboptimal technique [37, pp. 741] because it does not make use the channel memory, as opposed to ARQ which adapts itself to the channel memory. In ARQ, data is transferred using a high code rate (close to 1) whose only purpose is to detect errors. When the channel is quiet, data is transferred rapidly over the link. When the channel is noisy, the receiver detects the occurrence of errors and requests a retransmission of the corrupted codewords.

As a general rule, as the interleaving degree increases, the probability that the error clusters will be distributed uniformly also increases. However, interleaving introduces a systematic processing delay since the receiver must wait for the last bit of a codeword to arrive before decoding can begin. There is also a delay at the transmitter, in filling up the rectangular array of codewords. Hence, delay increases as the interleaving degree increases. This is an important consideration in voice applications where the maximum total delay that can be tolerated in commercial applications is about 30 ms. Furthermore, because error clusters in a fading channel often appear at roughly equal intervals, some values of I may keep errors clustered after interleaving and may even increase the concentration of errors (i.e. increase channel memory). The probability of this occurring usually decreases as I increases but this increases the processing delay.

4.5 Probability $P(m,n)$ of m errors in a block of n bits

In order to assess the performance of forward error correction codes over a channel, the probability of a decoding failure, (i.e. probability of having $t+1$ or more errors in a block of n bits) must be determined. If we define a decoding failure as the event where the decoded data vector at the receiver differs from the data vector at the transmitter, we can then formulate the following argument: for an (n,k,t) error correcting code, the probability of a decoding failure p_c is given by

$$p_c = \sum_{m=t+1}^n P(m,n) = 1 - \sum_{m=0}^t P(m,n) \quad (4-10)$$

where $P(m,n)$ is defined as the probability of having exactly m bits in error in a block of n bits. In this section, we provide the expressions for $P(m,n)$ for a memoryless binary symmetric channel, as well as for fading channels represented by the Gilbert and Fritchman models.

4.5.1 $P(m,n)$ for a binary symmetric channel

When errors occur randomly in a channel and are distributed uniformly over time, $P(m,n)$ can be shown to have this form [41, pp. 91]:

$$P(m,n) = \frac{n!}{(n-m)! m!} p^m (1-p)^{n-m} \quad (4-11)$$

where p is the channel bit error rate (BER).

4.5.2 $P(m, n)$ for the Gilbert's channel model

For a fading channel represented by a Gilbert model, the determination of $P(m, n)$ is more involved. Elliott [25, pp. 1988] developed the equations which define $P(m, n)$ for an Elliott channel model. As mentioned in chapter 2, the only difference between the Elliott model and the Gilbert model is that the former has a non-zero error probability in the good ('G') state. The equation for $P(m, n)$ is given by

$$P(m, n) = \frac{P_{bg}}{P_{bg} + P_{gb}} G(m, n) + \frac{P_{gb}}{P_{bg} + P_{gb}} B(m, n) \quad (4-12)$$

where

$$G(m, n) = G(m, n-1) P_{gg}(1 - P_{eg}) + B(m, n-1) P_{gb}(1 - P_{eg}) + G(m-1, n-1) P_{gg}P_{eg} + B(m-1, n-1) P_{gb}P_{eg} \quad (4-13)$$

and

$$B(m, n) = B(m, n-1) P_{bb}(1 - P_{eb}) + G(m, n-1) P_{bg}(1 - P_{eb}) + B(m-1, n-1) P_{bb}P_{eb} + G(m-1, n-1) P_{bg}P_{eb} \quad (4-14)$$

with the following conditions

$$\begin{aligned} G(0, 1) &= 1 - P_{eg}, & G(1, 1) &= P_{eg} \\ B(0, 1) &= 1 - P_{eb}, & B(1, 1) &= P_{eb} \\ G(m, n) &= B(m, n) = 0, & \text{if } m < 0, n < 0, m > n \end{aligned} \quad (4-15)$$

In the above equations, $G(m, n)$ is the probability of exactly m bit

errors in a block of n bits, given that the model is initially in the good state while $B(m,n)$ is the probability of m bits in error in a block of n bits when the model is initially in the bad state. As before, P_{bg} is the probability of transition from the bad to the good state and P_{bb} is the probability of transition from the bad to the bad state (and similarly for P_{gb} and P_{gg}). P_{eb} is the error probability in the bad state while P_{eg} is the error probability in the good state. For the Gilbert model, $P_{eg} = 0.0$.

4.5.3 $P(m,n)$ for the Fritchman's channel model

For a Fritchman channel model, the recursive equations leading to the equation for $P(m,n)$ can be found in [23] and are given below:

$$P(m,n) = \sum_{i=1}^n P_i f_i(m,n) \quad (4-16)$$

where

$$f_i(m,n) = \left(\sum_{j=1}^k P_{ij} f_j(m,n-1) \right) + \sum_{j=k+1}^n P_{ij} f_j(m-1,n-1) \quad (4-17)$$

with these conditions

$$\begin{aligned} f_j(0,0) &= 1 \\ f_j(m,n) &= 0, \text{ if } n < 0, m < 0, m > n \end{aligned} \quad (4-18)$$

$f_i(m,n)$ is the probability of exactly m transitions to the error states in n steps, given that the chain starts in state s_i . P_{ij} is the transition probability from state s_i to state s_j and P_i is the steady state probability of being in state s_i . It is assumed

that there are k error-free states and $n - k$ error states. For a simplified Fritchman model with one error state and no transitions allowed between the error-free states, the equations simplify to this form:

$$\begin{aligned} f_i(m, n) &= P_{ii} f_i(m, n-1) + P_{iN} f_N(m-1, n-1), \quad 1 \leq i \leq N-1 \\ f_N(m, n) &= \left(\sum_{i=1}^{N-1} P_{Ni} f_i(m, n-1) \right) + P_{NN} f_N(m-1, n-1) \end{aligned} \quad (4-19)$$

4.5.4 Comparison between analytical and experimental values of $P(m, n)$

To demonstrate the validity of the equations derived in the earlier sections, the probability $P(m, n)$ was calculated from channel files generated by the three models used in this work and compared to the values predicted by theory. The model parameters used are the same ones as those of the sample channels of chapter 2. The results are shown in Figure 4.1. The curves represent the theoretical $P(m, n)$ while the marked points indicate the $P(m, n)$ calculated from the corresponding channel file (based on 500,000 bits). It can be seen that there is excellent agreement between the theoretical and experimental values. Note that all three channels have roughly the same bit error rate³. Hence, the three curves for $P(m, n)$ can be compared

Figure 4.1: Theoretical $P(m, n)$ versus experimental values.

³ $BER_{BSC} = 0.0256$, $BER_{GIL} = 0.0253$, $BER_{FRI} = 0.0264$

directly and the impact of memory on the distribution of bit errors within a block of n bits is clearly shown.

4.6 Conclusion

In this chapter, we described the basic properties of linear block codes. It was shown that the error correcting capability of any such code is directly related to the minimum distance. It was also shown that, as the code rate decreases, the minimum distance increases and the error correcting capability of the code increases. Following this, the class of binary BCH codes was presented. The procedure to determine the generator polynomial for an (n, k, t) code was explained. The encoding and decoding process for these codes was then described. Subsequently, the advantages and disadvantages of using bit interleaving was examined briefly. It was further shown that interleaving is a suboptimal error control technique but which can provide a significant improvement in the performance of binary BCH codes over fading channels. The last section of this chapter described the equations required to compute the probability $P(m, n)$ of obtaining m errors within a block of n bits for a memoryless BSC and for fading channels represented by either the Gilbert or Fritchman models. The validity of the theoretical values of $P(m, n)$ was demonstrated and the importance of that parameter in determining the probability of a decoding failure p_e was also explained.

Chapter 5

Analysis and simulation results

5.1 Introduction

This chapter is divided into two main parts. In part one, equations are derived from theory to predict the bit error rate after decryption (*DBER*) for different cipher modes over binary symmetric channels with and without FEC coding. Part two provides an analysis of the simulation results and is further divided into four main sections:

- a) Simulation results over memoryless BSC,
- b) simulation results over fading channels generated by the Gilbert model,
- c) simulation results over fading channels generated by the Fritchman model, and
- d) simulation results involving bit interleaving over fading channels generated by the Fritchman model.

5.2 Error propagation characteristics of DES in a binary symmetric channel

In this section, equations are derived to model the error propagation of block ciphers and stream ciphers in a memoryless channel. The processing steps involved are illustrated in the block diagram shown in Figure 5.1.

5.2.1 Error propagation characteristics without channel coding

Error control coding is not used in this section in order to simplify the analysis. From Figure 5.1, it is seen that *DBER* can be formally defined as :

DBER = Bit Error Rate at the receiver, AFTER decryption

In contrast, the *BER* represents the channel bit error rate. In order to determine how *DBER* is related to *BER* for various cipher modes, we will need the following definition:

*P_b = Probability that a bit in a decrypted word
at the receiver is in error (no FEC coding)*

For stationary processes, $P_b = DBER$. Since all channel models considered in this thesis are stationary, the above statement will always be true. A prefix will be used to refer to a specific cipher mode (e.g. $_{cc}P_b$ for Cipher Block Chaining mode).

Another useful parameter is cryptographic degradation [67], defined as:

$$D = \frac{DBER}{BER} = \frac{P_b}{BER} \quad (5-1)$$

This parameter provides a useful measure of the degree of error propagation for a given cipher mode. When $D > 1.0$, the bit error rate after decryption is higher than the channel BER and error propagation has occurred.

Figure 5.1: Signal processing steps

5.2.1.1 Error propagation for Electronic Code Book mode

The configuration under analysis is shown in Figure 3.2 (page 85). Here, the receiver collects N bits of an encrypted word c'_m where $c'_m = \{c'_{i1}, c'_{i2}, \dots, c'_{iN-1}\}$ is the m -th encrypted word to be transmitted in the channel and where $M'_m = D_K(C'_m)$, and $C'_m = C_m \oplus E_m$. Hence, $E_m = \{e_{i1}, e_{i2}, \dots, e_{iN-1}\}$ and $c'_i = c_i \oplus e_i$ where $e_i = 1$ when a channel error occurs.

The following definition will also be required in the upcoming development:

A_m = Probability that one or more bits of the encrypted word $C'_m = \{c'_{i1}, c'_{i2}, \dots, c'_{iN-1}\}$ are received in error

Similarly, event A_{m-1} is the probability that one or more bits of the encrypted word c'_{m-1} are received in error. Conversely, $\bar{A}_m$ is the event where all N bits of encrypted word c'_m are received error-free. If $P(0, N)$ is the probability of having no errors in a block of N bits, then:

$$P(A_m) = 1 - P(0, N) \quad (5-2)$$

where

$$P(i, N) = \frac{N!}{(N-i)! i!} p^i (1-p)^{N-i} \quad (5-3)$$

for $0 \leq i \leq N$

for a binary symmetric channel with crossover probability p . The event B_{RCB} is defined as follows:

B_{RCB} = Event that a bit of the decrypted word $M'_m = \{m'_{i1}, m'_{i2}, \dots, m'_{iN-1}\}$ is in error, for ECB

Because of the avalanche effect of the DES described earlier, it is known that M'_m will have about $N/2$ bits in error if c'_m has one or more bits in error in it. Thus,

$$P(B_{ECB}|A_m) = 0.5 \quad (5-4)$$

Conversely $P(\bar{A}_m) = 1 - P(A_m)$ and $P(B_{ECB}|\bar{A}_m) = 0.0$ where A_m and $\bar{A}_m$ are mutually exclusive events. Hence,

$$\begin{aligned} {}_{ECB}P_b &= P(B_{ECB}) = P(B_{ECB}, A_m) + P(B_{ECB}, \bar{A}_m) \\ &= P(B_{ECB}|A_m) P(A_m) + P(B_{ECB}|\bar{A}_m) P(\bar{A}_m) \end{aligned} \quad (5-5)$$

Furthermore, $P(A_m) = \sum_{v_i=1}^N P(v_i, N) = 1 - P(0, N)$. Therefore,

$${}_{ECB}P_b = \frac{1 - P(0, N)}{2} \quad (5-6)$$

For a binary symmetric channel, we find that:

$${}_{ECB}P_b = \frac{1 - (1-p)^N}{2} \quad (5-7)$$

In Figure 5.2, the curve represents the theoretical values of ${}_{ECB}P_b$ as a function of p . The points shown as squares represent the measured values of ${}_{ECB}P_b$ based on simulation runs over the memoryless BSC. It can be seen that the measured values are in close agreement with the theoretical values. Note that ${}_{ECB}P_b$ approaches its maximum value of 0.5 when the channel BER is about 0.04.

Figure 5.2: Bit error probability after decryption for ECB mode in a BSC.

For ECB, cryptographic degradation in a memoryless BSC is given by:

$$D_{ECB} = \frac{{}_{ECB}P_b}{BER} = \frac{1 - (1-p)^N}{2p} \quad (5-8)$$

Cryptographic degradation varies from 32.0 to 1.0 as the channel BER increases from 0.0 to 0.5. Figure 5.3 illustrates the cryptographic degradation for ECB mode over BSC. Note that, as the

Figure 5.3: Cryptographic degradation for ECB Mode in a BSC.

channel BER increases, the average number of bits in error per received encrypted block increases. However, this does not result in a corresponding increase in the number of bits decrypted in error; there are always about $N/2$ bits decrypted in error regardless of the specific number of bits in error in the received block. Hence, D_{ECB} decreases towards 1.0 as the channel BER increases, but this cannot be interpreted as an

improvement in the system performance.

5.2.1.2 Error propagation for Cipher Block Chaining mode

The configuration for CBC is shown in Figure 3.3 (page 88). In the CBC mode of the DES, the receiver collects the N bits of codeword $C'_m = \{c'_1, c'_{1,1}, \dots, c'_{1,N-1}\}$ and performs the following operation:

$$M'_m = D_K(C'_m) \oplus C'_{m-1} \quad (5-9)$$

where $C'_{m-1} = \{c'_{1-N}, c'_{1-N,1}, \dots, c'_{1-1}\}$.

Let P_b be the probability that any one bit in the decrypted word M'_m is generated in error. From the decryption operation, it is clear that M'_m will contain bits in error if either C'_m or C'_{m-1} contain bits in error. Let B_{CBC} be the event that a bit is decrypted in error in the decrypted word M'_m of N bits. Thus,

$$P_{CBC} = P(B_{CBC}) \quad (5-10)$$

where

$$\begin{aligned} P(B_{CBC}) &= P[(B_{CBC}, A_{m-1}, \bar{A}_m) \cup (B_{CBC}, \bar{A}_{m-1}, A_m) \cup (B_{CBC}, A_{m-1}, A_m)] \\ &= P(B_{CBC}, A_{m-1}, \bar{A}_m) + P(B_{CBC}, \bar{A}_{m-1}, A_m) + P(B_{CBC}, A_{m-1}, A_m) \end{aligned} \quad (5-11)$$

given that the joint events are mutually exclusive. Using Bayes' theorem, we find that

$$\begin{aligned} P(B_{CBC}) &= P(B_{CBC}|A_{m-1}, \bar{A}_m) P(A_{m-1}, \bar{A}_m) + \\ &\quad P(B_{CBC}|\bar{A}_{m-1}, A_m) P(\bar{A}_{m-1}, A_m) + \\ &\quad P(B_{CBC}|A_{m-1}, A_m) P(A_{m-1}, A_m) \end{aligned} \quad (5-12)$$

Because the channel is memoryless, this simplifies to

$$\begin{aligned} P(B_{CBC}) &= P(B_{CBC}|A_{m-1}, \bar{A}_m) P(A_{m-1}) P(\bar{A}_m) + \\ &\quad P(B_{CBC}|\bar{A}_{m-1}, A_m) P(\bar{A}_{m-1}) P(A_m) + \\ &\quad P(B_{CBC}|A_{m-1}, A_m) P(A_{m-1}) P(A_m) \end{aligned} \quad (5-13)$$

where

$$\begin{aligned} P(A_m) &= P(A_{m-1}) = 1 - P(0, N) \\ P(\bar{A}_m) &= P(\bar{A}_{m-1}) = P(0, N) \end{aligned} \quad (5-14)$$

We now examine each one of the three joint events individually.

CASE 1: For joint event $(A_{m-1}, \bar{A}_m)$, assume that v_N bits have been received in error in the word $C'_{m-1} = \{c'_{i-N}, c'_{i-N+1}, \dots, c'_{i-1}\}$ (where $1 \leq v_N \leq N$). In addition, no bits are received in error in the word $C'_m = \{c'_i, c'_{i+1}, \dots, c'_{i+N-1}\}$. Here, $D_K(C'_m)$ is error-free since C'_m is error-free. However, $M'_m = D_K(C'_m) \oplus C'_{m-1}$ will have v_N bits in error in it. Thus, for a given value of v_N ,

$$P(B_{CBC}|v_N, A_{m-1}, \bar{A}_m) = \frac{v_N}{N} \quad (5-15)$$

The Chapman-Kolmogorov theorem states that [52, pp. 182]:

$$P(B_{CBC}|A_{m-1}, \bar{A}_m) = \sum_{v_N=1}^N P(B_{CBC}|v_N, A_{m-1}, \bar{A}_m) P(v_N|A_{m-1}, \bar{A}_m) \quad (5-16)$$

where $P(v_N|A_{m-1}, \bar{A}_m) = \frac{P(A_{m-1}, \bar{A}_m|v_N) P(v_N)}{P(A_{m-1}, \bar{A}_m)}$ by Bayes' theorem. Because of the

way A_{m-1} is defined, it is clear that $P(A_{m-1}|v_N) = 1$ for $v_N \geq 1$. Furthermore, for a memoryless channel, event $\bar{A}_m$ is independent of both A_{m-1} and v_N . As a result,

$$P(v_N|A_{m-1}, \bar{A}_m) = \frac{P(\bar{A}_m) P(v_N)}{P(A_{m-1}) P(\bar{A}_m)} = \frac{P(v_N)}{P(A_{m-1})} = \frac{P(v_N, N)}{1 - P(0, N)} \quad (5-17)$$

where $P(v_N, N)$ is the probability of having v_N bits in error in a block of N bits. Hence,

$$P(B_{CBC}|A_{m-1}, \bar{A}_m) = \frac{1}{N [1 - P(0, N)]} \sum_{v_N=1}^N v_N P(v_N, N) \quad (5-18)$$

for a memoryless channel. If we define $\bar{v}_N$ as

$$\bar{v}_N = \sum_{v_N=1}^N v_N P(v_N, N) \quad (5-19)$$

we get

$$P(B_{CBC}|A_{m-1}, \bar{A}_m) = \frac{\bar{v}_N}{N [1 - P(0, N)]} \quad (5-20)$$

CASE 2: Here, we consider the joint event $(\bar{A}_{m-1}, A_m)$. In other words, the word $c'_{m-1} = \{c'_{i-N}, c'_{i-N+1}, \dots, c'_{i-1}\}$ is error-free and the word $c'_m = \{c'_i, c'_{i+1}, \dots, c'_{i+N-1}\}$ contains v_N bits in error. Here, $c'_{m-1} = c_{m-1}$ but $D_K(c'_m) \neq D_K(c_m)$. Because of the avalanche effect of the DES, we know that $D_K(c'_m)$ at the receiver will differ from $M_m \oplus c_{m-1}$ at the transmitter in about $N/2$ bits, regardless of the number of bits in

error in the codeword c'_m . As a result,

$$P(B_{CBC}|v_N, \bar{A}_{m-1}, A_m) = P(B_{CBC}|\bar{A}_{m-1}, A_m) = \frac{1}{2} \quad (5-21)$$

for any value of v_N .

CASE 3: Finally, we consider the joint event (A_{m-1}, A_m) where v_1 bits are received in error in the word $c'_{m-1} = \{c'_{i-N}, c'_{i-N+1}, \dots, c'_{i-1}\}$ and v_2 bits are received in error in the word $c'_m = \{c'_i, c'_{i+1}, \dots, c'_{i+N-1}\}$ ($1 \leq v_1, v_2 \leq N$).

Because c'_m is corrupted, $D_K(c'_m)$ will have about half of its bits in error on average, regardless of the value of v_1 . The diffusion and confusion characteristics of the DES ensure that these bit errors are uniformly distributed over the block of N bits. Hence, it is reasonable to assume that about half of the v_1 bit errors of c'_{m-1} will accidentally correct bit errors of $D_K(c'_m)$ when the operation $M'_m = D_K(c'_m) \oplus c'_{m-1}$ is performed. It is also reasonable to assume that the other half of the v_1 bits will introduce bit errors where there were none before. It is therefore apparent that for the joint event (A_{m-1}, A_m) , about half the bits of M'_m will differ from those of M_m regardless of the exact value of v_1 and v_2 . Therefore, we conclude that:

$$P(B_{CBC}|A_{m-1}, A_m) = \frac{1}{2} \quad (5-22)$$

Using equation (5-13) given earlier (page 123), $_{CBC}P_b$ can be determined as follows:

$$\begin{aligned} _{CBC}P_b &= \frac{\overline{V_N}}{N [1 - P(0, N)]} [1 - P(0, N)] [P(0, N)] \\ &+ \frac{1}{2} [P(0, N)] [1 - P(0, N)] \\ &+ \frac{1}{2} [1 - P(0, N)]^2 \end{aligned} \quad (5-23)$$

Simplifying, we obtain

$$_{CBC}P_b = \frac{[1 - P(0, N)]}{2} + \frac{\overline{V_N}}{N} P(0, N) = _{ECB}P_b + \frac{\overline{V_N}}{N} P(0, N) \quad (5-24)$$

For a binary symmetric channel, we get

$$_{CBC}P_b = \frac{[1 - (1-p)^N]}{2} + \frac{\overline{V_N}}{N} (1-p)^N = _{ECB}P_b + \frac{\overline{V_N}}{N} (1-p)^N \quad (5-25)$$

where

$$\overline{V_N} = \sum_{v_N=1}^N v_N P(v_N, N), \text{ and} \quad (5-26)$$

$$P(v_N, N) = \frac{N!}{(N-v_N)! v_N!} p^{v_N} (1-p)^{N-v_N}$$

Here, it is clear that the probability of a bit being decrypted in error in a word of N bits is higher for Cipher Block Chaining mode than for Electronic Code Book mode. The bit error probability after decryption ($DBER$) for CBC mode in a BSC is shown in Figure 5.4. Measured values of $_{CBC}P_b$ based on simulation runs are also shown on the graph. Again, it is apparent that the simulated and theoretical values are very close to each other.

Figure 5.4: Bit error probability after decryption for CBC mode in a BSC.

The cryptographic degradation D_{CBC} for a memoryless channel is given by:

$$D_{CBC} = D_{ECB} + \frac{P(0, N) \overline{V_N}}{N BER} = \frac{1 - P(0, N)}{2 BER} + \frac{P(0, N) \overline{V_N}}{N BER} \quad (5-27)$$

For a binary symmetric channel, this becomes

$$D_{CBC} = D_{ECB} + \frac{(1-p)^N \overline{V_N}}{N p} = \frac{1 - (1-p)^N}{2 p} + \frac{(1-p)^N \overline{V_N}}{N p} \quad (5-28)$$

Again, it is apparent that the cryptographic degradation for CBC mode is higher than for ECB mode for all values of p and N .

5.2.1.3 Error propagation for Output Feedback mode

The configuration for this mode is shown in Figure 3.5 (page 92). It was mentioned in chapter 3 that OFB mode is essentially a substitution cipher which processes bits independently of each other. Hence, no error propagation takes place and therefore:

${}_{OFB}P_b = BER \quad (5-29)$

where BER is the channel bit error rate (p for a BSC). This is true for both memoryless and fading channels. Note also that there is no cryptographic degradation for this mode.

5.2.1.4 Error propagation for Cipher Feedback mode

The configuration for CFB mode is shown in Figure 3.6 (page 96). The development of the analytical expression for ${}_{CFB}P_b$ is considerably more involved than for the other modes and is not presented in this work. Nevertheless, an illustrative example is given in this section, in order to highlight the subtle differences between CFB and CBC modes.

For CFB mode, the receiver collects the s bits of codeword c'_m and performs the operation $M'_m = c'_m \oplus K'_m$ where the decrypted word is given by $M'_m = \{m'_1, m'_{1+1}, \dots, m'_{1+s-1}\}$. As before, $c'_m = c_m \oplus E_m$ where c_m is the encrypted word at the transmitter and E_m is a vector of s bits, which represents the errors added by the channel: $E_m = \{e_1, e_{1+1}, \dots, e_{1+s-1}\}$. Here, $e_j = 1$ if a channel bit error occurs,

otherwise, $e_j = 0$ where $i \leq j \leq i+s-1$. The value of N is the length of the memory buffer in bits and can be different from 64.

If the memory buffer contains bits in error, κ'_m will differ from κ_m (at the transmitter) in about $s/2$ bit positions and M'_m will have $s/2$ bit errors in it. Similarly, if c'_m has v_s bits in error in it, M'_m will have v_s bit errors in it. So far, this is very similar to CBC mode. However, the following example will show that the two modes behave differently when faced with an identical channel error sequence.

Consider the bit error sequence $\dots 00010000000001000\dots$ ¹. For CBC mode, two possibilities arise:

a) the error cluster is contained within one encrypted block of 64 bits, leading to about $\frac{64}{2} + 2 = 34$ bit errors (on average).

b) the error cluster spans two adjacent blocks, leading to $\frac{64}{2} + \frac{64}{2} + 1 = 65$ bit errors on average.

For CFB mode (with $s=1$), this error cluster will lead to $1 + \frac{64}{2} + \frac{10}{2} = 38$ bit errors (on average). The first error corresponds to the first corrupted bit c'_m being XOR-ed with κ'_m (which is error-free). Of the next 64 bits, about half are in error. As the first error bit leaves the memory buffer, the second error bit will remain in the buffer for an additional 10 bit intervals, leading to

¹ To simplify the argument, assume both cipher systems are initially error-free and that enough error-free bits follow the last bit error so that the latter will be cleared from both systems before the next error occurs.

5 more bit errors (on average). For $s = 8$, the two adjacent bytes '00010000' and '00000100' lead to about $1 + 8 \times \frac{8}{2} + \frac{8}{2} = 37$ bit errors.

In any event, it can be seen that the same error sequence may lead to CBC outperforming CFB, or vice-versa. It is also clear that the value of s has an impact on the performance of CFB mode².

Hence, the *DBER* of the two modes may differ significantly even when both are used under identical channel conditions.

5.2.2 Error propagation characteristics of DES in BSC with BCH channel coding

Figure 5.5: Signal processing steps, with FEC coding

In this section, the performance of binary BCH codes on the probability of a bit being decrypted in error at the receiver (*DBER*) for a memoryless channel is studied. The analysis is carried out for ECB, CBC and OFB cipher modes. Analytical expressions for the *DBER* are compared with simulation results. The communication link involved for the simulations is shown in

Figure 5.5.

For ECB, CBC and CFB modes, the probability of any bit being decrypted correctly is dependent on all $N = 64$ bits in a given block being received error-free at the receiver. For instance, a (7,4,1) block code requires 16 codewords for a block of 64

² Note that if $s = N$, CFB mode is identical to CBC mode in terms of *DBER*.

encrypted bits. For ECB mode, if any one of these codewords suffers a decoding failure, one or more of the 64 encrypted bits will be corrupted and about half of the bits will be wrong after decryption. For ECB and CBC, the boundaries of a block of 64 encrypted bits are clearly defined but this is not the case for CFB mode. For example, $\{c_1, c_2, \dots, c_{64}\}$ and $\{c_{65}, c_{66}, \dots, c_{128}\}$ are the first two blocks to be processed at the receiver for CBC and ECB modes. Note that a bit belonging to a block does not belong to another one. However, for CFB mode, the first two blocks will have this form when the word length is $s = 1$; $\{c_1, c_2, \dots, c_{64}\}$ and $\{c_2, c_3, \dots, c_{65}\}$. Here, block boundaries are not clearly defined. As a result, the average number of codewords required to encode a block of 64 encrypted bits may be different for CFB mode. Hence, we define

$\bar{N}_{bc}$ = Average number of codewords required to encode a block of N encrypted bits for block cipher modes ECB and CBC

and

$\bar{N}_{sc}$ = Average number of codewords required to encode a block of N encrypted bits for stream cipher mode CFB

where encrypted blocks are $N = 64$ bits in all our simulations. Note that OFB mode operates on single encrypted bits independently of each other. Hence, for that mode, successful decryption of any one bit involves only one codeword for all binary BCH codes.

For block cipher modes, consider the case when a BCH (127,57,11) code is used. The first block of 64 encrypted bits at the receiver is obtained from all 57 bits of the first transmitted codeword, plus 7 bits from the second one. We represent this situation as follows: $\{57, 7\}$. The next encrypted block will consist

of the last 50 bits from codeword two (bits 8 to 57) and the first 14 bits from the third codeword (i.e. {50,14}). We can then generate the following sequence:

{57,7}	{50,14}	{43,21}	{36,28}	{29,35}	{22,42}	{15,49}
{8,56}	{1,57,6}	{51,13}	{44,20}	{37,27}	{30,34}	{23,41}
{16,48}	{9,55}	{2,57,5}	{52,12}	{45,19}	{38,26}	{31,33}
{24,40}	{17,47}	{10,54}	{3,57,4}	{53,11}	{46,18}	{39,25}
{32,32}	{25,39}	{18,46}				

Note that after {32,32}, the sequence repeats itself with the terms reversed (e.g. {39,25} becomes {25,39}, {46,18} becomes {18,46}, etc), progressing backwards to the beginning. The last term is {7,57}, after which the entire sequence repeats itself. The average number of codewords required to encode an encrypted block of 64 bits is then given by:

$$\bar{N}_{sc} = \frac{(51) \times (2) + (6) \times (3)}{57} = 2.105$$

For CFB mode with $s=1$, the first block of 64 bits in the memory buffer has the form {57,7}. For the next decrypted bit, the sequence is shifted by one bit to the left and the memory buffer then contains the last 56 bits of the first encoded word (bits 2 to 57) and the first 8 bits of the second one (i.e. {56,8}). As a result, the sequence takes this form:

{57,7}	{56,8}	{55,7}		{2,57,5}	{1,57,6}
--------	--------	--------	-------	----------	----------

In this case, $\bar{N}_{sc} = 2.105$ also. Note that if the word length is $s=8$, the memory buffer will contain the last 49 bits of codeword one (bits 9 to 57) and the first 15 bits of codeword two when the second word of 8 bits is decrypted. Hence, the sequence starts out as follows:

{57,7}	{49,15}	{41,23}	
--------	---------	---------	-------

It can be verified that in this case also, $\bar{N}_{sc} = 2.105$.

However, consider what happens when the binary BCH code used is (127,64,10). For ECB and CBC modes, all encrypted blocks are encoded by exactly one codeword so that $\bar{N}_{bc} = 1.0$. However, for CFB mode with $s = 1$, the sequence takes this form:

{64} {63,1} {62,2} {61,1} ... {2,62} {1,63}

The average number of codewords required to encode a block of 64 encrypted bits is then given by:

$$\bar{N}_{sc} = \frac{(63) \times (2) + (1) \times (1)}{64} = 1.984$$

Furthermore, if $s = 8$, the sequence will take this form:

{64} {56,8} {48,16} {40,24} {32,32} {24,40} {16,48}
{8,56}

In this case, we find that

$$\bar{N}_{sc} = \frac{(7) \times (2) + (1) \times (1)}{8} = 1.875$$

which is lower than when $s = 1$. Note that, for ECB and CBC mode, going from the (127,57,11) code to the (127,64,10) code results in a reduction of 52% in the average number of codewords required to encode an encrypted block. For CFB, the same transition yields a reduction of 5.7% and 10.9% for $s = 1$ and for $s = 8$ respectively. Since a decryption failure will occur if one or more of the codewords (required to encode the block) is decoded incorrectly, it is clear that a decrease in $\bar{N}_{bc}$ or $\bar{N}_{sc}$ will decrease the probability of bits being decrypted in error (for a fixed code length and error correcting capability).

The average number of codewords required for ECB, CBC and CFB modes was determined for all values of k used in the simulations. These are shown in the following table:

Table 5.1: $\bar{N}_{bc}$ and $\bar{N}_{sc}$ of ECB, CBC and CFB modes for all (n, k, t) codes used in the simulations			
k	$\bar{N}_{bc}$ ECB, CBC	$\bar{N}_{sc}$ CFB ($s = 1$)	$\bar{N}_{sc}$ CFB ($s = 8$)
4	16.0	16.75	16.0
5	13.6	13.6	13.6
7	10.0	10.0	10.0
11	6.727	6.727	6.727
16	4.0	4.94	4.5
18	4.44	4.5	4.44
24	3.33	3.625	3.33
26	3.385	3.423	3.385
30	3.07	3.1	3.07
43	2.465	2.465	2.465
45	2.4	2.4	2.4
57	2.105	2.105	2.105
64	1.0	1.984	1.875
92	1.652	1.685	1.652

For CFB mode, it is clear that the average number of codewords is significantly higher than for block cipher modes when k can be divided evenly in 64. It is also affected by the word length s .

5.2.2.1 Error propagation: ECB mode with BCH coding

Electronic Code Book mode operates on blocks of N bits where $N = 64$. After the $\bar{N}_{bc}$ codewords have been transmitted over the channel (and corrupted by it), they are decoded at the receiver and the encrypted block retrieved is given by $\tilde{c}_m = \{\tilde{c}_1, \tilde{c}_{1,1}, \dots, \tilde{c}_{1,N-1}\}$ where the tilde emphasizes the fact that these encrypted bits were encoded before transmission. After decryption at the receiver, the

decrypted block $\tilde{M}_m = \{\tilde{m}_1, \tilde{m}_{i+1}, \dots, \tilde{m}_{i+N-1}\}$ is generated. If all the codewords were decoded correctly, this block is identical to the decrypted block at the transmitter. Otherwise, it will contain errors.

When FEC codes are used, each k bit sequence of encrypted bits is mapped into an n bit sequence which is then transmitted over the channel. With this in mind, we define the following event:

E_m = Event that one or more of the $\bar{N}_{BC}$ codewords required to encode the encrypted block $C_m = \{c_i, c_{i+1}, \dots, c_{i+N-1}\}$ are decoded in error

Note that this event involves $\bar{N}_{BC} \times n$ transmitted bits instead of only N .

Let $P_c(V_N, \bar{N}_{BC})$ be the probability that exactly v_N out of $[\bar{N}_{BC}]$ codewords will be decoded in error at the receiver. For a memoryless channel, this becomes

$$P_c(V_N, \bar{N}_{BC}) = \frac{[\bar{N}_{BC}]!}{([\bar{N}_{BC}] - V_N)! V_N!} P_c^{V_N} (1 - P_c)^{[\bar{N}_{BC}] - V_N} \quad (5-30)$$

where $1 \leq V_N \leq [\bar{N}_{BC}]$ and P_c is the probability that an (n, k, t) codeword has $t+1$ or more bits in error in it:

$$P_c = \sum_{v_n=t+1}^n P(v_n, n) = 1 - \sum_{v_n=0}^t P(v_n, n) \quad (5-31)$$

Note that using the ceiling function $(\lceil \cdot \rceil)$ for $\bar{N}_{BC}$ will result in $P_c(V_N, \bar{N}_{BC})$ being slightly higher than it is in reality when V_N is greater than zero. For $V_N = 0$, we will use the exact value of $\bar{N}_{BC}$. The above results lead to this equation:

$$P(E_m) = 1 - P(0, \bar{N}_{BC}) \quad (5-32)$$

For ECB mode, we know that if one or more bits in the decoded sequence $\{\hat{a}_1, \hat{a}_{1,1}, \dots, \hat{a}_{1,N-1}\}$ are in error, about $N/2$ bits will be decrypted in error in the sequence $\{\hat{m}_1, \hat{m}_{1,1}, \dots, \hat{m}_{1,N-1}\}$. Hence, it is clear that

$$P(B_{ECB}|E_m) = \frac{1}{2} \quad (5-33)$$

if we define P_{cb} as follows

P_{cb} = Probability that a bit in a decrypted word
at the receiver is in error (with FEC coding)

${}_{ECB}P_{cb}$ is then given by:

$${}_{ECB}P_{cb} = P(B_{ECB}) = P(B_{ECB}|E_m) P(E_m) \quad (5-34)$$

The above equation simplifies to

$${}_{ECB}P_{cb} = \frac{1 - P_c(0, \bar{N}_{BC})}{2} = \frac{1 - (1 - p_c)^{\bar{N}_{BC}}}{2} \quad (5-35)$$

For a binary symmetric channel, this becomes

$${}_{ECB}P_{cb} = \frac{1 - \left(\sum_{v_n=0}^t P(v_n, n) \right)^{\bar{N}_{BC}}}{2} = \frac{1 - \left(\sum_{v_n=0}^t \frac{n!}{(n - v_n)! v_n!} p^{v_n} (1-p)^{n-v_n} \right)^{\bar{N}_{BC}}}{2} \quad (5-36)$$

where p is the channel BER. Note that as the error correcting capability t of the code increases, the term in the brackets increases (approaches 1) and therefore the ${}_{ECB}P_{cb}$ tends to become smaller. However, as t increases, k decreases and $\bar{N}_{BC}$ increases. Hence, $(1 - p_c)^{\bar{N}_{BC}}$ decreases and ${}_{ECB}P_{cb}$ tends to increase. Thus, t and $\bar{N}_{BC}$ have opposite effects on the ${}_{ECB}P_{cb}$.

In Figure 5.6 the theoretical $E_b P_{cb}$ as a function of p for different codes is shown as smooth curves. The individual points shown were derived experimentally from simulation runs. Again, it is apparent that theoretical and experimental values of $E_b P_{cb}$ are in close agreement.

Figure 5.6: Theoretical and experimental $E_b P_{cb}$ of ECB mode over BSC, $n = 31$.

5.2.2.2 Error propagation: CBC mode with BCH coding

As before, it can be shown that

$$\begin{aligned} {}_{CBC}P_{cb} &= P(B_{CBC}|E_{m-1}, \bar{E}_m) P(E_{m-1}) P(\bar{E}_m) \\ &+ P(B_{CBC}|\bar{E}_{m-1}, E_m) P(\bar{E}_{m-1}) P(E_m) \\ &+ P(B_{CBC}|E_{m-1}, E_m) P(E_{m-1}) P(E_m) \end{aligned} \quad (5-37)$$

where E_{m-1} is defined exactly like event E_m but for encrypted block $C_{m-1} = \{c_{i-N}, c_{i-N+1}, \dots, c_{i-1}\}$. By using arguments similar to those given earlier, it is easy to demonstrate that

$$P(B_{CBC}|\bar{E}_{m-1}, E_m) = P(B_{CBC}|E_{m-1}, E_m) = \frac{1}{2} \quad (5-38)$$

However, the joint event $(E_{m-1}, \bar{E}_m)$ leads to a slightly different result when forward error correction codes are used. For this joint event, assume v_N codewords ($1 \leq v_N \leq \lceil \bar{N}_{BC} \rceil$) are overwhelmed by channel errors in the $\lceil \bar{N}_{BC} \rceil$ codewords which encode the encrypted sequence

$\{c_{i-N}, c_{i-N+1}, \dots, c_{i-1}\}$. Each codeword with $t+1$ or more channel bit errors in it will be decoded incorrectly at the receiver. When this occurs, no more than $t+v$ of the k data bits will be in error after decoding [54, pp. 405]. To simplify calculations, we use a loose upper bound and assume that, on average, $\frac{k}{2}$ data bits are in error (i.e. the decoded data vector is uncorrelated with the transmitted vector). If V_N codewords out of $[\bar{N}_{BC}]$ are corrupted with $t+1$ or more bit errors each, no more than $\frac{V_N k}{2}$ bits will be decoded in error in the sequence $\{\tilde{c}_{i-N}, \tilde{c}_{i-N+1}, \dots, \tilde{c}_{i-1}\}$. This will cause $\frac{V_N k}{2}$ bit errors in the word $\{\tilde{m}_i, \tilde{m}_{i+1}, \dots, \tilde{m}_{i+N-1}\}$ after decryption. As a result,

$$P(B_{CBC}|E_{m-1}, \bar{E}_m, V_N) = \frac{V_N k}{2N} \quad (5-39)$$

Again, we find that

$$P(B_{CBC}|E_{m-1}, \bar{E}_m) = \sum_{V_N=1}^{[\bar{N}_{BC}]} P(B_{CBC}|E_{m-1}, \bar{E}_m, V_N) P(V_N|E_{m-1}, \bar{E}_m) \quad (5-40)$$

where

$$P(V_N|E_{m-1}, \bar{E}_m) = \frac{P(E_{m-1}, \bar{E}_m|V_N) P(V_N)}{P(E_{m-1}, \bar{E}_m)} = \frac{P(V_N)}{P(E_{m-1})} \quad (5-41)$$

Therefore,

$$\begin{aligned} P(B_{CBC}|E_{m-1}, \bar{E}_m) &= \sum_{V_N=1}^{[\bar{N}_{BC}]} \frac{V_N P_C(V_N, \bar{N}_{BC}) k}{2N[1 - P_C(0, \bar{N}_{BC})]} \\ &= \frac{k \sum_{V_N=1}^{[\bar{N}_{BC}]} V_N P_C(V_N, \bar{N}_{BC})}{2N[1 - P_C(0, \bar{N}_{BC})]} \end{aligned} \quad (5-42)$$

If we define $\bar{V}_N$ as

$$\bar{V}_N = \sum_{V_N=1}^{[\bar{N}_{BC}]} V_N P_C(V_N, \bar{N}_{BC}) \quad (5-43)$$

we obtain

$$P(B_{CBC}|E_{m-1}, \bar{E}_m) = \frac{\bar{V}_N k}{2N[1 - P_c(0, \bar{N}_{BC})]} \quad (5-44)$$

The value of $_{CBC}P_{cb}$ can now be determined as follows:

$$\begin{aligned} _{CBC}P_{cb} &= P(B_{CBC}|\bar{E}_{m-1}, E_m) P(\bar{E}_{m-1}) P(E_m) \\ &+ P(B_{CBC}|E_{m-1}, E_m) P(E_{m-1}) P(E_m) \\ &+ P(B_{CBC}|E_{m-1}, \bar{E}_m) P(E_{m-1}) P(\bar{E}_m) \\ &= \frac{1}{2} [1 - P_c(0, \bar{N}_{BC})] [P_c(0, \bar{N}_{BC})] \\ &+ \frac{1}{2} [1 - P_c(0, \bar{N}_{BC})]^2 \\ &+ \left[\frac{\bar{V}_N k}{2N[1 - P_c(0, \bar{N}_{BC})]} \right] [1 - P_c(0, \bar{N}_{BC})] [P_c(0, \bar{N}_{BC})] \end{aligned} \quad (5-45)$$

which can be simplified to the form:

$$\begin{aligned} _{CBC}P_{cb} &= \frac{1 - P_c(0, \bar{N}_{BC})}{2} + \frac{\bar{V}_N k P_c(0, \bar{N}_{BC})}{2N} \\ &= _{ECB}P_{cb} + \frac{\bar{V}_N k P_c(0, \bar{N}_{BC})}{2N} \end{aligned} \quad (5-46)$$

For a BSC, this becomes

$$\begin{aligned} _{CBC}P_{cb} &= \frac{1 - \left(\sum_{v_n=0}^t \frac{n!}{(n-v_n)! v_n!} p^{v_n} (1-p)^{n-v_n} \right)^{\bar{N}_{BC}}}{2} + \\ &\frac{\bar{V}_N k \left(\sum_{v_n=0}^t \frac{n!}{(n-v_n)! v_n!} p^{v_n} (1-p)^{n-v_n} \right)^{\bar{N}_{BC}}}{2N} \end{aligned} \quad (5-47)$$

Figure 5.7: Theoretical and experimental ${}_{CBC}P_{cb}$ of CBC mode over BSC, $n = 31$

values of channel BER. In Figure 5.7, the theoretical and simulation results for ${}_{CBC}P_{cb}$ as a function of p are shown for different codes.

5.2.2.3 Error propagation: OFB mode with BCH coding

For OFB mode, individual cipher bits are encrypted and decrypted individually. Hence, the probability ${}_{OFB}P_{cb}$ that a bit will be decrypted incorrectly is dependent only on the probability of a decoding failure p_c for the codeword which contains this encrypted bit. Using the simplification given earlier, we can say that an (n, k, t) codeword which suffers a decoding failure will

have $\frac{k}{2}$ bits in error (on average) after decoding. Hence,

Note that the assumption of independence for joint events $(E_{m-1}, \bar{E}_m)$, $(\bar{E}_{m-1}, E_m)$ or (E_{m-1}, E_m) is not always true. For some values of k , it will often occur that an (n, k, t) codeword will contain encrypted bits from both $C_{m-1} = \{c_{t-N}, c_{t-N+1}, \dots, c_{t-1}\}$ and $C_m = \{c_t, c_{t+1}, \dots, c_{t+N-1}\}$. Hence, the above equations are inaccurate in that respect. Note also that CBC mode is again consistently worse than ECB mode for all

Figure 5.8: Theoretical and experimental ${}_{OFB}P_{cb}$ of OFB mode over BSC, $n = 15$

$$P(B_{OFB}|F_m) = \frac{1}{2} \quad (5-48)$$

where B_{OFB} is the event that cipher bit $\tilde{m}_i \neq m_i$ and F_m is the event that the m -th codeword (which contains encrypted bit c_i) is decoded incorrectly. As a result, we can say that

$$\begin{aligned} {}_{OFB}P_{cb} &= DBER = P(B_{OFB}) = P(B_{OFB}|F_m) P(F_m) \\ &= \frac{P_c}{2} \end{aligned} \quad (5-49)$$

For a binary symmetric channel, this becomes

$${}_{OFB}P_{cb} = \frac{\sum_{v_n=c+1}^n \frac{n!}{(n-v_n)! v_n!} p^{v_n} (1-p)^{n-v_n}}{2} \quad (5-50)$$

The theoretical and experimental values of ${}_{OFB}P_{cb}$ as a function of BSC crossover probability p are shown in Figure 5.8.

5.2.2.4 Error propagation: CFB mode with BCH coding

It was shown earlier that CFB mode behaves differently than CBC mode for identical error sequences. The use of channel coding leads to additional differences since the number of codewords required to encode the 64 bits of the memory buffer for CFB mode $\bar{N}_{sc}$ is sometimes different than $\bar{N}_{bc}$ for CBC mode. Thus, it can be expected that the *DBER* performance of these two modes can vary significantly when they are used under similar coding and channel conditions.

5.3 Analysis of simulation results over binary symmetric channels

In this section we analyze the experimental results of the

simulation runs of all four encryption modes over binary symmetric channels. The complete results are shown in table form in annex B. Annex A contains additional graphs based on the results in annex B.

The processing steps carried out during the simulations are shown in Figure 5.5 (page 129). For the case of bit interleaving, the interleaving step is carried out just before the channel pass step of Figure 5.5 (the de-interleaving is performed just after the channel pass). Note that only the channel itself is simulated in the simulations³; all the other steps are carried out as they would be in a real communication system. Encryption and decryption is performed as described in chapter 3 for the four cipher modes used. Encoding, decoding (Berlekamp-Massey algorithm) and interleaving is carried out as described in chapter 4. The decoding program always generates a data vector, even when the error correcting capability of the code has been exceeded (i.e. no erasures are allowed). The computer programs used to perform the various processing steps are provided in a separate document (annex D).

5.3.1 OFB mode simulation results

For OFB mode, the results of the simulation runs are shown in tables B.1 to B.3 in annex B, starting at page 244.

Figure 5.9 shows the *DBER* versus the channel BER for all codes whose rate is roughly equal to 0.75. The curve shown for the case where no BCH coding is used provides a reference since *DBER* is approximately equal to the BER. Hence, when a BCH code yields a value of *DBER* which is lower than the corresponding value for the curve with no coding, the BCH code in question provides a net reduction in the error rate. When the *DBER* is higher, the BCH code introduces more errors than it corrects.

³ This is done by using the Markov chain models described in chapter 2.

Beyond a given channel BER, all BCH codes result in a $DBER$ which is higher than if no coding had been used (i.e. $P_{cb} > P_b$). This is caused by an avalanche effect of the BCH code which is somewhat similar to that of the DES. When a codeword is corrupted by $t+1$ errors (for example), the decoding of the codeword will fail and may result in up to $2t+1$ of the decoded data bits being in error. This is an inherent property of all block codes.

From Figure 5.9 to Figure 5.11, these additional observations can be made about the avalanche effect of the BCH codes:

- i) As n and k increase for a fixed code rate R , the severity of the avalanche effect increases ($DBER$ increases),
- ii) As n and k increase (for a fixed code rate), the minimum channel BER at which $P_{cb} = P_b$ decreases. Hence, long codes become useless (i.e. $P_{cb} > P_b$) earlier (i.e. for lower values of BER) than shorter codes⁴.

As k increases, a decoding failure may result in more bit errors being generated. For example, a BCH (15,11,1) codeword may end up with about 5.5 bit errors (on average) after a decoding failure as opposed to about 13 bit errors when a BCH (31,26,1) code is used.

Furthermore, longer codes reach the point where $P_{cb} = P_b$ at a lower value of channel BER than shorter codes because of the significant increase in the severity of the avalanche effect (as k increases). This increase in the error rate more than offsets any possible decrease in the probability of a decoding failure ($p_c = 1 - \sum_{i=0}^t P(i,n)$) for a codeword.

⁴ Recall that for OFB mode, $P_b \approx BER$

Figure 5.9: OFB mode over BSC with BCH coding ($R = 0.75$).

Figure 5.10: OFB mode over BSC with BCH coding ($R = 0.50$).

Figure 5.11: OFB mode over BSC with BCH coding ($R = 0.33$).

The channel BER above which a BCH code ceases to be useful (when $P_{cb} \geq P_b$) for OFB mode can be determined as follows:

$$DBER = {}_{OFB}P_{cb} = \frac{P_c}{2} = P_{MAX} = BER$$

$$DBER = \frac{\left(1 - \sum_{i=0}^t \frac{n!}{i!(n-i)!} P_{MAX}^i (1-P_{MAX})^{n-i}\right)}{2} = P_{MAX} \quad (5-51)$$

where P_{MAX} is the value of channel BER for which $P_{cb} = P_b$. The values of P_{MAX} for OFB mode over binary symmetric channels are given in the following table for the codes used in the simulations, along

with the code rate R and the channel capacity.

Table 5.2: Maximum channel BER for which $P_{ch} \leq P_b$ for all codes, in memoryless BSC				
Code	Code rate R	P_{MAX}	P_c	Capacity (bits)
(7,4,1)	0.5714	0.170	0.3396	0.3423
(15,11,1)	0.7333	0.0233	0.0466	0.8404
(15,7,2)	0.4667	0.1090	0.2197	0.5031
(15,5,3)	0.3333	0.2270	0.4538	0.2272
(31,26,1)	0.8387	0.00471	0.0094	0.9568
(31,16,3)	0.5161	0.0623	0.1246	0.6635
(31,11,5)	0.3548	0.145	0.2888	0.4028
(63,45,3)	0.7143	0.0206	0.0411	0.8550
(63,30,6)	0.4762	0.0682	0.1363	0.6409
(63,24,7)	0.3810	0.0861	0.1721	0.5769
(63,18,10)	0.2857	0.1430	0.2854	0.4080
(63,16,11)	0.2540	0.1630	0.3260	0.3587
(127,92,5)	0.7240	0.0195	0.0390	0.8613
(127,64,10)	0.5039	0.0575	0.1150	0.6826
(127,57,11)	0.4488	0.0658	0.1314	0.6501
(127,43,14)	0.3386	0.0913	0.1826	0.5593

For a given code rate, P_{MAX} usually decreases as n increases⁵.

In addition, the table indicates that, as k decreases for a fixed n , the value of P_{MAX} increases. Hence, codes with a low rate suffer from a less severe avalanche effect than codes with a higher

⁵ The (31,26,1) code has the lowest value of P_{MAX} for all codes of rate $R = 0.75$ but this is probably due to the fact that its real rate ($R = 0.8387$) is not as close to 0.75 as the other codes.

rate (for a same n and BER). This higher crossover point is also apparent from Figure 5.12. As the code rate (and k) decreases for a fixed n , a decoding failure leads to less bit errors and the probability of a decoding failure p_c decreases (since t increases). For instance, a (127,92,5) codeword has $p_c = 0.9430$ for a BER of $p = 7.932e-2$ in contrast to $p_c = 0.4265$ for a (127,64,10) codeword.

Figure 5.12 illustrates the *DBER* versus BER for codes of a fixed length $n = 15$ and varying k (see also Figure A.1 to Figure A.3 in annex A).

Figure 5.12: OFB mode over BSC with BCH coding ($n = 15$).

For a fixed block length n , the code performance is directly related to the code rate. Hence, the lower the value of k is, the better the code is. As k decreases, t increases and the decoding failure probability p_c decreases.

5.3.2 ECB mode simulation results

The complete results of the simulation runs for ECB mode over binary symmetric channels are shown in tables B.4 to B.6 in annex B, starting at page 246. Comparison of OFB and ECB modes makes it immediately apparent that ECB mode has a *DBER* substantially higher than OFB mode.

Figure 5.13 to Figure 5.15 make it clear that there is no evidence of the BCH code avalanche effect for ECB mode. The use of any BCH code for any value of channel BER always results in $P_{cb} < P_b$ (in contrast with OFB). The reason behind this lies in the fact that ECB mode has a significant degree of error propagation whereas OFB mode has none, resulting in a *DBER* which is significantly higher for ECB mode. Hence, the avalanche effect of the codes is hidden by this additional error rate degradation; this implies that, for ECB mode, the error reduction achieved by channel coding is always superior to the increase in the error rate caused by the avalanche effect of the BCH code.

As is the case for OFB, Figure 5.13 indicates that, beyond a given channel BER, shorter codes outperform long ones for a given code rate. However, it is also clear by comparing the graphs for OFB and ECB modes of corresponding code rate, that the crossover point below which a longer code outperforms a shorter one of similar rate is significantly higher for ECB mode than it is for OFB mode; for instance, Figure 5.13 reveals that the crossover point for the (31,26,1) code relative to the (15,11,1) code is roughly $p = 1.5e-3$ for ECB mode versus $p = 8.0e-4$ for OFB mode⁶.

⁶ The theoretical value of the crossover point for any mode can be determined by setting P_{cb} equal for two different BCH codes and solving for the channel BER.

Figure 5.13: ECB mode over BSC with BCH coding ($R = 0.75$).

Figure 5.14: ECB mode over BSC with BCH coding ($R = 0.50$).

Figure 5.15: ECB mode over BSC with BCH coding ($R = 0.33$).

Similarly, the ECB mode (63,24,7) code outperforms the (15,5,3) code for any $p < 7.0e-2$ (Figure 5.15) as opposed to $p < 4.0e-2$ for the OFB mode (Figure 5.11, page 144). The fact that the (63,24,7) code outperforms the (15,5,3) code at a BER of $6.0e-2$ for ECB (for example) while the converse is true for OFB mode at that channel BER seems to indicate that long codes are more effective than shorter codes in reducing the *DBER* of ECB mode relative to OFB mode. The reason behind this behaviour has to do with the significant difference in how the *DBER* is related to the BCH codes for the two cipher modes. For OFB mode, the *DBER* is a function of the decoding failure probability p_c . For ECB mode, the *DBER* is also a function of the probability of one or more of the $\bar{N}_{BC}$ codewords (required to encode a block of N encrypted bits) being decoded incorrectly. For example, consider the results given below:

Table 5.3: <i>DBER</i> versus BER over memoryless BSC				
BER	<i>DBER</i> OFB (15, 5, 3)	<i>DBER</i> ECB (15, 5, 3)	<i>DBER</i> OFB (63, 24, 7)	<i>DBER</i> ECB (63, 24, 7)
3.977e-2	9.4e-4	1.49e-2	1.37e-3	5.62e-3
7.932e-2	1.2e-2	1.416e-1	5.55e-2	1.756e-1

where

Table 5.4: Decoding failure probability and probability of one or more of $\bar{N}_{BC}$ codewords being decoded incorrectly over memoryless BSC				
BER	(15, 5, 3) p_c	(15, 5, 3) $1 - P_c(0, \bar{N}_{BC})$	(63, 24, 7) p_c	(63, 24, 7) $1 - P_c(0, \bar{N}_{BC})$
3.977e-2	2.40e-3	3.21e-2	3.44e-3	1.15e-2
7.932e-2	2.66e-2	3.07e-1	1.24e-1	3.57e-1

Here, p_c increases by a factor of 1.43 as the code is changed from (15, 5, 3) to (63, 24, 7), for a channel BER of 3.977e-2. In contrast, the probability of one or more of $\bar{N}_{BC}$ codewords being incorrectly decoded decreases by a factor of 2.79 under the same conditions. Similarly, for a channel BER of 7.932e-2, p_c increases by a factor of 4.66 as the code is changed from (15, 5, 3) to (63, 24, 7) versus an increase of only 1.16 for the probability of one or more of $\bar{N}_{BC}$ codewords being decoded incorrectly. For a channel BER 3.977e-2, the change in code results in an increase in the *DBER* for OFB mode but leads to a decrease for ECB mode. For the channel BER of 7.932e-2, the *DBER* increases for both ECB and OFB modes, but the increase is higher for OFB than for ECB mode. Note that for ECB mode, $\bar{N}_{BC}$ changes from 13.6 to 3.33 as we go from the (15, 5, 3) to the (63, 24, 7) code while $\bar{N}_{sc}$ is always 1.0 for OFB mode. Recall that, for ECB mode,

$${}_{ECB}P_{cb} = \frac{1 - (1 - p_c)^{\bar{N}_{BC}}}{2} = \frac{1 - \left(\sum_{v_n=0}^t \frac{n!}{(n - v_n)! v_n!} p^{v_n} (1-p)^{n-v_n} \right)^{\bar{N}_{BC}}}{2} \quad (5-52)$$

As the exponent decreases, $(1 - p_c)^{\bar{N}_{BC}}$ tends to increase rapidly while p_c changes more slowly; hence, ${}_{ECB}P_{cb}$ decreases rapidly when a longer code is used.

The relative independence of ${}_{ECB}P_{cb}$ on the value of p_c can be seen more clearly in the following sample results from the simulation runs (channel $BER = 7.932e-2$ and $R = 0.50$):

Table 5.5: Decoding failure probability and probability of one or more of $\bar{N}_{BC}$ codewords being decoded incorrectly over a memoryless BSC ($BER = 7.932e-2$)			
Code	$DBER$	p_c	$1 - P_c(0, \bar{N}_{BC})$
(15, 7, 2)	3.31e-1	1.11e-1	6.91e-1
(31, 16, 3)	3.15e-1	2.29e-1	6.46e-1
(63, 30, 6)	2.60e-1	2.31e-1	5.54e-1
(127, 64, 10)	2.15e-1	4.27e-1	4.27e-1

Here, the (127, 64, 10) code has the highest value of p_c but leads to the lowest $DBER$ since it has the lowest probability that one or more of the $\bar{N}_{BC} = 1.0$ codeword required to encode an encrypted block of 64 bits will be corrupted. In contrast, the (15, 7, 2) code yields the highest value of $DBER$ (despite the fact the p_c is the lowest of the above codes) because the probability that one or more of the $\bar{N}_{BC} = 10.0$ required codewords (on average) will be corrupted is larger for this code.

Figure 5.16 illustrates the $DBER$ versus BER for ECB for a fixed n

Figure 5.16: ECB mode over BSC with BCH coding ($n = 127$).

($n = 127$) and varying code rate. It can be seen that, as the code rate decreases, the $DBER$ decreases; the lower the value of k for a given n , the better the code is (see also Figure A.4 to Figure A.6 in annex A). There is, however, an interesting exception to the above rule. This occurs for the (127,64,10) code (rate $R = 0.504$) which outperforms the (127,57,11) code (rate $R = 0.449$) for some values of channel BER. It was shown earlier that $\bar{N}_{BC} = 1.0$ for $k = 64$ but becomes $\bar{N}_{BC} = 2.105$ for $k = 57$. This change in k causes a large increase in the probability of one or more of $\bar{N}_{BC}$ codewords being decoded incorrectly. In contrast, p_c decreases only slightly because t increases by only one bit. Hence, the former outweighs the latter and the end result is a net increase in the $DBER$. This is also clear from the following simulation results table:

Table 5.6: Decoding failure probability and probability of one or more of $\bar{N}_{BC}$ codewords being decoded incorrectly over memoryless BSC			
	BER = 3.977e-2	BER = 7.932e-2	BER = 1.01e-1
(127,64,10) <i>DBER</i>	5.97e-3	2.15e-1	3.70e-1
(127,64,10) P_c	1.28e-2	4.27e-1	7.46e-1
(127,64,10) $1 - P_c(0, \bar{N}_{BC})$	1.28e-1	4.27e-1	7.46e-1
(127,57,11) <i>DBER</i>	2.79e-3	2.50e-1	4.30e-1
(127,57,11) P_c	4.89e-3	3.07e-1	6.39e-1
(127,57,11) $1 - P_c(0, \bar{N}_{BC})$	1.03e-2	5.38e-1	8.83e-1

It can be seen that the code which results in the lowest *DBER* for a given channel BER is the one with the lowest probability of decoding failures in the $\bar{N}_{BC}$ codewords.

5.3.3 CBC mode simulation results

The complete results of the simulation runs of CBC mode over binary symmetric channels are shown in tables B.7 to B.9 in annex B, starting at page 248 (see also Figure A.7 to Figure A.9 in annex A). Examination of these results indicate that the behaviour of the *DBER* as a function of the code and channel BER for CBC mode is very similar to that of ECB mode. This is hardly surprising since it was shown earlier that $_{CBC}P_{cb}$ is $_{ECB}P_{cb}$ plus an additional factor. Hence, most of the results and observations made in the previous section apply to CBC mode also. Comparison of the tables for CBC and ECB

mode confirm that CBC is slightly worse (i.e. higher *DBER*) than ECB.

There is, however, one very important difference between CBC and ECB modes. For CBC mode, it is apparent from Figure 5.17 that the (127,57,11) code outperforms the (127,64,10) code for all

Figure 5.17: CBC mode over BSC with BCH coding ($n = 127$).

values of channel BER despite the fact that the required number of codewords $\bar{N}_{BC}$ is identical for CBC and ECB modes. The reason for the difference is probably due to the additional factor in the equation for $_{CBC}P_{cb}$ which ECB mode does not have; recall that

$$_{CBC}P_{cb} = _{ECB}P_{cb} + \frac{\bar{V}_N k P_c(0, \bar{N}_{BC})}{2N} \quad (5-53)$$

where

$$P_c(0, \bar{N}_{BC}) = (1 - p_c)^{\bar{N}_{BC}} \quad (5-54)$$

This term is usually lower for $k = 57$ than for $k = 64$ because the exponent is 2.105 for the former versus 1.0 for the latter. By contrast, the term $\bar{N}_{cb}$ is higher for $k = 57$ than for $k = 64$. Hence, the two terms counteract each other to a certain degree and the advantage of having $\bar{N}_{bc} = 1.0$ when the (127,64,10) code is used is less pronounced for CBC mode than for ECB mode.

5.3.4 CFB mode simulation results

The complete results of the simulation runs of CFB mode over binary symmetric channels are shown in tables B.10 to B.12 in annex B, starting at page 250. See also Figure A.10 to Figure A.12 in annex A. Recall that CFB mode used for these simulations involves a memory buffer of $N = 64$ bits and a word length of $s = 8$ bits. It's *DBER* performance is similar to CBC mode but is marginally superior to it. Hence, most of the conclusions reached earlier for ECB and CBC mode apply here also.

The most significant difference of CFB mode relative to CBC and ECB mode is again related to codes (127,64,10) and (127,57,11). Figure 5.18 shows that the *DBER* for the (127,57,11) is consistently superior to that of the (127,64,10) code for this mode. Recall that for CFB mode, $\bar{N}_{sc}$ is different from $\bar{N}_{bc}$ (for block ciphers) when k is an even number. For CFB mode, $\bar{N}_{sc}$ ranges from 2.105 to 1.875 for $k = 57$ to 64. By contrast, $\bar{N}_{bc}$ changes from 2.105 to 1.0 for ECB and CBC modes. Hence, the effect of going from one code to the other is far less significant for CFB mode in terms of the number of codewords used for an encrypted block of 64 bits.

5.3.5 Comparison of *DBER* for all cipher modes

In this section, we compare the performance of the four cipher modes relative to each other when used over binary symmetric channels. Additional tables and graphs were created, based on the

Figure 5.18: CFB mode over BSC with BCH coding (n = 127).

simulation results mentioned earlier and are listed in annex B, starting at page 251.

The tables B.13 to B.21 compare the *DBER* of ECB, CBC and CFB mode relative to the *DBER* of OFB mode, when the channel BER and BCH code used are identical. These differences are given as percentage differences. The % difference in *DBER* is calculated as follows:

$$\Delta DBER_{mode} = \frac{DBER_{mode} - DBER_{OFB}}{DBER_{OFB}} \times 100 \quad (5-55)$$

where *mode* is either *ECB*, *CBC* or *CFB* as appropriate.

Figure 5.19 to Figure 5.22 indicate that OFB mode outperforms all other cipher modes for all code types and all channel BER values. This is not surprising since it was established in chapter 3 that OFB mode is the only cipher mode which does not cause error propagation.

Figure 5.19: All cipher modes over BSC with no BCH coding.

Figure 5.20: All cipher modes over BSC with BCH code (31,26,1).

Figure 5.21: All cipher modes over BSC with BCH code (31,16,3).

Figure 5.22: All cipher modes over BSC with BCH code (31,11,5).

5.3.5.1 Comparison of *DBER* of ECB, CBC, CFB with OFB mode

The difference in *DBER* is largest when no coding is used. As shown in Figure 5.19, the *DBER* of ECB, CBC and CFB can be up to about 33 times higher than that of OFB. By comparing Figure 5.19 with Figure 5.20 to Figure 5.22, it is clear that the difference in *DBER* decreases significantly when a BCH code is used. For example, the % *DBER* difference between ECB and OFB goes from 1069.33% for no code, to only 244% for a (31,26,1) code when the channel BER is $3.977e-2$. This is simply the result of channel errors being corrected by the BCH codes before decryption is done at the receiver. Since less encrypted blocks at the receiver contain errors, the impact of the inherent error propagation characteristics of the cipher modes is reduced by the use of channel coding.

Two other interesting trends emerge from the examination of the % difference in *DBER*. A few results from the tables of annex B are reproduced below to illustrate these trends (channel BER = $3.977e-2$):

Table 5.7: % <i>DBER</i> difference of ECB, CBC and CFB versus OFB over a memoryless BSC of BER = $3.977e-2$						
Mode	$\Delta DBER$ (15,11,1)	$\Delta DBER$ (15,7,2)	$\Delta DBER$ (15,5,3)	$\Delta DBER$ (31,26,1)	$\Delta DBER$ (31,16,3)	$\Delta DBER$ (31,11,5)
ECB	531.7%	996.4%	1483.9%	244.1%	414.1%	553.6%
CBC	571.0%	1068.2%	1606.2%	266.0%	506.7%	642.6%
CFB	566.0%	1040.6%	1629.6%	256.3%	523.5%	600.1%

One obvious trend is that the $\Delta DBER_{mode}$ increases as the code rate decreases, for a fixed n . This implies that reducing the code rate is not as efficient in reducing the *DBER* for cipher modes ECB, CBC, CFB as it is for cipher mode OFB. For example, at a BER = $3.977e-2$, the *DBER* for ECB mode goes from 0.2688 to 0.08089

as the code is changed from a (15,11,1) to a (15,7,2) code, the latter *DBER* being 3.32 times lower than the former, while for OFB mode under the same conditions, the *DBER* decreases from 0.04255 to 0.007378 (i.e. *DBER* of (15,7,2) code is 5.76 times lower than for the (15,11,1) code). The explanation for this behaviour is again related to the fact that, for ECB, CBC and CFB modes, the *DBER* is a function of the probability that one or more of the $\bar{N}_{BC}$ (or $\bar{N}_{SC}$) codewords required for a block of N encrypted bits are decoded incorrectly. As k decreases, with fixed n , the impact on $1 - P_c(0, \bar{N}_{BC})$ is less beneficial than on the probability of one codeword being decoded incorrectly p_c . This is illustrated by the values of p_c and $1 - P_c(0, \bar{N}_{BC})$ shown below (where the channel BER is $3.977e-2$):

Table 5.8: Decoding failure probability and probability of one or more of $\bar{N}_{BC}$ codewords being decoded incorrectly over a memoryless BSC of BER = $3.977e-2$					
	(15,11,1)		(15,7,2)		(15,5,3)
p_c	0.1180		0.01999		0.002399
Δp_c		-5.903		-8.333	
$1 - P_c(0, \bar{N}_{BC})$	0.5702		0.1828		0.032135
$\Delta (1 - P_c(0, \bar{N}_{BC}))$		-3.119		-5.689	

Note that p_c decreases by a factor of 5.903 as we go from (15,11,1) to (15,7,2) and by a factor of 8.33 as we go from the (15,7,2) code to the (15,5,3) code. Over the same transition range, $1 - P_c(0, \bar{N}_{BC})$ decreases by only 3.119 and 5.689 (as $\bar{N}_{BC}$ increases from 6.727 to 13.6). As a result, decreasing the code rate R is less effective in reducing the *DBER* for ECB, CBC and CFB modes than it is for OFB mode.

Another trend which is apparent from table 5.7 is that, as the value of n increases (for a fixed code rate R), the % difference in $DBER$ decreases. The table below contains results from annex-B which illustrate this trend more clearly (channel BER = $7.895e-2$):

Table 5.9: % $DBER$ difference of ECB, CBC and CFB versus OFB over a memoryless BSC of BER = $7.895e-2$				
Mode	$\Delta DBER$ (15,11,1)	$\Delta DBER$ (31,26,1)	$\Delta DBER$ (63,45,3)	$\Delta DBER$ (127,92,5)
ECB	702.5%	405.0%	226.8%	48.3%
CBC	797.4%	482.0%	285.7%	85.4%
CFB	817.4%	424.0%	250.0%	76.4%

The code rate for the above BCH codes is $R = 0.75$. An examination of the results presented in annex B shows that this trend is consistent for all code rates and all cipher modes. In this case, the reduction in the % $DBER$ as n is increased implies that increasing the block length n is more effective in reducing the $DBER$ for cipher modes ECB, CBC and CFB than it is for cipher mode OFB. In the present case, the increase in k usually results in lower values for $\bar{N}_{BC}$ (or $\bar{N}_{SC}$). The effect on $1 - P_c(0, \bar{N}_{SC})$ is then more beneficial than on the probability of one codeword being decoded incorrectly (p_c). This is well illustrated by the values of p_c and $1 - P_c(0, \bar{N}_{BC})$ shown below, where BER = $3.977e-2$, and the code rate is set at $R = 0.50$:

Table 5.10: Decoding failure probability and probability of one or more of $\bar{N}_{bc}$ codewords being Decoded incorrectly over a memoryless BSC of BER = 3.977e-2					
	(31,16,3)		(63,30,6)		(127,64,10)
p_c	0.03343		0.01241		0.01279
Δp_c		-2.690		+1.03	
<i>DBER OFB</i>	0.01215		0.004629		0.005048
$\Delta DBER$		-2.625		+1.09	
$1 - P_c(0, \bar{N}_{bc})$	0.1271		0.0376		0.01279
$\Delta(1 - P_c(0, \bar{N}_{bc}))$		-3.380		-2.940	
<i>DBER ECB</i>	0.06246		0.01509		0.005967
$\Delta DBER$		-4.139		-2.529	

Note that p_c decreases and increases alternatively as we progress to longer codes while $1 - P_c(0, \bar{N}_{bc})$ consistently decreases under the same conditions (as $\bar{N}_{bc}$ goes from 16.0 to 1.0). Furthermore, the latter always decreases by a larger factor than p_c when they both decrease. Hence, for OFB mode ($\bar{N}_{sc} = 1.0$) an increase in the code length, for a fixed code rate, is not as beneficial as it is for the other cipher modes.

5.3.5.2 Comparison of *DBER* of CBC, CFB with ECB mode

Tables B.22 to B.27 of annex B show the performance of CBC and CFB mode relative to ECB mode. The results in the tables show that the *DBER* of CBC is consistently higher than the *DBER* of ECB for all BCH codes and channel BER.

The % difference in *DBER* when no FEC coding is used is of special interest because it allows us to see how the cipher modes

compare to each other without channel coding. The average $\Delta DBER_{mode}$ is computed for all BCH codes and the results are shown below for CBC and CFB modes versus ECB mode⁷.

Table 5.11: Average % difference in $DBER$ for CBC and CFB modes versus ECB mode for BCH codes									
	none	n=7 k=4 t=1	n=15 k=11 t=1	n=15 k=7 t=2	n=15 k=5 t=3	n=31 k=26 t=1	n=31 k=16 t=3	n=31 k=11 t=5	n=63 k=45 t=3
CBC	1.7%	6.5%	11.6%	5.5%	6.8%	26.1%	12.0%	11.9%	16.8%
CFB	1.9%	5.3%	11.4%	1.0%	5.2%	21.0%	16.7%	5.7%	4.3%
	n=63 k=30 t=6	n=63 k=24 t=7	n=63 k=18 t=10	n=63 k=16 t=11	n=127 k=92 t=5	n=127 k=64 t=10	n=127 k=57 t=11	n=127 k=43 t=14	
CBC	16.4%	15.2%	20.3%	20.7%	16.7%	51.6%	18.1%	21.9%	
CFB	6.7%	7.5%	8.8%	19.3%	11.3%	54.3%	5.6%	6.2%	

It is clear that the difference is very small (below 2%) when no coding is used. Recall that the three modes have a similar value of N (number of bits per block or in the memory buffer, as appropriate). Thus, the three modes can be considered to be virtually identical in performance when no FEC codes are used. The slight additional degradation of CBC and CFB is consistent with theory as seen earlier during the development of expressions for $c_{bc}P_{cb}$. The chaining of adjacent words results in a slightly higher $DBER$ for these two modes than is the case for ECB mode.

However, the results in the table above also indicate that the use of BCH codes tends to magnify significantly the difference in $DBER$ between these modes. Hence, the results imply that BCH codes

⁷ The average of $\Delta DBER$ is calculated as $\frac{1}{n} \sum_{i=1}^n \Delta DBER_i$.

are more efficient at reducing the *DBER* of ECB mode than for CBC or CFB mode for a given channel BER. For example, at a channel BER of 2.064×10^{-3} , the *DBER* of ECB mode goes from 0.05578 (no coding) to 0.002728 (i.e. 20.45 times smaller) when a (31,26,1) BCH code is used while the *DBER* of CBC mode goes from 0.05594 to 0.003638 (15.38 times smaller) and from 0.05616 to 0.003627 (15.48 times smaller) for CFB mode.

The chaining of adjacent encrypted blocks (or words) for CBC and CFB modes makes them more vulnerable to decoding failures of codewords than ECB since the chaining operation results in the successful decryption involving more codewords than ECB mode. For CBC mode, this is reflected in the additional term of the equation for ${}_{CBC}P_{cb}$:

$${}_{CBC}P_{cb} = {}_{ECB}P_{cb} + \frac{\overline{V}_N k P_c(0, \overline{N}_{BC})}{2N} \quad (5-56)$$

For CBC mode, as n increases, with a fixed code rate R , the % difference in *DBER* increases with respect to ECB mode. Also, as the code rate decreases for a fixed n , $\Delta DBER$ also increases (see tables B.22 to B.24 in annex B). These results imply that increasing n (for a fixed code rate) or decreasing the code rate (with a constant blocklength n) is not as efficient in reducing the *DBER* of CBC mode as it is for ECB (since the difference in *DBER* increases when either action is performed).

Examination of the tables in annex B and of the table 5.11 reveal that CFB mode does not exhibit the same, simple trend as CBC, when compared to ECB. In some instances, a decrease in the code rate (fixed n) will result in an increase in the $\Delta DBER$ of CFB relative to ECB mode but in other cases a decrease will result

(e.g. it goes from 5.432 to 4.030 and on to 9.21 as the code is changed from (15,11,1) to (15,7,2) and then to (15,5,3) for a channel BER of $3.977e-2$). The same is true of an increase in n with a fixed code rate R (e.g. $\Delta DBER$ goes from 4.279 to 8.617 and then down to 1.430 as the code is changed from (15,11,1) to (31,26,1) and then to (63,45,3) for a channel BER of $2.019e-2$). Hence, no clear pattern emerges.

5.3.5.3 Comparison of $DBER$ of CFB with CBC mode

Tables B.28 to B.30 in annex B starting at page 260 indicate that the difference in performance between the two modes is relatively small.

Table 5.12: Average % difference in $DBER$ for CFB mode versus CBC mode for all codes								
none	n=7 k=4 t=1	n=15 k=11 t=1	n=15 k=7 t=2	n=15 k=5 t=3	n=31 k=26 t=1	n=31 k=16 t=3	n=31 k=11 t=5	n=63 k=45 t=3
0.3%	-1.1%	1.1%	-4.3%	0.1%	-4.0%	4.1%	-3.3%	-8.5%
n=63 k=30 t=6	n=63 k=24 t=7	n=63 k=18 t=10	n=63 k=16 t=11	n=127 k=92 t=5	n=127 k=64 t=10	n=127 k=57 t=11	n=127 k=43 t=14	
-8.3%	-6.6%	-9.5%	-1.2%	-11.4%	2.3%	-11.6%	-12.8%	

Table 5.12 gives the average % difference in $DBER$ for CFB mode versus CBC mode over all channels, for all codes used. It is apparent that CFB mode is usually superior to CBC as far as $DBER$ is concerned. However, the difference is small. As the code length increases (fixed rate) there appears to be a tendency for the % $DBER$ difference to increase. This indicates that increasing n for a fixed rate code rate R is more efficient in reducing the $DBER$ of CFB mode than it is for CBC mode. However, this trend is weak and there are exceptions when $k = 16$ or 64 . One possible reason for this behaviour is that, as n (and k) increase, a codeword decoding

failure (v errors, $v > t$) can lead to more errors after decryption for CBC mode than for CFB mode. A decoding failure will result in an error cluster (after decoding) of length k bits, with at most $v+t$ bit errors in it. As k increases (fixed code rate), the probability that this cluster will span two adjacent encrypted blocks (for CBC mode) increases, leading to a higher number of bit errors after decryption than for CFB mode (where there are no clearly defined block boundaries). This was illustrated earlier by an example, in section 5.2.1.4. This results, in CBC having a worse average *DBER* than for CFB mode when k increases. However, when k is 16 or 64, there is no possibility of a cluster spanning two blocks, leading to an improved performance for CBC mode. Furthermore, for these values of k , $\bar{N}_{sc} > \bar{N}_{bc}$.

5.3.6 Dependency of *DBER* on the data stream and the DES key

In this section, we examine the dependency between the *DBER* of the four cipher modes on the specific message stream encrypted and on the DES key used. Table B.31 of annex B indicates how the *DBER* of these modes changes when different data sequences (all of them being 97,874 bits long) are transmitted over the same channel, with the DES key remaining constant⁸. Table B.32 illustrates the effect of transmitting the same data sequence over the same channel, but with different DES keys used⁸. For ECB and CBC modes, the original key $K_{old} = A46F107B\ 38DE2C59$ was changed to $K_{new} = A46F127B\ 38DE2C59$ ($d_H(K_{old}, K_{new}) = 1$). For OFB and CFB mode, the original key $K_{old} = A56213F6\ C3D12879$ was changed to $K_{new} = A56213F6\ C3D42879$ ($d_H = 2$). It can be seen that the change in *DBER* is relatively small in all cases.

This change is a result of the confusion-diffusion

⁸ For OFB mode, there was absolutely no change in *DBER*. Hence, that mode was omitted from the table.

characteristics of these cipher modes. In the decryption operation, the inter-dependence between all the bits of the received cipher block ensures that any bit error introduced by the channel will result in different decrypted words M'_m , depending on the specific values of the bits which make up c'_m . Similarly, every bit of the DES key used has an impact on every bit of a decrypted word M'_m (confusion effect) so that two different keys used to decrypt the same corrupted block c'_m may result in a different number of bit errors in M'_m (although the average should still be close to $N/2$). However, OFB mode is a substitution cipher which processes data bits individually and independently of each other. Hence, for that mode, the DBER is completely independent of the data sequence transmitted and of the DES key used.

5.4 Simulated fading channels represented by Gilbert and Fritchman models

In this section, we start the analysis of the simulation results over fading channels, based on the Gilbert and Fritchman models. We restrict our attention to narrow-band mobile, digital, communications channels. The channel BER for such links is typically in the range $[10^{-3}, 10^{-1}]$ [16, 58, 60]. Furthermore, our main interest is to determine how the presence of cryptographic systems such as DES in the modes of operation ECB, CBC, CFB and OFB affects the error rate after decoding and decryption (i.e. the primary objective is not to determine the effectiveness of FEC codes over such a link⁹). Hence, we have limited our simulations to a range of channel BER which is realistic for a mobile communication channel, and which covers a range of conditions (i.e. mobile unit speed, Rayleigh or Rician pdf, etc) which is just wide enough to allow us to determine if (and how) the observations made earlier for

⁹ This has already been done by many researchers (e.g. [2, 15, 18, 23, 28, 42, 49, 51, 71]).

memoryless channels can apply for this type of channel.

5.5 Error propagation for OFB mode over fading channels with BCH coding

Figure 5.23: Theoretical and experimental $_{OFB}P_{cb}$ over Gilbert model channels.

the probability of m errors in n bits $P(m,n)$ is different for fading channels than for a BSC. For channels generated from the Gilbert and Fritchman models, $P(m,n)$ is as given earlier in chapter 4. Figure 5.23 illustrates the theoretical and experimental $_{OFB}P_{cb}$ as a function of channel BER for the Gilbert model (channels OPL, SUB, LOW, MED and HIG, as described in chapter 5 and annex C). The same functions are shown in Figure 5.24 for the Fritchman channel model (channels SU1, OP1, SU2, URB, SUH and HIG, as described in chapter 5 and annex C). In both cases the DBER

In a channel with memory, the probability of an encrypted block c'_m being error-free is not independent of the probability that the preceding block c'_{m-1} is also error-free. However, the DBER of OFB mode has the same form shown as in equation (5-49) (page 140) since the successful decryption of a bit depends only on the codeword which contains it (thus, it is independent of c'_{m-1}). However,

Figure 5.24: Theoretical and experimental $_{OFB}P_{cb}$ for Fritchman model channels.

predicted by equation (5-49) is often higher than the real $DBER$. This may be due in part to the fact that the assumption that $\frac{k}{2}$ bit errors occur when an (n,k,t) codeword is decoded incorrectly is a loose upper bound; the determination of the real number of bit errors is much more involved, as explained in [54, pp. 405].

5.6 Simulation results for the Gilbert channel model

This section describes the procedure followed to determine the Gilbert channel model parameters and presents the simulation results.

5.6.1 Determination of the Gilbert model parameters

The Gilbert model parameters used in this section are based on the experimental results of Lecours et al. [39]. The results are based on measurements of the slow and fast fading characteristics for a wide variety of urban density conditions in the region of Ottawa, Ontario. The channel operating frequency was $f_c = 910$ MHz and the mobile unit speed was about 7.5 km/hr on average. Figure 2.2 (page 20, chapter 2) shows a typical recording of the measured instantaneous power at the receiver for this type of environment.

We will use the experimentally determined value of level crossing rate (LCR) and average fade duration (AFD) to infer the parameters of the Gilbert model. Recall that the LCR and AFD are measured at a pre-determined threshold R_0 . This threshold is customarily selected to be $\eta - \sigma$, η , or $\eta + \sigma$ where η is the expected signal power and σ is its standard deviation. In Figure 2.2, the smooth line superimposed over the bottom figure is the average instantaneous signal power.

The Gilbert model is ideally suited to represent the channel

based on the LCR and AFD since its good state can be used to represent the error-free runs between error clusters while its bad state can simulate error clusters, where the error density within a cluster is controlled by its crossover probability p .

In order to prevent the shallow fades from yielding unrealistic values of AFD and LCR, a threshold of $R_0 = \eta - \sigma$ has been selected so that only deep fade notches of 20 to 40 dB below the mean signal power will be taken into account.

Examination of typical error sequences measured over such channels [16, pp. 60-61] indicate that the BER within a fade notch is typically in the range of 0.10 to 0.5, with the highest rate being associated with the longest and deepest fades. Hence, the Gilbert model parameter p will be selected to achieve this range of error density within an error cluster.

For a mobile binary digital communication channel of bandwidth $BW = 25 \text{ kHz}$, a transmission rate of 10 kbps is a typical value. We will use this data transmission rate to determine the length (in bits) of error-free runs and error-clusters. Based on these considerations, the following table of values is derived:

Table 5.13: Gilbert model parameters for different urban densities					
	Open area (OPL)	Suburban area (SUB)	Low density (LOW)	Med. density (MED)	High density (HIG)
LCR cross./m	1.56	1.65	1.92	2.86	3.07
AFD meters	0.11	0.10	0.09	0.07	0.06
1/LCR s/cross	0.3077	.2909	0.2500	0.1678	0.1564
AFD in ms	52.8	48.0	43.2	33.6	28.8
Run length (bits)	3077	2909	2500	1678	1564
Cluster length (bits)	528	480	432	336	288
P_{gg}	0.999675	0.999656	0.999600	0.999404	0.999361
P_{bb}	0.998110	0.997921	0.997691	0.997033	0.996540
P	0.10	0.20	0.30	0.40	0.50
BER	0.01467	0.02840	0.04430	0.06691	0.07795

The first four columns correspond to a Rician channel whose line-of-sight received signal power component is getting progressively weaker. The column associated with the high urban density corresponds to a Rayleigh fading channel. The values of the LCR and AFD for the Rician channels are experimental values. For the Rayleigh channel, the LCR (in crossings per meter) was extrapolated from Lecours' experimental results and is higher than theory would predict (theoretical LCR is 2.69). The AFD for the Rayleigh is the theoretical value while it is the experimental

value for the Rician channels¹⁰

In order to determine the average length of error-free runs and error clusters, the LCR was converted from crossings per meters to time between crossings, based on a mobile unit speed of 7.5 km/hr (2.083 meters/second). For example, 1.56 crossings/meter is equivalent to $2.083 \times 1.56 = 3.24948$ crossings/second or 0.3077 seconds between crossings. Similarly, an AFD of 0.11 meters is equivalent to a $0.11/2.083 = 52.28$ ms fade duration. Based on a 10 kbps data transmission rate, we find that there are, for instance, $0.3077 \times 10^3 = 3077$ error-free bits between error clusters for the open area urban density (OPL). Similarly, an error cluster would last $52.8 \text{ ms} \times 10^3$ or 528 bit periods in that channel.

To determine the values of P_{gg} and P_{bb} , we use the fact that, for a two-state, homogeneous, stationary Markov chain, the average amount of time (in bit periods) spent in a given state, given that the chain starts in that state, is equal to [52]:

$$E[A_G|X_0 = G] = \frac{P_{gg}}{1 - P_{gg}}, \quad E[A_B|X_0 = B] = \frac{P_{bb}}{1 - P_{bb}} \quad (5-57)$$

where $E[A_G|X_0 = G]$ is the expected amount of time spent in the good state 'G' (in number of bit periods) given that the initial state is 'G'. For example, when the urban density is that of an open area (OPL), we want the Gilbert model to spend, on average, 3077 bit periods in the good state when it enters it. Using that value as the average persistence of the state, we find that $P_{gg} = 3077/3078 = 0.999675$ by solving the above equation. Similarly, the average error cluster length for an open area (OPL) results in $P_{bb} = 528/529 = 0.998110$.

¹⁰ Lecours found the theoretical values to be close to the experimental values for the AFD but noted that the experimental values of the LCR were consistently higher than those predicted by theory.

The selection of the error cluster density (i.e. parameter p of the Gilbert model) is based on the following arguments. For a strongly Rician distributed channel (i.e. with large line-of-sight to multipath signal ratio), the fades are usually shallow. It was seen in chapter 2 that, for a Rician channel, the deviation between the peak and notch power is usually around 10 dB. Hence, the BER within a fade notch should be much lower than for a Rayleigh channel where the deviation between the peak and notch power is often around 40 dB. Thus, we associate the lowest error cluster density with the open area and the densest one with the Rayleigh channel (high urban density).

The above model parameters are used to simulate different urban density channel conditions. To determine the effect of a speed increase, we propose to simulate a doubling of the mobile unit speed from 7.5 km/hr to 15 km/hr and from 15 km/hr to 30 km/hr for the open area model. This results in doubling the LCR and halving the AFD each time the speed is doubled. The error cluster density will be increased from 0.10 to 0.20 and then to 0.30 for each speed increase, to simulate the expected increase in error density as a result of the increased irreducible error rate caused by random FM. For these extra simulations, the Gilbert model parameters can be shown to be:

Table 5.14: Gilbert model parameters for varying mobile unit speed		
Parameter	Speed = 15 km/hr (OPM)	Speed = 30 km/hr (OPH)
Error-free run length	1539 bits	770 bits
Error cluster length	264 bits	132 bits
P_{00}	0.999351	0.998703
P_{11}	0.996226	0.992381
p	0.20	0.30
Theoretical channel BER	0.02935	0.04413

Annex C illustrates the autocorrelation functions for all these channels. As expected, channels of high urban density have more memory (i.e. denser clusters) than those with a lower density. Channels corresponding to a high unit speed initially have more memory than those for lower speeds (because of denser clusters) but the curve for the former decreases more rapidly than for the latter (because clusters are shorter). The more or less periodic recurrence of error clusters is also evident from the graph of the long-term autocorrelation function for channel OPH shown in the same annex (i.e. periodic peaks and troughs in the curve). The simulation results for the Gilbert model are shown in annex B, starting at table B.33 (page 264).

5.6.2 Performance of binary BCH codes

The results of the simulations indicate clearly that the use of binary BCH codes in a fading channel often results in a *DBER* which is higher than if no coding had been used. This is well illustrated by the figures on the following pages which show the

Figure 5.25: OFB mode over Gilbert model channels ($R = 0.75$)

Figure 5.26: OFB mode over Gilbert model channels ($R = 0.50$).

Figure 5.27: OFB mode over Gilbert model channels ($R = 0.33$).

Figure 5.28: CBC mode over Gilbert model channels ($R = 0.75$).

Figure 5.29: CBC mode over Gilbert model channels ($R = 0.50$).

Figure 5.30: CBC mode over Gilbert model channels ($R = 0.33$).

performance of codes of similar rate and varying length n , for OFB and CBC cipher modes (best and worst $DBER$ performance respectively). By contrast, it was seen earlier that these codes, when used over a memoryless channel over the same BER range, often provided a significant improvement in the $DBER$.

The difference is even more apparent for ECB, CBC and CFB modes. For the memoryless BSC, it was seen earlier that the use of any binary BCH code at any channel BER always resulted in a reduction in the $DBER$ at the receiver. This is seldom true for the fading channels represented by the Gilbert model. The only exception occurs at the lowest BER channels (OPL, SUB) and for the codes with the lowest rates.

The reason for this poor performance is simply that binary BCH codes are not well suited for operation in fading channels. That is because the codewords which coincide with dense error clusters have a high probability of decoding failure p_c . This is clear from the Gilbert model parameter p which controls the error density within a cluster. When the error density exceeds 0.2, ($p > 0.2$) none of the codes used in the simulations yield a reduction in $DBER$. This is consistent with the results of table 5.2 (page 145), which indicates the highest BER (for a memoryless channel) at which a given (n, k, t) code will yield a $P_{cb} \leq P_b$ ¹¹. Hence, the binary BCH codes are mostly useless in fading channels characterized by dense error clusters followed by error-free periods since they are overwhelmed within the clusters (i.e. frequent decoding failures leading to $P_{cb} > P_b$) and serve no function during error-free runs.

¹¹ Although the Gilbert model represents a fading channel, the distribution of errors within a cluster is uniform (i.e. just like a memoryless channel).

5.6.3 Error propagation characteristics for all modes

The simulation results indicate that, under similar coding and channel conditions, OFB mode has the lowest *DBER* of all modes, followed by ECB, CFB and CBC in that order. This is identical to the earlier findings for memoryless BSC and indicates that the error propagation characteristics of these modes do not depend only on the channel characteristics. The figures on the following pages illustrate the *DBER* of the four modes for various coding conditions.

Examination of the simulation results for ECB, CBC and CFB modes also reveals that the cryptographic degradation $D = \frac{DBER}{BER}$ is significantly lower for fading channels than it is for memoryless links of corresponding BER. This is true despite the fact that binary BCH codes are less efficient in a fading channel than in a BSC, as explained earlier. For example, for ECB mode where no coding is used, the ratio $D = \frac{DBER}{BER}$ varies between 6.063 and 1.364 for the fading channels while it varies between 18.17 and 6.294 for binary symmetric channels over the same BER range. The situation is similar when coding is used. For example, when data is encoded with a (31,26,1) code, ECB mode over a BSC channel with $BER = 0.03977$ results in a $DBER = 0.352$ ($D = 8.851$) while, for the Gilbert model channel of $BER = 0.04487$ (OPH), the $DBER$ is 0.1357 ($D = 3.024$).

Figure 5.35 and Figure 5.36 provide a direct comparison of the severity of error propagation for BSC and fading channels of corresponding BER. The reason for that apparent performance gain in the fading link is that the clustering of channel errors results in less encrypted blocks being corrupted than in a memoryless BSC of identical BER. Since bit errors occur close to each other in time, a few encrypted blocks end up corrupted with many bit errors.

Figure 5.31: All modes over Gilbert model channels (no coding)

Figure 5.32: All modes over Gilbert model channels, (31,26,1) code

Figure 5.33: All modes over Gilbert model channels, (31,16,3) code

Figure 5.34: All modes over Gilbert model channels, (31,11,5) code

Figure 5.35: OFB mode over BSC and Gilbert model channels

Figure 5.36: CBC mode over BSC and Gilbert model channels

However, it was shown earlier that a corrupted block will produce $N/2$ bit errors after decryption regardless of the number of bit errors in the block. Hence, the numerous bit errors in a corrupted encrypted block do not result in a corresponding increase in the number of bit errors after decryption.

5.6.4 Avalanche effect of BCH codes

For memoryless BSC, there was no evidence of the avalanche effect of BCH codes (i.e. $P_{cb} > P_b$) for ECB, CBC and CFB modes. This is no longer true for the Gilbert model channels. There are two reasons for this:

a) the clustering of errors in a fading channel affects less encrypted blocks, leading to a less severe error propagation effect. Hence, the avalanche effect of a code is less likely to be masked by a high $DBER$.

b) The high error density within a cluster for a fading channel leads to a high decoding failure probability p_c for all codes. As a result, the avalanche effect of a code is much more pronounced than for a memoryless BSC of similar BER.

5.6.5 Relationship between $DBER$ and BER

For ECB, CBC and CFB modes, Figure 5.31 and Figure 5.32 depict an unusual behaviour; as the BER increases (i.e. urban density increases), the $DBER$ initially decreases before it increases again. For example, for CBC mode, as the BER goes from 0.01395 to 0.03085 and to 0.04789, the $DBER$ (no coding) progresses from 0.0877 to 0.07355 and on to 0.08664. Note that the last $DBER$ is lower than the first one, despite the fact that the last BER is roughly 3.4 times larger than the first one. The same behaviour is evident for BCH codes with a high code rate.

As the urban density increases, three factors come into play;

a) the probability of a decoding failure of one or more of $\bar{N}_{nc}$ (or $\bar{N}_{sc}$) codewords (required to encode a block of N encrypted bits) decreases because the cluster length decreases,

b) the probability of a decoding failure of one or more of $\bar{N}_{nc}$ (or $\bar{N}_{sc}$) codewords increases because error clusters occur more frequently (because error-free runs are shorter), and

c) there is almost no increase in the probability of a decoding failure for one codeword p_c within a cluster, even though the error density in a cluster increases.

The third factor comes about because the p_c is already very high for most codes when the parameter p of the Gilbert model is 0.1. Hence, it does not increase significantly as p is increased up to 0.5. This means that the other two factors control the $DBER$ to a large extent. The effect these have on the $DBER$ is best illustrated by the following example.

Consider the case where no coding is used. For 100,000 data bits transmitted over these channels, the following table of values can be constructed:

Table 5.15: Effect of cluster length on blocks of encrypted bits for different urban density channels					
	Open (OPL)	Suburban (SUB)	Low density (LOW)	Medium density (MED)	High density (HIG)
Cluster length	528	480	432	336	288
# Clusters in 100,000 bits	32.5	34.37	40.0	59.595	63.94
# Blocks of $N = 64$ bits affected	8.25	7.5	6.75	5.25	4.5
Max. <i>DBER</i>	0.08580	0.08252	0.0864	0.1001	0.09207
Simulation <i>DBER</i> for ECB	0.08464	0.07185	0.08293	0.1214	0.112

For instance, an open area channel file of 100,000 bits should have 32.5 clusters of about 528 bits each, on average. One cluster will span about 8.25 blocks of 64 encrypted bits. Assuming that each such block is corrupted with probability 1.0 and that each corrupted block produces about 32 bit errors after decryption, the maximum theoretical *DBER* is then

$$\frac{32.5 \times 8.25 \times 32}{100000} = 0.08580$$

It can be seen from the above table that the measured *DBER* from the simulations increases and decreases in step with the calculated *DBER* based on the three assumptions made earlier.

Hence, it seems that, as urban density increases, there may be a resultant decrease in the *DBER* even though the channel BER increases. This will occur if the reduction in the probability of an encrypted block coinciding with a cluster (because of shorter clusters) is larger than the increase in that probability because

of the increased frequency of occurrence of these clusters.

5.6.6 BCH code performance versus blocklength

The computer simulation results indicate that, for a fixed code rate R , short codes almost always outperform longer ones for OFB mode. This result is identical to that of memoryless BSC over the same channel BER range.

The same results are apparent for the other modes. However, this is much more consistent in the Gilbert channel than was the case for the BSC. For the latter, there were many instances where the $DBER$ would be lower as the BER decreased towards 10^{-3} . This rarely occurs in the Gilbert channel because a reduction in the code rate only yields a very small decrease in the decoding failure p_c , since error clusters are usually very dense. This decrease in p_c is more than offset by the increase in the number of codewords which suffer a decoding failure (i.e. a code rate reduction means that more codewords must be transmitted per encrypted block and, as a result, more of them will be affected by error clusters).

5.6.7 BCH code performance versus code rate

For OFB mode, the $DBER$ usually decreases as the code rate R decreases for a fixed n . But there are exceptions to this rule, in contrast with memoryless BSC where this was consistently true. For example, the $DBER$ increases from 0.06946 to 0.07112 and decreases to 0.05024 (channel LOW) as the code is changed from (15,11,1) to (15,7,2) and to (15,5,3). This is also caused by the poor performance of binary BCH codes over fading channels. As the code rate decreases, the modest decrease in p_c is sometimes more than offset by the corresponding increase in the number of codewords which suffer a decoding failure (more codewords required to transmit the same quantity of data). The same behaviour is apparent

for the other cipher modes, but it is much more common than for OFB mode.

As was the case for the memoryless BSC, it is again apparent that the $DBER$ of ECB mode for the (127,64,10) code is lower than for the (127,57,11) code. Furthermore, the effect of having $\bar{N}_{bc} = 1.0$ is not as important for CBC than it is for ECB mode. For CFB mode, the impact of changing k from 57 to 64 is even less important ($\bar{N}_{sc} \neq \bar{N}_{bc}$). This is also consistent with the results found for the memoryless BSC. For example, in the SUB channel, ($BER = 0.03085$), the transition from the (127,64,10) code to the (127,57,11) code results in an 18.5% increase in the $DBER$ for ECB mode, versus increases of 13% and 2.77% for CBC and CFB modes respectively. For OFB under the same conditions, the $DBER$ falls by 7.29%.

5.6.8 Comparison of $DBER$ of ECB, CBC and CFB with OFB mode

The % difference in $DBER$ for the ECB, CBC and CFB modes with OFB mode is significantly smaller for the fading channels (Gilbert model) than that of memoryless BSC of corresponding BER. For a BSC with no coding, the difference (for ECB mode) varies from 1756% to 389% over the BER range $[10^{-2}, 10^{-1}]$. Over the same range, the difference for the fading channels goes from 426% to 19%. This is again caused by the clustering of errors in the fading links which, by affecting less encrypted blocks than a BSC of similar BER, results in a less severe error propagation for these three modes.

The results for the Gilbert model are very consistent with those for the binary symmetric channel. Examination of tables B.45 to B.53 inclusive leads to the following, similar observations:

- a) the % difference ($\Delta DBER$) in $DBER$ increases as the code rate R decreases (for a fixed blocklength n) for all Gilbert

model channels, all four cipher modes and most BCH codes considered (exception is discussed below).

b) the $\Delta DBER$ decreases as n increases (for a fixed code rate R). This is observed for all channels, all cipher modes and all codes, with a few exceptions.

The only consistent exception to the first observation occurs for ECB mode, for the transition from the (127,92,5) code to the (127,64,10); the % difference decreases for all channels. This is due to the fact that the latter code yields a significant improvement in the $DBER$ for ECB mode ($\bar{N}_{bc} = 1.0$) but has no special effect on OFB mode (whose $\bar{N}_{sc} = 1.0$ for all codes).

For the Gilbert model simulations, exceptions to the second observation occur mostly in the case of the high urban density channel (HIG, BER = 0.08209); in some cases, the % difference in $DBER$ increases as the code length increases. Since HIG is the noisiest channel, this may be caused by the fact that the increase in blocklength n (and hence k) results in an increase in the $DBER$ (avalanche effect of the code) which is more important than the corresponding decrease in the $DBER$ because of the smaller number of codewords ($\bar{N}_{bc}$ or $\bar{N}_{sc}$) required to encode an encrypted block.

5.6.9 Comparison of $DBER$ of CBC and CFB with ECB mode

Tables B.54 to B.59 in annex B compare the performance of the CBC, CFB and ECB modes of DES. Again, the observations made for fading channels are very consistent with those made earlier for memoryless BSC:

a) the $DBER$ of CBC and CFB modes are consistently higher than for ECB mode,

b) the use of BCH codes tends to magnify these differences but not as much as for the case of a binary symmetric channel,

c) the chaining of adjacent encrypted blocks for CBC mode results in BCH codes being less effective in reducing the $DBER$ than is the case for to ECB mode. Again, this trend is less apparent for CFB mode.

The table below shows the average % difference in $DBER$ over all channels for the different BCH codes used:

Table 5.16: Average % difference in $DBER$ for CBC and CFB modes versus ECB mode for all codes									
	none	n=7 k=4 t=1	n=15 k=11 t=1	n=15 k=7 t=2	n=15 k=5 t=3	n=31 k=26 t=1	n=31 k=16 t=3	n=31 k=11 t=5	n=63 k=45 t=3
CBC	5.0%	7.8%	8.7%	9.4%	9.5%	7.8%	12.8%	12.3%	8.8%
CFB	1.5%	1.9%	2.5%	1.5%	4.1%	1.2%	6.7%	3.1%	1.3%
	n=63 k=30 t=6	n=63 k=24 t=7	n=63 k=18 t=10	n=63 k=16 t=11	n=127 k=92 t=5	n=127 k=64 t=10	n=127 k=57 t=11	n=127 k=43 t=14	
CBC	13.0%	14.3%	15.7%	21.0%	8.9%	24.8%	13.6%	15.6%	
CFB	2.5%	4.5%	5.0%	12.8%	1.0%	26.6%	1.5%	3.2%	

The fact that binary BCH codes do not magnify the $\Delta DBER$ as much as was the case for BSC is easily explained by the fact that these codes are not as efficient at correcting errors in fading channels as they are in memoryless ones.

The impact of $k = 64$ on the $\Delta DBER_{CFB}$ of CFB relative to ECB is again due to the fact that $\bar{N}_{BC}$ is much smaller than $\bar{N}_{SC}$ for this value of k .

5.6.10 Comparison of *DBER* of CFB with CBC mode

Tables B.60 to B.62 from annex B indicate the % difference in *DBER* ($\Delta DBER$) for CFB mode relative to CBC mode for all codes and channels. The table below indicates the average % difference in *DBER* for CFB mode relative to CBC mode, calculated over all channels.

Table 5.17: Average % difference in <i>DBER</i> for CFB mode versus CBC mode for all codes								
none	n=7 k=4 t=1	n=15 k=11 t=1	n=15 k=7 t=2	n=15 k=5 t=3	n=31 k=26 t=1	n=31 k=16 t=3	n=31 k=11 t=5	n=63 k=45 t=3
-3.2%	-5.3%	-5.7%	-7.1%	-4.9%	-5.9%	-5.5%	-8.1%	-6.8%
n=63 k=30 t=6	n=63 k=24 t=7	n=63 k=18 t=10	n=63 k=16 t=11	n=127 k=92 t=5	n=127 k=64 t=10	n=127 k=57 t=11	n=127 k=43 t=14	
-9.2%	-8.6%	-9.2%	-6.8%	-7.2%	1.4%	-10.5%	-10.7%	

CFB mode is marginally but consistently superior to CBC modes for all channels and all codes except the (127,64,10) code, where it has a higher *DBER* than CBC mode. Again, there is a tendency for the $\Delta DBER$ to increase as n increases (fixed rate) from (n_1, k_1, t_1) to (n_2, k_2, t_2) , except when $k_2 = 16$ or 64. This is consistent with the results obtained earlier for the memoryless BSC.

5.6.11 Effect of a speed increase

The simulation results indicate that the observations made earlier for the different urban channel densities are equally applicable to these channels.

The decrease in the *DBER* (no coding) for ECB, CBC and CFB from OPL to OPM ($BER = 0.01395$ to 0.02663) is consistent with

observations made earlier about the net impact of a reduction in the cluster length and the corresponding increase in the number of clusters in a sequence of a given length (in bits). Figure A.13 to Figure A.18 in annex A (starting at page 239) illustrate some results over Gilbert model channels of different mobile unit speeds.

5.7 Simulation results for the Fritchman channel model

This section describes the model parameters used and presents the simulation results.

5.7.1 Determination of the Fritchman model parameters

Simulations based on the Fritchman model were performed over 9 channels, covering a wide range of urban density and unit speed. The model parameters are based on measurements made in real channels. The parameters used in this work were taken from the work of Assia Semmar [61], whose channel measurements were carried out in the area of Quebec city.

Five of the nine models used were selected to represent a gradual increase in urban density with other factors being constant. The other four represent the effect of a speed increase. The parameters for the channel models and the corresponding conditions (e.g. urban density, speed) under which the experimental tests were carried out are given in annex C. As much as possible, the channels were selected to ensure that the average signal power was the same for each group of channels. In the simulation results given in annex B (tables B.63 to B.92), the channels are labelled as follows:

```
SU1 == Suburban Area (BER = 3.9875e-3)
OP1 == Open Area (BER = 8.97e-3)
SU2 == Suburban Area (BER = 1.457e-2)
URB == Urban Area (BER = 2.588e-2)
HIG == High Density Urban Area (BER = 1.605e-2)
SUL == Suburban Area, Speed = 80 km/hr (BER = 2.919e-2)
SUH == Suburban Area, Speed = 90 km/hr (BER = 8.92e-2)
OPL == Open Area, Speed = 50 km/hr (BER = 1.286e-2)
OPH == Open Area, Speed = 100 km/hr (BER = 3.585e-2)
```

5.7.2 Fritchman's model versus Gilbert's model

There is no simple procedure to determine the average cluster length and the average error-free run length for the Fritchman model. Error cluster length and density are jointly determined by the N -th state and its transition probabilities to the good states. Rather than attempt to derive approximate equations based on the model parameters, it was decided to calculate the average cluster and run lengths based on the channel files generated from the Fritchman model (each file is 500,000 bits long). This is accomplished by a program which utilizes these two conditions to determine when a cluster ends:

- a) A minimum BER (or error density) within a cluster. If the BER falls below that value, the cluster is deemed to have ended at the last '1' in the sequence.
- b) A maximum number of consecutive '0's allowed within a cluster. If that maximum is exceeded, the cluster is deemed to have ended at the last '1' in the sequence (regardless of the BER within the cluster when this limit is exceeded).

Selecting the values of these parameters is somewhat arbitrary. Nevertheless, the threshold BER was set at 0.1 and the maximum number of consecutive '0's allowed in a cluster was set at 80 because it was felt that these are reasonable values for

channels which usually exhibit a considerable degree of memory. The results from the analysis of the channel files are shown in the table below:

Table 5.18: Measured cluster length and error-free run length for Fritchman model channels					
Channel label	# of runs in file	Av. run length	Standard deviation	Av.clus. length	Standard deviation
SU1	503	788	794	7	10
OP1	522	730	831	37	52
SU2	800	468	428	19	26
URB	800	366	336	38	53
HIG	732	242	195	305	355
SUL	305	1198	1168	114	139
SUH	638	340	402	287	391
OPL	593	640	717	34	47
OPH	800	246	266	50	72

The number of clusters in each file is either one more or one less than the number of runs¹². The wide standard deviation (in bits) should not be surprising, given the random nature of the process which generates these files. To allow a comparison with the Gilbert model channels used earlier, the theoretical and calculated values of the cluster length and error-free runs for the latter are shown in the following table:

¹² An error-free run is a sequence of consecutive '0's between two error clusters and should not be confused with an error-free gap $P(0^n|1)$ as defined in chapter 2.

Table 5.19: Measured (and theoretical) cluster length and error-free run length for Gilbert model channels					
Channel label	# of Runs in file	Av. run length	Standard deviation	Av.clus. length	Standard deviation
OPL	588	657 (3077)	1891	82 (528)	220
SUB	159	2328 (2909)	2441	416 (480)	455
LOW	166	2201 (2500)	2192	428 (432)	448
MED	226	1586 (1678)	1533	344 (336)	328
HIG	218	1532 (1564)	1413	303 (288)	296
OPM	274	1380 (1539)	1424	212 (264)	266
OPH	473	789 (770)	783	133 (132)	134

The numbers in parenthesis are the theoretical values derived earlier and calculated from the Gilbert model parameters. These are generally in agreement with the measured values, except for the OPL channel where there is a significant difference between the two. This difference is probably due to the low error density for that channel ($p = 0.1$). This causes the program (with a threshold of 0.1) to underestimate the cluster length, leading to a low average value (i.e. there are many clusters of length = 1).

For the Fritchman model channels used, it is apparent that the error-free run length decreases as the urban density increases (just like the Gilbert model) but the error cluster length sometimes increases as the urban density increases (unlike the Gilbert model). The same behaviour is apparent for the case of a speed increase (fixed urban density). Furthermore, the run and cluster lengths for the Fritchman models are significantly lower than those of the Gilbert model channels.

Because of the wide variety of uncontrolled factors which affect the channels used and on which the Fritchman models are based, it is impossible to make generalizations based on these few results. Nevertheless, it can be said that the lower cluster and run lengths for the Fritchman model channels are probably due to the fact that clusters are not as clearly defined for this model as they are for the Gilbert model (with only two states). For the Fritchman model, the presence of two or more good states leads to a wider range of cluster lengths than for the Gilbert model. Furthermore, the error clusters generated by the Fritchman model are usually less dense than those of a corresponding Gilbert model.

The apparent increase in cluster length as speed or urban density increases (for the Fritchman model) may be due in part to a corresponding increase in the error cluster density in conjunction with a higher frequency of occurrence of short error clusters. For a cluster of a given length, an increase in the cluster density would result in the program (used to determine the cluster length) requiring more consecutive '0's to fall below the threshold. This increases the probability that clusters close to each other may be merged together, thereby increasing the average cluster length.

Annex C illustrates the autocorrelation functions for the Fritchman model channels used in the simulation runs. The long-term autocorrelation function is also shown for the SU2 and OP1 channel files. The peaks for these functions are very narrow and much lower than for the Gilbert channel files shown in the same annex. This indicates clearly that clusters are not as well defined and do not occur as regularly for the Fritchman model channels as they do for a channel based on the Gilbert model. Furthermore, the channels based on the Fritchman model generally exhibit less memory than Gilbert model channels of corresponding BER, indicating that the latter gives an estimate of the mobile links which is too pessimistic. Again, this is probably due to the fact that the

clusters are too well defined for the Gilbert model. Since the Fritchman model parameters used in this thesis are more closely related to real channels than the Gilbert model parameters, we consider the former to be more representative of the real distribution of error clusters in a digital mobile communication link.

The computer simulation results based on the Fritchman model are shown in annex B, in tables B.63 to B.98 inclusive.

5.7.3 Performance of binary BCH codes

As was the case for the Gilbert model simulations, it is clear that binary BCH codes do not perform as well in fading links as they do in memoryless channels. However, the code performance is not as poor as was the case for the Gilbert model channels of corresponding BER. This is due to the fact that the channels simulated by the Fritchman models exhibit less memory (i.e. shorter, less dense error clusters) than those simulated by the Gilbert model, leading to less frequent decoding failures.

The reduction in *DBER* achieved by the use of these codes is more evident for ECB, CBC and CFB modes. This can be seen from figures of the following pages (Figure 5.37 to Figure 5.42) which illustrate the *DBER* for OFB and CBC modes, for the code rates 0.75, 0.5 and 0.33.

It can be noted that the avalanche effect of the binary BCH codes (i.e. $P_{cb} > P_b$) is again evident for the Fritchman model channels, as was the case for the Gilbert model. However, it is much less pronounced because error clusters are less dense and generally shorter.

Figure 5.37: OFB mode over Fritchman model channels (R = 0.75)

Figure 5.38: OFB mode over Fritchman model channels (R = 0.50)

Figure 5.39: OFB mode over Fritchman model channels ($R = 0.33$)

Figure 5.40: CBC mode over Fritchman model channels ($R = 0.75$)

Figure 5.41: CBC mode over Fritchman mode channels ($R = 0.50$)

Figure 5.42: CBC mode over Fritchman model channels ($R = 0.33$)

5.7.4 Error propagation for the four DES modes

The simulation results indicate that OFB mode has the lowest *DBER* of the four DES modes of operation, followed by ECB, CFB and CBC in that order (under similar coding and channel conditions). This is identical to the earlier findings for Gilbert model channels and memoryless BSC.

However, the ratio $D = \frac{DBER}{BER}$ is higher than for the Gilbert model channels but still lower than for the memoryless BSC of similar BER (Figure 5.43). For example, the ratio varies from 6.79 to 2.1 over the BER range $[10^{-2}, 10^{-1}]$ for CBC mode, versus 6.2 to 1.1 for the Gilbert model channels over the same range and 18.4 to 4.9

Figure 5.43: CBC mode over BSC and fading channels, no coding

for BSC. It is higher for the Fritchman model than for the Gilbert

model; Fritchman's model channel experience more small clusters than the Gilbert's model channels, thereby affecting more encrypted blocks. Memoryless BSC have the highest ratio because channel errors are uniformly distributed, affecting even more encrypted blocks. Figure 5.44 to Figure 5.47 compares the *DBER* of the four modes under various coding conditions for the Fritchman model channels.

5.7.5 *DBER* versus BER for Fritchman's channels

For the Fritchman model channels, Figure 5.38 to Figure 5.42 (starting at page 197) illustrate an unusual behaviour for codes of medium to low rate (0.5 to 0.33). As the BER increases from $8.97\text{e-}3$ to $1.457\text{e-}2$ (OP1 to SU2), there is a significant drop in the *DBER*. When the BER is further increased to $2.588\text{e-}2$ (URB channel), the *DBER* increases normally again (see also Figure A.19 and Figure A.20 in annex A, starting at page 242). This is probably due to the unusually short average cluster length of the SU2 channel (19, versus 37 for OP1 and 38 for URB). This low average value indicates that the channel file for SU2 contains a lot of short clusters of low error density. This would result in a significant decrease in the probability of a decoding failure p_c for long codes with a relatively large value of t . This is supported by the autocorrelation function shown in annex C for these two channels; OP1 displays sharper and wider peaks, indicating that its error clusters are more dense and better defined than those of SU2. For high rate codes (i.e. $R = 0.75$) or short codes of lower rate, p_c does not decrease significantly as a result of the shorter clusters. Hence, the *DBER* versus BER curves for these codes show a normal, gradual increase in *DBER* as the BER increases.

Figure 5.44: All modes over Fritchman model channels, no coding

Figure 5.45: All modes over Fritchman model channels, (31,26,1) code

Figure 5.46: All modes over Fritchman model channels, (31,16,3) code

Figure 5.47: All modes over Fritchman model channels, (31,11,5) code

5.7.6 Code performance versus blocklength n

The simulation results also show that, for a fixed code rate R , codes having a small blocklength n outperform longer ones for OFB mode beyond a given BER. This result is identical to that of BSC over the same channel BER range.

However, this is not usually true for ECB, CBC and CFB modes. More often than not (for a given code rate R), long codes outperform shorter ones over the entire BER range, with a few exceptions. This behaviour is similar to that seen earlier for memoryless BSC but is different than for Gilbert model channels, where short codes almost always outperformed longer ones for all modes. The difference is probably due to the fact that binary BCH codes perform better on the Fritchman model channels than the Gilbert model channels used in the simulations; the former having less memory than the latter, codewords do not suffer decoding failures as often. Hence, the avalanche effect of BCH codes is less common.

5.7.7 Code performance versus code rate

For OFB mode, the simulation results indicate that a reduction in the code rate for a fixed blocklength n consistently results in a reduced $DBER$. This is in agreement with the findings for memoryless BSC but in contrast with the Gilbert model channels, where a decrease in R often resulted in an increase in the $DBER$. For the Gilbert model, this was caused by the very dense and well defined clusters which resulted in a marginal decrease in p_c , which was more than offset by the increased number of corrupted codewords of lower rate.

For ECB, CBC and CFB modes, a reduction in the code rate (fixed n) will often result in an increase in the $DBER$ for short

codes ($n \leq 31$) but not for longer codes, where the reduction in R usually results in a decrease in the $DBER$. One exception to this is the HIG channel, where the high $BER = 0.1605$ almost always results in an increase in the $DBER$ when the code rate is decreased. Recall that for the Gilbert model channels, it was almost always invariably observed for the BCH codes and the channels under consideration that a decrease in R would result in an increased $DBER$.

For short codes, the relatively small error correcting capability ($t \leq 5$) results in only a modest decrease in the probability of a decoding failure p_c as the code rate is decreased because the error clusters are often longer than the code length (see table 5.18). This decrease in p_c is usually more than offset by the increase in $\bar{N}_{bc}$ (or $\bar{N}_{sc}$) and the resultant increase in the probability that one or more of these codewords will be decoded incorrectly. For longer codes, where n is often larger than the cluster length, the higher values of t obtained as R decreases makes it more likely that a codeword coinciding with an error cluster will not suffer a decoding failure. However, in the case of the HIG channel ($BER = 0.1605$ and an average cluster length of 305 bits), t and n are still not high enough to lead to a sufficient decrease in p_c to compensate for the increase in $\bar{N}_{bc}$ (or $\bar{N}_{sc}$) and the resultant increase in the probability that one or more of these codewords will be decoded incorrectly.

As was the case for the memoryless BSC and the Gilbert model channels, changing the code from (127,64,10) to (127,57,11) has a significant negative impact on the $DBER$ of ECB and CBC modes. For example, in the urban channel (URB), going from $k = 64$ to $k = 57$ results in a 13.7% decrease in the $DBER$ for OFB, a 45.7% increase for ECB, a 13.4% increase for CBC and a 7.3% decrease for CFB mode. As before, it is clear that the impact is most severe for ECB mode, followed by CBC. Both OFB and CFB behave normally in this case

(i.e. $DBER$ decreases as R decreases for fixed n).

5.7.8 Comparison of $DBER$ of ECB, CBC and CFB with OFB mode

The % difference in $DBER$ ($\Delta DBER$) for these modes relative to OFB mode is higher for the Fritchman model channels than that of the Gilbert model channels but is still lower than for memoryless BSC of corresponding BER. For a BSC with no coding, the difference varies from 1756% to 389% over the BER range $[10^{-2}, 10^{-1}]$ (for the ECB mode). Over the same range, the difference for the Gilbert model channels goes from 426% to 19% as opposed to 456% to 107% for Fritchman model channels. These results are consistent with earlier observations to the effect that error clusters are more dense and better defined for the Gilbert model channels than they are for Fritchman model channels, leading to less error propagation as fewer encrypted blocks are affected.

The results for the Fritchman model channels are consistent with the observations made for the memoryless BSC and the Gilbert model channels:

a) for a fixed n the $\Delta DBER$ increases as the code rate R decreases for all Fritchman model channels, all cipher modes and most codes (except for the transition from the (127,92,5) code to the (127,64,10) code for ECB mode).

b) the % difference in $DBER$ ($\Delta DBER$) decreases as n increases, for a fixed code rate. This is observed for all channels, all codes and the 4 cipher modes, with a few exceptions (mostly for the noisiest channel; HIG).

Hence, the conclusions reached earlier for the other channel types apply here as well.

5.7.9 Comparison of *DBER* of CBC and CFB with ECB mode

Tables B.84 to B.89 in annex B give the simulation results. Again, the observations made for the Fritchman model channels are similar to those made BSC and for Gilbert model channels:

- a) the *DBER* of CBC and CFB modes are consistently higher than for ECB mode,
- b) the use of BCH codes tends to magnify these differences but not as much as was the case for memoryless BSC,
- c) for CBC, an increase in n (fixed rate R) or a decrease in the code rate (fixed n) results in an increase in $\Delta DBER$, relative to ECB,
- d) for CFB, no such pattern is evident. In fact, for CFB mode, the % difference in *DBER* relative to ECB changes very little over the entire range of (n,k,t) codes used. The only exceptions occur when $k = 16$ or 64 , where the $\Delta DBER$ is higher than for other BCH codes.

The table below shows the average % difference in *DBER* ($\Delta DBER$) over all channel realizations for the different BCH codes used:

Table 5.20: Average % difference in <i>DBER</i> for CBC and CFB modes versus ECB mode for all BCH codes									
	none	n=7 k=4 t=1	n=15 k=11 t=1	n=15 k=7 t=2	n=15 k=5 t=3	n=31 k=26 t=1	n=31 k=16 t=3	n=31 k=11 t=5	n=63 k=45 t=3
CBC	8.4%	8.3%	11.1%	9.4%	7.7%	14.2%	16.8%	11.6%	17.8%
CFB	4.0%	4.2%	4.9%	4.4%	5.8%	4.8%	13.4%	5.1%	5.6%
	n=63 k=30 t=6	n=63 k=24 t=7	n=63 k=18 t=10	n=63 k=16 t=11	n=127 k=92 t=5	n=127 k=64 t=10	n=127 k=57 t=11	n=127 k=43 t=14	
CBC	17.9%	18.8%	14.6%	18.9%	16.9%	47.3%	21.9%	18.8%	
CFB	4.6%	9.5%	6.9%	17.5%	2.4%	51.6%	4.0%	2.5%	

5.7.10 Comparison of *DBER* of CFB with CBC mode

Tables B.90 to B.92 show the % difference in *DBER* for CFB mode versus CBC mode. The results of these tables lead to observations which are identical to those made earlier for the Gilbert model channels; CFB mode is marginally but consistently superior to CBC modes for all channels and all codes except the (127,64,10) code, where it has a higher *DBER* than CBC mode. Again, there is a slight but consistent tendency for the $\Delta DBER$ to increase as n increases from (n_1, k_1, t_1) to (n_2, k_2, t_2) (for a constant code rate), except when $k_2 = 16$ or 64.

The table below indicates the average $\Delta DBER$ for CFB mode relative to CBC mode, calculated over all channels:

Table 5.21: Average % difference in <i>DBER</i> for CFB mode versus CBC mode for all BCH codes								
none	n=7 k=4 t=1	n=15 k=11 t=1	n=15 k=7 t=2	n=15 k=5 t=3	n=31 k=26 t=1	n=31 k=16 t=3	n=31 k=11 t=5	n=63 k=45 t=3
-4.0%	-3.8%	-5.5%	-4.6%	-1.7%	-8.2%	-2.9%	-5.8%	-10.3%
n=63 k=30 t=6	n=63 k=24 t=7	n=63 k=18 t=10	n=63 k=16 t=11	n=127 k=92 t=5	n=127 k=64 t=10	n=127 k=57 t=11	n=127 k=43 t=14	
-11.2%	-8.0%	-6.7%	-1.2%	-12.2%	2.8%	-14.4%	-12.1%	

5.7.11 Effect of a speed increase

As was the case for the Gilbert model, a change in speed will alter the average length of the error clusters and error-free runs. This will have an impact on the *DBER*, as explained earlier in the section on Gilbert model simulations. However, aside from the expected increase in *DBER* as the BER increases, the results show no unusual behaviour for these channels. This may be caused by the fact that error clusters have a wider variety of density and length for the Fritchman model, so that the overall effect of a change in speed is less evident. However, there are not enough results to reach a firm conclusion about this matter.

5.8 Bit interleaving in Fritchman's fading channels

In this section, we examine the effect of bit interleaving on the performance of binary BCH codes in fading channels. As explained in chapter 4, interleaving attempts to remove channel memory by breaking up error clusters and distributing bit errors uniformly over time. It is important to emphasize that interleaving does not change the channel BER; it merely redistributes bit errors.

As a general rule, bit interleaving will only be effective in

reducing the *DBER* over a fading channel when both of these conditions are met:

- a) the interleaving degree r must be high enough to remove most of the channel memory (i.e. after interleaving, bit errors should be uniformly distributed over time), and
- b) the error correcting capability t of the binary BCH code must be high enough to ensure that it corrects more bit errors in the interleaved channel than it generates as a result of decoding failures ($P_{cb} < P_b$).

If either one of these conditions is not met, the use of interleaving will lead to a net increase in the *DBER*.

The need for the first condition was seen earlier from the simulation runs over Fritchman or Gilbert model channels. In many instances, the BCH codes served no useful purpose because the density of bit errors in error clusters was too high for the code to handle. By breaking up clusters, interleaving increases the probability that a codeword will be able to correct errors (i.e. it increases the probability that the number of bit errors will be less or equal to t).

The second condition is necessary because of the avalanche effect of binary BCH codes. From the simulation results over memoryless BSC, it became apparent that beyond a given channel BER, all binary BCH codes result in a $P_{cb} > P_b$. This occurs because, as the BER increases, the number of bit errors introduced by the code (as a result of a decoding failure) eventually becomes higher than the number of channel bit errors corrected by the code. Since interleaving transforms a fading channel into a memoryless one, it is necessary to ensure that the code will lead to a net reduction in the *DBER* in the equivalent memoryless channel of a given BER.

5.8.1 Effect of bit interleaving on channel memory

It was demonstrated in chapter 2 that the autocorrelation function provides a useful measure of channel memory. The function is also useful in determining the interleaving degree required to remove this memory (i.e. break up error clusters and distribute bit errors uniformly). As a general rule, the lowest value of k for which $\Phi(k) = BER$ is the minimum interleaving degree required to remove all channel memory since it indicates the point where the correlation of the error sequence falls to its minimum value.

From figures C.4 and C.5 of annex C, it is clear that an interleaving degree of $r = 300$ should remove all the memory for all Fritchman model channels. For $r = 25$, none of channels will have their error clusters completely broken up but $r = 100$ should achieve this for some channels.

Recall from chapter 4 that the 'equivalent channel' refers to the distribution of bit errors perceived by an interleaved code. For example, if the first three codewords of a (7,4,1) code are as follows,

$$\begin{aligned}\tilde{C}_1 &= \{c_{1,1}, c_{1,2}, \dots, c_{1,7}\} \\ \tilde{C}_2 &= \{c_{2,1}, c_{2,2}, \dots, c_{2,7}\} \\ \tilde{C}_3 &= \{c_{3,1}, c_{3,2}, \dots, c_{3,7}\}\end{aligned}$$

and the sequence representing the effect of the channel is given by

$$\tilde{e} = \{e_1, e_2, \dots, e_{21}\}$$

an interleaving degree of $r = 3$ will lead to the de-interleaved codewords (at the receiver) having this form:

$$\begin{aligned}\tilde{C}'_1 &= \{c_{1,1}+e_1, c_{1,2}+e_4, \dots, c_{1,7}+e_{19}\} \\ \tilde{C}'_2 &= \{c_{2,1}+e_2, c_{2,2}+e_5, \dots, c_{2,7}+e_{20}\} \\ \tilde{C}'_3 &= \{c_{3,1}+e_3, c_{3,2}+e_6, \dots, c_{3,7}+e_{21}\}\end{aligned}$$

As a result, the 'equivalent channel' perceived by the interleaved codewords has this form:

$$\tilde{e}' = \{e_1, e_4, \dots, e_{19}, e_2, e_5, \dots, e_{20}, e_3, e_6, \dots, e_{21}\}$$

This sequence can easily be generated during simulations (i.e. by XOR-ing the de-interleaved codewords at the receiver with the original codewords before interleaving). Having done this, the autocorrelation function of the equivalent channel can be calculated. This allows us to determine to what extent channel memory has been removed. Figure 5.48 and Figure 5.49 illustrate the autocorrelation functions for channels OP1 (open area, BER = 0.00897) and SUH (suburban area, BER = 0.08920) with $r = 1$ (original channel) and $r = 25, 100$ and 300 (the equivalent channels)¹³ for $n = 31$. The reduction in channel memory as a result of using bit interleaving is evident.

¹³ The autocorrelation function was calculated based on a 200,000 bits sample from the channel file.

Figure 5.48: Autocorrelation function for OP1 channel with $I = 1, 25, 100$ and 300 .

Figure 5.49: Autocorrelation function for SUH channel with $I = 1, 25, 100$ and 300 .

Figure 5.50: Autocorrelation function for SUH channel with $I = 1$ and 25.

Figure 5.50 shows the autocorrelation function of channel SUH for $I = 1$ and $I = 25$, over a shift range of 800 bits. Although the sharp initial decrease for $I = 25$ indicates a significant reduction in the degree of channel memory, the use of interleaving has resulted in a train of periodic impulses at intervals of 31 bits. This behaviour was also apparent for $I = 100$ and $I = 300$, for all channels. The reason behind this peculiar behaviour can be explained as follows: consider the case where $I = 3$ and where the first three (7,4,1) codewords are transmitted over the channel with this bit sequence:

$$\bar{e} = \{00\bar{1}\bar{1}000\bar{1}\bar{1}\bar{1}00000\bar{1}\bar{1}0000\}$$

The adjacent bits in each error burst are flagged so that their relative position in the equivalent channel can easily be seen:

$$\bar{e}' = \{0\bar{1}0\bar{1}0\bar{1}000\bar{1}000\bar{1}0\bar{1}0\bar{1}0000\}$$

Note that for the first burst, the error bits are separated by 13 bit intervals after interleaving. For the second burst, they are

respectively 6 and 7 bit intervals apart. For the last burst, there are 7 bit intervals between the error bits. It can easily be verified that the adjacent bits for any combination of error bursts will be separated by a multiple of n bit intervals (plus or minus one) where n is the length of the codeword. Hence, if the autocorrelation of the equivalent channel is carried out, one would expect that function $\Phi(k)$ will have sharp peaks when k is close to a multiple of n . This is precisely what is happening in Figure 5.50, where the impulses appear at intervals of 31 bits since the equivalent channel was derived from the (31,16,3) code.

These impulses in $\Phi(k)$ indicate that bit errors have a strong tendency to be separated by exactly n bit intervals. For an (n,k,t) code, this implies that if, for instance, bit c_{3j} of the third codeword is corrupted by a channel error, bit c_{4j} of the fourth codeword will also be corrupted with a high probability ($j \leq n$). For block codes, the exact position of a bit error in a codeword is irrelevant; as long as a codeword contains t or less such errors, they will be corrected. Hence, the presence of these impulses in the autocorrelation function does not affect the code performance in any way.

5.8.2 Simulation results for bit interleaving

Simulations were carried out over the fading channels represented by the Fritchman model for all three codes of length $n = 31$. The interleaving degrees selected were 1, 25, 100 and 300 bits. The latter was selected as a maximum since it represents a delay of 30 ms in a digital channel with a transmission rate of 10 kbps; a longer delay is seldom acceptable in commercial voice applications. Simulations were done for OFB mode and CBC mode (lowest and highest error propagation respectively). The simulation results are presented in annex B, in tables B.93 to B.98 inclusively (starting at page 291).

Figure 5.51: Bit interleaving for OFB mode with (31,26,1) code

Figure 5.52: Bit interleaving for CBC mode with (31,26,1) code

For the (31,26,1) code, it can be seen that interleaving is only effective for the channel of BER = 0.003988 (SU1). For both cipher modes, it is effective only for a BER $\leq$ 0.0075. Recall that table 5.2 (page 145) indicates that for this code, $P_{cb} \leq P_b$ for BER $\leq$ 0.00471. Furthermore, the DBER for I of 100 or 300 is only marginally lower than for $I = 25$. Hence, little is gained in exchange for the increased delay. The graphs for the (31,26,1) code indicate clearly the need to ensure that the channel BER is lower than the maximum value P_{MAX} for which the code will result in a $P_{cb} \leq P_b$. Since the second condition required for interleaving to be effective is not met for all channels of BER $\geq$ 0.0075, its use leads to a net increase in the DBER for these links.

Figure 5.53 and Figure 5.54 illustrate the effect of interleaving on the performance of the (31,16,3) code for OFB and CBC modes respectively. For both modes, interleaving is effective up to a channel BER of about 0.07. For the (31,16,3) code, $P_{MAX} = 0.0623$. Also, for the low noise channels, the DBER decreases significantly as the interleaving degree increases. This is simply because a high value of I will generally break up error clusters more uniformly, thereby decreasing the decoding failure probability p_c . However, as the channel BER increases, the reduction in DBER as I increases becomes much less important.

Figure 5.53: Bit interleaving for OFB mode with (31,16,3) code

Figure 5.54: Bit interleaving for CBC mode with (31,16,3) code

For channel OP1 (open area with BER = 0.00897), an interesting situation arises. It seems that $r = 25$ is a bad choice since it leads to a $DBER$ which is higher than the $DBER$ for channel SU2 (suburban with BER = 0.01457) for both cipher modes. For $r = 100$, the $DBER$ is lower for OP1 than SU2 but it is still higher than it should be. However, for $r = 300$, the $DBER$ of OP1 is consistent with the slope of the curves of $DBER$ versus BER. Recall from table 5.18 that the average cluster length for OP1 is 37 bits, as opposed to 19 bits for SU2. Hence, $r = 25$ is clearly insufficient to break up the error clusters. This illustrates the fact that the interleaving degree must be selected as high as possible in order to maximize the probability that all error clusters will be broken up, independently of the channel conditions which control the average cluster length (i.e. urban density, mobile unit speed, etc).

For Figure 5.55 and Figure 5.56 of the (31,11,5) code, the effect of interleaving is even more dramatic. Its use results in a net reduction in $DBER$ for all channels except for the noisiest link (HIG), whose BER = 0.1605. Note that for the (31,11,5) code, $P_{MAX} = 0.145$.

Figure 5.55: Bit interleaving for OFB mode with (31,11,5) code

Figure 5.56: Bit interleaving for CBC mode with (31,11,5) code

From all results of this section, we can conclude that, the lower the code rate is, the more effective interleaving is likely to be. This is to be expected since a reduction in the rate R corresponds to an increase in the error correcting capability t . Hence, a code with a low rate can tolerate more memory (i.e. partially broken clusters resulting from a low value of r) and still lead to a net reduction in the $DBER$.

The validity of the two conditions required to ensure that interleaving will be effective has also been verified; interleaving will only be effective when the interleaving degree is high enough to remove most of the channel memory and when the channel BER is lower or equal to p_{max} for the code being used (so that $P_{cb} < P_b$).

Finally, it is necessary to use the highest possible value of r in order to minimize the possibility that the interleaving degree will lead to an ineffective redistribution of bit errors.

5.9 Conclusion

In this chapter, analytical expressions predicting the $DBER$ performance of stream and block cipher modes over memoryless BSC were derived. These were shown to be in agreement with results obtained by computer simulations. The observations based on the simulation results over BSC and fading channels are summarized below.

The following observations were common to all three channel types (BSC, Gilbert and Fritchman model channels):

- 1) The ranking of $DBER$ performance from best to worst is OFB, ECB, CFB and CBC mode.

2) Short codes outperform longer ones (same rate) beyond a given channel BER but this is not as consistently true for the Fritchman model as it was for the Gilbert model (because of shorter, less dense error clusters) or for the BSC.

3) Increasing n and k (fixed code rate) is more effective in reducing the $DBER$ of ECB, CBC and CFB modes than it is for OFB mode.

4) Decreasing the code rate (fixed n) is more effective in reducing the $DBER$ of OFB mode than it is for the other three modes.

5) Because of chaining of adjacent encrypted blocks (or words) the $DBER$ of CFB and CBC modes is consistently higher than that of ECB mode under similar coding and channel conditions.

6) The use of BCH codes tends to magnify the % difference in $DBER$ of CBC mode relative to ECB mode:

a) Increasing n (fixed rate) or decreasing R (fixed n) is less effective in reducing the $DBER$ of CBC mode than it is for ECB mode.

b) For CFB mode, there is no discernible pattern. Some codes appear to perform better for CFB mode than for ECB mode and vice-versa.

7) CFB mode is consistently superior to CBC mode. The only exception is for $k = 64$, where CBC outperforms CFB mode. Furthermore, increasing n (fixed code rate) is slightly more efficient in reducing the $DBER$ of CFB mode than it is for CBC mode but there are exceptions to this ($k = 16, 64$).

8) The avalanche effect of binary BCH codes (i.e. $P_{cb} > P_b$) is apparent for OFB mode.

9) For ECB and CBC modes, the *DBER* decreases as the code is changed from (127,64,10) to (127,57,11). For CFB mode over the Fritchman model channel, it decreases (like the BSC but unlike the Gilbert channel) but not as much as for OFB mode under the same conditions.

10) For ECB, CBC and CFB modes, the *DBER* is slightly dependent on the specific data sequence transmitted and on the DES key used. The *DBER* of OFB mode is completely independent of both.

These observations are common to the Fritchman model channels and the Gilbert model channels but are different than for the memoryless BSC:

11) The avalanche effect of BCH codes is visible for ECB, CBC and CFB modes ($P_{cb} > P_b$) but to a lesser extent for the Fritchman model than for the Gilbert model channels (none is apparent for the BSC).

12) For fading channel, it is possible to encounter instances where the *DBER* decreases as the BER increases when error clusters become more frequent but shorter. This never occurs for memoryless BSC.

13) A speed increase generally results in error clusters becoming shorter and more frequent for fading channels, which may lead to an increase or decrease in the *DBER* for ECB, CBC and CFB block cipher modes. The *DBER* will increase if the increase in the number of cipher blocks being corrupted (because of more frequent error clusters) is higher than the

decrease in the number of cipher blocks being corrupted (because of shorter error clusters). For memoryless BSC, a speed change has no effect on the BER.

The following observations for the Fritchman model fading channel were different than those of the Gilbert model fading channel:

14) For the Fritchman model, the error cluster length increases as urban density or unit speed increases, in contrast to the Gilbert model. This may be due in part to the fact that error clusters become more frequent and more dense as density (or speed) increases, leading to more clusters being merged into one by the program calculating the average length.

15) Clusters are not as well defined (i.e. more variation in the cluster length) and do not occur as regularly for the Fritchman model as is the case for the Gilbert model. This results in the former having generally less memory than a Gilbert model channel of corresponding BER. As a result of this, the following additional differences appear:

a) Binary BCH codes perform better (i.e. lead to a lower *DBER*) on Fritchman model channels than on a channel generated by the Gilbert model.

b) The ratio $\frac{DBER}{BER}$ is higher for the Fritchman model than the Gilbert model, under similar coding and channel conditions.

c) The % difference in *DBER* for ECB, CBC and CFB modes versus OFB mode is higher for the Fritchman model than for the Gilbert model channel.

Finally, it was observed that bit interleaving over fading channels will only be effective when the interleaving degree is high enough to remove most of the channel memory and when the channel BER is lower or equal to P_{MAX} for the code being used (so that $P_{ch} < P_b$).

Chapter 6

Conclusions and recommendations

In this chapter, the scope of this thesis and the results and conclusions reached in this work are summarized. This is followed by a series of recommendations for future research work in the domain of cryptography applied to digital mobile communication channels.

6.1 Thesis summary

The main objective of this work was to determine the impact of various encryption systems, based on the Data Encryption Standard, on the bit error rate at the receiver after decryption (*DBER*) in a digital mobile communication channel. The ability of error correcting codes and of bit interleaving to correct channel errors for the various DES block and stream cipher modes was examined in detail.

Chapter 2 provided a comprehensive review of the characteristics of digital mobile communication channels. The models used to simulate such communication links were also discussed in that chapter. Chapter 3 described important aspects of cryptographic theory and examined the advantages and disadvantages of the four DES-based cipher modes used in this work. Chapter 4 covered briefly the theory of linear block codes in general and the properties of binary BCH codes in particular. The technique of bit interleaving was also discussed. In chapter 5, a theoretical analysis of the error propagation characteristics of the cipher modes was carried out. This was followed by an analysis of computer simulation results. The main observations and conclusion reached in that chapter are reiterated below.

The results mentioned earlier in this work make it clear that the use of cryptographic systems in a communication channel may lead to a significant degree of cryptographic degradation (i.e. error propagation). These additional bit errors will often transform a marginally acceptable link into a useless one. Hence, error control techniques such as forward error correction coding must be used to reduce the effect of the additional degradation.

Binary BCH codes proved to be reasonably effective in correcting errors for all four cipher modes of DES. However, the performance of these codes over fading channels was shown to be significantly lower than for memoryless links. Nevertheless, the use of bit interleaving enhances the code performance over fading links. In some cases, the use of binary BCH codes leads to a *DBER* which is higher than if no coding is used. For noisy links, short codes almost invariably outperformed longer codes for the same code rate.

For block encryption systems (i.e. ECB and CBC mode), the number of information bits k for an (n,k,t) block code has a significant impact on the ability of the code to correct errors. It was seen that a value of k which is divisible into N without remainder usually leads to the best code performance. For stream cipher systems (i.e. OFB and CFB modes) the value of k is less important since these modes do not have clearly defined block boundaries.

For ECB, CBC and CFB modes, the *DBER* is dependent on the average number of codewords required to encode a full block of N encrypted bits. For CFB mode, this value is sometimes different than for the other two modes because the block boundaries are not clearly defined for that stream cipher mode. For OFB mode, encrypted bits are processed individually and independently so that the *DBER* depends only on the probability of a codeword decoding failure.

It was seen that binary BCH codes are usually most efficient for OFB mode (which experiences no error propagation), followed by ECB mode. Because of the chaining of adjacent encrypted blocks (or words), binary BCH codes are less efficient for CBC and CFB mode than for ECB mode. For all channel types, it was apparent that increasing n and k , for a fixed code rate R , is more effective in reducing the $DBER$ of ECB, CBC and CFB modes than for OFB mode. By contrast, reducing the code rate R , with fixed blocklength n , proved to be more effective in reducing the $DBER$ of OFB mode than for the other modes.

The simulation results indicate that Cipher Block Chaining mode has the highest degree of error propagation, followed by Cipher FeedBack mode, Electronic Code Book mode and Output FeedBack mode, in that order. These observations were consistent for both fading and memoryless channels, over the entire channel BER range, and for most binary BCH codes. This indicates that error propagation characteristics are mostly inherent to these systems as opposed to being determined by the channel or coding conditions.

Bit interleaving is an effective way to reduce the unwanted effects of channel memory. However, it leads to an improved code performance (i.e. lower $DBER$) only when the interleaving degree r is high enough to remove most of the channel memory and when the error correcting capability t of the code is high enough to lead to a net reduction in the $DBER$ over the interleaved channel ($P_{cb} < P_b$). If either one of these conditions is not met, interleaving may lead to an increase in the $DBER$.

6.2 Recommendations for further research

As is often the case in a work of this nature, a few questions were answered, many more remain unresolved and new questions were raised. In this section, some areas which would benefit from

further research work are discussed briefly.

1) It was mentioned in chapter 2 that the use of a renewal model to represent a non-renewal channel process may lead to the simulation results being slightly more optimistic (i.e. lower *DBER*) than one should expect in a real channel. Because of time and space limitations, this factor was not analyzed in this work. It is possible to assess the impact of that factor without having to resort to statistical modelling of the multi-gap distribution. This can be done by comparing the performance of codes over the error sequence (i.e., channel file) measured in a real channel and comparing it to the code performance over error sequences simulated by the renewal model¹.

2) Block codes appear to be naturally well suited for operation with cipher modes which have clearly defined encrypted block boundaries (i.e. ECB and CBC modes). But they are not as efficient for CFB mode, where the block boundaries change at each bit period. One might then wonder if convolutional codes might not prove to be more effective for CFB mode than block codes of comparable error correcting capability.

3) Reed-Solomon codes are maximal distance codes which are well suited for operation over fading channels where errors tend to cluster. As a result, their performance over fading links simulated by the Gilbert or Fritchman models is of interest and warrants further investigation.

4) Similarly, the effectiveness of code concatenation for the four DES cipher modes over fading channels needs to be

¹ Where the parameters of the model are based on the error sequence measured over the real channel.

assessed. As mentioned earlier, block codes are not optimal for stream cipher mode CFB. Hence, one might expect that concatenation of convolutional codes with block codes or of binary block codes and non-binary block codes will lead to significant differences in the *DBER* performance for the stream and block cipher modes.

5) The need to preserve bandwidth and optimize the data throughput over a channel has led to the widespread use of source coding systems. Since source decoding at the receiver would necessarily follow the decryption system, it appears that channel errors propagated by the decryption operation would lead to further error propagation in the source decoder. On the other hand, data compression results in less data bits being transmitted and hence less possibilities for decoding failures. The relationship between these two factors and the degree of error propagation inherent in different source coding systems (e.g. vocoders, Code Excited Linear Predictors, etc) needs to be analyzed in detail.

6) The impact of varying the mobile unit speed on the performance of FEC codes has yet to be determined clearly. In order to minimize the effect of other factors (such as varying urban density, terrain features, etc), error sequences must be experimentally measured for a mobile unit travelling over the same road but at different speeds. Channel models can then be constructed for each error sequence. The simulation results obtained over these (simulated) channels should lead to more detailed observations and conclusions than was possible in this work.

7) The development of analytical expressions to predict the *DBER* performance of CFB mode would provide additional insight on the effectiveness of FEC codes for this cipher system. It would also allow a more detailed comparison with other modes

such as ECB or CBC.

8) Finally, the performance of CFB mode versus ECB and CBC modes has to be explored in more detail in order to reach more detailed observation and conclusions than was possible in this work.

There are many more worthwhile subjects of study than were mentioned above. In any event, the growing demand for mobile communications equipment and the corresponding need for encryption systems will ensure the relevance of any research endeavour in this field.

Annex A

Supplemental graphs of the simulation results

A.1 Introduction

This annex contains additional graphs based on the simulation results shown in annex B. These graphs are referred to in chapter 5.

A.2 Graphs for simulations over BSC

Figure A.1: OFB mode over BSC with BCH coding ($n = 31$).

Figure A.2: OFB mode over BSC with BCH coding ($n = 63$).

Figure A.3: OFB mode over BSC with BCH coding ($n = 127$).

Figure A.4: ECB mode over BSC with BCH coding ($n = 15$).

Figure A.5: ECB mode over BSC with BCH coding ($n = 31$).

Figure A.6: ECB mode over BSC with BCH coding ($n = 63$).

Figure A.7: CBC mode over BSC with BCH coding ($R = 0.75$).

Figure A.8: CBC mode over BSC with BCH coding ($R = 0.50$).

Figure A.9: CBC mode over BSC with BCH coding ($R = 0.33$).

Figure A.10: CFB mode over BSC with BCH coding ($R = 0.75$).

Figure A.11: CFB mode over BSC with BCH coding ($R = 0.50$).

Figure A.12: CFB mode over BSC with BCH coding ($R = 0.33$).

A.3 Graphs for simulations over Gilbert model channels

Figure A.13: OFB mode over Gilbert model channels ($R = 0.50$)

Figure A.14: OFB mode over Gilbert model channels ($R = 0.33$)

Figure A.15: ECB mode over Gilbert model channels ($R = 0.50$)

Figure A.16: ECB mode over Gilbert model channels ($R = 0.33$)

Figure A.17: OFB mode over Gilbert model channels, (63,24,7) code

Figure A.18: ECB mode over Gilbert model channels, (63,24,7) code

A.4 Graphs for simulations over Fritchman model channels

Figure A.19: OFB mode over Fritchman model channels (n = 127)

Figure A.20: ECB mode over Fritchman model channels (n = 127)

Annex B

Simulation results

This annex contains a complete listing of the results of all simulation runs carried out over the simulated channels. The results shown here are analyzed and referred to in chapter 5 of the thesis.

B.1 Simulation runs over binary symmetric channels

This section contains the experimental results of the simulation runs carried out over all BSC in table form. All runs involved the transmission of a data file of 12.233 kbytes (97,864 bits). All channel files are 49 kbytes long (392,000 bits) in order to accommodate the BCH code with the lowest rate.

B.1.1 Output feedback mode simulation results

Table B.1: OFB mode simulations over binary symmetric channels with BCH coding						
BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
0.000199	0.0002146	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0003985	0.0	5.109e-05	0.0	0.0	0.0
0.0007423	0.000797	0.0	0.0	0.0	0.0	0.0001533
0.002064	0.001839	1.022e-05	0.0002248	0.0	0.0	0.0007868
0.004225	0.004312	0.000235	0.0007459	0.0	0.0	0.002534
0.007895	0.008175	0.0007153	0.00235	0.0001533	0.0	0.007725
0.02019	0.01977	0.003556	0.01011	0.001165	0.0001022	0.03508
0.03977	0.03933	0.01415	0.04255	0.007378	0.0009401	0.1023
0.07932	0.0787	0.04948	0.1184	0.04353	0.01203	0.2303
0.1009	0.1015	0.07137	0.166	0.07409	0.02623	0.2867
0.2499	0.251	0.2747	0.3802	0.3341	0.2455	0.4287
0.4995	0.4998	0.4974	0.4985	0.4979	0.5005	0.4998

Table B.2: OFB mode simulations over binary symmetric channels with BCH coding						
BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)	(63,18,10)
0.000199	0.0	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0	0.0
0.007895	9.196e-05	0.0	0.0005722	0.0	0.0	0.0
0.02019	0.0007562	0.0	0.01245	0.0002044	0.0	0.0
0.03977	0.01215	0.0002861	0.09416	0.004629	0.001369	0.0
0.07932	0.08601	0.01356	0.3039	0.1029	0.05555	0.004741
0.1009	0.1456	0.03707	0.3772	0.1964	0.1289	0.02183
0.2499	0.4112	0.3658	0.4777	0.4782	0.4669	0.434
0.4995	0.4986	0.4979	0.5025	0.4993	0.5018	0.5015

Table B.3: OFB mode simulations over binary symmetric channels with BCH coding

BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
0.000199	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0
0.007895	0.0	0.0009094	0.0	0.0	0.0
0.02019	0.0	0.01798	0.0	0.0	0.0
0.03977	0.0	0.1733	0.005048	0.001298	0.0
0.07932	0.001328	0.4373	0.1897	0.1429	0.02991
0.1009	0.01092	0.4732	0.3296	0.2956	0.1277
0.2499	0.424	0.4905	0.484	0.4912	0.4716
0.4995	0.5005	0.5002	0.5017	0.5013	0.4991

B.1.2 Electronic code Book mode simulation results

Table B.4: ECB mode simulations over binary symmetric channels with BCH coding						
BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
0.000199	0.006734	0.0	0.0	0.0	0.0	0.0
0.0004082	0.01312	0.0	0.000327	0.0	0.0	0.0
0.0007423	0.02587	0.0	0.0	0.0	0.0	0.0003168
0.002064	0.05578	0.0002861	0.00186	0.0	0.0	0.002728
0.004225	0.12	0.004547	0.007214	0.0	0.0	0.01112
0.007895	0.2023	0.01212	0.01886	0.001666	0.0	0.03901
0.02019	0.367	0.06079	0.09231	0.01321	0.001328	0.1555
0.03977	0.4599	0.1916	0.2688	0.08089	0.01489	0.352
0.07932	0.4993	0.4153	0.4579	0.3309	0.1406	0.4832
0.1009	0.4973	0.4639	0.4886	0.4215	0.2573	0.4964
0.2499	0.5023	0.5004	0.4991	0.4981	0.4993	0.4983
0.4995	0.5016	0.4982	0.4978	0.4995	0.5024	0.5005

Table B.5: ECB mode simulations over binary symmetric channels with BCH coding						
BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)	(63,18,10)
0.000199	0.0	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0	0.0
0.007895	0.0006744	0.0	0.00187	0.0	0.0	0.0
0.02019	0.004169	0.0	0.03426	0.0006744	0.0	0.0
0.03977	0.06246	0.00187	0.2161	0.01509	0.00562	0.0
0.07932	0.3145	0.09344	0.4628	0.2599	0.1756	0.02092
0.1009	0.4251	0.2122	0.4941	0.4025	0.3379	0.09575
0.2499	0.5008	0.5009	0.4983	0.501	0.4984	0.4995
0.4995	0.4983	0.5002	0.498	0.5	0.5023	0.4994

Table B.6: ECB mode simulations over binary symmetric channels with BCH coding

BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
0.000199	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0
0.007895	0.0	0.001349	0.0	0.0	0.0
0.02019	0.0	0.03097	0.0	0.0	0.0
0.03977	0.0	0.2603	0.005967	0.00279	0.0
0.07932	0.006775	0.4879	0.2147	0.2499	0.07723
0.1009	0.04413	0.5044	0.3703	0.4304	0.2681
0.2499	0.5012	0.5	0.4992	0.5006	0.5017
0.4995	0.5007	0.5017	0.5	0.5013	0.5017

B.1.3 Cipher Block Chaining mode simulation results

Table B.7: CBC mode simulations over binary symmetric channels with BCH coding						
BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
0.000199	0.007071	0.0	0.0	0.0	0.0	0.0
0.0004082	0.01291	0.0	0.0004189	0.0	0.0	0.0
0.0007423	0.02644	0.0	0.0	0.0	0.0	0.0005211
0.002064	0.05594	0.0003781	0.002013	0.0	0.0	0.003638
0.004225	0.124	0.004271	0.007745	0.0	0.0	0.01337
0.007895	0.2063	0.01272	0.02109	0.001747	0.0	0.04496
0.02019	0.3716	0.06429	0.09934	0.01448	0.001328	0.1813
0.03977	0.4649	0.2011	0.2855	0.08619	0.01604	0.3744
0.07932	0.5003	0.4225	0.4669	0.3435	0.1512	0.4928
0.1009	0.5003	0.4691	0.4967	0.4334	0.2707	0.4978
0.2499	0.5003	0.5008	0.5016	0.5017	0.5036	0.4992
0.4995	0.4977	0.4978	0.4996	0.4981	0.5033	0.4996

Table B.8: CBC mode simulations over binary symmetric channels with BCH coding						
BER	(31, 16, 3)	(31, 11, 5)	(63, 45, 3)	(63, 30, 6)	(63, 24, 7)	(63, 18, 10)
0.000199	0.0	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0	0.0
0.007895	0.0006744	0.0	0.002207	0.0	0.0	0.0
0.02019	0.004803	0.0	0.04279	0.0008277	0.0	0.0
0.03977	0.07372	0.002125	0.2585	0.01824	0.006478	0.0
0.07932	0.3443	0.1018	0.4835	0.2985	0.21	0.02571
0.1009	0.4475	0.2338	0.4997	0.4314	0.374	0.1126
0.2499	0.5009	0.501	0.4981	0.5	0.5011	0.497
0.4995	0.4988	0.5003	0.4984	0.5014	0.5013	0.4986

Table B.9: CBC mode simulations over binary symmetric channels with BCH coding

BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
0.000199	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0
0.007895	0.0	0.001686	0.0	0.0	0.0
0.02019	0.0	0.03863	0.0	0.0	0.0
0.03977	0.0	0.3	0.01086	0.003627	0.0
0.07932	0.007929	0.4973	0.3205	0.2962	0.09775
0.1009	0.05485	0.5007	0.4577	0.4553	0.3143
0.2499	0.4992	0.4998	0.4987	0.4991	0.4991
0.4995	0.5011	0.4997	0.4989	0.5003	0.4992

B.1.4 Cipher FeedBack mode simulation results

Table B.10: CFB mode simulations over binary symmetric channels with BCH coding						
BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
0.000199	0.007081	0.0	0.0	0.0	0.0	0.0
0.0004082	0.01326	0.0	0.0003781	0.0	0.0	0.0
0.0007423	0.02625	0.0	0.0	0.0	0.0	0.0005313
0.002064	0.05616	0.0003372	0.002452	0.0	0.0	0.003627
0.004225	0.1238	0.004476	0.007653	0.0	0.0	0.01217
0.007895	0.2096	0.01248	0.02156	0.001563	0.0	0.04048
0.02019	0.3702	0.06498	0.09626	0.0134	0.001359	0.1689
0.03977	0.4654	0.1993	0.2834	0.08415	0.01626	0.3645
0.07932	0.4976	0.4222	0.4687	0.341	0.1496	0.4867
0.1009	0.5005	0.4673	0.4929	0.4324	0.2642	0.4992
0.2499	0.5009	0.4957	0.5017	0.5008	0.4996	0.4993
0.4995	0.4997	0.4994	0.5014	0.4998	0.499	0.5026

Table B.11: CFB mode simulations over binary symmetric channels with BCH coding						
BER	(31, 16, 3)	(31, 11, 5)	(63, 45, 3)	(63, 30, 6)	(63, 24, 7)	(63, 18, 10)
0.000199	0.0	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0	0.0
0.007895	0.0006744	0.0	0.002003	0.0	0.0	0.0
0.02019	0.005262	0.0	0.03475	0.0007562	0.0	0.0
0.03977	0.07576	0.002003	0.23	0.0162	0.006111	0.0
0.07932	0.3561	0.09942	0.473	0.2717	0.1914	0.02274
0.1009	0.4508	0.2299	0.4936	0.4133	0.3538	0.1043
0.2499	0.4992	0.5017	0.5003	0.4992	0.5	0.5024
0.4995	0.499	0.4998	0.5025	0.4979	0.4984	0.4973

Table B.12: CFB mode simulations over binary symmetric channels with BCH coding					
BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
0.000199	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0
0.007895	0.0	0.001604	0.0	0.0	0.0
0.02019	0.0	0.03209	0.0	0.0	0.0
0.03977	0.0	0.2625	0.0111	0.003014	0.0
0.07932	0.007837	0.4903	0.3284	0.2578	0.0824
0.1009	0.05423	0.5035	0.4592	0.433	0.2833
0.2499	0.4991	0.5001	0.4984	0.4977	0.4989
0.4995	0.4971	0.5013	0.4999	0.4987	0.4988

B.1.5 Relative performance of ECB, CBC, CFB modes versus OFB mode

The tables shown on the following pages illustrate the difference in $DBER$ for ECB, CBC and CFB modes (respectively) relative to OFB mode. The calculations involve the simulation results listed earlier in this annex (sections B.1.1 to B.1.4). The difference is given as a percentage, which is calculated as follows (for CBC):

$$\Delta DBER_{CBC} = \frac{DBER_{CBC} - DBER_{OFB}}{DBER_{OFB}} \times 100 \quad (B-2)$$

The formula has the same form for ECB or CFB mode. If the denominator is zero, $\Delta DBER$ is set to zero.

Table B.13: % Difference in DBER for ECB mode relative to OFB in a BSC for various BCH codes						
BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
0.000199	3037.93	0.0	0.0	0.0	0.0	0.0
0.0004082	3192.34	0.0	540.047	0.0	0.0	0.0
0.0007423	3145.92	0.0	0.0	0.0	0.0	106.654
0.002064	2933.17	2699.412	727.402	0.0	0.0	246.721
0.004225	2682.93	1834.893	867.154	0.0	0.0	338.832
0.007895	2374.61	1594.393	702.553	0.0	0.0	404.984
0.02019	1756.34	1609.505	813.056	986.758	0.0	343.273
0.03977	1069.33	1254.063	531.727	1033.91	1199.413	244.086
0.07932	534.434	739.329	286.740	996.367	1483.874	109.813
0.1009	389.950	549.992	194.337	660.165	1068.745	73.143
0.2499	100.119	82.162	31.273	468.902	880.937	16.235
.4995	0.360	0.161	-0.140	49.087	103.380	0.140

Table B.14: % Difference in DBER for ECB mode relative to OFB in a BSC for various BCH codes						
BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)	(63,18,10)
0.000199	0.0	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0	0.0
0.007895	633.362	0.0	226.809	0.0	0.0	0.0
0.02019	451.309	0.0	175.181	229.941	0.0	0.0
0.03977	414.074	553.618	129.503	225.988	310.519	0.0
0.07932	265.655	589.086	52.287	152.575	216.112	341.257
0.1009	191.964	472.431	30.992	104.939	162.141	338.617
0.2499	21.789	36.933	4.312	4.768	6.747	15.092
0.4995	-0.060	0.462	-0.896	0.140	0.100	-0.419

Table B.15: % Difference in DBER for ECB mode relative to OFB in a BSC for various BCH codes					
BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
0.000199	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0
0.007895	0.0	48.340	0.0	0.0	0.0
0.02019	0.0	72.247	0.0	0.0	0.0
0.03977	0.0	50.202	18.205	114.946	0.0
0.07932	410.166	11.571	13.179	74.878	158.208
0.1009	304.121	6.593	12.348	45.602	109.945
0.2499	18.208	1.937	3.140	1.914	6.383
0.4995	0.040	0.300	-0.339	0.0	0.521

Table B.16: % Difference in DBER for CBC mode relative to OFB in a BSC for various BCH codes						
BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
0.000199	3194.967	0.0	0.0	0.0	0.0	0.0
0.0004082	3139.648	0.0	719.926	0.0	0.0	0.0
0.0007423	3217.440	0.0	0.0	0.0	0.0	239.922
0.002064	2941.870	3599.608	795.463	0.0	0.0	362.379
0.004225	2775.695	1717.446	938.343	0.0	0.0	427.624
0.007895	2423.547	1678.274	797.447	0.0	0.0	482.006
0.02019	1779.615	1707.930	882.59	1039.596	0.0	416.819
0.03977	1082.049	1321.201	570.975	1142.918	1199.413	265.982
0.07932	535.705	753.880	294.341	1068.203	1606.201	113.982
0.1009	392.906	557.278	199.217	689.111	1156.857	73.631
0.2499	99.322	82.307	31.931	484.964	932.024	16.445
0.4995	-0.420	0.080	0.221	50.165	105.132	-0.040
				0.040	0.559	

Table B.17: % Difference in DBER for CBC mode relative to OFB in a BSC for various BCH codes					
BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,18,10)
0.000199	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0
0.007895	633.362	0.0	285.704	0.0	0.0
0.02019	535.149	0.0	243.694	304.941	0.0
0.03977	506.749	642.747	174.532	294.038	0.0
0.07932	300.302	650.737	59.098	190.087	442.291
0.1009	207.349	530.699	32.476	119.654	415.804
0.2499	21.814	36.960	4.270	4.559	14.516
0.4995	0.040	0.482	-0.816	0.421	-0.578

Table B.18: % Difference in DBER for CBC mode relative to OFB in a BSC for various BCH codes					
BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
0.000199	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0
0.007895	0.0	85.397	0.0	0.0	0.0
0.02019	0.0	114.850	0.0	0.0	0.0
0.03977	0.0	73.110	115.135	179.430	0.0
0.07932	497.063	13.721	68.951	107.278	226.814
0.1009	402.289	5.811	38.865	54.026	146.124
0.2499	17.736	1.896	3.037	1.608	5.831
0.4995	0.120	-0.100	-0.558	-0.199	0.020

Table B.19: % Difference in DBER for CFB mode relative to OFB in a BSC for various BCH codes						
BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
0.000199	3199.627	0.0	0.0	0.0	0.0	0.0
0.0004082	3227.478	0.0	640.067	0.0	0.0	0.0
0.0007423	3193.601	0.0	0.0	0.0	0.0	246.575
0.002064	2953.833	3199.413	990.747	0.0	0.0	360.981
0.004225	2771.057	1804.681	926.009	0.0	0.0	380.268
0.007895	2463.914	1644.722	817.447	919.569	0.0	424.013
0.02019	1772.534	1727.334	852.127	1050.210	1229.746	381.471
0.03977	1083.321	1308.480	566.040	1040.553	1629.603	256.305
0.07932	532.274	753.274	295.861	683.368	1143.558	111.333
0.1009	393.103	554.757	196.928	483.615	907.244	74.119
0.2499	99.562	81.908	31.957	49.895	103.503	16.468
0.4995	-0.020	0.402	0.582	0.382	-0.299	0.560

Table B.20: % Difference in DBER for CFB mode relative to OFB in a BSC for various BCH codes						
BER	(31, 16, 3)	(31, 11, 5)	(63, 45, 3)	(63, 30, 6)	(63, 24, 7)	(63, 18, 10)
0.000199	0.0	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0	0.0
0.007895	633.362	0.0	250.052	0.0	0.0	0.0
0.02019	595.848	0.0	179.116	269.961	0.0	0.0
0.03977	523.539	600.105	144.265	249.968	346.384	0.0
0.07932	314.022	633.186	55.643	164.043	244.554	379.646
0.1009	209.615	520.178	30.858	110.438	174.476	377.783
0.2499	21.400	37.151	4.731	4.391	7.089	15.760
0.4995	0.080	0.382	0.0	-0.280	-0.678	-0.837

Table B.21: % Difference in <i>DBER</i> for CFB mode relative to OFB in a BSC for various BCH codes					
BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
0.000199	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0
0.007895	0.0	76.380	0.0	0.0	0.0
0.02019	0.0	78.476	0.0	0.0	0.0
0.03977	0.0	51.471	119.890	132.203	0.0
0.07932	490.136	12.120	73.115	80.406	175.493
0.1009	396.612	6.403	39.320	46.482	121.848
0.2499	17.712	1.957	2.975	1.323	5.789
0.4995	-0.679	0.220	-0.359	-0.519	-0.060

B.1.6 Relative performance of CBC, CFB modes versus ECB mode

The tables on the following pages illustrate the difference in *DBER* between CBC and CFB modes respectively, relative to ECB mode. The calculations involve the simulation results listed earlier in this annex (see sections B.1.1 to B.1.4).

Table B.22: % Difference in DBER for CBC mode relative to ECB in a BSC for various BCH codes						
BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
0.000199	5.004	0.0	0.0	0.0	0.0	0.0
0.0004082	-1.601	0.0	28.104	0.0	0.0	0.0
0.0007423	2.203	0.0	0.0	0.0	0.0	64.487
0.002064	0.287	32.157	8.226	0.0	0.0	33.358
0.004225	3.333	-6.070	7.361	0.0	0.0	20.234
0.007895	1.977	4.950	11.824	4.862	0.0	15.252
0.02019	1.253	5.758	7.616	9.614	0.0	16.592
0.03977	1.087	4.958	6.213	6.552	7.723	6.364
0.07932	0.200	1.734	1.965	3.808	7.539	1.987
0.1009	0.603	1.121	1.658	2.823	5.208	0.282
0.2499	-0.398	0.080	0.501	0.723	0.861	0.181
0.4995	-0.778	-0.080	0.362	-0.280	0.179	-0.180

Table B.23: % Difference in DBER for CBC mode relative to ECB in a BSC for various BCH codes						
BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)	(63,18,10)
0.000199	0.0	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0	0.0
0.007895	0.0	0.0	18.021	0.0	0.0	0.0
0.02019	15.207	0.0	24.898	22.731	0.0	0.0
0.03977	18.028	13.636	19.621	20.875	15.267	0.0
0.07932	9.475	8.947	4.473	14.852	19.590	22.897
0.1009	5.269	10.179	1.133	7.180	10.684	17.598
0.2499	0.020	0.020	-0.040	-0.200	0.542	-0.501
0.4995	0.100	0.020	0.080	0.280	-0.200	-0.160

Table B.24: % Difference in DBER for CBC mode relative to ECB in a BSC for various BCH codes					
BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
0.000199	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0
0.007895	0.0	24.981	0.0	0.0	0.0
0.02019	0.0	24.734	0.0	0.0	0.0
0.03977	0.0	15.252	82.001	30.0	0.0
0.07932	17.033	1.927	49.278	18.527	26.570
0.1009	24.292	-0.734	23.602	5.785	17.232
0.2499	-0.399	-0.040	-0.100	-0.300	-0.518
0.4995	0.080	-0.399	-0.220	-0.199	-0.498

Table B.25: % Difference in DBER for CFB mode relative to ECB in a BSC for various BCH codes					
BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)
0.000199	5.153	0.0	0.0	0.0	0.0
0.0004082	1.067	0.0	15.627	0.0	0.0
0.0007423	1.469	0.0	0.0	0.0	67.708
0.002064	0.681	17.861	31.828	0.0	32.955
0.004225	3.167	-1.561	6.085	0.0	9.442
0.007895	3.609	2.970	14.316	0.0	3.768
0.02019	0.872	6.893	4.279	-6.182	8.617
0.03977	1.196	4.019	5.432	1.438	3.551
0.07932	-0.340	1.661	2.359	4.030	0.724
0.1009	0.643	0.733	0.880	3.052	0.564
0.2499	-0.279	-0.140	0.521	2.586	0.201
0.4995	-0.379	0.241	0.723	0.542	-0.677
				0.060	0.420

Table B.26: % Difference in DBER for CFB mode relative to ECB in a BSC for various BCH codes					
BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,18,10)
0.000199	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0
0.007895	0.0	0.0	7.112	0.0	0.0
0.02019	26.217	0.0	1.430	12.129	0.0
0.03977	21.294	7.112	6.43	7.356	0.0
0.07932	13.227	6.400	2.20	4.540	8.700
0.1009	6.045	8.341	-0.101	2.683	8.930
0.2499	-0.319	0.160	0.401	-0.360	0.581
0.4995	0.140	-0.080	0.903	-0.420	-0.421

Table B.27: % Difference in DBER for CFB mode relative to ECB in a BSC for various BCH codes					
BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
0.000199	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0
0.007895	0.0	18.903	0.0	0.0	0.0
0.02019	0.0	3.616	0.0	0.0	0.0
0.03977	0.0	0.845	86.023	8.029	0.0
0.07932	15.675	0.492	52.958	3.161	6.694
0.1009	22.887	-0.178	24.008	0.604	5.670
0.2499	-0.419	0.020	-0.160	-0.579	-0.558
0.4995	-0.719	-0.080	-0.020	-0.519	-0.578

B.1.7 Relative performance of CFB mode versus CBC mode

Table B.28: % Difference in DBER for CFB mode relative to CBC in a BSC for various BCH codes

BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
0.000199	0.141	0.0	0.0	0.0	0.0	0.0
0.0004082	2.711	0.0	-9.740	0.0	0.0	0.0
0.0007423	-0.719	0.0	0.0	0.0	0.0	1.957
0.002064	0.393	-10.817	21.808	0.0	0.0	-0.302
0.004225	-0.161	4.800	-1.188	0.0	0.0	-8.975
0.007895	1.600	-1.887	2.229	-10.532	0.0	-9.964
0.02019	-0.377	1.073	-3.100	-7.459	2.334	-6.839
0.03977	0.108	-0.895	-0.736	-2.367	1.372	-2.644
0.07932	-0.540	-0.071	0.386	-0.728	-1.058	-1.238
0.1009	0.040	-0.384	-0.765	-0.231	-2.401	0.281
0.2499	0.120	-0.220	0.020	-0.179	-0.794	0.020
0.4995	0.402	0.321	0.360	0.341	-0.854	0.600

Table B.29: % Difference in DBER for CFB mode relative to CBC in a BSC for various BCH codes

BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)	(63,18,10)
0.000199	0.0	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0	0.0
0.007895	0.0	0.0	-9.241	0.0	0.0	0.0
0.02019	9.556	0.0	-18.789	-8.638	0.0	0.0
0.03977	2.767	-5.741	-11.025	-11.184	-5.665	0.0
0.07932	3.427	-2.338	-2.172	-8.978	-8.857	-11.552
0.1009	0.737	-1.668	-1.221	-4.196	-5.401	-7.371
0.2499	-0.339	0.140	0.442	-0.160	-0.220	1.087
0.4995	0.040	-0.100	0.823	-0.698	-0.578	-0.261

Table B.30: % Difference in *DBER* for CFB mode relative to CBC
in a BSC for various BCH codes

BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
0.000199	0.0	0.0	0.0	0.0	0.0
0.0004082	0.0	0.0	0.0	0.0	0.0
0.0007423	0.0	0.0	0.0	0.0	0.0
0.002064	0.0	0.0	0.0	0.0	0.0
0.004225	0.0	0.0	0.0	0.0	0.0
0.007895	0.0	-4.864	0.0	0.0	0.0
0.02019	0.0	-16.930	0.0	0.0	0.0
0.03977	0.0	-12.500	2.210	-16.901	0.0
0.07932	-1.160	-1.408	2.465	-12.964	-15.703
0.1009	-1.130	0.560	0.328	-4.898	-9.863
0.2499	-0.020	0.060	-0.060	-0.281	-0.040
0.4995	-0.798	0.320	0.200	-0.320	-0.080

B.1.8 Additional simulations over BSC for the four DES cipher modes

Table B.31: Additional simulation runs for ECB, CBC and CFB modes using the original DES key and new data over BSC

CIPHER TYPE	channel BER	ORIG. DBER	NEW DBER	Δ DBER
ECB	0.000199	0.006734	0.006741	0.149
CBC	0.000199	0.007071	0.007183	1.590
CFB	0.000199	0.007081	0.007000	-1.151
ECB	0.0004082	0.01312	0.012477	-4.905
CBC	0.0004082	0.01291	0.013365	3.528
CFB	0.0004082	0.01326	0.013417	1.181
ECB	0.0007423	0.02587	0.025474	-1.530
CBC	0.0007423	0.02644	0.025995	-1.682
CFB	0.0007423	0.02625	0.026915	2.533
ECB	0.002064	0.05578	0.055536	-0.437
CBC	0.002064	0.05594	0.055986	0.082
CFB	0.002064	0.05616	0.057069	1.619
ECB	0.004225	0.12	0.121025	0.854
CBC	0.004225	0.124	0.123794	-0.166
CFB	0.004225	0.1238	0.124704	0.730
ECB	0.007895	0.2023	0.198919	-1.671
CBC	0.007895	0.2063	0.205663	-0.309
CFB	0.007895	0.2096	0.207298	-1.098
ECB	0.02019	0.367	0.367970	0.264
CBC	0.02019	0.3716	0.373549	0.524
CFB	0.02019	0.3702	0.370289	0.024
ECB	0.03977	0.4599	0.462346	0.532
CBC	0.03977	0.4649	0.459893	-1.077
CFB	0.03977	0.4654	0.467179	0.382
ECB	0.07932	0.4993	0.497609	-0.339
CBC	0.07932	0.5003	0.497497	-0.560
CFB	0.07932	0.4976	0.500245	0.532
ECB	0.1009	0.4973	0.499080	0.358
CBC	0.1009	0.5003	0.499387	-0.183
CFB	0.1009	0.5005	0.497241	-0.651
ECB	0.2499	0.5023	0.501083	-0.242
CBC	0.2499	0.5003	0.499234	-0.213
CFB	0.2499	0.5009	0.500562	-0.067
ECB	0.4995	0.5016	0.496066	-1.103
CBC	0.4995	0.4977	0.499346	0.331
CFB	0.4995	0.4997	0.500010	0.062

Table B.32: Additional simulation runs for ECB, CBC and CFB modes using a new DES key and the original data over BSC

CIPHER TYPE	channel BER	ORIG. DBER	NEW DBER	Δ DBER
ECB	0.000199	0.006734	0.006335	-5.920
CBC	0.000199	0.007071	0.007030	-0.578
CFB	0.000199	0.007081	0.006897	-2.594
ECB	0.0004082	0.01312	0.012129	-7.553
CBC	0.0004082	0.01291	0.013222	2.420
CFB	0.0004082	0.01326	0.013457	1.489
ECB	0.0007423	0.02587	0.024881	-3.821
CBC	0.0007423	0.02644	0.026230	-0.793
CFB	0.0007423	0.02625	0.026261	0.042
ECB	0.002064	0.05578	0.055311	-0.840
CBC	0.002064	0.05594	0.056936	1.781
CFB	0.002064	0.05616	0.057079	1.637
ECB	0.004225	0.12	0.120401	0.335
CBC	0.004225	0.124	0.125204	0.971
CFB	0.004225	0.1238	0.124397	0.482
ECB	0.007895	0.2023	0.201126	-0.580
CBC	0.007895	0.2063	0.205929	-0.180
CFB	0.007895	0.2096	0.207574	-0.967
ECB	0.02019	0.367	0.369247	0.612
CBC	0.02019	0.3716	0.374489	0.777
CFB	0.02019	0.3702	0.369063	-0.307
ECB	0.03977	0.4599	0.460844	0.205
CBC	0.03977	0.4649	0.463981	-0.198
CFB	0.03977	0.4654	0.463623	-0.382
ECB	0.07932	0.4993	0.4966638	-0.533
CBC	0.07932	0.5003	0.496802	-0.699
CFB	0.07932	0.4976	0.499275	0.337
ECB	0.1009	0.4973	0.499193	0.381
CBC	0.1009	0.5003	0.498600	-0.340
CFB	0.1009	0.5005	0.501145	0.129
ECB	0.2499	0.5023	0.497394	-0.977
CBC	0.2499	0.5003	0.499510	-0.158
CFB	0.2499	0.5009	0.495678	-1.043
ECB	0.4995	0.5016	0.494584	-0.427
CBC	0.4995	0.4977	0.501512	0.766
CFB	0.4995	0.4997	0.499602	-0.020

B.2 Simulation runs over fading channels (Gilbert model)

This section contains the results of the simulations carried out over fading channels of varying urban density and varying mobile unit speed. The Gilbert model was used in all cases. See annex C, starting at page 295 for a description of the model parameters used and the channel autocorrelation function. All runs involved the transmission of 12.333 kbytes of data. All channel files are 50 kbytes long (500,000 bits) to accommodate the BCH code with the lowest rate. In the tables of the following pages, channels are identified as follows:

OPL == Open area, low mobile unit speed (7.5 km/hr)
OPM == Open area, medium mobile unit speed (15 km/hr)
OPH == Open area, high mobile unit speed (30 km/hr)
SUB == Suburban area, mobile unit speed = 7.5 km/hr
LOW == Low density urban area, mobile unit speed = 7.5 km/hr
MED == Medium density urban area, mobile unit speed = 7.5 km/hr
HIG == High density urban area, mobile unit speed = 7.5 km/hr

All channel models are associated with a Rician signal power distribution, except for the high density urban area channel (HIG) which is represented by a Rayleigh distribution.

B.2.1 Output Feedback mode simulation results

The results of the simulation runs for OFB mode over fading channels of varying urban and varying speed are shown on the following pages.

Table B.33: OFB mode simulations over Gilbert model channels with BCH coding							
TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
OPL	0.01396	0.01608	0.009656	0.02378	0.009932	0.002749	0.04144
SUB	0.03085	0.02609	0.02799	0.04368	0.03663	0.02308	0.05539
LOW	0.04789	0.04413	0.0553	0.06946	0.07112	0.05024	0.06635
MED	0.07203	0.08189	0.0815	0.08571	0.08943	0.07804	0.09009
HIG	0.08209	0.09418	0.08849	0.09318	0.08636	0.08347	0.1006
OPM	0.02663	0.02639	0.02708	0.04481	0.03217	0.02088	0.05563
OPH	0.04487	0.0557	0.05566	0.07081	0.06307	0.04682	0.0834

Table B.34: OFB mode simulations over Gilbert model channels with BCH coding							
TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)	(63,18,10)
OPL	0.01396	0.01868	0.004598	0.05158	0.02469	0.01584	0.002738
SUB	0.03085	0.05036	0.03889	0.06101	0.06314	0.06205	0.04455
LOW	0.04789	0.07905	0.06341	0.08418	0.09065	0.07731	0.06915
MED	0.07203	0.0924	0.08431	0.0997	0.09747	0.08443	0.08271
HIG	0.08209	0.08694	0.08433	0.09488	0.08636	0.08745	0.08078
OPM	0.02663	0.0473	0.03353	0.06523	0.05616	0.05239	0.03991
OPH	0.04487	0.07282	0.06105	0.09418	0.0785	0.07064	0.05858

Table B.35: OFB mode simulations over Gilbert model channels with BCH coding						
TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
OPL	0.01396	0.001359	0.0692	0.03635	0.03263	0.01083
SUB	0.03085	0.04198	0.06743	0.06446	0.06399	0.06145
LOW	0.04789	0.06861	0.09027	0.08747	0.08477	0.07064
MED	0.07203	0.08395	0.1092	0.104	0.1032	0.08886
HIG	0.08209	0.07585	0.1076	0.09143	0.09438	0.08692
OPM	0.02663	0.03451	0.07649	0.05992	0.05523	0.04937
OPH	0.04487	0.05657	0.1073	0.08374	0.07897	0.06236

B.2.2 Electronic code Book mode simulation results

Table B.36: ECB mode simulations over Gilbert model channels with BCH coding

TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
OPL	0.01396	0.08464	0.0684	0.07705	0.06093	0.02941	0.08428
SUB	0.03085	0.07185	0.07882	0.07632	0.08666	0.08763	0.07647
LOW	0.04789	0.08293	0.1008	0.09599	0.1128	0.1046	0.08464
MED	0.07203	0.1214	0.1208	0.1156	0.1329	0.1359	0.1126
HIG	0.08209	0.112	0.1157	0.1147	0.1234	0.1384	0.1208
OPM	0.02663	0.08233	0.09229	0.08963	0.0938	0.09488	0.08893
OPH	0.04487	0.1265	0.1393	0.1307	0.1458	0.1485	0.1357

Table B.37: ECB mode simulations over Gilbert model channels with BCH coding

TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)	(63,18,10)
OPL	0.01396	0.05485	0.02851	0.07823	0.05366	0.0427	0.01071
SUB	0.03085	0.08047	0.08587	0.0769	0.08586	0.09137	0.0857
LOW	0.04789	0.1073	0.1028	0.1021	0.1141	0.1085	0.1107
MED	0.07203	0.1256	0.1311	0.1217	0.1292	0.1198	0.1322
HIG	0.08209	0.1178	0.1311	0.1201	0.1226	0.1308	0.1317
OPM	0.02663	0.08796	0.09469	0.09123	0.09034	0.0924	0.09318
OPH	0.04487	0.1344	0.1459	0.1401	0.1358	0.1369	0.1422

Table B.38: ECB mode simulations over Gilbert model channels with BCH coding

TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
OPL	0.01396	0.004792	0.08177	0.04169	0.0529	0.0272
SUB	0.03085	0.08639	0.07858	0.07184	0.08515	0.08646
LOW	0.04789	0.1029	0.1046	0.09598	0.1058	0.09816
MED	0.07203	0.1321	0.131	0.1121	0.1333	0.1309
HIG	0.08209	0.126	0.1331	0.09888	0.1308	0.1288
OPM	0.02663	0.0847	0.1009	0.06862	0.08398	0.08356
OPH	0.04487	0.1309	0.1441	0.0996	0.13	0.126

B.2.3 Cipher Block Chaining simulation results

Table B.39: CBC mode simulations over Gilbert model channels with BCH coding						
TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)
OPL	0.01396	0.0877	0.07027	0.08124	0.06382	0.03075
SUB	0.03085	0.07355	0.08118	0.08099	0.09145	0.09329
LOW	0.04789	0.08664	0.107	0.1022	0.1225	0.1148
MED	0.07203	0.1278	0.1351	0.1257	0.1465	0.1538
HIG	0.08209	0.12	0.1307	0.1285	0.1395	0.1558
OPM	0.02663	0.08586	0.09824	0.0983	0.1035	0.1013
OPH	0.04487	0.1364	0.1551	0.1465	0.1646	0.1683

Table B.40: CBC mode simulations over Gilbert model channels with BCH coding						
TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)
OPL	0.01396	0.06168	0.03095	0.08302	0.06287	0.04838
SUB	0.03085	0.08631	0.09449	0.08124	0.0914	0.09917
LOW	0.04789	0.1182	0.1132	0.1065	0.1233	0.1203
MED	0.07203	0.1406	0.1487	0.1328	0.144	0.1396
HIG	0.08209	0.1336	0.1508	0.1313	0.1404	0.1481
OPM	0.02663	0.1021	0.1069	0.1031	0.1043	0.1097
OPH	0.04487	0.1589	0.1691	0.1592	0.1599	0.1632

Table B.41: CBC mode simulations over Gilbert model channels with BCH coding						
TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
OPL	0.01396	0.006591	0.08764	0.05645	0.06222	0.03232
SUB	0.03085	0.09907	0.08329	0.08021	0.09113	0.09599
LOW	0.04789	0.121	0.1116	0.1111	0.1174	0.1107
MED	0.07203	0.156	0.1428	0.1345	0.1517	0.1497
HIG	0.08209	0.1492	0.143	0.1242	0.1453	0.1481
OPM	0.02663	0.1003	0.1124	0.08948	0.09724	0.09999
OPH	0.04487	0.1595	0.1649	0.1345	0.1542	0.1484

B.2.4 Cipher FeedBack mode simulation results

Table B.42: CFB mode simulations over Gilbert model channels with BCH coding

TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
OPL	0.01396	0.08797	0.06954	0.07793	0.06241	0.03203	0.08296
SUB	0.03085	0.07262	0.0793	0.07662	0.08762	0.08998	0.07725
LOW	0.04789	0.08378	0.102	0.09784	0.1128	0.1079	0.08718
MED	0.07203	0.1219	0.124	0.1175	0.1357	0.1383	0.1144
HIG	0.08209	0.1129	0.1212	0.1203	0.1264	0.1416	0.1242
OPM	0.02663	0.0821	0.09312	0.09257	0.09447	0.09944	0.08984
OPH	0.04487	0.1319	0.1415	0.1358	0.1485	0.1555	0.1366

Table B.43: CFB mode simulations over Gilbert model channels with BCH coding

TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)	(63,18,10)
OPL	0.01396	0.06248	0.02961	0.07806	0.0549	0.04621	0.01248
SUB	0.03085	0.08446	0.09007	0.0774	0.08494	0.09212	0.08848
LOW	0.04789	0.1105	0.1054	0.1019	0.1152	0.1097	0.1092
MED	0.07203	0.1299	0.1343	0.1242	0.1352	0.127	0.1375
HIG	0.08209	0.1232	0.1345	0.1212	0.1254	0.134	0.1369
OPM	0.02663	0.09543	0.0967	0.09497	0.0937	0.09922	0.09743
OPH	0.04487	0.1457	0.151	0.1423	0.1425	0.1443	0.1481

Table B.44: CFB mode simulations over Gilbert model channels with BCH coding

TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
OPL	0.01396	0.006325	0.08431	0.05689	0.05379	0.02743
SUB	0.03085	0.09481	0.07855	0.08165	0.08364	0.088
LOW	0.04789	0.1115	0.1049	0.1111	0.1089	0.1033
MED	0.07203	0.1427	0.1327	0.136	0.1377	0.1335
HIG	0.08209	0.1362	0.133	0.1255	0.1313	0.1336
OPM	0.02663	0.09353	0.1012	0.09145	0.08432	0.08885
OPH	0.04487	0.1474	0.1472	0.138	0.135	0.1291

B.2.5 Relative performance of ECB, CBC and CFB modes versus OFB mode

Table B.45: % Difference in DBER for ECB mode relative to OFB mode in Gilbert model channels for various BCH codes

TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
OPL	0.01396	426.368	608.368	224.012	513.472	969.844	103.378
SUB	0.03085	175.392	181.601	74.725	136.582	279.679	38.057
LOW	0.04789	87.922	82.278	38.195	58.605	108.201	27.566
MED	0.07203	48.248	48.221	34.873	48.608	74.141	24.986
HIG	0.08209	18.922	30.762	23.544	42.830	65.798	20.089
OPM	0.02663	211.974	240.805	100.022	191.576	354.406	59.860
OPH	0.04487	127.110	150.269	84.578	131.172	217.172	62.710

Table B.46: % Difference in DBER for ECB mode relative to OFB mode in Gilbert model channels for various BCH codes

TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)	(63,18,10)
OPL	0.01396	193.630	520.052	51.667	117.335	169.571	291.161
SUB	0.03085	59.790	120.802	26.045	35.984	47.252	92.368
LOW	0.04789	35.737	62.120	21.288	25.869	40.344	60.087
MED	0.07203	35.931	55.498	22.066	32.554	41.893	59.836
HIG	0.08209	35.472	55.434	26.537	42.014	49.626	62.998
OPM	0.02663	85.962	182.404	39.859	60.862	76.370	133.475
OPH	0.04487	84.565	138.984	48.758	72.994	93.800	142.745

Table B.47: % Difference in DBER for ECB mode relative to OFB mode in Gilbert model channels for various BCH codes

TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
OPL	0.01396	252.612	18.165	14.691	62.121	151.154
SUB	0.03085	105.788	16.536	11.449	33.068	40.700
LOW	0.04789	49.978	15.875	9.729	24.808	38.958
MED	0.07203	57.356	19.963	7.788	29.167	47.310
HIG	0.08209	66.092	23.675	8.157	38.588	48.225
OPM	0.02663	145.436	31.913	14.519	52.055	69.253
OPH	0.04487	131.395	34.296	18.940	64.619	102.053

Table B.48: % Difference in DBER for CBC mode relative to OFB mode in Gilbert model channels for various BCH codes

TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
OPL	0.01396	445.398	627.734	241.631	542.570	1018.589	106.274
SUB	0.03085	181.909	190.032	85.417	149.659	304.203	48.402
LOW	0.04789	96.329	93.490	47.135	72.244	128.503	36.292
MED	0.07203	56.063	65.767	46.657	63.815	97.078	33.866
HIG	0.08209	27.384	47.725	37.866	61.548	86.706	32.328
OPM	0.02663	225.351	262.777	119.371	221.728	385.153	73.234
OPH	0.04487	144.883	178.656	106.892	160.980	259.462	83.693

Table B.49: % Difference in DBER for CBC mode relative to OFB mode in Gilbert model channels for various BCH codes

TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)	(63,18,10)
OPL	0.01396	230.193	573.119	60.954	154.638	205.429	372.973
SUB	0.03085	71.386	142.967	33.158	44.758	59.823	116.566
LOW	0.04789	49.526	78.521	26.515	36.018	55.607	81.779
MED	0.07203	52.165	76.373	33.200	47.738	65.344	85.830
HIG	0.08209	53.644	78.771	38.406	62.601	69.304	87.641
OPM	0.02663	115.856	218.819	58.056	85.719	109.391	164.846
OPH	0.04487	118.209	176.986	69.038	103.694	131.031	185.763

Table B.50: % Difference in DBER for CBC mode relative to OFB mode in Gilbert model channels for various BCH codes

TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
OPL	0.01396	384.989	26.647	55.296	90.683	198.430
SUB	0.03085	135.993	23.521	24.434	42.413	56.208
LOW	0.04789	76.359	23.629	27.015	38.492	56.710
MED	0.07203	85.825	30.769	29.327	46.996	68.467
HIG	0.08209	96.726	32.896	35.863	53.995	56.973
OPM	0.02663	190.640	46.947	49.332	76.064	102.532
OPH	0.04487	181.952	53.681	60.616	95.264	137.973

Table B.51: % Difference in DBER for CFB mode relative to OFB mode in Gilbert model channels for various BCH codes						
TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3) (31,26,1)
OPL	0.01396	447.077	620.174	227.712	528.373	1065.151 100.193
SUB	0.03085	178.344	183.315	75.412	139.203	289.861 39.466
LOW	0.04789	89.848	84.448	40.858	58.605	114.769 31.394
MED	0.07203	48.858	52.147	37.090	51.739	77.217 26.984
HIG	0.08209	19.822	36.998	29.060	46.391	69.605 23.441
OPM	0.02663	211.103	243.870	106.583	193.659	376.245 61.496
OPH	0.04487	136.804	154.222	91.781	135.453	232.123 63.789

Table B.52: % Difference in DBER for CFB mode relative to OFB mode in Gilbert model channels for various BCH codes						
TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7) (63,18,10)
OPL	0.01396	234.475	543.976	51.338	122.357	191.730 355.807
SUB	0.03085	67.712	131.602	26.864	34.526	48.461 98.608
LOW	0.04789	39.785	66.220	21.050	27.082	41.896 57.918
MED	0.07203	40.584	59.293	24.574	38.709	50.420 66.244
HIG	0.08209	41.667	59.433	27.744	45.149	53.202 69.462
OPM	0.02663	101.754	188.398	45.593	66.845	89.387 144.124
OPH	0.04487	100.082	147.338	51.094	81.529	104.275 152.817

Table B.53: % Difference in DBER for CFB mode relative to OFB mode in Gilbert model channels for various BCH codes						
TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
OPL	0.01396	365.416	21.835	56.506	64.848	153.278
SUB	0.03085	125.846	16.491	26.668	30.708	43.206
LOW	0.04789	62.513	16.207	27.015	28.465	46.234
MED	0.07203	69.982	21.520	30.769	33.430	50.236
HIG	0.08209	79.617	23.637	37.237	39.173	53.703
OPM	0.02663	171.023	32.305	52.620	52.671	79.968
OPH	0.04487	160.562	37.185	64.796	70.951	107.024

B.2.6 Relative performance of CBC and CFB modes versus ECB mode

Table B.54: % Difference in DBER for CBC mode relative to ECB mode in Gilbert model channels for various BCH codes

TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
OPL	0.01396	3.615	2.734	5.438	4.743	4.556	1.424
SUB	0.03085	2.366	2.994	6.119	5.527	6.459	7.493
LOW	0.04789	4.474	6.151	6.469	8.599	9.751	6.841
MED	0.07203	5.272	11.838	8.737	10.233	13.171	7.105
HIG	0.08209	7.116	12.972	11.990	13.105	12.611	10.191
OPM	0.02663	4.288	6.447	9.673	10.341	6.766	8.366
OPH	0.04487	7.826	11.342	12.089	12.894	13.333	12.896

Table B.55: % Difference in DBER for CBC mode relative to ECB mode in Gilbert model channels for various BCH codes

TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)	(63,18,10)
OPL	0.01396	12.452	8.558	6.123	17.164	13.302	20.915
SUB	0.03085	7.257	10.038	5.644	6.452	8.537	12.579
LOW	0.04789	10.158	10.117	4.310	8.063	10.876	13.550
MED	0.07203	11.943	13.425	9.121	11.455	16.528	16.263
HIG	0.08209	13.413	15.014	9.380	14.496	13.151	15.118
OPM	0.02663	16.076	12.895	13.011	15.453	18.723	13.436
OPH	0.04487	18.229	15.901	13.633	17.747	19.211	17.722

Table B.56: % Difference in DBER for CBC mode relative to ECB mode in Gilbert model channels for various BCH codes

TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
OPL	0.01396	37.542	7.179	35.404	17.618	18.824
SUB	0.03085	14.678	5.994	11.651	7.023	11.022
LOW	0.04789	17.590	6.692	15.753	10.964	12.775
MED	0.07203	18.092	9.008	19.982	13.803	14.362
HIG	0.08209	18.444	7.456	25.628	11.117	14.990
OPM	0.02663	18.418	11.397	30.399	15.789	19.663
OPH	0.04487	21.849	14.434	35.040	18.615	17.778

Table B.57: % Difference in DBER for CFB mode relative to ECB mode in Gilbert model channels for various BCH codes						
TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3) (31,26,1)
OPL	0.01396	3.934	1.667	1.142	2.429	8.909
SUB	0.03085	1.072	0.609	0.393	1.108	2.682
LOW	0.04789	1.025	1.190	1.927	0.0	3.155
MED	0.07203	0.412	2.649	1.644	2.108	1.766
HIG	0.08209	0.057	4.769	4.837	2.493	2.296
OPM	0.02663	-0.279	0.899	3.280	0.714	4.806
OPH	0.04487	4.269	1.579	3.902	1.852	4.714

Table B.58: % Difference in DBER for CFB mode relative to ECB mode in Gilbert model channels for various BCH codes						
TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7) (63,18,10)
OPL	0.01396	13.911	3.858	-0.217	2.311	8.220
SUB	0.03085	4.958	4.891	0.650	-1.072	0.821
LOW	0.04789	2.982	2.529	-0.196	0.964	1.106
MED	0.07203	3.424	2.441	2.054	4.644	6.010
HIG	0.08209	4.572	2.572	0.953	2.208	2.390
OPM	0.02663	8.492	2.123	4.100	3.719	7.381
OPH	0.04487	8.408	3.496	1.570	4.934	5.405

Table B.59: % Difference in DBER for CFB mode relative to ECB mode in Gilbert model channels for various BCH codes						
TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
OPL	0.01396	31.991	3.106	36.460	1.682	0.845
SUB	0.03085	9.746	-0.038	13.655	-1.773	1.781
LOW	0.04789	8.358	0.287	15.753	2.930	5.236
MED	0.07203	8.024	1.298	21.320	3.301	1.986
HIG	0.08209	8.143	-0.031	26.899	0.422	3.696
OPM	0.02663	10.425	0.297	33.270	0.405	6.331
OPH	0.04487	12.605	2.151	38.554	3.846	2.460

B.2.7 Relative performance of CFB mode versus CBC Mode

Table B.60: % Difference in DBER for CFB mode relative to CBC mode in Gilbert model channels for various BCH codes							
TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
OPL	0.01396	0.308	-1.039	-4.074	-2.209	4.163	-2.948
SUB	0.03085	-1.264	-2.316	-5.396	-4.188	-3.548	-6.022
LOW	0.04789	-3.301	-4.673	-4.266	-7.918	-6.010	-3.594
MED	0.07203	-4.617	-8.216	-6.523	-7.372	-10.078	-5.141
HIG	0.08209	-5.936	-7.262	-6.387	-9.382	-9.159	-6.176
OPM	0.02663	-4.380	-5.212	-5.829	-8.725	-1.836	-6.776
OPH	0.04487	-3.299	-8.769	-7.304	-9.781	-7.605	-10.836

Table B.61: % Difference in DBER for CFB mode relative to CBC mode in Gilbert model channels for various BCH codes							
TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)	(63,18,10)
OPL	0.01396	1.297	-4.330	-5.974	-12.677	-4.485	-3.630
SUB	0.03085	-2.143	-4.678	-4.727	-7.068	-7.109	-8.292
LOW	0.04789	-6.514	-6.890	-4.319	-6.569	-8.811	-13.126
MED	0.07203	-7.610	-9.684	-6.476	-6.111	-9.026	-10.540
HIG	0.08209	-7.795	-10.817	-7.704	-10.733	-9.511	-9.688
OPM	0.02663	-6.533	-9.542	-7.886	-10.163	-9.553	-7.824
OPH	0.04487	-9.060	-10.704	-10.616	-10.882	-11.581	-11.529

Table B.62: % Difference in DBER for CFB mode relative to CBC mode in Gilbert model channels for various BCH codes						
TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
OPL	0.01396	-4.036	-3.800	0.779	-13.549	-15.130
SUB	0.03085	-4.300	-5.691	1.795	-8.219	-8.324
LOW	0.04789	-7.851	-6.004	0.0	-7.240	-6.685
MED	0.07203	-8.526	-7.073	1.115	-9.229	-10.822
HIG	0.08209	-8.697	-6.967	1.012	-9.625	-9.822
OPM	0.02663	-6.750	-9.964	2.202	-13.287	-11.141
OPH	0.04487	-7.586	-10.734	2.602	-12.451	-13.005

B.3 Simulation runs over fading channels (Fritchman model)

This section contains the results of the simulations carried out over fading channels of varying urban density and varying mobile unit speed. The Fritchman model was used in all cases. See annex C for a description of the model parameters used and the channel autocorrelation function. All runs involved the transmission of 12.333 kbytes of data. All channel files are 50 kbytes long (500,000 bits) to accommodate the BCH code with the lowest rate. In the tables of the following pages, channels are identified as follows:

SU1 == Suburban area (60 km/hr.)
OP1 == Open area (50 km/hr.)
SU2 == Suburban area (70 km/hr.)
URB == Urban area (50 km/hr.)
HIG == High density urban area (30 km/hr.)
SUL == Suburban area, low speed (80 km/hr.)
SUH == Suburban area, high speed (90 km/hr.)
OPL == Open area, low speed (50 km/hr.)
OPH == Open area, high speed (100 km/hr.)

B.3.1 Output Feedback mode simulation results

The results of the simulation runs for OFB mode over fading channels of varying urban density and varying speed are shown on the following pages.

Table B.63: OFB mode simulations over Fritchman model channels with BCH coding						
TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3) (31,26,1)
SU1	0.003988	0.004414	0.003995	0.004987	0.003546	0.001972 0.008346
OP1	0.008970	0.009656	0.009748	0.01447	0.01039	0.007071 0.02212
SU2	0.01457	0.01582	0.01453	0.01851	0.01426	0.01055 0.02657
URB	0.02588	0.02877	0.02629	0.03672	0.02651	0.01865 0.05061
HIG	0.1605	0.1596	0.1658	0.1885	0.1665	0.1522 0.2219
SUL	0.02919	0.02603	0.03118	0.04068	0.03811	0.0303 0.04341
SUH	0.08920	0.09356	0.09471	0.1236	0.09938	0.08059 0.1679
OPL	0.01286	0.01185	0.01141	0.01739	0.01133	0.008338 0.02558
OPH	0.03585	0.03406	0.03232	0.04992	0.03388	0.02313 0.07086

Table B.64: OFB mode simulations over Fritchman model channels with BCH coding						
TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7) (63,18,10)
SU1	0.003988	0.002994	0.001318	0.007919	0.001451	0.001073 0.0003883
OP1	0.008970	0.01337	0.008471	0.02378	0.01556	0.01306 0.007378
SU2	0.01457	0.01686	0.01096	0.03184	0.01528	0.01122 0.005906
URB	0.02588	0.03679	0.02208	0.0578	0.03884	0.03045 0.01887
HIG	0.1605	0.2025	0.1796	0.2623	0.233	0.2212 0.1911
SUL	0.02919	0.04373	0.03526	0.05246	0.05145	0.0425 0.03588
SUH	0.08920	0.1373	0.101	0.1961	0.1664	0.1449 0.1054
OPL	0.01286	0.01582	0.01008	0.02908	0.0191	0.01322 0.007756
OPH	0.03585	0.05068	0.02989	0.08558	0.0579	0.04818 0.02625

Table B.65: OFB mode simulations over Fritchman model channels with BCH coding						
TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
SU1	0.003988	0.0002963	0.00702	0.001134	0.0009503	0
OP1	0.008970	0.005201	0.03111	0.01596	0.0146	0.009166
SU2	0.01457	0.004741	0.0454	0.01423	0.01035	0.00513
URB	0.02588	0.01471	0.07786	0.04271	0.03686	0.02223
HIG	0.1605	0.1805	0.3005	0.2644	0.2613	0.2336
SUL	0.02919	0.03242	0.06442	0.0556	0.05424	0.03946
SUH	0.08920	0.08937	0.2368	0.205	0.1918	0.1641
OPL	0.01286	0.005804	0.03966	0.01792	0.01571	0.01
OPH	0.03585	0.01866	0.1157	0.06133	0.05186	0.03864

B.3.2 Electronic code Book mode simulation results

Table B.66: ECB mode simulations over Fritchman model channels with BCH coding							
TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
SU1	0.003988	0.04385	0.04366	0.03795	0.03788	0.02274	0.04058
OP1	0.008970	0.05036	0.0631	0.05926	0.06216	0.06141	0.06061
SU2	0.01457	0.08801	0.1106	0.09735	0.1031	0.09345	0.09047
URB	0.02588	0.1328	0.1506	0.1377	0.148	0.1459	0.1397
HIG	0.1605	0.3308	0.3596	0.3427	0.3702	0.3892	0.3457
SUL	0.02919	0.06113	0.08619	0.08088	0.09638	0.09763	0.07516
SUH	0.08920	0.2783	0.2934	0.2806	0.2948	0.3122	0.2873
OPL	0.01286	0.06097	0.07442	0.06938	0.07205	0.07138	0.06964
OPH	0.03585	0.1646	0.1888	0.1788	0.1903	0.173	0.1744

Table B.67: ECB mode simulations over Fritchman model channels with BCH coding							
TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)	(63,18,10)
SU1	0.003988	0.0174	0.01218	0.0247	0.007204	0.004731	0.00232
OP1	0.008970	0.04977	0.04787	0.05361	0.04405	0.0464	0.03023
SU2	0.01457	0.07324	0.06791	0.08148	0.05437	0.0431	0.03077
URB	0.02588	0.1168	0.113	0.1318	0.1089	0.1017	0.08313
HIG	0.1605	0.3535	0.3747	0.3574	0.3589	0.3638	0.3772
SUL	0.02919	0.08228	0.08976	0.08143	0.09148	0.08603	0.08686
SUH	0.08920	0.2851	0.2958	0.2844	0.2872	0.286	0.2794
OPL	0.01286	0.05617	0.05655	0.06424	0.05533	0.04318	0.03555
OPH	0.03585	0.1571	0.1513	0.1675	0.1459	0.1477	0.1032

Table B.68: ECB mode simulations over Fritchman model channels with BCH coding						
TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
SU1	0.003988	0.001666	0.01393	0.002207	0.002268	0.0
OP1	0.008970	0.02275	0.05763	0.02071	0.03196	0.02713
SU2	0.01457	0.023	0.08315	0.02027	0.02513	0.01731
URB	0.02588	0.0684	0.1356	0.05677	0.08272	0.06109
HIG	0.1605	0.3623	0.3669	0.3089	0.3638	0.3619
SUL	0.02919	0.07932	0.09049	0.06499	0.08793	0.07829
SUH	0.08920	0.2494	0.2911	0.2415	0.277	0.2811
OPL	0.01286	0.02713	0.06623	0.02495	0.03725	0.02954
OPH	0.03585	0.07753	0.1824	0.0811	0.1098	0.09896

B.3.3 Cipher Block Chaining mode simulation results

Table B.69: CBC mode simulations over Fritchman model channels with BCH coding						
TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3) (31,26,1)
SU1	0.003988	0.04781	0.04636	0.04243	0.04165	0.02459 0.04719
OP1	0.008970	0.05594	0.07017	0.06683	0.069	0.06696 0.0706
SU2	0.01457	0.09891	0.1205	0.11	0.1147	0.1028 0.1095
URB	0.02588	0.1464	0.1665	0.1571	0.1623	0.1558 0.1596
HIG	0.1605	0.3464	0.3816	0.3643	0.3873	0.4089 0.3688
SUL	0.02919	0.06789	0.09454	0.09165	0.1094	0.1086 0.08636
SUH	0.08920	0.2831	0.3081	0.2896	0.3082	0.326 0.3012
OPL	0.01286	0.06667	0.08192	0.0791	0.07995	0.07566 0.0825
OPH	0.03585	0.1746	0.202	0.1995	0.2083	0.1872 0.2002

Table B.70: CBC mode simulations over Fritchman model channels with BCH coding						
TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7) (63,18,10)
SU1	0.003988	0.02045	0.01295	0.03024	0.008686	0.006346 0.002647
OP1	0.008970	0.05836	0.05533	0.06647	0.05404	0.05477 0.03567
SU2	0.01457	0.08728	0.07569	0.1021	0.06542	0.0523 0.03531
URB	0.02588	0.1432	0.1282	0.1549	0.1319	0.1204 0.09578
HIG	0.1605	0.3852	0.4054	0.388	0.3896	0.3987 0.4068
SUL	0.02919	0.09735	0.1047	0.09411	0.1074	0.1005 0.1032
SUH	0.08920	0.309	0.3149	0.3017	0.3115	0.3127 0.305
OPL	0.01286	0.06808	0.06392	0.07941	0.06839	0.05337 0.04178
OPH	0.03585	0.1845	0.1718	0.1967	0.1728	0.1744 0.1198

Table B.71: CBC mode simulations over Fritchman model channels with BCH coding						
TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
SU1	0.003988	0.002095	0.01754	0.003178	0.003035	0
OP1	0.008970	0.0273	0.06888	0.03435	0.04059	0.03292
SU2	0.01457	0.02761	0.1018	0.03361	0.03194	0.02094
URB	0.02588	0.08152	0.1613	0.09036	0.1025	0.07699
HIG	0.1605	0.3981	0.3943	0.3663	0.3959	0.4004
SUL	0.02919	0.09495	0.1026	0.08814	0.1058	0.09371
SUH	0.08920	0.2828	0.3085	0.288	0.3038	0.3128
OPL	0.01286	0.03334	0.08084	0.04089	0.04538	0.03555
OPH	0.03585	0.09263	0.2127	0.1246	0.1363	0.1192

B.3.4 Cipher FeedBack mode simulation results

Table B.72: CFB mode simulations over Fritchman model channels with BCH coding							
TYPE	BER	(none)	(7, 4, 1)	(15, 11, 1)	(15, 7, 2)	(15, 5, 3)	(31, 26, 1)
SU1	0.003988	0.04737	0.04616	0.04158	0.04041	0.02536	0.04255
OP1	0.008970	0.05366	0.06594	0.0635	0.06619	0.06791	0.06308
SU2	0.01457	0.09186	0.1156	0.1018	0.1108	0.1038	0.1004
URB	0.02588	0.1359	0.1565	0.142	0.1559	0.1538	0.1426
HIG	0.1605	0.3351	0.3685	0.352	0.3736	0.3978	0.3523
SUL	0.02919	0.06576	0.08755	0.08306	0.09935	0.1002	0.0794
SUH	0.08920	0.2802	0.2982	0.2877	0.2997	0.3167	0.2895
OPL	0.01286	0.06412	0.08052	0.07408	0.07482	0.07213	0.07473
OPH	0.03585	0.1648	0.1989	0.188	0.198	0.1839	0.1839

Table B.73: CFB mode simulations over Fritchman model channels with BCH coding						
TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,18,10)
SU1	0.003988	0.01976	0.01258	0.02632	0.007602	0.006315
OP1	0.008970	0.0572	0.04994	0.05708	0.0474	0.04765
SU2	0.01457	0.08546	0.0726	0.09081	0.05674	0.04951
URB	0.02588	0.1416	0.1217	0.137	0.1159	0.1083
HIG	0.1605	0.3695	0.38274	0.3645	0.3677	0.3779
SUL	0.02919	0.09132	0.09463	0.08664	0.09552	0.09022
SUH	0.08920	0.3033	0.3004	0.2886	0.2932	0.2945
OPL	0.01286	0.06717	0.06093	0.06827	0.05727	0.04735
OPH	0.03585	0.1775	0.1616	0.1767	0.1532	0.1569
						0.03795
						0.1091

Table B.74: CFB mode simulations over Fritchman model channels with BCH coding						
TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
SU1	0.003988	0.002044	0.01452	0.003505	0.002524	0
OP1	0.008970	0.02756	0.05631	0.03592	0.0341	0.02787
SU2	0.01457	0.02814	0.087	0.03437	0.02577	0.01708
URB	0.02588	0.08111	0.141	0.09196	0.08529	0.06564
HIG	0.1605	0.388	0.3737	0.3689	0.3705	0.3709
SUL	0.02919	0.08966	0.09116	0.08988	0.09209	0.08127
SUH	0.08920	0.2754	0.2925	0.2928	0.283	0.2899
OPL	0.01286	0.03405	0.06982	0.0411	0.0377	0.02957
OPH	0.03585	0.09033	0.1869	0.1269	0.112	0.1004

B.3.5 Relative performance of ECB, CBC and CFB modes versus OFB mode

Table B.75: % Difference in DBER for ECB mode relative to OFB mode in Fritchman model channels for various BCH codes							
TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
SU1	0.003988	893.430	992.866	660.979	968.246	1053.144	386.104
OP1	0.008970	421.541	547.312	309.537	498.268	768.477	174.005
SU2	0.01457	456.321	661.184	425.932	623.001	785.782	240.497
URB	0.02588	361.592	472.841	275.00	458.280	682.306	176.032
HIG	0.1605	107.268	116.888	81.804	122.342	155.716	55.791
SUL	0.02919	134.844	176.427	98.820	152.900	222.211	73.140
SUH	0.08920	197.456	209.788	127.023	196.639	287.393	71.114
OPL	0.01286	414.515	552.235	298.965	535.922	756.081	172.244
OPH	0.03585	383.265	484.158	258.173	461.688	647.946	146.119

Table B.76: % Difference in DBER for ECB mode relative to OFB mode in Fritchman model channels for various BCH codes							
TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)	(63,18,10)
SU1	0.003988	481.162	824.127	211.908	396.485	340.913	497.476
OP1	0.008970	272.251	465.104	125.442	183.098	255.283	309.732
SU2	0.01457	334.401	519.617	155.905	255.825	284.135	420.996
URB	0.02588	217.478	411.775	128.028	180.381	233.990	340.540
HIG	0.1605	74.568	108.630	36.256	54.034	64.467	97.384
SUL	0.02919	88.155	154.566	55.223	77.804	102.424	142.085
SUH	0.08920	107.647	192.871	45.028	72.596	97.378	165.085
OPL	0.01286	255.057	461.012	120.908	189.686	226.626	358.355
OPH	0.03585	209.984	406.189	95.723	151.986	206.559	293.143

Table B.77: % Difference in DBER for ECB mode relative to OFB mode in Fritchman model channels for various BCH codes					
TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11) (127,43,14)
SU1	0.003988	462.268	98.433	94.621	138.661
OP1	0.008970	337.416	85.246	29.762	118.904
SU2	0.01457	385.130	83.150	42.446	142.802
URB	0.02588	364.990	74.159	32.920	124.417
HIG	0.1605	100.720	22.097	16.831	39.227
SUL	0.02919	144.664	40.469	16.888	62.113
SUH	0.08920	179.065	22.931	17.805	44.421
OPL	0.01286	367.436	66.994	39.230	137.110
OPH	0.03585	315.488	57.649	32.235	111.724
					195.400
					156.108

Table B.78: % Difference in DBER for CBC mode relative to OFB mode in Fritchman model channels for various BCH codes					
TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2) (15,5,3) (31,26,1)
SU1	0.003988	983.145	1060.451	750.812	1074.563
OP1	0.008970	479.329	619.840	361.852	564.100
SU2	0.01457	525.221	729.319	494.273	704.348
URB	0.02588	408.863	533.321	327.832	512.222
HIG	0.1605	117.043	130.157	93.263	132.613
SUL	0.02919	160.814	203.207	125.295	187.064
SUH	0.08920	202.587	225.309	134.304	210.123
OPL	0.01286	462.616	617.967	354.859	605.649
OPH	0.03585	412.625	525.00	299.639	514.817
					807.412
					709.339
					222.518
					182.529

Table B.79: % Difference in DBER for CBC mode relative to OFB mode in Fritchman model channels for various BCH codes						
TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,18,10)
SU1	0.003988	583.032	882.549	281.866	498.622	491.426
OP1	0.008970	336.500	553.170	179.521	247.301	319.372
SU2	0.01457	417.675	590.602	220.666	328.141	366.132
URB	0.02588	289.236	480.616	167.993	239.598	295.402
HIG	0.1605	90.222	125.724	47.922	67.210	80.244
SUL	0.02919	122.616	196.937	79.394	108.746	136.471
SUH	0.08920	125.055	211.782	53.850	87.200	115.804
OPL	0.01286	330.341	534.127	173.074	258.063	303.707
OPH	0.03585	264.049	474.774	129.843	198.446	261.976
						438.680
						356.381

Table B.80: % Difference in DBER for CBC mode relative to OFB mode in Fritchman model channels for various BCH codes						
TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
SU1	0.003988	607.054	149.858	180.247	219.373	0
OP1	0.008970	424.899	121.408	115.226	178.014	259.153
SU2	0.01457	482.367	124.229	136.191	208.599	308.187
URB	0.02588	454.181	107.167	111.566	178.079	246.334
HIG	0.1605	120.554	31.215	38.540	51.512	71.404
SUL	0.02919	192.875	59.267	58.525	95.059	137.481
SUH	0.08920	216.437	30.279	40.487	58.394	90.615
OPL	0.01286	474.431	103.833	128.181	188.861	255.500
OPH	0.03585	396.409	83.838	103.163	162.823	208.489

Table B.81: % Difference in DBER for CFB mode relative to OFB mode in Fritchman model channels for various BCH codes							
TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
SU1	0.003988	973.176	1055.444	733.768	1039.594	1186.004	409.703
OP1	0.008970	455.717	576.446	338.839	537.055	860.402	185.172
SU2	0.01457	480.657	695.595	449.973	676.999	883.886	277.870
URB	0.02588	372.367	495.283	286.710	488.080	724.665	181.762
HIG	0.1605	109.962	122.256	86.737	124.384	161.367	58.765
SUL	0.02919	152.632	180.789	104.179	160.693	230.693	82.907
SUH	0.08920	199.487	214.856	132.767	201.570	292.977	72.424
OPL	0.01286	441.097	605.697	325.992	560.371	765.076	192.142
OPH	0.03585	383.852	515.408	276.603	484.416	695.071	159.526

Table B.82: % Difference in DBER for CFB mode relative to OFB mode in Fritchman model channels for various BCH codes							
TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)	(63,18,10)
SU1	0.003988	559.987	854.476	232.365	423.915	488.537	639.377
OP1	0.008970	327.823	489.541	140.033	204.627	264.855	325.861
SU2	0.01457	406.880	562.409	185.207	271.335	341.266	456.553
URB	0.02588	284.887	451.178	137.024	198.404	255.665	367.409
HIG	0.1605	82.469	113.107	38.963	57.811	70.841	100.0
SUL	0.02919	108.827	168.378	65.154	85.656	112.282	154.515
SUH	0.08920	120.903	197.426	47.170	76.202	103.244	171.632
OPL	0.01286	324.589	504.464	134.766	199.843	258.169	389.299
OPH	0.03585	250.237	440.649	106.473	164.594	225.654	315.619

Table B.83: % Difference in DBER for CFB mode relative to OFB mode in Fritchman model channels for various BCH codes						
TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
SU1	0.003988	589.841	106.838	209.083	165.600	0.0
OP1	0.008970	429.898	81.003	125.063	133.562	204.058
SU2	0.01457	493.546	91.630	141.532	148.986	232.943
URB	0.02588	451.394	81.094	115.313	131.389	195.277
HIG	0.1605	114.958	24.359	39.523	41.791	58.776
SUL	0.02919	176.558	41.509	61.656	69.782	105.955
SUH	0.08920	208.157	23.522	42.829	47.550	76.661
OPL	0.01286	486.664	76.046	129.353	139.975	195.700
OPH	0.03585	384.084	61.538	106.913	115.966	159.834

B.3.6 Relative performance of CBC and CFB modes versus ECB mode

Table B.84: % Difference in DBER for CBC mode relative to ECB mode in Fritchman model channels for various BCH codes							
TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
SU1	0.003988	9.031	6.184	11.805	9.952	8.135	16.289
OP1	0.008970	11.080	11.204	12.774	11.004	9.038	16.482
SU2	0.01457	12.385	8.951	12.994	11.251	10.005	21.035
URB	0.02588	10.241	10.558	14.089	9.662	6.785	14.245
HIG	0.1605	4.716	6.118	6.303	4.619	5.062	6.682
SUL	0.02919	11.058	9.688	13.316	13.509	11.236	14.902
SUH	0.08920	1.725	5.010	3.207	4.546	4.420	4.838
OPL	0.01286	9.349	10.078	14.010	10.965	5.996	18.466
OPH	0.03585	6.075	6.992	11.577	9.459	8.208	14.794

Table B.85: % Difference in DBER for CBC mode relative to ECB mode in Fritchman model channels for various BCH codes						
TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7) (63,18,10)
SU1	0.003988	17.529	6.322	22.429	20.572	34.137
OP1	0.008970	17.259	15.584	23.988	22.679	18.039
SU2	0.01457	19.170	11.456	25.307	20.324	21.346
URB	0.02588	22.603	13.451	17.527	21.120	18.387
HIG	0.1605	8.967	8.193	8.562	8.554	9.593
SUL	0.02919	18.316	16.644	15.572	17.403	16.820
SUH	0.08920	8.383	6.457	6.083	8.461	9.336
OPL	0.01286	21.203	13.033	23.615	23.604	23.599
OPH	0.03585	17.441	13.549	17.433	18.437	18.077
						17.525
						16.085

Table B.86: % Difference in DBER for CBC mode relative to ECB mode in Fritchman model channels for various BCH codes						
TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
SU1	0.003988	25.750	25.915	43.996	33.818	0.0
OP1	0.008970	20.0	19.521	65.862	27.003	21.342
SU2	0.01457	20.043	22.429	65.812	27.099	20.971
URB	0.02588	19.181	18.953	59.169	23.912	26.027
HIG	0.1605	9.881	7.468	18.582	8.824	10.638
SUL	0.02919	19.705	13.383	35.621	20.323	19.696
SUH	0.08920	13.392	5.977	19.255	9.675	11.277
OPL	0.01286	22.890	22.059	63.888	21.826	20.345
OPH	0.03585	19.476	16.611	53.637	24.135	20.453

Table B.87: % Difference in DBER for CFB mode relative to ECB mode in Fritchman model channels for various BCH codes							
TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
SU1	0.003988	8.027	5.726	9.565	6.679	11.522	4.855
OP1	0.008970	6.553	4.501	7.155	6.483	10.585	4.075
SU2	0.01457	4.375	4.521	4.571	7.468	11.075	10.976
URB	0.02588	2.334	3.918	3.123	5.338	5.415	2.076
HIG	0.1605	1.300	2.475	2.714	0.918	2.210	1.909
SUL	0.02919	7.574	1.578	2.695	3.082	2.632	5.64
SUH	0.08920	0.683	1.636	2.530	1.662	1.441	0.766
OPL	0.01286	5.166	8.197	6.774	3.845	1.051	7.309
OPH	0.03585	0.122	5.350	5.145	4.046	6.301	5.447

Table B.88: % Difference in DBER for CFB mode relative to ECB mode in Fritchman model channels for various BCH codes							
TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,24,7)	(63,18,10)
SU1	0.003988	13.563	3.284	6.559	5.525	33.481	23.750
OP1	0.008970	14.929	4.324	6.473	7.605	2.694	3.936
SU2	0.01457	16.685	6.906	11.451	4.359	14.872	6.825
URB	0.02588	21.233	7.699	3.945	6.428	6.490	6.099
HIG	0.1605	4.526	2.146	1.987	2.452	3.876	1.326
SUL	0.02919	10.987	5.426	6.398	4.416	4.870	5.135
SUH	0.08920	6.384	1.555	1.477	2.089	2.972	2.470
OPL	0.01286	19.583	7.745	6.273	3.506	9.657	6.751
OPH	0.03585	12.985	6.808	5.493	5.003	6.229	5.717

Table B.89: % Difference in DBER for CFB mode relative to ECB mode in Fritchman model channels for various BCH codes						
TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
SU1	0.003988	22.689	4.235	58.813	11.287	0.0
OP1	0.008970	21.143	-2.290	73.443	6.696	2.728
SU2	0.01457	22.348	4.630	69.561	2.547	-1.329
URB	0.02588	18.582	3.982	61.987	3.107	7.448
HIG	0.1605	7.094	1.853	19.424	1.842	2.487
SUL	0.02919	13.036	0.740	38.298	4.731	3.806
SUH	0.08920	10.425	0.481	21.242	2.166	3.131
OPL	0.01286	25.507	5.421	64.729	1.208	0.102
OPH	0.03585	16.510	2.467	56.473	2.004	1.455

B.3.7 Relative performance of CFB mode versus CBC mode

Table B.90: % Difference in DBER for CFB mode relative to CBC mode in Fritchman model channels for various BCH codes							
TYPE	BER	(none)	(7,4,1)	(15,11,1)	(15,7,2)	(15,5,3)	(31,26,1)
SU1	0.003988	-0.920	-0.431	-2.003	-2.977	3.131	-9.833
OP1	0.008970	-4.076	-6.028	-4.983	-4.072	1.419	-10.652
SU2	0.01457	-7.128	-4.066	-7.454	-3.400	0.973	-8.311
URB	0.02588	-7.172	-6.006	-9.612	-3.943	-1.284	-10.652
HIG	0.1605	-3.262	-3.433	-3.376	-3.537	-2.715	-4.474
SUL	0.02919	-3.137	-7.394	-9.373	-9.186	-7.735	-8.059
SUH	0.08920	-1.024	-3.213	-0.656	-2.758	-2.853	-3.884
OPL	0.01286	-3.825	-1.709	-6.346	-6.417	-4.666	-9.418
OPH	0.03585	-5.613	-1.535	-5.764	-4.945	-1.763	-8.142

Table B.91: % Difference in DBER for CFB mode relative to CBC mode in Fritchman model channels for various BCH codes						
TYPE	BER	(31,16,3)	(31,11,5)	(63,45,3)	(63,30,6)	(63,18,10)
SU1	0.003988	-3.374	-2.857	-12.963	-12.480	-0.488
OP1	0.008970	-1.988	-9.742	-14.127	-12.287	-13.000
SU2	0.01457	-2.085	-4.082	-11.058	-13.268	-5.335
URB	0.02588	-1.117	-5.070	-11.556	-12.130	-10.050
HIG	0.1605	-4.076	-5.590	-6.057	-5.621	-5.217
SUL	0.02919	-6.194	-9.618	-7.938	-11.061	-10.229
SUH	0.08920	-1.845	-4.604	-4.342	-5.875	-5.820
OPL	0.01286	-1.337	-4.678	-14.028	-16.260	-11.280
OPH	0.03585	-3.794	-5.937	-10.168	-11.343	-10.034
						-9.167
						-8.932

Table B.92: % Difference in DBER for CFB mode relative to CBC mode in Fritchman model channels for various BCH codes						
TYPE	BER	(63,16,11)	(127,92,5)	(127,64,10)	(127,57,11)	(127,43,14)
SU1	0.003988	-2.434	-17.218	10.289	-16.837	0.0
OP1	0.008970	0.952	-18.249	4.571	-15.989	-15.340
SU2	0.01457	1.920	-14.538	2.261	-19.317	-18.434
URB	0.02588	-0.503	-12.585	1.771	-16.790	-14.742
HIG	0.1605	-2.537	-5.224	0.710	-6.416	-7.368
SUL	0.02919	-5.571	-11.150	1.974	-12.958	-13.275
SUH	0.08920	-2.617	-5.186	1.667	-6.847	-7.321
OPL	0.01286	2.130	-13.632	0.514	-16.924	-16.821
OPH	0.03585	-2.483	-12.130	1.846	-17.828	-15.772

B.3.8 Bit interleaving for OFB mode (Fritchman model channel)

Table B.93: Bit interleaving for OFB mode in Fritchman model channels with BCH code (31,26,1)					
TYPE	BER	no coding $I = 1$	(31,26,1) $I = 1$	(31,26,1) $I = 25$	(31,26,1) $I = 100$
SU1	0.003988	0.004414	0.008348	0.002115	0.001839
OP1	0.008970	0.009656	0.02212	0.020212	0.012487
SU2	0.01457	0.01582	0.02657	0.028877	0.024463
URB	0.02588	0.02877	0.05061	0.060891	0.062822
HIG	0.1605	0.1596	0.2219	0.320251	0.359816
SUL	0.02919	0.02603	0.04341	0.056640	0.068677
SUH	0.08920	0.09356	0.1679	0.218293	0.236359
OPL	0.01286	0.01185	0.02558	0.022541	0.017259
OPH	0.03585	0.03406	0.07086	0.084832	0.088991
					0.017320
					0.092731

Table B.94: Bit interleaving for OFB mode in Fritchman model channels with BCH code (31,16,3)					
TYPE	BER	no coding $I = 1$	(31,16,3) $I = 1$	(31,16,3) $I = 25$	(31,16,3) $I = 100$
SU1	0.003988	0.004414	0.002994	0.0	0.0
OP1	0.008970	0.009656	0.01337	0.001533	0.000215
SU2	0.01457	0.01582	0.01686	0.000961	0.000439
URB	0.02588	0.02877	0.03679	0.006642	0.004449
HIG	0.1605	0.1596	0.2025	0.254149	0.282443
SUL	0.02919	0.02603	0.04373	0.031973	0.015256
SUH	0.08920	0.09356	0.1373	0.134013	0.127820
OPL	0.01286	0.01185	0.01582	0.001032	0.000061
OPH	0.03585	0.03406	0.05068	0.014847	0.008144
					0.000092
					0.008869

Table B.95: Bit interleaving for OFB mode in Fritchman model channels with BCH code (31,11,5)						
TYPE	BER	no coding $I = 1$	(31,11,5) $I = 1$	(31,11,5) $I = 25$	(31,11,5) $I = 100$	(31,11,5) $I = 300$
SU1	0.003988	0.004414	0.001318	0.0	0.0	0.0
OP1	0.008970	0.009656	0.008471	0.0	0.0	0.0
SU2	0.01457	0.01582	0.01096	0.0	0.0	0.0
URB	0.02588	0.02877	0.02208	0.000450	0.000061	0.0
HIG	0.1605	0.1596	0.1796	0.156881	0.154173	0.156268
SUL	0.02919	0.02603	0.03526	0.012691	0.002054	0.000930
SUH	0.08920	0.09356	0.101	0.059297	0.036234	0.029163
OPL	0.01286	0.01185	0.01008	0.000041	0.000061	0.0
OPH	0.03585	0.03406	0.02989	0.001962	0.000378	0.000593

B.3.9 Bit interleaving for CBC mode (Fritchman model channel)

Table B.96: Bit interleaving for CBC mode in Fritchman model channels with BCH code (31,26,1)						
TYPE	BER	no coding $I = 1$	(31,26,1) $I = 1$	(31,26,1) $I = 25$	(31,26,1) $I = 100$	(31,26,1) $I = 300$
SU1	0.003988	0.04781	0.04719	0.011271	0.012017	0.012385
OP1	0.008970	0.05594	0.0706	0.076177	0.060043	0.055598
SU2	0.01457	0.09891	0.1095	0.107139	0.101406	0.099536
URB	0.02588	0.1464	0.1596	0.204069	0.217792	0.219877
HIG	0.1605	0.3464	0.3688	0.484346	0.498508	0.501308
SUL	0.02919	0.06789	0.08636	0.147940	0.203548	0.227029
SUH	0.08920	0.2831	0.3012	0.410652	0.463633	0.489158
OPL	0.01286	0.06667	0.0825	0.095244	0.075094	0.074593
OPH	0.03585	0.1746	0.2002	0.264888	0.294695	0.296268

Table B.97: Bit interleaving for CBC mode in Fritchman model channels with BCH code (31,16,3)					
TYPE	BER	no coding $I = 1$	(31,16,3) $I = 1$	(31,16,3) $I = 25$	(31,16,3) $I = 100$
SU1	0.003988	0.04781	0.02045	0.0	0.0
OP1	0.008970	0.05594	0.05836	0.007521	0.000674
SU2	0.01457	0.09891	0.08728	0.004537	0.002636
URB	0.02588	0.1464	0.1432	0.032443	0.017821
HIG	0.1605	0.3464	0.3852	0.439242	0.493910
SUL	0.02919	0.06789	0.09735	0.081961	0.046830
SUH	0.08920	0.2831	0.309	0.308449	0.376410
OPL	0.01286	0.06667	0.06808	0.006111	0.000807
OPH	0.03585	0.1746	0.1845	0.069157	0.043070

Table B.98: Bit interleaving for CBC mode in Fritchman model channels with BCH code (31,11,5)					
TYPE	BER	no coding $I = 1$	(31,11,5) $I = 1$	(31,11,5) $I = 25$	(31,11,5) $I = 300$
SU1	0.003988	0.04781	0.01295	0.0	0.0
OP1	0.008970	0.05594	0.05533	0.0	0.0
SU2	0.01457	0.09891	0.07569	0.0	0.0
URB	0.02588	0.1464	0.1282	0.003791	0.0
HIG	0.1605	0.3464	0.4054	0.349720	0.434808
SUL	0.02919	0.06789	0.1047	0.041353	0.005692
SUH	0.08920	0.2831	0.3149	0.205765	0.151731
OPL	0.01286	0.06667	0.06392	0.000388	0.0
OPH	0.03585	0.1746	0.1718	0.014602	0.003209

Annex C

Digital channel model parameters

In this annex, a description of the channel files used in the simulations is provided. Relevant model parameters used to generate these files are also given.

C.1 Memoryless binary symmetric channels

All channel files are 392,000 bits long.

Table C.1: Memoryless binary symmetric channels, BER and channel capacity			
Theoretical BER (used by program to make file)	Number of bits in error out of 392,000 bits	Measured bit error rate (BER)	Calculated capacity (based on theoretical BER)
0.0002	78	0.0001990	0.997254
0.0004	160	0.0004082	0.994908
0.0008	291	0.0007423	0.990616
0.002	809	0.002064	0.979186
0.004	1656	0.004225	0.962378
0.008	3095	0.007895	0.932778
0.02	7916	0.02019	0.858559
0.04	15588	0.03977	0.757708
0.08	31094	0.07932	0.597821
0.10	39556	0.1009	0.531004
0.25	97969	0.2499	0.188722
0.50	195797	0.4995	0.0

The autocorrelation function for all these channels is an

impulse at $k = 0$.

C.2 Fading channels based on the Gilbert model

All channel files are 500,000 bits long. The model parameters are derived from results by [39], as explained in chapter 5.

Table C.2: Gilbert model parameters for fading channels				
Channel label	Channel description	P_{gg}	P_{bb}	p
OPL	Open area, speed = 7.5 km/hr	0.999675	0.998110	0.10
SUB	Suburban area, speed = 7.5 km/hr	0.999656	0.997921	0.20
LOW	Low density urban area, speed = 7.5 km/hr	0.999600	0.997691	0.30
MED	Medium density urban area, speed = 7.5 km/hr	0.999404	0.997033	0.40
HIG	High density urban area, speed = 7.5 km/hr	0.999361	0.996540	0.50
OPM	Open area, speed = 15 km/hr	0.999351	0.996226	0.20
OPH	Open area, speed = 30 km/hr	0.998703	0.992381	0.30

Table C.3: Fading channels (Gilbert model) BER and capacity				
Channel Label	Theoretical BER	Bits in Error out of 500,000 bits	Measured BER	Calculated Capacity (based on theoretical BER)
OPL	0.0146727	6979	0.01396	0.927365
SUB	0.0283946	15426	0.03085	0.892826
LOW	0.0442968	23943	0.04789	0.864115
MED	0.0669099	36013	0.07203	0.829159
HIG	0.0779458	41044	0.08209	0.834648
OPM	0.0293466	13316	0.02663	0.886335
OPH	0.0441357	22433	0.04487	0.855652

The capacity was calculated from equation (2-82) derived in chapter 2 (page 66). The longest gap length considered was 200 bits.

The graphs shown on the next pages illustrate the autocorrelation function for the above channels. It was calculated from a 200,000 points sample from each channel file, with a maximum time shift of 800 bits.

Figure C.1: Autocorrelation function for fading channels (Gilbert model) for different urban environments.

Figure C.2: Autocorrelation function for fading channels (Gilbert model) for varying mobile unit speeds.

The next graph illustrates the autocorrelation function for channel OPH, based on a 300,000 bits sample and with a maximum time shift of 2700 bits.

Figure C.3: Long term autocorrelation function for Gilbert model channel OPH.

C.3 Fading channels based on the Fritchman model

All channel files are 500,000 bits long. The model parameters are derived from results by [61].

Table C.4: Fritchman model parameters for fading channels						
Label	α_1	β_1	α_2	β_2	α_3	β_3
SU1	0.551425	0.685453	0.314832	0.998711		
OP1	0.556226	0.857084	0.096573	0.998904		
SU2	0.629742	0.643552	0.161549	0.997519		
URB	0.566301	0.815557	0.107606	0.997096		
HIG	0.729324	0.592677	0.119140	0.915145	0.014270	0.994939
SUL	0.646204	0.751814	0.026310	0.999231		
SUH	0.640335	0.743248	0.129841	0.950203	0.013075	0.997700
OPL	0.578578	0.841538	0.099554	0.998448		
OPH	0.569142	0.860644	0.092019	0.996160		

Table C.5: Fading channels (Fritchman model) description, BER and capacity					
Label	Channel description	Theo. BER	Bits in error per 500,000	Measured BER	Capacity (based on theo. BER)
SU1	Suburban 60 km/hr $\eta = -72$ dBm	0.004063	1994	0.003988	0.97737
OP1	Open 50 km/hr $\eta = -81$ dBm	0.010828	4485	0.008970	0.955289
SU2	Suburban 70 km/hr $\eta = -81$ dBm	0.014905	7284	0.01457	0.938863
URB	Urban 50 km/hr $\eta = -81$ dBm	0.024721	12938	0.02588	0.898502
HIG	High Dens. Urban, 30 km/hr $\eta = -78$ dBm	0.162564	80259	0.1605	0.526088
SUL	Suburban 80 km/hr $\eta = -68$ dBm	0.026922	14596	0.02919	0.922318
SUH	Suburban 90 km/hr $\eta = -69$ dBm	0.090885	44601	0.08920	0.676383
OPL	Open 50 km/hr $\eta = -82$ dBm	0.014680	6429	0.01286	0.93808
OPH	Open 100 km/hr $\eta = -83$ dBm	0.035228	17926	0.03585	0.852485

In the above table η refers to the average signal power level at the receiver. The capacity was calculated from equation (2-91) derived in chapter 2 (page 68), with a maximum error-free gap length of 2000 bits.

The following figures show the autocorrelation function for the channels described in the preceding pages. It was calculated from a 200,000 points sample from each channel file, with a maximum shift of 800 bits.

Figure C.4: Autocorrelation function for fading channels (Fritchman model) for different Urban environments.

Figure C.5: Autocorrelation function for fading channels (Fritchman model) for varying mobile unit speeds.

The next graph illustrates the autocorrelation function for the channels OP1 and SU2, based on a 300,000 bits sample with a maximum shift of 2700 bits. Note that an artificial offset of 0.02 was added to all the Y-axis values for SU2 so that the two curves would not overlap.

Figure C.6: Long term autocorrelation function for Fritchman model channels OP1 and SU2 (an artificial offset of 0.02 was added to the Y-axis values of SU2 so that the curves would not overlap).

List of references

1. Carlisle Adams, Stafford Tavares, "The Structured Design of Cryptographically Good S-Boxes", Journal of Cryptology, Vol. 3, 1990, pp. 27-41.
2. Adel A. Ali, Ibrahim A. Al-Kadi, "On the Use of Repetition Coding with Binary Digital Modulations on Mobile Channels", IEEE Transactions on Vehicular Technology, Vol. 38, No. 1, February 1989, pp. 14-18.
3. Ross J. Anderson, "Tree Functions and Cipher Systems", CRYPTOLOGIA Journal, Rose-Hulman Institute of Technology, Terre Haute, Indiana, Vol. XV, number 3, July 1991, pp. 194-201.
4. Tor Aulin, "Characteristics of a Digital Mobile Radio Channel". IEEE Transactions on Vehicular Technology, Vol-30, # 2, May 1981, pp. 45-53.
5. Henry Beker, Fred Piper, "Cipher Systems: The Protection of Communications", Northwood Books, London, 1982.
6. J.M. Berger, B. Mandelbrot, "A New Model for Error Clustering in Telephone Circuits", IBM Journal of Resource Development, Vol. 7, July 1963, pp. 224-236.
7. Eli Biham, Adi Shamir, "Differential Cryptanalysis of DES-Like Cryptosystems", Proceedings of the CRYPTO 90 conference, pp. 3-21.
8. Eli Biham, Adi Shamir, "Differential Cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and Lucifer", Proceedings of CRYPTO 91, pp.156-171.
9. Richard E. Blahut, "Principles and Practice of Information Theory", Addison-Wesley, Don Mills Ontario, 1987.
10. C. Boyd, "Applying Compression Coding in Cryptography", article in "Cryptography and Coding II", Clarendon Press, New York, 1992, ISBN 0198533934.
11. Lawrence Brown, Jennifer Seberry, "On the Design of Permutation P in DES Type Cryptosystems", Proceedings of EUROCRYPT 1989, pp. 696-705.
12. John M. Carroll, Lynda E. Robbins, "Using Binary Derivatives to Test an Enhancement of DES", CRYPTOLOGIA journal, Rose-Hulman Institute of Technology, Terre-Haute Indiana, October 1988, Vol. XII, number 4, pp. 193-208.

13. John M. Carroll, Jeff Verhagen, Perry T. Wong, "Chaos in Cryptography: The Escape From the Strange Attractor", CRYPTOLOGIA Journal, Rose-Hulman Institute of Technology, Terre Haute, Indiana, Vol. XVI, number 1, January 1992, pp. 52-70.
14. U. Charash. "Reception Through Nakagami Fading Multipath Channels with Random Delays". IEEE Transactions on Communications, Vol COM-27, # 4, April 1979, pp. 657-670.
15. Kar-Ming Cheung, "On the Decoder Error Probability of Block Codes", IEEE Transactions on Communications, Vol. 40, No. 5, May 1992, pp. 857-859.
16. Jean-Yves Chouinard, "Mesures et Caractérisation d'un Canal de Transmission Radio-Mobile Numérique", Ph.D. thesis, Laval University, Quebec, January 1987.
17. Jean-Yves Chouinard, Michel Lecours, Gilles Y. Delisle, "Estimation of Gilbert's and Fritchman's Models Parameters Using the Gradient Method for Digital Mobile Radio Channels", IEEE Transactions on Vehicular Technology, Vol. 37, No. 3, August 1988, pp.158-166.
18. David L. Cohn, Allen H. Levesque, John H. Meyn, Alan W. Pierce, "Performance of Selected Block and Convolutional Codes on a Fading HF Channel", IEEE Transactions on Information Theory, Vol. IT-14, No. 5, September 1968, pp. 627-639.
19. Charles Connell, "An Analysis of NEWDES: A modified Version of DES", CRYPTOLOGIA journal, Rose-Hulman Institute of Technology, Terre-Haute, Indiana, July 1990, Vol. XIV, number 3, pp. 217-223.
20. Thomas W. Cusick, Michael C. Wood, "The REDOC II CryptoSystem", Proceedings of CRYPTO 90, pp. 545-563.
21. D.E.R Denning, "Cryptography and Data Security", Addison-Wesley, Reading, Massachusetts, 1983.
22. W. Diffie, M.E. Hellman, "Cryptanalysis of the NBS Data Encryption Standard", Stanford University Report MEH-76-2, 1976.
23. Alexander Drukarev, Kai P. Yiu, "Performance of Error Correcting Codes on Channels with Memory", IEEE Transactions on Communications, Vol. COM-34, No. 6, June 1986, pp. 513-521.
24. E.O. Elliott, "A Model of the Switched Telephone Network for Data Communications", Bell Systems Technical Journal, January 1965, pp. 89-109.

25. E.O. Elliott, "Estimates of Error Rates for Codes on Burst-Noise Channels", Bell Systems Technical Journal, September 1963, pp. 1977-1996.
26. Gilles Ferland, "Burst Distribution for Memory Channels: Effect of Interleaving", Proceedings of the Canadian Conference on Electrical and Computer Engineering, 13-16 September 1992, Volume 1, ISBN 0-9694170-3-9, pp. MM2.1.1 to MM2.1.4.
27. Guy Ferland, Jean-Yves Chouinard, "Error Rate Performance Analysis of Stream and Block Ciphers in a Digital Mobile Communication Channel", Third Annual Conference on Vehicular Navigation and Information Systems (VNIS 92, Oslo, Norway), September 1992, pp. 426-433.
28. G. David Forney Jr, "Burst-Correcting Codes for the Classic Bursty Channel", IEEE Transactions on Communications Technology, Vol. COM-19, No. 5, October 1971, pp. 772-781.
29. Bruce D. Fritchman, "A Binary Channel Characterization Using Partitioned Markov Chains", IEEE Transactions on Information Theory, Vol. IT-13, No. 2, April 1967, pp. 221-227.
30. Robert G. Gallager, "Information Theory and Reliable Communication", John Wiley and Sons, 1968.
31. E.N. Gilbert, "Capacity of a Burst-Noise Channel", Bell System Technical Journal, September 1960, pp. 1253-1265.
32. Henri Gilbert, Guy Chasse, "A Statistical Attack of the FEAL-8 Cryptosystem", Proceedings of CRYPTO 90, pp. 22-33.
33. Helen Gustafson, Ed Dawson, Bill Caelli, "Comparison of Block Ciphers", Proceedings of AUSCRYPT 90, pp. 208-220.
34. Abraham H. Haddad, Stephen Tsai, Bernard Goldberg, Gregory C. Ranieri, "Markov Gap Models for Real Communication Channels", IEEE Transactions on Communications Vol. COM-23, No. 11, November 1975, pp. 1189-1197.
35. Tzonelih Hwang, "Secret Error-Correcting Codes", Proceedings of CRYPTO 88, pp. 540-563.
36. N.S. Jayant, Peter Noll, "Digital Coding of Waveforms", Prentice-Hall, 1984.
37. Laveen N. Kanal, A.R.K. Sastry, "Models for Channels with Memory and Their Applications to Error Control", Proceedings of the IEEE, Vol. 66, No. 7, July 1978, pp. 724-744.
38. Xuejia Lai, James L. Massey, "A Proposal for a New Block Encryption Standard", EuroCrypt 1990, pp. 389-404.

39. M. Lecours, J.Y. Chouinard, G.Y. Delisle, J. Roy. "Statistical Modelling of the Received Signal Envelope in a Mobile Radio Channel". IEEE Transactions on Vehicular Technology, Vol 37, # 4, November 1988, pp. 204-212.
40. Michel Lecours, Gilles-Y. Delisle, Jean-Yves Chouinard, John Ahern "Envelope, Phase and Data-Transmission Characterization for Mobile Radio Channels at UHF Frequencies", Canadian Journal of Electrical and Computer Engineering, Vol-17, No. 4, 1992, pp.175-182.
41. Lee W.C. "Mobile Communications Engineering". McGraw-Hill, 1982.
42. Cyril Leung, Jimmy Ng, "Estimation of Block Error Rates for NCFSK Modulation on a VHF/UHF Mobile Radio Channel", IEEE Transactions on Vehicular Technology, Vol. 38, No. 2, May 1989, pp. 46-49.
43. Shu Lin, Daniel, J. Costello, "Error Control Coding: Fundamentals and Applications", Prentice-Hall, New Jersey, 1983.
44. Robert Matthews, "On the Derivation of a Chaotic Encryption Algorithm", CRYPTOLOGIA Journal, Rose-Hulman Institute of Technology, Terre Haute, Indiana, Vol. XIII, number 1, January 1989, pp. 29-41.
45. Ueli M. Maurer, "New Approaches to the Design of Self-Synchronizing Stream Ciphers", Proceedings of EUROCRYPT 91, pp. 458-471.
46. R.H. McCullough, "The Binary Regenerative Channel", Bell Systems Technical Journal, vol. 47, October 1968, pp. 1713-1735.
47. Robert McLaughlin, "Yet Another Machine to Break DES", CRYPTOLOGIA journal, Rose-Hulman Institute of Technology, Terre-Haute, Indiana, April 1992, Vol. XVI, number 2, pp. 136-143.
48. Shoji Miyaguchi, "The FEAL Cipher Family", Proceedings of CRYPTO 90, pp. 627-638.
49. Michael Munter, Jack K. Wolf, "Predicted Performance of Error-Control Techniques Over Real Channels", IEEE Transactions on Information Theory, Vol. IT-14, No. 5, September 1968, pp.640-650.
50. M. Nakagami, "The m-distribution- A general formula of intensity distribution of rapid fading". From "Statistical Methods of Radio Wave Propagation", Pergamon Press, 1960

51. Katharyn E. Olson, Per K. Enge, "Forward Error Correction for an Atmospheric Noise Channel", IEEE Transactions on Communications, Vol. 40, No. 5, May 1992, pp. 863-872.
52. Athanasios Papoulis, "Probability, Random Variables and Stochastic Processes", 2nd edition, McGraw-Hill, 1985.
53. Charles P. Pfleeger, "Security in Computing", Prentice-Hall, 1989.
54. John G. Proakis, "Digital Communications" Second Edition. McGraw-Hill, 1989.
55. D.O. Reudink, "Preliminary Investigation of Mobile Radio Transmission at X-Band in an Urban Area". 1967 Fall URSI Meeting at Ann Arbor, Mich.
56. D.O. Reudink, M.F. Wazowicz, "Some Propagation Experiments Relating Foliage Loss and Diffraction Loss at X-Band at UHF Frequency", Joint IEEE Communication Society and Vehicular Technology Society Special Transactions on Mobile Radio Communication, November 1973, pp. 1198-1206.
57. Rainer A. Rueppel, "Security Models and Notions for Stream Ciphers", article in "Cryptography and Coding-II" , Clarendon Press, New York, 1992, ISBN 0198533934.
58. Jean Roy, "Modelisation Statistique d'un Canal de Transmission Radio-Mobile à Bande Étroite", Master's thesis, Laval University, Quebec city, Canada, September 1986.
59. Jennifer Seberry, Josef Pieprzyk, "Cryptography: An introduction to Computer Security", Prentice-Hall, 1989.
60. Assia Semmar, Michel Lecours, Jean-Yves Chouinard, John Ahern, "Characterization of Error Sequences in UHF Digital Mobile Radio Channels", IEEE Transactions on Vehicular Technology, Vol. 40, No. 4, November 1991, pp. 769-775.
61. Assia Semmar, "Caractérisation Des Erreurs de Transmission en Radio-Mobile Numérique", M.Sc. thesis, Laval University (Quebec City), May 1990 (in french).
62. C.E. Shannon, "Communication Theory of Secrecy Systems", Bell Systems Technical Journal, Vol. 28, 1949, pp. 656-715.
63. Akihiro Shimizu, Shoji Miyaguchi, "Fast Data Encipherment Algorithm FEAL", Proceedings of EUROCRYPT 87, pp. 267-278.
64. Murray R. Spiegel, "Mathematical Handbook", Schaum's outline series in mathematics, McGraw-Hill, Toronto, 1968.

65. Hirofumo Suzuki. "A Statistical Model for Urban Radio Propagation". IEEE Transactions on Communications, Vol COM-25, # 7, July 1977, pp. 673-679.
66. Anne Tardy-Corffdir, Henri Gilbert, "A Known Plaintext Attack on FEAL-4 and FEAL-6", Proceedings of CRYPTO 91, pp. 172-182.
67. Don J. Torrieri, "Principles of Secure Communication Systems", first edition, Artech House.
68. Rodney G. Vaughan. "Signals in Mobile Communications: A Review". IEEE Transactions on Vehicular Technology, Vol-35, # 4, November 1986, pp. 133-145.
69. Daniel D. Wheeler, Robert A.J. Matthews, "Supercomputer Investigations of a Chaotic Encryption Algorithm", CRYPTOLOGIA Journal, Rose-Hulman Institute of Technology, Terre Haute, Indiana, Vol. XV, number 2, April 1991, pp. 140-149.
70. Daniel D. Wheeler, "Problems with Chaotic Cryptosystems", CRYPTOLOGIA Journal, Rose-Hulman Institute of Technology, Terre Haute, Indiana, Vol. XIII, number 3, July 1989, pp. 243-250.
71. Stephen B. Wicker, " Reed-Solomon Error Control Coding for Rayleigh Fading Channels with Feedback", IEEE Transactions on Vehicular Technology, Vol. 41, No. 2, May 1992, pp. 124-133.