Fast Attacks on Tree-structured Ciphers

William Millan
Signal Processing Research Center
Information Security Research Center

E. P. Dawson and L. J. O'Connor*
CRC for Distributed Systems Technology Center
Information Security Research Center

March 17, 1994

Queensland University of Technology
GPO Box 2434, Brisbane, Queensland, Australia 4001

Abstract

Tree-structures have been proposed for both the construction of block ciphers by Kam and Davida [7], and self-synchronous stream ciphers by Kühn [9]. Attacks on these ciphers have been given by Anderson [2], and Heys and Tavares [6]. In this paper it is demonstrated that a more efficient attack can be conducted when the underlying boolean functions for the cells are known. It is shown that this attack requires less than $\frac{1}{3}$ the chosen ciphertext of Anderson's original attack on Kühn's cipher. We also comment on an improved version of Kühn's cipher that was modified in light of Anderson's original attack.

*The work reported in this paper has been funded in part by the Cooperative Research Centres program through the Department of the Prime Minister and Cabinet of Australia.

1 Introduction

This paper deals with the cryptanalysis of ciphers which can be reduced to a boolean function which has a *tree-structure*, such as the cipher proposed by Kühn [9], and Kam and Davida's construction [7] of substitution-permutation networks (SPNs) which ensure the completeness property. Anderson [2] has shown that a tree-structured network is vulnerable to a divide-and-conquer style chosen-ciphertext attack. Heys and Tavares [6] have provided an extension of Anderson's attack which is applicable to tree-structured SPNs.

A framework is provided for discussion of tree-structured function networks. Anderson's attack is restated, along with an improvement which is both faster and more effective. Against Kühn's cipher, the improved attack requires less than $\frac{1}{3}$ of the chosen ciphertext required by Anderson's attack. In addition, the improved version recovers the secret key explicitly, whereas the original attack provided sufficient information to mimic the cipher, with the key remaining implicit. Both attacks rely on the cipher's tree-structure for their operation.

In [3], Anderson reports that Kühn has suggested that the insertion of a random bit-position permutation between rows of a tree-structured network will prevent the operation of Anderson's attack. However we present an approach which exploits the tree-structure to quickly unravel the permutation to a degree sufficient to allow Anderson's attack to proceed.

2 Tree-Structured Networks

Large functions $F : Z_2^N \mapsto Z_2$ can be constructed as a network of smaller sub-functions called *cells*, where each cell has m inputs, $m < N$. With each network we can associate a graph where cells become vertices and connections become edges. In this paper we will be concerned with functions for which the corresponding graph is a complete m-ary tree [8]. Kühn's cipher uses a 125-bit function constructed as a complete 5-ary tree of depth three, and is presented in Section 2.2. From now on in this paper we will use the term tree-structure to mean complete m-ary tree-structure.

A tree-structured network of depth R can be considered as R rows of m-input cells, with m^{R-r} cells $C_{r,i}$ in row r. The total number of cells in a tree-structured network is $\frac{m^R-1}{m-1}$. The network function is denoted $F_{net}(\vec{X}) = Z$.

Each N-bit network input will induce a corresponding m^{R+1-r} bit value $\vec{Z}_r$ at the output of row r, which is the input to row $r+1$. Adams [1] first observed that tree-structured functions perform poorly with respect to the strict avalanche criterion [12]. It can be shown that for a function arranged as a tree-structured network of random balanced m-bit sub-functions, the probability of the output changing in response to a single input bit change is given by $(\frac{1}{2}+\varepsilon)^R$, where $\varepsilon = \frac{1}{2}\left(\frac{1}{2^m-1}\right)$.

2.1 Isolation of Cells

The output value of any cell $C_{r,i}$ in a tree-structured network is dependant on a subset of the network input bits, called the *vital set* $V_{r,i}$. The remaining input bits $P_{r,i} = X - V_{r,i}$ will be called the *propagation set*. The output $z_{r,i}$ of cell $C_{r,i}$ is a function of the contents of the vital set $V_{r,i}$, so that $z_{r,i} = f_{r,i}(V_{r,i})$. If the network is a complete m-ary tree, then $|V_{r,i}| = m^r$. Changes to the contents of a vital set may effect no more than one cell in each row.

Consider a series of network inputs which hold the contents of $P_{r,i}$ constant while the contents of $V_{r,i}$ are varied, causing the output of cell $C_{r,i}$ to change. For some values of $P_{r,i}$, the propagation of this change will be halted at an internal row, and the network output will remain constant. However, other values of $P_{r,i}$ allow the cell's output change to propagate to the network output. If such a value for $P_{r,i}$ is found then cell $C_{r,i}$ is said to be *isolated*. Once cell $C_{r,i}$ is isolated,

$$F_{net}(V_{r,i}, P_{r,i}) \quad = \quad f_{r,i}(V_{r,i}) + c$$

where $c \in \{0,1\}$. Thus we can determine the truth table for $f_{r,i}$ up to complementation by cycling through the values of $V_{r,i}$.

Using a simple algorithm, the average number of trials required to isolate a cell is $\left(\frac{2}{1+2\varepsilon}\right)^{R-r+1}$, and the time to isolate a top row cell is exponential in the depth of the network. However, since the propagation sets are not disjoint, $P_{r,i}$ can be partially constructed from $P_{r+1,j}$, where cell $C_{r,i}$ provides an input to cell $C_{r+1,j}$. This observation leads to a faster algorithm, first presented by Heys and Tavares [5], in which the expected number of iterations to isolate a cell is $\left(\frac{2}{1+2\varepsilon}\right)^2 \approx 4$. Approximately, an average of eight chosen inputs are required for each cell, regardless of the depth of the network.

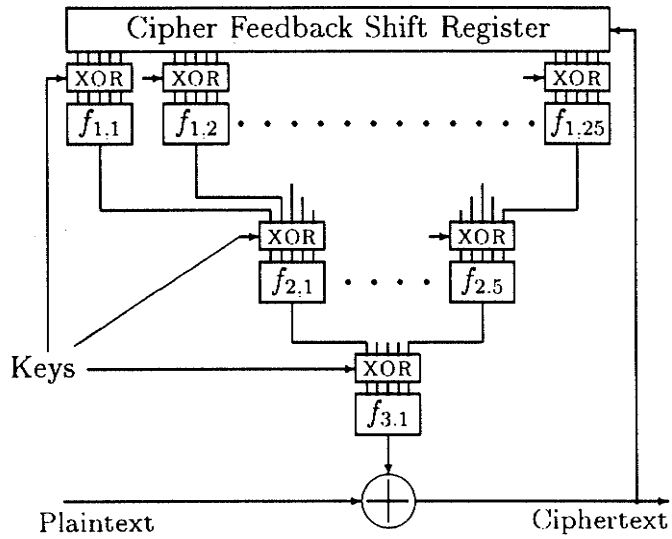

Figure 1: Kühn's Cipher

2.2 Kühn's Cipher

An example of a tree-structured cipher was proposed by Kühn in [9], and is shown in Figure 1. The boolean function of Kühn's cipher is a complete 5-ary tree of depth three. It uses the last 125 ciphertext bits as input to the 31 cell function. The cipher keying is achieved by masking the input to each cell with a 5-bit subkey. Each bit of the key is exclusive-ored with a single bit at the input to one of the cells. The key length is 155 bits. Each cell in the cipher performs the same 5-bit function

$$f(x_1, x_2, x_3, x_4, x_5) = x_1 + x_2 + (x_1 + x_3)(x_2 + x_4 + x_5) + (x_1 + x_4)(x_2 + x_3)x_5 \quad (1)$$

which was chosen to be 1-0 balanced and 1^{st}-order correlation immune [11]. The full network function is 7^{th}-order correlation immune.

3 Anderson's Attack

In this section, Anderson's attack [2] is described, which is general enough to be applied to any tree-structured cipher. The attack does not make any

assumptions about the basic cell function, or the method of keying, so that each cell could be selected from the set of all m-bit functions. The attack has two stages: stage 1 isolates each cell (except the last), and stage 2 replaces each cell with a look-up table (LUT). The resulting network of tables correctly mimics the input-output mapping of the cipher, for the key in use at the time the attack is made. The key is hidden implicitly in the look-up tables obtained. A very efficient algorithm for Anderson's attack was presented by Heys and Tavares in [5] and [6].

Stage 1 of the attack is probabilistic, and against Kühn's cipher requires an average of 240 chosen inputs. Stage 2 is deterministic, and requires 992 chosen inputs (32 inputs for each of the 31 cells). The full attack can be expected to require 1232 chosen inputs each of 125 bits. In general, the attack requires

$$(8 + 2^m)\left(\frac{m^R - 1}{m - 1}\right) - 8$$

chosen inputs. For an arbitrary network, the attack requires $O(2^m m^{R-1})$ chosen inputs.

4 The Improved Attack

This section presents a modification to Anderson's attack which takes advantage of the known cell function from [9] to bypass construction of the LUTs and recover each subkey directly. Kühn's cipher is based on a cell function which was selected to satisfy certain essential cryptographic criteria. One undesirable property of this function, which has been previously overlooked in the literature, is that it has a linear structure [4]. The output of Kühn's function remains unchanged when the first four input bits are simultaneously complemented:

$$f(\vec{X}) = f(\vec{X} + 11110). \tag{2}$$

so that the information content of each subkey is reduced from five bits to four. This is true for each of the 31 cells in the network, so that the total effective key length is actually 124. Each 155-bit key used is cryptographically equivalent to $2^{31} - 1$ other keys.

Since the look-up tables of Anderson's attack can appear in one of two complemented forms, let the *standardised* look-up table be such that each of

its elements has been XORed with the look-up table value for the zero input. A standardised look-up table defines a partition of the possible cell input-values into two sets: those with a table entry 0, and those with a table entry 1. The subkey mask uniquely determines the partition, since the cell functions are fixed. For any given cell function, one-to-one mappings exist between the input masks, partition transformations, and the standardised look-up tables. The partition can be precomputed for each of the cryptographically distinct input masks, so that the subkey can be found explicitly from the standardised look-up table.

Table 1. shows the standardised truth tables obtained for Kühn's function when the input has been masked with the indicated subkey. Because of the linear structure in (2), a standard form of subkey notation has been adopted here in which the first bit is set to 0 and the other four bits constitute the effective subkey. The first row of this table shows the unmodified truth table for Kühn's function, given the natural binary representation order along the horizontal (inputs 0..31).

Given that, in general, the input mask contains only m bits of information, it is possible to determine the subkey using $O(m)$ chosen inputs. In the case of Kühn's function it is possible to dynamically choose four inputs which will uniquely determine the subkey. Let the first two chosen inputs be 1 and 12, which are 00001 and 01100 in binary. This choice allows the cryptanalyst to reduce the number of subkeys to a set of four candidates. The next two inputs to choose are decided by the standardised result of input 12. There are several equally effective choices, one of which is presented here. If $r(12) = 0$ then the next two chosen inputs should be 2 and 4. If $r(12) = 1$ then choose 3 and 5. The subkey of the cell under scrutiny can now be found from Table 2.

In general, the improved stage 2 of the attack requires m chosen inputs per cell, rather than the original 2^m. With stage 1 unaltered, the full attack can now be expected to require

$$(8+m)\left(\frac{m^R-1}{m-1}\right)-8$$

chosen inputs to complete. If Kühn's cipher is implemented using a 5-bit sub-function without a linear structure, then the improved attack is expected to require an average of 382 chosen inputs, compared to the 1232 inputs required

Subkey	Standardised Truth Table
00000	00000111111001101001101111010000
00001	00001011110110010110011111100000
00010	00001101101110010110111001110000
00011	00001110011101101001110110110000
00100	01110000011011101011100100001101
00101	01001111011000101000100111110001
00110	00101111011001000001100111111000
00111	00011111100110000010011011110100
01000	00011001111110000010111101100100
01001	00100110111101000001111110011000
01010	01000110111100101000111110010001
01011	01110110000011101011000010011101
01100	01101110011100000000110110111001
01101	01100010010011111111000110001001
01110	01100100001011111111100000011001
01111	01100111111000000000101111011001

Table 1: Standardized Truth Tables for Kühn's Function when Input Masked

$r(1), r(12)$	$r(2), r(4)$	Subkey	$r(1), r(12)$	$r(3), r(5)$	Subkey
0,0	0,0	0	0,1	0,0	1
	0,1	3		0,1	2
	1,0	9		1,0	8
	1,1	6		1,1	7
1,0	0,0	10	1,1	0,0	13
	0,1	5		0,1	14
	1,0	15		1,0	4
	1,1	12		1,1	11

Table 2: Full Subkey Identification for Kühn's Function

by the original attack. The linear structure further reduces the number of inputs required by 31 (one per cell) to 351.

It should be noted that the fast attack on Kühn's cipher described above can be applied to any tree-structured cipher in the case where the underlying boolean functions for the cells are known. This includes SPNs.

5 Random Permutations in Tree-structured Networks

One way which has been suggested in [10] to improve the design of Kühn's original cipher is to insert a random bit position permutation between rows 1 and 2. This conceals the vital set of any cell in a row lower than the permutation, so that these cells cannot be isolated in the same way as in Section 2.1. Note that the tree-structure of the network is preserved, so that the subkeys of the cells above the permutation can be found. The cryptanalyst can effectively make chosen inputs to the permutation. The output of the permutation is the input to five cells, so that the permutation partitions the 25 input bits into five sets of five positions each. The method now presented will reveal the five set partition of the bit positions. The original look-up table attack can then proceed, since the vital sets of all cells will be known. Each look-up table can then be determined as before, this time replacing the cell-function, input mask and 5-bit permutation all at once.

Bit position i is said to be *active* for input $\vec{I}$ in boolean function f if $f(\vec{I}) \neq f(\vec{I} + e_i)$, where e_i is the unit vector with 1 in position i and 0 elsewhere. Any bit position which is not active for some input is said to be *idle* for that input. Each 25-bit input to the permutation can be associated with an *activity set*, which is the set of all bit positions which are active for that input. An activity set can be determined with a group of 26 chosen inputs (the input itself and the other 25 at Hamming distance one from it). The activity sets of two inputs at Hamming distance one can be found with 50 chosen inputs. For Kühn's cipher, it turns out that the way these two activity sets intersect provides a lot of information about the permutation. This is possible due to an interaction between the tree-structure of the network and properties of Kühn's function relating to the intersection of activity sets

which are associated with inputs at Hamming distance one.

5.1 Activity Sets in Kühn's Function

It is a universal property of bit activity that, given two inputs which differ in a single bit, the activity type of that bit is the same for the two inputs. Whether a particular bit position is active or idle will depend on the contents of the rest of the input vector (the other $m-1$ bits). When flipped, an active bit will remain active, and an idle bit will remain idle.

In general, whenever an input bit is flipped producing a new input, the associated activity set is changed. A function can be characterised by the constraints on possible activity sets, and the manner in which the activity sets change in response to a single input bit change. The activity set changes for Kühn's function have been fully explored, show interesting properties of symmetry, and will now be presented.

Every 5-bit input to Kühn's function has either 2 or 3 members in its associated activity set, with equal frequency. Activity sets of other sizes do not occur for this function, although sizes from 0 to 5 would be possible in general for 5-bit functions. Whenever a single input bit is changed, the associated activity set will change, in accordance with the state transition diagram of Figure 2. In Figure 2a, the arrows indicate possible transitions between activity sets. In Figure 2b, those activity sets with similar transition behaviour are grouped together. Within the group, each set is an equally frequent outcome of transitions to that group, given that all inputs are equally likely. The arrows in Figure 2b are labeled with the probability of those transitions for the complementation of a randomly chosen bit.

For every transition to or from a set with two members, exactly one element will be in common. Every transition between 3-member sets replaces one member, leaving exactly two members the same. These rules holds regardless of the activity type of the bit flipped. A related property of the transitions is that in every case either exactly one bit is lost from the active set, or exactly one bit is added to the active set, or both. The fact that the size and contents of the activity set change in such a regular manner has been exploited in developing the following method for partially solving the permutation.

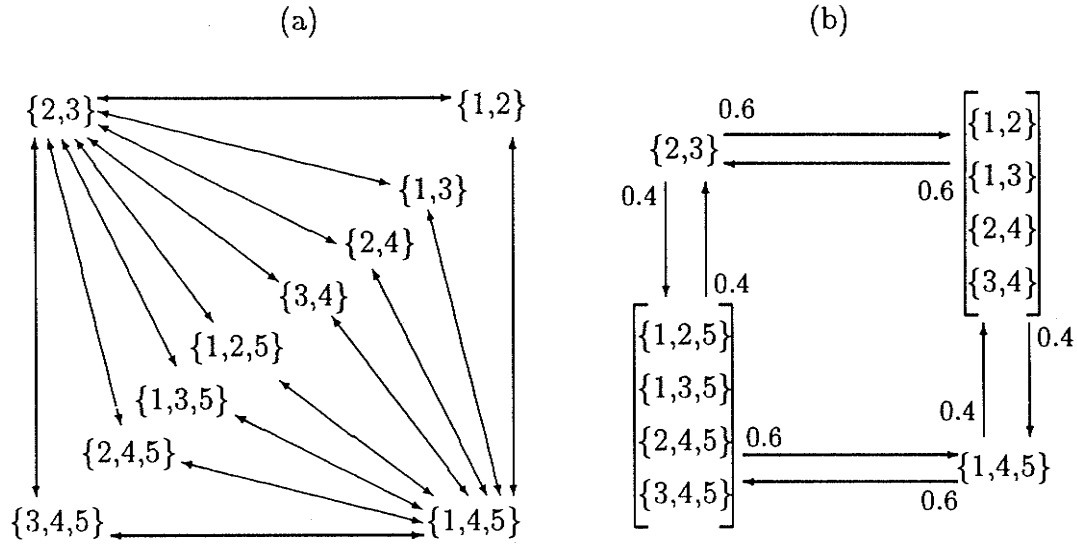

Figure 2: Activity Set Transitions for Kühn's Function

5.2 Finding a Partial Solution of the Permutation

After being permuted, a 25-bit input will induce some 5-bit value at the input to the last cell, which (for Kühn's sub-function) will make two or three of the five row 2 cells active. All of the elements of the active set A, associated with a 25-bit input $\vec{I}$, will be bit positions which lead to one of these active cells. Each of these cells will have two or three *locally active* input positions which are active for that cell. The size $n(A)$ of the active set A, will therefore be bounded by $4 \leq n(A) \leq 9$.

Let A_0 and A_1 be the two active sets associated with inputs $\vec{I}_0$ and $\vec{I}_1$ which differ only in one place (the *flipped bit F*). Let $S = A_0 \cap A_1$, $L = A_0 - S$ and $G = A_1 - S$. It follows that the sets L, S, G are mutually disjoint.

On its own, a single cell has two types of input bits: active and idle. Expanding to a tree-structured network, each additional row of cells doubles the number of input bit types. In Kühn's cipher, two rows occur after the permutation, so there are four types of input bits: active and idle bits each for active and idle cells.

The correct inference about the permutation is determined by the number of elements in the sets L and G. Each interpretation is uniquely associated with a particular type of flipped bit. Note that at the time the flipped bit

is selected, its type is not known. Once the two activity sets are found, the number of elements in the sets L and G will be known, and then the type of flipped bit can be inferred. This allows the correct interpretation to be made, and a 25 by 25 table can be further filled in, reflecting the increased knowledge about the permutation.

The form of activity set change is distinct for each of the types of bits chosen to be flipped. The cases with their effects are:

(i) The flipped bit is a locally idle bit of an idle cell. Whenever the activity set does not change, the cryptanalyst knows that the bit chosen to be flipped must lead to a different cell than all bits in the active set.

(ii) The flipped bit is a locally idle bit of an active cell. The active set will loose and gain only bits which lead to the same cell, which is the cell led to by the flipped bit.

(iii) The flipped bit is a locally active bit for an idle cell. The set of active cells is changed, with the restriction that the cell led to by the flipped bit remains idle. The result is that the bits lost from the active set, those bits gained by the active set, and those bits which remained in the active set, must all lead to different cells, all of which are different cells to that led to by the flipped bit.

(iv) The flipped bit is a locally active bit in an active cell. This is the case of flipping a bit in the active set of the initial arbitrary input. This case alters both the set of active cells and the set of active bits in a cell which remains active. This case can be confused with the above two cases, so it is avoided by choosing to restrict the flipped bit to those *not* in the active set of the initial arbitrary input.

Given case (iv) is prohibited, the other three cases can be identified and interpreted as follows. If the active set does not change (case (i)), then we conclude that the flipped bit leads to a cell different from that of any bit in the active set. If $n(L) = 1$ or $n(G) = 1$ or both, then we conclude that the bit flipped F, the bit(s) lost L, and the bit(s) gained G, must all lead to the same cell. This corresponds to case (ii) above. Any other resulting change in the elements of the active set corresponds to case (iii): each bit which is an element of any one of the sets L, S, G, F must lead to a cell different to each of the bits in the other three sets.

With these simple interpretations, a 25 by 25 table can be filled in. Any element (i, j) in the table can take on one of three values representing the

known relation between the bit positions i and j. Of course, by symmetry, $(i,j) = (j,i)$ for all i and all j. The table can be initialised by setting all $(i,j) = 0$, indicating no knowledge. The information that bit positions i and j lead to the same cell is represented as $(i,j) = 1$. Finally, $(i,j) = -1$ can indicate that bit positions i and j lead to different cells. Each of the three cases will allow several of the table elements to be set at 1 or -1 as applicable. The table will be complete when each row contains either five $1s$ or twenty $-1s$.

The table can be filled in faster by using the transitive property of the 'leads to same cell' relation. In other words, if we know already that $(a,b) = 1$ and we now discover that $(b,c) = 1$, then we can immediately state that $(a,c) = 1$. The opportunity for this type of inference should be checked every time a new table element is set from 0 to 1. As the table becomes almost full, the contribution from this inference will become greater than the contribution from new groups of chosen inputs.

This attack on the permutation has been implemented on PC and found to require an average of 1350 chosen inputs to complete.

6 Conclusion

Ciphers based on tree-structured boolean functions are not secure. Although tree-structures make hardware implementation simple, a fundamental weakness is introduced which allows a divide-and-conquer attack to be made. The cipher can be mimicked or fully broken, depending on the information available to the cryptanalyst. Insertion of a random permutation provides some protection from cell-replacement attacks, however one example is known where the permutation can be partialy unravelled to allow the attack to proceed. This unravelling process exploited both the tree structure and properties of the particular sub-function. Further research is required to assess the general applicability of these methods to ciphers with other forms of sub-function selection, or which are merely similar to trees in structure.

7 References

[1] C. Adams, A Formal and Practical Design Procedure for Substitution-Permutation Network Cryptosystems, *PhD. Thesis*, Queens University, Kingston, Ontario 1990

[2] R.J. Anderson, Tree Functions and Cipher Systems, *Cryptologia* Vol. 15, No. 3, July 1991, pp.194-202.

[3] R.J. Anderson, Use of Idle Variables in Cryptanalysis, *unpublished manuscript.*

[4] D. Chaum, J.-H. Evertse, Cryptanalysis of DES with a reduced number of rounds; Sequences of Linear Factors in Block Ciphers, *Advances in Cryptology: Proc. Crypto '85*, Lecture Notes in Computer Science 218, Springer Verlag, 1986, pp.192-211.

[5] H.M. Heys and S.E. Tavares, Chosen Plaintext attacks on Tree-Structured Substitution-Permutation Networks, *Technical Report*, Queens University, Kingston, Ontario, 16 Sept 1992.

[6] H.M. Heys, S.E. Tavares, Cryptanalysis of Tree-Structured Substitution-Permutation Networks, *Electronics Letters*, Vol. 29, No. 1, 7 Jan. 1993, pp.40-41.

[7] J.B. Kam and G.I. Davida, Structured Design of Substitution-Permutation Encryption Networks, *IEEE Transactions on Computers*, Vol. C-28, No. 10, Oct. 1979, pp.747-753.

[8] D.E.Knuth, Fundamental Algorithms: The Art of Computer Programming, Vol. 1. Addison-Wesley.

[9] G.K. Kühn, Algorithms for Self-Synchronizing Ciphers, *Comsig 88, Southern Africa Conference on Communications*, pp.159-164.

[10] G.K. Kühn, F.J. Brouwer, W. Smit, A Fast Multipurpose Encryption Chip, *Infosec '90 Symposium, CSIR, Pretoria*, 16 March 1990.

[11] T. Siegenthaler, Correlation Immunity of Nonlinear Combining Functions for Cryptographic Applications, *IEEE Transactions on Information Theory*, Vol. 34, No. 3, 1988, pp.569-571

[12] A.F. Webster, S.E. Tavares, On the Design of S-Boxes, *Advances in Cryptology: Proc. Crypto '85*, Lecture Notes in Computer Science 218, Springer Verlag, pp. 523-534.

Cryptographic Degradation of DES in Block and Stream Cipher Modes in a Digital Mobile Communication Link

Jean-Yves Chouinard and Guy Ferland
Department of Electrical Engineering, University of Ottawa
161 Louis-Pasteur, Ottawa, Ontario, Canada, K1N 6N5
Email: chouinar@trix.genie.uottawa.ca

Abstract

This paper presents a performance analysis of different block and stream cipher modes of operation of the DES (Data Encryption Standard) encryption in a digital mobile radio channel environment. Each DES mode offers a different level of cryptographic protection, but also leads to a different cryptographic degradation. The objective here is to determine the degradation effects of data enciphering on the channel reliability and to analyze the error correction capability of BCH (Bose-Chaudhuri-Hocquenghem) error correcting codes. The digital mobile communication channel is computer simulated using Fritchman's error burst channel model. It is shown that the performance of BCH codes depends strongly on the burst error length distribution and also on the error propagation properties of the specific DES mode used. Bit interleaving allows for a more efficient use of the error control code.

1 Introduction

To protect sensitive data transfers against passive or active attacks, it is necessary to encrypt the data before transmission over the communication channel. The DES is a cryptographic algorithm which can be used for low to medium security applications. High data encipherment throughput rates are achievable and inexpensive hardware to implement it is readily available. In a digital channel, DES can be operated in block cipher and stream cipher modes, each one providing a different level of cryptographic protection. Unfortunately, enciphering of data often results in a significant increase in the bit error rate (BER) at the

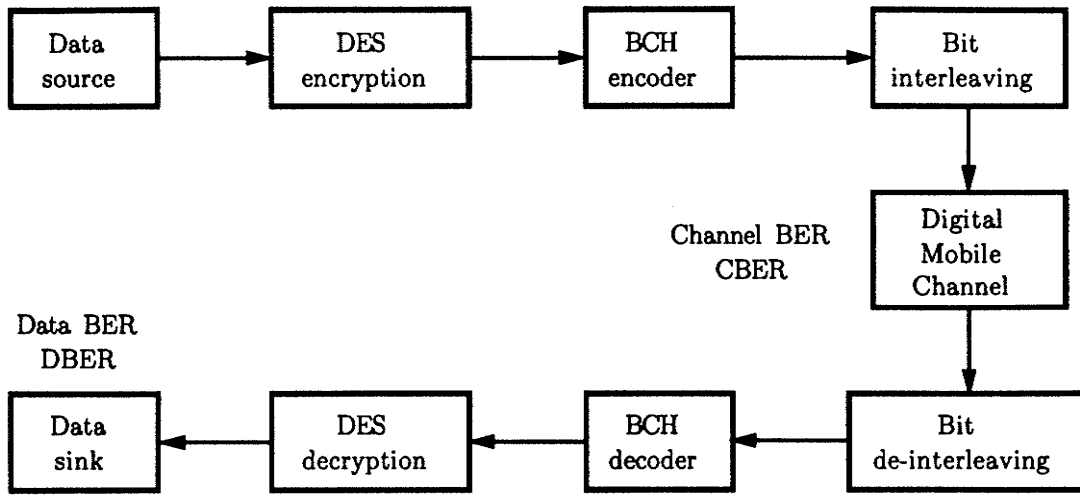

Figure 1: Block diagram of DES secure communication link

receiver, due to error propagation, which is inherent in most encryption modes. This additional cryptographic degradation of the channel reliability is usually unacceptable in a mobile communication channel where the signal already suffers from a relatively high BER caused by multipath fading. To reduce the BER at the receiver to acceptable levels, Forward Error Correction (FEC) techniques can be implemented to detect and correct channel errors. The objective here is to determine the cryptographic degradation of DES encipherment on the data bits at the receiver and to analyze the ability of BCH block codes to correct errors, without and with bit interleaving. To avoid confusion concerning the BER, the following definitions are used hereafter: $DBER$ is defined as the (information) data bit error rate at the receiver after FEC decoding and decryption, whereas $CBER$ represents the channel bit error rate and refers to the number of bits that are corrupted by the channel.

2 Digital Mobile Communication Link

Figure 1 illustrates the block diagram of a communication link in which DES encryption/decryption and BCH channel coding are used. The mobile communication channel is characterized by fast fading of the signal power envelope at the receiver [15] due to multipath propagation caused by scatterers nearby the mobile. The drop in signal power can be in the order of 20 to 30 dB [14]. This periodic variation in the signal strength results in time intervals during which the channel bit error rate ($CBER$) is low, followed by usually shorter intervals when the $CBER$ can be higher by many orders of magnitude. To represent the behavior of a digital channel with memory, many models have been proposed over the years [10, 12, 13].

The channel model proposed by Fritchman [10] can represent a wide variety of bursty channel conditions. It consists of a Markov chain with N states $\{S_1, \ldots, S_N\}$ which is partitioned into two groups of states: states S_1 to S_k represent error-free states while the remaining states, S_{k+1} to S_N, are error states. Fritchman has proposed a simplified version of the channel model where the Markov chain consists of $(N-1)$ error-free states and one error state, i.e. S_N. Transitions are only allowed to the same state (e.g., from S_i to S_i, for $1 \leq i \leq N$), from the good states to the bad one, and vice-versa (i.e., from S_i to S_N and S_N to S_i). The simplified Fritchman channel model can still accurately describe the error distributions observed in most digital channels with memory [4, 21]. The error distribution is determined by the channel transition probability's matrix $\mathbf{P} = \{P_{i,j}\}$, where $P_{i,j}$ indicates the transition probability from the state S_i to the state S_j. The error-free interval length distribution $P(0^m|1)$ can be expressed as a sum of exponential functions:

$$P(0^m|1) = \sum_{i=1}^{N-1} \frac{P_{N,i}}{P_{i,i}} (P_{i,i})^m \qquad \text{for } m \geq 1 \tag{1}$$

The error-free interval length distribution $P(0^m|1)$ gives the probability of having m or more consecutive error-free bits, provided that an error had occurred. The transition probability matrix, $\mathbf{P}$, controls the persistence of the states, and therefore, determines the distribution of errors.

3 Modes of Operation of DES and Cryptographic Degradation

The Data Encryption Standard constitutes an interesting encryption approach for applications requiring low to medium levels of security because of the high throughput rates achievable (up to 100 $Mbits/s$). DES is a substitution-permutation encryption algorithm [17] which encrypts blocks of 64 plaintext bits with a 56-bit key into a 64-bit ciphertext block through a set of substitution and permutation transformations [1, 6]. In a digital communication channel, DES can be implemented in either block cipher modes (i.e., Electronic Codebook (ECB) and Cipher Block Chaining (CBC) modes), where the plaintext data bits are encrypted as 64-bit data blocks; or in stream cipher modes (i.e., Output Feedback (OFB) and Cipher Feedback (CFB) modes) for which the data bits are enciphered individually. Reference [18] provides a complete description of DES modes of operation. The following discussion on cryptographic error degradation in the different modes of DES is based on material readily available in the literature: the reader is referred to [20, 6, 19] for further information.

In the ECB mode, a message stream is broken into blocks of 64 data bits and, then, each block is encrypted as a unit, $C_i = DES_K(M_i)$, and transmitted over the channel. The

receiver simply decrypts the received cipher block C_i', where $C_i' = C_i \oplus E_i$, with the same key (the element e_i of the 64-bit error vector E_i is 1 when the channel introduces a bit error, or 0 otherwise). If $C_i' = C_i$, i.e., no errors, then:

$$DES_K^{-1}(C_i') = DES_K^{-1}(C_i \oplus E_i) = DES_K^{-1}(C_i) = M_i \tag{2}$$

Because of the complex relationship of all data bits within the encrypted block, a single bit in error within a block will invariably corrupt about half of the 64 message bits: bit errors propagate within the block in which they occur.

In CBC mode, the encryption/decryption of a block of data is made dependent on the previously transmitted block. The plaintext block is added to the previous cipher block, that is $C_i = DES_K(M_i \oplus C_{i-1})$, prior encryption. At the receiver, C_i' is decrypted and then added to the previous cipher block C_{i-1}'.

$$DES_K^{-1}(C_i') \oplus C_{i-1}' = DES_K^{-1}(C_i \oplus E_i) \oplus (C_{i-1} \oplus E_{i-1}) = M_i \tag{3}$$

if $C_i' = C_i$ and $C_{i-1}' = C_{i-1}$. In general, j bits received in error within a block of 64 bits will corrupt $(64 + j)$ bits, that is, errors propagate within the block in which they occur and in the next 64-bit block. An advantage of CBC is that the chaining of adjacent data blocks prevents the insertion, deletion or replacement of cipher blocks [19], whereas ECB is vulnerable to these attacks.

In the OFB stream cipher mode, the generated bit stream added to the data bits (to form the cipher) is independent of the data bits being encrypted. At the transmitter, bit $c_i = m_i \oplus k_i$, where k_i is a function of the 64 DES output bits:

$$k_i = f_1\left[DES_K(k_{i-64}, k_{i-63}, \ldots, k_{i-1})\right] \tag{4}$$

where $f_1(\bullet)$ indicates that only one of the 64 bits generated by DES, k_i, is used while the other bits are discarded. A bit received in error $c_i' = c_i \oplus e_i$ is decrypted incorrectly at the receiver since $m_i' = c_i' \neq c_i \oplus k_i$. External bit synchronization is required between the transmitter and receiver. However, OFB is the only cipher mode without any inherent error propagation.

In CFB mode, bits are still encrypted individually but not independently of each other. At the transmitter, bit $c_i = m_i \oplus k_i$ where k_i is now a function of the 64 *previous ciphertext bits*:

$$k_i = f_1\left[DES_K(c_{i-64}, c_{i-63}, \ldots, c_{i-1})\right] \tag{5}$$

A erroneous bit c_i' will be decrypted incorrectly. Furthermore, since the key stream bits k_{i+1} up to k_{i+64} at the receiver are functions of the corrupted bit c_i', the next 64 consecutive bits will be affected by that one error. Because of the diffusion property of DES, on average, about half of those 64 bits will be erroneous. The cipher feedback mode is a self-synchronous stream cipher: if bit synchronization is lost, this is equivalent to a bit error: once the erroneous digit is cleared from the receiver memory, correct deciphering operation resumes. As for ECB and CBC block modes, CFB is also resistant to bit tampering, while OFB is vulnerable to these attacks.

4 Channel Control Coding Simulation Results

A series of computer simulations has been done to assess the performances, in terms of data bit error rate (i.e., $DBER$) versus the channel bit error rate ($CBER$), for digital channels when DES encryption is used to protect the information. Figure 1 shows the components of the digital mobile communication link under consideration. For the simulations, Fritchman's error bursts' channel model is studied to determine the cryptographic degradation of the block and stream cipher modes of DES. Six different sets of channel parameters are used: these are based on actual measurements carried out in Quebec city [21]. Cryptographic degradation and information bit error rate $DBER$ are estimated over the range [9]: $10^{-1} \leq CBER \leq 10^{-3}$. Figure 2 show that, under identical channel and coding conditions, the ranking in terms of the data bit error rate (i.e., $DBER$), obtained with the four DES modes of operation is, from the lowest to highest $DBER$: OFB, ECB, CFB and CBC. Figure 3 shows the error rate obtained with a $BCH(31, 16, 3)$ triple error correcting code over the Fritchman's channel. The ranking, from the lowest to highest $DBER$, is the same as before. Note that, in this particular case, the grouping of errors into high error density clusters leads to almost no improvement with this triple error correcting code due to the relatively low correcting capability of the BCH code.

To reduce the data bit error rate $DBER$ caused by DES cryptographic degradation, Forward Error Correction codes, such as BCH block codes [2, 16], can be used, where k bits of enciphered data are encoded in a vector of n bits ($n > k$) prior to transmission over the channel, allowing the receiver to correct up to t errors within the n bit block. The error correcting capability t increases as the number of parity bits $(n - k)$ increases, but this results in a decrease in the actual code rate, $R \equiv k/n$, hence reducing the real throughput of information across the digital communication channel. The $BCH(n, k, t)$ codes are chosen such that the code rate $R \approx 75\%, 50\%$ or 33% for different codeword blocklenghts. Figure 4 shows the $DBER$, as a function of $CBER$, when DES is used in the ECB mode, for BCH codes having the same code rate $R \approx 50\%$, for different blocklengths n. It is observed that an increase in n leads to a smaller data bit error rate. As shown for the CBC mode of operation (see Figure 5), given a fixed blocklength $n = 63$, an increase in the error correcting capability t of the BCH code, and therefore a reduction in its code rate R, decreases the $DBER$ as expected.

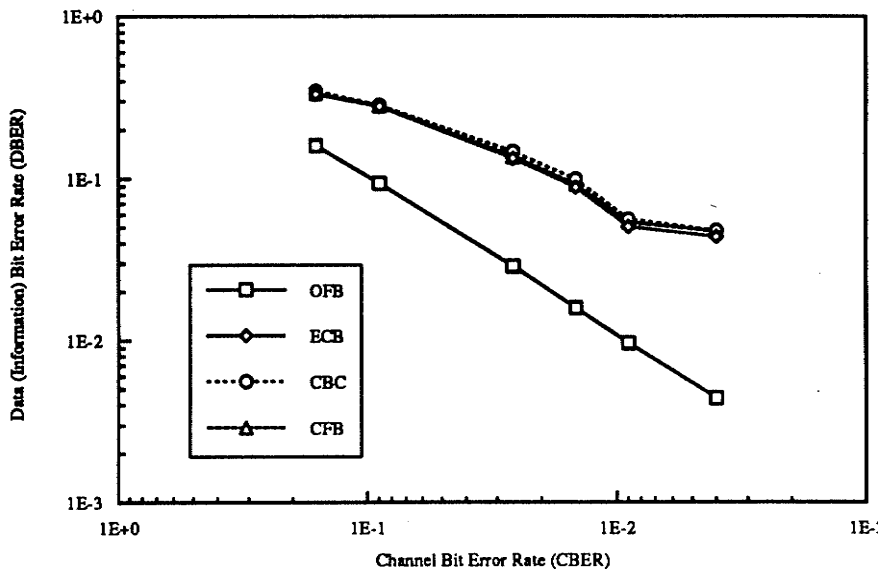

Figure 2: BER performance without coding (four DES modes).

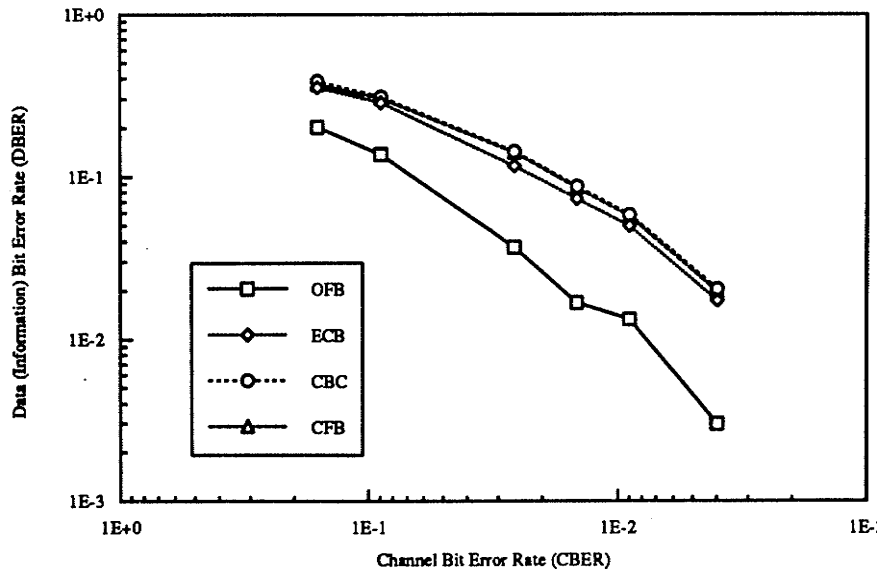

Figure 3: BER performance over a Fritchman channel for the four DES modes with a $BCH(31, 16, 3)$ block code

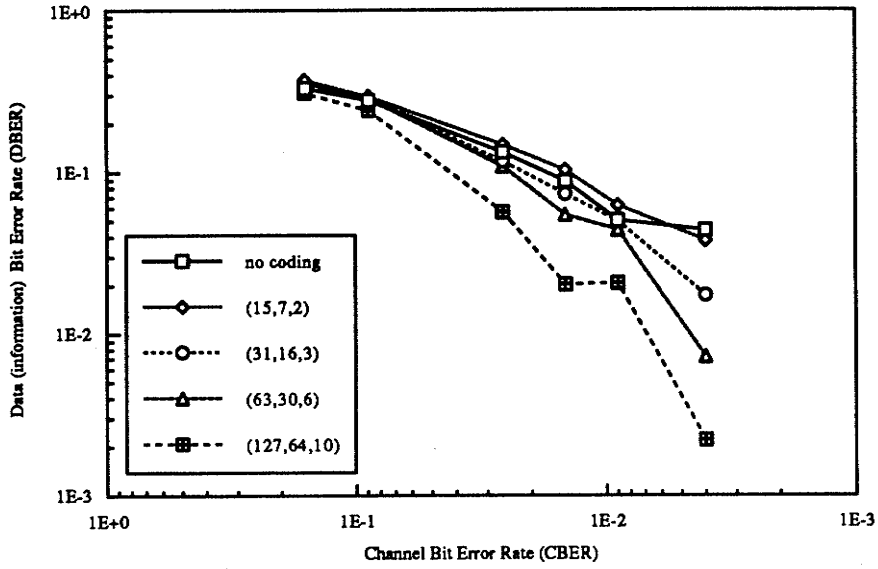

Figure 4: $BCH(n, k, t)$ code performance with $R \approx 50\%$ (ECB mode).

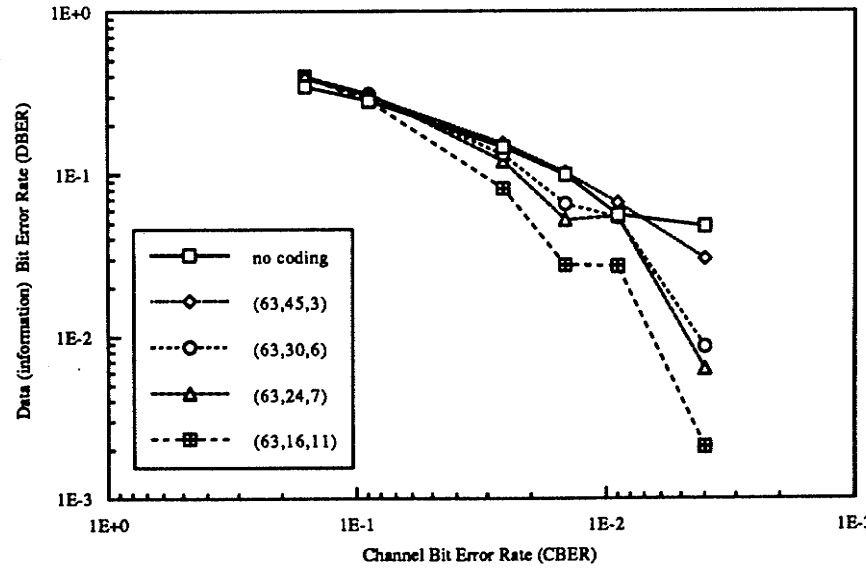

Figure 5: $BCH(63, k, t)$ code performance with (CBC mode).

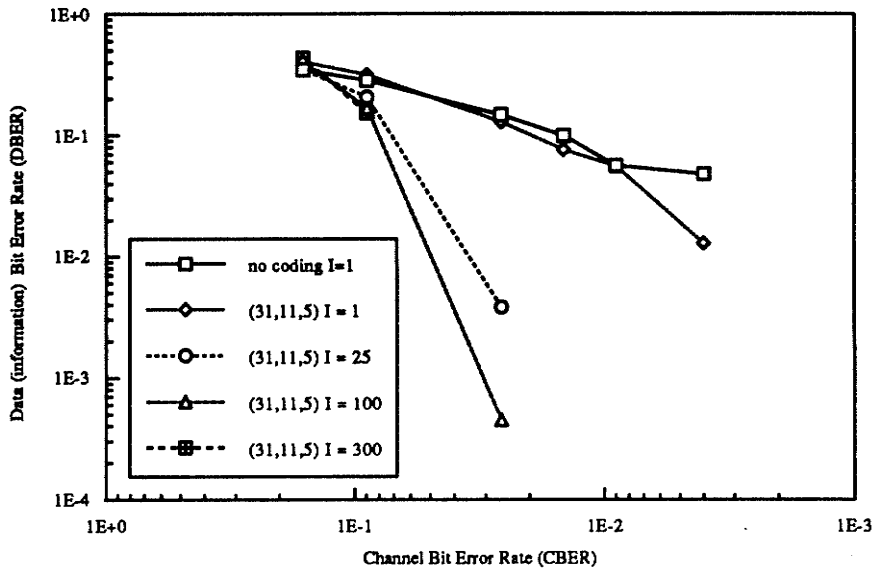

Figure 6: $BCH(31, 11, 5)$ code performance with bit interleaving (CBC mode).

In some circumstances, the use of BCH codes may result in a $DBER$ which is higher than if no coding had been used. The average number of codewords N_c required [8] to encode a complete block of N encrypted bits ($N_c \approx \lceil N/k \rceil$) has a significant impact on the performance of the BCH code: given two BCH codes (n, k_1, t_1) and (n, k_2, t_2) of the same length n, where $R_1 > R_2$, it is possible for the code with highest rate to outperform the other one when the former results in a lower value of N_c than the latter. For instance, even if a $BCH(127, 64, 10)$ code is less powerful in terms of error correcting capability t than a $BCH(127, 57, 11)$ code, an incorrectly decoded 127-bit codeword will affect only a single 64-bit ciphertext block for the $BCH(127, 64, 10)$ code while affecting the decipherment of one or two 64-bit ciphertext blocks for the $BCH(127, 57, 11)$ code.

BCH channel coding provides better results when the channel is corrupted by random errors. By applying bit interleaving on I consecutive $BCH(n, k, t)$ codewords, the errors in bursts are redistributed more evenly over the $(I \times n)$ bits [16]. For channels with memory, bit interleaving is effective if, the error correcting capability t of the error control code is sufficient to correct most channel errors after deinterleaving, and if, the interleaving degree I is large enough to reduce the memory effect of the channel. However, a systematic delay of $(I \times n)$ bits is introduced; for large values of interleaving degrees I and blocklengths n, this delay may be unacceptable for real-time processing applications. The interleaving degrees I employed for the simulations are $I = 1$ (i.e., no interleaving), 25, 100, and 300. Figure 6 shows the results obtained for the CBC block mode with the $BCH(31, 11, 5)$ code. The

largest interleaving degree provides the lowest data bit error rate, since t is greater than the channel error density.

5 Concluding Remarks

Simple encipherment methods, such as the Data Encryption Standard, can offer protection of data against wiretapping at high data rates for digital mobile communication applications. The four different modes of operation of DES, Electronic Codebook, Cipher Block Chaining, Output Feedback and Cipher Feedback, provide different levels of cryptographic protection of information. CBC constitutes the most secure protection scheme among them. However, the use of DES encryption in the ECB, CBC and CFB modes leads to a cryptographic degradation that translates into a significant bit error rate increase in the data, while no error propagation occurs in the OFB mode. On the other hand, OFB requires bit synchronization. Error control coding techniques, such as BCH codes, reduce the actual information error rate, provided that their error correcting capability is larger than the error density. As shown, the performance of BCH codes is strongly dependent on the burst error length distribution. For channels with memory, channel errors can be spread more evenly in time by interleaving the encoded bits before transmission and then deinterleaving the corrupted received bit stream, thus making a more efficient use of the error control code capability. The bit error rate performance with bit interleaving depends, however, on the actual error correcting capability of the error correction code and, thus, its code rate. The tradeoff is a systematic interleaving delay that may prove unacceptable for real-time mobile applications such as digital speech transmission.

References

[1] H. Beker, F. Piper, *"Cipher Systems: The Protection of Communications"*, Northwood Books, London, 1982.

[2] R.E. Blahut, *"Theory and Practice of Error Control Codes"*, Addison-Wesley, Reading, Massachusetts, 1984.

[3] R.E. Blahut, *"Principles and Practice of Information Theory"*, Addison-Wesley, Reading, Massachusetts, 1987.

[4] J.-Y. Chouinard, M. Lecours, G.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.

[5] D.W. Davies and W.L. Price, *"Security for Computer Networks"*, John Wiley and Sons, 1984.

[6] D.E.R. Denning, *"Cryptography and Data Security"*, Addison-Wesley, Reading, Massachusetts, 1983.

[7] G. Ferland, J.-Y. 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.

[8] G. Ferland, *"Error Rate Performance Analysis of Stream and Block Ciphers in a Digital Mobile Communication Channel"*, M.A.Sc. thesis, University of Ottawa, Ottawa, Canada, July 1993.

[9] G. Ferland, J.-Y. Chouinard, *"Performance of BCH Codes with DES Encryption in a Digital Mobile Channel"*, Proceedings of the 1993 Canadian Workshop on Information Theory, to appear in the Lecture Notes in Computer Science Series by Springer-Verlag, 1994.

[10] B.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.

[11] R.G. Gallager, *"Information Theory and Reliable Communication"*, John Wiley and Sons, 1968.

[12] E.N. Gilbert, *"Capacity of a Burst-Noise Channel"*, Bell System Technical Journal, September 1960, pp. 1253-1265.

[13] L.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.

[14] 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, No. 4, November 1988, pp. 204-212.

[15] W.C. Lee *"Mobile Communications Engineering"*, McGraw-Hill, 1982.

[16] S. Lin, D.J. Costello, *"Error Control Coding: Fundamentals and Applications"*, Prentice-Hall, New Jersey, 1983.

[17] NBS (National Bureau of Standards), *"Data Encryption Standard"*, Federal Information Processing Standards, Publication 46, US Department of Commerce, Washington DC, January 1977.

[18] NBS (National Bureau of Standards), *"DES Modes of Operation"*, Federal Information Processing Standards, Publication 81, US Department of Commerce, Washington DC, December 1980.

[19] C.P. Pfleeger, *"Security in Computing"*, Prentice-Hall, 1989.

[20] B. Schneier, *"Applied Cryptography: Protocols, Algorithms, and Source Code in C"*, John Wiley and Sons, 1994.

[21] A. Semmar, M. Lecours, J.-Y. Chouinard, J. 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.

[22] C.E. Shannon, *"Communication Theory of Secrecy Systems"*, Bell System Technical Journal, Vol. 28, October 1949, pp. 656-715.

[23] D.J. Torrieri, *"Principles of Secure Communication Systems"*, first edition, Artech House.