Resistance of a CAST-Like Encryption Algorithm to Linear and Differential Cryptanalysis

J. Lee[†], H. M. Heys[‡] and S. E. Tavares[†]

[†]Department of Electrical and Computer Engineering
Queen's University
Kingston, Ontario, Canada K7L 3N6
email: tavares@ee.queensu.ca

[‡]Faculty of Engineering and Applied Science
Memorial University of Newfoundland
St. John's, Newfoundland, Canada A1B 3X5
email: howard@engr.mun.ca

Abstract

Linear cryptanalysis and differential cryptanalysis are two recently introduced, powerful methodologies for attacking private-key block ciphers. In this paper, we examine the application of these two cryptanalysis techniques to a CAST-like encryption algorithm based on randomly generated s-boxes. It is shown that, when randomly generated substitution boxes (s-boxes) are used in a CAST-like algorithm, the resulting cipher is resistant to both the linear attack and the differential attack.

1 Introduction

As the need for privacy and authentication is now generally recognized by the telecommunications community, a widely adopted private-key encryption algorithm is becoming an increasingly important objective in the development and analysis of cryptographic algorithms. For some time, the Data Encryption Standard (DES) [16] has been the most widely used and trusted encryption algorithm. However, DES is about twenty years old and has recently become vulnerable to cryptanalysis due to its small key size. In addition, DES was designed explicitly for fast hardware implementation, making the current extensive use of DES in software rather anomalous.

The most successful attack on DES to date is an exhaustive key search machine that can be built for about one million U.S. dollars and which can find a DES key in about 3.5 hours [20]. In recent years, DES has also been subjected to two powerful attacks known as linear cryptanalysis [11] and differential cryptanalysis [4]. Although these attacks have been very successful against many encryption algorithms such as FEAL [19] and Khafre [14], DES has been resistant, in a practical sense, to both attacks. Nonetheless, it seems inevitable that we need an alternative to DES which can be easily implemented in software, has a long enough key, a fast encryption/decryption rate, and is resistant to known attacks. The CAST encryption algorithm [1, 3] was developed with these objectives in mind.

Similarly to DES and other proposed block ciphers, the CAST algorithm consists of a series of rounds of substitutions in order to achieve the "confusion" and "diffusion" principles suggested by Shannon [18]. In CAST, the substitutions are accomplished using large $m \times n$ mappings, referred to as s-boxes, which have m input bits and n output bits such that $m < n$. The large s-boxes are implemented to efficiently eliminate the permutations found in DES between rounds of substitutions and, as a result, CAST is a very efficient algorithm for software implementations [3]. CAST is currently employed in several computer security products [17].

In this paper, we consider a CAST-like cipher constructed with randomly generated s-boxes and examine the resistance of such a cipher to both linear cryptanalysis and differential cryptanalysis. The resulting analysis suggests that random s-boxes can be used effectively to create a CAST-like cipher resistant to both types of attacks.

2 Structure of CAST

The initial implementation of CAST [1, 3] is a 64-bit private-key block cipher with a 64-bit key which encrypts by using a number of rounds consisting of 8×32 s-boxes. The flow of data between consecutive rounds in CAST is similar to that of DES. Both algorithms implement a round function F which operates on the right half of the data block. The output of F is XORed bit-by-bit with the left half of the data block to produce a new left half-block and then the left and right half-blocks are swapped. An R-round algorithm is illustrated in Figure 1 and may be viewed as the following iterated operation:

$$L_i = R_{i-1} \tag{1}$$

and

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i) \tag{2}$$

for $1 \leq i < R$ where R_i and L_i represent the right and left half-blocks, respectively, and K_i represents the sub-key associated with round i. In the last round, we have

$$R_R = R_{R-1} \tag{3}$$

and

$$L_R = L_{R-1} \oplus F(R_R, K_R). \tag{4}$$

CAST and DES differ significantly in the implementation of the round function F. In DES, the round function F expands 32 bits of input data to 48 bits using an expansion table E. The expanded data is then XORed with 48 key bits and fed into eight 6×4 s-boxes.

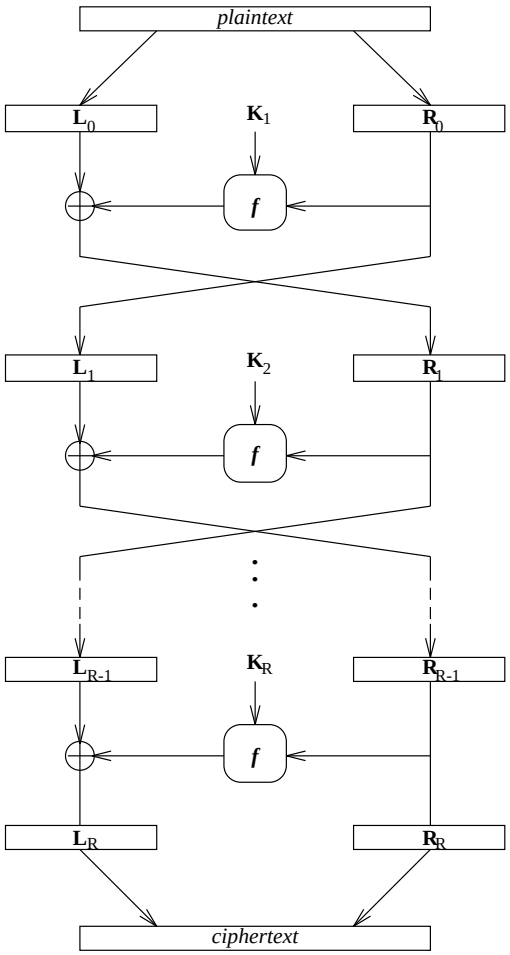

Figure 1: Enciphering Algorithm of CAST

The output of the 8 s-boxes are concatenated together and then permuted according to a permutation function P to form the 32 output bits of F. In CAST, the round function XORs 32 bits of input data with 32 key bits and feeds the result into four 8×32 s-boxes. The 32 output bits of the four s-boxes are XORed together to form the 32 output bits of F. The round functions of DES and CAST are shown in Figure 2. The original version of CAST [1, 3] uses s-boxes based on a class of highly nonlinear boolean functions referred to as "bent" functions [13]. As a result, the generation of the s-boxes is complicated and it

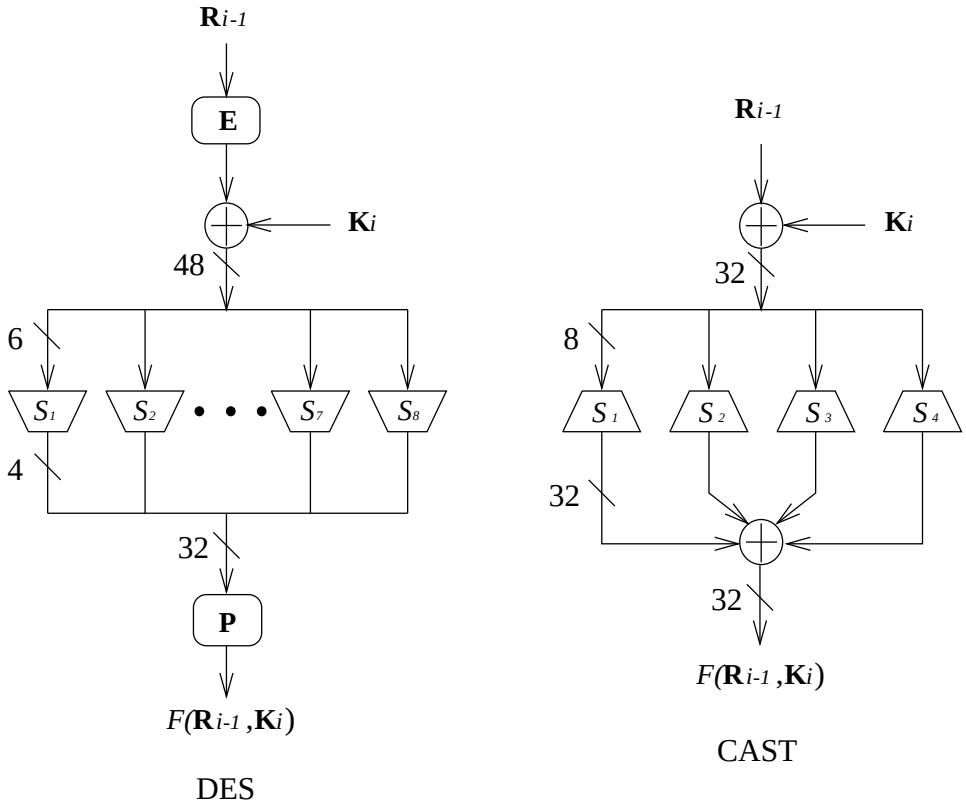

Figure 2: Round Functions of DES and CAST

is difficult to analyze the security of the cipher. In addition, a key scheduling algorithm is used to assign keys to the various rounds [2]. In this paper, to simplify the analysis and promote a simple s-box generation procedure, we consider a CAST-like algorithm which uses randomly generated s-boxes instead of s-boxes generated from bent functions and uses independent keys in each round of substitution.

3 Linear Cryptanalysis of the CAST-like Cipher

In this section, following the development of [6], we examine the resistance of the CAST-like encryption algorithm to linear cryptanalysis. In particular, we establish a relationship between the probability that a linear approximation of a given number of rounds of a CAST-

like cipher holds and the minimum nonlinearity of the s-boxes used in the construction of the round function.

3.1 Application of Linear Cryptanalysis

In [11] and [12], Matsui describes the first known plaintext attack that can break the full 16-rounds of DES faster than exhaustive search. The attack described by Matsui is known as linear cryptanalysis and was experimentally shown to break DES using 2^{43} known plaintexts [12].

The fundamental principle of linear cryptanalysis is to find a linear approximation that relates subsets of plaintext bits, ciphertext bits, and key bits in the following manner:

$$P_{i_1} \oplus P_{i_2} \oplus \cdots \oplus P_{i_a} \oplus C_{j_1} \oplus C_{j_2} \oplus \cdots \oplus C_{j_b} = K_{k_1} \oplus K_{k_2} \oplus \cdots \oplus K_{k_c} \tag{5}$$

where $i_1, i_2, \ldots, i_a, j_1, j_2, \ldots, j_b$ and $k_1, k_2, \ldots, k_c$ denote fixed bit positions of the plaintext P, ciphertext C, and key K, respectively.

Suppose p_l is the probability that equation (5) holds. The effectiveness of the linear approximation depends on the magnitude of $|p_l - \frac{1}{2}|$. If $|p_l - \frac{1}{2}|$ is large enough and sufficient plaintext-ciphertext pairs are known, it is possible to determine one equivalent key bit in the form of the XOR sum of the key bits on the right-hand-side of equation (5) as the value that most often satisfies the equation.

In general, a linear approximation of the cipher as represented by equation (5) is formed by combining a number of s-box linear approximations for different rounds such that any terms that do not involve plaintext bits, ciphertext bits, or key bits are cancelled. Suppose that equation (5) is formed by combining α s-box linear approximations and that the best s-box linear approximation has a probability p_β (i.e., the magnitude $|p_\beta - \frac{1}{2}|$ is the largest

among all the α s-box linear approximations). If we assume that the inputs to the s-boxes involved in the linear approximation are independent and uniformly distributed random variables, from [11] it follows that

$$|p_l - \frac{1}{2}| \lesssim 2^{\alpha-1} \cdot |p_\beta - \frac{1}{2}|^\alpha \ . \tag{6}$$

It is also shown in [11] that the number of known plaintexts in a linear attack is inversely related to $|p_l - 1/2|$ and, hence, the number of known plaintexts required in the analysis can be increased by selecting s-boxes such that $p_\beta \rightarrow \frac{1}{2}$ and by increasing the number of s-box linear approximations, α, involved in the overall approximation.

Let us denote an m-bit affine boolean function as

$$t(X) = a_0 \oplus a_1 X_1 \oplus \ldots \oplus a_m X_m \tag{7}$$

where $X = [X_1 \ldots X_m]$ represents the m-bit input and $a_i \in \{0,1\}$, $0 \leq i \leq m$. The distance between two m-bit boolean functions, s and t, can be defined to be

$$d(s,t) = \sum_{X \in \{0,1\}^m} [s(X) \oplus t(X)] \, . \tag{8}$$

The nonlinearity of an m-bit boolean function s is defined in the following way:

$$NL(s) = \min_{t \in \mathcal{A}} d(s,t) \tag{9}$$

where $\mathcal{A}$ is the set of all m-bit affine boolean functions. The definition of nonlinearity can be extended to an $m \times n$ s-box S as follows [15]:

$$NL(S) = \min_{\substack{a_1, a_2, \ldots, a_n \in \{0,1\} \\ \exists i, a_i \neq 0}} NL(\bigoplus_{i=1}^{n} a_i s_i) \tag{10}$$

where s_i is the m-bit boolean function of the i-th output bit of the s-box S. Hence, the nonlinearity of an $m \times n$ s-box is just the minimum nonlinearity over all non-zero linear combinations of the n m-bit boolean functions of the s-box.

Consider a cipher that consists of s-boxes of size $m \times n$ and assume that each s-box in the cipher has a nonlinearity greater than or equal to NL_{min}. It then follows that the best s-box linear approximation has a probability p_β where

$$|p_\beta - \frac{1}{2}| = \frac{2^{m-1} - NL_{min}}{2^m} \ . \tag{11}$$

From Figure 2, an output bit of the round function F of CAST is the XOR sum of the corresponding output bit of all 4 s-boxes. Consequently, since each output bit depends on the input from all 4 s-boxes, a linear approximation of F must involve linear approximations of all 4 s-boxes. If one of the s-boxes is not included in the approximation, then the output bit from that s-box that is used in the XOR to determine the output of the round function will occur randomly with respect to the linear approximation and the approximation will hold with a probability of 1/2, negating its usefulness in a linear attack. Referring to Figure 3, if the values of L_{i-1} and R_{i-1} in round i are known, a linear approximation of L_{i+1} must involve a linear approximation of the output of the F function in round i since $L_{i+1} = L_{i-1} \oplus F_i$. This means that the 4 s-boxes in round i must be involved in the linear approximation. Similarly, it is not difficult to see that a linear approximation of R_{i+1} must involve at least the 4 s-boxes of round $i+1$. Regardless of whether both L_{i+1} and R_{i+1} or just one of them is involved in the linear approximation, a 2-round linear approximation must involve at least 4 s-boxes. Since an r-round linear approximation must involve at least as many s-boxes as $r/2$ iterations of the best 2-round approximation, the number of s-boxes

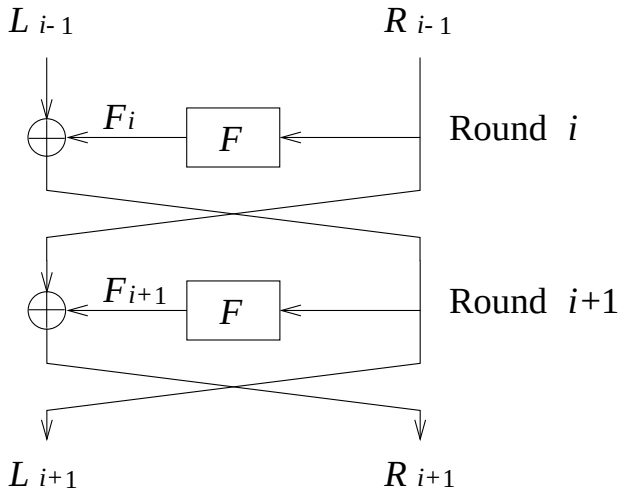

Figure 3: Flow of Data in Round i and Round $i+1$

involved in an r-round linear approximation is at least $\alpha = (r/2) \cdot 4 = 2r$. Putting $\alpha = 2r$ in equation (6) results in the following estimate:

$$|p_l - \frac{1}{2}| \lesssim 2^{2r-1} |p_\beta - \frac{1}{2}|^{2r} . \tag{12}$$

If 8×32 s-boxes with $NL_{min} = 64$ are used in a cipher, the best s-box linear approximation for such a cipher will have probability p_β such that $|p_\beta - \frac{1}{2}| = \frac{2^8 - 1 - 64}{2^8} = \frac{1}{4}$. Consequently, using (12), it is possible to construct Table 1 which shows the upper bound on the value of $|p_l - 1/2|$ for an r-round linear approximation of a CAST-like cipher that uses 8×32 s-boxes with $NL_{min} = 64$. As a comparison, the corresponding value of $|p_l - 1/2|$ for an r-round linear approximation of DES [11] is also shown in the table. [1] Note that the linear approximation of the CAST-like cipher occurs with a probability much closer to $1/2$ than the linear approximation of DES for the same number of rounds. Moreover, the

[1] One must be cautious in drawing conclusions about the immunity of a cipher to linear cryptanalysis based on an upper bound on the value of $|p_l - 1/2|$ since it is still conceivable that multiple linear approximations may be combined to effectively attack a cipher [7]. However, clearly it is a desirable objective to minimize $|p_l - 1/2|$ for the linear approximations of a cipher and, therefore, $|p_l - 1/2|$ is a valid metric with which to compare the performances of different ciphers.

Number of Rounds r	$\lvert p_l - 1/2 \rvert$	
	CAST	**DES**
8	2^{-17}	1.22×2^{-11}
12	2^{-25}	1.19×2^{-17}
16	2^{-33}	1.49×2^{-24}

Table 1: $|p_l - 1/2|$ for Linear Approximation of CAST-like Cipher (with s-boxes having $NL_{min} = 64$) and DES

numbers in Table 1 for the CAST-like cipher are just an upper bound. In practice, it is likely that a linear approximation of a CAST-like cipher will be much closer in probability to 1/2.

3.2 Selecting S-boxes of High Nonlinearity

In the previous section, we considered a CAST-like cipher using 8×32 s-boxes with a minimum nonlinearity greater than or equal to 64. For a general $m \times n$ s-box this corresponds to a nonlinearity of 2^{m-2}. In this section, we examine the likelihood of randomly selecting an s-box S of size $m \times n$ such that its nonlinearity $NL(S) < 2^{m-2}$.

The number of ways of choosing two m-bit boolean functions, s and t, so that the distance between them, $d(s,t)$, equals 2^{m-2} is given by

$$\mathcal{N}(d = 2^{m-2}) = \begin{pmatrix} 2^m \\ 2^{m-2} \end{pmatrix} \tag{13}$$

Also it is not difficult to show that, in general, $\sum_{i=0}^{k/4-1} \begin{pmatrix} k \\ i \end{pmatrix} < \begin{pmatrix} k \\ k/4 \end{pmatrix}$. Hence, the number of ways of selecting s and t so that $d(s,t) < 2^{m-2}$ is bounded as

$$\mathcal{N}(d < 2^{m-2}) < \mathcal{N}(d = 2^{m-2}) \, . \tag{14}$$

Since any two m-bit linear functions have a distance of exactly 2^{m-1} from each other, the events of a function s having a distance of less than 2^{m-2} from two different linear functions

are mutually exclusive. Also, since there are 2^{m+1} m-bit affine functions, we can get a bound on the number of ways to select a function with a nonlinearity of less than 2^{m-2}:

$$\mathcal{N}(NL(s) < 2^{m-2}) < \rho \tag{15}$$

where $\rho = 2^{m+1} \cdot \mathcal{N}(d = 2^{m-2})$.

Consider the modulo-2 sum of k output functions of an s-box. We may consider that a particular $k-1$ of the functions in the sum are chosen independently and there are $2^{(k-1)2^m}$ ways to choose these functions. The remaining function can then be chosen so that the nonlinearity of the sum is less than 2^{m-2} and the number of ways this can be done is upper bounded by ρ. Since there are k ways to choose which function will be used to satisfy the nonlinearity bound, an upper bound on the number of ways of selecting k functions so that their sum satisfies a nonlinearity of less than 2^{m-2} is given by $k \cdot \rho \cdot 2^{(k-1)2^m}$. There are $\left(\begin{array}{c} n \\ k \end{array} \right)$ ways of selecting the k bits from the n output functions of the s-box. Hence, summing over all values of k, $1 \leq k \leq n$, and dividing by the number of ways to choose n random m-bit boolean functions gives an upper bound on the probability that an s-box S has a nonlinearity less than 2^{m-2}:

$$P(NL(S) < 2^{m-2}) < \rho \cdot \sum_{k=1}^{n} \left[k \left(\begin{array}{c} n \\ k \end{array} \right) 2^{(k-1)2^m} \right] / 2^{n2^m} \tag{16}$$

By applying Stirling's approximation of $k! \approx \sqrt{2\pi k} \cdot (\frac{k}{e})^k$, it can be shown that, for 8×32 s-boxes, the right side of (16) evaluates to less than 2^{-33} and, hence, $P(NL(S) < 64) < 2^{-33}$. Hence, the probability of randomly generating an s-box with a nonlinearity of less than 64 is very unlikely. In recent experiments, no randomly generated 8×32 s-boxes were found for which $NL(S) < 72$ [21].

4 Differential Cryptanalysis of the CAST-like Cipher

In this section, we examine the resistance of the CAST-like encryption algorithm to differential cryptanalysis. An analytical model for the distribution of entries in the XOR difference distribution table for the round function is first developed. Based on the distribution of entries, we can form highly probable characteristics. In particular, we focus on finding the best iterative characteristic that is applicable to the CAST-like cipher.

4.1 Distribution of Entries in the XOR Table

Differential cryptanalysis is a chosen plaintext attack which makes use of the highly probable occurrences of sequences of output XOR differences at each round given a particular plaintext XOR difference. The foundation for the differential attack is the ability to predict the output XOR difference of the round function F given the knowledge of the input XOR difference to that round. Information on the likelihood of possible output XOR values given particular input XOR values is available in an XOR difference distribution table [4]. In the XOR table, each row corresponds to a particular input XOR value, each column corresponds to a particular output XOR value, and the entries themselves represent the number of possible input/output pairs corresponding to the input and output XOR values. The XOR table can be used to determine the probability that a particular output XOR will occur given an input XOR. We refer to this as the difference probability and it is derived by dividing the value of the entry in the XOR table by the number of possible output XORs. By concatenating highly probable XORs together, one can construct a highly probable sequence of differences referred to as a characteristic. The higher the likelihood of

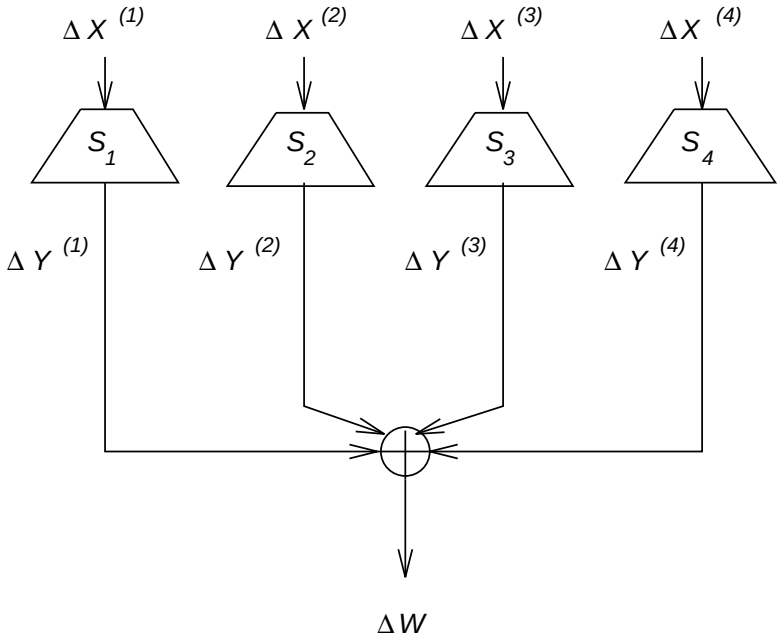

Figure 4: Flow of Data in Round Function of CAST

a characteristic, the fewer chosen plaintexts required for differential cryptanalysis. [2]

In reference to the CAST-like cipher, consider a set of four 8×32 randomly generated s-boxes where we denote the 8-bit inputs to the 4 s-boxes as $X^{(1)}, X^{(2)}, X^{(3)}$, and $X^{(4)}$, and the corresponding outputs of the 4 s-boxes as $S_1(X^{(1)}), S_2(X^{(2)}), S_3(X^{(3)})$, and $S_4(X^{(4)})$. Let $\Delta X^{(i)}$ represent the bitwise XOR of two values for $X^{(i)}$. For the complete 32-bit round function, given an input XOR value $\Delta X = [\Delta X^{(1)}, \Delta X^{(2)}, \Delta X^{(3)}, \Delta X^{(4)}]$, the output XOR

[2] In a strict sense, differential cryptanalysis only requires the existence of highly probable differentials where a differential refers to an output XOR difference after a particular round given a plaintext XOR difference [9]. This differs from a characteristic since a characteristic specifies the exact output XOR difference of each round required to achieve the final output XOR. In practice, it is difficult for a cryptanalyst to discover the existence of a highly probable differential without examining the cipher for highly probable characteristics. Hence, it is appropriate to consider the likelihood of the existence of highly probable characteristics since, without them, discovery of highly probable differentials is unlikely. It should noted, however, that an upper bound on the probability of a characteristic is not an upper bound on the probability of a differential and cannot be used to prove that the cipher cannot be attacked.

value, ΔW, in Figure 4 is given by the following equation:

$$\Delta W = \bigoplus_{i=1}^{4} \Delta Y^{(i)} \tag{17}$$

where $\Delta Y^{(i)} = S_i(X^{(i)}) \oplus S_i(X^{(i)} \oplus \Delta X^{(i)})$.

Let $wt(\cdot)$ represent the Hamming weight of the specified argument and define the following functions:

$$f(\Delta X^{(i)}) = \begin{cases} 0 & \text{if } wt(\Delta X^{(i)}) = 0 \\ 1 & \text{if } wt(\Delta X^{(i)}) \neq 0 \end{cases} \tag{18}$$

and

$$g(\Delta X) = \sum_{i=1}^{4} f(\Delta X^{(i)}). \tag{19}$$

Thus, $g(\Delta X)$ is the number of s-boxes that have non-zero XOR inputs when ΔX is applied to F. Consider that an entry in the XOR table of the round function corresponding to input XOR value $\Delta X = [\Delta X^{(1)}, \Delta X^{(2)}, \Delta X^{(3)}, \Delta X^{(4)}]$ and output XOR value ΔW is given as the following sum of products:

$$\begin{aligned} \sum_{a} \sum_{b} \sum_{c} [&\#\{X^{(1)} | \Delta Y^{(1)} = a\} \times \#\{X^{(2)} | \Delta Y^{(2)} = b\} \\ &\times \#\{X^{(3)} | \Delta Y^{(3)} = c\} \times \#\{X^{(4)} | \Delta Y^{(4)} = \Delta W \oplus a \oplus b \oplus c\}] \end{aligned} \tag{20}$$

Since each of the product terms must be an even number (or zero), each of products must be a multiple of 16 (or zero). If $\Delta X^{(i)} = 0$, $\Delta Y^{(i)} = 0$ for all values of $X^{(i)}$ and the corresponding term in the product is 2^8. Thus, the XOR table contains zeroes or multiples of $2^{32-7g(\Delta X)}$.

Since ΔW is 32 bits long, it can assume at most 2^{32} distinct values. However, each $\Delta Y^{(i)}$ can assume at most 2^7 values for a particular $\Delta X^{(i)}$. This occurs because, for a fixed

$\Delta X^{(i)}$, there will be $2^8/2 = 2^7$ unordered pairs of $(X^{(i)}, X^{(i)} \oplus \Delta X^{(i)})$. If each of these pairs gives rise to a distinct value for $\Delta Y^{(i)}$, then $\Delta Y^{(i)}$ can take at most 2^7 distinct values.

Since the output vectors of the s-boxes are randomly generated, the values obtained by the XOR sum of the $\Delta Y^{(i)}$'s will also be randomly distributed. This results because the j-th bit of the output XOR, ΔW_j, is just the XOR sum of the j-th bit of the four s-box output XORs, $\bigoplus_{i=1}^{4} \Delta Y_j^{(i)}$. Since the output bits are randomly generated, it follows that each output XOR bit of an s-box has an equal chance of being 0 or 1. Assuming independence between the output XOR bits of different s-boxes, the XOR sum of the j-th bit of the four s-box output XORs will also have an equal chance of being 0 or 1. Consequently, one can conclude that ΔW may assume any one of the 2^{32} possible values with equal probability.

In fact, as the possible values of ΔW can be found by trying all the $2^{7g(\Delta X)}$ different combinations of $\Delta Y^{(1)} \oplus \Delta Y^{(2)} \oplus \Delta Y^{(3)} \oplus \Delta Y^{(4)}$, the distribution of output XORs for a given input XOR is equivalent to tossing $2^{7g(\Delta X)}$ balls randomly into 2^{32} bins with each ball having a weight of $2^{32-7g(\Delta X)}$. We wish to determine the distribution of the balls in the bins.

Let Λ_k be a random variable representing the number of bins having k balls when N balls are being tossed randomly into M bins. It has been shown that for large N and M [5],

$$E[\Lambda_k] \approx M \frac{e^{-\frac{N}{M}}}{k!} (\frac{N}{M})^k . \tag{21}$$

For $M = 2^{32}$ and $N = 2^{28}$, $E[\Lambda_k]$ will be the expected number of ΔW values that have XOR entries of 16*k for a particular ΔX when $g(\Delta X) = 4$. By dividing E[Λ_k] by M, one can get the expected fraction of ΔW values that have XOR entries of 16*k. For choices of ΔX such that $g(\Delta X) = 3$, the corresponding entries in the XOR table are multiples of

$g(\Delta X)$	**Entry Value**	**% of Entries**
4	0	93.94
4	16	5.87
4	32	0.183
4	48	$3.83 * 10^{-3}$
4	64	$5.97 * 10^{-5}$
4	80	$7.47 * 10^{-7}$
4	96	$7.78 * 10^{-9}$
4	112	$6.94 * 10^{-11}$
⋮	⋮	⋮
3	0	99.95
3	2048	0.0488
3	4096	$1.19 * 10^{-5}$
3	6144	$1.94 * 10^{-9}$
3	8192	$2.37 * 10^{-13}$
⋮	⋮	⋮

Table 2: Expected Distribution of Entry Value in XOR Table for $g(\Delta X)$=4 and $g(\Delta X)$=3

2048 and, in this case $N = 2^{21}$. A summary of the results for $g(\Delta X) = 4$ and $g(\Delta X) = 3$ is listed in Table 2.

For ΔX's which have $g(\Delta X) = 2$ or 1, the corresponding non-zero entries in the XOR table will be multiples of 2^{18} and 2^{25}, respectively. In both cases, the probability that a non-zero entry is not 2^{18} or 2^{25}, respectively, is negligible. Finally, for the trivial case where $g(\Delta X) = 0$ (i.e., $wt(\Delta X) = 0$), there is an entry of magnitude 2^{32} for the column corresponding to $\Delta W = 0$ and the entries are zero for all the other columns.

Note that as the value of $g(\Delta X)$ decreases, the corresponding magnitudes of the entries in the XOR table will increase and it is extremely unlikely for an XOR table corresponding to $g(\Delta X) = i$ to have non-zero entries that are greater than those of an XOR table for $g(\Delta X) = j$ where $j < i$.

4.2 Iterative Characteristics

Input XOR differences of zero to the round function F always lead to output XOR differences of zero with a probability 1. This is called the 1-round trivial characteristic. If such a trivial characteristic appears in every k rounds of encryption and the plaintext is equal to the ciphertext after k rounds of encryption, then we say we have an k-round iterative characteristic.

4.2.1 2-Round Iterative Characteristics

Let Φ represent a particular 32-bit XOR vector. The flow of data in a 2-round iterative characteristic is shown in Figure 5 and has the following form [8]:

$(\Phi, 0)$	[input]
$0 \leftarrow 0$ with probability 1	[round 1]
$0 \leftarrow \Phi$ with probability p	[round 2]
$(\Phi, 0)$	[output]

where the elements in brackets represent the left and right half XOR values, respectively, and the arrow represents the mapping of the round function F.

Define the probability $q = \mathrm{P}(h(\Delta X, 0) = 16 \mid g(\Delta X) = 4)$ where $h(\Delta X, \Delta W)$ is the entry in the XOR table corresponding to an input XOR value of ΔX and a value of ΔW for the output XOR. Since ΔW is randomly distributed among the 2^{32} possible values, for a particular ΔX with $g(\Delta X) = 4$, one can use Table 2 to predict that $q = 0.0587$.

Let $\mu(h, g)$ represent the expected number of rows in the XOR table that will have the entry value $h(\Delta X, 0)$ given $g(\Delta X)$. Note that the number of ΔX such that $g(\Delta X) = 4$ is

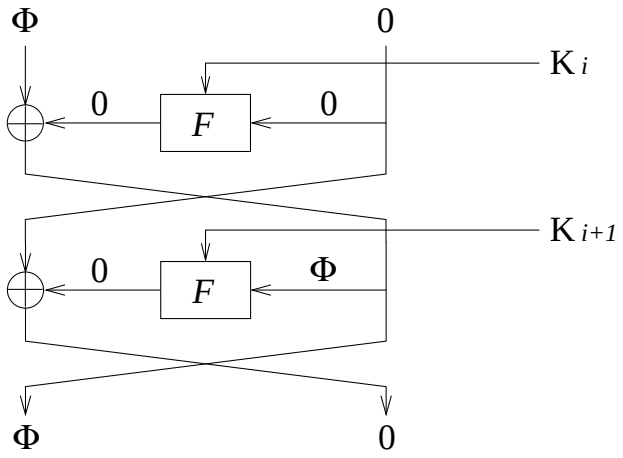

Figure 5: 2-round Iterative Characteristic

$(2^8-1)^4$. Thus, for $g(\Delta X)=4$ and $h(\Delta X,0)=16$, $\mu(h,g)=q*(2^8-1)^4\approx 2.5*10^8$ if we assume that the occurrence of $h(\Delta X,0)=16$ for different ΔX's are independent. An entry of 16 in the XOR table means that the corresponding difference probability p will be $\frac{16}{2^{32}}=2^{-28}$ and the resulting probability for the 2-round characteristic is $p_{\Omega_2}=p=2^{-28}$.

Using a similar analysis for $g(\Delta X)=3$, 2 and 1, one can get the values listed in Table 3. Note that it is highly unlikely to find an s-box with which to construct a 2-round iterative characteristic so that $g(\Delta X)=1$ and $p=2^{-7}$. In fact, it would not be difficult to select only injective s-boxes thereby preventing the occurrence of a 2-round iterative characteristic based on a difference with a probability of $p=2^{-7}$. However, it is very likely that an s-box will exhibit a 2-round iterative characteristic with $g(\Delta X)=2$ and $p=2^{-14}$. Therefore, we shall assume that the probability per round for the best 2-round iterative characteristic is $(2^{-14})^{\frac{1}{2}}=2^{-7}$.

4.2.2 Iterative Characteristics with more than 2 Rounds

According to [8], the flow of data in a 3-round iterative characteristic is as follows:

$g(\Delta X)$	$h(\Delta X, 0)$	μ	p
4	16	$2.5*10^{8}$	2^{-28}
4	32	$7.8*10^{6}$	2^{-27}
4	48	$1.6*10^{5}$	$2^{-26.4}$
4	64	$2.5*10^{3}$	2^{-26}
4	80	32	$2^{-25.7}$
4	96	$3.3*10^{-1}$	$2^{-25.4}$
⋮	⋮	⋮	⋮
3	2048	$3.2*10^{4}$	2^{-21}
3	4096	7.9	2^{-20}
3	6144	$1.3*10^{-3}$	$2^{-19.4}$
⋮	⋮	⋮	⋮
2	2^{18}	1.5	2^{-14}
2	2^{19}	$2.8*10^{-6}$	2^{-13}
⋮	⋮	⋮	⋮
1	2^{25}	$3.0*10^{-5}$	2^{-7}
⋮	⋮	⋮	⋮

Table 3: Likelihood of Occurrence of 2-round Iterative Characeristic

$(\Gamma, 0)$	[input]
$0 \leftarrow 0$ with probability 1	[round 1]
$\Phi \leftarrow \Gamma$ with probability p_1	[round 2]
$\Gamma \leftarrow \Phi$ with probability p_2	[round 3]
$(\Phi, 0)$	[output]

where Φ, Γ represent 32-bit XOR vectors. Although the inputs and outputs are not the same, one can concatenate this 3-round characteristic with itself interchanging Φ and Γ to get a 6 round iterative characteristic such that the inputs and the outputs will be the same.

Since the probability per round for the best 2-round iterative characteristic is 2^{-7}, a 3-round iterative characteristic needs to have $p_{\Omega_3} = p_1 \cdot p_2 > 2^{-21}$ in order to be more useful

than the 2-round iterative characteristic. [3]

A 3-round iterative characteristic is made up of three 1-round characteristics. In section 4.1, it was shown that the entries in the XOR table depend on the values of $g(\Delta X)$. The most likely maximum entry when $g(\Delta X) = 1$ is 2^{25} and hence the highest one-round difference probability that is likely to occur in such a case is $\frac{2^{25}}{2^{32}} = 2^{-7}$. Similarly, as suggested by the results of Table 3, the highest one-round difference probability that is likely to occur when $g(\Delta X) = 2$, 3 or 4 will be less than or equal to 2^{-14}. Hence, in order for $p_{\Omega_3} = p_1 \cdot p_2 > 2^{-21}$, we only need to consider the case when $p_1 = 2^{-7}$ and $p_2 = 2^{-7}$. This means that $g(\Gamma) = 1$ and $g(\Phi) = 1$. If the XOR input value of the round function F is denoted by ΔX and the output XOR value by ΔW, we can denote $g(\Delta W)$ as the number of s-boxes in the next round that have non-zero XOR inputs when ΔW is used as the XOR input. Consider $\mathcal{E}$ to be the event the XOR table contains a value of ΔW for which $g(\Delta W) = 1$ given that $g(\Delta X) = 1$. Then it can be shown that, using the assumption of independence between rows in the XOR table, $P(\mathcal{E}) = 3.1 * 10^{-2}$ [10]. Hence, the probability of an s-box having a Φ and Γ so that they can be used in round number 2 of the 3-round iterative characteristic is no greater than $3.1 * 10^{-2}$. Therefore, s-boxes which cannot be used in the 3-round iterative characteristics are plentiful and it would be easy to apply a screening process on the s-boxes, taking time in $O(2^m)$, to ensure that event $\mathcal{E}$ does not

[3] Note that our analysis does not consider that multiple characteristics may be used in a differential attack by combining the chosen plaintexts into a structure such as a quartet or octet [4] . Using such a structure reduces the number of chosen plaintexts and, hence, if multiple 3-round characteristics were used where $p_1 \cdot p_2 = 2^{-21}$, it might be possible to mount a more efficient differential attack than an attack using a characteristic based on a 2-round iterative characteristic. However, it is also conceivable that multiple 2-round characteristics may be combined in a structure to reduce the number of plaintexts required in the attack. In either case, while the number of chosen plaintexts required in the attack is reduced, it is a relatively modest reduction. For example, an quartet structure based on 3 characteristics can be used to reduce the number of chosen plaintexts by 2/3. Hence, we shall only be interested in comparing the relative likelihoods of individual characteristics.

occur. Hence, there would not be any 3-round iterative characteristics that would have a better probability per round than the 2-round iterative characteristic.

In general, the format for a k-round iterative characteristic would involve a trivial round where the input XOR value of that round is zero, followed by $k-1$ non-trivial rounds. Since the trivial round would have a probability of 1 (zero XOR inputs always give zero XOR outputs), the k-round iterative characteristic would thus have a probability of

$$p_{\Omega_k} = \prod_{i=1}^{k-1} p_i. \tag{22}$$

where p_i is the probability of the 1-round characteristic in round $i+1$ of the k-round characteristic.

The probability per round for the 2-round iterative characteristic is 2^{-7}, and so an k-round iterative characteristic would have a better probability per round than the 2-round iterative characteristic only if $p_{\Omega_k} > (2^{-7})^k$. This implies that $p_i = 2^{-7}$ for $i = 1$ to $k-1$. This results because the next best likely one-round difference probability is 2^{-14} and the incorporation of just one such XOR difference would make p_{Ω_k} have the same value as $(2^{-7})^k$. Hence we need to have a 1-round difference probability of 2^{-7} for each non-trivial round. This means that for non-trivial rounds, the input pairs differ in at most 1 s-box and the output pairs differ in at most 1 s-box as well. This is equivalent to event $\mathcal{E}$ and the screening process mentioned earlier would ensure that this event does not happen. Hence, an k-round iterative characteristic that gives a better probability per round than the 2-round iterative characteristic will not occur.

4.3 Differential Characteristics of r Rounds

One can construct an r-round characteristic by concatenating the 2-round iterative characteristic with itself. This r-round characteristic will then have a particular plaintext XOR value ΔP and a probability p_{Ω_r} for a particular sequence of XOR output values to appear from round 1 to round r. If a plaintext pair having XOR value ΔP does indeed produce the same sequence of XOR values in the intermediate rounds as we would expect from the r-round characteristic, then it is a right pair. Otherwise, it is a wrong pair.

By concatenating a 2-round iterative characteristic to produce an r-round characteristic, the probability of a right pair occurring will be

$$p_{\Omega_r} = 2^{-7r} \tag{23}$$

for even r. As a result, for a 6-round characteristic of the CAST-like cipher, the upper bound on the probability of a right pair occurring is 2^{-42}. A CAST-like cipher characteristic of 8 rounds will reduce that probability further to 2^{-56}, a value which is achieved for a 15-round characteristic of DES.

5 Conclusion

In this paper, we have examined the resistance of a CAST-like encryption algorithm to both linear cryptanalysis and differential cryptanalysis.

In our analysis of the resistance of the CAST-like cipher to linear cryptanalysis, we have derived a bound on the probability of satisfaction of a linear approximation based on the minimum nonlinearity of the s-boxes used in the round function of the CAST-like cipher. Our results suggest that for randomly generated 8×32 s-boxes, a 64-bit CAST-like cipher

consisting of 12 rounds has a better degree of resistance to the linear attack than 16 rounds of DES.

In our analysis of the resistance of the CAST-like cipher to differential cryptanalysis, a method for predicting the entries in the XOR table of the round function F in a CAST-like cipher using randomly generated s-boxes has been presented. Based on this method, we have shown that by using randomly generated s-boxes and a simple screening process, the best iterative characteristic available is the 2-round iterative characteristic. For a 64-bit CAST-like algorithm using 8×32 s-boxes, the best 2-round iterative characteristic has a probability of 2^{-14} and this value is almost 70 times smaller than that of the best 2-round iterative characteristic in DES, which has a probability of $\frac{1}{234}$. As a result, an 8 round characteristic of a CAST-like cipher will reduce the probability of the occurrence of a right pair to 2^{-56}, a value which is better than a 15 round characteristic of DES.

6 Acknowledgements

The authors are grateful to the referee whose insightful comments form much of the basis for the analysis of Section 3.2.

References

[1] C. M. Adams. *A Formal and Practical Design Procedure for Substitution-Permutation Network Cryptosystems.* PhD thesis, Queen's University, Kingston, Canada, 1990.

[2] C. M. Adams. Simple and effective key scheduling for symmetric ciphers. In *Workshop on Selected Areas of Cryptography (SAC) '94*, pages 129–133, Queen's University, Kingston, Ontario, Canada, May 1994.

[3] C. M. Adams and S. E. Tavares. Designing s-boxes resistant to differential cryptanalysis. In *Proceedings of 3rd Symposium on the State and Progress of Research in Cryptography*, pages 181–190, Rome, Italy, 1994.

[4] E. Biham and A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard.* New York: Springer-Verlag, 1993.

[5] W. Feller. *An Introduction to Probability Theory and its Applications.* New York: Wiley, 1968.

[6] H. M. Heys and S. E. Tavares. On the security of the CAST encryption algorithm. In *Canadian Conference on Electrical and Computer Engineering*, pages 332–335, Halifax, Nova Scotia, Canada, Sept. 1994.

[7] B. S. Kaliski and M. J. B. Robshaw. Linear Cryptanalysis Using Multiple Approximations. In *Advances in Cryptology: Proceedings of CRYPTO '94*, pages 26–39. Springer-Verlag, Berlin, 1994.

[8] L. R. Knudsen. *Block Ciphers - Analysis, Design and Applications.* PhD thesis, Aarhus University, Denmark, July 1994.

[9] X. Lai, J. L. Massey, and S. Murphy. Markov Ciphers and Differential Cryptanalysis. In *Advances in Cryptology: Proceedings of EUROCRYPT '91*, pages 17–38. Springer-Verlag, Berlin, 1991.

[10] J. Lee. An Investigation of Some Security Aspects of the CAST Encryption Algorithm. Master's thesis, Queen's University, Kingston, Ontario, Canada, 1995.

[11] M. Matsui. Linear Cryptanalysis Method for DES Cipher. In *Advances in Cryptology: Proceedings of EUROCRYPT '93*, pages 386–397. Springer-Verlag, Berlin, 1994.

[12] M. Matsui. The First Experimental Cryptanalysis of the Data Encryption Standard. In *Advances in Cryptology: Proceedings of CRYPTO '94*, pages 1–11. Springer-Verlag, Berlin, 1994.

[13] W. Meier and O. Staffelbach. Nonlinearity Criteria for Cryptographic Functions. In *Advances in Cryptology: Proceedings of EUROCRYPT '89*, pages 549–562. Springer-Verlag, Berlin, 1990.

[14] R. C. Merkle. Fast software encryption functions. In *Advances in Cryptology: Proceedings of CRYPTO '90*, pages 476–501. Springer-Verlag, Berlin, 1991.

[15] K. Nyberg. On the Construction of Highly Nonlinear Permutations. In *Advances in Cryptology: Proceedings of EUROCRYPT '92*, pages 92–98. Springer-Verlag, Berlin, 1992.

[16] National Bureau of Standards. *Data Encryption Standard.* Federal Information Processing Standard Publication 46, 1977.

[17] B. O'Higgins. BNR leads industry in client/server network security. *Telesis*, pages 181–190, Feb. 1994.

[18] C. E. Shannon. Communication Theory of Secrecy System. *Bell System Technical Journal*, volume 28:pages 656–715, 1949.

[19] A. Shimizu and S. Miyaguchi. Fast Data Encryption Algorithm FEAL. In *Advances in Cryptology: Proceedings of EUROCRYPT '87*, pages 267–278. Springer-Verlag, Berlin, 1988.

[20] M. J. Wiener. Efficient DES key search. Technical Report TR-244, School of Computer Science, Carleton University, Ottawa, Canada, May 1994. (Also presented at the Rump Session of CRYPTO '93).

[21] A. Youssef, S.E. Tavares, S. Mister, and C.A. Adams. On the Linear Approximation of Injective S-boxes. presented in the Rump Session of CRYPTO '95, Santa Barbara, California, Aug. 1995.