Private-Key Cipher Design Principles I

Practical S-Box Design

Serge Mister, Carlisle Adams

Nortel, P.O. Box 3511, Station C, Ottawa, Ontario, Canada, K1Y 4H7
cf744@freenet.carleton.ca, cadams@nortel.ca

1. Introduction

Much of the security of a block cipher based on the Feistel network [8, 9] depends on the properties of the substitution boxes (s-boxes) used in the round function. Although many desirable properties have been studied, relatively little work has been done to determine to what degree these properties are achievable in practice. This paper presents one effort to construct large, cryptographically secure s-boxes, contrasting theoretical and practical limitations, and highlighting areas for future research.

2. Background

An $n \times m$ s-box S is a mapping $S: \{0,1\}^n \rightarrow \{0,1\}^m$. S can be represented as 2^n m-bit numbers, denoted $r_0, \ldots, r_{2^n-1}$, in which case $S(x) = r_x$, $0 \le x < 2^n$ and the r_i are the rows of the s-box. Alternatively, $S(x) = [c_{m-1}(x)\ c_{m-2}\ (x) \ldots c_0(x)]$ where the c_i are fixed Boolean functions $c_i: \{0,1\}^n \rightarrow \{0,1\}\ \forall i$; these are the columns of the s-box. Finally, S can be represented by a $2^n \times m$ binary matrix M with the i, j entry being bit j of row i. All three representations will be used in this paper.

The linear combination of two functions $f, g: \{0,1\}^n \rightarrow \{0,1\}$ is defined to be

$$(f \oplus g)(x) = f(x) \oplus g(x)$$

where $\oplus$ denotes modulo 2 addition. Let V_n denote the set of functions mapping $\{0,1\}^n \rightarrow \{0,1\}$. Let L_n denote the set of linear functions mapping $\{0,1\}^n \rightarrow \{0,1\}$. Let A_n denote the set of affine functions mapping $\{0,1\}^n \rightarrow \{0,1\}$.

The following definitions will be useful in the discussion.

2.1 Walsh Transform

The Walsh transform of a function $f: \{0,1\}^n \rightarrow \{0,1\}$ is defined by

$$\mathcal{W}(f)(w) = \sum_{x \in \{0,1\}^n} (-1)^{f(x) + w \cdot x}$$

where $w \cdot x = w_{n-1}x_{n-1} \oplus \cdots \oplus w_0 x_0$ (note that the transform is usually normalized by multiplying by $2^{-n/2}$).

The set of bent functions, denoted B_n with n even, is the set of functions $f:\{0,1\}^n \to \{0,1\}$ such that [14]

$$\mathcal{W}(f)(w) = \pm 2^{n/2} \ \forall w \in \{0,1\}^n .$$

2.2 Inverse Walsh Transform

The inverse Walsh transform of a function $F:\{0,1\}^n \to \mathbf{Z}$ is

$$\mathcal{W}^{-1}(F)(x) = \frac{1}{2^n} \sum_{w \in \{0,1\}^n} F(w)(-1)^{w \cdot x} .$$

2.3 Nonlinearity

The nonlinearity of a function $f:\{0,1\}^n \to \{0,1\}$ is

$$\mathrm{nl}(f) = \min_{l \in A_n} \mathrm{wt}(f \oplus l)$$

where wt() denotes the Hamming weight of the function.

The nonlinearity of an s-box S is

$$\mathrm{nl}(S) = \min_{f \in C} \mathrm{nl}(f)$$

where C is the set of all nontrivial linear combinations of the columns of S.

2.4 XOR Table

Let $\alpha \in \{0,1\}^n \setminus \{\mathbf{0}\}$, $\beta \in \{0,1\}^m$. The XOR table entry of an s-box S corresponding to (α,β) is

$$\mathrm{XOR}(\alpha,\beta) = \#\{x \in \{0,1\}^n : S(x) \oplus S(x \oplus \alpha\} = \beta\}$$

where # denotes the cardinality of the set. The XOR value of an s-box is the highest XOR table entry:

$$\mathrm{XOR}(S) = \max_{\alpha,\beta} \mathrm{XOR}(\alpha,\beta) .$$

2.5 Dynamic Distance

We define the dynamic distance of order j of a function $f:\{0,1\}^n \to \{0,1\}$ as follows:

$$\mathrm{DD}_j(f) = \max_{\substack{d \in \{0,1\}^n \\ 1 \le wt(d) \le j}} \frac{1}{2} \left| 2^{n-1} - \sum_{x=0}^{2^n-1} f(x) \oplus f(x \oplus d) \right| .$$

It provides a measure from which can be defined other dynamic properties such as the strict avalanche criterion, bit independence criterion, and a precise "distance" from these properties.

2.6 Strict Avalanche Criterion (SAC)

A Boolean function $f:\{0,1\}^n \to \{0,1\}$ satisfies the SAC if $\mathrm{DD}_1(f)=0$. The distance to SAC is defined by $\mathrm{DSAC}(f)=\mathrm{DD}_1(f)$. Similarly, $f:\{0,1\}^n \to \{0,1\}$ satisfies higher-order SAC (HOSAC) of order j if $\mathrm{DD}_j(f)=0$ and the distance to higher-order SAC of order j is defined by $\mathrm{DHOSAC}_j(f)=\mathrm{DD}_j(f)$. Maximum order SAC (MOSAC) and the distance to maximum order SAC (DMOSAC) correspond to the case $j=n$. The concepts of SAC, higher-order SAC, and maximum order SAC coincide with those of [4, 15] (see also [13]). An s-box satisfies the (HO, MO)SAC if all of its columns satisfy (HO, MO)SAC.

2.7 Bit Independence Criterion (BIC)

For an s-box S, the distance to higher order BIC is defined by

$$\mathrm{DHOBIC}_{i,j}(S)=\max_{\substack{c\in\{0,1\}^m \\ 1<wt(c)\leq i}} \mathrm{DD}_j(Mc)$$

where M is the binary matrix corresponding to S and the matrix multiplication is done using modulo 2 addition. S satisfies BIC[1] [15] if $\mathrm{DHOBIC}_{2,1}(S)=0$ and satisfies $\mathrm{HOBIC}_{i,j}$ if $\mathrm{DHOBIC}_{i,j}(S)=0$ (note that HOBIC can be defined in terms of varying i, or varying j, or both). Distances to BIC and HOBIC are given by $\mathrm{DHOBIC}_{2,1}(S)$ and $\mathrm{DHOBIC}_{i,j}(S)$ respectively. Maximum order BIC (MOBIC) and the distance to MOBIC (DMOBIC) correspond to HOBIC and DHOBIC with $i=m,\ j=n$.

2.8 Ideal S-Box Properties

The following are properties which we feel that an ideal s-box would possess:

I1. All linear combinations of s-box columns are bent.
I2. All entries in the s-box XOR table are 0 or 2.
I3. The s-box satisfies MOSAC.
I4. The s-box satisfies MOBIC.
I5. The set of weights of rows has a binomial distribution with mean $m/2$.

[1] The (output) Bit Independence Criterion (BIC) states that s-box output bits j and k should change independently when any single input bit i is inverted, for all i, j, and k (note that for a given i, j, and k the independence is computed over the set of all pairs of input vectors which differ only in bit i).

I6. The set of weights of all pairs of rows has a binomial distribution with mean $m/2$.

I7. The columns each have Hamming weight 2^{n-1}.

Property I1 will aid in protection against linear cryptanalysis (see [5]), and I2 against differential cryptanalysis (see [6]). Properties I1, I5, and I7 help to ensure a good static characteristic, and properties I2, I3, I4, and I6 help to ensure a good dynamic characteristic. Not all of these properties can be achieved simultaneously.

3. General S-Box Construction Methods

We observe the following properties of s-boxes:

3.1 Property S1

The nonlinearity distribution of an s-box is not affected when affine functions are added to columns of the s-box.

Proof:

This follows directly from the definition of nonlinearity.

3.2 Property S2

$\max\limits_{\substack{\alpha\in\{0,1\}^n\setminus\{0\}\\ \beta\in\{0,1\}^m}} \mathrm{XOR}(\alpha,\beta)$ for an s-box S is not affected when affine functions are added to the columns of S.

Proof:

Let $b(x)$ be a column of S, $l(x)$ be a linear function, $\delta \in \{0,1\}$, and $\gamma(x) = l(x)+\delta$ be an affine function. Let $h(x) = b(x) \oplus \gamma(x)$ be the s-box column modified by adding an affine function.

For a given input XOR α, the output XOR at the bit position corresponding to column $b(x)$ for the modified s-box is:

$$\begin{aligned} h(x)\oplus h(x\oplus\alpha) &= b(x)\oplus\gamma(x)\oplus b(x\oplus\alpha)\oplus\gamma(x\oplus\alpha) \\ &= b(x)\oplus b(x\oplus\alpha)\oplus\gamma(x)\oplus\gamma(x)\oplus l(\alpha) \quad \text{(since } \gamma(x) \text{ is affine)} \\ &= b(x)\oplus b(x\oplus\alpha)\oplus l(\alpha)\,. \end{aligned}$$

Thus for a given input XOR, the output XORs with the modified column are the output XORs of the original s-box, except that each is XORed with the constant $l(\alpha)$ at the bit position corresponding to $b(x)$. Therefore, the number of output XORs β which correspond to a given input XOR α is unchanged.

3.3 Property S3

Dynamic distance is unaffected by the addition of affine functions.

Proof:

Let $f:\{0,1\}^n \rightarrow \{0,1\}$ be a Boolean function. Let $\gamma(x) = l(x) + \delta$ be an affine function. Let $h(x) = f(x) \oplus \gamma(x)$.

For a fixed input change c, let $B = \frac{1}{2}\left|2^{n-1} - \sum_{x=0}^{2^n-1} f(x) \oplus f(x \oplus c)\right|$ be the dynamic distance of f corresponding to that input change and T be the summation term. The dynamic distance of the modified function is given by

$$\begin{aligned}
\frac{1}{2}\left|2^{n-1} - \sum_{x=0}^{2^n-1} h(x) \oplus h(x \oplus c)\right| &= \frac{1}{2}\left|2^{n-1} - \sum_{x=0}^{2^n-1} f(x) \oplus \gamma(x) \oplus f(x \oplus c) \oplus \gamma(x \oplus c)\right| \\
&= \frac{1}{2}\left|2^{n-1} - \sum_{x=0}^{2^n-1} f(x) \oplus \gamma(x) \oplus f(x \oplus c) \oplus \gamma(x) \oplus l(c)\right| \\
&= \frac{1}{2}\left|2^{n-1} - \sum_{x=0}^{2^n-1} f(x) \oplus f(x \oplus c) \oplus l(c)\right| \\
&= \frac{1}{2}\left|2^{n-1} - T\right| \\
&= B
\end{aligned}$$

where the second to last equality follows because $l(c)$ is a constant with respect to the summation. If $l(c) = 0$, the result is immediate. If $l(c) = 1$,

$$\begin{aligned}
\left|2^{n-1} - \sum_{x=0}^{2^n-1} f(x) \oplus f(x \oplus c) \oplus l(c)\right| &= \left|2^{n-1} - \sum_{x=0}^{2^n-1} f(x) \oplus f(x \oplus c) \oplus 1\right| \\
&= \left|2^{n-1} - \left(2^n - \sum_{x=0}^{2^n-1} f(x) \oplus f(x \oplus c)\right)\right| \\
&= \left|-2^{n-1} + \sum_{x=0}^{2^n-1} f(x) \oplus f(x \oplus c)\right| \\
&= \left|2^{n-1} - T\right|.
\end{aligned}$$

3.4 Property S4

The distances to (HO, MO)SAC and to (HO, MO)BIC for an s-box are not affected when affine functions are added to its columns.

Proof:

This follows directly from Property S3.

Properties S1-S4 show that an s-box with high nonlinearity, low distance to BIC, and a good XOR table can be modified by adding affine functions to the columns without disturbing these properties.

We will now focus on the case $n = 8$, $m = 32$, the dimensions typically used in CAST algorithms [1, 2]. Property I1 cannot be achieved for $2m > n$ [12]. I1 is then replaced with I1': the s-box has the highest possible nonlinearity. Property I2 can be achieved without much difficulty because m is sufficiently greater than n; ideal XOR tables for the s-box dimensions given have $2^{n-1} \cdot (2^n - 1) = 32640$ entries containing 2 and all other entries 0. Property I3 is guaranteed if bent functions are chosen for the c_i. Property I4 cannot be achieved for $2m > n$ [12]. We have reduced I4 to I4': the s-box minimizes $\text{DHOBIC}_{32,1}$.

This leads us to construct s-boxes using bent functions as columns (Property I3), and emphasizing only nonlinearity and $\text{HOBIC}_{32,1}$ (properties I1' and I4'). Once complete, the s-box can be inspected to see that it satisfies property I2, and can be modified by adding affine functions to columns to approximately obtain properties I5 and I6. Note that with $n = 8$, the bent functions have Hamming weight 120 or 136, which is close to the weight 128 required by property I7. These constructed s-boxes, therefore, represent what appears to be a good approximation to our definition of ideal s-boxes of these dimensions.

3.5 S-Box Construction Algorithm

The following construction algorithm (Alg. 1) was used:

1. Set $ncols = 0$.
2. Load a bent function into column $ncols$ of the s-box.
3. Test the nonlinearity and DD_1 of all combinations of columns $0\text{-}ncols$ that involve column $ncols$.

4. If the minimum nonlinearity observed is greater than or equal to the minimum desired nonlinearity and the highest dynamic distance observed is at most the maximum allowable $\text{DHOBIC}_{32,1}$, increment *ncols*.

5. If $ncols < 32$ go back to step 2.

The following algorithm (the lazy counting algorithm) can be used to generate combinations of columns. It generates all 2^{ncols} combinations of $ncols+1$ columns involving c_{ncols} as needed in step 3 of the algorithm above. In addition, only one XOR operation is required to generate the next combination whose nonlinearity and $\text{DHOBIC}_{32,1}$ are to be tested.

```
1. Set $f = c_{ncols}$.
2. Test the nonlinearity of $f$.
3. For $i = 1$ to $2^{ncols} - 1$
   a) Find the least significant bit of the binary representation of $i$ which is 1. Let $b$ be the associated bit position number.
   b) Set $f = f \oplus c_b$. ($f_{new} = f_{previous} \oplus c_b$)
   c) Test the nonlinearity and $\text{DHOBIC}_{32,1}$ of $f$.
```

4. Constructions of Bent Functions

The method of s-box generation proposed in the previous section requires a set of bent functions. Several construction techniques are known [7] (see also [3, 10, 13]). The ones which we used are described below.

4.1 Generation from 6 input bent functions:

Given a set of bent functions in B_6, bent functions in B_8 can be constructed using either of the following two methods:

4.1.1 Method 1:

Let $a,b \in B_6$. Then the function $f:\{0,1\}^8 \rightarrow \{0,1\}$ defined by

$$f(x_7 \ldots x_0) = \begin{cases} a(x_5 \ldots x_0), & x_6 = 0, x_7 = 0 \\ a(x_5 \ldots x_0), & x_6 = 0, x_7 = 1 \\ b(x_5 \ldots x_0), & x_6 = 1, x_7 = 0 \\ b(x_5 \ldots x_0) \oplus \mathbf{1}, & x_6 = 1, x_7 = 1 \end{cases}$$

is bent [3]. Rearrangements of the 64 bit blocks in the expression above also result in bent functions.

4.1.2 *Method 2:*

Let $a, b, c \in B_6$ and let A, B, C be their respective Walsh Transforms. If

$$D(w_7 \ldots w_0) = \begin{cases} A(w_5 \ldots w_0), & w_6 = 0, w_7 = 0 \\ B(w_5 \ldots w_0), & w_6 = 0, w_7 = 1 \\ C(w_5 \ldots w_0), & w_6 = 1, w_7 = 0 \\ -2^{2n}\left[A(w_5 \ldots w_0)B(w_5 \ldots w_0)C(w_5 \ldots w_0)\right]^{-1}, & w_6 = 1, w_7 = 1 \end{cases}$$

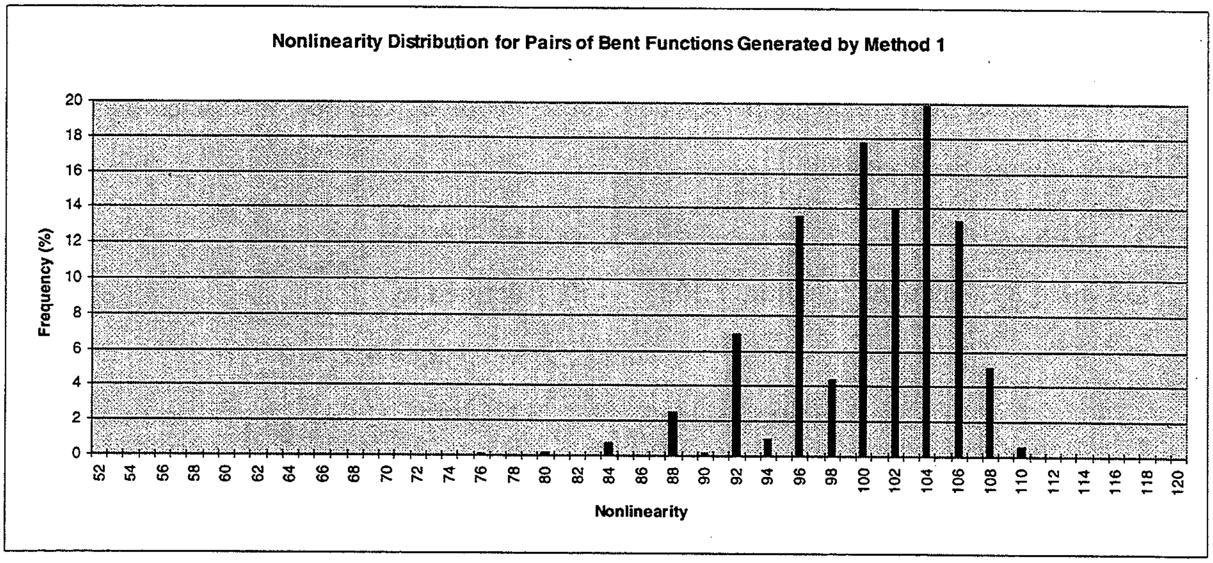
represents the Walsh transform of a function $d: \{0,1\}^n \to \{0,1\}$ then d is bent [13].

4.2 Maiorana Functions

Let $\pi(x)$ be a bijective mapping from $\{0,1\}^{n/2} \to \{0,1\}^{n/2}$, $g(x)$ be a function in $V_{n/2}$, and x_H and x_L denote the high $n/2$ and low $n/2$ bits of x respectively. Let the $\cdot$ operator denote the dot product $\left[a_{n/2-1}\ a_{n/2-2} \cdots a_0\right] \cdot \left[b_{n/2-1}\ b_{n/2-2} \cdots b_0\right] = a_{n/2-1}b_{n/2-1} \oplus a_{n/2-2}b_{n/2-2} \oplus \cdots \oplus a_0 b_0$. Then the function $f \in V_n : f(x) = f(x_H, x_L) = \pi(x_H) \cdot x_L \oplus g(x_H)$ with n even is bent, and is called a Maiorana function [7, 10].

4.3 Properties of Bent Functions Constructed by Method 1

A set of 100000 functions, $f_0 \ldots f_{99999}$, was generated using Method 1. For random, distinct, $i, j \in \{0, 1, \ldots, 99999\}$, the nonlinearity of $f_i \oplus f_j$ was calculated. The following chart shows the resulting nonlinearity distribution.



Two similar distributions are present. The lower one corresponds to nonlinearities divisible by 2 but not by 4, and the upper to nonlinearities divisible by four. This set of functions has the greatest nonlinearity spread of all those considered in this paper. In our experiments, it was found that it is possible to build 8×25 s-boxes with nonlinearity 80 and 8×29 s-boxes with nonlinearity 76 with columns in this set. The minimum nonlinearity of a pair of functions was 52.

The set of functions was filtered by taking only groups of 4 functions forming 8×4 s-boxes with nonlinearity at least 100. It was hoped that if one function from a group of four had high nonlinearity when added to each member of a set of functions, the other functions in that group would also have high nonlinearity with respect to that set. Experimentally, it was found that such filtering did not support this hypothesis.

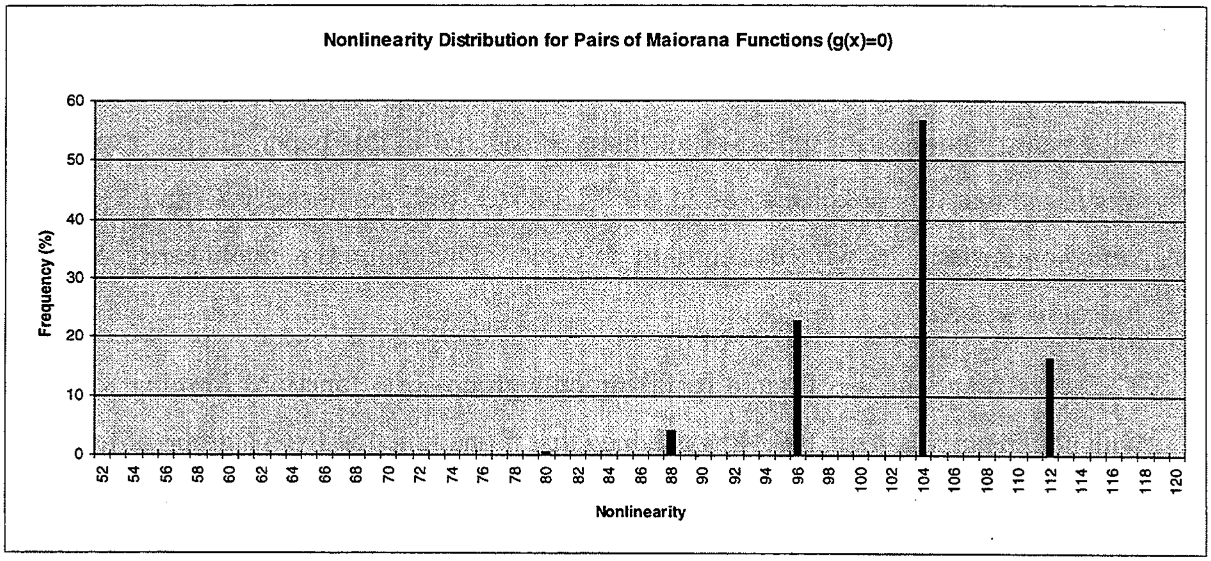
S-Boxes having more than three columns and built with functions constructed by Method 1 have a DBIC value of 64, the worst (highest) possible.. This can be observed as follows. Let a_1, b_1 and a_2, b_2 be the truth tables corresponding to two pairs of bent functions in B_6. Let $(a_1, a_1, b_1 \oplus \alpha, b_1 \oplus \beta)$ and $(a_2, a_2, b_2 \oplus \chi, b_2 \oplus \delta)$ be elements of B_8 constructed by Method 1, where exactly one of $\alpha, \beta \in \{\mathbf{0},\mathbf{1}\}$ is $\mathbf{1}$ and exactly one of $\chi, \delta \in \{\mathbf{0},\mathbf{1}\}$ is $\mathbf{1}$. The linear combination of these two functions is $(a_1 \oplus a_2\, , a_1 \oplus a_2\, , b_1 \oplus b_2 \oplus \alpha \oplus \chi\, , b_1 \oplus b_2 \oplus \beta \oplus \delta)$ which achieves a DBIC of 64 for an input change of 01000000 (binary) because the resulting change vector is:

$$\begin{aligned}(a_1 \oplus a_2 \oplus a_1 \oplus a_2\ , b_1 \oplus b_2 \oplus \alpha \oplus \chi \oplus b_1 \oplus b_2 \oplus \beta \oplus \delta) &= (\mathbf{0}, \alpha \oplus \chi \oplus \beta \oplus \delta) \\ &= (\mathbf{0},\mathbf{0}).\end{aligned}$$

This argument is valid for any rearrangement of the 64 bit blocks provided that both are rearranged in the same way. Thus, an s-box with columns constructed using Method 1 will have a DBIC of 64 if any two columns have the same block arrangement (ignoring complementation). Because there are only three distinct rearrangements, the result follows.

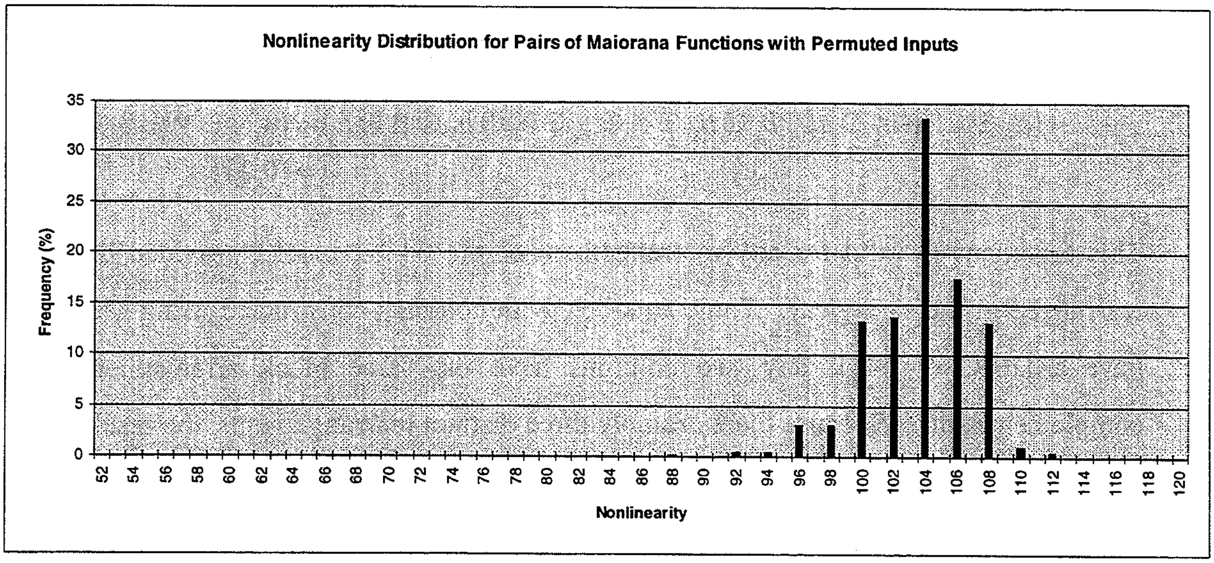
4.4 Properties of Maiorana Functions

For randomly generated pairs of Maiorana functions $m(x)$, $p(x)$ with $g(x) = \mathbf{0}$, the nonlinearity of $m \oplus p$ was calculated. The resulting nonlinearity distribution follows:



The fact that the nonlinearity of linear combinations of Maiorana functions is divisible by $2^{n/2-1}$ is explained in [11]. In our experiments, we found that Maiorana functions cannot be used to generate s-boxes larger than 8×17 of nonlinearity greater than 80 with Alg. 1. The lowest nonlinearity observed for a pair of Maiorana functions was 64.

Let $a,b,\ldots,h$ be a permutation of $0,1,\ldots,7$. Then it is easy to show that the function $f(x_7 \ldots x_0) = f_m(x_a \ldots x_h)$, where $f_m()$ is a Maiorana function, is also bent. We will refer to these functions as Maiorana functions with permuted inputs. The nonlinearity distribution of pairs of these functions follows:



With this set of functions, our experiments generated s-boxes with nonlinearity 76 of size 8×30. This is the largest s-box of nonlinearity 76 or greater that was generated from any single set of functions described in this paper. With a nonlinearity of 80, the dimension 8×25 could not be exceeded using our pool of functions. The lowest nonlinearity observed for a pair of Maiorana functions with permuted inputs was 82, the highest of all sets considered here.

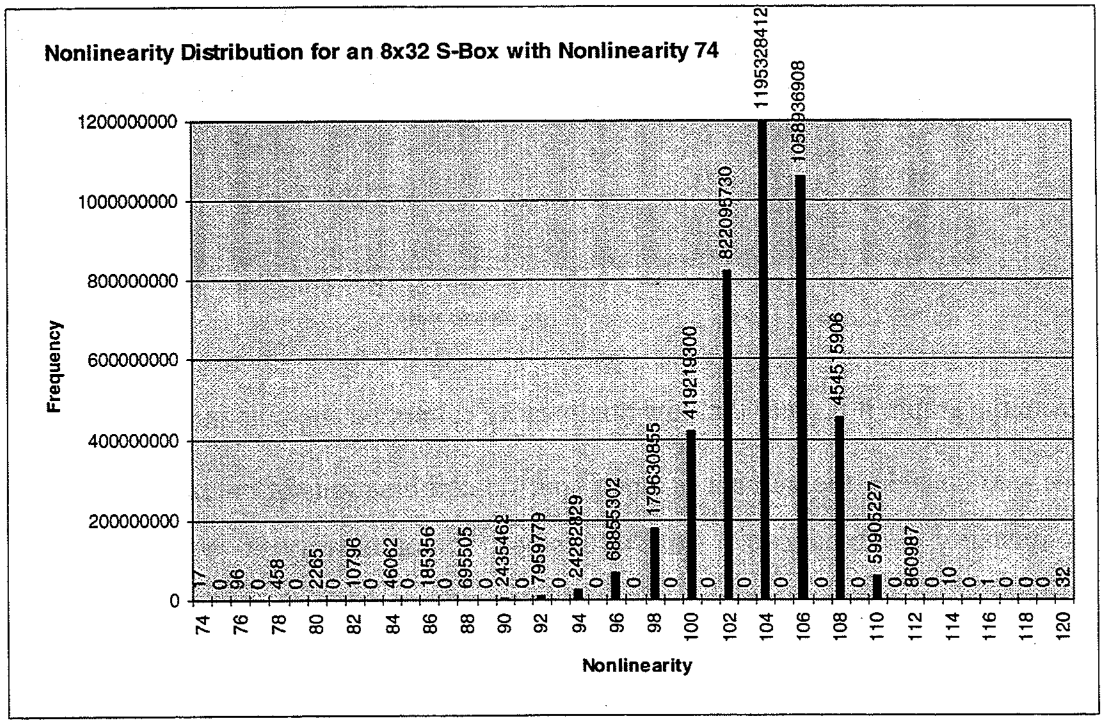
5. Bent Functions in S-Box Design

Experimentally, it was found that 8×32 s-boxes with nonlinearity 74 were best constructed by the following method:

1. build an 8×29 s-box with nonlinearity 76 from bent functions generated by Method 1
2. Append two Maiorana functions with permuted inputs, keeping the s-box nonlinearity at 76
3. Append a Maiorana function with permuted inputs, reducing the s-box nonlinearity to 74

Using this technique, construction of a single s-box takes 15 to 30 days on a Pentium 90. However, the computation can readily be distributed across many computers, reducing the construction time to a few hours.

The following chart shows a typical distribution of nonlinearities of all combinations of the columns of an s-box with nonlinearity 74 generated by the method described above:

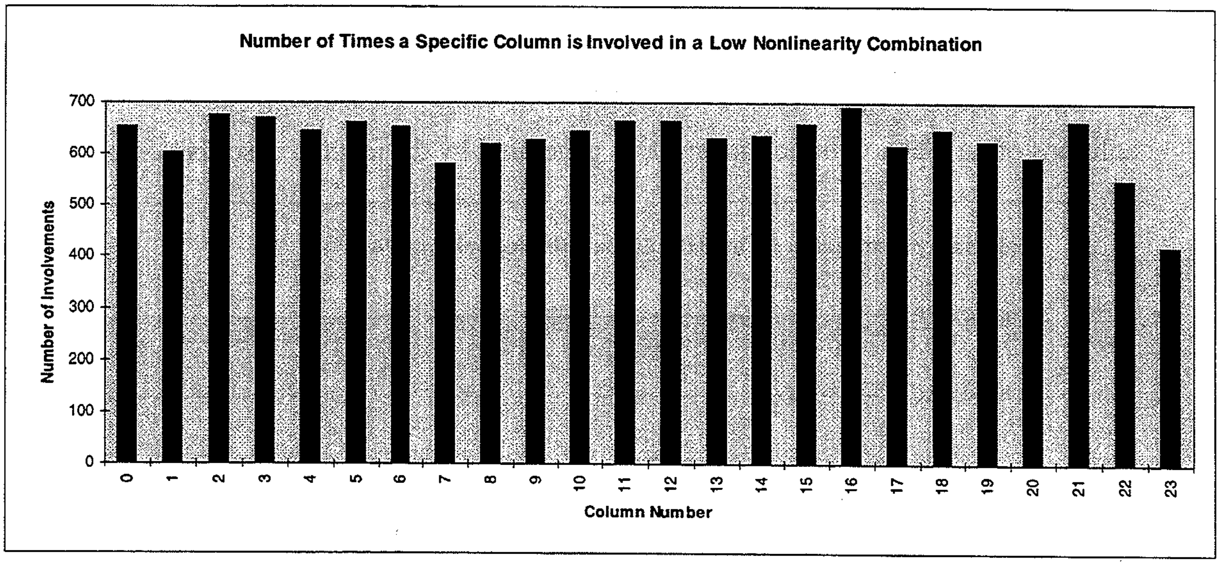


We observe that the nonlinearity of most linear combinations is much higher than 74. Only 17 linear combinations have nonlinearity 74, and 571 have nonlinearity below 80 (out of 2^{32} possible combinations). However, no 8×32 s-boxes with bent functions as columns with nonlinearity 76 or above have yet been found using this construction method.

6. Improvements on the Construction Algorithm

The algorithm described so far simply discards columns which do not have the desired minimum nonlinearity with respect to any combination of columns already in the s-box. Although this method works, it is disturbing that until the technique has found a new column, it is no more likely to find a suitable column than it was immediately after locating the previous one. The following development leads to an improvement over this situation which could possibly be exploited further.

Below is a plot of the number of times a particular s-box column is included in a linear combination having nonlinearity less than 80 during the construction process.



From the graph, it is clear that low nonlinearity cannot be attributed to a few s-box columns.

We next consider the number of times specific combinations of columns are involved in low nonlinearity combinations. An 8×10 s-box of nonlinearity 80 was constructed, and the number of times a combination of these 10 functions when combined with an eleventh bent function resulted in a nonlinearity under 80 was recorded. This was repeated three times. A summary of the results follows:

	Exp. 1	Exp. 2	Exp. 3
Number of functions tested	200 393	18 259	5 908
Combination producing low nonlinearity most often (lazy counting algorithm counter value given)	127	15	31
Maximum number of low nonlinearities caused by a single combination	63 633	2 533	1 058
Percentage of low nonlinearities caused by a single combination	31.8	13.9	17.9
Total number of combinations having generated a low nonlinearity (out of 1024)	115	328	208
Number of combinations which generate low nonlinearities in all experiments	36		
Number of combinations which generate low nonlinearities in last two experiments	162		

We observe that up to 30% of columns could be rejected by performing a nonlinearity test with a fixed combination of the existing s-box columns. Because the combination to be checked differed for all three experiments, the construction algorithm would need to determine the combination which most frequently is involved in low nonlinearities. This is difficult because the full nonlinearity test must be completed even if it is already known part way through the test that the nonlinearity of the s-box with the function being considered is lower than the minimum required.

6.1 Effective Use of Combinations and Subspaces

We now consider an s-box nonlinearity test which checks first that a candidate function has the desired nonlinearity with respect to

1. all combinations of columns not involving the most recently added one
2. all combinations of columns involving the most recently added one.

Suppose that two candidate functions f and g complete part 1 successfully and fail part 2. Then if the function $f \oplus g$ passes part 1, the most recently added column can be replaced with f and g can be added to the s-box. Verifying that $f \oplus g$ passes part 1 involves only half the effort required to verify that a new candidate function passes both parts 1 and 2. This scheme has been implemented and causes a noticeable decrease in the time required to find columns for large s-boxes. The method could be extended, for example, to save time when three functions have the desired nonlinearity for all combinations not involving two specific columns of an s-box.

6.2 Calculating the Nonlinearity of Functions

The speed of s-box construction is greatly dependent on the time required for a nonlinearity calculation. The nonlinearity of a function f can be computed using:

$$nl(f) = 2^{n-1} - \frac{1}{2} \max_{w \in \{0,1\}^n} \left| \mathcal{W}(f)(w) \right|.$$

The Walsh transform of f can be calculated by multiplying the truth table of $(-1)^{f(x)}$ by a Hadamard Matrix of order n. This matrix is defined recursively by:

$$H_0 = 1$$
$$H_n = \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix}.$$

Let $f[a...b]$ represent the truth table of $(-1)^{f(x)}$ for inputs between a and b inclusive. The equation $H_n f[0...2^n - 1] = \begin{bmatrix} A+B \\ A-B \end{bmatrix}$, where $A = H_{n-1} f[0...2^{n-1} - 1]$ and $B = H_{n-1} f[2^{n-1}...2^n - 1]$, can be used to efficiently calculate the Walsh transform. In our current implementation, $H_4 f$ is calculated by table lookup. In the final stage of the recursion, $|A| + |B|$ is computed instead of $A + B$ and $A - B$ in view of the maximization specified in the formula for nonlinearity.

The improvements described in 6.1 and 6.2 combine to give a nonlinearity calculation requiring approximately $100 \mu s$ on a Pentium 90 (roughly 30-50 times better than a naive implementation of the nonlinearity calculation).

7. Conclusions

The following table summarizes the properties obtained for 8×32 s-boxes constructed according to the methods described in this paper and for random 8×32 s-boxes:

Property	Random S-Box	Constructed S-Box
I1'. Nonlinearity	72*	74
I2. Largest XOR table entry	2	2
I3. DMOSAC	17	0
I4'. DHOBIC$_{32,1}$	37	36
I5. Row weight distribution	Approximately binomial	Approximately binomial
I6. Row pair distribution	Approximately binomial	Approximately binomial
I7. Average column weight	128	128**

* See [16]

** Half the columns have weight 120 and the other half have weight 136

The constructed s-boxes are equivalent to random s-boxes with respect to properties I2, I5, and I6, and are superior to random s-boxes with respect to properties I1', I3, and I4'. It is therefore conjectured that using constructed s-boxes in CAST-like ciphers will increase security (compared with the use of random s-boxes). Note that although improvements in nonlinearity and DHOBIC$_{32,1}$ seem minor (74 vs. 72 and 36 vs. 37), it has so far proven impossible to construct 8×32 s-boxes with nonlinearity greater than 74 or DHOBIC$_{32,1}$ less than 36, and has, on the other hand, been almost trivial to construct 8×32 s-boxes with nonlinearity less than 74 or DHOBIC$_{32,1}$ greater than 36. Thus, these "minor" improvements may in fact represent significant advances in terms of s-box strength (especially considering that property I3 is so much improved and properties I2, I5, and I6 are not degraded). Also, the construction time for an s-box is only about twice the time required to find the nonlinearity and DBIC for a given s-box. Construction would then typically be advantageous over random generation if the probability of a randomly generated s-box having unsatisfactory nonlinearity or DBIC is not negligible. In most cases, the increase in security would justify the extra time required for construction.

The creation of large, cryptographically good s-boxes with bent functions as columns has proven more difficult than originally expected, although extensive experimentation (as summarized in this paper) has given rise to a method which appears to produce satisfactory results. The current algorithm takes between 15 and 30 days on a Pentium 90 for an 8×32 s-box with the properties listed in the table above, and each additional column would cause the generation time to double. However, the process can readily be distributed over a number of computers to significantly reduce the time required.

Finally, the concept of dynamic distance presented here defines a quantitative measure of how close an s-box is to satisfying dynamic properties such as SAC, BIC, and their higher orders, and provides an intuitive, unified framework for these properties.

References

[1] C. M. Adams, *Constructing Symmetric Ciphers Using the CAST Design Procedure*, (submitted for publication).

[2] C. M. Adams, *Designing DES-Like Ciphers with Guaranteed Resistance to Differential and Linear Attacks*, Workshop Record of the Workshop on Selected Areas in Cryptography (SAC 95), May 18-19, 1995, pp. 133-144.

[3] C. M. Adams and S. E. Tavares, *Generating and Counting Binary Bent Sequences*, IEEE Transactions on Information Theory, vol. IT-36, 1990, pp. 1170-1173.

[4] C. M. Adams and S. E. Tavares, *The Use of Bent Sequences to Achieve Higher-Order Strict Avalanche Criterion in S-Box Design*, Technical Report TR 90-013, Dept. of Electrical Engineering, Queen's University, Kingston, Ontario, Jan., 1990.

[5] E. Biham, *On Matsui's Linear Cryptanalysis*, Advances in Cryptology - Proceedings of EUROCRYPT '94, Springer-Verlag, Berlin, 1995, pp. 341-355.

[6] E. Biham and A. Shamir, *Differential Cryptanalysis of DES-like Cryptosystems*, Advances in Cryptology: Proceedings of CRYPTO '90, Springer-Verlag, Berlin, 1991, pp. 1-21.

[7] J. F. Dillon, *A Survey of Bent Functions*, NSA Technical Journal, Special Issue, 1972, pp. 191-215.

[8] H. Feistel, *Cryptography and Computer Privacy*, Scientific American, 228 (1973), pp. 15-23.

[9] H. Feistel, W. Notz, and J. L. Smith, *Some Cryptographic Techniques for Machine-to-Machine Data Communications*, Proceedings of the IEEE, 63 (1975), pp. 1545-1554.

[10] J. A. Maiorana, *A Class of Bent Functions*, R41 Technical Paper, June 1971. (see [7])

[11] S. Mister, *Notes on Maiorana Functions and S-Box Design*, (in progress).

[12] K. Nyberg, *Perfect Nonlinear S-boxes*. Advances in Cryptology - Proceedings of EUROCRYPT '91, Springer-Verlag, Berlin, 1991, pp. 378-385.

[13] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, and J. Vandewalle, *Propagation characteristics of boolean functions*, Advances in Cryptology: Proceedings of EUROCRYPT '90, Springer-Verlag, Berlin, 1991, pp. 161-173.

[14] O. S. Rothaus, *On 'Bent' Functions*, Journal of Combinatorial Theory, 20(A), 1976, pp. 300-305.

[15] A. F. Webster and S. E. Tavares, *On the Design of S-Boxes*, Advances in Cryptology: Proceedings of CRYPTO '85, Springer-Verlag, New York, 1986, pp. 523-534.

[16] A. Youssef, S. Tavares, S. Mister, C. Adams, *Linear Approximation of Injective S-boxes*, Electronics Letters, Vol. 31 No. 25, Dec. 7, 1995, pp. 2165-2166.

Modelling Avalanche in DES-Like Ciphers

Howard M. Heys

Electrical Engineering
Memorial University of Newfoundland
St.John's, Newfoundland, Canada A1B 3X5
Email: howard@engr.mun.ca

Abstract: In this paper, we examine the avalanche characteristics of private-key block ciphers constructed using a DES-like architecture. Avalanche is a desirable cryptographic property that is necessary to ensure that a small difference between two plaintexts results in a seemingly random difference between the two corresponding ciphertexts. In order to examine the behaviour of DES-like ciphers in relation to the avalanche property, a model of the cipher is developed which allows us to analyze the avalanche characteristics of the cipher for different cipher parameter values. In particular, the results suggest that large, symmetric S-boxes which satisfy the guaranteed avalanche property are effective in combining efficiency and good avalanche characteristics of the cipher.

I. Introduction

Private-key block ciphers are typically implemented as a product cipher, using a number of rounds of substitutions and linear transformations. One such class of ciphers, introduced in [1] and referred to as DES-like or Feistel ciphers, uses the general structure of the Data Encryption Standard (DES) [2].

The concept of avalanche in block ciphers was informally introduced by Feistel [3] and Feistel, Notz, and Smith [1], as the property of a small number of bit changes in the plaintext input leading to an "avalanche" of changes in subsequent rounds resulting in a large number of ciphertext bit changes. More precisely, in our analysis, we consider the following definition of the avalanche criterion [4]:

Definition 1 : A cipher is said to satisfy the *avalanche criterion* if, for all keys, on average, half of the ciphertext bits change when one plaintext bit is changed.

Note that this definition is very similar to (but a little looser than) the strict avalanche criterion [5] which states that each ciphertext bit must change with a probability of exactly one half given a particular one bit plaintext change. As a measure of a cipher's adherence to the avalanche criterion we define avalanche probability.

Definition 2 : The ***avalanche probability***, P_{av}, of a cipher is the average fraction of ciphertext bits that change when one plaintext bit is changed and the key remains fixed.

For a cipher which perfectly satisfies the avalanche criterion, $P_{av} = 1/2$. The avalanche probability can be used as one measure of the performance of a cipher: the fewer rounds it takes for the avalanche probability to converge to $1/2$, the stronger the cipher (with respect to avalanche), implying a cipher of more efficient construction consisting of fewer rounds.

In [4], the avalanche characteristics of basic substitution-permutation networks (SPNs) (which are not DES-like) are modelled and the effect of varying cipher parameters are examined. In this paper, we extend this work and develop a model of the avalanche characteristics of DES-like ciphers. The value of this model is that it allows us to examine the relationship between avalanche and various parameters of a DES-like cipher such as the amount of expansion and the S-box dimensions and properties. As well, the performance of DES-like ciphers and the basic SPN ciphers of [4] are compared.

II. Modelling the Cipher

As shown in Figure 1, an R-round DES-like cipher encrypts by dividing the N-bit plaintext input block into two halves: left half $\mathbf{L}_1$ and right half $\mathbf{R}_1$. [1] The right half block $\mathbf{R}_1$ is transformed by the keyed round function f and XORed bit-by-bit to the left half block $\mathbf{L}_1$ to form a new left half block. The right and left halves are then swapped. Consequently, for a round i, $1 \leq i \leq R$ of the cipher, letting $\mathbf{L}_i$ and $\mathbf{R}_i$ represent the left and right half-blocks, respectively, and $\mathbf{K}_i$ represent the key bits applied to the round function, the DES-like algorithm may be viewed as the following iterated operation:

$$\begin{aligned} \mathbf{L}_{i+1} &= \mathbf{R}_i \\ \mathbf{R}_{i+1} &= \mathbf{L}_i \oplus f(\mathbf{R}_i, \mathbf{K}_i). \end{aligned} \tag{1}$$

After the last round, since the half-blocks are not swapped, we have $\mathbf{R}_{R+1}$ and $\mathbf{L}_{R+1}$ represent-

[1] Note that the initial and final permutations of DES have been ignored.

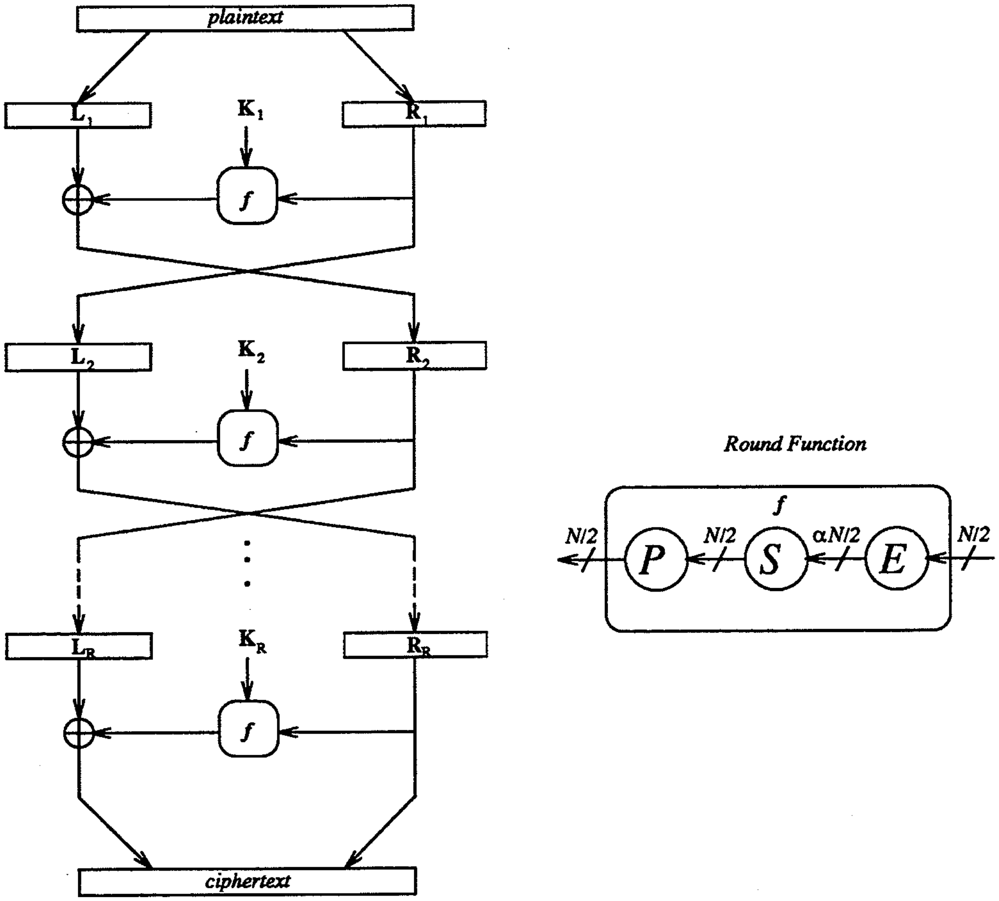


Figure 1: DES-like Cipher Structure

ing the left and right halves of the ciphertext, respectively.

As illustrated in Figure 1, there are generally three components in the round function f: the expansion (E), the substitution (S), and the permutation (P). The cipher is keyed by applying a subset of cipher key bits to the round function, typically by XORing with the data bits before the substitution is performed.

(a) S-box Model

The substitution component operates by dividing the block into a number of smaller sub-blocks and then replacing the bits of these sub-blocks according to a predefined mapping referred to as an S-box. In this paper we consider S-boxes of dimension $m \times n$, $m \geq n$, where m represents the number of input bits and n represents the number of output bits. DES has eight 6×4 S-boxes which are used in the substitution component of the round function.

In general, we represent the input to an S-box as $\mathbf{X} = [X_1 X_2 ... X_m]$, $X_i \in \{0, 1\}$, and the output as $\mathbf{Y} = [Y_1 Y_2 ... Y_n]$, $Y_i \in \{0,1\}$. The input and output differences or change vectors of an S-box corresponding to the bit-wise XOR of two different values for $\mathbf{X}$ and the bit-wise XOR of the resulting two values for $\mathbf{Y}$ are represented by $\Delta\mathbf{X}$ and $\Delta\mathbf{Y}$, respectively.

We model the S-box in the cipher by treating the number of output changes of the S-box as a random variable. Representing the Hamming weight operation by $wt(\cdot)$ and letting $D = wt(\Delta\mathbf{Y})$ represent the random variable corresponding to the number of output bit changes, the model uses the probability distribution of D given by

$$P_D(D=0) = \begin{cases} 1 & , wt(\Delta\mathbf{X}) = 0 \\ \frac{2^{m-n}-1}{2^m-1} & , wt(\Delta\mathbf{X}) \geq 1 \end{cases} \tag{2}$$

and

$$P_D(D=d) = \begin{cases} 0 & , wt(\Delta\mathbf{X}) = 0 \\ \frac{\binom{n}{d} \cdot 2^{m-n}}{2^m-1} & , wt(\Delta\mathbf{X}) \geq 1 \end{cases} \tag{3}$$

for $1 \leq d \leq n$.

To understand the origin of (2), consider that there are 2^{m-n} times more input change vector values for $\Delta\mathbf{X}$ than output change vector values for $\Delta\mathbf{Y}$. Hence, we expect a particular value of the output change vector to occur 2^{m-n} times more often than an input change vector value. Clearly, if there are no input bit changes, then there are no output bit changes resulting the probability of 1 in the first case of (2). The remaining $2^{m-n} - 1$ occurrences of the all-zeros output change vector can be expected to occur when there are input bit changes. Since there are $2^m - 1$ non-zero input changes, the probability of a zero output change given an input change is given by the second case of (2).

Consider now the derivation of the probability distribution of D for $D > 0$ as given in (3). The first case arises from the fact that if there are no input changes, then there are no output changes and $P_D(d)$ must be zero. If there is an input change, as in the second case, then the total number of possible output changes corresponding to a weight of d is given by the number of selections of the d changes from the n output bits multiplied by the factor 2^{m-n} to account for the ratio of possible inputs to outputs. This is divided by the total number of non-zero input changes given by $2^m - 1$.

Note that this stochastic model of the S-box is not intended to characterize the behaviour of an actual, physically realizable S-box, but rather represents an aggregate behaviour over all randomly selected S-boxes. In this sense, it represents a typical S-box.

(b) Permutation Model

The permutation component transposes the bits within the block. In DES, the 32-bit permutation has the property that no two outputs of an S-box are connected to the input of the same S-box.

To model the permutation component of the round function we represent the permutation by a random variable where all possible permutations are considered equally likely. Hence, this model does not make any assumptions about the permutation properties. Instead, we model the cipher by averaging over all possible values that the permutation component can take on. Therefore, it is quite reasonable to expect that a well chosen permutation might display better characteristics than the averaging model used for the analysis. However, in general, the analysis of specific permutations is very difficult and, since the permutation depends greatly on the block and S-box sizes, it is not clear how to generalize an optimal permutation for the purposes of our model of DES-like ciphers.

(c) Expansion Model

The expansion component duplicates an appropriate number of input bits before they are presented to the substitution component and is required if asymetric $m \times n$ S-boxes with $m > n$ are used. The expansion factor, α, of the round function is given by the ratio of the number of bits entering the substitution component to the block size at the input of the round function. In DES, the round function input is 32 bits and the substitution takes 48 bits as its input. Hence DES has an expansion factor of $\alpha = 1.5$.

In our model of the avalanche characteristics of DES-like ciphers, we treat the expansion as a random variable and average over all possible values of the random variable. It is assumed that α is fixed and the expansion randomly selects the appropriate bits for duplication from the set of all bits entering the round function. For example, for the parameter values of DES, 16 out of 32 bits are arbitrarily selected as the set of duplicated bits. The model updates the avalanche characteristics based on averaging over all possible selections for those 16 bits at each round of the cipher.

III. Computation of Avalanche

In this section, we detail the computational model that is used to examine the avalanche characteristics of a cipher of R rounds. Since the avalanche probability is calculated iteratively from 1 to R rounds, ciphers may be analyzed in relation to their satisfaction of the avalanche criterion as a function of the number of rounds. Note that in the following development the key is assumed fixed and, hence, is not a factor in the computation of avalanche probability.

Consider the determination of the distribution of the number of bit changes at the input to a round given the distribution of the number of bit changes at the input to the previous round. Let η_L and η_R represent the number of bit changes in the left and right half blocks, respectively, at the input to a round i, i.e., $\eta_L = wt(\Delta\mathbf{L}_i)$ and $\eta_R = wt(\Delta\mathbf{R}_i)$. Using the total probability theorem, the probability of η_L^* and η_R^* bit changes in the left and right half inputs to round $i+1$ is given by

$$P(\eta_L^*, \eta_R^*) = \sum_{\eta_L=0}^{N/2} \sum_{\eta_R=0}^{N/2} P(\eta_L^*, \eta_R^* | \eta_L, \eta_R) \cdot P(\eta_L, \eta_R). \tag{4}$$

Since, in a DES-like structure, $\eta_L^* = \eta_R$, this can be simplified to

$$P(\eta_L^*, \eta_R^*) = \sum_{\eta_L=0}^{N/2} P(\eta_R^* | \eta_L, \eta_R = \eta_L^*) \cdot P(\eta_L, \eta_R = \eta_L^*). \tag{5}$$

Let η_f represent the number of bit changes at the output of the round function. Then

$$P(\eta_R^* | \eta_L, \eta_R) = \sum_{\eta_f=0}^{N/2} P(\eta_R^* | \eta_L, \eta_R, \eta_f) \cdot P(\eta_f | \eta_L, \eta_R). \tag{6}$$

Since η_R^* is determined directly by η_L and η_f, and η_f is not affected by η_L, we have

$$P(\eta_R^* | \eta_L, \eta_R) = \sum_{\eta_f=0}^{N/2} P(\eta_R^* | \eta_L, \eta_f) \cdot P(\eta_f | \eta_R). \tag{7}$$

Let $\mu = max(\eta_L, \eta_f)$ and $\lambda = min(\eta_L, \eta_f)$. Now $P(\eta_R^* | \eta_L, \eta_f)$ can be determined from

$$P(\eta_R^* | \eta_L, \eta_f) = \begin{cases} \dfrac{\binom{\mu}{i} \cdot \binom{N/2-\mu}{\lambda-i}}{\binom{N/2}{\lambda}} & , \eta_R^* = \eta_L + \eta_f - 2i \\ 0 & , otherwise \end{cases} \tag{8}$$

for all i, $0 \leq i \leq \lambda$. To understand the origin of (8), consider the general bit-wise XOR of two random b-bit vectors: $w = u \oplus v$ where $\eta_w = wt(w)$, $\eta_u = wt(u)$, and $\eta_v = wt(v)$. Without loss of generality, assume that $\eta_u > \eta_v$ and that the first η_u bits of u are ones and the remaining bits are zeroes. Consider now the placement of the ones in the vector v and the effect on the vector w. If i ones of the η_v ones of v are located in the first η_u bits of v, the vector w will have $\eta_u - i$ ones in the first η_u bits and $\eta_v - i$ ones in the remaining bits. Hence, $\eta_w = \eta_u + \eta_v - 2i$. The probability of $\eta_w = \eta_u + \eta_v - 2i$ given η_u and η_v is determined as the fraction of arrangements of η_v ones for which i ones are in the first η_u bits and the remaining $\eta_v - i$ ones are in the remaining $b - \eta_u$ bits. Equation (8) is derived by letting $b = N/2$, $\eta_u = \mu$, $\eta_v = \lambda$, and $\eta_w = \eta_R^*$.

Consider now the probability of the number of output bit changes given the number of input bit changes to the round function, $P(\eta_f|\eta_R)$. Let η_e represent the number of bit changes at the output of the expansion and let l represent the number of S-boxes which have at least one bit change at the input. Using total probability and the chain rule, it can be shown that

$$P(\eta_f|\eta_R) = \sum_{\eta_e=0}^{T} \sum_{l=0}^{M} P(\eta_f|l) \cdot P(l|\eta_e) \cdot P(\eta_e|\eta_R) \tag{9}$$

where T represents the number of bits at the output of the expansion component and M is the number of S-boxes in the substitution component. Hence, $T = \alpha \cdot (N/2) = M \cdot m$ represents the number of bits entering the substitution component.

The probability distribution of the number of changes at the output of the expansion given the number of input changes is given by

$$P(\eta_e|\eta_R) = \frac{\binom{T-N/2}{\eta_e-\eta_R} \cdot \binom{N-T}{2\eta_R-\eta_e}}{\binom{N/2}{\eta_R}} \tag{10}$$

where we have assumed that $T \leq N$. The first term in the numerator represents the number of selections of extra bit changes in the expanded vector from the bits that have been duplicated by the expansion function. To compute the number of arrangements of η_e bits from η_R bit changes at the expansion input, this is multiplied by the number of selections of the remaining bit changes in the expanded vector from the bits in the round function input which have not been duplicated. The probability is then calculated by dividing the number of suitable arrangements by the total number of selections of η_R bits from the round function input of $N/2$ bits.

The probability $P(l|\eta_e)$ is the probability that l S-boxes are affected by changes given that there are η_e changes at the output of the expansion component. This can be determined by computing the fraction of the number of selections of η_e bit changes that affect only l S-boxes. Letting $\mathcal{N}(\eta_e)$ represent the total number of selections of η_e bit changes and $\mathcal{N}(l,\eta_e)$ represent the number of selections that have bit changes at the input to l S-boxes, we get

$$P(l|\eta_e) = \mathcal{N}(l,\eta_e)/\mathcal{N}(\eta_e) \tag{11}$$

where

$$\mathcal{N}(\eta_e) = \binom{T}{\eta_e}. \tag{12}$$

From Lemma 2 in [4], $\mathcal{N}(l,\eta_e)$ may be determined by

$$\mathcal{N}(l,\eta_e) = \sum_{i=M-l}^{M} (-1)^{i-(M-l)} \binom{i}{M-l} \binom{M}{i} \binom{(M-i)m}{\eta_e}. \tag{13}$$

The probability distribution of round function output changes given the number of affected S-boxes, represented by $P(\eta_f|l)$, can be determined by counting over all combinations of output changes η_f from l S-boxes. Let $\mathbf{d} = [d_1 d_2 ... d_l]$ where $d_i \in \{1, ..., n\}$ is the number of output changes, $wt(\Delta\mathbf{Y})$, in the i-th S-box that has a non-zero input change. Now define

$$\Lambda = \{\mathbf{d} | \sum_{i=1}^{l} d_i = \eta_f\} \tag{14}$$

to represent the values of $\mathbf{d}$ for which there are a total of η_f output bit changes. Hence,

$$P(\eta_f|l) = \sum_{\mathbf{d}\in\Lambda} P(\mathbf{d}) \tag{15}$$

where $P(\mathbf{d})$ represents the probability of a particular $\mathbf{d}$ occurring and is given by

$$P(\mathbf{d}) = \prod_{i=1}^{l} P_D(d_i). \tag{16}$$

Using equations (4) to (16), we can now iteratively determine the probability distribution of bit changes, $P(\eta_L, \eta_R)$, for each round in the cipher and, subsequently, the expected number of bit changes after each round. Consequently, letting $E\{\cdot\}$ represent the expectation operation, the avalanche probability after a particular round can be determined from

$$P_{av} = E\{\eta_L + \eta_R\}/N \tag{17}$$

given $\eta_L = 1$ and $\eta_R = 0$ at the input to round 1 if the one bit change occurs in the left half of the plaintext, or $\eta_L = 0$ and $\eta_R = 1$ at the input to round 1 if the one bit change occurs in the right half of the plaintext.

IV. Analysis of the Results

In this section, we present the results of the computations of the preceding section for various ciphers with different parameters. We shall use a 64-bit cipher as a basis for comparison.

When computing the avalanche probability, one might consider one bit changes on either the left or right side of the plaintext. In fact, it can be shown that

$$P_{av}[right] = P^*_{av}[left] \tag{18}$$

where P_{av} and P^*_{av} represent the avalanche probabilities after i and $i+1$ rounds, respectively, and the keywords *left* and *right* indicate which half of the plaintext has the one bit change. This results from the fact that a one bit change on the left side manifests itself as a one bit change to the right side of the input to round 2. Since a DES-like cipher will have weaker avalanche for a bit change on the left, we shall consider the left side avalanche probability to be the property of interest.

In Figure 2, we present a plot of the avalanche probability versus the number of rounds in the cipher where the S-box dimensions are of the form $m \times 4$. Three cases are illustrated: $m = 4$, $m = 6$, and $m = 8$. All results presented are based on the plaintext bit change occurring in the left half. In all three cases, the avalanche probability is converging towards the desired value of 1/2. However, it is clear that the larger the value of m, the faster the convergence. Similar results were observed for ciphers based on $m \times 8$ S-boxes where $m = 8$, $m = 12$, and $m = 16$.

It is not surprising that performance is improved as m increases since a larger S-box input increases the diffusion of bit changes. Note also that Figure 3, which compares ciphers using 8×4 and 8×8 S-boxes, suggests that the S-boxes with larger number of outputs improve the avalanche probability convergence. Again this is perhaps not surprising and is due to the improved diffusion of bit changes. However, Figures 2 and 3 suggest that the effects of increasing the S-box input size appear to be more dramatic than an increase in the number of output bits. Unfortunately, for cipher implementations where look-up tables are used for

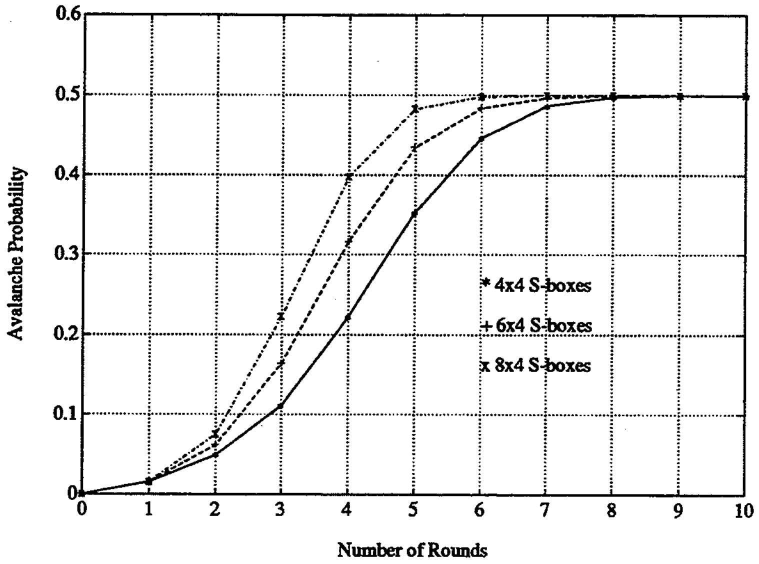


Figure 2: Theoretical Avalanche for DES-like Ciphers with $m \times 4$ S-boxes

S-boxes, the amount of memory required increases exponentially in the size of S-box input and only linearly in the size of the output. Hence, while the size of the S-box input is limited by practical considerations, the output can be more freely expanded to improve the cipher security properties. This suggests that the best combination of efficiency and security (in relation to avalanche) is given by ciphers using symmetric $n \times n$ S-boxes.

Consider now a comparison between the performance of a DES-like cipher versus a basic substitution-permutation network such as discussed in [4]. (Basic SPNs do not have the structure of Figure 1: each round consists of substitution on the entire block using N/n $n \times n$ S-boxes followed by a permutation on the entire block.) The cases for 4×4 and 8×8 S-boxes are illustrated in Figure 4. For 4×4 S-boxes there is little difference between the two ciphers. In fact, considering the relationship of (18), if the change is on the right side, then the DES-like cipher actually performs significantly better. This is perhaps surprising: since, in a basic SPN, the round function operates on the full block and, in a DES-like cipher, the round function only operates on half the block, it seems reasonable to assume that an SPN would display good cryptographic properties in fewer rounds. For the case of the larger 8×8 S-boxes, the SPN clearly has better performance than the DES-like cipher.

Of course, any cipher is a deterministic structure based on a fixed set of S-boxes, permutation

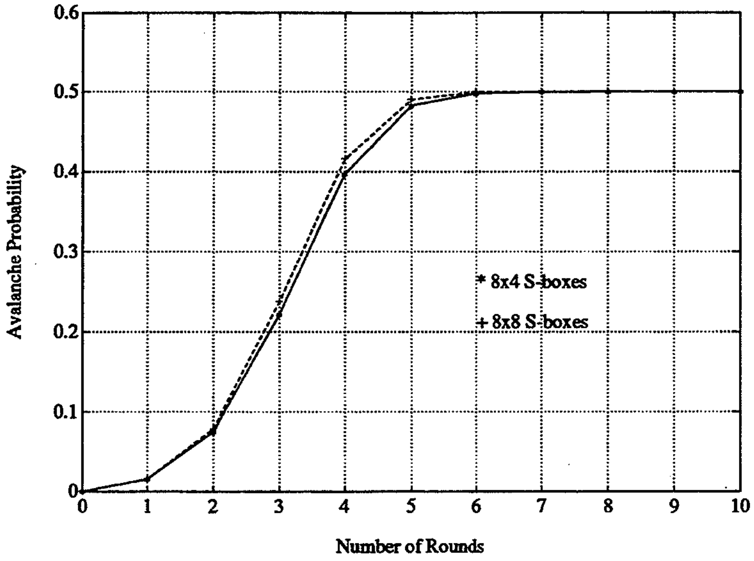


Figure 3: Theoretical Avalanche for DES-like Ciphers with 8 × 4 and 8 × 8 S-boxes

and expansion. The effect of the model, which treats these components as random variables, is to smooth out the advantages or disadvantages of particular fixed components. Although it is impossible to exactly model all ciphers using a general model, it appears that the generalizations made in this model are not only intuitively reasonable but experimental results suggest that they provide a reasonable approximation of the behaviour of a DES-like cipher. Nevertheless, it seems reasonable to expect that the model is, in fact, pessimistic and that the careful selection of S-boxes, permutations, and expansion mappings is likely to improve the performance of the cipher in relation to the avalanche characteristics [4]. In the next section, we examine the modelling of "diffusive" S-boxes and demonstrate that, indeed, S-box properties can be utilized to improve the avalanche characteristics of a DES-like cipher.

V. Improving Avalanche by Using Diffusive S-boxes

A discussion on improving the avalanche characteristics of an SPN by selecting diffusive S-boxes is contained in [4]. In this section, we consider the application of such S-boxes to a DES-like cipher. Consider the following S-box diffusion property referred to as guaranteed avalanche [4] and note that guaranteed avalanche order 2 is an acknowledged DES S-box criterion [6].

Definition 3: An S-box satifies the property of *guaranteed avalanche* of order γ if, for a one bit

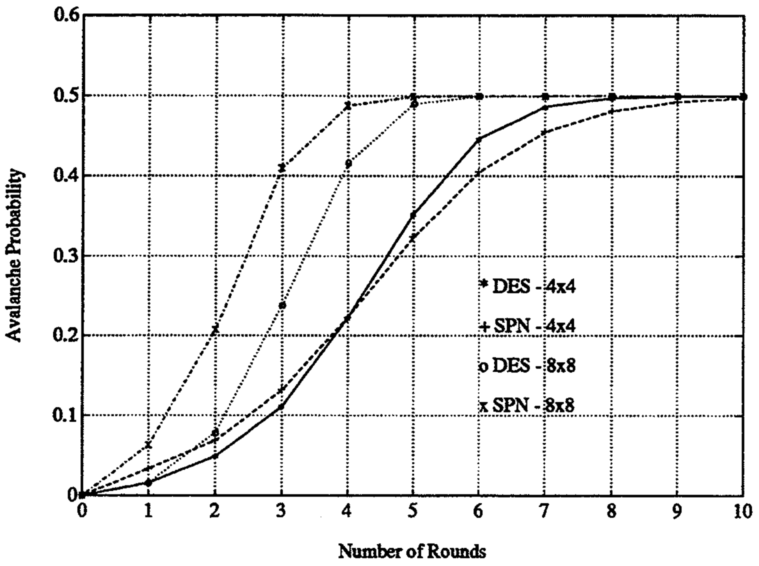


Figure 4: Theoretical Avalanche of DES-like Ciphers vs. Basic SPN Ciphers

input change, at least γ output bits change, i.e., $wt(\Delta\mathbf{X}) = 1 \Rightarrow wt(\Delta\mathbf{Y}) \geq \gamma$.

Consider now the development of a model for an S-box satisfying guaranteed avalanche order γ, $\gamma \geq 1$. The probability distribution $P_D(d)$ can be replaced by probability distributions for the number of output bit changes conditioned on the number of input changes. Clearly, $P_D(D = 0|wt(\Delta\mathbf{X}) = 0) = 1$ and $P_D(D = d|wt(\Delta\mathbf{X}) = 0) = 0$ for $d > 0$. Now let $P'_D(d) \equiv P_D(D = d|wt(\Delta\mathbf{X}) = 1)$ and $P''_D(d) \equiv P_D(D = d|wt(\Delta\mathbf{X}) > 1)$. The conditional probabilities for the number of output changes is then given by

$$P'_D(d) = \begin{cases} 0 & ,d < \gamma \\ \frac{\binom{n}{d}}{\sum_{i=\gamma}^{n} \binom{n}{i}} & ,d \geq \gamma \end{cases} \tag{19}$$

and

$$P''_D(d) = \begin{cases} \frac{2^{m-n}-1}{2^m-1-m} & ,d = 0 \\ \frac{\binom{n}{d}2^{m-n}-mP'_D(d)}{2^m-1-m} & ,d \geq 1. \end{cases} \tag{20}$$

Consider first the expression for P'_D in (19). The case of $d < \gamma$ arises simply from the definition of guaranteed avalanche; the case for $d \geq \gamma$ is derived by assuming that the selection of $\Delta\mathbf{Y}$ is uniformly distributed over the set of values such that $D = wt(\Delta\mathbf{Y}) \geq \gamma$. Considering now the expression for P''_D, the denominator of (20) represents the number of values of $\Delta\mathbf{X}$ for which $wt(\Delta\mathbf{X}) > 1$ and the numerator represents the number of values of $\Delta\mathbf{Y}$ for which $wt(\Delta\mathbf{Y}) = d$, scaled by factor 2^{m-n} and adjusted to remove the expected number of $\Delta\mathbf{Y}$ values used for the

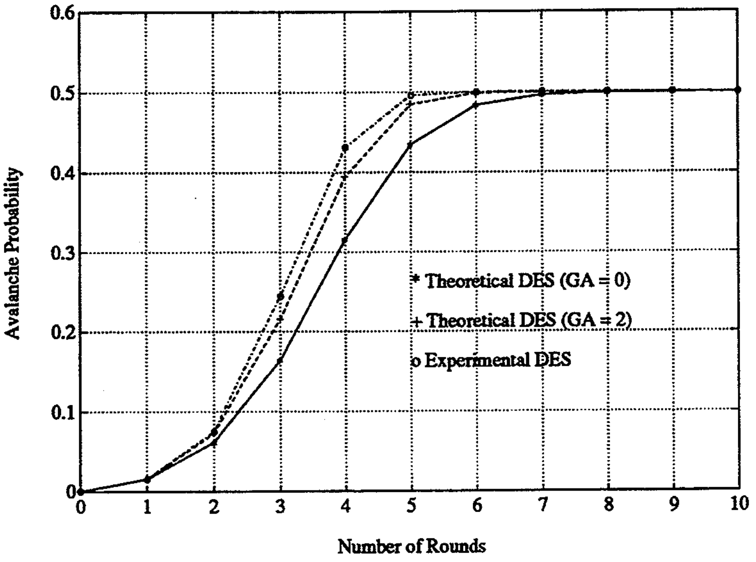


Figure 5: Avalanche for DES-like Ciphers with Diffusive S-boxes

values of $\Delta\mathbf{X}$ for which $wt(\Delta\mathbf{X}) \leq 1$.

The iterative computation of the avalanche probability follows similarly to the previous development: equations (4) through (8) are equally applicable. However, (9) must be modified to consider separately the cases of one bit input changes and more than one bit input changes to the S-boxes. Let l' represent the number of S-boxes for which $wt(\Delta\mathbf{X}) = 1$ and, as before, let l represent the number of S-boxes for which $wt(\Delta\mathbf{X}) \geq 1$. Hence, (9) becomes

$$P(\eta_f|\eta_R) = \sum_{\eta_e=0}^{T} \sum_{l=0}^{M} \sum_{l'=0}^{l} P(\eta_f|l',l) \cdot P(l',l|\eta_e) \cdot P(\eta_e|\eta_R). \tag{21}$$

The probability $P(\eta_e|\eta_R)$ may be computed as previously outlined in equation (10).

The probability $P(l',l|\eta_e)$ can be determined by

$$P(l',l|\eta_e) = \mathcal{N}(l',l,\eta_e)/\mathcal{N}(\eta_e) \tag{22}$$

where $\mathcal{N}(\eta_e)$ is the number of selections of η_e bit changes and $\mathcal{N}(l',l,\eta_e)$ is the number of selections of changes of η_e bits such that l S-boxes are affected by changes and l' S-boxes have a exactly a one bit input change. $\mathcal{N}(\eta_e)$ is given by (12) and, based on Lemma 4 in [4], $\mathcal{N}(l',l,\eta_e)$ is computed by

$$\mathcal{N}(l',l,\eta_e) = \binom{M}{l} \sum_{i=l'}^{l} (-1)^{i-l'} \binom{i}{l'} \binom{l}{i} m^i \sum_{j=0}^{l-i} (-1)^j \binom{l-i}{j} \binom{(l-i-j)m}{\eta_e - i}. \tag{23}$$

In order to determine $P(\eta_f|l',l)$, define the vector $\mathbf{d}' = [d'_1 d'_2 ... d'_{l'}]$ such that $d'_i \in \{\gamma, ..., n\}$ represents the number of output changes, $wt(\Delta\mathbf{Y})$, of the i-th S-box for which $wt(\Delta\mathbf{X}) = 1$. Similarly, define the vector $\mathbf{d}'' = [d''_1 d''_2 ... d''_{l-l'}]$ such that $d''_i \in \{1, ..., n\}$ represents the number of output changes, $wt(\Delta\mathbf{Y})$, of the i-th S-box for which $wt(\Delta\mathbf{X}) > 1$. Then

$$P(\eta_f|l',l) = \sum_{(\mathbf{d}',\mathbf{d}'')\in\Lambda^*} P(\mathbf{d}',\mathbf{d}'') \tag{24}$$

where

$$\Lambda^* = \left\{ (\mathbf{d}',\mathbf{d}'') | \sum_{i=1}^{l'} d'_i + \sum_{i=1}^{l-l'} d''_i = \eta_f \right\} \tag{25}$$

with the probability $P(\mathbf{d}',\mathbf{d}'')$ given by

$$P(\mathbf{d}',\mathbf{d}'') = \left[\prod_{i=1}^{l'} P'_D(d'_i) \right] \left[\prod_{i=1}^{l-l'} P''_D(d''_i) \right]. \tag{26}$$

Methods for improving the efficiency of the computation are given in [4].

Similarly to the previous development, (4) can be used to iteratively compute the avalanche probability given a plaintext bit change in either the left or right half. For example, results have been computed for a 64-bit cipher using 6 × 4 S-boxes, both for the original S-box model with no diffusion (i.e., $\gamma = 0$) and for the S-box model based on guaranteed avalanche order $\gamma = 2$. This is illustrated in Figure 5. There is a clear improvement in the avalanche performance for the cipher constructed using diffusive S-boxes over a cipher without difffusive S-boxes. As well, for comparison, since DES S-boxes satisfy $\gamma = 2$, experimental results for DES based on 10^4 pairs of plaintexts are also shown. While the theoretical and experimental results for DES are close, it is not surprising that experimental results on DES are slightly better than the theoretical model with diffusive S-boxes. To more accurately model DES, the model would have to incorporate a fixed representation of the DES expansion and permutation instead of treating the expansion and permutation as random and averaging over all possibilities, good and bad.

VI. Conclusion

We have modelled the avalanche characteristics for DES-like block ciphers and, consequently, analyzed the performance of the ciphers in response to variations in parameters such as the

S-box dimensions and properties. The results suggest that large, symmetric S-boxes provide the best combination of cipher efficiency and the strength of a cipher's avalanche. As well, the model is extended and used to demonstrate that selecting diffusive S-boxes is also effective in improving the avalanche characteristics of a DES-like cipher.

References

[1] H. Feistel and W.A. Notz and J.L. Smith, "Some cryptographic techniques for machine-to-machine data communications", *Proceedings of the IEEE*, vol. 63, no. 11, pp. 1545-1554, 1975.

[2] "Data Encryption Standard (DES)", *Federal Information Processing Standard Publication 46*, National Bureau of Standards, 1977.

[3] H. Feistel, "Cryptography and computer privacy", *Scientific American*, vol. 228, no. 5, pp. 15-23, 1973.

[4] H.M. Heys and S.E. Tavares, "Avalanche characteristics of substitution-permutation encryption networks", *IEEE Transactions on Computers*, vol. 44, no. 9, pp. 1131-1139, 1995.

[5] A.F. Webster and S.E. Tavares, "On the design of s-boxes", *Proceedings of CRYPTO '85*, Springer-Verlag, Berlin, pp. 523-534, 1986.

[6] E.F. Brickell and J.H. Moore and M.R. Purtill, "Structures in the s-boxes of DES ", *Advances in Cryptology: Proceedings of CRYPTO '86*, Springer-Verlag, Berlin, pp. 3-8, 1987.

Towards Provable Security for Feistel Ciphers

Abstract

Serge Vaudenay*

Ecole Normale Supérieure — DMI
45, rue d'Ulm
75230 Paris Cedex 5 France
Serge.Vaudenay@ens.fr

In the early 40's, the second world war raised the big issue of providing secrecy for electronic telecommunication. Of course, this was for military purposes, but it also initiated similar inquiries for comercial use. In 1949, Shannon published the first (public) scientific treatment of encryption [15]. He proved that the very simple Vernam Cipher (also known as *one-time pad*) provides *perfect secrecy* [18]. This encryption algorithm is however not really an encryption function because the sender and the reciever need to be synchronized. It also uses a quite cumbersome huge secret key.

For banking matters, the U.S. Government developped in 1977 the Data Encryption Standard (DES) which is a 64-bit to 64-bit encryption function that uses a small secret key (namely 56 bits) [1]. This was build around the Feistel scheme (from the name of the designer of the preliminary version), which appears to be very efficient. It however has no proven security, and its development criteria have been kept secret. Since then, researchers from this area have been working on its cryptanalysis.

During twenty years, only two important approachs inspired new knowledge about the security of DES. First, Biham and Shamir discovered in the early 90's the notion of *differential cryptanalysis* which gave the first (theoretical) attack against DES better than exhaustive search (namely with complexity 2^{47} instead of 2^{55}) [3, 4]. Later on, Matsui discovered a quite dual approach, called *linear cryptanalysis* which raised the first experimental attack of DES (with complexity 2^{43}) [8].

Now, the DES is getting old, and we have new applications for encryption. For instance, security purposes on Internet require high rate encryption. We thus need a global approach for security on block ciphers.

In this talk, we investigate different approachs for providing security against both differential and linear cryptanalysis. First, we review a heuristic approach

Laboratoire d'Informatique de l'Ecole Normale Supérieure, research group affiliated with the CNRS that consists in designing strong computation boxes in strong computation networks. We use the strength analysis from Nyberg [11], Nyberg and Knudsen [12] and Chabaud and Vaudenay [5]. We also use the analysis of the network from Heys and Tavares [6].

Second, we review another approach for getting proven security against differential and linear cryptanalysis from a Theorem due to Nyberg and Knudsen [12] recently improved by Aoki and Ohta [2]. We also discuss about a nice construction from Matsui [9].

We also try to generalize the analysis method into statistical attacks (by using the general frame independently found by Vaudenay [16, 17] and Murphy, Piper, Walker and Wild [10]) to get other design criteria.

Finally, we discuss about a theoretical approach due to Luby and Rackoff [7] which has been improved by Patarin [13] proved that a n-bit to n-bit three-round Feistel scheme is secure for about $2^{n/4}$ uses when using a $3\frac{n}{2}2^{n/2}$-bit secret key. Patarin also conjectured that some special four-round Feistel scheme may be secure for $2^{n/2}$ uses with a $n2^{n/2}$-bit key [14]. This would correspond to the strongest possible security.

References

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

[2] K. Aoki, K. Ohta. Strict evaluation of the maximum average of differential probability and the maximum average of linear probability. *IEICE Transactions on Fundamentals*, vol. E80-A, pp. 1–8, 1997.

[3] E. Biham, A. Shamir. Differential cryptanalysis of the full 16-round DES. In *Advances in Cryptology CRYPTO'92*, Santa Barbara, California, U.S.A., Lectures Notes in Computer Science 740, pp. 487–496, Springer-Verlag, 1993.

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

[5] F. Chabaud, S. Vaudenay. Links between differential and linear cryptanalysis. In *Advances in Cryptology EUROCRYPT'94*, Perugia, Italy, Lectures Notes in Computer Science 950, pp. 356–365, Springer-Verlag, 1995.

[6] H. M. Heys, S. E. Tavares. Substitution-Permutation Networks resistant to differential and linear cryptanalysis. *Journal of Cryptology*, vol. 9, pp. 1–19, 1996.

[7] M. Luby, C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, vol. 17, pp. 373–386, 1988.

[8] M. Matsui. The first experimental cryptanalysis of the Data Encryption Standard. In *Advances in Cryptology CRYPTO'94*, Santa Barbara, California, U.S.A., Lectures Notes in Computer Science 839, pp. 1–11, Springer-Verlag, 1994.

[9] M. Matsui. New structure of block ciphers with provable security against differential and linear cryptanalysis. In *Fast Software Encryption*, Cambridge, United Kingdom, Lectures Notes in Computer Science 1039, pp. 205–218, Springer-Verlag, 1996.

[10] S. Murphy, F. Piper, M. Walker, P. Wild. Likehood estimation for block cipher keys. Unpublished.

[11] K. Nyberg. Perfect nonlinear S-boxes. In *Advances in Cryptology EUROCRYPT'91*, Brighton, United Kingdom, Lectures Notes in Computer Science 547, pp. 378–385, Springer-Verlag, 1991.

[12] K. Nyberg, L. R. Knudsen. Provable security against a differential cryptanalysis. *Journal of Cryptology*, vol. 8, pp. 27–37, 1995.

[13] J. Patarin. *Etude des Générateurs de Permutations Basés sur le Schéma du D.E.S.*, Thèse de Doctorat de l'Université de Paris 6, 1991.

[14] J. Patarin. In *Advances in Cryptology EUROCRYPT'92*, Balatonfüred, Hungary, Lectures Notes in Computer Science 658, pp. 256–266, Springer-Verlag, 1993.

[15] C. E. Shannon. Communication theory of secrecy systems. *Bell system technical journal*, vol. 28, pp. 656–715, 1949.

[16] S. Vaudenay. *La Sécurité des Primitives Cryptographiques*, Thèse de Doctorat de l'Université de Paris 7, Technical Report LIENS-95-10 of the Laboratoire d'Informatique de l'Ecole Normale Supérieure, 1995.

[17] S. Vaudenay. An experiment on DES — Statistical cryptanalysis. In *3rd ACM Conference on Computer and Communications Security*, New Delhi, India, pp. 139–147, ACM Press, 1996.

[18] G. S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *Journal of the American Institute of Electrical Engineers*, vol. 45, pp. 109–115, 1926.