A New Class of Substitution-Permutation Networks

A. M. Youssef[†], S. E. Tavares[†], and H.M. Heys[††]

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

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

Abstract: In this paper we propose a special class of substitution-permutation encryption networks. This class has the advantage that the same network can be used to perform both the encryption and the decryption operations. We determine the cryptographic properties of these networks such as avalanche characteristics, expected cycle length and the resistance to both differential and linear cryptanalysis. Further, it is shown that using an appropriate linear transformation between rounds is effective in improving the resistance in relation to these two attacks. A key scheduling algorithm which satisfies certain design principles is also proposed.

1. Introduction

Feistel [6] was the first to suggest that a basic substitution-permutation network (SPN) consisting of iterative rounds of nonlinear substitutions (s-boxes) connected by bit permutations was a simple, effective implementation of a private-key block cipher. The SPN structure is directly based on Shannon's principle of a mixing transformation using the concepts of "confusion" and "diffusion" [22]. Letting N represent the block size of a basic SPN consisting of R rounds of $n \times n$ s-boxes, a simple example of an SPN with $N = 16$, $n = 4$, and $R = 3$ is illustrated in Figure 1. Keying the network can be accomplished by XORing the key bits with the data bits before each round of substitution and after the last round. The key bits associated with each round are derived from the master key according to the key scheduling algorithm.

One advantage of the basic SPN model is that it is a simple, yet elegant, structure for which it is generally possible to prove security properties. Indeed, it has been shown that a basic SPN can be constructed to possess good cryptographic properties such as completeness or nondegeneracy [10], adherence to the avalanche criterion [9], and resistance to differential and linear cryptanalysis [8].

The basic SPN architecture differs from a DES-like architecture in which the substitutions and permutations, used as a mixing transformation, operate on only half of the block at a time. Since SPNs do not have this last property, in general, SPNs need two different modules for the encryption and the decryption operations. In an SPN, decryption is performed by running the data backwards through the inverse network (i.e., applying the key scheduling algorithm in reverse and using the inverse s-boxes and the inverse permutation layer). In a DES-like cipher, the inverse s-boxes and inverse permutation are not required. Hence, a practical disadvantage of the basic SPN architecture compared with the DES-like architecture is that both the s-boxes and their inverses must be located in the same encryption hardware or software. The resulting extra memory or power consumption requirements may render this solution less attractive in some situations especially for hardware implementations.

One proposal to overcome this problem is to use a single s-box and its inverse for both the encryption and the decryption. This idea was employed in SAFER[13]. Unfortunately, in SAFER, the encryption and the decryption are different and one still needs two different hardware modules.

In this paper, we introduce a special class of substitution-permutation networks. This class has the advantage that the same network can be used to perform both the encryption and the decryption operations. The basic idea is to use involution substitution layers and involution permutation layers or linear transformations. We investigate the resistance of these networks to both differential and linear cryptanalysis: it is shown that using an appropriate linear transformation between rounds is effective in improving the security of the SPNs in relation to these two attacks. This paper also demonstrates the effectiveness of the proposed linear transformation in improving the avalanche properties of the cipher and further results suggest that the cyclic properties of the overall network are not negatively influenced by the cyclic properties of the involution s-boxes. As well, a key scheduling algorithm is proposed that has the advantages of preventing weak keys and ensuring that, given that key bits in a particular round are compromised, it is hard to get any information about the key bits of other rounds.

2. S-boxes

2.1 Semi-Involution Functions

It is possible to construct SPNs which do not require inverse s-boxes if the s-boxes in the network belong to the class of functions that we refer to as semi-involution functions. Such functions have the property that their inverses can be easily obtained by a simple XOR operation on the function input and output. Hence, differences between the s-boxes in the encryption network and the decryption network can be accommodated by incorporating the XOR into the application of the round key bits.

Definition: A bijective function $\pi : Z_2^n \rightarrow Z_2^n$ is called a semi-involution function if

$$\pi^{-1}(X) = \pi(X \oplus a) \oplus b \tag{1}$$

for some constants $a, b \in Z_2^n$.

Involution functions are the sub-class of semi-involution functions for which $a = b = 0$.

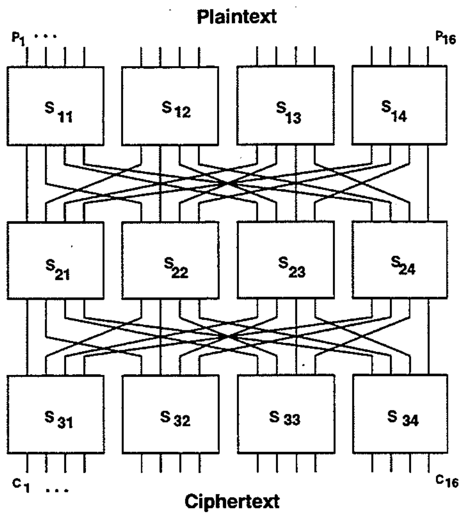

Figure 1: SPN with $N = 16$, $n = 4$, and $R = 3$.

Lemma 2.1: A semi-involution function as defined above has $a \oplus b$ as a linear structure.

Proof: Let $Y = \pi^{-1}(X)$ and, from (1), we have $Y \oplus b = \pi(X \oplus a)$. Therefore, $X = \pi^{-1}(Y \oplus b) \oplus a$. Hence, $\pi(Y) = \pi^{-1}(Y \oplus b) \oplus a$. Now replacing $Y \oplus b$ with X gives $\pi(X \oplus b) = \pi^{-1}(X) \oplus a$. From (1), $\pi(X \oplus a) \oplus b = \pi(X \oplus b) \oplus a$. Replacing X with $X \oplus b$ gives

$$\pi(X \oplus a \oplus b) = \pi(X) \oplus a \oplus b, \tag{2}$$

which is the definition of a linear structure [5], [16]. □

Thus a semi-involution function has $N_{\triangle X \triangle Y} = 2^n$ where $N_{\triangle X \triangle Y}$ is the XOR difference distribution table entry[3] for input $\triangle X = a \oplus b$ and $\triangle Y = a \oplus b$. For $a \oplus b \neq \mathbf{0}$ this renders the SPN trivially broken by differential cryptanalysis. This means that, if we want to use the same SPN for both the encryption and decryption, then only semi-involution s-boxes with $a = b$ can be used.

The following lemma shows how the useful class of semi-involution functions can be obtained from involution functions.

Lemma 2.2: Let $\phi : Z_2^n \rightarrow Z_2^n$ be an involution function, then the function $\pi(X) = \phi(X) \oplus a$ is a semi-involution function such that $a = b$, i.e., $\pi^{-1}(X) = \pi(X \oplus a) \oplus a$.

Proof: From the definition of involution functions, $\phi^2(X) = X$. Hence, $\pi(\pi(X) \oplus a) \oplus a = X$. Replacing X with $X \oplus a$ gives $\pi(X \oplus a) \oplus a = \pi^{-1}(X)$. □

Lemma 2.2 is important, not only because it provides an easy way to generate the useful class of semi-involution functions from involution functions, but also because it implies that the functions $\phi(X)$ and $\pi(X)$ belong to the same cryptographic class and hence they have the same linear approximation table[15], and the same XOR difference distribution table[3].

The only cryptographic difference between involution s-boxes and semi-involution s-boxes with $a = b, a \neq \mathbf{0}$, is their cyclic properties. All cycles of involution functions have length one or two. In SPNs where the key bits are XORed with the data bits at the s-box input, if we assume that all the key bits are equi-probable, then both the SPNs built using semi-involution s-boxes with $a = b \neq 0$ and the SPNs built using involution s-boxes will have the same cryptographic properties. In the rest of the paper we will focus on the class of SPNs that use involution s-boxes.

Remark: In an SPN where the s-boxes are keyed by selecting between sets of mappings (and not XORing the key bits with the data bits), then the cyclic properties of involution and semi-involution s-boxes may be an important difference in their cryptographic properties. Unfortunately this class of SPNs requires storage for a large set of s-boxes and, hence, is not attractive for practical implementations.

An interesting class of involution mappings is the inversion mapping in $\boldsymbol{GF(2^n)}$ defined as [18]:

$$\pi(X) = \begin{cases} X^{-1}, & X \neq 0 \\ 0, & X = 0. \end{cases} \tag{3}$$

Different cryptographic properties of this mapping were studied in [18]. This inversion mapping is differentially 2-uniform if n is odd and it is differentially 4-uniform if n is even. The nonlinearity of this mapping is given by $\mathcal{NL}(\pi) \geq 2^{n-1} - 2^{n/2}$.

The above class of s-boxes can be generated using different irreducible polynomials. The number of monic polynomials of degree n which are irreducible over $GF(q)$, where q is any prime power, is given by [1], [12]:

$$\frac{1}{n} \sum_{d|m} \mu(d) q^{m/d} \tag{4}$$

where $\mu(d)$ is the Möbius function given by

$$\mu(d) \begin{cases} 1 & , d = 1 \\ (-1)^r & , d \; is \; a \; product \; of \; r \; distinct \; primes \\ 0 & , otherwise. \end{cases} \tag{5}$$

For $n = 8$, we have 30 irreducible polynomials of degree 8 and hence we can generate 30 such s-boxes. All these 30 s-boxes have nonlinearity equal to 112 and maximum XOR table entry equal to 4. In order to frustrate possible algebraic attacks, the SPN should use s-boxes generated using different irreducible polynomials. Another approach is to use randomly generated s-boxes so that the overall cipher would not have any easy algebraic description. In section 2.3 we study some of the cryptographic properties of such randomly generated involution s-boxes.

Lemma 2.3: The number of involution functions $\pi : Z_2^n \rightarrow Z_2^n$ is given by

$$\sum_{i=0}^{2^{(n-1)}} \frac{2^n!}{(2^{n-1} - i)! \, (2i)! \, 2^{2^{n-1} - i}} \; . \tag{6}$$

Proof: See the appendix. □

2.2 Equivalence Classes

Two s-boxes π_1, π_2 are said to belong to the same cryptographic class if

$$\pi_2(X) = \pi_1(X \oplus a) \oplus b \tag{7}$$

for arbitrary constants $a, b \in Z_2^n$.

The use of s-boxes within the same cryptographic classes was suggested as a means to design SPNs that are resistant to differential cryptanalysis [23]. Unfortunately, involution s-boxes can not be used in such SPNs because, as shown in the following lemma, if two involution s-boxes belong to the same cryptographic class then they posse a linear structure.

Lemma 2.4: If π_1 and π_2 are both involution mappings and

$$\pi_2(X) = \pi_1(X \oplus a) \oplus b \tag{8}$$

then π_1, π_2 have $a \oplus b$ as a linear structure.

Proof: By noting that $\pi_2(X) = \pi_1(X \oplus a) \oplus b$ then we have $\pi_2^2(X) = \pi_1(\pi_1(X \oplus a) \oplus a \oplus b) \oplus b$. But we also have $\pi_2^2(X) = X$ and, hence, $\pi_1(\pi_1(X \oplus a) \oplus a \oplus b) \oplus b = X$. Thus, we have $\pi_1(X \oplus a) \oplus a \oplus b = \pi_1^{-1}(X \oplus b)$. Replacing $X \oplus b$ by X and noting that $\pi_1^{-1}(X) = \pi_1(X)$ gives

$$\pi_1(X \oplus a \oplus b) = \pi_1(X) \oplus a \oplus b \tag{9}$$

which is the definition of a linear structure. By a similar argument, one can show that π_2 also has $a \oplus b$ as a linear structure. □

2.3 Number of Fixed Points

Involution s-boxes have the characteristic that all cycles are of length one or two and, as will be shown, have a larger expected number of fixed points than a randomly chosen s-box. Although there is no known effective cryptanalytic attack directly based on the existence of fixed points in the s-boxes, it is of interest to determine if a large number of fixed points affects other cryptographic properties, such as the nonlinearity and the maximum XOR table entry, that lead to other cryptographic attacks .

Figure 2 shows the experimental results for the average nonlinearity and the average maximum XOR table entry as a function of the number of fixed points for 8-bit random bijections and 8-bit random involutions. One thousand random bijective s-boxes and one thousand random involution s-boxes were tested for each point. The graphs were derived by incrementing the number of fixed points by 2. The graphs clearly indicate a strong correlation between the cryptographic properties and the number of fixed points and suggest that the s-boxes should be chosen to contain few fixed points.

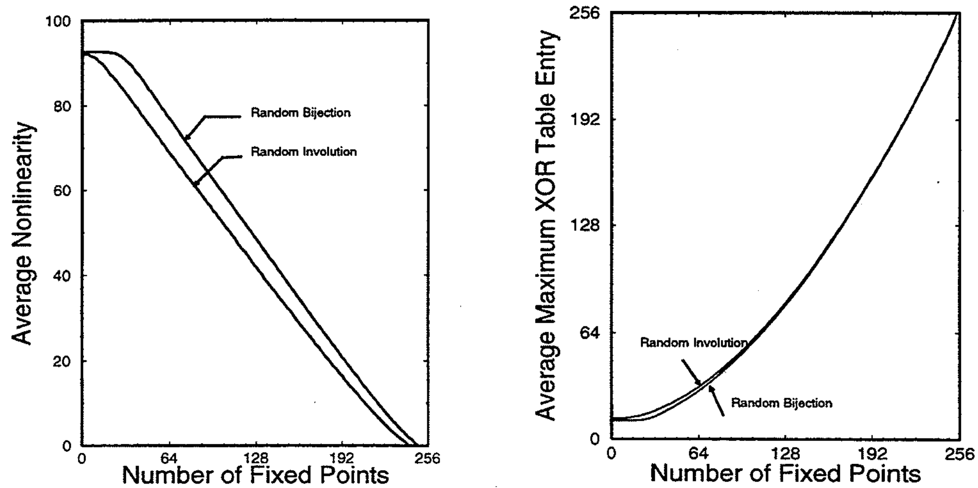

Figure 2: Average Nonlinearity and Average Maximum XOR Table Entry Versus the Number of Fixed Points ($n = 8$)

We now calculate the expected number of fixed points for a random bijection and for a random involution.

Lemma 2.5: The expected value of the number of fixed points for a random bijective mapping is 1 .

Proof: See the appendix. □

Similarly, one can show that the variance of the number of fixed points is also 1.

Lemma 2.6: The expected number of fixed points for a random involution mapping is given by

$$E(N_{fp}) = \sum_{i=0}^{2^{n-1}} 2i\Phi(n,i) \Big/ \sum_{i=0}^{2^{n-1}} \Phi(n,i) \tag{10}$$

where

$$\Phi(n,i) = \frac{2^i}{(2^{n-1}-i)!\,(2i)!}. \tag{11}$$

Proof: See the appendix. □

Numerical substitution in the formula above shows that the expected number of fixed points of a random involution exceeds that of a random injective mapping by a large factor. For example, an 8-bit involution mapping is expected to have about 16 fixed points. Fortunately, the construction proof of Lemma 2.3 can be used to generate involution functions with a predetermined number of fixed points. A special case of interest is involution functions with zero fixed points since this seems to optimize their cryptographic properties (see Figure. 2). The number of such functions follows from the proof of Lemma 2.3 and can be approximated using Stirling's formula as follows

$$\frac{2^n!}{2^{n-1}!2^{2^{n-1}}} \approx \sqrt{2}\left(\frac{2^n}{e}\right)^{2^{n-1}}. \tag{12}$$

3. S-box Interconnection Layer

In order to use the same SPN to perform both the encryption and the decryption operations, the s-box inter-connection layer should also be an involution mapping. One permutation layer, applicable to networks for which $N = n^2$, with nice cryptographic properties [8] and which satisfies the involution requirement is described by: output bit i of s-box j at round r is connected to input bit j of s-box i at round $r+1$.

In [8] it was shown that with such a permutation layer we can develop upper bounds on the differential characteristic probability [3] and on the probability of a linear approximation [15] as a function of the number of rounds of substitution. Unfortunately, to achieve good bounds, with a relatively small number of rounds, it is suggested to have s-boxes with a large diffusion order [8]. Letting $\triangle X$ and $\triangle Y$ denote the input change vector and the output change vector, respectively, an s-box satisfies diffusion order of λ, $\lambda \geq 0$, if for $wt(\triangle X) > 0$,

$$wt(\triangle Y) > \begin{cases} \lambda & wt(\triangle X) < \lambda + 1, \\ 0 & otherwise. \end{cases} \tag{13}$$

where $wt(\cdot)$ denotes the Hamming weight of the enclosed argument.

Our depth-first search algorithm could not find any 8×8 involution s-boxes with diffusion order greater than 1 (without the involution constraint, some 8×8 s-boxes with $\lambda = 2$ were found in [8]). As an alternative to this, the authors in [8] proposed the use of an invertible linear transformation between rounds. The SPN resistance to linear and differential cryptanalysis was very encouraging. Unfortunately, their proposed linear transformation is not very attractive in practice as it requires a bit XORing operation of all the output bits of the round.

We propose a more efficient linear transformation that runs much faster. Moreover it has improved bounds for the linear approximation and the differential characteristic. The linear transformation between rounds of s-boxes is described by

$$z(i) = \bigoplus_{l=1, l \neq i}^{M} w(l), \quad 1 \leq i \leq M \tag{14}$$

where $z(i)$ represents the i^{th} n-bit output word of the transformation, $w(i)$ is the i^{th} input word, $M = \frac{N}{n}$ denotes the number of s-boxes, and $\oplus$ denotes a bit-wise XOR operation. It is assumed that M is even. For 8×8 s-boxes this is a byte oriented operation. One can easily check that this linear transformation operation is an involution.

The linear transformation described above may be efficiently implemented by noting that each $z(i)$ could be simply determined by XORing $w(i)$ with the XOR sum of all $z(j)$, $1 \leq j \leq M$, i.e.,

$$z(i) = Q \oplus w(i), \tag{15}$$

where

$$Q = \bigoplus_{l=1}^{M} w(l). \tag{16}$$

Equation (16) above requires $(M-1)$ word-oriented XORs (which can be done in parallel in $log_2 M$ steps) and equation (15) requires M word-oriented XORs (which can be done in one step). Hence for a 64-bit SPN using 8×8 s-boxes, the above linear transformation requires $7 + 8 = 15$ byte-oriented XORs compared to $63 + 64 = 127$ bit-oriented XORs required for the linear transformation of [8].

4. Resistance to Differential and Linear Cryptanalysis

Using an approach similar to the analysis in [8], it is possible to establish upper bounds on the most likely differential characteristic and linear approximation expression using the linear transformation of (14). The results of this section are obtained by assuming that all the round keys are independent.

4.1 Differential Cryptanalysis

The following lemma gives a lower bound on the number of s-boxes involved in any 2 rounds of a differential characteristic.

Lemma 4.1: Consider an SPN with M s-boxes, $M \geq 4$. If the SPN employs the linear transformation described in (14), then the number of s-boxes involved in any 2 rounds of a differential characteristic is greater than or equal to 4.

Proof: (Sketch) From the linear transformation expression one can check that if only one s-box is involved in round r this implies that $M-1$ s-boxes are involved in round $r+1$. If 2 s-boxes are involved in round r, (14) ensures that at least 2 s-boxes will be involved in round $r+1$. The rest of the proof follows by noting that the minimum number of s-boxes involved per round is 1. □

The number of chosen plaintext/ciphertext pairs required for differential cryptanalysis of an R round SPN (based on the best *characteristic* and not the best *differential* [19],[14]) may be approximated by [3], [8]

$$N_D = \frac{1}{P_{\Omega_{R-1}}}, \tag{17}$$

where $P_{\Omega_{R-1}}$ is the probability of the best $R-1$ round characteristic. This probability can be bounded by

$$P_{\Omega_{R-1}} \leq (P_\delta)^\alpha \tag{18}$$

where the maximum s-box XOR pair probability is given by $P_\delta = \frac{M_\oplus}{2^n}$ with $M_\oplus$ denoting the maximum entry in the XOR distribution tables of the s-boxes used in the SPN and α is the total number of s-boxes involved in the characteristic. For even R, from Lemma 4.1 and assuming that only one s-box will be involved in round $R-1$ then we have

$$\alpha \geq 4\left(\frac{R}{2} - 1\right) + 1 = 2R - 3, \tag{19}$$

and, hence,

$$N_D \geq \frac{1}{(P_\delta)^{2R-3}}. \tag{20}$$

Using 8×8 involution s-boxes with maximum XOR table entry of 10 (easily found by randomly selecting involution s-boxes), an 8 round 64-bit SPN that utilizes the proposed linear transformation will have $N_D \geq 2^{60.8}$ chosen plaintext/ciphertext pairs required for differential cryptanalysis. If we use the inversion s-boxes given by (3), then we will have $N_D \geq 2^{78}$.

4.2 Linear Cryptanalysis

The following lemma gives a lower bound on the number of s-boxes involved in any 2 round linear approximation and is based on the assumption of independence between linear approximation of different rounds.

Lemma 4.2: Consider an SPN with M s-boxes, $M \geq 4$. If the SPN employs the linear transformation described in (14) then the number of s-boxes involved in any 2 rounds of a linear approximation is greater than or equal to 4.

Proof: (Sketch) If the number of s-boxes involved in round $r+1$, l, is odd, then the number of s-boxes involved in round r is $M - l$. If l is even, then the number of s-boxes involved in round r is l. The lemma above follows by considering different values for l. □

For an SPN based on $n \times n$ s-boxes, the number of known plaintexts required for the *basic* linear cryptanalysis (algorithm 1 in [15]) may be approximated by [8]

$$N_L = \frac{1}{\left|P_L - \frac{1}{2}\right|^2} \tag{21}$$

where

$$\left|P_L - \frac{1}{2}\right| \leq 2^{\alpha-1}(P_\epsilon)^\alpha \tag{22}$$

and

$$P_\epsilon = \left(\frac{2^{n-1} - \mathcal{NL}}{2^n}\right), \tag{23}$$

with $\mathcal{NL}$ denoting the minimum nonlinearity [17] of the s-boxes used in the SPN and α is the total number of s-boxes involved in the linear approximation. From the above argument we have

$$\alpha \geq 4\left(\frac{R}{2}\right) = 2R, \tag{24}$$

and, hence,

$$N_L \geq \frac{1}{2^{4R-2}\ P_\epsilon^{4R}}. \tag{25}$$

Using 8×8 involution s-boxes with nonlinearity of 98 (easily found by randomly selecting involution s-boxes), an 8 round 64-bit SPN that utilizes the proposed linear transformation will have $N_L \geq 2^{68.98}$ known plaintext/ciphertext pairs required for the *basic* linear attack. Since this number is greater than the size of the plaintext set, we interpret this to mean that the *basic* linear attack is not effective against this class of SPNs, even if we use all possible plaintexts. If we use the inversion s-boxes given by (3), then we will have $N_L \geq 2^{98}$.

Remark: There are other types of linear transformations that greatly improve the resistance of the algorithm to differential and linear cryptanalysis. An example of such transformations is the one based on Maximum Distance Separable (MDS) codes [12] described in [21]. In this case, the number of s-boxes per round involved in any linear approximation expression or a differential characteristic is equal to the number of s-boxes per round + 1, which is the maximum theoretical possible number. Unfortunately, the above linear transformation is not an involution. Moreover, it is not efficient for hardware implementation.

5. Avalanche Characteristics of the Network

An SPN is considered to display good avalanche characteristics if, for a fixed key, one bit change in the plaintext input is expected to result in close to half the ciphertext output bits changing. Good avalanche characteristics are important to ensure that a cipher is not susceptible to statistical attacks and the strength of an SPN's avalanche characteristic may be considered as a measure of the randomness of the ciphertext.

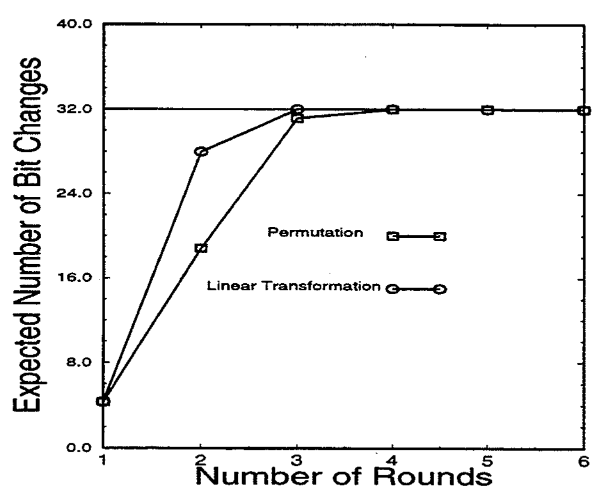

Figure 3: Expected Number of Bit Changes Versus the Number of Rounds

Figure 3 shows the experimental results for the average number of output bit changes as a function of the number of rounds for a 64-bit SPN with a permutation layer and a linear transformation layer. One thousand random chosen input pairs, different in one randomly selected bit, were used to obtain the result. The SPN used for the experiments employed 8×8 involution s-boxes with zero fixed points, nonlinearity of 96, maximum XOR table entry of 10, and a diffusion order equal to 1. The results of Figure 3 suggest that the linear transformation significantly improves the avalanche characteristics of the cipher. Analytical model for the SPN avalanche characteristics is developed in [24].

6. Cyclic Properties of the Proposed SPN

A significant difference between an involution s-box and a non-involution s-box is likely to be their cyclic properties. For a randomly chosen n-bit bijective mapping, the expected value and the variance of the number of cycles are both approximately equal to $log_e(2^n) \approx 0.69\,n$ [7]. The expected value of the cycle length is equal to $2^{n-1} + 1/2$ [4].

For an involution mapping with N_{fp} fixed points, the expected cycle length is given by

$$\frac{N_{fp} \times 1 \; + \; \left(2^n - N_{fp}\right) \times 2}{2^n} = 2 - \frac{N_{fp}}{2^n}. \tag{26}$$

and the number of cycles is given by $2^{n-1} + N_{fp}/2$.

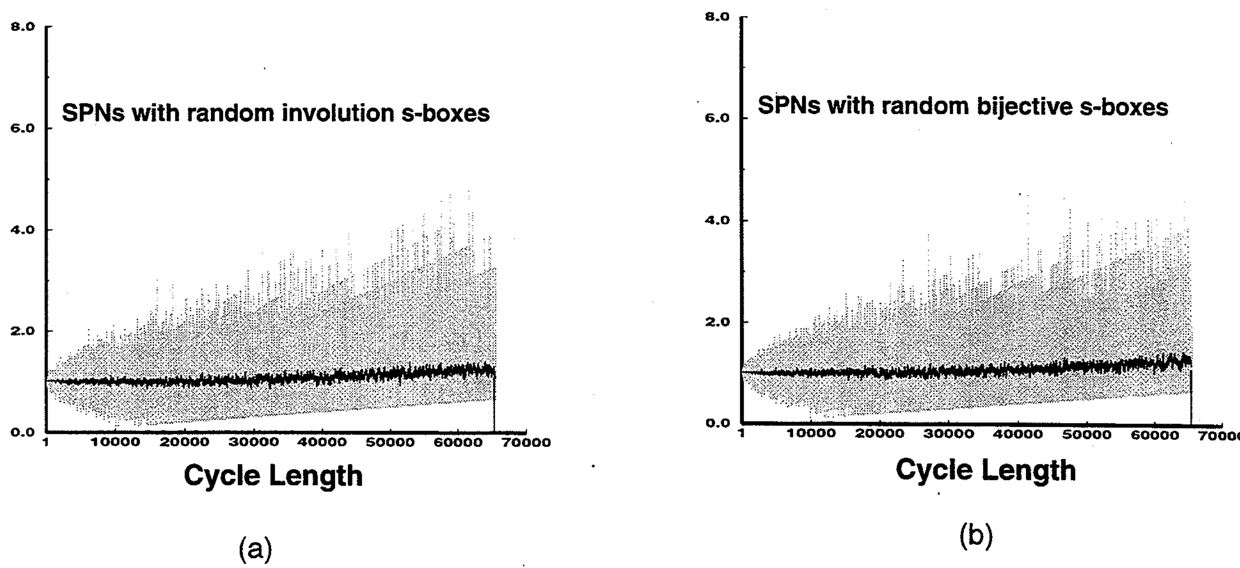

Figure 4: Distribution of Cycle Length for all 2^{16} Starting Points

In order to investigate whether the cyclic properties of involution s-boxes affect the cyclic properties of the SPN, we measured the cycle distribution for $100,000$ 16-bit SPNs with 4-rounds. Each SPN uses four 4×4 random involution s-boxes with zero fixed points, nonlinearity greater than or equal to 4 and maximum XOR table entry equal to 4. The cycle length distribution is shown in Figure 4(a) (the dark line shows the average distribution over 100 adjacent points). In this case, the average cycle length over all SPNs is equal to 32779. We performed the same experiment on $100,000$ SPNs using random bijective mappings with the same constraints on the nonlinearity and the XOR table. The simulation results are shown in Figure 4(b). The average cycle length over all SPNs is equal to 32766. It is clear that the two distributions are almost indistinguishable. This suggests that the involution s-boxes do not have a negative impact on the cyclic properties of the SPN.

7. Key Scheduling Algorithm

In our discussion, we assume that the SPN is keyed by XORing the key bits before each substitution and after the last substitution. A weak key, k_W, is any key for which $E_{k_W}(E_{k_W}(p)) = p$ for every plaintext vector p where $E_{k_W}(\cdot)$ denotes the encryption operation using the key k_W. In this section we propose a simple key scheduling algorithm for the SPN. Three design principles were employed:

(i) Prevent weak keys.

(ii) Given that some or all of the key bits at round r are compromised, it is hard to get any information about the other round keys.

Although the above key scheduling can be controlled to be relatively slow in order to make brute force attack harder [20], it is far easier and more effective to use a larger key. Using a larger key has the advantage that it does not penalize implementations which must change the key often.

In the following algorithm key denotes the user supplied key which is assumed to be of the same length as the block length of the SPN, $E_k^*(p)$ denotes the output of the SPN when it has p as an input, and the round keys are all set to k. Consider the key scheduling algorithm shown below.

$$
\begin{array}{l}
x_0 = \mathbf{0}; \\
for \quad i = 1 \quad to \quad (R+1) \\
\{ \\
\quad k_i = E^*_{key}(x_{i-1}); \\
\quad x_i = Op(x_{i-1}); \\
\}
\end{array}
$$

Figure 5: Key Scheduling Algorithms

One can assign any other arbitrary value to x_0. $Op(\cdot)$ denotes any simple operation that guarantees that all x_i's are different. By noting that E_k^* is a bijective mapping for any fixed key then all k_i's will be different which guarantees that we do not have any weak keys. An example of operation $Op(\cdot)$ is the complementing of different bits in x_0 for each i. Note that we control the key scheduling speed by controlling the number of rounds used in the encryption operation E_k^*. Also, this scheme is similar to the scheme proposed in [11].

It is also worth noting that the above keying scheme does not have the complementation property; this property makes DES susceptible to exhaustive key search of 2^{55} rather than 2^{56}. This scheme also ensures that there are no simply related keys which leads to Biham's related keys attack [2].

The key scheduling described above can be extended to accommodate the case where the user supplied key size is a multiple of the SPN block length (Keys which are not multiples can be padded to be so) .

Performance

While the usefulness of a cryptographic algorithm is based on assumptions about its security, the complexity of the cryptographic function is another feature that should not be overlooked. Table 1 shows the relative speed of Q-CAST[1] and SPNs on three platforms: an 8–bit microcontroller (Motorola 6811), a SUN SPARC workstation and a SUN ULTRA workstation. All algorithms operate on a 64–bit blocks and implemented 16 rounds.

In considering these numbers, one should take into account that the proposed SPN is a hardware oriented cipher (while Q-CAST is a software oriented cipher), and the round function of the proposed SPN provides a better degree of security than the round function of Q-CAST.

	SPN	Q-CAST
Motorola 6811	1	0.46
SUN SPARC-20	1	7.5
SUN ULTRA-1	1	1.56

Table 1 Relative Speed of the proposed SPN and Q-CAST

9. Conclusion

We have presented a special class of SPNs that have the advantage that the same network can be used to perform both the encryption and the decryption operation. The s-boxes used are involution mappings and the permutation layer is replaced by an efficient involution linear transformation layer. In a few seconds on a SPARC-20 workstation, we were able to obtain tens of 8×8 involution s-boxes with nonlinearity of 98 and maximum XOR table entry of 10. Using these s-boxes, an 8 round 64-bit SPN that utilizes the proposed linear transformation will be resistant to both the basic linear cryptanalysis and to the differential cryptanalysis based on the best $(R-1)$-round characteristic. We also confirmed that the avalanche characteristics and the cyclic properties of this special class of SPNs reveal no apparent weakness. A key scheduling algorithm which satisfies certain design principles was also proposed.

References

[1] E. R. Berlekamp. *Algebraic coding theory*. McGraw-Hill Book Inc., 1968.

[2] E. Biham. New type of cryptanalytic attacks using related keys. *Advances in cryptology: Proc. of EUROCRYPT '93, Springer-Verlag, Berlin, pp. 398–409*, 1994.

[3] E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology, vol. 4, no. 1, pp. 3–72*, 1991.

[4] D.W. Davis and G.I.P. Parkin. The average cycle size of the key stream in output feedback encipherment. *Lecture Notes in Computer Science : Proc. of the Workshop on Cryptography, Springer-Verlag, Berlin, pp.263–279*, 1982.

[1] Queen's University version of the CAST encryption algorithm

[5] J.H. Evertse. Linear structures in block ciphers. *Advances in Cryptology: Proc. of EUROCRYPT '87, Springer-Verlag, Berlin, pp.249–266*, 1988.

[6] H. Feistel. Cryptography and computer privacy. *Scientific American, 228, pp. 15–23*, 1973.

[7] W. Feller. *An Introduction to Probability Theory with Applications, Vol. I, 3rd edition.* John Wiley and Sons, New York, 1968.

[8] H.M. Heys and S.E. Tavares. The design of product ciphers resistant to differential and linear cryptanalysis. *Journal of Cryptology, Vol. 9, no. 1, pp. 1–19*, 1996.

[9] H.M. Heys and S.E. Tavares. Avalanche characteristics of substitution-permutation encryption networks. *IEEE Trans. Comp., Vol. 44, pp.1131–1139*, Sept. 1995.

[10]J.B. Kam and G.I. Davida. Structured design of substitution-permutation encryption networks. *IEEE Trans. Comp. C-28, pp.747–753*, 1979.

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

[12]F.J MacWilliams and N.J.A. Sloane. *The theory of error correcting codes.* North-Holland Publishing Company, 1977.

[13]J. Massey. SAFER K-64: a byte-oriented block-ciphering algorithm. *Fast Software Encryption, LNCS 809, Springer-Verlag, pp. 1–17*, 1994.

[14]J. L. Massey. An introduction to contemporary cryptology. *Proc. of IEEE, 76, pp.533–549*, 1988.

[15]M. Matsui. Linear cryptanalysis method for DES cipher. *Advances in Cryptology: Proc. of EUROCRYPT '93, Springer-Verlag, Berlin. pp. 386–397*, 1994.

[16]W. Meier and O. Staffelbach. Nonlinearity criteria for cryptographic functions. *Advances in Cryptology: Proc. of EUROCRYPT' 89, Springer-Verlag, pp. 549–562*, 1990.

[17]K. Nyberg. Perfect nonlinear S-boxes. *Advances in Cryptology: Proc. of EUROCRYPT '91 , Springer-Verlag, pp. 378–386*, 1992.

[18]K. Nyberg. Differentially uniform mappings for cryptography. *Advances in Cryptology: Proc. of EUROCRYPT '93, Springer-Verlag, Berlin, pp.55–64*, 1994.

[19]K. Nyberg and L.R. Knudsen. Provable security against differential cryptanalysis. *Advances in Crytology: Proc. of CRYPTO '92, Springer-Verlag, pp.566–574*, 1993.

[20]J. Quisquater, Y. Desmedt, and M. Davio. The importance of "good" key scheduling schemes. *Advances in Cryptology: Proc. of CRYPTO '85, Springer-Verlag, Berlin, pp. 537–542*, 1986.

[21]V. Rijmen, J. Daemen, B. Preneel, A. Bosselaers, and E. De Win. The cipher SHARK. *Fast Software Encryption, LNCS 1039, D. Gollmann, Ed., Springer-Verlag, pp. 99-112.*, 1996.

[22]C.E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal, Vol.28, pp. 656–715*, 1949.

[23]M. Sivabalan, S.E. Tavares, and L.E. Peppard. On the design of SP networks from an information theoretic point of view. *Advances in Cryptology: Proc. of CRYPTO '92, Springer-Verlag, Berlin, pp. 260–279*, 1993.

[24]A.M. Youssef and S.E. Tavares. Avalanche characteristics of a special class of substitution permutation networks. Technical report, Electrical and Computer Engineering department, Queen's University, Kingston, Ontario, December, 1995.

Appendix

Proof of Lemma 2.3: An involution function can only have an even number of fixed points. There are $\binom{2^n}{2i}$, $0 \leq i \leq 2^{n-1}$ ways to specify any of these $2i$ fixed points. Note also that an involution function with $2i$ fixed points must have $2^{n-1} - i$ cycles of length 2. An involution function is completely defined by specifying its fixed points and a single point on each of its $2^{n-1} - i$ cycles. Now, we will count the number of ways of assigning these $2^n - 2i$ points along the $2^{n-1} - i$ cycles. To choose the first point, pick any arbitrary point $x_0 \in Z_2^n$ such that x_0 is not equal to any of the assigned fixed points. Choose a random value $r_0 \in Z_2^n$ for $\pi(x_0)$. r_0 should not be equal to any of the fixed points. It also should not be equal to x_0. Thus there are $(2^n - 2i - 1)$ ways to choose r_0. To choose a second point, pick another arbitrary point x_1 such that $\pi(x_1)$ has not been assigned yet (this also ensures that it belongs to a new distinct cycle) and pick a random $r_1 \in Z_2^n$ for $\pi(x_1)$. Again, r_1 should satisfy the following conditions: $r_1 \neq x_1$ and it should not be equal to any of the previously assigned values for π. Proceeding as above, we have

$$\prod_{j=0}^{2^{n-1}-i-1} (2^n - 2i - 1 - 2j) = \frac{(2^n - 2i)!}{(2^{n-1} - i)!\, 2^{2^{n-1}-i}} \tag{28}$$

ways of assigning these points. Hence, the number of involution functions is given by

$$\begin{aligned} \binom{2^n}{2^n} + \sum_{i=0}^{2^{n-1}-1} \binom{2^n}{2i} \prod_{j=0}^{2^{n-1}-i-1} (2^n - 2i - 1 - 2j) \\ = \sum_{i=0}^{2^{n-1}} \binom{2^n}{2i} \frac{(2^n - 2i)!}{(2^{n-1} - i)!\, 2^{2^{n-1}-i}} \\ = \sum_{i=0}^{2^{n-1}} \frac{2^n!}{(2^{n-1} - i)!\, (2i)!\, 2^{2^{n-1}-i}}. \end{aligned} \tag{29}$$

The first term in the equation above stands for the unity bijection mapping with 2^n fixed points. □

Proof of Lemma 2.5: The number of bijective mappings with exactly t fixed points is given by (this result follows by using the inclusion-exclusion principle)

$$\sum_{i=t}^{2^n} (-1)^{i+t} \binom{i}{t} \binom{2^n}{i} (2^n - i)! \ . \tag{30}$$

The probability of having exactly t fixed points is given by the above formula divided by $2^n!$. Hence, the expected number of fixed points is given by

$$\sum_{t=0}^{2^n} t \sum_{i=t}^{2^n} \frac{(-1)^{i+t}\binom{i}{t}\binom{2^n}{i}(2^n-i)!}{2^n!}$$

$$\begin{aligned}
&= \sum_{t=0}^{2^n}\sum_{i=t}^{2^n} \frac{(-1)^{i+t}\, t \binom{i}{t}}{i!} \\
&= \sum_{t=0}^{2^n}\sum_{i=0}^{2^n} \frac{(-1)^{i+t}\, t \binom{i}{t}}{i!} \qquad (31) \\
&= \sum_{i=0}^{2^n} \frac{(-1)^i}{i!} \sum_{t=0}^{2^n} (-1)^t t \binom{i}{t} \\
&= 1.
\end{aligned}$$

The last step in the equation above follows by noting that

$$\sum_{t=0}^{i} (-1)^t \; t \binom{i}{t} = \begin{cases} -1 & i = 1, \\ 0 & otherwise. \end{cases} \qquad (32)$$

which completes the proof of the lemma. □

Proof of Lemma 2.6:

From the proof of Lemma 2.3, the number of involution functions with $2i$ fixed points is given by

$$\frac{2^n!}{(2^{n-1}-i)!\,(2i)!\,2^{2^{n-1}-i}}, \quad 0 \le i \le 2^{n-1}. \qquad (33)$$

The probability of randomly selecting an involution function with $2i$ fixed points is obtained by dividing (33) by the total number of involution functions. Thus, the expected number of fixed points for a random involution function is given by

$$\frac{\sum_{i=0}^{2^{n-1}} \frac{2^n!\; 2i}{(2^{n-1}-i)!\;(2i)!\;2^{2^{n-1}-i}}}{\sum_{i=0}^{2^{n-1}} \frac{2^n!}{(2^{n-1}-i)!\;(2i)!\,2^{2^{n-1}-i}}} = \frac{\sum_{i=0}^{2^{n-1}} \frac{2^i\;\; 2i}{(2^{n-1}-i)!\;(2i)!}}{\sum_{i=0}^{2^{n-1}} \frac{2^i}{(2^{n-1}-i)!\;(2i)!}}. \qquad (34)$$

which completes the proof of the lemma. □