Key Clustering in Substitution-Permutation Network Cryptosystems†

Draft

H. M. Heys and S. E. Tavares

Department of Electrical Engineering
Queen's University
Kingston, Ontario, Canada

December 7, 1993

†Support for this research was provided by the Natural Sciences and Engineering Research Council and by the Telecommunications Research Institute of Ontario.

Key Clustering in Substitution-Permutation Network Cryptosystems

H. M. Heys and S. E. Tavares
Department of Electrical and Computer Engineering
Queen's University
Kingston, Ontario, Canada K7L 3N6
email: tavares@ee.queensu.ca

Abstract — In this paper we examine the key clustering characteristics of a class of block cryptosystems referred to as substitution-permutation networks or SPNs. Specifically, we investigate the relationship between the property of key avalanche and the success of a key clustering attack. Further, we develop an analytical model of the key avalanche property and use this to estimate a lower bound on the complexity of a key clustering attack as a function of the number of rounds of substitutions.

I. Introduction

Substitution-permutation networks (SPNs) evolved from the work of Shannon [1] and Feistel [2] and form the foundation for many modern private key block cryptosystems such as DES [3], FEAL [4], and LOKI [5]. Such cryptosystems belong to the class of product ciphers which obtain their cryptographic strength by iterating a cryptographic operation several times. The basic SPN consists of a number of rounds of nonlinear subsitutions connected by bit position permutations. The substitutions are performed by dividing the block of bits into small sub-blocks and using a mapping stored as a table lookup and referred to as an S-box. It has been shown that this basic SPN structure can be used to construct ciphers which possess good cryptographic properties such as completeness [6] and resistance to differential and linear cryptanalysis [7][8].

We shall consider a general N-bit SPN as consisting of R rounds of $n \times n$ S-boxes. The plaintext and ciphertext are N-bit vectors denoted as $\mathbf{P} = [P_1\ P_2\ ...\ P_N]$ and $\mathbf{C} = [C_1\ C_2\ ...\ C_N]$, respectively. An S-box in the network is defined as an n-bit bijective mapping $S\ :\ \mathbf{X} \rightarrow \mathbf{Y}$ where $\mathbf{X} = [X_1\ X_2\ ...\ X_n]$ and $\mathbf{Y} = [Y_1\ Y_2\ ...\ Y_n]$. We shall assume that an S-box is keyed by XORing n bits of key with the S-box input vector, $\mathbf{X}$, before the substitution operation is performed. Hence, the network is keyed from a τ-bit key $\mathbf{K} = [K_1\ K_2\ ...\ K_\tau]$ by XORing N bits of the key with the network bits before each round of substitutions. The method for determining where each key bit is applied in the network is referred to as the key scheduling algorithm. Decryption is performed by running the data "backwards" through the network (i.e., applying the key scheduling algorithm in reverse and using the inverse S-boxes). A simple example of an SPN cryptosystem with $N = 16$, $n = 4$, and $R = 4$ is illustrated in Figure 1.

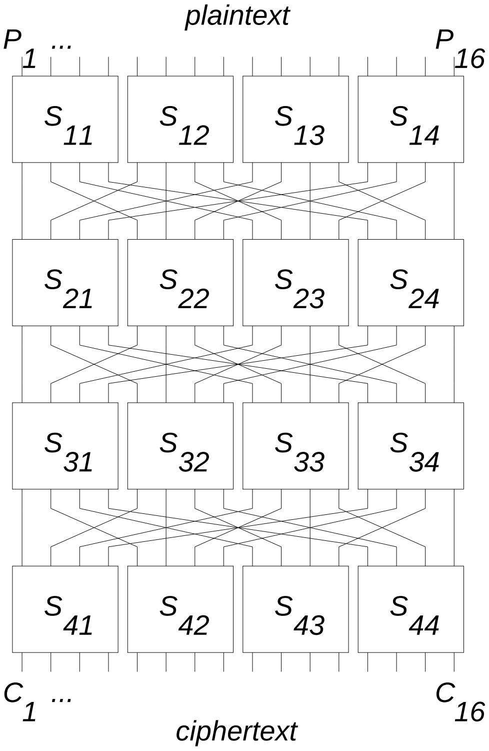

Figure 1. Simple SPN with $N = 16$, $n = 4$, and $R = 4$

II. Key Clustering Attacks

A block cryptosystem is considered weak if keys which are close to each other in Hamming distance result in a number of corresponding ciphertexts which are also close in distance. For example, consider two keys, $\mathbf{K}'$ and $\mathbf{K}''$, for which $wt(\Delta\mathbf{K} = \mathbf{K}' \oplus \mathbf{K}'')$ is small where $wt(\cdot)$ represents the Hamming weight of the specified argument, $\oplus$ is the bit-wise XOR operator, and Δ is used to indicate the XOR difference of the specfied vector. The encryption of l plaintexts, $\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_l$, under the two different keys results in l ciphertext pairs, $(\mathbf{C}'_1, \mathbf{C}''_1), (\mathbf{C}'_2, \mathbf{C}''_2), ..., (\mathbf{C}'_l, \mathbf{C}''_l)$. If there are a number of ciphertexts that are close in distance due to the proximity of the two keys — for example, if $E\{wt(\Delta\mathbf{C} = \mathbf{C}' \oplus \mathbf{C}'')\}$ is small where $E\{\cdot\}$ is the expectation operator — then we refer to the cryptosystem as having key clustering.

Key clustering can be exploited by a cryptanalyst to improve upon an exhaustive key search. Such an attack requires an appropriate number of known plaintexts to be able to determine whether a key is close to the correct key. The cryptanalyst proceeds by randomly selecting and testing keys until a key is found that is in the neighborhood of the correct key. Once such a key is found, the cryptanalyst can test all keys within a suitable distance of this key until the correct key is established. As an example, consider a cryptosystem with a key of 64–bits and which has the property that keys within distance 5 of the actual key (about 2^{23} keys) result in ciphertexts that are (on average) close enough to the actual ciphertexts to be distinguishable using only about 1000 known plaintext-ciphertext pairs. By randomly executing about $2^{64}/2^{23} = 2^{41}$ trials of

1000 encryptions we expect to be able to discover a key in the neighorhood (i.e., within distance 5) of the actual key. Testing all keys in the neighborhood of this experimental key (about 2^{23} encryptions) will reveal the correct key. As a result, the complexity of the key clustering attack is approximately $(1000)2^{41} + 2^{23} \approx 2^{51}$ which is a significant improvement on the complexity of about 2^{64} required for an exhaustive key search.

III. Key Avalanche Property

The relationship between the key avalanche property of a cryptosystem and key clustering attacks was noted in [9]. In this section we develop a model for the key avalanche property of an SPN as a function of the number of rounds of substitutions.

(a) Definitions

Consider the following definition of key avalanche.

Definition 1: A cryptosystem is said to satisfy the *key avalanche criterion* if each ciphertext bit changes with a probability of 1/2 when a single key bit is changed.

This definition is analagous to the definition of the strict avalanche criterion (SAC) for a cryptosystem [10] which refers to the probability of a ciphertext bit change given a one bit plaintext change. The key avalanche criterion, of course, refers to key rather than plaintext changes. As well, we can extend Definition 1 to consider the effect of changes involving more than one key bit.

Definition 2: A cryptosystem is said to satisfy the *extended key avalanche criterion order* κ if each ciphertext bit changes with a probability of 1/2 when a set of κ key bits are changed.

Let the *key avalanche probability*, p_{ka}, represent the probability that a particular ciphertext bit changes given a particular set of κ key bit changes. Ideally, we desire a cryptosystem to exactly satisfy the extended key avalanche criterion and $p_{ka} = 1/2$ for any ciphertext bit and set of κ key bits. In reality, cryptosystems will likely only approximately satisfy the criterion and the key avalanche probability can be represented as

$$p_{ka} = (1/2) - \epsilon. \tag{1}$$

We refer to ϵ as the *key avalanche error* and note that the value of $|\epsilon|$ is typically very small. A key clustering attack may be mounted against a cryptosystem which has poor extended key avalanche characteristics (i.e., a large value for $|\epsilon|$). The complexity of the key clustering attack is essentially a product of the number of key trials required to find a key in the neighborhood of the correct key and the number of known plaintexts required to determine that a test key is in the neighborhood. Cryptosystems with large values of κ and large corresponding values of $|\epsilon|$ require few key tests before finding a key close to the actual key and few known plaintexts to determine that a tested key is in the neighborhood of the actual key and, hence, are susceptible to key clustering attacks.

(b) Network Model Assumptions

An analytical model for the strict avalanche characteristics of SPNs is presented in [11]. We now extend the methodology to develop a model of the key avalanche characteristics for a one bit key change, i.e., $\kappa = 1$. The model approximates each S-box in the network as a stochastic mapping and calculates the key avalanche probability for each round recursively assuming a one bit key change.

We shall assume that the cryptosystem of interest is an N-bit block cryptosystem constructed using $n \times n$ S-boxes such that $N = n^2$. For example, a 64–bit SPN constructed using 8×8 S-boxes is a practical cryptosystem that satisfies these constraints. The network of Figure 1 is also a simple illustration of such a network with $N = 16$ and $n = 4$.

In the model, each S-box behaves as a random variable selected uniformly from the set of possible bijective mappings. Hence, an input change to an S-box results in a number of output changes represented by the random variable $D = wt(\Delta\mathbf{Y})$ where all possible values of $\Delta\mathbf{Y}$ belonging to the set of $2^n - 1$ non-zero changes are equally likely. Therefore, the probability distribution of D is given by

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

and

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

for $1 \leq d \leq n$. For ease of notation, we will represent $P_D(D = d)$ by $P_D(d)$.

We assume that the network uses a simple, effective permutation defined by bit i of the output of round r being connected to bit j of the input of round $r + 1$ such that

$$j = n \cdot ((i-1) \;\; mod \;\; n) + \lfloor (i-1)/n \rfloor + 1. \tag{4}$$

This permutation belongs to the class of permutations identified Kam and Davida [6] as providing provable completeness in networks for which $N = n^R$. Ayoub [12] further identified this permutation as belonging to a class of permutations cryptographically equivalent to Kam and Davida's structure.[1]

The cryptosystem that we shall consider in the model is keyed using $\tau = N$ key bits which are all applied at each round by XORing with the S-box input bits. We assume that the ordering of the key bits at the input to each round can be represented as a random variable. Hence, the model determines probabilities by averaging over all possible placements of the one bit key change in each round.

[1] The class of permutations identified by Ayoub are particularly useful for networks of an arbitrary number of rounds because they are optimal (in the sense that they provide completeness in the fewest number of contiguous rounds) and are easy to implement since they allow the use of the same permutation for each round.

(c) Computation of Key Avalanche Property

The recursive model is based on finding the probability distribution of the number of S-boxes in a round which have changes at their outputs given the probability distribution for the previous round when a one bit key change is applied. From the probability distribution of the number of S-boxes with output changes, it is possible to derive, under the assumptions of the model, a probability distribution of the number of bit changes at the output of a particular round. From this, the expected number of bit changes and, hence, the key avalanche probability are easily determined.

Let W_r represent the random variable corresponding to the number of bit changes (caused by the complementation of one key bit) after round r, i.e.,

$$W_r = \sum_{s=1}^{n} wt(\Delta\mathbf{Y}_{rs}) \tag{5}$$

where $\Delta\mathbf{Y}_{rs}$ is the output change of an S-box numbered s, $1 \le s \le n$, in round r, $1 \le r \le R$. Assuming symmetry in the location of a key bit change and the resulting output bit changes, the key avalanche probability after r rounds corresponding to any ciphertext bit and key bit change is given by the expected value of the number of bit changes divided by the block size, i.e., $p_{ka} = E\{W_r/N\}$. Therefore, we are interested in determining the probability distribution $P(W_r)$. For notational convenience, we let $P(W_r = w_r) \equiv P(w_r)$.

Let L_r be a random variable representing the number of S-boxes in round r which have changes at their outputs, i.e., $L_r = \#\{s \mid wt(\Delta\mathbf{Y}_{rs}) \ne 0\}$. The probability distribution of bit changes for the output of round r may be determined from

$$P(w_r) = \sum_{l_r=0}^{n} P(w_r \mid l_r) \cdot P(l_r) \tag{6}$$

where the variable l_r represents a particular value of the random variable L_r. Since the single key bit change must cause a change in the output of one, and only one, of the first round S-boxes, we can write the first round probability distribution of L_r as

$$P(L_1 = l_1) = \begin{cases} 1 & , l_1 = 1 \\ 0 & , l_1 \ne 1. \end{cases} \tag{7}$$

Consequently, the probability of l_{r+1} S-boxes in round $r+1$ with output changes may be determined recursively using

$$P(l_{r+1}) = \sum_{l_r=0}^{n} P(l_{r+1} \mid l_r) \cdot P(l_r). \tag{8}$$

In order to determine $P(W_r)$ and, subsequently, the expected number of bit changes, we must therefore derive expressions for $P(w_r \mid l_r)$ and $P(l_{r+1} \mid l_r)$.

Consider first the determination of $P(w_r \mid l_r)$. Let $\mathbf{d} = [d_1\ d_2\ ...\ d_{l_r}]$ where $d_i \in \{1, ..., n\}$ is the number of output changes, $wt(\Delta\mathbf{Y})$, of the i-th S-box of round r that has an output change. Define

$$\Lambda = \left\{ \mathbf{d} \;\middle|\; \sum_{i=1}^{l_r} d_i = w_r \right\} \tag{9}$$

to represent the values of $\mathbf{d}$ for which there are a total of w_r bit changes at the output of round r. The probability of w_r output bit changes given that l_r S-boxes have output bit changes is given by

$$P(w_r \mid l_r) = \sum_{\mathbf{d} \in \Lambda} P(\mathbf{d}) \tag{10}$$

where

$$P(\mathbf{d}) = \prod_{i=1}^{l_r} P_D(d_i). \tag{11}$$

Consider now the determination of $P(l_{r+1} \mid l_r)$. Let t be the number of S-boxes in round $r + 1$ that do not have any input changes and let b be the number of S-boxes in round $r + 1$ that have input changes of one bit only, i.e., $t = \#\{s \mid wt(\Delta\mathbf{X}_{(r+1)s}) = 0\}$ and $b = \#\{s \mid wt(\Delta\mathbf{X}_{(r+1)s}) = 1\}$. In order to determine $P(l_{r+1} \mid l_r)$, we initially consider the joint probability of b and t given l_r S-boxes with output changes in round r, $P(b, t \mid l_r)$.

Define the probability $P(\delta \mid \mathbf{d})$ to be the probability that at least δ particular S-boxes in round $r + 1$ are not affected by input changes given that a specific $\mathbf{d}$ occurs. Further, let $P(\rho \mid t, \mathbf{d})$ represent the probability that at least ρ particular S-boxes have only one input bit changing given that t S-boxes do not have any input changes and a specific $\mathbf{d}$ occurs. Letting $\mathrm{T} = \{1, ..., n\}^{l_r}$, the probability of interest is given by

$$\begin{aligned} P(b, t \mid l_r) &= \sum_{\mathbf{d} \in \mathrm{T}} P(b, t, \mathbf{d}) \\ &= \sum_{\mathbf{d} \in \mathrm{T}} P(b \mid t, \mathbf{d}) \cdot P(t \mid \mathbf{d}) \cdot P(\mathbf{d}) \\ &= \sum_{\mathbf{d} \in \mathrm{T}} \sum_{\rho=b}^{n-t} (-1)^{\rho-b} \binom{\rho}{b} \binom{n-t}{\rho} P(\rho \mid t, \mathbf{d}) \\ &\qquad \cdot \sum_{\delta=t}^{n} (-1)^{\delta-t} \binom{\delta}{t} \binom{n}{\delta} P(\delta \mid \mathbf{d}) \cdot P(\mathbf{d}). \end{aligned} \tag{12}$$

Equation (12) is derived by considering the application of the extension of the inclusion-exclusion principle [13, p.271] in order to determine $P(b \mid t, \mathbf{d})$ and $P(t \mid \mathbf{d})$.

The probability $P(\mathbf{d})$ is determined as in equation (11) and from [11], we have

$$P(\delta \mid \mathbf{d}) = \prod_{i=1}^{l_r} \frac{\binom{n-\delta}{d_i}}{\binom{n}{d_i}}. \tag{13}$$

We shall determine the probability $P(\rho \mid t, \mathbf{d})$ in the following manner. Without loss of generality, consider that the last t S-boxes in round $r+1$ are the S-boxes which do not have any input changes. Let N_t represent the number of arrangements for the output bit changes of round r satisfying $\mathbf{d}$ such that the last t S-boxes in round $r+1$ have no input changes and the remaining $n-t$ S-boxes each have one or more input bit changes. Hence, N_t can be determined by computing the number of arrangements with exactly zero of the remaining $n-t$ S-boxes having no input changes. Using the inclusion-exclusion principle this is given by

$$N_t = \sum_{\mu=0}^{n-t} (-1)^{\mu} \binom{n-t}{\mu} \prod_{i=1}^{l_r} \binom{n-t-\mu}{d_i}. \tag{14}$$

Further, assume that the first ρ S-boxes in round $r+1$ have only one input change. Define the vector $\mathbf{h} = [h_1 \; h_2 \; ... \; h_{l_r}]$ where $h_i = \{0, ..., \rho\}$ represents the number of outputs of the i-th S-box in round r with output changes which provide an input change to the first ρ S-boxes. Hence, $\mathbf{h} \in \mathrm{H}$ where

$$\mathrm{H} = \left\{ \mathbf{h} \mid \sum_{i=1}^{l_r} h_i = \rho \right\}. \tag{15}$$

Let N_ρ represent the number of arrangements of the S-boxes in round r which originate input changes to the first ρ S-boxes. Hence

$$N_\rho = \rho! / \left[\prod_{i=1}^{l_r} h_i! \right]. \tag{16}$$

Lastly, define N_γ as the number of valid arrangements of bit changes such that the remaining $n-t-\rho$ S-boxes in round $r+1$ have one or more bit changes at their input. Once again applying the inclusion-exclusion principle, this is given by

$$N_\gamma = \sum_{\mu=0}^{n-t-\rho} (-1)^{\mu} \binom{n-t-\rho}{\mu} \prod_{i=1}^{l_r} \binom{n-t-\rho-\mu}{d_i - h_i}. \tag{17}$$

The probability $P(\rho \mid t, \mathbf{d})$ can now be computed as

$$P(\rho \mid t, \mathbf{d}) = \sum_{\mathbf{h} \in \mathrm{H}} N_\rho N_\gamma / N_t. \tag{18}$$

Using the probability $P(b, t \mid l_r)$ it is possible to compute $P(l_{r+1} \mid l_r)$ by determining the expected result given that a key bit change is randomly XORed to one input bit of round $r+1$. If we ignore the effect of the key bit change, the number of S-boxes with output changes is simply given as $l_{r+1} = n - t$. However, in determining the effect of the key bit change we must consider the following three cases, their probability of occurrence, and their resulting implications:

1. Key change XORed with bit from S-box with no input changes (probability = t/n) $\Rightarrow$ $l_{r+1} = n - t + 1$.
2. Key change XORed with bit from S-box with one input change (probability = b/n) $\Rightarrow$ $l_{r+1} = n - t - 1$ with probability $1/n$, or $l_{r+1} = n - t$ with probability $(n-1)/n$.
3. Key change XORed with bit from S-box with more than one input change (probability = $1 - t/n - b/n$) $\Rightarrow$ $l_{r+1} = n - t$.

Hence, given $P(b, t \mid l_r)$, we can compute $P(l_{r+1} \mid l_r)$ by:

$$\begin{aligned}
P(l_{r+1} = n - t + 1 \mid l_r) &= \sum_{b=0}^{n-t} \frac{t}{n} \cdot P(b, t \mid l_r) \\
P(l_{r+1} = n - t - 1 \mid l_r) &= \sum_{b=0}^{n-t} \frac{b}{N} \cdot P(b, t \mid l_r) \\
P(l_{r+1} = n - t \mid l_r) &= \sum_{b=0}^{n-t} \left(1 - \frac{t}{n} - \frac{b}{N}\right) \cdot P(b, t \mid l_r).
\end{aligned} \tag{19}$$

Using this analysis, we have estimated the key avalanche probability for a 64–bit SPN and, subsequently, determined the key avalanche error ϵ as a function of the number of rounds. The results are listed in the second column of Table 1.

IV. Methods of Determining Close Keys

Using the key avalanche model presented in the previous section, we can now determine the security of an SPN against exploitation of weak key avalanche for a key clustering attack as a function of the number of rounds of substitutions. In this section we examine two methods for determining that an experimental key is close to the actual key. Specifically, for each method we determine the number of known plaintexts, $\mathcal{N}_P$, required to reveal that two keys are close to each other.

In developing a lower bound on the complexity of the key clustering attack, we only consider the number of known plaintexts required to determine if an experimental key is close to the actual key; we do not consider how many trials are required before we expect to pick a close experimental key. Hence, although $\mathcal{N}_P$ gives an approximate lower bound on the complexity of the key clustering attack, in practice, the complexity of the attack will be much higher than $\mathcal{N}_P$ since it will typically take a large number of trials before a selected key is close to the actual key. In the approach we make the reasonable assumption that the magnitude of the key avalanche error $|\epsilon|$ is maximized for $\kappa = 1$.

(a) Ciphertext Correlation

In a cryptosystem with weak key avalanche, an obvious method for determining whether an experimental key is close to the actual key is to search for correlation in the ciphertext output bits. If there is a high enough degree of correlation it is likely that the experimental key is a small Hamming distance from the actual key.

The problem may be considered to be a hypothesis testing problem with one hypothesis, H_0, being that the test key comes from the neighborhood (i.e., for $\kappa = 1$, within one bit) of the correct key and the other hypothesis, H_1, being that the key is not in the neighborhood of the correct key. Let ϵ_R represent the value of the key avalanche error for $\kappa = 1$ after R rounds of substitutions. Assume that the probability that a ciphertext bit changes under hypothesis H_0 or H_1 is given by $p_0 = 1/2 - \epsilon_R$ or $p_1 = 1/2$, respectively[2]. Let η represent the number of samples of ciphertext bit changes required to test a key and, hence, the number of known plaintexts required to test a key is given by $\mathcal{N}_P = \eta/N$. The number of bit changes in η ciphertext bit change samples follows the binomial distribution for each hypothesis. Therefore the expected number of bit changes and variances are given by

$$\begin{aligned} H_0: \quad & \mu_0 = \eta/2 - \eta\epsilon_R, \quad \sigma_0^2 = \eta\left(1/4 - \epsilon_R^2\right) \\ H_1: \quad & \mu_1 = \eta/2, \qquad\qquad \sigma_1^2 = \eta/4. \end{aligned} \tag{20}$$

Since ϵ_R is typically very small, $\epsilon_R^2 \ll 1/4$ and $\sigma_0^2 \approx \sigma_1^2 = \eta/4$. Therefore, let $\sigma^2 = \eta/4$ represent the variance of both hypothesis distributions.

Since η is typically large, the binomial distribution for each hypothesis may be approximated as a Gaussian distribution with the means, μ_0 and μ_1, and variance σ^2. For convenience, we shall assume that the acceptable probability of error in selecting a hypothesis is the same for both H_0 and H_1. Hence, considering the symmetry of the hypotheses, we require η large enough so that

$$\mu_0 + \alpha\sigma \approx \mu_1 - \alpha\sigma \tag{21}$$

with the significance level α selected to provide a suitably small probability of error in the hypothesis test where the probability of error is given by

$$Q(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\infty} e^{-x^2/2} dx. \tag{22}$$

Hence, $\eta\epsilon_R - \sqrt{\eta}\alpha \approx 0$ and, consequently, $\eta \approx (\alpha/\epsilon_R)^2$.

For an R round SPN, the number of known plaintexts required to test a key is, therefore,

$$\mathcal{N}_P \approx \frac{\alpha^2}{N\epsilon_R^2} \tag{23}$$

where α is selected to provide a suitably small probability of error in the hypothesis test.

[2] Note that $p_1 = 1/2$ implies that $\epsilon = 0$ for $\kappa \neq 1$. In practice, the key avalanche error ϵ for different values of $\kappa \neq 1$ would not necessarily be exactly zero. However, since the assumption results in a lower bound on the security analysis, it is therefore suitable for our purposes.

(b) Meet-in-the-Middle Correlation

Similarly to the ciphertext correlation approach we may consider identifying close keys by using an experimental key to encrypt the known plaintexts for the first $R/2$ rounds and to decrypt the known ciphertexts for the last $R/2$ rounds. This generates two values for a middle block of N bits which can be checked for correlation. Let $\epsilon_{R/2}$ represent the value of ϵ for $R/2$ rounds. If $\epsilon_{R/2}$ is large, then the two sets of middle bits are significantly correlated to the actual bits and, therefore, are highly likely to be the same. We refer to the correlation of the two sets of middle bits as meet-in-the-middle correlation. Note that it is not necessary or possible for the cryptanalyst to know the actual middle bits.

Let $p_{R/2} = 1/2 - \epsilon_{R/2}$ represent the probability that a middle bit is different than the actual middle bit given that the experimental key selected is within distance one of the actual key. Assume that the key avalanche probability is the same backwards and forwards. The probability that two experimental middle bits are the same, p_M, is given by the probability that both bits are correct or that both bits are incorrect. That is,

$$\begin{aligned} p_M &= \left(p_{R/2}\right)^2 + \left(1 - p_{R/2}\right)^2 \\ &= 1/2 + 2\epsilon_{R/2}^2. \end{aligned} \tag{24}$$

As before, we define hypothesis H_0 to be that an experimental key is close to the actual key and hypothesis H_1 to be that it is not. The expected number of ciphertext bit changes and variances for each hypothesis are

$$\begin{aligned} &H_0 : \quad \mu_0 = \eta/2 - 2\eta\epsilon_{R/2}^2, \quad \sigma_0^2 = \eta\left(1/4 - 4\epsilon_{R/2}^4\right) \\ &H_1 : \quad \mu_1 = \eta/2, \qquad\qquad\quad \sigma_1^2 = \eta/4 \end{aligned} \tag{25}$$

Using an analysis similar to the previous case of ciphertext correlation, for an R round SPN, we may determine the number of known plaintexts required to test a key to be

$$\mathcal{N}_P \approx \frac{\alpha^2}{4N\epsilon_{R/2}^4} \tag{26}$$

where α is selected to provide a suitably small probability of error in the hypothesis test.

V. Results

Clearly the advantage of using one form of the attack over the other depends on the relative values of ϵ_R and $\epsilon_{R/2}$. For an SPN with $N = 64$ and a key size of $\tau = 64$, using the values of the key avalanche error determined by the model of Section III and presented in the second column of Table 1, we have calculated the number of known plaintexts required in order to test a key to determine whether it is within distance one of the actual key as a function of the number of substitution rounds. The results for both methods of determining close keys are presented

Rounds R	Key Avalanche Error ϵ	Ciphertext Correlation $\mathcal{N}_P$	Meet-in-Middle Correlation $\mathcal{N}_P$
1	4.37×10^{-1}	5	-
2	2.21×10^{-1}	20	7
3	2.98×10^{-2}	1124	-
4	1.59×10^{-3}	3.98×10^{5}	105
5	6.75×10^{-5}	2.20×10^{8}	-
6	2.55×10^{-6}	1.54×10^{11}	3.16×10^{5}
7	9.09×10^{-8}	1.21×10^{14}	-
8	3.12×10^{-9}	1.03×10^{17}	3.96×10^{10}
9	1.05×10^{-10}	9.16×10^{19}	-
10	3.45×10^{-12}	8.42×10^{22}	1.21×10^{16}

Table 1. Results for SPN with $N = 64$, $n = 8$, and $\alpha = 8$

in columns 3 and 4 of Table 1. The significance level for the hypothesis test was selected to be $\alpha = 8$. Note that it can be shown [14, p.569] that $Q(\alpha) \leq e^{-\alpha^2/2}/2$ and, hence, although increasing the value of α does not significantly change the value of $\mathcal{N}_P$, it does significantly decrease the likelihood of an error in the hypothesis test.

From the table we can determine the number of rounds required by an SPN in order to provide a level of security against key clustering equivalent to exhaustive key search. The results suggest that the meet-in-the-middle approach requires fewer known plaintexts to identify a close key. For both methods, when $R = 10$, the number of plaintexts required to test a key satisfies $\mathcal{N}_P \gtrsim 2^{53}$. Combining this bound on $\mathcal{N}_P$ with the number of trials required to select a key close to the actual key results in a complexity much greater than the 2^{64} key trials required in exhaustive key search. We conclude that a 64–bit 10–round SPN with a 64–bit key and 8×8 S-boxes is expected to be unbreakable using a key clustering attack exploiting a key avalanche weakness. Further, since the complexity of the attack is likely to be far greater than $\mathcal{N}_P$, our analysis suggests that, in practice, an 8–round SPN with $\mathcal{N}_P \gtrsim 2^{35}$ will have adequate resistance to key clustering.

VI. Conclusion

We have presented an analysis of the relationship between the key avalanche property and key clustering. Using a stochastic model of the key avalanche property we are able to determine the minimum number of rounds required for an SPN to ensure that a key clustering attack, exploiting weak key avalanche, will fail.

References

[1] C. E. Shannon, “Communication theory of secrecy systems,” *Bell System Technical Journal*, vol. 28, pp. 656–715, 1949.

[2] H. Feistel, “Cryptography and computer privacy,” *Scientific American*, vol. 228, no. 5, pp. 15–23, 1973.

[3] National_Bureau_of_Standards, “Data Encryption Standard (DES),” *Federal Information Processing Standard Publication 46*, 1977.

[4] A. Shimizu and S. Miyaguchi, “Fast data encipherment algorithm: FEAL,” *Advances in Cryptology: Proceedings of EUROCRYPT ’87*, Springer-Verlag, Berlin, pp. 267–278, 1988.

[5] L. Brown, J. Pieprzyk, and J. Seberry, “LOKI - a cryptographic primitive for authentication and secrecy applications,” *Advances in Cryptology: Proceedings of AUSCRYPT ’90*, Springer-Verlag, Berlin, pp. 229–236, 1990.

[6] J. B. Kam and G. I. Davida, “A structured design of substitution-permutation encryption networks,” *IEEE Transactions on Computers*, vol. 28, no. 10, pp. 747–753, 1979.

[7] L. J. O’Connor, “On the distribution of characteristics in bijective mappings,” *Advances in Cryptology: Proceedings of EUROCRYPT ’93*, Springer-Verlag, Berlin, pp. 360–370, 1994.

[8] H. M. Heys and S. E. Tavares, “The design of product ciphers resistant to differential and linear cryptanalysis,” *presented at CRYPTO ’93*, Santa Barbara, Calif., Aug. 1993.

[9] W. Diffie and M. E. Hellman, “Privacy and authentication: An introduction to cryptography,” *Proceedings of the IEEE*, vol. 67, no. 3, pp. 397–427, 1979.

[10]A. F. Webster and S. E. Tavares, “On the design of S-boxes,” *Advances in Cryptology: Proceedings of CRYPTO ’85*, Springer-Verlag, Berlin, pp. 523–534, 1986.

[11]H. M. Heys and S. E. Tavares, “Avalanche characteristics of a class of product ciphers,” tech. rep., Department of Electrical Engineering, Queen’s University, Aug. 30, 1993.

[12]F. Ayoub, “The design of complete encryption networks using cryptographically equivalent permutations,” *Computers and Security*, vol. 2, pp. 261–267, 1982.

[13]F. S. Roberts, *Applied Combinatorics*. Englewood Cliffs, N.J.: Prentice-Hall, 1984.

[14]S. S. Haykin, *Digital Communications*. New York: Wiley, 1988.