# A Tutorial on

# Linear and Differential Cryptanalysis

by

Howard M. Heys

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

**Abstract:** In this paper, we present a detailed tutorial on linear cryptanalysis and differential cryptanalysis, the two most significant attacks applicable to symmetric-key block ciphers. The intent of the paper is to present a lucid explanation of the attacks, detailing the practical application of the attacks to a cipher in a simple, conceptually revealing manner for the novice cryptanalyst. The tutorial is based on the analysis of a simple, yet realistically structured, basic Substitution-Permutation Network cipher. Understanding the attacks as they apply to this structure is useful, as the Rijndael cipher, recently selected for the Advanced Encryption Standard (AES), has been derived from the basic SPN architecture. As well, experimental data from the attacks is presented as confirmation of the applicability of the concepts as outlined.

## 1. Introduction

In this paper, we present a tutorial on two powerful cryptanalysis techniques applied to symmetric-key block ciphers: linear cryptanalysis [1] and differential cryptanalysis [2]. Linear cryptanalysis was introduced by Matsui at EUROCRYPT '93 as a theoretical attack on the Data Encryption Standard (DES) [3] and later successfully used in the practical cryptanalysis of DES [4]; differential cryptanalysis was first presented by Biham and Shamir at CRYPTO '90 to attack DES and eventually the details of the attack were packaged as a book [5]. Although the early target of both attacks was DES, the wide applicability of both attacks to numerous other block ciphers has solidified the pre-eminence of both cryptanalysis techniques in the consideration of the security of all block ciphers. For example, many of the candidates submitted for the recent Advanced Encryption Standard process undertaken by the National Institute of Standards and Technology [6] were designed using techniques specifically targeted at thwarting linear and differential cryptanalysis. This is evident, for example, in the Rijndael cipher [7], the encryption algorithm selected to be the new standard. The concepts discussed in this paper could be used to form an initial understanding required to comprehend the design principles and security analysis of the Rijndael cipher, as well as many other ciphers proposed in recent years.

The paper is structured as a tutorial and, as such, is intended to not be rigorously mathematical. It introduces the basic concepts of linear and differential cryptanalysis but is by no means a definitive source for understanding all the many refinements and improvements of the attacks over the years. The basic purpose of the paper is to use a simple (yet somewhat realistic) cipher structure to study the most basic concepts of the two attacks. Other more formal discussions exist on the topic. For example, overviews of the attacks as applied to Substitution-Permutation Networks (the cipher structured to be considered in this paper) are presented in [8] and [9]. For a general introduction to block ciphers and their analysis, see [10].

The need for a tutorial on the attacks arises from the very difficult nature of both attacks and the lack of *simplified*, yet detailed, reference material describing the attacks. Conventional cryptographic references and texts [11][12][13][14] generally present material on block ciphers in a very descriptive manner, with little detail illustrating the concepts of the attacks. Consequently, most published material detailing the attacks has a research focus and gives little intuition and explanation for the non-expert. When the basic concepts of the attack are described in the literature (as in Matsui's and Biham and Shamir's original papers), they are typically presented in reference to DES which is, in nature, somewhat convoluted in a manner which interferes with the understanding the cryptanalytic concepts.

## 2. A Basic Substitution-Permutation Network Cipher

The cipher that we shall use to present the concepts is a basic Substitution-Permutation Network (SPN). We will focus our discussion on a cipher, illustrated in Figure 1, that takes a 16-bit input block and processes the block by repeating the basic operations of a round four times. Each round consists of (1) substitution, (2) a transposition of the bits (i.e., permutation of the bit positions), and (3) key mixing. This basic structure was presented by Feistel back in 1973 [15] and these basic operations are similar to what is found in DES and many other modern ciphers, including Rijndael. So although, we are considering a somewhat simplified structure, an analysis of the attack of such a cipher presents valuable insight into the security of larger, more practical constructions.

### *2.1 Substitution*

In our cipher, we break the 16-bit data block into four 4-bit sub-blocks. Each sub-block forms an input to a 4×4 S-box (a substitution with 4 input and 4 output bits), which can be easily implemented with a table lookup of sixteen 4-bit values, indexed by the integer represented by the 4 input bits. The most fundamental property of an S-box is that it is a nonlinear mapping, i.e., the output bits cannot be represented as a linear operation on the input bits.

For our cipher, we shall use the same nonlinear mapping for all S-boxes. (In DES all the S-boxes in a round are different, while all rounds use the same set of S-boxes.) The attacks of linear and differential cryptanalysis apply equally to whether there is one mapping or all S-boxes are different mappings. The mapping chosen for our cipher, given in Table 1, is chosen from the S-boxes of DES. (It is the first row of the first S-box.) In the table, the most significant bit of the hexadecimal notation represents the leftmost bit of the S-box in Figure 1.

| input | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| output | E | 4 | D | 1 | 2 | F | B | 8 | 3 | A | 6 | C | 5 | 9 | 0 | 7 |

**Table 1.** S-box Representation (in hexadecimal)

### *2.2 Permutation*

The permutation portion of a round is simply the tranposition of the bits or the permutation of the bit positions. The permutation of Figure 1 is given in Table 2 (where the numbers represent bit positions in the block, with 1 being the leftmost bit and 16 being the rightmost bit) and can be simply described as: the output *i* of S-box *j* is connected to input *j* of S-box *i*. Note that there would be no purpose for a permutation in the last round and, hence, our cipher does not have one.

| input | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| output | 1 | 5 | 9 | 13 | 2 | 6 | 10 | 14 | 3 | 7 | 11 | 15 | 4 | 8 | 12 | 16 |

**Table 2.** Permutation

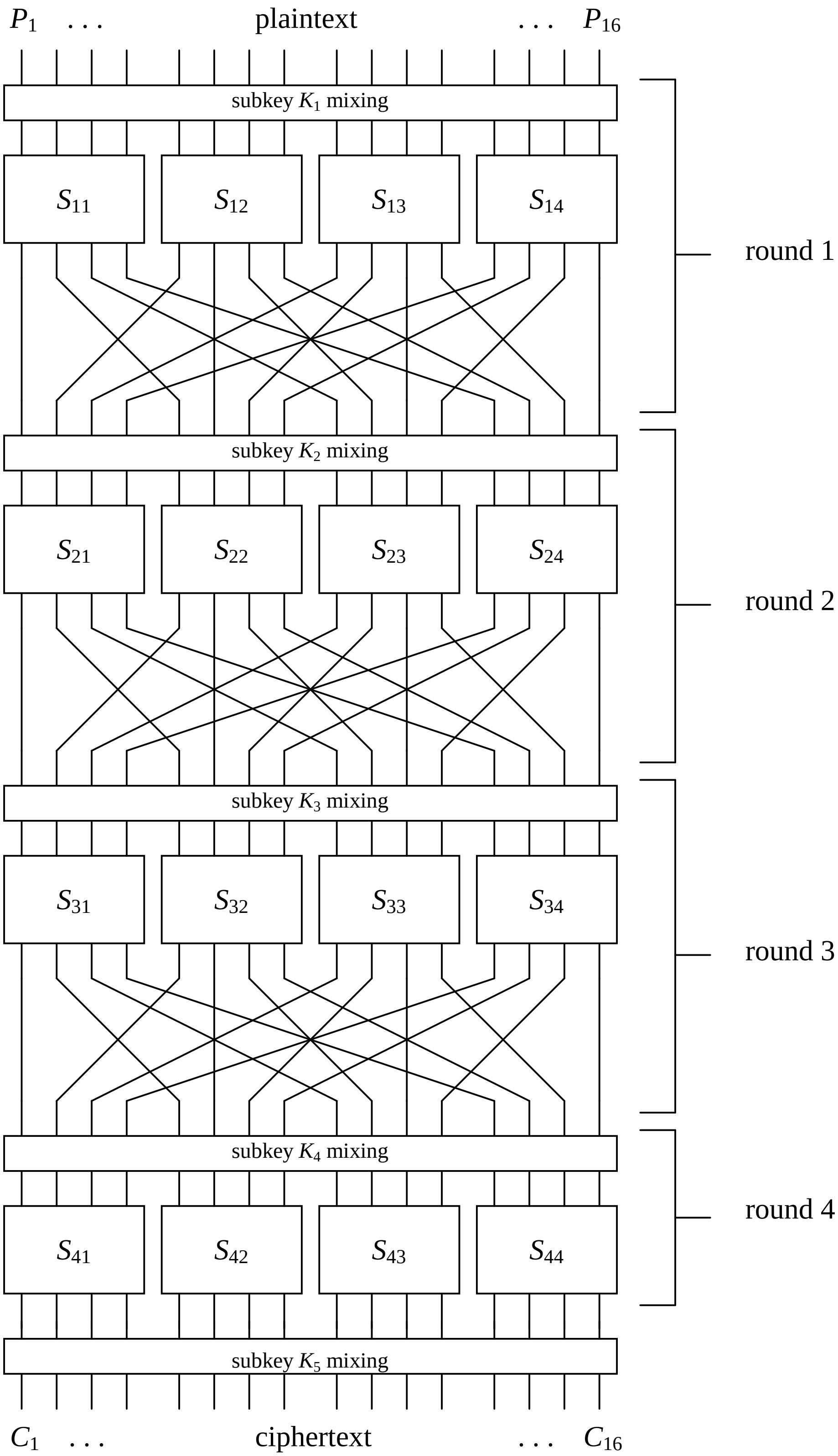


**Figure 1.** Basic Substitution-Permutation Network (SPN) Cipher

### *2.3 Key Mixing*

To achieve the key mixing, we use a simple bit-wise exclusive-OR between the key bits associated with a round (referred to as a subkey) and the data block input to a round. As well, a subkey is applied following the last round, ensuring that the last layer of substitution cannot be easily ignored by a cryptanalyst that simply works backward through the last round's substitution. Normally, in a cipher, the subkey for a round is derived from the cipher's master key through a process known as the key schedule. In our cipher, we shall assume that all bits of the subkeys are independently generated and unrelated.

### *2.4 Decryption*

In order to decrypt, data is essentially passed backwards through the network. Hence, decryption is also of the form of an SPN as illustrated in Figure 1. However, the mappings used in the S-boxes of the decryption network are the inverse of the mappings in the encryption network (i.e., input becomes output, output becomes input). This implies that in order for an SPN to allow for decryption, all S-boxes must be bijective, that is, a one-to-one mapping with the same number input and output bits. As well, in order for the network to properly decrypt, the subkeys are applied in reverse order and the bits of the subkeys must be moved around according to the permutation, if the SPN is to look similar to Figure 1. Note also that the lack of the permutation after the last round ensures that the decryption network can be the same structure as the encryption network. (If there was a permutation after the last substitution layer in the encryption, the decryption would require a permutation before the first layer of substitution.)

## 3. Linear Cryptanalysis

In this section, we outline the approach to attacking a cipher using linear cryptanalysis based on the example cipher of our basic SPN.

### *3.1 Overview of Basic Attack*

Linear cryptanalysis tries to take advantage of high probability occurrences of linear expressions involving plaintext bits, "ciphertext" bits (actually we shall use bits from the 2nd last round output), and subkey bits. It is a known plaintext attack: that is, it is premised on the attacker having information on a set of plaintexts and the corresponding ciphertexts. However, the attacker has no way to select which plaintexts (and corresponding ciphertexts) are available. In many applications and scenarios it is reasonable to assume that the attacker has knowledge of a random set of plaintexts and the corresponding ciphertexts.

The basic idea is to approximate the operation of a portion of the cipher with an expression that is linear where the linearity refers to a mod-2 bit-wise operation (i.e., exclusive-OR denoted by "⊕"). Such an expression is of the form:

$$X_{i_1} \oplus X_{i_2} \oplus \ldots \oplus X_{i_u} \oplus Y_{j_1} \oplus Y_{j_2} \oplus \ldots \oplus Y_{j_v} = 0 \qquad (1)$$

where $X_i$ represents the $i$-th bit of the input $X = [X_1, X_2, ...]$ and $Y_j$ represents the $j$-th bit of the output $Y = [Y_1, Y_2, ...]$. This equation is representing the exclusive-OR "sum" of $u$ input bits and $v$ output bits.

The approach in linear cryptanalysis is to determine expressions of the form above which have a high or low probability of occurrence. (No obvious linearity such as above should hold for all input and output values or the cipher would be trivially weak.) If a cipher displays a tendency for equation (1) to hold with high probability or not hold with high probability, this is evidence of the cipher's poor randomization abilities. Consider that if we randomly selected values for $u + v$ bits and placed them into the equation above, the probability that the expression would hold would be exactly 1/2. It is the deviation or bias from the probability of 1/2 for an expression to hold that is exploited in linear cryptanalysis: the further away that a linear expression is from holding with a probability of 1/2, the better the cryptanalyst is able to apply linear cryptanalysis. In the remainder of the paper, we refer to the amount by which the probability of a linear expression holding deviates from 1/2 as the *linear probability bias*. Hence, if the expression above holds with probability $p_L$ for randomly chosen plaintexts and the corresponding ciphertexts, then the probability bias is $p_L - 1/2$. The higher the magnitude of the probability bias, $|p_L - 1/2|$, the better the applicability of linear cryptanalysis with fewer known plaintexts required in the attack.

There are several ways to mount the attack of linear cryptanalysis. In this paper, we shall focus on what Matsui calls Algorithm 2 [1]. We investigate the construction of a linear approximation involving plaintext bits as represented by $X$ in (1) and the input to the last round of the cipher (or equivalently the output of the 2nd last round of the cipher) as represented by $Y$ in (1). The plaintext bits are random and consequently so are the input bits to the last round.

Equation (1) could be equivalently reformulated to have the right side being the sum of a number of subkey bits. However, in (1) as written with the right side of "0", the equation implicitly has subkey bits involved: these bits are fixed but unknown (as they are determined by the key under attack) and implicity absorbed into the "0" on the right side of equation (1) and the probability $p_L$ that the linear expression holds. If the sum of the involved subkey bits is "0", the bias of (1) will have the same sign (+ or –) as the bias of the expression involving the subkey sum and, if the sum of the involved subkey bits is "1", the bias of (1) will have the opposite sign.

Note that $p_L = 1$ implies that linear expression (1) is a perfect representation of the cipher behaviour and the cipher has a catastrophic weakness. If $p_L = 0$, then (1) represents an affine relationship in the cipher, also an indication of a catastrophic weakness. For mod-2 addition systems, an affine function is simply the complement of a linear function. Both linear and affine approximations, indicated by $p_L > 1/2$ and $p_L < 1/2$, respectively, are equally susceptible to linear cryptanalysis and we shall generally use the term linear to refer to both linear and affine relationships.

The natural question to ask is: How do we construct expressions which are highly linear and, hence, can be exploited? This is done by considering the properties of the cipher's only nonlinear component: the S-box. When the nonlinearity properties of the S-box are enumerated, it is possible to develop linear approximations between sets of input and output bits in the S-box. Consequently, it is possible to concatenate linear approximations of the S-boxes together so that intermediate bits (i.e., data bits from within the cipher) can be cancelled out and we are left with a linear expression which has a large bias and involves only plaintext and the last round input bits.

### *3.2 Piling-Up Principle*

Before we consider constructing a linear expression for the example cipher of this paper, we need some basic tools. Consider two random binary variables, $X_1$ and $X_2$. We begin by noting the simple relationships: $X_1 \oplus X_2 = 0$ is a linear expression and is equivalent to $X_1 = X_2$; $X_1 \oplus X_2 = 1$ is an affine expression and is equivalent to $X_1 \neq X_2$.

Now, assume that the probability distributions are given by

$$\Pr(X_1 = i) = \begin{cases} p_1 & , i = 0 \\ 1 - p_1 & , i = 1 \end{cases}$$

and

$$\Pr(X_2 = i) = \begin{cases} p_2 & , i = 0 \\ 1 - p_2 & , i = 1. \end{cases}$$

If the two random variables are independent, then

$$\Pr(X_1 = i, X_2 = j) = \begin{cases} p_1 p_2 & , i = 0, j = 0 \\ p_1(1-p_2) & , i = 0, j = 1 \\ (1-p_1)p_2 & , i = 1, j = 0 \\ (1-p_1)(1-p_2) & , i = 1, j = 1 \end{cases}$$

and it can be shown that

$$\begin{aligned} \Pr(X_1 \oplus X_2 = 0) &= \Pr(X_1 = X_2) \\ &= \Pr(X_1 = 0, X_2 = 0) + \Pr(X_1 = 1, X_2 = 1) \\ &= p_1 p_2 + (1-p_1)(1-p_2). \end{aligned}$$

Another perspective is to let $p_1 = 1/2+\varepsilon_1$ and $p_2 = 1/2+\varepsilon_2$, where $\varepsilon_1$ and $\varepsilon_2$ are the probability biases and $-1/2 \le \varepsilon_1,\varepsilon_2 \le +1/2$. Hence, it follows that

$$\Pr(X_1 \oplus X_2 = 0) = 1/2 + 2\varepsilon_1\varepsilon_2$$

and the bias $\varepsilon_{1,2}$ of $X_1 \oplus X_2 = 0$ is

$$\varepsilon_{1,2} = 2\varepsilon_1\varepsilon_2.$$

This can be extended to more than two random binary variables, $X_1$ to $X_n$, with probabilities $p_1 = 1/2+\varepsilon_1$ to $p_n = 1/2+\varepsilon_n$. The probability that $X_1 \oplus ... \oplus X_n = 0$ holds can be determined by the *Piling-Up Lemma* which assumes that all $n$ random binary variables are independent.

*Piling-Up Lemma* (Matsui [1])
For $n$ independent, random binary variables, $X_1, X_2, ...X_n$,

$$\Pr(X_1 \oplus ... \oplus X_n = 0) = 1/2 + 2^{n-1}\prod_{i=1}^{n} \varepsilon_i$$

or, equivalently,

$$\varepsilon_{1,2,..,n} = 2^{n-1}\prod_{i=1}^{n} \varepsilon_i$$

where $\varepsilon_{1,2,..,n}$ represents the bias of $X_1 \oplus ... \oplus X_n = 0$.

Note that if $p_i = 0$ or $1$ for all $i$, then $\Pr(X_1 \oplus ... \oplus X_n = 0) = 0$ or $1$. If only one $p_i = 1/2$, then $\Pr(X_1 \oplus ... \oplus X_n = 0) = 1/2$.

In developing the linear approximation of a cipher, the $X_i$ values will actually represent linear approximations of the S-boxes. For example, consider four independent random binary variables, $X_1$, $X_2$, $X_3$ and $X_4$. Let $\Pr(X_1 \oplus X_2 = 0) = 1/2 + \varepsilon_{1,2}$ and $\Pr(X_2 \oplus X_3 = 0) = 1/2 + \varepsilon_{2,3}$ and consider the sum $X_1 \oplus X_3$ to be derived by adding $X_1 \oplus X_2$ and $X_2 \oplus X_3$ together. Hence,

$$\Pr(X_1 \oplus X_3 = 0) = \Pr([X_1 \oplus X_2] \oplus [X_2 \oplus X_3] = 0).$$

So we are combining linear expressions to form a new linear expression. Since we may consider random variables $X_1 \oplus X_2$ and $X_2 \oplus X_3$ to be independent, we can use the Piling-Up Lemma, to determine

$$\Pr(X_1 \oplus X_3 = 0) = 1/2 + 2\varepsilon_{1,2}\varepsilon_{2,3}$$

and, consequently,

$$\varepsilon_{1,3} = 2\varepsilon_{1,2}\varepsilon_{2,3}.$$

As we shall see, the expressions $X_1 \oplus X_2 = 0$ and $X_2 \oplus X_3 = 0$ are analogous to linear approximations of S-boxes and $X_1 \oplus X_3 = 0$ is analogous to a cipher approximation where the intermediate bit $X_2$ is eliminated. Of course, the real analysis will be more complex involving many S-box approximations.

### *3.3 Analyzing the Cipher Components*

Before considering the attack in any more detail on the overall cipher, we first require knowledge of the linear vulnerabilities of an S-box. Consider the S-box representation of Figure 2 with input $X = [X_1\ X_2\ X_3\ X_4]$ and a corresponding output $Y = [Y_1\ Y_2\ Y_3\ Y_4]$. All linear approximations can be examined to determine their usefulness by computing the probability bias for each. Hence, we are examining all expressions of the form of equation (1) where $X$ and $Y$ are the S-box input and outputs, respectively.

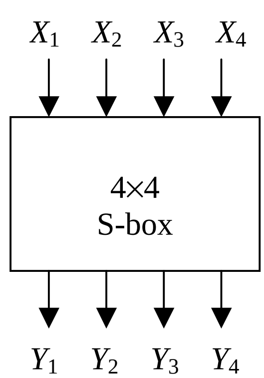


**Figure 2.** S-box Mapping

For example, for the S-box used in our cipher, consider the linear expression $X_2 \oplus X_3 \oplus Y_1 \oplus Y_3 \oplus Y_4 = 0$ or equivalently

$$X_2 \oplus X_3 = Y_1 \oplus Y_3 \oplus Y_4.$$

Applying all 16 possible input values for $X$ and examining the corresponding output values $Y$, it may be observed that for exactly 12 out the 16 cases, the expression above holds true. Hence, the probability bias is 12/16–1/2 = 1/4. This is presented in Table 3.

Similarly, for equation

$$X_1 \oplus X_4 = Y_2$$

it may be seen that the probability bias is 0 and for equation

$$X_3 \oplus X_4 = Y_1 \oplus Y_4$$

the probability bias is 2/16–1/2 = –3/8. In the last case, the best approximation is an affine approximation as indicated by the minus sign. However, the success of the attack is based on magnitude of the bias and, as we shall see, affine approximations can be used equivalently to linear approximations.

| $X_1$ | $X_2$ | $X_3$ | $X_4$ | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ | $X_2 \oplus X_3$ | $Y_1 \oplus Y_3 \oplus Y_4$ | $X_1 \oplus X_4$ | $Y_2$ | $X_3 \oplus X_4$ | $Y_1 \oplus Y_4$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |

**Table 3.** Sample Linear Approximations of S-box

| | Output Sum | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Input Sum | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 0 | +8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | –2 | –2 | 0 | 0 | –2 | +6 | +2 | +2 | 0 | 0 | +2 | +2 | 0 | 0 |
| 2 | 0 | 0 | –2 | –2 | 0 | 0 | –2 | –2 | 0 | 0 | +2 | +2 | 0 | 0 | –6 | +2 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | +2 | –6 | –2 | –2 | +2 | +2 | –2 | –2 |
| 4 | 0 | +2 | 0 | –2 | –2 | –4 | –2 | 0 | 0 | –2 | 0 | +2 | +2 | –4 | +2 | 0 |
| 5 | 0 | –2 | –2 | 0 | –2 | 0 | +4 | +2 | –2 | 0 | –4 | +2 | 0 | –2 | –2 | 0 |
| 6 | 0 | +2 | –2 | +4 | +2 | 0 | 0 | +2 | 0 | –2 | +2 | +4 | –2 | 0 | 0 | –2 |
| 7 | 0 | –2 | 0 | +2 | +2 | –4 | +2 | 0 | –2 | 0 | +2 | 0 | +4 | +2 | 0 | +2 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | –2 | +2 | +2 | –2 | +2 | –2 | –2 | –6 |
| 9 | 0 | 0 | –2 | –2 | 0 | 0 | –2 | –2 | –4 | 0 | –2 | +2 | 0 | +4 | +2 | –2 |
| A | 0 | +4 | –2 | +2 | –4 | 0 | +2 | –2 | +2 | +2 | 0 | 0 | +2 | +2 | 0 | 0 |
| B | 0 | +4 | 0 | –4 | +4 | 0 | +4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| C | 0 | –2 | +4 | –2 | –2 | 0 | +2 | 0 | +2 | 0 | +2 | +4 | 0 | +2 | 0 | –2 |
| D | 0 | +2 | +2 | 0 | –2 | +4 | 0 | +2 | –4 | –2 | +2 | 0 | +2 | 0 | 0 | +2 |
| E | 0 | +2 | +2 | 0 | –2 | –4 | 0 | +2 | –2 | 0 | 0 | –2 | –4 | +2 | –2 | 0 |
| F | 0 | –2 | –4 | –2 | –2 | 0 | +2 | 0 | 0 | –2 | +4 | –2 | –2 | 0 | +2 | 0 |

**Table 4.** Linear Approximation Table

A complete enumeration of all linear approximations of the S-box in our cipher is given in the *linear approximation table* of Table 4. Each element in the table represents the number of matches between the linear equation represented in hexadecimal as "Input Sum" and the sum of the output bits represented in hexadecimal as "Output Sum" minus 8. Hence, dividing an element value by 16 gives the probability bias for the particular linear combination of input and output bits. The hexadecimal value representing a sum, when viewed as a binary value indicates the variables involved in the sum. For a linear combination of input variables represented as $a_1 \cdot X_1 \oplus a_2 \cdot X_2 \oplus a_3 \cdot X_3 \oplus a_4 \cdot X_4$ where $a_i \in \{0,1\}$ and "$\cdot$" represents binary AND, the hexadecimal value represents the binary value $a_1 a_2 a_3 a_4$, where $a_1$ is the most significant bit. Similarly, for a linear combination of output bits $b_1 \cdot Y_1 \oplus b_2 \cdot Y_2 \oplus b_3 \cdot Y_3 \oplus b_4 \cdot Y_4$ where $b_i \in \{0,1\}$, the hexadecimal value represents the binary vector $b_1 b_2 b_3 b_4$. Hence, the bias of linear equation $X_3 \oplus X_4 = Y_1 \oplus Y_4$ (hex input 3 and hex output 9) is –6/16 = –3/8 and the probability that the linear equation holds true is given by 1/2 – 3/8 = 1/8.

Some basic properties of the linear approximation table can be noted. For example, the probability that any sum of a non-empty subset of output bits is equal to the sum involving no input bits is exactly 1/2 since any linear combination of output bits must have an equal number of zeros and ones for a bijective S-box. Also, the linear combination involving no output bits will always equal the linear combination of no input bits resulting in a bias of +1/2 and a table value of +8 in the top left corner. Hence, the top row of the table is all zeros, except for the leftmost value. Similarly, the first column is all zeros except for the topmost value. It can also be noted the sum of any row or any column must be either +8 or –8. We leave the proof of this as an exercise to the reader.

### *3.4 Constructing Linear Approximations for the Complete Cipher*

Once the linear approximation information has been compiled for the S-boxes in an SPN, we have the data to proceed with determining linear approximations of the overall cipher of the form of equation (1). This can be achieved by concatenating appropriate linear approximations of S-boxes. By constructing a linear approximation involving plaintext bits and data bits from the output of the second last round of S-boxes, it is possible to attack the cipher by recovering a subset of the subkey bits that follow the last round. We illustrate with an example.

Consider an approximation involving $S_{12}$, $S_{22}$, $S_{32}$, and $S_{34}$ as illustrated in Figure 3. Note that this actually develops an expression for the first 3 rounds of the cipher and not the full 4 rounds. We shall see how this is useful in deriving the subkey bits after the last round in the next section.

We use the following approximations of the S-box:

| | |
|---|---|
| $S_{12}$: $X_1 \oplus X_3 \oplus X_4 = Y_2$ | with probability 12/16 and bias +1/4 |
| $S_{22}$: $X_2 = Y_2 \oplus Y_4$ | with probability 4/16 and bias –1/4 |
| $S_{32}$: $X_2 = Y_2 \oplus Y_4$ | with probability 4/16 and bias –1/4 |
| $S_{34}$: $X_2 = Y_2 \oplus Y_4$ | with probability 4/16 and bias –1/4 |

Letting $U_i$ ($V_i$) represent the 16-bit block of bits at the input (output) of the round $i$ S-boxes and $U_{i,j}$ ($V_{i,j}$) represent the $j$-th bit of block $U_i$ ($V_i$) (where bits are numbered from 1 to 16 from left to right in the figure of the cipher). Similarly, let $K_i$ represent the subkey block of bits exclusive-ORed at the input to round $i$, with the exception that $K_5$ is the key exclusive-ORed at the output of round 4.

Hence, $U_1 = P \oplus K_1$ where $P$ represents the block of 16 plaintext bits and "$\oplus$" represents the bit-wise exclusive-OR. Using the linear approximation of the 1st round, we then have

$$\begin{aligned} V_{1,6} &= U_{1,5} \oplus U_{1,7} \oplus U_{1,8} \\ &= (P_5 \oplus K_{1,5}) \oplus (P_7 \oplus K_{1,7}) \oplus (P_8 \oplus K_{1,8}) \end{aligned} \qquad (2)$$

with probability 3/4. For the approximation in the 2nd round, we have

$$V_{2,6} \oplus V_{2,8} = U_{2,6}$$

with probability 1/4. Since $U_{2,6} = V_{1,6} \oplus K_{2,6}$, we can get an approximation of the form

$$V_{2,6} \oplus V_{2,8} = V_{1,6} \oplus K_{2,6}$$

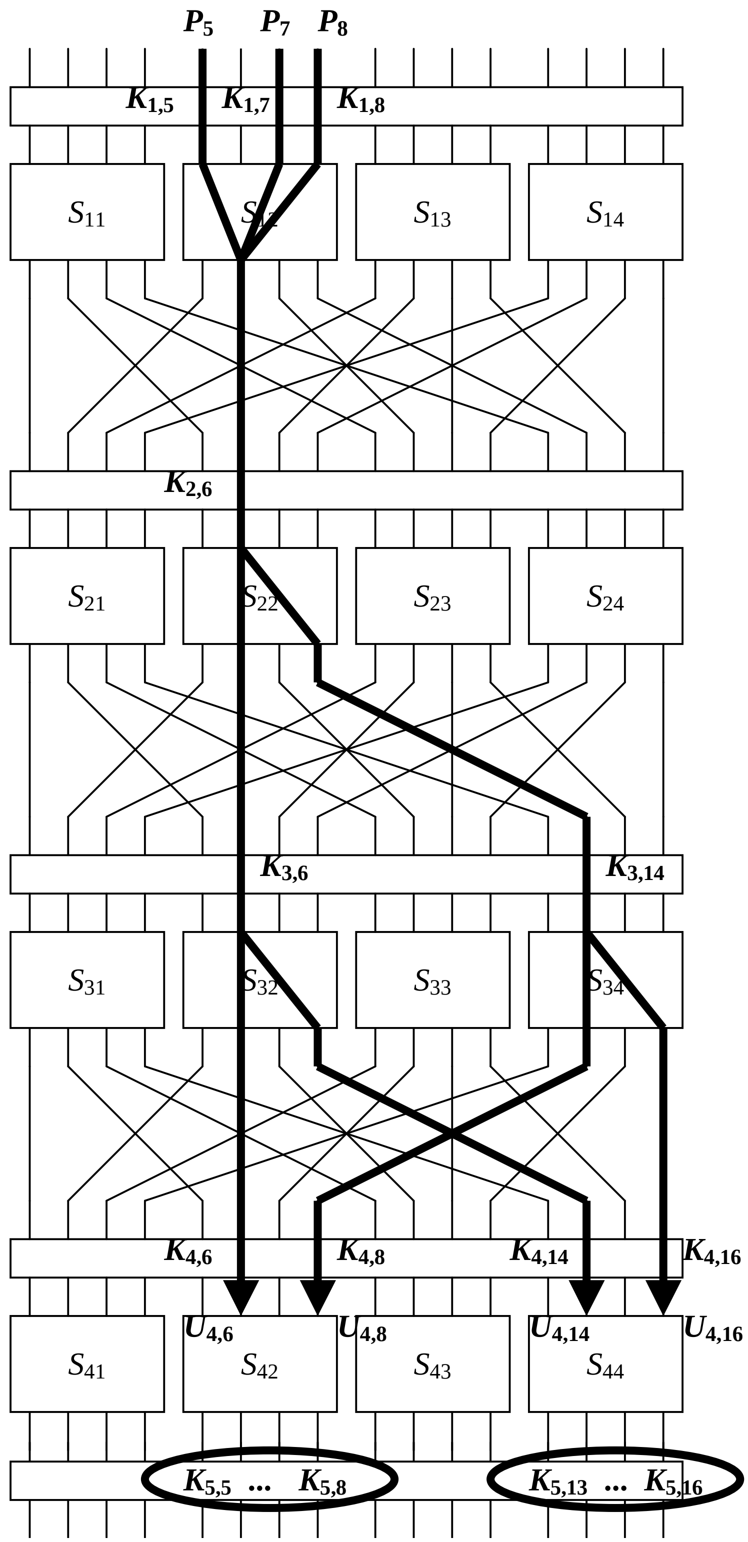


**Figure 3.** Sample Linear Approximation

with probability 1/4 and combining this with (2) which holds with probability of 3/4 gives

$$V_{2,6} \oplus V_{2,8} \oplus P_5 \oplus P_7 \oplus P_8 \oplus K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} = 0 \qquad (3)$$

which holds with probability of 1/2 + 2(3/4–1/2)(1/4–1/2) = 3/8 (that is, with a bias of –1/8) by application of the Piling-Up Lemma. Note that we are using the assumption that the approximations of S-boxes are independent which, although not strictly correct, works well in practice for most ciphers.

For round 3, we note that

$$V_{3,6} \oplus V_{3,8} = U_{3,6}$$

with probability 1/4 and

$$V_{3,14} \oplus V_{3,16} = U_{3,14}$$

with probability 1/4. Hence, since $U_{3,6} = V_{2,6} \oplus K_{3,6}$ and $U_{3,14} = V_{2,8} \oplus K_{3,14}$,

$$V_{3,6} \oplus V_{3,8} \oplus V_{3,14} \oplus V_{3,16} \oplus V_{2,6} \oplus K_{3,6} \oplus V_{2,8} \oplus K_{3,14} = 0 \qquad (4)$$

with probability of $1/2 + 2(1/4–1/2)^2 = 5/8$ (that is, with a bias of +1/8). Again, we have applied the Piling-Up Lemma.

Now combining (3) and (4), to incorporate all four S-box approximations, we get

$$V_{3,6} \oplus V_{3,8} \oplus V_{3,14} \oplus V_{3,16} \oplus P_5 \oplus P_7 \oplus P_8 \\ \oplus K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \oplus K_{3,6} \oplus K_{3,14} = 0.$$

Noting that $U_{4,6} = V_{3,6} \oplus K_{4,6}$, $U_{4,8} = V_{3,14} \oplus K_{4,8}$, $U_{4,14} = V_{3,8} \oplus K_{4,14}$, and $U_{4,16} = V_{3,16} \oplus K_{4,16}$, we can then write

$$U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus P_5 \oplus P_7 \oplus P_8 \oplus \Sigma_K = 0.$$

where

$$\Sigma_K = K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \oplus K_{3,6} \oplus K_{3,14} \oplus K_{4,6} \oplus K_{4,8} \oplus K_{4,14} \oplus K_{4,16}$$

and $\Sigma_K$ is fixed at either 0 or 1 depending on the key of the cipher. By application of the Piling-Up Lemma, the above expression holds with probability $1/2+2^3(3/4–1/2)(1/4–1/2)^3 = 15/32$ (that is, with a bias of –1/32).

Now since $\Sigma_K$ is fixed, we note that

$$U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus P_5 \oplus P_7 \oplus P_8 = 0 \tag{5}$$

must hold with a probability of either 15/32 or (1−15/32) = 17/32, depending on whether $\Sigma_K = 0$ or 1, respectively. In other words, we now have a linear approximation of the first three rounds of the cipher with a bias of magnitude 1/32. We must now discuss how such a bias can be used to determine some of the key bits.

### *3.5 Extracting Key Bits*

Once an $R$−1 round linear approximation is discovered for a cipher of $R$ rounds with a suitably large enough linear probability bias, it is conceivable to attack the cipher by recovering bits of the last subkey. In the case of our example cipher, it is possible to extract bits from subkey $K_5$ given a 3 round linear approximation. We shall refer to the bits to be recovered from the last subkey as the *target partial subkey*. Specifically, the target partial subkey bits are the bits from the last subkey associated with the S-boxes in the last round influenced by the data bits involved in the linear approximation.

The process followed involves partially decrypting the last round of the cipher. Specifically, for all possible values of the target partial subkey, the corresponding ciphertext bits are exclusive-ORed with the bits of the target partial subkey and the result is run backwards through the corresponding S-boxes. This is done for all known plaintext/ciphertext samples and a count is kept for each value of the target partial subkey. The count for a particular target partial subkey value is incremented when the linear expression holds true for the bits into the last round's S-boxes (determined by the partial decryption) and the known plaintext bits. The target partial subkey value which has the count which differs the greatest from half the number of plaintext/ciphertext samples is assumed to represent the correct values of the target partial subkey bits. This works because it is assumed that the correct partial subkey value will result in the linear approximation holding with a probability significantly different from 1/2. (Whether it is above or below 1/2 depends on whether a linear or affine expression is the best approximation and this depends on the unknown values of the subkey bits implicitly involved in the linear expression.) An incorrect subkey is assumed to result in a relatively random guess at the bits entering the S-boxes of the last round and as a result, the linear expression will hold with a probability close to 1/2.

Let's now put this into the context of our example. The linear expression of (5) affects the inputs to S-boxes $S_{42}$ and $S_{44}$ in the last round. For each plaintext/ciphertext sample, we would try all 256 values for the target partial subkey [$K_{5,5}$...$K_{5,8}$, $K_{5,13}$...$K_{5,16}$]. For each partial subkey value, we would increment the count whenever equation (5) holds true, where we determine the value of [$U_{4,5}$...$U_{4,8}$, $U_{4,13}$...$U_{4,16}$] by running the data backwards through the target partial subkey and S-boxes $S_{24}$ and $S_{44}$. The count which deviates the largest from half of the number of plaintext/ciphertext samples is assumed to the correct value. Whether the deviation is positive or negative will depend on the values of the subkey bits involved in $\Sigma_K$. When $\Sigma_K = 0$, the linear approximation of (5) will serve as the estimate (with probability < 1/2) and when $\Sigma_K = 1$, (5) will hold with a probability > 1/2.

We have simulated attacking our basic cipher by generating 10000 known plaintext/ciphertext values and following the cryptanalytic process described for partial subkey values $[K_{5,5}...K_{5,8}]$ = [0010] (hex 2) and $[K_{5,13}...K_{5,16}]$ = [0100] (hex 4). As expected, the count which differed the most from 5000 corresponded to target partial subkey value $[2,4]_{hex}$, confirming that the attack has successfully derived the subkey bits. Table 5 highlights a partial summary of the data derived from the subkey counts. (The complete data involves 256 data entries, one for each target partial subkey value.) The values in the table indicate the bias magnitude derived from

$$| \text{bias} | = | \text{count} - 5000 | / 10000$$

where the count is the count corresponding to the particular partial subkey value.

As can be seen from the partial results in the table, the largest bias occurs for partial subkey value $[K_{5,5}...K_{5,8}, K_{5,13}...K_{5,16}]$ = [2,4] and this observation was, in fact, found to be true for the complete set of partial subkey values.

| *partial subkey* $[K_{5,5}...K_{5,8}, K_{5,13}...K_{5,16}]$ | \| bias \| | *partial subkey* $[K_{5,5}...K_{5,8}, K_{5,13}...K_{5,16}]$ | \| bias \| |
|---|---|---|---|
| 1 C | 0.0031 | 2 A | 0.0044 |
| 1 D | 0.0078 | 2 B | 0.0186 |
| 1 E | 0.0071 | 2 C | 0.0094 |
| 1 F | 0.0170 | 2 D | 0.0053 |
| 2 0 | 0.0025 | 2 E | 0.0062 |
| 2 1 | 0.0220 | 2 F | 0.0133 |
| 2 2 | 0.0211 | 3 0 | 0.0027 |
| 2 3 | 0.0064 | 3 1 | 0.0050 |
| **2 4** | **0.0336** | 3 2 | 0.0075 |
| 2 5 | 0.0106 | 3 3 | 0.0162 |
| 2 6 | 0.0096 | 3 4 | 0.0218 |
| 2 7 | 0.0074 | 3 5 | 0.0052 |
| 2 8 | 0.0224 | 3 6 | 0.0056 |
| 2 9 | 0.0054 | 3 7 | 0.0048 |

**Table 5.** Experimental Results for Linear Attack

The experimentally determined bias value of 0.0336 is very close to the expected value of 1/32 = 0.03125. Note that, although the correct target partial subkey has clearly the highest bias, other large bias values occur indicating that the examination of incorrect target partial subkeys is not precisely equivalent to comparing random data to a linear expression (where the bias could be expected to be very close to zero). Inconsistencies in the experimental biases can occur for several reasons including the S-box properties influencing the partial decryption for different partial subkey values, the imprecision of the independence assumption required for use in the Piling-Up Lemma, and the influence of *linear hulls* (to be discussed in the next section).

### *3.6 Complexity of Attack*

We refer to the S-boxes involved in the linear approximation as *active* S-boxes. In Figure 3, the four S-boxes in rounds 1 to 3 influenced by the highlighted lines are active. The probability that a linear expression holds true is related to the linear probability bias in the active S-boxes and the number of active S-boxes. In general, the larger the magnitude of the bias in the S-boxes, the larger the magnitude of the bias of the overall expression. Also, the fewer active S-boxes, the larger the magnitude of the overall linear expression bias.

Let $\varepsilon$ represent the bias from 1/2 of the probability that the linear expression for the complete cipher holds. In his paper, Matsui shows that the number of known plaintexts required in the attack is proportional to $\varepsilon^{-2}$ and, letting $N_L$ represent the number of known plaintexts required, it is reasonable to approximate $N_L$ by

$$N_L \approx 1/\varepsilon^2.$$

In practice, it is generally reasonable to expect some small multiple of $\varepsilon^{-2}$ known plaintexts are required. Although strictly speaking, the complexity of the cryptanalysis could be characterized in both time and space (or memory) domains, we refer to the data required to mount the attack when considering the complexity of the cryptanalysis. That is, we assume that if we are able to acquire $N_L$ plaintexts, we are able to process them.

Since the bias is derived using the Piling-Up Lemma where each term in the product refers to an S-box approximation, it is easy to see that the bias is dependent on the biases of the S-box linear approximations and the number of active S-boxes involved. General approaches to providing security against linear cryptanalysis have focused on optimizing the S-boxes (i.e., minimizing the largest bias) and finding structures to maximize the number of active S-boxes. The design principles of Rijndael are an excellent example of such an approach.

It must be cautioned, however, the concept of a "proof" of security to linear cryptanalysis is usually premised on the nonexistence of highly likely linear approximations. However, the computation of the probability of such linear approximations is based on the assumption that each S-box approximation is independent (so that the Piling-Up Lemma can be used) and on the assumption that one linear approximation scenario (i.e., a particular set of active S-boxes) is sufficient to determine the best linear expression between plaintext bits and data bits at the input to the last round. The reality is that the S-box approximations are not independent and this can have significant impact on the computation of the probability. Also, linear approximation scenarios involving the same plaintext and last round input bits but different sets of active S-boxes can combine to give a linear probability higher than that predicted by one set of active S-boxes. This concept is referred to as a *linear hull* [16]. Most notably for example, a number of linear approximation scenarios may have very small biases and on their own seem to imply that a cipher might be immune to a linear attack. However, when these scenarios are combined, the resulting linear expression of plaintext and last round input bits might have a very high bias. Nevertheless, the approach outlined in this paper, tends to work well for many ciphers because the independence assumption is a reasonable approximation and when one linear approximation scenario of a particular set of active S-boxes has a high bias, it tends to dominate the linear hull.

# 4. Differential Cryptanalysis

In this section, we now turn our focus to the application of differential cryptanalysis to the basic SPN cipher.

### *4.1 Overview of Basic Attack*

Differential cryptanalysis exploits the high probability of certain occurrences of plaintext differences and differences into the last round of the cipher. For example, consider a system with input $X = [X_1\ X_2\ ...\ X_n]$ and output $Y = [Y_1\ Y_2\ ...\ Y_n]$. Let two inputs to the system be $X'$ and $X''$ with the corresponding outputs $Y'$ and $Y''$, respectively. The input difference is given by $\Delta X = X' \oplus X''$ where "$\oplus$" represents a bit-wise exclusive-OR of the $n$-bit vectors and, hence,

$$\Delta X = [\Delta X_1\ \Delta X_2\ ...\ \Delta X_n]$$

where $\Delta X_i = X_i' \oplus X_i''$ with $X_i'$ and $X_i''$ representing the $i$-th bit of $X'$ and $X''$, respectively. Similarly, $\Delta Y = Y' \oplus Y''$ is the output difference and

$$\Delta Y = [\Delta Y_1\ \Delta Y_2\ ...\ \Delta Y_n]$$

where $\Delta Y_i = Y_i' \oplus Y_i''$.

In an ideally randomizing cipher, the probability that a particular output difference $\Delta Y$ occurs given a particular input difference $\Delta X$ is $1/2^n$ where $n$ is the number of bits of $X$. Differential cryptanalysis seeks to exploit a scenario where a particular $\Delta Y$ occurs given a particular input difference $\Delta X$ with a very high probability $p_D$ (i.e., much greater than $1/2^n$). The pair $(\Delta X, \Delta Y)$ is referred to as a *differential*.

Differential cryptanalysis is a chosen plaintext attack, meaning that the attacker is able to select inputs and examine outputs in an attempt to derive the key. For differential cryptanalysis, the attacker will select pairs of inputs, $X'$ and $X''$, to satisfy a particular $\Delta X$, knowing that for that $\Delta X$ value, a particular $\Delta Y$ value occurs with high probability.

In this paper, we investigate the construction of a differential $(\Delta X, \Delta Y)$ involving plaintext bits as represented by $X$ and the input to the last round of the cipher as represented by $Y$. We shall do this by examining high likely *differential characteristics* where a differential characteristic is a sequence of input and output differences to the rounds so that the output difference from one round corresponds to the input difference for the next round. Using the highly likely differential characteristic gives us the opportunity to exploit information coming into the last round of the cipher to derive bits from the last layer of subkeys.

As with linear cryptanalysis, to construct highly likely differential characteristics, we examine the properties of individual S-boxes and use these properties to determine the complete differential characteristic. Specifically, we consider the input and output differences of the S-boxes in order to determine a high probability difference pair. Combining S-box difference pairs from round to round so that the nonzero output difference bits from one round correspond to the non-zero input difference bits of the next round, enables us to find a high probability differential consisting of the plaintext difference and the difference of the input to the last round. The subkey bits of the cipher end up disappearing from the difference expression because they are involved in both data sets and, hence, considering their influence on the difference involves exclusive-ORing subkey bits with themselves, the result of which is zero.

### *4.2 Analyzing the Cipher Components*

We examine now the difference pairs of an S-box. Consider the 4×4 S-box representation of Figure 2 with input $X = [X_1\ X_2\ X_3\ X_4]$ and output $Y = [Y_1\ Y_2\ Y_3\ Y_4]$. All difference pairs of an S-box, $(\Delta X, \Delta Y)$, can be examined and the probability of $\Delta Y$ given $\Delta X$ can be derived by considering input pairs $(X', X'')$ such that $X' \oplus X'' = \Delta X$. Since the ordering of the pair is not relevant, for a 4×4 S-box we need only consider all 16 values for $X'$ and then the value of $\Delta X$ constrains the value of $X''$ to be $X'' = X' \oplus \Delta X$.

Considering the S-box of our cipher given in Section 2, we can derive the resulting values of $\Delta Y$ for each input pair $(X', X''=X' \oplus \Delta X)$. For example, the binary values of $X$, $Y$, and the corresponding values for $\Delta Y$ for given input pairs $(X, X \oplus \Delta X)$ are presented in Table 6 for $\Delta X$ values of 1011 (hex B), 1000 (hex 8), and 0100 (hex 4). The last three columns of the table represent $\Delta Y$ values for the value of $X$ (as given by the row) and the particular $\Delta X$ value for each column. From the table, we can see that the number of occurrences of $\Delta Y$ = 0010 for $\Delta X$ = 1011 is 8 out of 16 possible values (i.e., a probability of 8/16); the number of occurrences of $\Delta Y$ = 1011 given $\Delta X$ = 1000 is 4 out of 16; the number of occurrences of $\Delta Y$ = 1010 given $\Delta X$ = 0100 is 0 out of 16. If the S-box could be "ideal", the number of occurrences of difference pair values would all be 1 to give a probability of 1/16 of the occurrence of a particular $\Delta Y$ value given $\Delta X$. (It turns out that such an "ideal" S-box is not mathematically possible.)

We can tabularize the complete data for an S-box in a *difference distribution table* in which the rows represent $\Delta X$ values (in hexadecimal) and the columns represent $\Delta Y$ values (in hexadecimal). The difference distribution table for the S-box of Table 1 is given in Table 7. Each element of the table represents the number of occurrences of the corresponding output difference $\Delta Y$ value given the input difference $\Delta X$. Note that, besides the special case of $(\Delta X = 0, \Delta Y = 0)$, the largest value in the table is 8, corresponding to $\Delta X$ = B and $\Delta Y$ = 2. Hence, the probability that $\Delta Y$ = 2 given an arbitrary pair of input values that satisfy $\Delta X$ = B is 8/16. The smallest value in the table is 0 and occurs for many difference pairs. In this case, the probability of the $\Delta Y$ value occurring given the $\Delta X$ value is 0.

| $X$ | $Y$ | $\Delta Y$ | | |
|---|---|---|---|---|
| | | $\Delta X$ = 1011 | $\Delta X$ = 1000 | $\Delta X$ = 0100 |
| 0000 | 1110 | 0010 | 1101 | 1100 |
| 0001 | 0100 | 0010 | 1110 | 1011 |
| 0010 | 1101 | 0111 | 0101 | 0110 |
| 0011 | 0001 | 0010 | 1011 | 1001 |
| 0100 | 0010 | 0101 | 0111 | 1100 |
| 0101 | 1111 | 1111 | 0110 | 1011 |
| 0110 | 1011 | 0010 | 1011 | 0110 |
| 0111 | 1000 | 1101 | 1111 | 1001 |
| 1000 | 0011 | 0010 | 1101 | 0110 |
| 1001 | 1010 | 0111 | 1110 | 0011 |
| 1010 | 0110 | 0010 | 0101 | 0110 |
| 1011 | 1100 | 0010 | 1011 | 1011 |
| 1100 | 0101 | 1101 | 0111 | 0110 |
| 1101 | 1001 | 0010 | 0110 | 0011 |
| 1110 | 0000 | 1111 | 1011 | 0110 |
| 1111 | 0111 | 0101 | 1111 | 1011 |

**Table 6.** Sample Difference Pairs of the S-box

| | Output Difference | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Input Difference | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 2 | 0 | 2 | 4 | 0 | 4 | 2 | 0 | 0 |
| 2 | 0 | 0 | 0 | 2 | 0 | 6 | 2 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 2 | 0 |
| 3 | 0 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 4 | 2 | 0 | 2 | 0 | 0 | 4 |
| 4 | 0 | 0 | 0 | 2 | 0 | 0 | 6 | 0 | 0 | 2 | 0 | 4 | 2 | 0 | 0 | 0 |
| 5 | 0 | 4 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 4 | 0 | 2 | 0 | 0 | 2 |
| 6 | 0 | 0 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 |
| 7 | 0 | 0 | 2 | 2 | 2 | 0 | 2 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 4 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 4 | 0 | 4 | 2 | 2 |
| 9 | 0 | 2 | 0 | 0 | 2 | 0 | 0 | 4 | 2 | 0 | 2 | 2 | 2 | 0 | 0 | 0 |
| A | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 2 | 0 | 0 | 4 | 0 |
| B | 0 | 0 | 8 | 0 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| C | 0 | 2 | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 2 | 0 | 6 | 0 | 0 |
| D | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 4 | 2 | 0 | 2 | 0 | 2 | 0 | 2 | 0 |
| E | 0 | 0 | 2 | 4 | 2 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| F | 0 | 2 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 4 | 0 | 2 | 0 | 0 | 2 | 0 |

**Table 7.** Difference Distribution Table

There are several general properties of the difference distribution table that should be mentioned. First, it should be noted that the sum of all elements in a row is $2^n$ =16; similarly the sum of any column is $2^n$ =16. Also, all element values are even: this results because a pair of input (or output) values represented as $(X', X'')$ has the same $\Delta X$ value as the pair $(X'', X')$ since $\Delta X = X' \oplus X'' = X'' \oplus X'$. As well, the input difference of $\Delta X = 0$ must lead to an output difference of $\Delta Y = 0$ for the one-to-one mapping of the S-box. Hence, the top right corner of the table has a value of $2^n = 16$ and all other values in the first row and first column are 0. Finally, if we could construct an ideal S-box, which gives no differential information about the output given the input value, the S-box would have all elements in the table equal to 1 and the probability of occurrence of a particular value for $\Delta Y$ given a particular value of $\Delta X$ would be $1/2^n = 1/16$. However, as the properties discussed above must hold, this is clearly not achievable.

Before we proceed to discuss the combining of S-box difference pairs to derive a differential characteristic and an estimate of a good differential to use in the attack, we must discuss the influence of the key on the S-box differential. Consider Figure 4. The input to the "unkeyed" S-box is $X$ and the output $Y$. However, in the cipher structure we must consider the keys applied at the input of each S-box. In this case, if we let the input to the "keyed" S-box be $W = [W_1\ W_2\ W_3\ W_4]$, we can consider the input difference to the keyed S-box to be

$$\Delta W = [W_1' \oplus W_1''\ \ W_2' \oplus W_2''\ \ \ldots\ \ W_n' \oplus W_n'']$$

where $W' = [W_1'\ W_2' \ldots W_n']$ and $W'' = [W_1''\ W_2'' \ldots W_n'']$ represent the two input values.

Since the key bits remain the same for both $W'$ and $W''$,

$$\begin{aligned} \Delta W_i &= W_i' \oplus W_i'' = (X_i' \oplus K_i) \oplus (X_i'' \oplus K_i) \\ &= X_i' \oplus X_i'' = \Delta X_i \end{aligned}$$

since $K_i \oplus K_i = 0$. Hence, the key bits have no influence on the input difference value and can be ignored. In other words, the keyed S-box has the same difference distribution table as the unkeyed S-box.

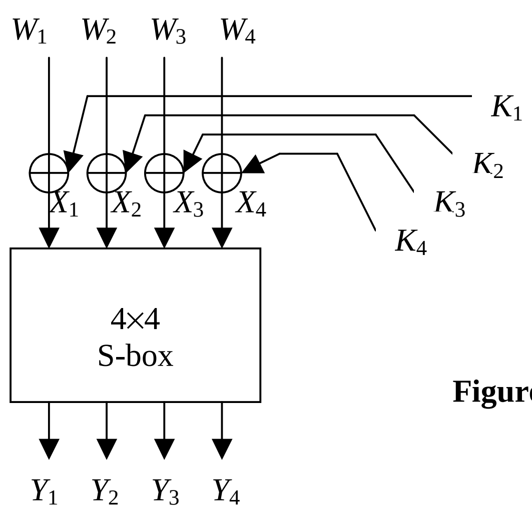


**Figure 4.** Keyed S-box

### *4.3 Constructing Differential Characteristics*

Once the differential information has been compiled for the S-boxes in an SPN, we have the data to proceed with determining a useful differential characteristic of the overall cipher. This can be done by concatenating appropriate difference pairs of S-boxes. By constructing a differential characteristic of certain S-box difference pairs in each round, such that a differential involves plaintexts bits and data bits to the input of the last round of S-boxes, it is possible to attack the cipher by recovering a subset of the subkey bits following the last round. We illustrate the construction of a differential characteristic with an example.

Consider a differential characteristic involving $S_{12}$, $S_{23}$, $S_{32}$, and $S_{33}$. As was the case for linear cryptanalysis, it is useful to visualize the differential characteristic in the form of a diagram as shown in Figure 5. The diagram illustrates the influence of non-zero differences in bits as they traverse the network, highlighting the S-boxes that may be considered as active (i.e., for which there is a non-zero differential). Note that this develops a differential characteristic for the first 3 rounds of the cipher and not the full 4 rounds. We shall see how this is useful in deriving bits from the last subkey in the next section.

We use the following difference pairs of the S-box:

$S_{12}$: $\Delta X = \mathrm{B} \rightarrow \Delta Y = 2$ with probability 8/16
$S_{23}$: $\Delta X = 4 \rightarrow \Delta Y = 6$ with probability 6/16
$S_{32}$: $\Delta X = 2 \rightarrow \Delta Y = 5$ with probability 6/16
$S_{33}$: $\Delta X = 2 \rightarrow \Delta Y = 5$ with probability 6/16

All other S-boxes will have zero input difference and consequently zero output difference.

The input difference to the cipher is equivalent to the input difference to the first round and is given by

$$\Delta P = \Delta U_1 = [0000\ 1011\ 0000\ 0000]$$

where again, as with our presentation of linear cryptanalysis in Section 3, we are using $U_i$ to represent the input to the $i$-th round S-boxes and $V_i$ to represent the output of the $i$-th round S-boxes. Hence, $\Delta U_i$ and $\Delta V_i$ represent the corresponding differences. As a result,

$$\Delta V_1 = [0000\ 0010\ 0000\ 0000]$$

considering the difference pair for $S_{12}$ listed above and following the round 1 permutation

$$\Delta U_2 = [0000\ 0000\ 0100\ 0000]$$

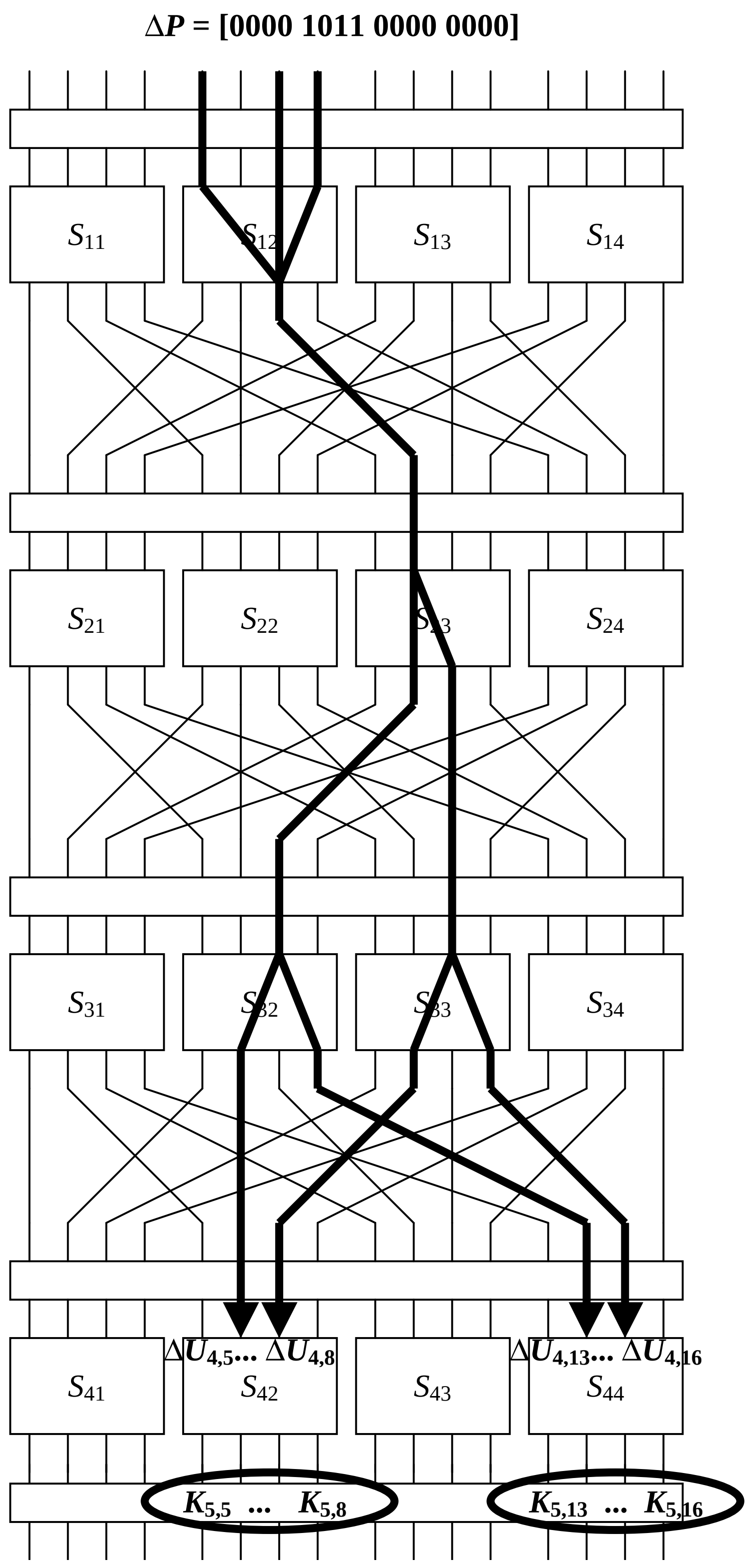


**Figure 5.** Sample Differential Characteristic

with probability of 8/16 = 1/2 given the plaintext difference $\Delta P$.

Now the second round differential using the difference pair for $S_{23}$ results in

$$\Delta V_2 = [0000\ 0000\ 0110\ 0000]$$

and the permutation of round 2 gives

$$\Delta U_3 = [0000\ 0010\ 0010\ 0000]$$

with probability 6/16 given $\Delta U_2$ and a probability of $8/16 \times 6/16 = 3/16$ given $\Delta P$. In determining the probability given plaintext difference $\Delta P$, we have assumed that the differential of the first round is independent of the differential of the 2nd round and, hence, the probability of both occurring is determined by the product of the probabilities.

Subsequently, we can use the differences for the S-boxes of the third round, $S_{32}$ and $S_{33}$, and the permutation of the third round to arrive at

$$\Delta V_3 = [0000\ 0101\ 0101\ 0000]$$

and

$$\Delta U_4 = [0000\ 0110\ 0000\ 0110] \tag{6}$$

with a probability of $(6/16)^2$ given $\Delta U_3$ and, hence, a probability of $8/16 \times 6/16 \times (6/16)^2 = 27/1024$ given plaintext difference $\Delta P$, where again we have assumed independence between the difference pairs of S-boxes in all rounds.

During the cryptanalysis process, many pairs of plaintexts for which $\Delta P$ = [0000 1011 0000 0000] will be encrypted. With high probability, 27/1024, the differential characteristic illustrated will occur. We term such pairs for $\Delta P$ as *right pairs*. Plaintext difference pairs for which the characteristic does not occur are referred to as *wrong pairs*.

### *4.4 Extracting Key Bits*

Once an $R$−1 round differential characteristic is discovered for a cipher of $R$ rounds with a suitably large enough probability, it is conceivable to attack the cipher by recovering bits from the last subkey. In the case of our example cipher, it is possible to extract bits from subkey $K_5$. The process followed involves partially decrypting the last round of the cipher and examining the input to the last round to determine if a right pair has probably occurred. We shall refer to the subkey bits following the last round at the output of S-boxes in the last round influenced by non-zero differences in the differential output as the *target partial subkey*. A partial decryption of the last round would involve, for all S-boxes in the last round influenced by non-zero differences in the differential, the exclusive-OR of the ciphertext with the target partial subkey bits and running the data backwards through the S-boxes, where all possible values for the target subkey bits would be tried.

A partial decryption is executed for each pair of ciphertexts corresponding to the pairs of plaintexts used to generate the input difference $\Delta P$ for all possible target partial subkey values. A count is kept for each value of the target partial subkey value. The count is incremented when the difference for the input to the last round corresponds to the value expected from the differential characteristic. The partial subkey value which has the largest count is assumed to indicate the correct values of the subkey bits. This works because it is assumed that the correct partial subkey value will result in the difference to the last round being frequently as expected from the characterstic (i.e., the occurrence of a right pair) since the characteristic has a high probability of occurring. (When a wrong pair has occurred, even with the partial decryption with the correct subkey, the count for the correct subkey will likely not be incremented.) An incorrect subkey is assumed to result in a relatively random guess at the bits entering the S-boxes of the last round and as a result, the difference will be as expected from the characteristic with a very low probability.

Considering the attack on our example cipher, the differential characteristic affects the inputs to S-boxes $S_{42}$ and $S_{44}$ in the last round. For each ciphertext pair, we would try all 256 values for $[K_{5,5}...K_{5,8},\ K_{5,13}...K_{5,16}]$. For each partial subkey value, we would increment the count whenever the input difference to the final round determined by the partial decryption is the same as (6), where we determine the value of $[\Delta U_{4,5}...\ \Delta U_{4,8}, \Delta U_{4,13}...\ \Delta U_{4,16}]$ by running the data backwards through the partial subkey and S-boxes $S_{24}$ and $S_{44}$. For each partial subkey value, the count represents the number of occurrences of differences that are consistent with right pairs (assuming that the partial subkey is the correct value). The count that is the largest is taken to be the correct value since we assume that we are observing the high probability occurrence of the right pair.

Note that it is not necessary to execute the partial decryption for every ciphertext pair. Since the input difference to the last round only influences 2 S-boxes, when the characteristic has occurred (i.e., for right pairs), the ciphertext bit differences corresponding to S-boxes $S_{41}$ and $S_{43}$ must be zero. Hence, we can *filter* out many wrong pairs by discarding ciphertext pairs for which zeros do not appear in the appropriate sub-blocks of the ciphertext difference. In these cases, since the ciphertext pair cannot correspond to a right pair, it is not necessary to examine $[\Delta U_{4,5}...\ \Delta U_{4,8}, \Delta U_{4,13}...\ \Delta U_{4,16}]$.

We have simulated attacking our basic cipher keyed using randomly generated subkeys by generating 5000 chosen plaintext/ciphertext pairs (i.e., 10000 encryptions with plaintext pairs satisfying $\Delta P$ = [0000 1011 0000 0000]) and following the process described above. The correct target partial subkey value was $[K_{5,5}...K_{5,8}, K_{5,13}...K_{5,16}]$ = [0010,0100] = $[2,4]_{hex}$. As expected, the largest count was observed for partial subkey value $[2,4]_{hex}$, confirming that the attack successfully derived the subkey bits. Table 8 highlights a partial summary of the data derived from the subkey counts. (The complete data involves 256 data entries, one for each partial subkey value.) The values in the table indicate the estimated probability of the occurrence of right pairs for the candidate partial subkey derived from

$$\text{prob} = \text{count} / 5000.$$

where the count is the count corresponding to the particular partial subkey value.

As can be seen from the sample results of the table, the largest probability occurs for partial subkey value $[K_{5,5}...K_{5,8}, K_{5,13}...K_{5,16}] = [2,4]_{hex}$ and this observation was, in fact, found to be true for the complete set of partial subkey values.

In our example, we would expect the probability of the occurrence of the right pair to be $p_D$ = 27/1024 = 0.0264 and we found experimentally the probability for the correct subkey value [2,4] gave $p_D$ = 0.0244. Note that sometimes other large count values occur for incorrect target partial subkeys. This indicates that the examination of incorrect target partial subkeys is not precisely equivalent to comparing random differences to the expected differential value. There are several factors which influence the counts to be different then our theorized expectations including the S-box properties influencing the partial decryption for different partial subkeys, the imprecision of the independence assumption required for determination of the characteristic probability, and the concept that *differentials* are composed of multiple differential characteristics (to be discussed in the next section).

| *partial subkey* $[K_{5,5}...K_{5,8}, K_{5,13}...K_{5,16}]$ | prob | *partial subkey* $[K_{5,5}...K_{5,8}, K_{5,13}...K_{5,16}]$ | prob |
|---|---|---|---|
| 1 C | 0.0000 | 2 A | 0.0032 |
| 1 D | 0.0000 | 2 B | 0.0022 |
| 1 E | 0.0000 | 2 C | 0.0000 |
| 1 F | 0.0000 | 2 D | 0.0000 |
| 2 0 | 0.0000 | 2 E | 0.0000 |
| 2 1 | 0.0136 | 2 F | 0.0000 |
| 2 2 | 0.0068 | 3 0 | 0.0004 |
| 2 3 | 0.0068 | 3 1 | 0.0000 |
| **2 4** | **0.0244** | 3 2 | 0.0004 |
| 2 5 | 0.0000 | 3 3 | 0.0004 |
| 2 6 | 0.0068 | 3 4 | 0.0000 |
| 2 7 | 0.0068 | 3 5 | 0.0004 |
| 2 8 | 0.0030 | 3 6 | 0.0000 |
| 2 9 | 0.0024 | 3 7 | 0.0008 |

**Table 8.** Experimental Results for Differential Attack

### *4.5 Complexity of the Attack*

For differential cryptanalysis, we refer to the S-boxes involved in a characteristic which have a non-zero input difference (and hence a non-zero output difference) as *active* S-boxes. In general, the larger the differential probabilities of the active S-boxes, the larger the characteristic probability for the complete cipher. Also, the fewer active S-boxes, the larger the characteristic probability. As with linear cryptanalysis, we refer to the data required to mount the attack when considering the complexity of the cryptanalysis. That is, we assume that if we are able to acquire $N_D$ plaintexts, we are able to process them.

In general it is very complex to determine exactly the number of chosen plaintext pairs required to mount the attack. However, it can be shown that a good rule-of-thumb for the number of chosen plaintext pairs, $N_D$, required to distinguish right pairs when trying subkey candidates is

$$N_D \approx c \,/\, p_D \tag{7}$$

where $p_D$ is the differential characteristic probability for the $R$–1 rounds of the $R$-round cipher and $c$ is a small constant. Assuming that the occurrences of difference pairs in each active S-box are independent, the differential characteristic probability is given by

$$p_D = \prod_{i=1}^{\gamma} \beta_i \tag{8}$$

where the number of active S-boxes is represented by γ and the occurrence of the particular difference pair in the *i*-th active S-box of the characteristic has a probability represented by $\beta_i$.

It is not difficult to rationalize that (7) is true. It simply indicates that a few occurrences of the right pair are enough to give a count to the correct target partial subkey value that is significantly greater than the counts for the incorrect target partial subkey values. Since a right pair is expected to occur for about every $1/p_D$ pairs examined, in practice, it is generally reasonable to use some small multiple of $1/p_D$ chosen plaintext pairs to successfully mount the attack.

Approaches to providing resistance to differential cryptanalysis have focused on the S-box properties (i.e., minimizing the difference pair probability of an S-box) and finding structures to maximize the number of active S-boxes. Rijndael is a good example of a cipher designed to provide high resistance to differential cryptanalysis.

As with linear cryptanalysis, caution must be exercised in "proving" immunity to differential cryptanalysis. The computation of the differential characteristic probability is premised on the independence of the S-boxes involved in the approximation and in a real cipher, there is a dependence between the data entering different S-boxes. Hence, the probability $p_D$ is an estimate only. In practice, in many ciphers it has proven to be reasonably accurate.

Most importantly, different differential characteristics with the same input difference and output difference (i.e., the same differential) can combine to imply a probability for the differential that is larger than is implied by the consideration of one differential characteristic alone [17]. (This is analogous to the concept of linear hulls.) In order to prove security to differential cryptanalysis, it is necessary to prove that the probability of all differentials are below some acceptable threshold, not just that the probability of all differential characteristics are below some acceptable threshold. Generally, though, it is a reasonable assumption that, when a differential characteristic has a high probability, it dominates the occurrence of a differential and the probability of the characteristic gives a good approximation of the differential probability.

## 5. Advanced Concepts

Several extensions and modifications to the basic attacks of linear and differential cryptanalysis have been proposed and analyzed since the original presentation of the attacks. We do not present these advanced concepts and analyses in detail here, but encourage the reader to pursue these concepts further. Note the papers cited in this section represent a small sampling of work built on the two attacks.

We have mentioned, for example, the concepts of linear hulls [16] and differentials [17]: both concepts are integral to understanding the nature of provable security to the two cryptanalysis methods (an elusive goal!). There have been discussions of the similarities of concepts between the two attacks [18][19] and the analysis of the combination of the attacks into what is referred to as linear-differential cryptanalysis [20].

Several refinements to the cryptanalyses have attempted to improve the attacks for some circumstances. Truncated differential cryptanalysis [21] proposes the exploitation of differences at the cipher output where only some of the ciphertext bits have their differences predicted. Higher order differential cryptanalysis [21] attempts to exploit higher order differentials and is applicable to ciphers where the ciphertext bits are represented as functions of low nonlinear order. Impossible differential cryptanalysis [22] uses the non-existence of differences to derive cipher subkey bits, as opposed to the existence of highly likely differences that are exploited in normal differential cryptanalysis. Also, it should be noted that, in general, in differential cryptanalysis, differences need not be based on bit-wise exclusive-OR but may be a difference of another form, such as a difference as in the subtraction of one word of size $n$ from another modulo $2^n$ [17]. Extensions to linear cryptanalysis have included the utilization of multiple linear approximations [23] and the use of nonlinear approximations in combination with linear cryptanalysis [24].

Many papers in recent years have discussed the application of linear and differential cryptanalysis to ciphers proposed before the existence of the attacks was known. As well, many techniques in cipher design have been proposed to make the application of the attacks difficult, focusing on the constructions of cipher components such as S-boxes [25][8] and the interconnection between layers of S-boxes [8][26][27]. As a result, the attacks and their extensions are now very well understood and proposals such as Rijndael [7] have been especially constructed with security against the attacks in mind.

Finally, we note that our presentation of the attacks does not discuss the method for determining the best linear approximation and differential characteristic. However, this is discussed, for example, in [28].

## 6. Conclusion

In this paper, we have strived to present the fundamental concepts of linear and differential cryptanalysis as applied to a basic cipher. This cipher is a basic Substitution-Permutation Network and is not of a realistic scale to be used as a practical cipher. However, the structure is useful in examining the applicability of the attacks and this example cipher has formed the cornerstone for the explanation of the two attacks.

## 7. References


[1] M. Matsui, "Linear Cryptanalysis Method for DES Cipher", *Advances in Cryptology - EUROCRYPT '93 (Lecture Notes in Computer Science no. 765)*, Springer-Verlag, pp. 386-397, 1994.

[2] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems", *Journal of Cryptology*, vol. 4, no. 1, pp. 3-72, 1991.

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

[4] M. Matsui, "The First Experimental Cryptanalysis of the Data Encryption Standard", *Advances in Cryptology - CRYPTO '94 (Lecture Notes in Computer Science no. 839)*, Springer-Verlag, pp. 1-11, 1994.

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

[6] National Institute of Standards, Advanced Encryption Standard (AES) web site: www.nist.gov/aes.

[7] J. Daemen and V. Rjimen, "AES Proposal: Rijndael", *First Advanced Encryption Standard (AES) Conference*, California, Aug. 1998. (See also [6].)

[8] H.M. Heys and S.E. Tavares, "Substitution-Permutation Networks Resistant to Differential and Linear Cryptanalysis", *Journal of Cryptology*, vol. 9, no.1, pp. 1-19, 1996.

[9] L. Keliher, "Linear and Differential Cryptanalysis of SPNs", unpublished.

[10] L. Knudsen, "Block Ciphers: A Survey", *State of the Art in Applied Cryptography: Course on Computer Security and Industrial Cryptography (Lecture Notes in Computer Science no. 1528)*, Springer-Verlag, pp. 18-48, 1998.

[11] D.R. Stinson, *Cryptography: Theory and Practice*, CRC Press, 1995.

[12] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed., John Wiley & Sons, 1995.

[13] A. J. Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

[14] W. Stallings, *Cryptography and Network Security: Principles and Practices*, 2nd ed., Prentice Hall, 1999.

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

[16] K. Nyberg, "Linear Approximations of Block Ciphers", *Advances in Cryptology - EUROCRYPT '94 (Lecture Notes in Computer Science no. 950)*, Springer-Verlag, pp. 439-444, 1995.

[17] X. Lai, J.L. Massey, and S. Murphy, "Markov Ciphers and Differential Cryptanalysis", *Advances in Cryptology - EUROCRYPT '91 (Lecture Notes in Computer Science no. 547)*, Springer-Verlag, pp. 17-38, 1991.

[18] E. Biham, "On Matsui's Linear Cryptanalysis", *Advances in Cryptology - EUROCRYPT '94 (Lecture Notes in Computer Science no. 950)*, Springer-Verlag, pp. 341-355, 1995.

[19] F. Chabaud and S. Vaudenay, "Links Between Differential and Linear Cryptanalysis", *Advances in Cryptology - EUROCRYPT '94 (Lecture Notes in Computer Science no. 950)*, Springer-Verlag, pp. 356-365, 1995.

[20] M. Hellman and S. Langford, "Differential-Linear Cryptanalysis", *Advances in Cryptology - CRYPTO '94 (Lecture Notes in Computer Science no. 839),* Springer-Verlag, pp. 26-39, 1994.

[21] L.R. Knudsen, "Truncated and Higher Order Differentials", *Fast Software Encryption (Lecture Notes in Computer Science no. 1008)*, Springer-Verlag, pp. 196-211, 1995.

[22] E. Biham, A. Biryukov, and A. Shamir, " Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials", *Advances in Cryptology - EUROCRYPT '99 (Lecture Notes in Computer Science no. 1592),* Springer-Verlag, pp. 55-64, 1996.

[23] M.J.B. Robshaw and B.S. Kaliski, "Linear Cryptanalysis Using Multiple Approximations", *Advances in Cryptology - CRYPTO '94 (Lecture Notes in Computer Science no. 839),* Springer-Verlag, pp. 1-11, 1994.

[24] L. Knudsen and M.J.B. Robshaw, "Nonlinear Approximations in Linear Cryptanalysis", *Advances in Cryptology - EUROCRYPT '96 (Lecture Notes in Computer Science no. 1070),* Springer-Verlag, pp. 224-236, 1996.

[25] K. Nyberg, "Differentially Uniform Mappings for Cryptography, *Advances in Cryptology - EUROCRYPT '93 (Lecture Notes in Computer Science no. 765),* Springer-Verlag, pp. 55-64, 1994.

[26] E. De Win, A. Bosselaers, B. Preneel, J. Daemen, and V. Rijmen, "The Cipher SHARK", *Fast Software Encryption (Lecture Notes in Computer Science no. 1039)*, Springer-Verlag, pp. 99-112, 1996.

[27] A.M. Youssef, S. Mister, and S.E. Tavares, "On the Design of Linear Transformations for Substitution Permutation Encryption Networks", *Workshop on Selected Areas of Cryptography (SAC '96): Workshop Record*, pp. 40-48, 1997.

[28] M. Matsui, "On Correlation Between the Order of S-boxes and the Strength of DES", *Advances in Cryptology - EUROCRYPT '94 (Lecture Notes in Computer Science no. 950),* Springer-Verlag, pp. 366-375, 1995.