December 2001

A New Cryptanalytic Method Using the Distribution Characteristics of Substitution Distances

B. Song
University of Wollongong

H. Wang
University of Wollongong, huaxiong@uow.edu.au

Jennifer Seberry
University of Wollongong, jennie@uow.edu.au

A New Cryptanalytic Method Using the Distribution Characteristics of Substitution Distances

Abstract

In this paper, we suggest a new method for cryptanalysis of the basic structures of the block ciphers having SP network structure. The concept of the substitution difference is introduced and the distribution characteristics of substitution distances in an S-box is developed. This gives clues for cryptanalysis of the cipher. We then examine if this method is applicable to cryptanalysis of Rijndael. We present the method for cryptanalysis of the first round of Rijndael including the initial Round-Key addition part in order to illustrate our new method.

Keywords

Cryptanalysis, Substitution-Permutation(SP) Networks, Substitution Distance, Rijndael.

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was orignally published as: Song, B, Wang, H & Seberry, J, A New Cryptanalytic Method Using the Distribution Characteristics of Substitution Distances, Information Security and Cryptology - ICISC'2001, 6-7 December 2001, Seoul,Korea, 2001. Lecture Notes in Computer Science, Vol 2288, Springer-Verlag, Berlin, 2001, 277-286. The original publication is available here through Springerlink.

A New Cryptanalytic Method Using the Distribution Characteristics of Substitution Distances

Beomsik Song, Huaxiong Wang, Jennifer Seberry

Centre for Computer Security Research
School of Information Technology and Computer Science
University of Wollongong
Wollongong 2522, AUSTRALIA
{bs81, huaxiong, jennifer_seberry}@uow.edu.au

Abstract. In this paper, we suggest a new method for cryptanalysis of the basic structures of the block ciphers having SP network structure. The concept of the substitution difference is introduced and the distribution characteristics of substitution distances in an S-box is developed. This gives clues for cryptanalysis of the cipher. We then examine if this method is applicable to cryptanalysis of $Rijndael$. We present the method for cryptanalysis of the first round of $Rijndael$ including the initial Round-Key addition part in order to illustrate our new method.

Key words: Cryptanalysis, Substitution-Permutation(SP) Networks, Subtitution Distance, Rijndael.

1 Introduction

Cryptanalytic research has been used to promote the development of more secure ciphers. Some powerful cryptanalytic tools including DC ($Differential\ Cryptanalysis$) [3, 4] and LC ($Linear\ Cryptanalysis$) [7] can be said to have played an important role in strengthening the design criteria for block ciphers. For this reason, we assert that these cryptanalytic methods have brought about the design of new block ciphers believed to be more secure than DES ($Data\ Encryption\ Standard$) [2]. However, it does not be necessarily follow that because these newer ciphers are designed to resist the existing cryptanalytic methods, they are permanently secure. This is because cryptanalytic methods are developing as well.

The purpose of this paper is to suggest a new method for cryptanalysis of the basic structures of the block ciphers having SP network structure. To do this, some basic concepts are defined to be used to establish the main ideas, and the main concept of the Substitution Distance is established on the basis of these basic concepts. And after that, the method of applying this concept to cryptanalysis is introduced. This method consists of two parts, such as finding

possible keys with a known plaintext and the determination of the keys actually used among those possible keys with some additional chosen (or known) plaintexts. Especially, this method is focused on how to choose plaintexts which are the most likely to select the keys actually used among possible keys.

We find out that suitable input differences between the known plaintext used for finding possible keys and additional chosen (or known) plaintexts used for the determination of the keys actually used can be obtained from the distribution characteristics of the values of substitution distances in the S-box of the cipher algorithm. This means that if we know the substitution distances in the S-box of a given cipher algorithm, in cryptanalysis of the cipher algorithm we can choose the plaintexts having the suitable input differences with the plaintext used for finding possible keys. It does not depend on the keys.

In the last part of this paper, we introduce the method for cryptanalysis of the first round of the AES algorithm ($Rijndael$) including the initial Round-Key addition part in order to briefly illustrate this cryptanalytic method. The new method we establish is unlike previous cryptanalytic methods on $Rijndael$ [5,6].

2 Preliminaries

2.1 Definitions

Definition 1 (Sequence Difference, Input Difference, Input-Output Difference). *The sequence difference is defined as the bitwise XOR of two binary sequence X and X'. If X and X' are input sequences of an algorithm, we say $X \oplus X'$ is the input difference denoted by ΔX. If X is an input sequence of an algorithm, and Y is the corresponding output sequence, then we say that $X \oplus Y$ is the input-output difference denoted by $IOD(X, Y)$.*

Definition 2 (Substitution Difference). *Let a sequence X be an input sequence of a substitution table (S-box) or function and the sequence $S(X)$ be the corresponding output sequence of the S-box. Then we say that the XOR of X and $S(X)$ is the substitution difference between X and $S(X)$ denoted by $SD(X, S(X))$. This means that*

$$SD(X, S(X)) = X \oplus S(X) = (x_0, x_1, x_2, \cdots, x_n) \oplus (s_0, s_1, s_2, \cdots, s_{n-1}).$$

Example 1. Suppose that substitution of each 3 bits is given by $(0\ 0\ 0 \rightarrow 1\ 1\ 0), (0\ 0\ 1 \rightarrow 0\ 1\ 0), (0\ 1\ 0 \rightarrow 1\ 1\ 1), (0\ 1\ 1 \rightarrow 0\ 0\ 0), (1\ 0\ 0 \rightarrow 1\ 0\ 1), (1\ 0\ 1 \rightarrow 0\ 0\ 1), (1\ 1\ 0 \rightarrow 1\ 0\ 0), (1\ 1\ 1 \rightarrow 0\ 1\ 1)$. Then we say that the substitution difference $SD((1\ 1\ 0), (1\ 0\ 0))$ between the input sequence $X = (1\ 1\ 0)$ and the corresponding output sequence $S(X) = (1\ 0\ 0)$ is

$$(1\ 1\ 0) \oplus (1\ 0\ 0) = (0\ 1\ 0).$$

The substitution differences are

$$(1\ 1\ 0), (0\ 1\ 1), (1\ 0\ 1), (0\ 1\ 1), (0\ 0\ 1), (1\ 0\ 0), (0\ 1\ 0), \text{and } (1\ 0\ 0) \qquad (1)$$

in turn.

Definition 3 (Substitution Distance). *Let*

$$SD(X, S(X)) = X \oplus S(X) = (d_0, d_1, d_2, \cdots, d_{n-1})$$

be a substitution difference between the input sequence X and the corresponding output sequence $S(X)$. And let

$$SD(X', S(X')) = X' \oplus S(X') = (d'_0, d'_1, d'_2, \cdots, d'_{n-1})$$

be another substitution difference between the input sequence X' and the corresponding output sequence $S(X')$. Then the substitution distance between $SD(X, S(X))$ and $SD(X', S(X'))$ is defined as

$$SD(X, S(X)) \oplus SD(X', S(X'))$$

and denoted by

$$SDT[SD(X, S(X)),\ SD(X', S(X'))].$$

This means that

$$SDT[SD(X, S(X)), SD(X', S(X'))] = X \oplus S(X) \oplus X' \oplus S(X')$$
$$= (d_0, d_1, d_2, \cdots, d_{n-1}) \oplus (d'_0, d'_1, d'_2, \cdots, d'_{n-1}).$$

Example 2. Suppose that substitution differences are described as (1). Then we say that the substitution distance $DST[(1\ 1\ 0), (0\ 1\ 1)]$ between the first substitution difference (1 1 0) and the second substitution difference (0 1 1) is (1 0 1). The substitution distances between the first substitution difference and other substitution differences are

$$(1\ 0\ 1), (0\ 1\ 1), (1\ 0\ 1), (1\ 1\ 1), (0\ 1\ 0), (1\ 0\ 0), \text{and } (0\ 1\ 0)$$

in turn.

2.2 Notation

Notations are defined as follows:

$X = (x_0, x_1, x_2, \cdots, x_{n-1})$: a binary sequence.
$X' = (x'_0, x'_1, x'_2, \cdots, x'_{n-1})$: another binary sequence.
$\Delta X = X \oplus X'$: the input difference between X and X'.
$S(X) = (s_0, s_1, s_2, \cdots, s_{n-1})$: the output sequence, of the S-box, corresponding to X.
$S(X') = (s'_0, s'_1, s'_2, \cdots, s'_{n-1})$: the output sequence, of the S-box, corresponding to X'.
$SD(X, S(X)) = X \oplus S(X)$: the substitution difference between X and $S(X)$.
$SD(X', S(X')) = X' \oplus S(X')$: the substitution difference between X' and $S(X')$.
$SDT[SD(X, S(X)), SD(X', S(X'))]$: the substitution distance between $SD(X, S(X))$ and $SD(X', S(X'))$.
$Y = (y_0, y_1, y_2, \cdots, y_{n-1})$: the output sequence, of the algorithm, corresponding to X.
$Y' = (y'_0, y'_1, y'_2, \cdots, y'_{n-1})$: the output sequence, of the algorithm, corresponding to X'.
$IOD(X, Y)$: the input-output difference between X and Y.

2.3 Relationships Between the Basic Concepts

In order to describe the *Substitution-Distance Cryptanalysis* clearly, the relationships between the basic concepts described above are shown in Figure 1 on the basic structure of the SP-network-structure block cipher algorithm. This structure will be used throughout this paper.

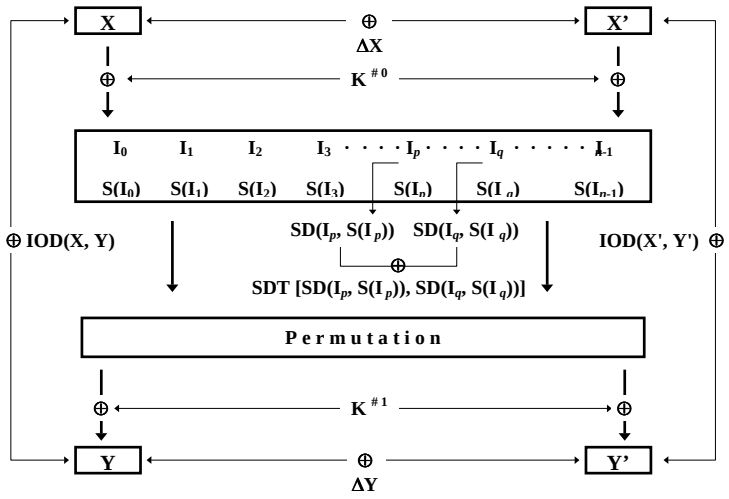

Fig. 1. The factors influencing the input-output difference.

3 Principles of Substitution-Distance Cryptanalysis

3.1 Requirements for Substitution-Distance Cryptanalysis

In this paper, we call the structures like Figure 1 the basic structures of the SP-network-structure block cipher algorithms. The purpose of *Substitution-Distance Cryptanalysis* is to analyse the basic structures of SP-network-structure block cipher algorithms. In other words, it is the purpose of *Substitution-Distance Cryptanalysis* to find the keys used in the basic structures of SP-network block cipher algorithms consisting of the substitution, the permutation and the addition of round keys. In order to do this, a known pair of a plaintext and the corresponding ciphertext are used for finding possible keys, and some other chosen (or known) pairs of a plaintext and the corresponding ciphertext are used for the determination of the keys actually used among those possible keys. The number of pairs of a plaintext and the corresponding ciphertext for the finding of keys actually used depends on the distribution characterics of the values of the substitution distances in the S-box of the given cipher. Of course, carefully chosen pairs of a plaintext and the corresponding ciphertext will be more helpful than known pairs of a plaintext and the corresponding ciphertext to lighten the difficulty of cryptanalysis.

3.2 Principle of Substitution-Distance Cryptanalysis

The principle of *Substitution-Distance Cryptanalysis* consists of two parts, which are the finding of all possible keys with a known plaintext, and the determination of the keys actually used among those possible keys with some additional known (or chosen) plaintexts. We describe how to choose plaintexts having the suitable input differences with the plaintext used to find the possible keys. For the purpose of this principle, we ignore the step of the permutation because it has no effect on this cryptanalytic method.

Finding the Possible Keys. For a given algorithm like Figure 1, if a pair of a plaintext X and the corresponding ciphertext Y is known, we can find all possible key pairs of $K_i^{\#0}$ and $K_i^{\#1}$ ($0 \leq i < n$, n is the number of the substitution paths in the S-box) to transform X into Y by the following method. Let the number of the substitution paths in the S-box of the algorithm be n, an input be X, and the corresponding output be Y. Then, if we suppose that the substitution path $(I_e, S(I_e))$ in the S-box is used for the encryption, we can make the formulas

$$I_e = X \oplus K_e^{\#0} \tag{2}$$

and

$$Y = S(I_e) \oplus K_e^{\#1}. \tag{3}$$

Also, because we know the values of X, Y, I_e, and $S(I_e)$, we can obtain $K_e^{\#0}$ and $K_e^{\#1}$ from (2) and (3). Now, if we consider all possible substitution paths

from $(I_0, S(I_0))$ to $(I_{n-1}, S(I_{n-1}))$, we can obtain all possible key pairs of $K_i^{\#0}$ and $K_i^{\#1}$ $(0 \leq i < n)$ to transform X into Y. The number of all possible key pairs of $K_i^{\#0}$ and $K_i^{\#1}$ will be equal to the number of substitution paths in the S-box if the S-box uses a one-to-one correspondence. For this reason, we can describe the set of all possible key pairs

$$K_{set} = \{(K_i^{\#0}, K_i^{\#1}) | I_i = X \oplus K_i^{\#0},\ Y = S(I_i) \oplus K_i^{\#1}\},$$

where $S_{set} = \{(I_i, S(I_i)) | 0 \leq i < n\}$. The cardinality of the K_{set} is equal to that of S_{set}, that is

$$|K_{set}| = |S_{set}|.$$

Determination of the Keys Actually Used. Now, we have to select the key pair actually used of $K_r^{\#0}$ and $K_r^{\#1}$ among the elements of K_{set} of the possible key pairs. In order to do this we need more clues. One method may be to use another known pair of a plaintext and the corresponding ciphertext. This is because if we know another pair of a plaintext X' and the corresponding ciphertext Y', we can obtain not only another set of possible key pairs

$$K'_{set} = \{(K_i'^{\#0}, K_i'^{\#1}) | I_i = X' \oplus K_i'^{\#0},\ Y' = S(I_i) \oplus K_i'^{\#1}\}$$

but also the intersection

$$PK_{set} = \{(K_j^{\#0}, K_j^{\#1}) | (K_j^{\#0}, K_j^{\#1}) \in K_{set} \cap K'_{set}\}$$

of sets K_{set} and K'_{set}. It is obvious that $|PK_{set}| \leq |K_{set}|$ and $|PK_{set}| \leq |K'_{set}|$. This means that the number of possible key pairs can be reduced by the consideration of another known pair of a plaintext and the corresponding ciphertext. However, because the number of elements of PK_{set} must be at least 2 by Theorem 1, that is $|PK_{set}| \geq 2$, we can not perfectly determine the key pair actually used even now. This is because we can only select the key pair actually used when the cardinality of PK_{set} is 1.

Theorem 1. *For the basic structure of the SP-network-structure block cipher algorithm, let X and X' be encrypted with the key pair $K_r^{\#0}$ and $K_r^{\#1}$, giving outputs Y and Y' respectively. Then there must be another key pair $K_r'^{\#0}$ and $K_r'^{\#1}$ which gives the same output.*

Proof. Let the set consisting of all n substitution paths in the S-box be S_{set}, that is,

$$S_{set} = \{(I_i, S(I_i)) | 0 \leq i < n\}.$$

Then the set consisting of all possible key pairs to transform X into Y can be described by

$$K_{set} = \{(K_i^{\#0},\ K_i^{\#1}) | I_i = X \oplus K_i^{\#0},\ Y = S(I_i) \oplus K_i^{\#1}\},$$

and another set consisting of all possible key pairs to transform X' into Y' can be described by

$$K'_{set} = \{(K_i'^{\#0},\ K_i'^{\#1}) | I_i = X' \oplus K_i'^{\#0},\ Y' = S(I_i) \oplus K_i'^{\#1}\}.$$

Now, let the key pair $(K_r^{\#0}, K_r^{\#1})$ among the elements of K_{set} be the key pair actually used to transform X into Y and X' into Y'. Then the path $(I_r, S(I_r))$ and the path $(I_r \oplus \Delta X, S(I_r \oplus \Delta X))$ in the S-box have to be separately used (ΔX is an input difference between X and $X' : \Delta X = X \oplus X'$). On the other hand, it is obvious that among the elements of K'_{set} there is already the key pair $(K_r'^{\#0}, K_r'^{\#1})$ to transform X' into Y' with the substitution path $(I_r, S(I_r))$. This key pair can also transform the input X into the output Y with the substitution path $(I_r \oplus \Delta X, S(I_r \oplus \Delta X))$. After all, both of these two key pairs $(K_r^{\#0}, K_r^{\#1})$ and $(K_r'^{\#0}, K_r'^{\#1})$ can transform not only X into Y, but also X' into Y'.

For this reason, it is necessary to use at least two known plaintexts to determine the key pair actually used among the possible key pairs to transform X into Y. In fact, in many cases, we may need many known plaintexts for the selection of the key pair actually used because we have to obtain the sets of K'_{set}, $K''_{set} \cdots$ to make the cardinality of the intersection $PK_{set}(K_{set} \cap K'_{set} \cap K''_{set} \cdots) = 1$.

However, it is not easy even to collect enough pairs of a plaintext and the corresponding ciphertext. Because of this, in order to minimize the effort in the determination of the key pair actually used we need to carefully choose the plaintexts, so that the cardinality of PKset becomes 1 easily. Theorem 2 shows the relation between X', $|K_{set}|$, $|K'_{set}|$ and $|PK_{set}| = |K_{set} \cap K'_{set}|$.

Theorem 2. *For the basic structure of the SP-network-structure block cipher algorithm, let X be a plaintext, Y be the corresponding ciphertext,*

$$S_{set} = \{(I_i, S(I_i)) | 0 \le i < n\}$$

be a set consisting of all n substitution paths in the S-box, and

$$K_{set} = \{(K_i^{\#0}, K_i^{\#1}) | I_i = X \oplus K_i^{\#0},\ Y = S(I_i) \oplus K_i^{\#1}\}$$

be the set consisting of all possible key pairs to transform X into Y. And let another plaintext be X' and the corresponding ciphertext by each key pair in K_{set} be $Y'_0, Y'_1, Y'_2, \cdots, Y'_{n-1}$ respectively. Then $|PK_{set}| = |K_{set} \cap K'_{setp}|$ is equal to the number of key pairs in K_{set} to transform X' into Y'_p ,where

$$K'_{setp} = \{(K_i'^{\#0}, K_i'^{\#1}) | I_i = X' \oplus K_i'^{\#0},\ Y'_p = S(I_i) \oplus K_i'^{\#1},\ \textit{and}\ 0 \le i < n\}.$$

Proof. Trivial by the definition of intersection.

As shown in Theorem 2, in order to make $|PK_{set}| = |K_{set} \cap K'_{set}|$ as small as possible, we need to use plaintexts each of which can be transformed into diverse ciphertexts by key pairs in K_{set} when we select the key pair actually used among the possible key pairs. In other words, we need to use plaintexts

that may bring well distributed ciphertexts by key pairs in K_{set} (ciphertexts which are not concentrated on a certain value). These plaintexts can bring the $K'_{set}, K''_{set}, K'''_{set}, \cdots$ to make $|PK_{set}|$ as small as possible. The next step is to find how to obtain these plaintexts each of which can be transformed into well distributed ciphertexts by the key pairs in K_{set}. Even if these plaintexts can be obtained from the direct computation for all plaintexts with the key pairs in K'_{set}, we need to establish a more effective method of finding these plaintexts. Theorem 6 shows which plaintexts can be transformed into well distributed ciphertexts. This idea is based on the relation between the input differences of the S-box and the distribution characteristics of the substitution distances in the S-box of a given cipher algorithm. Theorems 3, 4, and 5 enable Theorem 6 to be established.

Theorem 3. *For the basic structure of the SP-network-structure block cipher algorithm, let X be a plaintext, Y be the corresponding ciphertext,*

$$S_{set} = \{(I_i, S(I_i)) | 0 \leq i < n\}$$

be the set consisting of all n substitution paths in the S-box, and

$$K_{set} = \{(K_i^{\#0}, K_i^{\#1}) | I_i = X \oplus K_i^{\#0},\ Y = S(I_i) \oplus K_i^{\#1}\}$$

be the set consisting of all possible key pairs to transform X into Y. And let another plaintext,having the input difference ΔX with X, be X' and the corresponding ciphertexts for each key pair in K_{set} be $Y'_0, Y'_1, Y'_2, \cdots, Y'_{n-1}$ respectively. Then

$$Y'_p = X' \oplus SD(I_p \oplus \Delta X, S(I_p \oplus \Delta X)) \oplus (K_p^{\#0}, K_p^{\#1}).$$

Proof. Because Input-Output Difference $IOD(X, Y)$ = Substitution Difference $SD(I, S(I)) \oplus$ Key Value $(K^{\#0}, K^{\#1})$ in this algorithm, we can describe

$$Y = X \oplus SD(I_p, S(I_p)) \oplus (K_p^{\#0}, K_p^{\#1}).$$

If there is another plaintext X' having the input difference ΔX with X, then $X' = X \oplus \Delta X$. Here, because X passes through the substitution path $(I_p, S(I_p))$ and is transformed into Y by XOR with a key pair $(K_p^{\#0}, K_p^{\#1})$, $X' (= X \oplus \Delta X)$ has to pass through the substitution path $(I_p \oplus \Delta X, S(I_p \oplus \Delta X))$ and Y'_p is equal to $X' \oplus SD(I_p \oplus \Delta X, S(I_p \oplus \Delta X)) \oplus (K_p^{\#0}, K_p^{\#1})$ by XOR with the same key pair $(K_p^{\#0}, K_p^{\#1})$.

Theorem 4. *In Theorem 3,*

$$Y'_p = X' \oplus SD(I_p, S(I_p)) \oplus SDT[SD(I_p, S(I_p)), SD(I_p \oplus \Delta X, S(I_p \oplus \Delta X))] \oplus (K_p^{\#0}, K_p^{\#1}).$$

Proof. As $SDT[SD(I_p, S(I_p)), SD(I_p \oplus \Delta X, S(I_p \oplus \Delta X))] = SD(I_p, S(I_p)) \oplus SD(I_p \oplus \Delta X, S(I_p \oplus \Delta X))$ by Definition 3, it is trivial.

Theorem 5. *For the basic structure of the SP-network-structure block cipher algorithm, let X be a plaintext, Y be the corresponding ciphertext,*

$$S_{set} = \{(I_i, S(I_i)) | 0 \leq i < n\}$$

be the set consisting of all n substitution paths in the S-box, and

$$K_{set} = \{(K_i^{\#0}, K_i^{\#1}) | I_i = X \oplus K_i^{\#0},\ Y = S(I_i) \oplus K_i^{\#1}\}$$

be the set consisting of all possible key pairs to transform X into Y. And let another plaintext having the input difference ΔX with X be X' and the corresponding ciphertexts by each key pair in K_{set} be $Y'_0, Y'_1, Y'_2, \cdots, Y'_{n-1}$ respectively. Then, if

$$\begin{aligned} &SDT[SD(I_p, S(I_p)), SD(I_p \oplus \Delta X, S(I_p \oplus \Delta X))] \\ &= SDT[SD(I_q, S(I_q)), SD(I_q \oplus \Delta X, S(I_q \oplus \Delta X))], \end{aligned}$$

$$Y'_p = Y'_q.$$

Proof. By Theorem 4

$$\begin{aligned} Y'_p = X' &\oplus SD(I_p, S(I_p)) \oplus SDT[SD(I_p, S(I_p)), SD(I_p \oplus \Delta X, S(I_p \oplus \Delta X))] \\ &\oplus (K_p^{\#0}, K_p^{\#1}) \end{aligned}$$

and

$$\begin{aligned} Y'_q = X' &\oplus SD(I_q, S(I_q)) \oplus SDT[SD(I_q, S(I_q)), SD(I_q \oplus \Delta X, S(I_q \oplus \Delta X))] \\ &\oplus (K_q^{\#0}, K_q^{\#1}). \end{aligned}$$

Also, $SD(I_p, S(I_p)) \oplus (K_p^{\#0}, K_p^{\#1}) = SD(I_q, S(I_q)) \oplus (K_q^{\#0}, K_q^{\#1})$ in the above two formulas because both parts are the input-output difference between X and Y. Therefore, if

$$\begin{aligned} &SDT[SD(I_p, S(I_p), SD(I_p \oplus \Delta X, S(I_p \oplus \Delta X))] \\ &= SDT[SD(I_q, S(I_q)), SD(I_q \oplus \Delta X, S(I_q \oplus \Delta X))], \end{aligned}$$

$$Y'_p = Y'_q.$$

Theorem 6. *For the basic structure of the SP-network-structure block cipher algorithm, let X be a plaintext, Y be the corresponding ciphertext,*

$$S_{set} = \{(I_i, S(I_i)) | 0 \leq i < n\}$$

be the set consisting of all n substitution paths in the S-box, and

$$K_{set} = \{(K_i^{\#0}, K_i^{\#1}) | I_i = X \oplus K_i^{\#0},\ Y = S(I_i) \oplus K_i^{\#1}\}$$

be the set consisting of all possible key pairs to transform X into Y. And let another plaintext, having the input difference ΔX with X, be X' and the corresponding ciphertexts by each key pair in K_{set} be $Y'_0, Y'_1, Y'_2, \cdots, Y'_{n-1}$ respectively. Then, the distribution characteristics of the ciphertexts $Y'_0, Y'_1, Y'_2, \cdots, Y'_{n-1}$ is equivalent to that of

$$SDT[SD(I_i, S(I_i)),\ SD(I_i \oplus \Delta X, S(I_i \oplus \Delta X))]s, where\ 0 \leq i < n.$$

Proof. Trivial by Theorem 5.

As shown in Theorem 3, 4, 5 and 6, if we know ΔXs each of which makes the substitution distances $SDT[SD(I_i, S(I_i)), SD(I_i \oplus \Delta X, S(I_i \oplus \Delta X)]$s in a S-box as well distributed as possible, where $0 \leq i < n$, we also obtain the plaintexts each of which can be transformed into well distributed ciphertexts for all key pairs in K_{set} . Because it does not depend on keys to choose ΔXs ,we can obtain these off-line without any restriction of time. Hence, we can know which plaintexts have to be additionally chosen in the determination of the key pair really used, when we have a pair of a plaintext and the corresponding ciphertext used for finding the possible key pairs.

This means that we can choose the plaintexts to make $|PK_{set}| = |K_{set} \cap K'_{set}|$ smallest by the distribution characteristics of substitution distances, and reduce the number of additional plaintexts that we need for the selection of the key pair actually used. Figure 2 is a pseudo code showing the above idea of selecting ΔXs in the S-box substituting 8-bit inputs into 8-bit outputs.

```
for(ΔX=0x01; ΔX≤0xff; ΔX++)   {
    for (t=0; t<256 ;t++ ) count[t]=0x00;
    for (i=0; i<256 ;i++ ){
      t=SDT [SD(Ii,S(Ii)), SD((Ii⊕ΔX), S(Ii⊕ΔX)];
      count[t]= count[t]+1;
                          }
     value[ΔX]=the greatest count[t];
                                  }
arrange ΔXs in increasing order of value[ ];

select the ΔX to make the number of  value[ΔX]
smallest;
```

Fig. 2. Finding the most suitable input differences ΔXs.

4 Applying to *Rijndael* in Standard Case

4.1 Outline of *Rijndael* in Standard Case

As figure 3 shows, *Rijndael* has SP network structure and processes data blocks of 128bits in the standard case [1, 5] . The bytewise substitution by the ByteSub transformation , the cyclic shift of the four bytes in each row by the ShiftRow transformation, and the mix of the four bytes in each column by the MixColumn transformation are performed every round. After these operations, a 128-bit round key extended from an user key is XORed in the last part of every round. The MixColumn transformation is omitted in the last round, but before the first round a 128-bit round key is XORed.

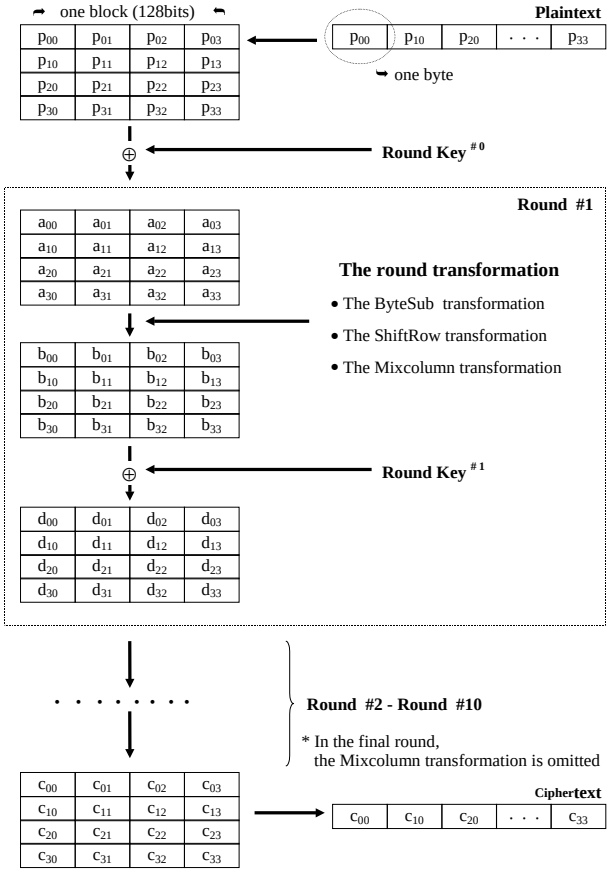

Fig. 3. The structure of *Rijndael* in standard case.

4.2 Structure of the First Round of *Rijndael*

Figure 4 is showing the first round of *Rijndael* including the initial Round-Key addition part. Because the ShiftRow transformation in *Rijndael* has no effect on the cryptanalytic method suggested on this paper, that operation is omitted in Figure 4.

4.3 Finding *Round Key*$^{\#0}$ and *Round Key*$^{\#1}$

Preparations. To find *Round Key*$^{\#0}$ and *Round Key*$^{\#1}$ start with finding the substitution differences in the S-box. In the case of *Rijndael*, the S-box has a non-linear byte substitution. The S-box consists of 256 substitution paths, and this S-box substitutes the value of each input byte into new value (eg : $0x46 \rightarrow 0x5a$) every round through the ByteSub transformation. However, after the substitution, because all four bytes in each column are mixed affecting each

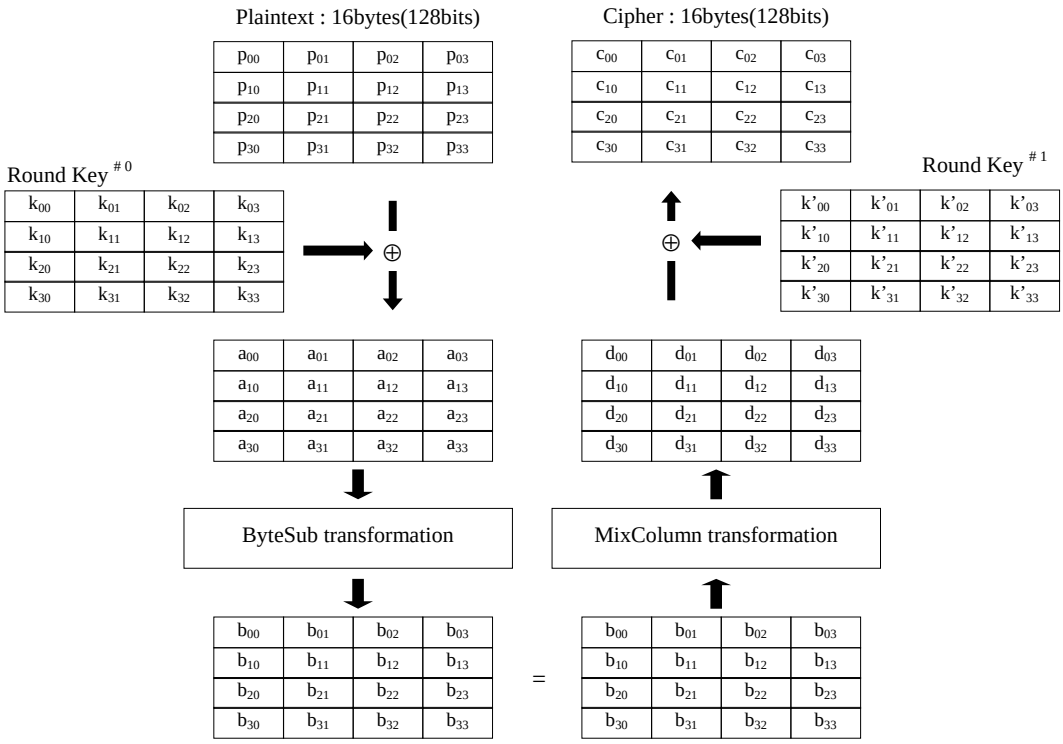

Fig. 4. The first round of *Rijndael* including the initial Round-Key addition.

other by the matrix formula in Figure 5 through the MixColumn transformation, all four bytes in each column need to be considered together in finding the substitution characteristics of the S-box.

$$\begin{pmatrix} d_{0c} \\ d_{1c} \\ d_{2c} \\ d_{3c} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} b_{0c} \\ b_{1c} \\ b_{2c} \\ b_{3c} \end{pmatrix}$$

Fig. 5. The MixColumn transformation of *Rijndael*.

In other words, we need to modify the round transformation properly to obtain the four byte substitution differences and the substitution distances between these substitution differences. Because we know both the subtitution differences in the S-box used in the ByteSub transformation and the matrix

formula used in the MixColumn transformation, we can obtain the outputs of these operations for all sorts of four byte inputs from $(0x00, 0x00, 0x00, 0x00)$ to $(0xff, 0xff, 0xff, 0xff)$. Now, the new S-box having all $(256)^4$ substitution paths can be obtained. This new S-box can replace the above two operations because the substitution effect of this new S-box and the substitution effect of the above two operation is the same for all four byte inputs. As shown above, it will be helpful in the application of *Substitution-Distance Cryptanalysis* to change complicated round transformations in a given algorithm into a simple S-box. The next step is to find the ΔXs to obtain the well distributed values of the substitution distances in the newly generated S-box. This is because if we know these ΔXs, by Theorem 6 we can choose the plaintexts to be used for the determination of the key pair actually used when we obtain the possible key pairs with a known plaintext.

Figure 6 is a pseudo code to describe this step briefly.

```
for(ΔX=0x00000001; ΔX≤0xffffffff; ΔX++){
   for (t=0; t<256^4 ;t++ ) count[t]=0x00000000;
   for (i=0; i<256^4 ;i++ ){
      t=SDT [SD(I_i,S(I_i)), SD((I_i⊕ΔX), S(I_i⊕ΔX)];
      count[t]= count[t]+1;
                        }
   value[ΔX]=the greatest count[t];
                                          }
arrange ΔXs in increasing order of value[ ];
select the ΔX to make the number of  value[ΔX]
smallest;
```

Fig. 6. Finding the most suitable input differences ΔXs in *Rijndael*.

Finding Possible Key Pairs of *Round Key*$^{\#0}$ and *Round Key*$^{\#1}$. In order to simplify the discussion, the method to find the first column (4 bytes) in *Round Key*$^{\#0}$ and the first column (4 bytes) in *Round Key*$^{\#1}$ is described first. In other words, the method of finding $(k_{00}, k_{10}, k_{20}, k_{30})$ and $(k'_{00}, k'_{10}, k'_{20}, k'_{30})$ is explained. Let a pair comprising $X = (p_{00}, p_{10}, p_{20}, p_{30})$ and the corresponding ciphertext $Y = (c_{00}, c_{10}, c_{20}, c_{30})$ be known as in Figure 4. Then, by the method referred in 3.2.1 we can obtain all $(256)^4$ possible key pairs of $(k_{00}, k_{10}, k_{20}, k_{30})$ and $(k'_{00}, k'_{10}, k'_{20}, k'_{30})$ by cinsidering all $(256)^4$ possible substitution paths from I_i to $S(I_i)$ (I_i is $0x00000000$ to $0xffffffff$). The result is the set

$$K_{set} = \{(K_i^{\#0},\ K_i^{\#1}) | I_i = X \oplus K_i^{\#0},\ Y = S(I_i) \oplus K_i^{\#1} \text{and } 0 \le i < 256^4\}.$$

Determining the Key Pairs Actually Used of *Round Key*$^{\#0}$ and *Round Key*$^{\#1}$. As mentioned above, we can find the set K_{set} consisting of all possible key pairs of the first column in *Round Key*$^{\#0}$and the first column in *Round Key*$^{\#1}$. Now, we have to determine the key pair actually used among those 256^4 key pairs. In order to do this, we need to use some more known (or chosen) plaintexts. This is because if we know these plaintexts we can select the key pair actually used from the intersection of $K_{set}, K'_{set}, K''_{set} \cdots$ (these sets can be made by the step of finding all possible keys). If we can obtain known plaintexts having the input differences ΔXs which yield the well distributed values of

$$SDT[SD(I_i, S(I_i)), SD(I_i \oplus \Delta X, S(I_i \oplus \Delta X))]\text{s, where } 0 \leq i < 256^4,$$

we can determine the key pair actually used with fewer plaintexts by Theorems 2 and 6. In this case, if we have plaintexts, which ensure $|K_{set} \cap K'_{set} \cap K''_{set} \cdots|$=1, we may determine the key pair actually used. On the other hand, if we do not have known plaintexts satisfing the above condition, we need to obtain chosen plaintexts, which have the input differences ΔXs with the known plaintext used for finding possible key pairs. These plaintexts are just the plaintexts to yield the well distributed values of

$$SDT[SD(I_i, S(I_i)), SD(I_i \oplus \Delta X, S(I_i \oplus \Delta X))]\text{s, where } 0 \leq i < 256^4.$$

If we know the most suitable ΔXs from the step of the Preparation, we can choose the plaintexts. And then, we can finally find the key pair actually used by the intersection of $K_{set} \cap K'_{set} \cap K''_{set} \cdots$ when $|K_{set} \cap K'_{set} \cap K''_{set} \cdots| = 1$. Remember that the plaintexts having the suitable input differences ΔXs with the plaintext X used for finding possible key pairs have to be chosen. Otherwise, we may need many more chosen plaintexts than minium. At the moment, because we have not tried to look over the substitution distances and to inspect the suitable input differences ΔXs, we can not know exactly the number of plaintexts we need for the selection of the key pair actually used. However, if we find out these characteristics, we may know the number of plaintexts actually needed

Applying to All Columns. As shown in 4.3.2 and 4.3.3, we can select the key pairs actually used in the first columns in *Round Key*$^{\#0}$ and *Round Key*$^{\#1}$. By the application of the same method to the second column, the third column and the last column, we can find the all columns in *Round Key*$^{\#0}$ and *Round Key*$^{\#1}$ (256 bytes). This means that we can find the *Round Key*$^{\#0}$ and *Round Key*$^{\#1}$ by considering 4×256^4 possible key pairs of 4 bytes.

5 Conclusions

We have shown that the Substitution Distances between the substitution paths in the S-box can be used as the clues for cryptanalysis of the block ciphers having SP network structure. Specifically, we show how to find all possible keys with a known plaintext on the basic structure of the SP network block cipher

algorithms. We explain how to select the key actually used from these possible keys by the use of the distribution characteristics of the values of the substitution distances in the S-box of the algorithm. In other words, we show which input differences ΔXs are the most suitable in the selection of the plaintexts to be used for the determination of the keys actually used. Along with this, we introduce the possibility of applying these principles to cryptananalysis of the first round of *Rijndael* in order to briefly illustrate this cryptanalytic method. As a result of our efforts, we find out that *Substitution-Distance Cryptanalysis* can be apply for cryptanalysis of the basic structure of SP-network-structure block cipher algorithms. Remember that if we know the most suitable input differences (ΔXs : they do not depend on keys), we may find the keys used for the encryption on the basic round of the SP-network-structure block cipher algorithms. For this reason, it is useful for the further study to find suitable ΔXs for the SP-network-structure block cipher algorithms including *Rijndael*.

References

1. "Advanced Encryption Standard(*AES*)", FIPS-Pub. ZZZ, NIST, http://csrc.nist.gov/publications/drafts/dfips-AES.pdf, 2001.
2. "Data Encryption Standard", FIPS 46-2, NIST, 1993.
3. E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems", *J. Cryptology*, Vol.4, pp.3-72, 1991.
4. E. Biham and A. Shamir, "Differential cryptanalysis of the full 16-round DES", *Advances in Cryptology-Crypto'92*, Lecture Notes in Computer Science, Springer-Verlag, pp.487-496, 1992.
5. J. Daemen and V. Rijmen, "*AES* Proposal: Rijndael", http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf, 1999.
6. N. Ferguson, et al, "Improved Cryptanalysis of Rijndael", *the Fast Software Encryption Workshop '2000*, Preproceeding, 2000.
7. M. Matsui, "Linear cryptanalysis method for DES cipher", *Advances in Cryptology -Eurocrypt'93*, Lecture Notes in Computer Science, Springer-Verlag, pp.386-397, 1993.