Journal of Information Processing Vol.21 No.3 572–579 (July 2013)

[DOI: 10.2197/ipsjjip.21.572]

Regular Paper

Scan-based Attack against DES and Triple DES Cryptosystems Using Scan Signatures

Hirokazu Kodera[1,a)] Masao Yanagisawa[1] Nozomu Togawa[1,b)]

Received: October 15, 2012, Accepted: April 5, 2013

Abstract: A scan-path test is one of the useful design-for-test techniques, in which testers can observe and control registers inside the target LSI chip directly. On the other hand, the risk of side-channel attacks against cryptographic LSIs and modules has been pointed out. In particular, scan-based attacks which retrieve secret keys by analyzing scan data obtained from scan chains have been attracting attention. In this paper, we propose two scan-based attack methods against DES and Triple DES using *scan signatures*. Our proposed methods are based on focusing on particular bit-column-data in a set of scan data and observing their changes when giving several plaintexts. Based on this property, we introduce the idea of a scan signature first and apply it to DES cryptosystems. In DES cryptosystems, we can retrieve secret keys by partitioning the S-BOX process into eight independent sub-processes and reducing the number of the round key candidates from 2^{48} to $2^6 \times 8 = 512$. In Triple DES cryptosystems, three secret keys are used to encrypt plaintexts. Then we retrieve them one by one, using the similar technique as in DES cryptosystems. Although some problems occur when retrieving the second/third secret key, our proposed method effectively resolves them. Our proposed methods can retrieve secret keys even if a scan chain includes registers except a crypto module and attackers do not know when the encryption is really done in the crypto module.Experimental results demonstrate that we successfully retrieve the secret keys of a DES cryptosystem using at most 32 plaintexts and that of a Triple DES cryptosystem using at most 36 plaintexts.

Keywords: side-channel attacks, data encryption standard, triple data encryption standard, scan chain, scan-based attack

1. Introduction

Data encryption standard (DES) is a common-key-encryption method developed by IBM. It was certified by NIST in 1977, and since then it has been used internationally. However, the safety of DES was threatened by the computational power improvement in recent years and there is a move to migrate to Triple DES, in which we apply the DES encryption/decryption algorithms three times to plaintexts to increase the encryption strength. Triple DES is widely used as a mutual authentication such as basic resident registration cards and electronic money cards.

By the way, the importance of design-for-test techniques which inspect electrical transistor faults and validations has been increasing. A scan-path test is one of the design-for-test techniques which makes it possible to easily perform the test and verification of LSIs. It can observe and control register data from outside using scan chains which connect registers in a serial fashion. Testers insert test data into registers from scan-in pins and obtain register data from the scan-out pins as *scan data*. Note that, a bit sequence of scan data itself has no logical meanings, since scan chains are designed such that registers which are placed close to each other are connected physically. This means that only the scan chain designer knows the correspondence between scan data and registers inside the scan chain.

Recently, the risk of side-channel attacks against crypto LSIs and modules has been pointed out. In particular, scan-based attacks which retrieve secret keys by analyzing scan data obtained from scan chains has been attracting attention. Scan-based attacks against DES, AES, RSA, and ECC cryptosystems have been proposed in Refs. [2], [3], [4], [6], [7]. In this paper, we pick up DES and Triple DES as a target cryptosystem.

One of the scan-based attack problems against DES is that attackers cannot know the correspondence between scan data and registers inside the scan chain. Yang et al. solved this problem by inputting several different plaintexts and observing the difference between their corresponding scan data. However, they have several unreasonable assumptions here that the scan chain includes only registers of the target DES cryptosystem and that they can obtain scan data at any time. As far as we know, there are no research reports for scan-based attacks against Triple DES.

In this paper, we propose two scan-based attack methods against DES and Triple DES[*1]. Our proposed method against a DES cryptosystem is based on the property that a particular register data appears in the same bit position of its scan data. Based on this property, we introduce the idea of a scan signature first and apply it to DES cryptosystems. In addition, we retrieve its secret key by partitioning the S-BOX process into eight independent sub-processes and reducing the number of the round key candi-

[1] Waseda University, Shinjuku, Tokyo 169–8555, Japan
[a)] hirokazu.kodera@togawa.cs.waseda.ac.jp
[b)] ntogawa@waseda.jp

[*1] The preliminary version of this paper appeared in Ref. [1]. In Ref. [1], a scan-based method against DES is proposed and its experimental evaluation results are shown.

dates from 2^{48} to $2^6 \times 8 = 512$. Our proposed method successfully retrieves its secret key, if its scan chain includes registers that store the intermediate data in the DES encryption process. Furthermore, our proposed method successfully retrieves its secret key, even if we do not know when the encryption is really done in the crypto module. Triple DES consists of three *crypto blocks* and our proposed method against Triple DES retrieves three 64-bit secret keys making use of our scan-based attack against DES, one by one. We can delete the scan data corresponding to the first crypto block, since they are not needed when retrieving the secret key of the second crypto block. Then, we retrieve the secret keys of the second and third crypto block, in the same way as that of the secret key of the first crypto block. Experimental results demonstrate that we successfully retrieve the secret keys of a DES cryptosystem using at most 32 plaintexts and that of a Triple DES cryptosystem using at most 36 plaintexts.

This paper is organized as follows: Section 2 summarizes our target DES and Triple DES encryption algorithms; Section 3 proposes a scan-based attack method against DES cryptosystems and Section 4 proposes a scan-based attack method against Triple DES cryptosystems; Section 5 demonstrates that our proposed scan-based attacks can successfully retrieves secret keys from DES and Triple DES; Section 6 gives concluding remarks.

2. DES and Triple DES Encryption Algorithms [8]

Data Encryption Standard (DES), one of common-key-encryption methods, has been used internationally as the de facto standard in encryption technology. However, the safety of DES was threatened by the computational power improvement in recent years and there is a move to migrate to Triple DES, in which we apply the DES encryption/decryption algorithms three times to plaintexts to increase the encryption strength. Nowadays DES and Triple DES are very generally used as an encryption/decryption standard.

In this section, we briefly summarize our target DES and Triple DES algorithms.

2.1 Data Encryption Standard

DES is a block-based cryptosystem using a 64-bit secret key and encrypts a 64-bit plaintext. Actually, its secret key is 56 bits long since a parity bit is inserted at every seven bits in its 64-bit secret key. The DES encryption algorithm has *Feistal structure*, in which the *Round* is repeatedly executed.

The Algorithm

The DES encryption algorithm is shown in **Fig. 1**. We first generate 16 *round keys* $k_1, \cdots, k_{16}$ from the secret key. This process is called *Key schedule*. After that, we encrypt a plaintext through 1st round, 2nd round, $\cdots$, and 16th round. Each ith round i uses the round key k_i and is realized by the F-function. F-function is composed of Expansion E, EXOR with the round key k_i, S-Box Permutation, and Permutation P:

Expansion E converts a 32-bit input into a 48-bit output based on the table E.

EXOR with Round Key performs EXOR of the output of E and the round key k_i.

S-Box Permutation performs nonlinear transformation from 6-bit inputs to 4-bit outputs based on the eight tables $S_1, S_2, \cdots, S_8$.

Permutation P converts a 32-bit input to a 32-bit output based on the table P.

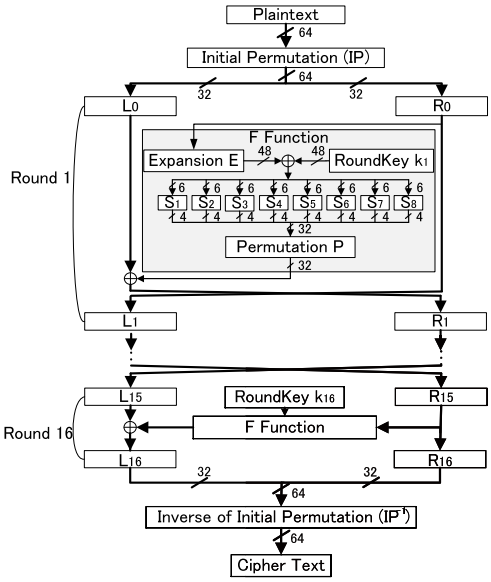

Fig. 1 DES encryption algorithm.

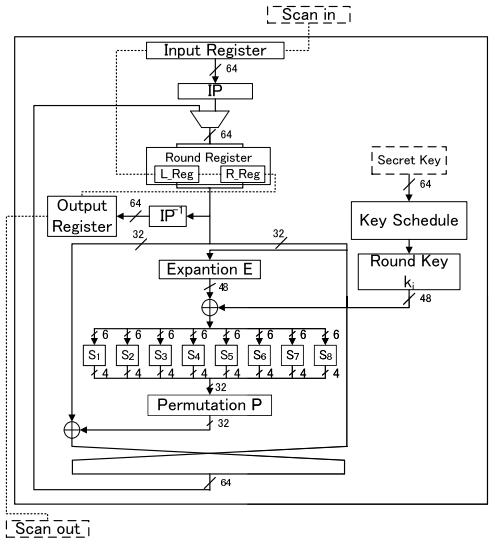

Fig. 2 DES cryptosystem with loop architecture.

Note that, Initial Permutation (IP) is performed before the 1st round is done and Inverse of Initial Permutation (IP^{-1}) is performed after the 16th round is over.

Hardware Architecture

In order to implement DES cryptosystems, we can use both a loop architecture and a pipeline architecture but a loop architecture is very often used. In this paper, we assume that DES cryptosystems are implemented by a loop architecture as shown in **Fig. 2**. First, a plaintext is stored in Input Register at Cycle 1; Output of initial permutation IP is stored into Round Register at Cycle 2; From Cycle 3 to Cycle 18, intermediate data just after each round function is stored into Round Register; Finally,

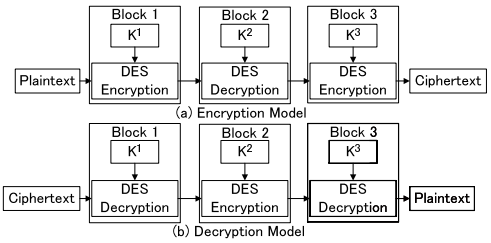

Fig. 3 Triple DES encryption/decryption algorithms.

output of inverse of initial permutation IP^{-1} is stored into Output Register as a ciphertext at Cycle 19.

As in Fig. 2, our scan chain includes at least 192 bits, that is Input Register, Round Register and Output Register. In other words, our scan chain may include other registers than Input Register, Round Register and Output Register.

2.2 Triple Data Encryption Standard

The Triple DES encryption algorithm improves the cryptological strength by applying the DES encryption/decryption algorithm three times. Triple DES is composed of three *crypto blocks* and uses three 64-bit secret keys as in the DES algorithm. Each of three secret keys is applied into each crypto block.

The Algorithm

The Triple DES encryption/decryption algorithms are shown in **Fig. 3**. Let P and C be a plaintext and a ciphertext encrypted by Triple DES, respectively. Then Triple DES encryption can be shown as:

$$C = E_{K^3}(D_{K^2}(E_{K^1}(P))), \tag{1}$$

where $E_K(I)$ and $D_K(I)$ shows DES encryption and decryption with the secret key K, respectively, and K^1, K^2 and K^3 are the three secret keys in Triple DES.

In the same way, Triple DES decryption is shown as:

$$P = D_{K^3}(E_{K^2}(D_{K^1}(C))). \tag{2}$$

The secret keys K^1, K^2 and K^3 here have the following three options:

(1) $K^1 \neq K^2 \neq K^3$

(2) $K^1 \neq K^2$ and $K^1 = K^3$

(3) $K^1 = K^2 = K^3$

If we set the secret keys so that $K^1 = K^2 = K^3$, Triple DES encryption can be equivalent to DES encryption.

Hardware Architecture

In this paper, we assume that Triple DES cryptosystems are also implemented by a loop architecture as in Fig. 2 and its scan chain includes at least 192 bits, that is Input Register, Round Register and Output Register.

3. Scan-based Attack against DES

Scan-based attack is one of the side channel attacks, which retrieves secret keys using scan data which can be obtained from the scan chain. Yang et al. proposed scan-based attack against DES in 2004 and these are the only research results reported so far as the scan-based attack against DES.

In this section, we introduce Yang's scan-based attack method and point out its inherent severe concerns. In order to resolve these concerns, we propose a new scan-based attack against DES using scan signatures. We assume that the scan chain is full scan and connects the flip-flops other than the ones including the secret key data. In addition, we can obtain flip-flop data which are connected to the scan chain as the uncompressed scan data at every cycle.

3.1 Yang's Method

First Yang's method has the following assumptions [6], [7]:

(Y1) Attackers can input any plaintexts into the target DES cryptosystem.

(Y2) Attackers know the DES encryption algorithm.

(Y3) Scan chain is essentially composed of just 192 bits, that is Round Register, Input Register and Output Register of the DES cryptosystem.

(Y4) Attackers can obtain scan data at any time.

Based on these assumptions (Y1)–(Y4), Yang's method retrieves a secret key from a DES cryptosystem.

When an attacker executes a scan-based attack against DES, there arise the two problems: *Encyption timing problem* and *Register connection order problem*. The first one is that, attackers do not know when the encryption is really done in the crypto module. The second one is that, attackers do not know the correspondence between scan data and registers inside the scan chain.

Yang et al. [6] resolved *Encyption timing problem* by giving the assumption (Y4). Similarly, Yang et al. resolved *Register connection order problem* by making use of the assumptions (Y3) and (Y4) and by inputting into the target DES cryptosystem several plaintexts whose contents are different at one bit to each other.

However, we cannot say that the assumptions (Y3) and (Y4) are reasonable. Scan chains always include not only registers inside a crypto module but registers inside peripheral modules. In addition, attackers cannot know when the encryption is really done in the crypto module. This means that attackers find it too hard to obtain scan data at any time.

3.2 Scan-based Attack against DES

We pointed out in the previous subsection that the two assumptions (Y3) and (Y4) of Yang's method are the critical concerns. Our proposed algorithm relaxes these two assumptions but resolves the two problems: *Encyption timing problem* and *Register connection order problem*.

In our proposed scan-based attack method, we have the following five assumptions:

(K1) Attackers can input any plaintexts into the target DES cryptosystem.

(K2) Attackers know the DES encryption algorithm.

(K3) The scan chain includes at least 32-bit Round Register of the target DES cryptosystem.

(K4) Attackers can obtain scan data over several cycles which include encryption processing at least.

The assumptions (K1) and (K2) are the same as Yang's assumptions (Y1) and (Y2). But we do not need the assumptions (Y3) and (Y4), which are an essential part of Yang's method. Instead we assume (K3) and (K4). (K3) says that our scan chain includes at least Round Register in Fig. 2. This means that the

scan chain can include registers other than Round Register, such as peripheral module registers. (K4) says that scan data includes several cycles which include encryption processing cycles. This means that we do not always know the exact timing of the encryption processing cycles.

We show why the exact timing of the encryption processing cycles is not known even if an attacker can input any plaintexts into the target cryptosystem at an arbitrary timing. We show an example as follows: **Fig. 4** shows an LSI chip which has DES crypto module and two other modules, A and B. We assume that the three modules have the conditions as shown below:

- The modules A and B execute some process in 19 cycles which are the same as DES.
- The execution sequence is A, DES and B as shown in **Fig. 5**.
- The number of flip-flops in the module A as well as B is the same as that of DES.
- An attacker does not know the execution sequence nor the connection sequence of the scan chain.

After the LSI executes the sequence of A, DES and B in 19×3 = 57 cycles, we have the scan data as shown in Fig. 4. From the 1st cycle to the 19th cycle, the scan data bits corresponding to the module A are changing. However an attacker does not know which module these scan data bits correspond to. Similary, from the 20th cycle to the 38th cycle and from the 39th cycle to the 57th cycle, the scan data bits corresponding to the DES module and to module B, respectively, are changing. However an attacker does not know which module these scan data bits correspond to, either.

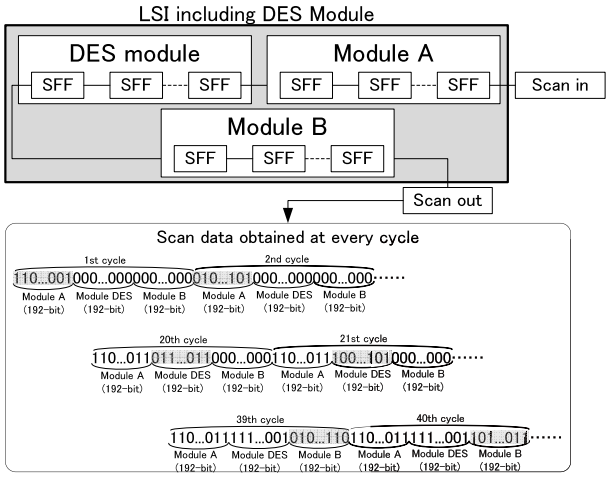

Fig. 4 Block diagram of a target LSI.

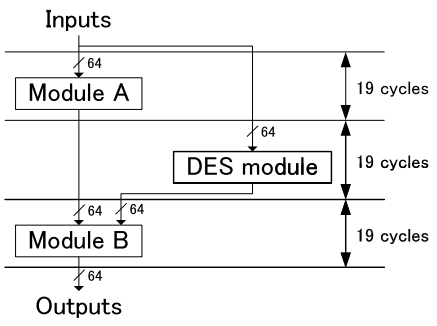

Fig. 5 Execution sequence.

In (Y3), scan chain includes only flip-flops of DES crypto module. Therefore, after obtaining 57-cycle scan data, DES encryption processing cycles are easily found, since only the part of scan data corresponding to DES is changing in encryption processing cycles. In (K3), on the other hand, the scan chain can include flip-flops other than DES crypto module. Therefore, after obtaining 57-cycle scan data as in Fig. 4, the part of scan data corresponding to DES are changing in DES encryption processing cycles, but scan data correponding to modules A and B are also changing. Since we do not know the execution sequence, the encryption processing cycles are not easily found.

Therefore, when we assume the condition (K3), the timing of each module executing is not clear. Even if an attacker can input any plaintexts at an arbitrary time, encryption processing cycles are not clear. We can say that the assumptions (K3) and (K4) are relaxed from the Yang's assumptions (Y3) and (Y4).

The DES encryption algorithm does not directly use the secret key, but uses 16 round keys of 48 bits long, which are generated from the secret key. Conversely, if we have these round keys, we can retrieve the secret key by using the key schedule algorithm. Particularly, it is known that a secret key can be retrieved by using just the three round keys k_1, k_2 and k_3 [6].

In the rest of this section, we consider that retrieving the secret key K is equivalent to retrieving its round keys k_1, k_2 and k_3 and propose a scan-based attack method which retrieves the round key k_1. k_2 and k_3 can be retrieved in a similar way and then we can retrieve the secret key K using k_1, k_2 and k_3.

3.2.1 Scan Signature

Now let's define a scan signature. Assume that we have a cryptosystem including registers storing intermediate calculating data. We *guess* a key k as a *guessed* key g inside the target cryptosystem and prepare n plaintexts to be encrypted. We input each of these plaintexts into a crypto simulator using the guessed key g and obtain intermediate calculating data. We can obtain a set of *comparison data* stored in registers storing intermediate calculating data as shown in **Fig. 6** (a). In this figure, cd_i $(i = 0, \cdots, n-1)$ corresponds to the data in registers storing intermediate calcu-

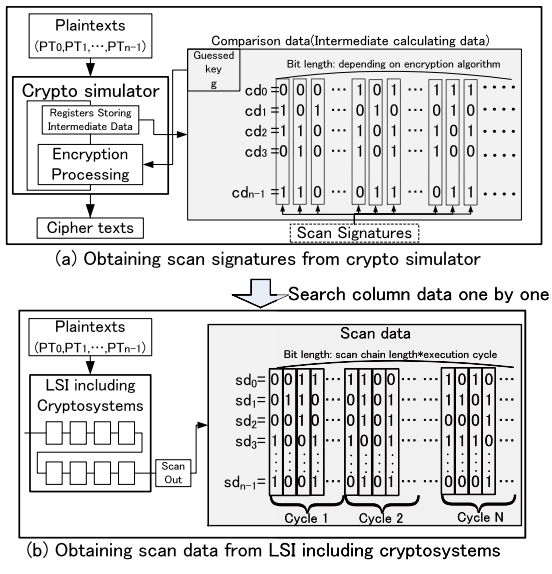

Fig. 6 Scan signature.

lated data when inputting the plaintext PT_i.

A *scan signature* here means a particular bit-column data when a set of comparison data is arranged as a matrix as in Fig. 6 (a). A scan signature indicates a particular 1-bit register change when inputting the prepared plaintexts into the cryptosystem.

Next, let us prepare n plaintexts which are the same as the previous ones and input each of them into the target cryptosystem. While encrypting each of these plaintexts, we can obtain scan data at every clock cycle and finally have a set of scan data as in Fig. 6 (b). In this figure, sd_i $(i = 0, \cdots, n-1)$ corresponds to all the scan data while encrypting the plaintext PT_i.

If we prepare plaintexts whose number n is large enough, the scan signatures are dependent on an input set of n plaintexts and the key used inside the cryptosystem. Then, if we correctly guess the key, we can find out scan signatures inside the set of scan data. But if we mistakenly guess the key, we cannot find out scan signatures inside the set of scan data.

3.2.2 Retrieving the Round Key k_1 Using Scan Signatures

Now we will apply our scan signatures to DES cryptosystems. First, we guess the round key k_1 as a guessed round key g_1 inside the target DES cryptosystem and prepare n plaintexts to be encrypted. We input each of these plaintexts into a crypto simulator using g_1 and finish the 1st round of DES encryption. In this case, we can obtain a set of comparison data stored in Round Register just after the 1st round as shown in **Fig. 7** (a). Scan signatures are particular bit-column data of a set of comparison data as in Fig. 7 (a).

Based on this idea, attackers can retrieve the round key k_1 without knowing specific positions of Round Register in the scan chain as follows:

(1) An attacker inserts n $(n \geq 1)$[*2] plaintexts into the target DES cryptosystem and obtains the scan data at every clock as shown in Fig. 7 (b). As in (K3) we assume that the scan data at least includes Round Register data.

(2) An attacker guesses the round key k_1 as the guess key g_1. An attacker encrypts the n plaintexts by a crypto simulator using g_1 and obtains comparison data and picks up column data of them as scan signatures as in Fig. 7 (a).

(3) An attacker searches whether the scan signature obtained in (2) is included in any column of scan data. If the scan signature is included somewhere in scan data, the guess key g_1 is correct and complete. Otherwise, the guess key g_1 is not correct. Go to (2) and try another guess key.

Since we only focus on 1-bit column of comparison data and generate scan signatures, attackers need not know the correspondence between scan data and registers of the scan chain. This means that we can resolve the *Register connection order problem* here. Furthermore, this scan-based attack method successfully finds out a scan signature when the scan data includes scan signature somewhere. This means that we can also resolve *Encyption timing problem*.

Now let us focus on retrieving the round key k_1 using the scan-based attack described above. The important point here is that, the number of candidate guessed keys is 2^{48} since the round key

[*2] As our experimental results indicate, the number n of plaintexts required to successfully retrieve the secret key is at most 32.

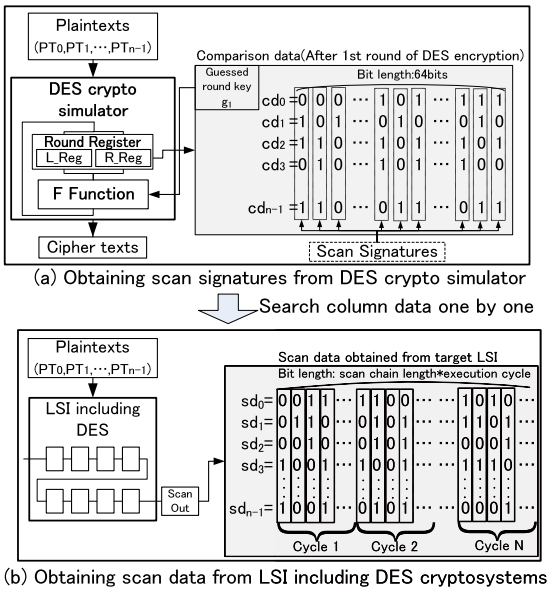

Fig. 7 Scan signature in DES cryptosystems.

k_1 is 48 bits long. It is, however, impossible to try all these 2^{48} candidate guessed keys. This problem is called **Guess key pattern explosion problem**. In the next subsection, we propose a method to effectively resolve this problem.

3.2.3 A Solution to the Guess Key Pattern Explosion Problem[*3]

As shown in Fig. 2, the inputs to S-Box are 48 bits long but they are partitioned into eight 6-bit data and each of them are inputted into the subprocesses of S_1, S_2, and S_8. In addition to that, each 6-bit input data of S_1, S_2, and S_8 is independent of each other. Now the 48-bit round key k_1 is partitioned into eight 6-bit *split keys* $k_{1,1}, k_{1,2}, \cdots, k_{1,8}$ and then we can retrieve each split key in this order. Then the number of guess key patterns can be reduced from 2^{48} to $2^6 \times 8 = 512$ and the **guess key pattern explosion problem** can be too much relaxed and resolved.

Now let's discuss how to retrieve the split key $k_{1,1}$. As just described before, the number of candidate guess keys for $k_{1,1}$ is $2^6 = 64$, which is reasonably small. After obtaining a set of scan data and a set of comparison data, attackers need to determine which column of comparison data should be a scan signature to effectively identify the split key. As depicted in **Fig. 8**, the input of S_1 in S-Box is dependent on $k_{1,1}$. Furthermore, the inputs of subprocesses other than S_1 is independent of $k_{1,1}$. The output of S_1 affects the 9th, 17th, 23th, and 31st bits of the Permutation P output. They also affects the 41st, 49th, 55th, and 63rd bits of Round Register. Therefore, we use four scan signatures, the 41st, 49th, 55th, and 63rd column of comparison data as shown in **Fig. 9**.

The proposed scan-based attack method against the split key $k_{1,1}$ is summarized as follows:

(1) An attacker prepares n plaintexts, $PT^0, PT^1, PT^2, \ldots, PT^{n-1}$, encrypts them using the target DES cryptosystem and

[*3] The technique of splitting the round key has been used in the other side-channel analysis [5]. However, the technique which focuses on particular bits of Round Register (ex. 41st, 49th, 55th and 63rd bits when retrieving $k_{1,1}$) to search the correpondence between scan data and comparison data is our original one proposed in this paper.

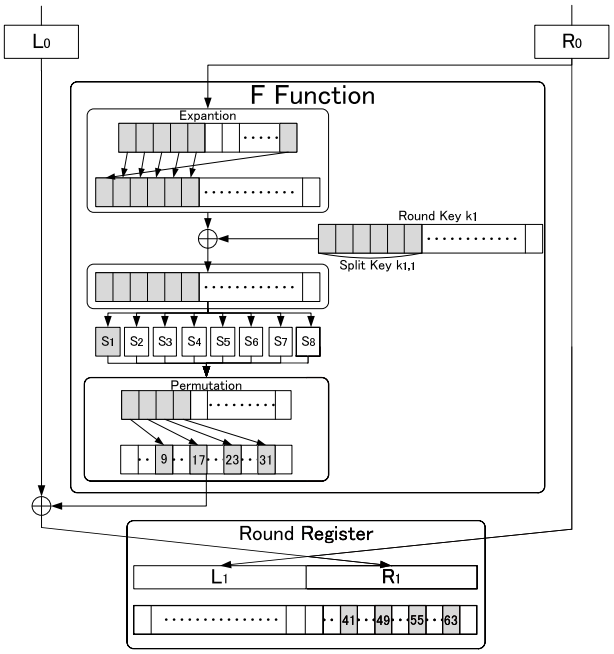

Fig. 8 Retrieving the split key $k_{1,1}$.

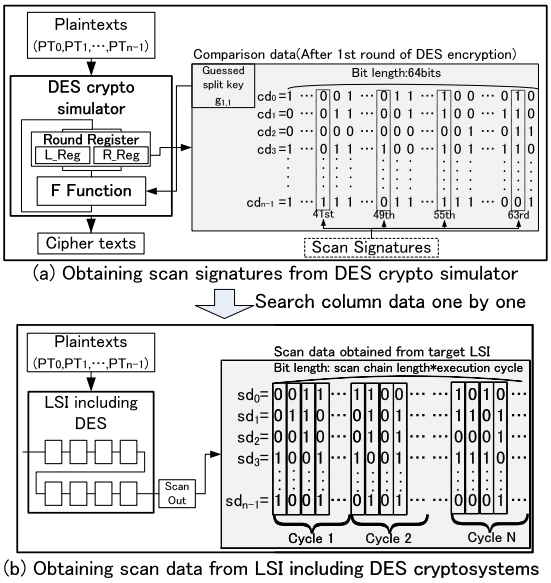

Fig. 9 Scan signature when retrieving $k_{1,1}$.

obtains a set of scan data $sd^0, sd^1, sd^2, \ldots, sd^{n-1}$.

(2_1) An attacker guesses the split key $k_{1,1}$ as $g_{1,1}$. An attacker encrypts the n plaintexts by a crypto simulator using $g_{1,1}$ and obtains Round Register data as comparison data $(cd^0, cd^1, cd^2, \ldots, cd^{n-1})$ after the 1st round 1 is over.

(2_2) Let four scan signatures be 41st, 49th, 55th, and 63rd columns of comparison data as shown in Fig. 9.

(3) If the four scan signatures are included in the scan data, the guess key $g_{1,1}$ is correct. Otherwise, $g_{1,1}$ is not correct and try another one.

In the same way, we can retrieve the other split keys $k_{1,2}, \ldots, k_{1,8}$ and finally have the round key k_1.

4. Scan-based Attack Method against Triple DES

In this section, we also assume the four assumptions of (K1)–(K4) in the Section 3 and propose a scan-based attack method against Triple DES. Because Triple DES encryption applies the DES encryption/decryption algorithm three times, three secret keys are used there. We refer to the three secret keys as K^1, K^2, and K^3.

The four assumptions (K1)–(K4) hold when we see the entire Triple DES encryption algorithm, but when we see each of the three crypto blocks in Triple DES individually, all of these four assumptions do not hold. When we see the first crypto block, the four assumptions (K1)–(K4) hold. This means that the secret key K^1 can be easily retrieved by using the scan-based attack method against DES. However, when we see the second/third crypto block, the assumption (K1) no longer holds true since its input is the output of its previous block. In this section, we propose a scan-based attack method that retrieves secret keys K^2 and K^3 assuming that the secret key K^1 is already retrieved.

When we see the structure of Triple DES as in Fig. 3, the input of the second crypto block is connected to the output of the first crypto block. Then, if we know in advance the secret key K^1 and input n plaintexts into the first crypto block, we can also know the n input data to the second crypto block. However, the scan data corresponding to the first crypto block may interfere with retrieving the secret key K^2 and it really does[*4]. We have to know the scan data positions corresponding to the first crypto block and delete them from the oridinal scan data to successfully retrieve K^2.

In order to obtain the scan data positions corresponding to the first crypto block, we have to obtain the bit positions of scan data corresponding to Round Register data just after the 16th round of the first crypto block is over. They can be obtained by using the technique as shown in **Fig. 10**, which is the same as the one retrieving K^1 of the first crypto block. After we obtain them, we can discard scan data from the first bit to the bit positions corresponding to the 16th round of the first crypto block. This means that the scan data does not include the first crypto block scan data and begins with the second crypto block scan data[*5]. By using this scan data and applying the same method as the one retrieving K^1 to it, we can retrieve K^2 of the second crypto block. Similarly, we can retrieve K^3 of the third crypto block.

[*4] In fact, let $k_1^1, k_2^1, \cdots, k_{16}^1$ be the 16 round keys of the first crypto block and $k_{1,1}^1, k_{1,2}^1, \cdots, k_{1,8}^1$ be the split keys of the round key k_1^1. When we retrieve $k_{1,1}^2$ $(\neq k_{16,1}^1)$, assume that we guess its guess key $g_{1,1}^2$ to be $k_{16,1}^1$ and make its scan signatures. Since $k_{1,1}^2$ is not $k_{16,1}^1$, $g_{1,1}^2 = k_{16,1}^1$ is wrong and we must not find out its scan signatures in the scan data. However, we mistakenly find out these scan signatures in the scan data corresponding to the first crypto block. This is because the process in the 16th round of the first crypto block is the opposite of the process in the 1st round of the second crypto block in the Triple DES cryptosystem.

[*5] The first crypto block consists of 19 cycles and its 16th round is executed in the 18th cycle. Then, if we discard scan data from the first bit to the bit positions corresponding to the 16th round of the first crypto block,it still includes the 19th cycle of the first crypto block. But it can just be ignored, since the scan data corresponding to the 19th cycle does not affect anything to retrieving K^2 of the second crypto block.

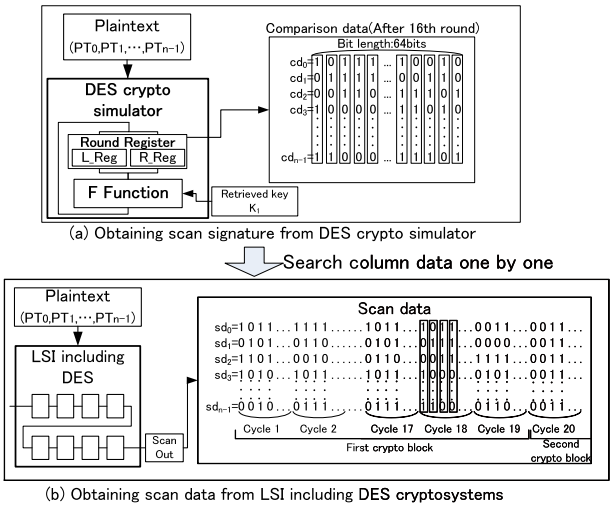

Fig. 10 Decision of scan data from LSI including cryptosystems.

5. Experimental Evaluations

We have implemented our proposed scan-based attack methods against DES and Triple DES and evaluated them through experimental results. In this experiment, we have generated 1,000 and $3 \times 1{,}000 = 3{,}000$ random secret keys and given them DES and Triple DES cryptosystems, respectively, and tried to retrieve them[*6]. We assumed DES and Triple DES cryptosystems as shown in Fig. 2. Throughout this experiment, we have used AMD Quad-Core Opteron 2360 SE 2.5 GHz × 2 with 16 GB main memory. We used gcc with compiling option of -O3.

In Fig. 2, the scan chain is assumed to be 192 bits long, that is Input Register, Round Register, and Output Register, but we have added several random data to the scan chain to show that our method successfully retrieves secret keys even if the scan data includes registers other than Round Register.

5.1 Results on Scan-based Attack against DES

In this subsection, we show the evaluation results on scan-based attack against DES. Since our proposed scan-based attack method is based on scan signatures and they are composed of several plaintexts, the number of plaintexts required to retrieve secret keys correctly is one of the important factors. If we prepare too small number of plaintexts, i.e., n is too small in the algorithm of Section 3.2.1, we cannot correctly find out the target secret key.

Figure 11 shows the average and worst number of plaintexts required to successfully retrieve the secret key out of 1,000 trials when varying the scan chain length. As shown in this figure, we require 32 plaintexts at worst, which can be quite reasonable. **Figure 12** shows the success rate of retrieving correct secret keys out of 1,000 trials when varying the plaintexts.

When the scan chain length is 4,096 bit, the CPU time to retrieve the secret key is just 16 seconds when 25 plaintexts are inputted.

As mentioned in Section 3, Yang's method also retrieves the secret key in DES when the assumptions (Y1)–(Y4) are given. In their results, 67 plaintexts and 6 scan-in data are required to retrieve the secret key. In their experiment, scan chain length is 192 bits. Note that, our scan-based attack method successfully retrieves the secret key in DES using just less than 30 plaintexts as in Fig. 11, which means that our proposed method is superior to Yang's method in terms of the number of required plaintexts as well as the required assumptions.

[*6] Since Triple DES requires three secret keys, we prepared 3,000 secret keys in this experiment and perform 1,000 trials.

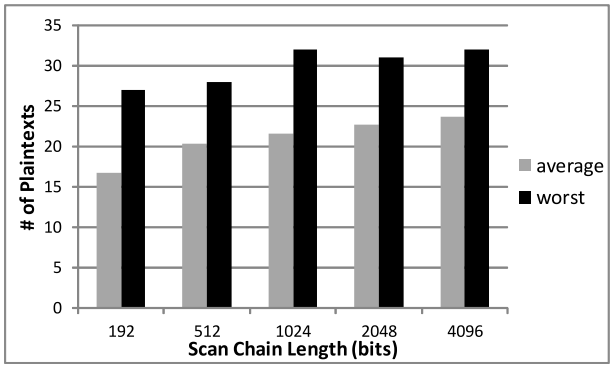

Fig. 11 The number of required plaintexts (average and worst) to retrieve secret keys in DES.

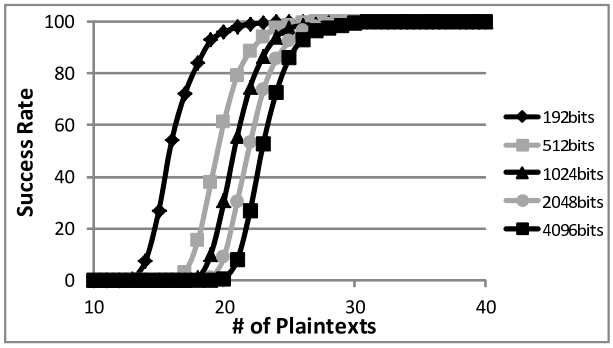

Fig. 12 Number of plaintexts vs Success rate ((# of secret keys successfully retrieved)/(# of trials)) in DES.

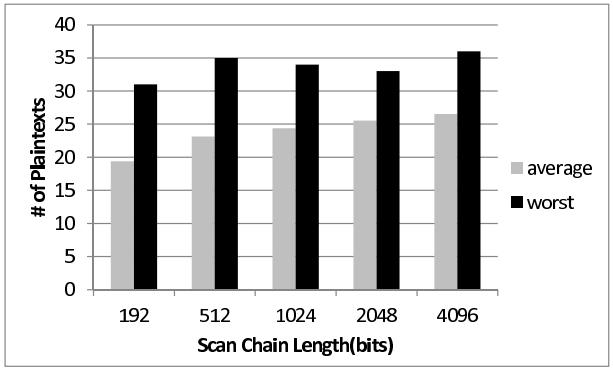

Fig. 13 The number of required plaintexts (average and worst) to retrieve secret keys in Triple DES.

5.2 Results on Scan-based Attack against Triple DES

In this subsection, we show the evaluation results on scan-based attack against Triple DES. **Figure 13** shows the average and worst number of plaintexts required to successfully retrieve the secret key out of 1,000 trials when varying the scan chain length. As shown in this figure, we require 36 plaintexts at worst, which can be quite reasonable. **Figure 14** shows the success rate of retrieving correct secret keys out of 1,000 trials when varying the plaintexts.

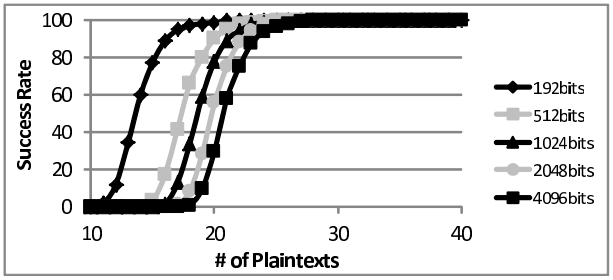

Fig. 14 Number of plaintexts vs Success rate ((# of secret keys successfully retrieved)/(# of trials)) in Triple DES.

According to the experimental results, the number of plaintexts for retrieving secret keys is not so changed even when the scan chain length is doubled. When the scan chain length is 4,096 bit, the CPU time to retrieve the secret key is just 105 seconds when 27 plaintexts are inputted.

Overall, our experimental results show that our proposed scan-based attack methods can successfully retrieve secret keys in a reasonable amount of CPU time.

6. Conclusions

In this paper, we proposed a scan-based attack method against DES and Triple DES using scan signatures. Our proposed method is based on focusing on particular bit-column-data of scan data and the secret keys in DES and Triple DES can be retrieved even if attackers do not know the correspondence between scan data and registers inside the scan chain. In addition, if a scan chain includes registers other than Round Register such as peripheral module registers, the secret keys can be successfully retrieved.

In the future, we will consider scan-based attack methods against other cryptosystems such as stream ciphers and further generalize our scan signatures to other block ciphers.

Acknowledgments This research is supported in part by Secom Science and Technology Foundation.

References

[1] Kodera, H., Yanagisawa, M. and Togawa, N.: Scan-based Attack against DES Cryptosystems Using Scan Signatures, *IEEE Asia Pacific Conference on Circuits and Systems*, pp.599–602 (Dec. 2012).

[2] Nara, R., Togawa, N., Yanagisawa, M. and Ohtsuki, T.: A scan-based attack based on discriminators for AES cryptosystems, *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E92-A, No.12, pp.3229–3237 (Dec. 2009).

[3] Nara, R., Satoh, K., Yanagisawa, M., Ohtsuki, T. and Togawa, N.: Scan-based side-channel attack against RSA cryptosystems using scan signatures, *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E93-A, No.12, pp.2481–2489 (Dec. 2010).

[4] Nara, R., Yanagisawa, M., Ohtsuki, T. and Togawa, N.: Scan vulnerability in elliptic curve cryptosystems, *IPSJ Trans. System LSI Design Methodology*, Vol.4, pp.47–59 (Feb. 2011).

[5] Kocher, P., Jaffe, J. and Jun, B.: Differential Power Analysis, *Lecture Notes in Computer Science*, Vol.1666, pp.388–397 (1999).

[6] Yang, B., Wu, K. and Karri, R.: Scan based side channel attack on dedicated hardware implementations of data encryption standard, *Proc. International Test Conference*, pp.339–344 (2004).

[7] Yang, B., Wu, K. and Karri, R.: Secure scan: A design-for-test architecture for crypto chips, *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, Vol.25, No.10, pp.2287–2293 (2006).

[8] FIPS 46-3, Data Encryption Standard (DES), available from ⟨http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf⟩.

Hirokazu Kodera received his B. Eng. and M. Eng. degrees from Waseda University in 2011 and 2013, respectively, all in computer science and engineering. His research interests include design and verification of LSIs, especially scan-based attack against crypto LSIs.

Masao Yanagisawa received his B. Eng., M. Eng., and Dr. Eng. degrees from Waseda University in 1981, 1983 and 1986, respectively, all in electrical engineering. He was with University of California, Berkeley from 1986 through 1987. In 1987, he joined Takushoku University. In 1991, he left Takushoku University and joined Waseda University, where he is presently a Professor in the Department of Computer Science and Engineering. His research interests are combinatorics and graph theory, computational geometry, LSI design and verification, and network analysis and design. He is a Fellow of IEICE and a member of IEEE and ACM.

Nozomu Togawa received his B. Eng., M. Eng., and Dr. Eng. degrees from Waseda University in 1992, 1994 and 1997, respectively, all in electrical engineering. He is presently a Professor in the Department of Computer Science and Engineering, Waseda University. His research interests are LSI design, graph theory, and computational geometry. He is a member of IEEE and IEICE.