IPSJ Transactions on System LSI Design Methodology Vol. 4 47–59 (Feb. 2011)

Regular Paper

Scan Vulnerability in Elliptic Curve Cryptosystems

Ryuta Nara,[†1] Nozomu Togawa,[†1]
Masao Yanagisawa[†2] and Tatsuo Ohtsuki[†1]

A scan-path test is one of the most important testing techniques, but it can be used as a side-channel attack against a cryptography circuit. Scan-based attacks are techniques to decipher a secret key using scanned data obtained from a cryptography circuit. Public-key cryptography, such as RSA and elliptic curve cryptosystem (ECC), is extensively used but conventional scan-based attacks cannot be applied to it, because it has a complicated algorithm as well as a complicated architecture. This paper proposes a scan-based attack which enables us to decipher a secret key in ECC. The proposed method is based on detecting intermediate values calculated in ECC. We focus on a 1-bit sequence which is specific to some intermediate values. By monitoring the 1-bit sequence in the scan path, we can find out the register position specific to the intermediate value in it and we can know whether this intermediate value is calculated or not in the target ECC circuit. By using several intermediate values, we can decipher a secret key. The experimental results demonstrate that a secret key in a practical ECC circuit can be deciphered using 29 points over the elliptic curve E within 40 seconds.

1. Introduction

Smart cards used as credit cards and banking cards contain an LSI chip to achieve secure communication and reject counterfeit cards. The LSI chip usually includes cryptography circuits and encrypt/decrypt important data such as ID numbers and electronic money information. However, there is a threat that a secret key may be deciphered in the cryptography LSI chip. Scan-based attack is a method to retrieve a secret key from the scanned data obtained from the scan path in the cryptography LSI chip. Yang, et al. first showed a scan-based attack against DES in 2004 and deciphered secret keys in DES [1]. The scan-based attack against AES was also presented in 2006 [2] and 2009 [3].

Symmetric-key cryptosystems such as DES and AES are very popular and widely used. They make use of the same secret key in encryption and decryption. However, it may be difficult to securely share the same secret key, such as in communicating on the Internet. Public-key cryptosystems, on the other hand, make use of different keys to encrypt and decrypt so that it solves the key sharing problem.

An elliptic curve cryptosystem (ECC) [4),5] is well known as a public-key cryptosystem with low-cost and high throughput. Finite field arithmetic is used in ECC where field multiplication requires most of the time in decryption and encryption and thus many research have been done in field multiplication [6)–10]. Also many research on an ECC circuit implementation are reported as in Refs. 8)–16). For instance, architectures including memories storing all ECC parameters and field multipliers which can execute the arbitrary polynomial reduction are proposed in Refs. 8), 10) for high-throughput ECC applications. On the contrary, architectures including minimal memories storing fixed polynomial reduction and a field-dedicated multiplier are proposed in Refs. 9), 15) for low-area and low-cost ECC applications.

Deciphering a secret key in a security LSI chip by using a scan path, we have to find out positions of registers storing the secret key in the scan path. There are, however, many architectures and implementations as above in ECC and then there can be many scan-path structures as well. This means that it is very difficult to find out positions of registers storing a secret key in a scan path in the ECC circuit. In other words, it is very difficult to retrieve a secret key from the scanned data. For that reason, scan-based attacks against symmetric-key cryptosystem succeed as reported in Refs. 1)–3), but a scan-based attack against public-key cryptosystem such as ECC has not been proposed yet.

In this paper, we propose a scan-based attack against ECC which is almost independent of a scan-path structure [⋆1]. The proposed method is based on detecting intermediate values calculated in an ECC circuit. We focus on a 1-bit sequence which is specific to some intermediate values. Then we check whether

†1 Department of Computer Science and Engineering, Waseda University
†2 Department of Electronic and Photonic Systems, Waseda University

⋆1 The preliminary version of this paper appeared in Ref. 17).

data dependent on this intermediate value is included in the scanned data. As long as a scan path is implemented on the ECC circuit and it includes at least 1-bit of each intermediate value, we can decipher a secret key in the target ECC circuit even if we do not know a scan path structure. The proposed method reveals the vulnerability of a scan path in the ECC circuit.

The purpose of our proposed method is, not to break-through secure scan architecture but to decipher a secret key using scanned data in an ECC LSI with as few limitations as possible. In fact, our scan-based attack method without any modification might not work against ECC LSIs using some secure scan architecture.

This paper is organized as follows: Section 2 briefly explains an elliptic curve cryptosystem; Section 3 describes an intermediate-value-analysis attack against an ECC circuit and points out the critical problem when applying it to a scan-based attack. Section 4 proposes our scan-based attacking method against the ECC circuit based on the discussion on Section 3; Section 5 demonstrates experiments on a practical ECC architecture; and Section 6 gives several concluding remarks.

2. Elliptic Curve Cryptosystem

An elliptic curve cryptosystem makes use of the difficulty in solving the discrete logarithm problem defined in the elliptic curve additive group. This problem is called the *elliptic curve discrete logarithm problem* (ECDLP). The 160-bit key in ECC provides the equivalent security level as the 1024-bit key in RSA [18]. An ECC circuit can have higher throughput and smaller area than an RSA circuit. This section briefly explains ECC [4),5)].

2.1 Elliptic Curve Arithmetic

An elliptic curve E with non-supersingular over a field $\mathbb{F}_{2^m}$ is defined by Eq. (1).

$$E : y^2 + xy = x^3 + ax^2 + b. \quad (1)$$

Let $E(\mathbb{F}_{2^m})$ be a group of points on the elliptic curve E. $E(\mathbb{F}_{2^m})$ has the four properties shown below and forms a group.

(1) *Identity.* $\infty \in E(\mathbb{F}_{2^m})$ is called the *identity* and it satisfies $P + \infty = \infty + P = P$ for all $P \in E(\mathbb{F}_{2^m})$.

(2) *Negatives.* If $P = (x, y) \in E(\mathbb{F}_{2^m})$, then $(x, y) + (x, x + y) = \infty$. The point $(x, x + y)$ is denoted by $-P$ and is called the *negative* of P.

(3) *Point addition.* Let $P_1 = (x_1, y_1) \in E(\mathbb{F}_{2^m})$ and $P_2 = (x_2, y_2) \in E(\mathbb{F}_{2^m})$, where $P_1 \neq \pm P_2$. Then $P_1 + P_2 = (x_3, y_3) = Q \in E(\mathbb{F}_{2^m})$, where

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a$$
$$y_3 = \lambda(x_1 + x_3) + x_3 + y_1$$

with $\lambda = (y_1 + y_2) / (x_1 + x_2)$. **Figure 1** shows the point addition.

(4) *Point doubling.* Let $P = (x_1, y_1) \in E(\mathbb{F}_{2^m})$, where $P \neq -P$. Then $2P = (x_3, y_3) = Q \in E(\mathbb{F}_{2^m})$ where

$$x_3 = \lambda^2 + \lambda + a = x_1^2 + \frac{b}{x_1^2}$$
$$y_3 = x_1^2 + \lambda x_3 + x_3$$

with $\lambda = x_1 + y_1/x_1$. **Figure 2** shows the point doubling.

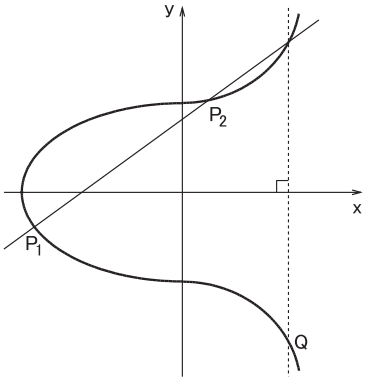

Fig. 1 Point Addition $P_1 + P_2 = Q$.

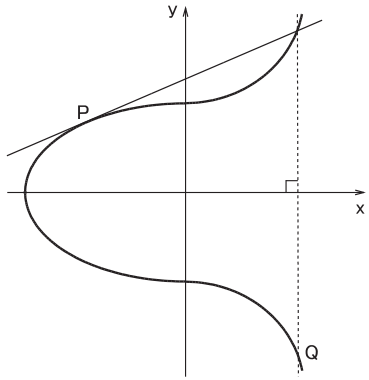

Fig. 2 Point Doubling $2P = Q$.

2.2 Point Multiplication

Let k be an m-bit integer and denoted as $k = k_{m-1}2^{m-1} + k_{m-2}2^{m-2} + \cdots + k_1 2 + k_0$. A *point multiplication* is defined by computing kP with k and $P \in E(\mathbb{F}_{2^m})$. The point multiplication is calculated in polynomial time by using point addition and point doubling. Given P, Q, where Q is a result of the point multiplication with k and P. To determine an integer k satisfying the equation $[kP \equiv Q \mod f(z)]$ ⋆1 is an *elliptic curve discrete logarithm problem* (ECDLP).

⋆1 $f(z)$ is an irreducible polynomial.

Algorithm 1 Montgomery method

Require: $k = (1, k_{m-2}, \ldots, k_1, k_0)_2, P \in E(\mathbb{F}_{2^m})$
Ensure: $Q_0 = kP$
1: $Q_0 \Leftarrow P$
2: $Q_1 \Leftarrow 2P$
3: **for** $i = m - 2$ to 0 **do**
4: $\quad Q_{1-k_i} \Leftarrow Q_0 + Q_1$
5: $\quad Q_{k_i} \Leftarrow 2Q_{k_i}$
6: **end for**
7: **return** Q_0

Solving the elliptic curve discrete logarithm problem requires exponential time. If the integer k is large enough, the point multiplication $Q = kP$ can be calculated easily. However determining k from the point P and $Q \in E(\mathbb{F}_{2^m})$ requires very long time. Q can be used as a public key and k can be used as a secret key in ECC.

The point multiplication kP dominates the execution time of ECC so that several efficient algorithms have been proposed. *Montgomery method* [19] is one of point multiplication algorithms. This algorithm has two advantages. One is that it does not require any extra storage with a low calculation time. The other is that the same operations are performed in every iteration of the main loop, therefore it has a resistance against power analysis attacks [20].

The Montgomery method is first proposed in Ref. 19) and shown in **Algorithm 1**. It converts affine coordinate (x, y) into projective coordinates (X, Y, Z) to reduce total calculation amount. Algorithms in Refs. 21)–24) are also based on the original Montgomery method. In this algorithm, the secret key k is written as $2^{m-1} + k_{m-2}2^{m-2} + \cdots + k_1 2 + k_0$. k_{m-1} will be always one to achieve the same number of iterations in the main loop.

3. Attack Against Elliptic Curve Cryptosystem

A scan path connects registers in an circuit serially so that a tester can observe the register values inside the circuit easily. The scan path is widely used in recent circuit implementations due to its testability and easiness.

Scan path test needs to replace standard flip-flops (FFs) with scan flip-flops (SFFs). An SFF usually consists of an FF and a multiplexer. The multiplexer output pin is connected to the FF input pin. It selects one from its two inputs.

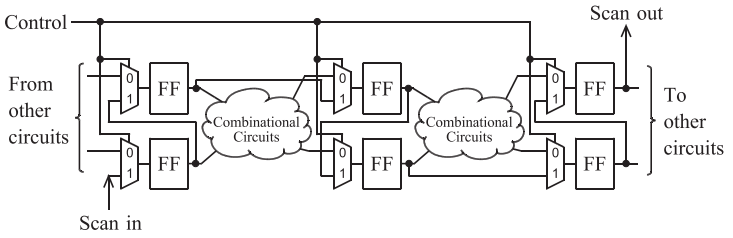

Fig. 3 Scan path model.

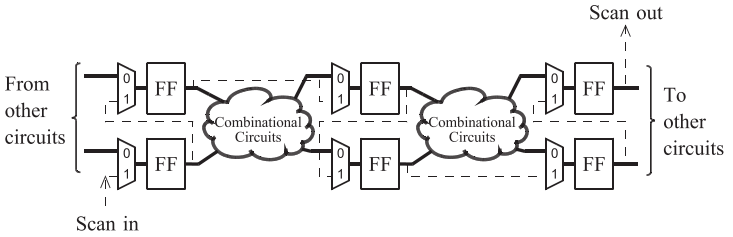

Fig. 4 System mode (Control = 0).

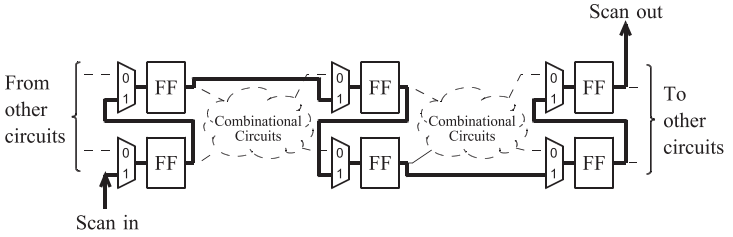

Fig. 5 Test mode (Control = 1).

When the select line of the multiplexer is 0, it outputs the combinational circuits output. When the select line of the multiplexer is 1, it outputs the SFF output. A scan path model is shown in **Fig. 3**. Control pin is used to choose between the system mode or the test mode. While Control pin is 0, normal operation is performed in the system mode as shown in **Fig. 4**. While Control pin is 1, SFFs are connected serially and we obtain scanned data stored in each FF from the scan out as shown in **Fig. 5**. By using a scan-in pin in the test mode, a test

pattern is inputted to the SFFs.

The purpose of a scan-based attack is to decipher a secret key from scanned data in an ECC circuit. Scan-based attack here requires several assumptions as in the previous research in Refs. 1)–3) which are summarized as shown below:

(1) Attackers can input an arbitrary point $P = (x, y) \in E(\mathbb{F}_{2^m})$ into a target ECC circuit.

(2) Attackers can obtain scanned data from the target circuit.

In this section, we explain the scan-based attack against ECC.

3.1 Deciphering a Secret Key Using Intermediate Values During the Point Multiplication

In order to decipher a secret key k, we have to solve the discrete logarithm problem in the elliptic curve additive group. If the bit length of secret key k is more than 160, it is impossible to solve this problem within realistic time. However, if we know all the "intermediate values" during the point multiplication in Algorithm 1, we can decipher a secret key k in a polynomial time [25].

Let $k = k_{m-1}2^{m-1} + k_{m-2}2^{m-2} + \cdots + k_1 2 + k_0$. Assume that all the intermediate values in Algorithm 1 are obtained. Let $Q_0(i)$ and $Q_1(i)$ be the intermediate values of Q_0 and Q_1 at the end of loop i in Algorithm 1, respectively.

Assume also that $k_{m-1}, k_{m-2}, \cdots, k_{i+1}$ are already deciphered. An attacker tries to reveal the next bit k_i. In this case, if and only if $k_i = 0$, either $Q_0(i-1)$ or $Q_1(i-1)$ is equal to Eq. (2) below:

$$\left(\sum_{j=i}^{m-1} k_j 2^{j-i+1} + 1 \right) P. \tag{2}$$

Similarly, if and only if $k_i = 1$, either $Q_0(i-1)$ or $Q_1(i-1)$ is equal to Eq. (3) below:

$$\left(\sum_{j=i}^{m-1} k_j 2^{j-i+1} + 3 \right) P. \tag{3}$$

In Ref. 25), differential power analysis attack is proposed based on the above ECC properties. Notice that, $Q_0(i-1) \neq Q_1(i-1)$ for any $1 \leq i \leq m-1$ and that $Q_0(i-1) \neq Q_0(j-1)$ and $Q_1(i-1) \neq Q_1(j-1)$ for $1 \leq i, j \leq m-1$ and $i \neq j$.

Table 1 Intermediate values at the end of i-th loop of Algorithm 1 with input P and $k = 10_{10}$.

i	Q_0	Q_1
3	P	$2P$
2	$2P$	$3P$
1	$5P$	$6P$
0	$10P^{*1}$	$11P$

*1: The result of the point multiplication.

Based on the above discussion, we employ $V(i)$ defined by Eq. (4) as a *selective function*:

$$V(i) = \sum_{j=i}^{m-1} k_j 2^{j-i+1} + 1. \tag{4}$$

When using the selective function above, we have to know $k_{m-1}, k_{m-2}, \cdots, k_{i+1}$. In addition to that, we assume that $k_i = 0$. $V(i) \neq V(j)$ always holds true for $i \neq j$ for $1 \leq i, j \leq m-1$. Given a point P over the elliptic curve E and $k_{m-1}, k_{m-2}, \cdots, k_{i+1}$, we assume that $k_i = 0$ and check whether $V(i)P$ appears somewhere in intermediate values. If it appears in them, we determine k_i as zero. If not, we determine k_i as one.

Finally, the LSB of a secret key k is determined by using the final point multiplication result. Since a point multiplication result $Q = kP$ is a public key itself, it must be obtained easily.

Example 1 Let us consider that the 4-bit secret key $k = 10_{10} = 1010_2$, i.e., $k_3 = 1$, $k_2 = 0$, $k_1 = 1$, $k_0 = 0$, and $m = 4$ but assume that we do not know k except for its bit length. The intermediate values $Q_0(i)$ and $Q_1(i)$ in Algorithm 1 are summarized in **Table 1**.

Now we try to decipher the 4-bit secret key k using intermediate values. Since we know that k has four bits, k can be written as $k = $ xxxx, where x shows the unknown bit. In Algorithm 1, MSB of k is defined by one. Then $k = \underline{\mathbf{1}}$xxx.

Next we try to decipher the second bit k_2 ($i = 2$) of k. The MSB of k is one by definition ($k_3 = 1$). We assume here that $k_2 = 0$. Then $V(1)$ is calculated as $V(1) = 5$. Since $5P$ appears in Table 1, then k_2 is deciphered as zero, i.e., $k = 1\underline{\mathbf{0}}$xx.

After that we try to decipher the third bit k_1 ($i = 1$) of k. We have already

known that $k_3 = 1$ and $k_2 = 0$. We assume here that $k_1 = 0$. $V(0)$ is calculated as $V(0) = 9$. Since $9P$ does not appear in Table 1, then k_1 is deciphered as one, i.e., $k = 10\underline{\mathbf{1}}$x.

Finally, we can have the point multiplication result $10P$ as in Table 1. If $k = 101\underline{\mathbf{0}}$, then $kP = 10P$. If $k = 101\underline{\mathbf{1}}$, then $kP = 11P$. Since the result is $10P$, then we can have $k = 101\underline{\mathbf{0}}$.

3.2 Problems to Decipher a Secret Key Using a Scan Path

If we decipher an m-bit secret key using an exhaustive search, we have to try 2^m possible values to do it. On the other hand, the method explained in Section 3.1 deciphers a secret key one-bit by one-bit from MSB to LSB. It tries at most $2\,m$ possible values to decipher an m-bit secret key. Further, the method just checks whether $V(i)P$ is in intermediate values of Algorithm 1.

In order to apply this method to a scan-based attack, we have to know which registers store intermediate values, i.e., we have to know correspondence between scanned data and (Q_0, Q_1).

However, a scan path is usually designed automatically by CAD tools so that nearby registers are connected together to shorten the scan path length. Only designers can know the correspondence between scanned data and registers and thus retrieved scanned data can be considered to be "random" for attackers. Therefore, it is very difficult to find out the values of $V(i)P$ in scanned data for attackers. As indicated before, an ECC circuit have very complicated architecture, its scan path can include too many registers other than those storing intermediate values.

We have to find out only $V(i)P$ somehow in the scanned data to decipher a secret key k using the method in Section 3.1.

4. Analysis Scanned Data Obtained from an ECC Circuit

In order to solve the problem that attackers do not know the correspondence between registers of the scanned data and ones storing intermediate values during point multiplication, we focus on the general property on a scan path below:

Property 1 A *bit position* of a particular register r in a scanned data when giving one input data is exactly the same as that when giving another input data.

This property is clearly true, since a scan path is fixed in an LSI chip and the order of connected registers in its scan path is unchanged.

If we execute point multiplication for each of n points on an ECC circuit, a bit pattern of a *particular* bit position in scanned data for these n points gives n-bit data. Based on the above property, this n-bit data also may give a bit pattern of a particular bit in some intermediate values when we give each of these n points to the ECC circuit.

By using the same n points we can calculate $V(i)P$ from k_{m-2} down to k_1 of the secret key k. By picking up a particular bit (LSB, for example) in each of $V(i)P$ values for n points, we also have an n-bit data. If n is large enough, this n-bit data gives information completely unique to $V(i)P$. We can use this n-bit data as a *discriminator* D_i to $V(i)P$ in scanned data.

Our main idea in this section is that we find out a discriminator D_i to $V(i)P$ in scanned data to decipher the secret key k from k_{m-2} down to k_1. If an n-bit discriminator D_i appears in the scanned data for n points, k_i is determined as zero. If not, it is determined as one.

In the rest of this section, we firstly propose a discriminator D_i to $V(i)P$. Secondly we propose an overall method to decipher a secret key k using discriminators. Thirdly we analyze the probabilities of successfully deciphering a secret key by using our method.

4.1 Calculating a Discriminator to $V(i)P$

Assume that n points $P_1, \cdots, P_n$ over the elliptic curve E are given. Also assume that we have already known $k_{m-2}, \cdots, k_{i+1}$ for a secret key k. Assuming that $k_i = 0$, we can calculate $V(i)P_r$ for $1 \le r \le n$. As **Fig. 6** shows, we define a *discriminator* D_i to be a set of LSBs of $V(i)P_r$ [⋆1]. If n is large enough, the discriminator D_i must give information unique to $V(i)P_r$ for $1 \le r \le n$. Consequently, if D_i appears in scanned data, k_i is determined as zero. If not, k_i is determined as one. After k_i is determined, we can continue to determine next bit of the secret key k in the same way.

Our proposed method has two advantages compared to conventional scan based attacks [1),2)]. One is that our method is effective in the case of partial scan archi-

⋆1 Since $V(i)P_r$ shows the point in XZ-plane, it has its X-coordinate and Z-coordinate. In our method, we just pick up LSB of its X-coordinate as in Fig. 6.

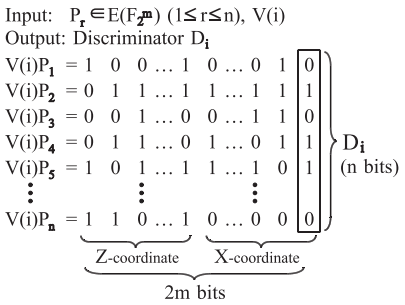

Fig. 6 Discriminator D_i.

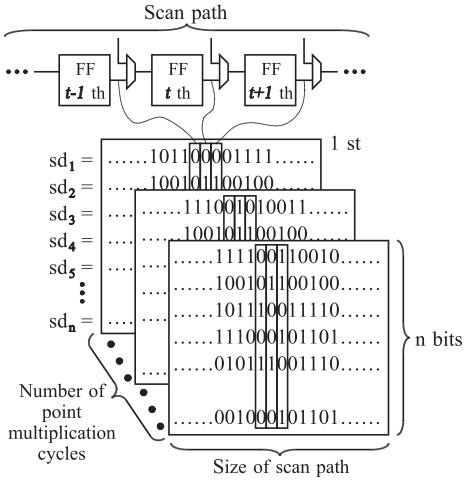

Fig. 7 Scanned data.

tecture. As long as a scan path includes at least 1-bit of each intermediate value, we can check whether the discriminator whether exists or not in the scanned data.

The other is that our method can crack the secure scan technique by Ref. 26), which inserts inverters into the internal scan path to complicate the scan structure. It protects Yang's method [1),2)] with low area cost. However, the value of a 1-bit register sequence is only changed to its inverted value. The variation of scanned data obtained by Ref. 26) is not enough to prevent our proposed method from deciphering a secret key. The detailed discussion will be described in Section 5.4.

4.2 Scanned Data Analysis Method

First we prepare n points $P_1, \cdots, P_n$ over the elliptic curve E and give them to an ECC circuit. For each of these points, we obtain all the scanned data from the scan out of the ECC circuit until the ECC circuit outputs the point multiplication result. As **Fig. 7** shows, the size of scanned data for each of these points is ("scan path length" × "number of point multiplication cycles".)

Now we check whether a discriminator D_i to $V(i)P$ appears in the obtained scanned data under the assumption that we do not know a secret key k in the ECC circuit as follows:

(1) Prepare n points $P_1, P_2, \cdots, P_n \in E(\mathbb{F}_{2^m})$, where $P_r \neq P_s$ for $1 \le r, s \le n$ and $r \neq s$.

(2) Input P_r $(1 \le r \le n)$ into the target ECC circuit and obtain scanned data every one cycle during point multiplication until the ECC circuit outputs the result. Let sd_r denote the obtained scanned data for the point P_r $(1 \le r \le n)$.

(3) Calculate $V(m-2)P_r$ assuming $k_{m-2} = 0$ for each P_r $(1 \le r \le n)$ and obtain the discriminator D_{m-2} to $V(m-2)P_r$.

(4) Check whether the discriminator D_{m-2} exists in the scanned data $sd_1, \cdots, sd_n$. If it exists, then we can find out that k_{m-2} is equal to 0, and if it does not exist, then we can find out that k_{m-2} is equal to 1.

(5) We can determine $k_{m-3}, k_{m-4}, \cdots, k_1$ in the same way as Step 4.

(6) k_0 (LSB of a secret key k) is determined by comparing the expected kP value with the point multiplication result outputted by the ECC circuit.

We show the example below to explain how the method above works.

Example 2 As in Example 1, let us consider that the 4-bit secret key $k = 10_{10} = 1010_2$, i.e., $k_3 = 1$, $k_2 = 0$, $k_1 = 1$, $k_0 = 0$, and $m = 4$ but assume that we do not know k except for its bit length and $k_3 = 1$. k can be written as $k = \underline{\mathbf{1}}$xxx, where x shows an unknown bit. Assume that the cycle counts of point multiplication are 4 and the size of the scan path is 62 in the target ECC circuit. First we prepare 8 points $P_1, P_2, \cdots, P_8 \in E(\mathbb{F}_{2^4})$, where $P_r \neq P_s$ for $1 \le r, s \le 8$ and $r \neq s$. The target ECC circuit executes the point multiplication as in Table 1. We input P_r $(1 \le r \le 8)$ into the target ECC circuit and obtain scanned data every one cycle during point multiplication until the ECC circuit outputs the result. Let sd_r denote the obtained scanned data for the point P_r $(1 \le r \le 8)$. The total size of scanned data is $4 \times 62 = 248$ (see **Fig. 8**).

The MSB of k is one by definition ($k_3 = 1$). Let us start to determine k_2. We calculate $V(2)P_r = 5P_r$ assuming $k_2 = 0$ for each P_r $(1 \le r \le 8)$ and obtain the discriminator D_2 to $5P_r$ (see **Fig. 9**). As Fig. 9 shows, the discriminator D_2 becomes "10011011". Since we find out that the discriminator D_2 exists in bit patterns of scanned data $sd_r (1 \le r \le 8)$ in Fig. 8, we can determine that k_2 is

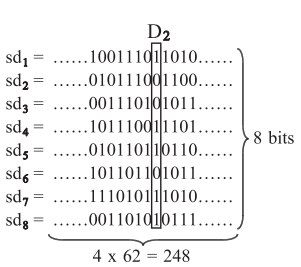

Fig. 8 Scanned data example.

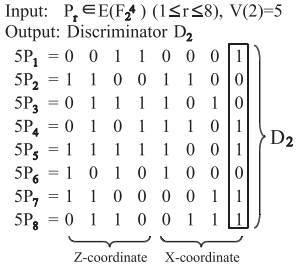

Fig. 9 Discriminator D_2.

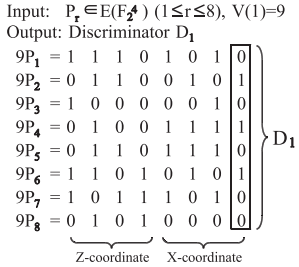

Fig. 10 Discriminator D_1.

equal to zero, i.e., $k = 1\underline{\mathbf{0}}$xx.

Next let us determine k_1. We calculate $V(1)P_r = 9P_r$ assuming $k_1 = 0$ for each P_r $(1 \le r \le 8)$ and obtain the discriminator D_1 to $9P_r$ (see **Fig. 10**). As Fig. 10 shows, the discriminator D_1 becomes "01010100". Since we find out that the discriminator D_1 does not exist in bit patterns of scanned data $sd_r (1 \le r \le 8)$ in Fig. 8, we can determine that k_1 is equal to one, i.e., $k = 10\underline{\mathbf{1}}$x.

Finally let us determine k_0. If $k = 101\underline{\mathbf{0}}$, then $kP = 10P$. If $k = 101\underline{\mathbf{1}}$, then $kP = 11P$. We calculate $10P_1$ and $11P_1$ and compare each of them with the point multiplication result kP_1. The point multiplication result obtained by the ECC circuit is $10P_1$ and we can determine that k_0 is equal to zero, i.e., $k = 101\underline{\mathbf{0}}$. Therefore we can decipher the secret key $k = 10_{10} = 1010_2$.

4.3 Possibility of Successfully Deciphering a Secret Key

Given that the scan size is α bits and the cycle counts to obtain point multiplication is T. Assume that scanned data are completely random data.

Even though $V(i)P_r$ for $1 \le r \le n$ is not calculated in the target ECC architecture, its discriminator may exist in scanned data. When $\alpha T < 2^n$, the probability that the discriminator D_i to $V(i)P_r$ exists in somewhere in bit patterns of scanned data sd_r $(1 \le r \le n)$ is $\alpha T/2^n$ despite $V(i)P_r$ does not calculate.

Sufficiently large n can decrease the probability that we mistakenly find out the discriminator D_i in scanned data. For instance, If α is 2,520, T is 15,137, and n is 32 ⋆1, then the probability that we mistakenly find out the discriminator D_i in scanned data is $2520 \times 15137/2^{32} \simeq 8.88 \times 10^{-3}$, which is low enough. If α is 25,200, T is 15,137, and n is 36, then the probability that we mistakenly find out the discriminator D_i in scanned data is $25200 \times 15137/2^{36} \simeq 5.55 \times 10^{-3}$, which is also low enough.

⋆1 These values are derived from the experiments in Section 4.

5. Experiments and Performance Analysis

Let us analyze the number of points n required to decipher a secret key k by using our proposed method. n must be large enough to be unique to $V(i)P_r$ $(1 \le r \le n)$. But it must be small enough to make deciphering time as short as possible.

In this section, we decipher some secret keys in the practical ECC architecture to determine the appropriate number of points n by using our method. We generate randomly 1,000 secret keys and decipher each of them. Then we calculate the number of points required to correctly decipher the secret keys.

5.1 Architecture of an Elliptic Curve Cryptography Circuit

Block diagram of the target ECC architecture for our scan-based attack is shown as in **Fig. 11** and **Fig. 12**. Its architecture is based on Refs. 11), 27) and it executes point multiplication using the López's method [22], an improved version of the Montgomery method. The method requires only one inversion and reduces the number of the multiplications compared with other point multiplication algorithm. The ECC architecture has an adder, a multiplier and a square unit over $\mathbb{F}_{2^m}$. These computing units can operate in parallel so that they can improve throughput effectively. Registers are used for input data, temporary data, and parameters for ECC. The ECC architecture also has registers for a secret key k and attackers cannot access these registers directly. In this ECC architecture, its secret key k can be set to be an arbitrary value beforehand.

We have designed the ECC architecture in Verilog HDL and synthesized it using Synopsys Design Compiler A-2007.12-SP3 with STARC 90 nm process library. A scan path has been implemented automatically using Synopsys DFT Compiler. We have obtained scanned data from the gate-level ECC circuit using HDL simulator Synopsys VCS-MX B-2008.12 ⋆2.

⋆2 This work is supported by VLSI Design and Education Center (VDEC), the University of Tokyo in the collaboration with Synopsys Corporation and with STARC.

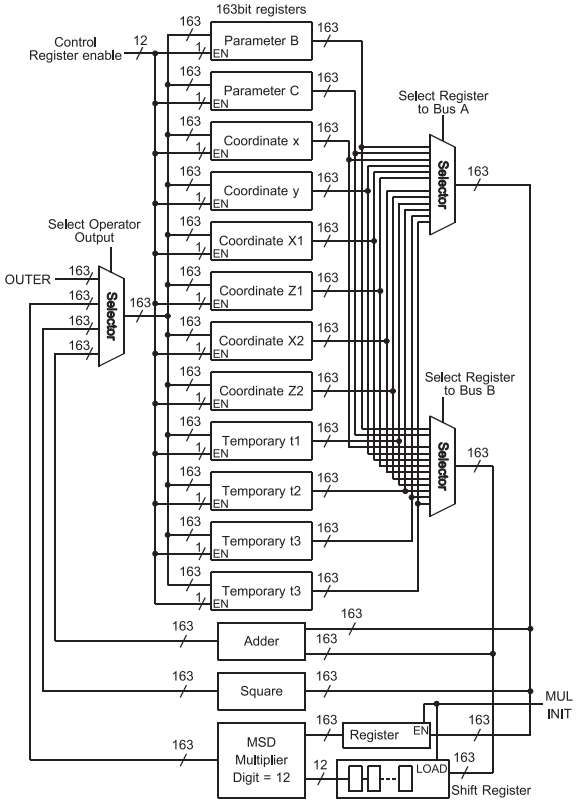

Fig. 11 Block diagram of the elliptic curve cryptosystem (Data path).

The implementation result indicates that the delay time is 1.66 ns, the area is 32.5 k gates and the total number of registers is 2,520 bits. Using this ECC architecture, the point multiplication requires 15,137 cycles.

5.2 Target Scan Path Architecture

For simplicity, the scan path used by our experiment just includes all the registers in the target ECC architecture. This means that it also includes the shift registers storing the secret key and registers for the controller in our experiment. However, we assume that attackers just attack scanned data in the data path in the ECC circuit. This is because of the following reasons:

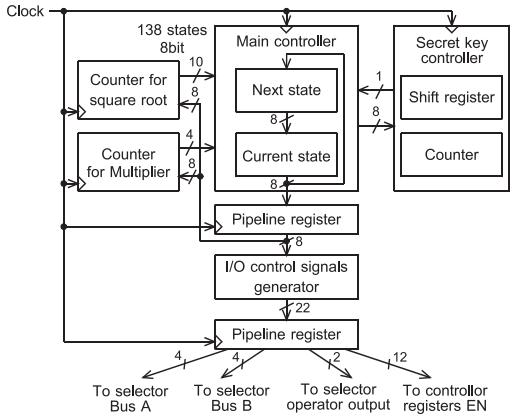

Fig. 12 Block diagram of the elliptic curve cryptosystem (Controller).

A controller architecture depends on implementation approach and is essentially unrelated to cryptography algorithm. For example, our ECC circuit uses a state machine as a controller but the ECC architecture in Ref. 8) uses a user-configurable circuit as a controller. Unlike cryptosystem algorithm, the controller architecture does not have to be open, and it is very hard for attackers to know what kind of controllers are used in a cryptosystem circuit. On the other hand, a modern cryptosystem algorithm has to be open to check its security and we need to know it to realize a secure communication. Attackers can easily know cryptosystem algorithm used by a target cryptosystem LSI.

Our proposed attacking method is based on an ECC algorithm and attackers know its algorithm using a target ECC LSI much easier than its controller architecture. We can say that scan-based attacks analyzing a data path is more practical than those analyzing a controller.

5.3 Results

We have implemented the analysis method proposed in Section 4 in C on the SuSE Linux 9, Intel Xeon 3.4 GHz, and 4 GB memories and performed the following experiments.

First, we have generated 1,000 secret keys randomly. Each of the generated secret keys has a bit length of 163. Next, we have given each of the 1,000 secret

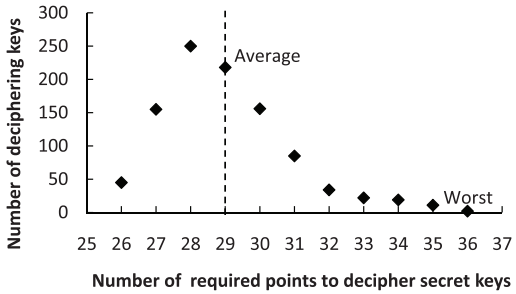

Fig. 13 Number of required points to decipher secret keys.

Table 2 The experimental results.

Key bit length *bit*	163
Number of deciphering keys	1,000
Number of required points (Average)	29
Number of required points (Worst)	36
Deciphering time *second*	≤40

keys into the target ECC circuit and obtained scanned data. Total size of the obtained scanned data for each secret key is 2,520 × 15,137 = 38,145,240 bits. Using these scanned data, we have deciphered each of the secret keys by using our proposed analysis method. **Figure 13** and **Table 2** show the deciphering results. Figure 13 shows a histogram which demonstrates the number n of required points to decipher each secret key versus its frequency. For example, the 572th secret key is 0x7e5f91be081095bf9eb1bc5d1e46f0001cb1d7b32. In order to decipher this secret key, we need 28 points, i.e., n is 28. In this case, we can successfully decipher the 572nd secret key using 28 points but fail to decipher it using 27 points or less. Throughout this experiment, the required number of points is 29 on average and 36 in the worst case. A deciphering time is at most 40 seconds when analyzing one secret key.

5.4 Discussions

Some secure scan architecture without consideration of a 1-bit sequence which is specific to some intermediate values cannot protect against our method. Here, we consider secure scan architecture against our proposed scan-based attack proposed so far.

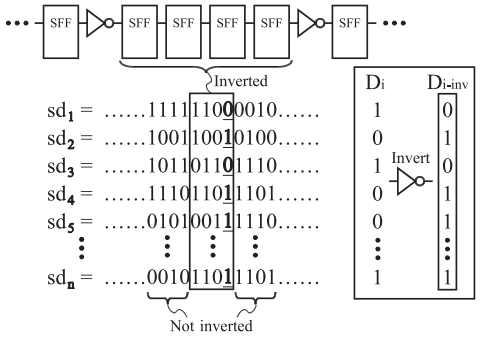

Fig. 14 Scanned data modified by Ref. 26).

Firstly, the most straightforward method against our proposed scan-based attack is to keep scan path open after testing the chip. However, scan path can be reconnected and be accessed by cracking the package [28].

Secondly, the secure scan architecture proposed in Ref. 26) cannot protect against our proposed method from deciphering a secret key. Reference 26) inserts some inverters into a scan path to invert scanned data as shown in **Fig. 14**. However, since the value of a 1-bit register sequence is only changed to its inverted value, the variation of scanned data is not enough to prevent attackers from checking whether the discriminator exists or not. For instance, assume that the discriminator D_i is $10100\ldots1$ and we check whether the discriminator D_i exists or not in the scanned data $sd_1, sd_2, \cdots, sd_n$ modified by Ref. 26) as shown in Fig. 14. If the discriminator D_i exists in the modified scanned data, we can successfully find out that k_i is zero. If not, we check whether the inverted discriminator $D_{i-inv} = 01011\ldots1$ exists or not. If the inverted discriminator D_{i-inv} exists in the modified scanned data, we can find out that k_i is zero. If the inverted discriminator D_{i-inv} does not exist, we can find out that k_i is one.

Reference 29) adds unrelated data to scanned data to confuse attackers as shown in **Fig. 15**. However, a sequence of scanned data to which unrelated data are added is fixed in each LSI chip and it just confuses only a part of scanned data to achieve lower area overhead. In other words, unmodified bits exist in the scanned data $sd_1, sd_2, \cdots, sd_n$ modified by Ref. 29). In this case, if the discriminator D_i exists in the modified scanned data, we can successfully find

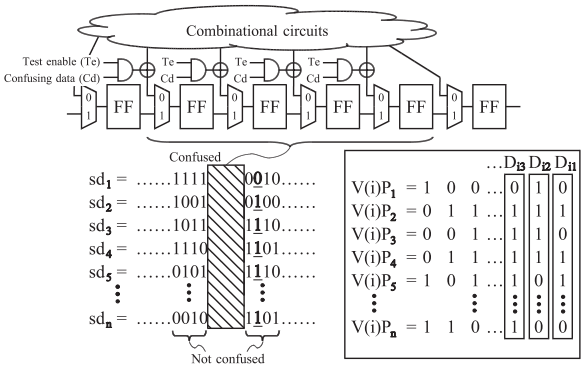

Fig. 15 Scanned data modified by Ref. 29).

out that k_i is zero. If not, we check whether the disicriminator D_i^1 calculated when k_i is one exists or not in the modified scanned data because a discriminator is defined as not only when k_i is zero but also when k_i is one. If the discriminator D_i^1 exists in the modified scanned data, we can successfully find out that k_i is one. Even if these discriminators do not exist in the modified scanned data, we can use other discriminators like $D_{i1}, D_{i2}, \cdots$ as shown in Fig. 15, which are defined as a set of other bits of $V(i)Pr$ for $1 \leq r \leq n$. If one of these other discriminators exists in the modified scanned data, we can find out that k_i is zero. Consequently, Ref. 29) cannot completely protect against our method.

Thirdly, Refs. 30)–37) require authentication to transfer between system mode and test mode, and their security depends on authentication methods. If authentication would be broken-through and attackers could obtain scanned data, a secret key in an ECC LSI could be deciphered by using our proposed method. We consider that authentication strength is a different issue from the purpose of this paper.

Yang's method [2] limits transition between test mode and system mode to prevent attackers from obtaining scanned data during encryption/decryption using the secret key in their cryptography circuit. However, it could not support in-field testing required for high reliable LSI.

Finally, Refs. 38)–40) use a compactor so as not to output scanned data corresponding to registers directly. Reference 41) proposes AES-based BIST, whereby there is no need for scan path test. However, applying these methods effectively to an ECC LSI is quite unclear because these methods implement only an AES circuit or just a sample circuit not for cryptography.

6. Conclusions

In this paper, we have focused on a scan-based attack against an ECC circuit. Three scan-based attacks against symmetric-key cryptosystems are reported [1)–3)] but those against public key cryptosystem are not reported yet. Since public-key cryptosystem are more complex than symmetric-key cryptosystem, scan-based attacks against symmetric-key cryptography cannot directly applied to decipher a secret key in public-key cryptosystem circuit.

Our proposed scan-based attack can effectively decipher a secret key k in an ECC circuit, since we just focus on the variation of 1-bit of intermediate values. By monitoring it in the scan path, we can find out the register position specific to intermediate values. The experimental results demonstrate that a secret key in a practical ECC circuit architecture can be deciphered by using 29 points over the elliptic curve E within 40 seconds. We can say that the proposed method reveals the vulnerability of a scan path in an ECC circuit.

In this paper, we deal with an elliptic curve cryptosystem over $GF(2^m)$. But even if we deal with an elliptic curve cryptosystem over $GF(p)$, where p is prime, the intermediate values during the point multiplication are determined by its inputs and a secret key, and consequently, our proposed method can decipher a secret key in the similar way.

Suppose that we attack other public key cryptography algorithms using our scan-based attack. For example, we pick up RSA, which is one of the most important public key cryptography algorithms. It encrypts and decrypts data with modular multiplication. Since its intermediate values may depend on a secret key, our scan-based attack method might be able to decipher its secret key in an RSA LSI.

However, architecture of RSA is quite different from that of ECC and we have a wide variety of its architecture compared with ECC. It may be very hard to know whether specific information unique to a secret key exists or not, unlike the discriminator of our scan-based attack. Applying our scan-based attack method

to other public key cryptography LSI is one of our future works.

Our future works are summarized as follows:

(1) New scan-based attack against compressed scan data.
(2) Countermeasures against the proposed scan-based attacking method.
(3) New scan-based attack against multiple scan paths.
(4) Experimental evaluation using our proposed method against scanned data modified by Refs. 26), 29).
(5) New scan-based attack by analyzing scanned data obtained from a controller circuit.

Acknowledgments This research was supported in part by Waseda University Grant for Special Research Projects (Nos. 2009A-504 and 2010B-222) and by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), 21560370. This work was partially supported by Secom Science and Technology Foundation and Kayamori Foundation of Information Science Advancement.

References

1) 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).
2) Yang, B., Wu, K. and Karri, R.: Secure Scan: A Design-for-test Architecture for Crypto Chips, *IEEE Trans. Comput. Aided Des. Integrated Circuits and Systems*, Vol.25, No.10, pp.2287–2293 (2006).
3) 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 (2009).
4) Miller, V.: Uses of Elliptic Curves in Cryptography, *The Advances in Cryptology*, Williams, H.(ed.), pp.417–426 (1986).
5) Koblitz, N.: Elliptic Curve Cryptosystems, *Mathematics of Computation*, Vol.48, pp.203–209 (1987).
6) Beth, T. and Gollmann, D.: Algorithm Engineering for Public Key algorithms, *IEEE Journal of Selected Areas in Communications*, Vol.7, pp.458–465 (1989).
7) Song, L. and Parhi, K.K.: Low Energy Digit-serial/parallel Finite Field Multipliers, *Journal VLSI Signal Processing*, Vol.19, No.2, pp.149–166 (1998).
8) Satoh, A. and Takano, K.: A Scalable Dual-field Elliptic Curve Cryptographic Processor, *IEEE Trans. Comput.*, Vol.52, No.4, pp.449–460 (2003).
9) Kumar, S., Wollinger, T. and Paar, C.: Optimum Digit Serial GF (2^m) Multipliers for Curve-based Cryptography, *IEEE Trans. Comput.*, Vol.55, pp.1306–1311 (2006).
10) Sakiyama, K., Batina, L., Preneel, B. and Verbauwhede, I.: Multicore Curve-based Cryptoprocessor with Reconfigurable Modular Arithmetic Logic Units over GF (2^n), *IEEE Trans. Comput.*, Vol.56, No.9, pp.1269–1282 (2007).
11) Orlando, G. and Paar, C.: A High-performance Reconfigurable Elliptic Curve Processor for GF (2^m), *Proc. Workshop on Cryptographic Hardware and Embedded Systems*, LNCS 1965, pp.41–56 (2000).
12) Gura, N., Shantz, S.C., Eberle, H., Gupta, S., Gupta, V., Finchelstein, D., Goupy, E. and Stebila, D.: An End-to-end Systems Approach to Elliptic Curve Cryptography, *Proc. Workshop on Cryptographic Hardware and Embedded Systems*, LNCS 2523, pp.349–365 (2002).
13) Eberle, H., Gura, N. and Chang-Shantz, S.: A Cryptographic Processor for Arbitrary Elliptic Curves over GF (2^m), *Proc. IEEE International Conference on Application-Specific Systems, Architectures*, pp.444–454 (2003).
14) Cohen, A.E. and Parhi, K.K.: Implementation of Scalable Elliptic Curve Cryptosystem Crypto-accelerators for GF (2^m), *Proc. Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, pp.471–477 (2004).
15) Moon, S.: A 193-bit Encryption Processor for Elliptic Curve Cryptosystem using Fast VLSI Algorithms in Finite Fields, *Proc. Consumer Communications and Networking Conference*, pp.611–613 (2005).
16) Chelton, W.N. and Benaissa, M.: Fast Elliptic Curve Cryptography on FPGA, *IEEE Trans. Very Large Scale Integration* (*VLSI*) Systems, Vol.16, No.2, pp.198–205 (2008).
17) Nara, R., Togawa, N., Yanagisawa, M. and Ohtsuki, T.: Scan-Based Attack against Elliptic Curve Cryptosystems, *Proc. IEEE Asia South Pacific Design Auto Conference*, pp.407–412 (2010).
18) Rivest, R.L., Shamir, A. and Adelman, L.: A Method for Obtaining Digital Signature and Public-key Cryptsystems, *Comm. ACM*, Vol.21, pp.120–126 (1978).
19) Montgomery, P.L.: Speeding the Pollard and Elliptic Curve Methods for Factorizations, *Mathematics of Computation*, Vol.48, pp.243–264 (1987).
20) Coron, J.-S.: Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems, *Workshop on Cryptographic Hardware and Embedded Systems Proceedings*, pp.292–302 (1999).
21) Agnew, G.B., Mullin, R.C. and Vanstone, S.A.: An Implementation of Elliptic Curve Cryptosystems over $\mathbb{F}_{2^{155}}$, *Journal IEEE*, Vol.11, No.5, pp.804–813 (1993).
22) López, J. and Dahab, R.: Fast Multiplication on Elliptic Curves over GF(2^m) without Precomputation, *The First International Workshop on Cryptographic Hardware and Embedded Systems Proceedings* LNCS 1717, pp.316–327 (1999).
23) Okeya, K., Kurumatani, H. and Sakurai, K.: Elliptic Curve with the Montgomery

Form and Their Cryptographic Applications, *The International Conference on Theory and Practice of Public-Key Cryptography Proceedings*, LNCS 1751, pp.238–257 (2000).
24) Okeya, K. and Sakurai, K.: Efficient Elliptic Curve Cryptosystems from a Scalar Multiplication Algorithm with Recovery of the y-coordinate on a Montgomery-form Elliptic Curve, *Workshop on Cryptographic Hardware and Embedded Systems Proceedings*, LNCS 2162, pp.126–124 (2001).
25) Goubin, L.: A Refined Power-analysis Attack on Elliptic Curve Cryptosystems, *The International Conference on Theory and Practice of Public-Key Cryptography Proceedings*, LNCS 2567, pp.199–211 (2003).
26) Sengar, G., Mukhopadhyay, D. and Chowdhury, D.R.: Secured Flipped Scan-chain Model for Crypto-architecture, *IEEE Trans. Very Large Scale Integration System*, Vol.26, No.11, pp.2080–2084, (2007).
27) Daley, W.M. and Kammer, R.G.: Digital signature standard (DSS), *Federal Information Processing Standards Publication* (*FIPS*), PUB 186-2 (2000).
28) Kömmerling, O. and Kuhn, M.G.: Design Principles for Tamper-Resistant Smartcard Processors, *Proc. USENIX Workshop on Smartcard Technology* (*Smartcard '99*), 1999.
29) Inoue, M., Yoneda, T., Hasegawa, M. and Fujiwara, H.: Partial Scan Approach for Secret Information Protection, *Proc. European Test Symposium*, pp.143–148 (2009).
30) Hely, D., Bancel, F., Flottes, M.L. and Rouzeyre, B.: Test Control for Secure Scan Designs, *Proc. European Test Symposium*, pp.190–195 (2005).
31) Gomułkiewicz, M., Nikodem, M. and Tomczak, T.: Low-cost and Universal Secure Scan: A Design- Architecture for Crypto Chips, *Proc. International Conference on Dependability of Computer Systems*, pp.282–288 (2006).
32) Hely, D., Bancel, F., Flottes, M.L. and Rouzeyre, B.: Secure Scan Techniques: A Comparison, *Proc. 12th IEEE International On-Line Testing Symposium*, p.6 (2006).
33) Lee, J., Tehranipoor, M. and Plusquellic, J.: A Low-cost Solution for Protecting IPs against Scan-based Side-channel Attacks, *Proc. 24th IEEE VLSI Test Symposium*, pp.94–99 (2006).
34) Hely, D., Bancel, F., Flottes, M.L. and Rouzeyre, B.: Securing Scan Control in Crypto Chips, *Journal of Electron Test*, pp.457–464 (2007).
35) Lee, J., Tehranipoor, M., Patel, C. and Plusquellic, J.: Securing Designs against Scan-Based Side-Channel Attacks, *IEEE Trans. Dependable and Secure Computing*, pp.325–336 (2007).
36) Paul, S., Chakraborty, R.S. and Bhunia, S.: Vim-scan: A Low Overhead Scan Design Approach for Protection of Secret Key in Scan-based Secure Chips, *Proc. 25th IEEE VLSI Test Symmposium* pp.455–460 (2007).
37) Chandran, U. and Zhao, D.: SS-KTC: A High-Testability Low-Overhead Scan Architecture with Multi-Level Security Intergration, *Proc. 27th IEEE VLSI Test Symposium*, pp.321–326 (2009).
38) Mukhopadhyay, D., Banerjee, S., RoyChowdhury, D. and Bhattacharya, B.B.: CryptoScan: A Secured Scan Chain Architecture, *Proc. 14th Asian Test Symposium*, pp.348–358 (2005).
39) Sengar, G., Mukhopadhyay, D. and RoyChowdhury, D.: An Efficient Approach to Develop Secure Scan Tree for Crypto-Hardware, *Proc. 15th International Conference of Advanced Computing and Communications*, pp.21–26 (2007).
40) Shi, Y., Togawa, N., Yanagisawa, M. and Ohtsuki, T.: A Secure Test Technique for Pipelined Advanced Encryption Standard, *IEICE Trans. Inf. Syst.*, Vol.E91–D, No.3, pp.776–780 (2008).
41) Doulcier, M., Flottes, M.L. and Rouzeyre, B.: AES-based BIST: Self-test, Test Pattern Generation and Signature Analysis, *Proc. 25th IEEE VLSI Test Symposium*, pp.94–99 (2007).

(Received May 11, 2010)
(Revised August 23, 2010)
(Accepted October 22, 2010)
(Released February 8, 2011)

(Recommended by Associate Editor: *Xiaoqing Wen*)

Ryuta Nara received his B. Eng. and M. Eng. degrees from Waseda University in 2005 and 2007, respectively, all in electronics and communication engineering. He is presently working toward Ph.D. degree there. His research interests are VLSI design and cryptography architecture. He is a member of IEICE.

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 VLSI design, graph theory, and computational geometry. He is a member of IEEE and IEICE.

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, VLSI design and verification, and network analysis and design. He is a member of IEEE, ACM, and IEICE.

Tatsuo Ohtsuki received his B. Eng., M. Eng., Dr. Eng. degrees from Waseda University in 1963, 1965 and 1970, respectively, all in electrical engineering. In 1965, he joined the NEC Corporation Ltd., Tokyo, Japan. From 1978 to 1980, he served as a Research Manager, Application System Research Laboratory, at Central Research Laboratories. In 1980, he left NEC and Joined Waseda University, where he is presently a Professor in the Department of Computer Science and Engineering. His research interests are algorithm and hardware engines for VLSI design and verification, computer algorithms for combinatorial problems, and network analysis/design. He is a fellow of IEEE, and a fellow of IEICE.