Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

# Why current secure scan designs fail and how to fix them?[☆]

Aijiao Cui[a,*], Yanhui Luo[b], Huawei Li[c], Gang Qu[d]

[a] *School of Electronic and Information Engineering, Harbin Institute of Technology Shenzhen Graduate School, China*
[b] *Advanced Micro Devices, Inc. (AMD), Shanghai, China*
[c] *SKLCA, Institute of Computing Technology, Chinese Academy of Sciences, China*
[d] *Department of Electrical and Computer Engineering, University of Maryland College Park, USA*

ARTICLE INFO

*Keywords:*
Scan-based side-channel attacks
Secure scan design
Cryptographic chip
Cipher key

ABSTRACT

Scan design has become another side channel of leaking confidential information inside cryptographic chips. Methods based on obfuscating scan chain order have been proposed as countermeasures for such scan-based attacks. In this paper, we first analyze the existing secure scan designs from the angle that whether they need a complete chain state or rely on any specific scan chain order. We show that all existing attacks do not rely on specific scan chain order and therefore any secure scan design with obfuscated scan chain order cannot provide sufficient security. We then propose a new approach which clears the states of all sensitive scan cells whenever the circuit under test is switched to test mode. It will also block the access to cipher key throughout the entire testing process. Our experimental results show that the proposed scan design can effectively insulate all the information related to cipher key from the scan chain with little design overhead, thus it can successfully defend all the existing scan-based attacks.

## 1. Introduction

Scan design has been regarded as the best design-for-testability (DFT) discipline as it can significantly improve the controllability and observability of the design under test (DUT). However, it is also a double-edged sword when it is used by attackers as a side channel to pry the confidential information residing in the DUT. The scan-based side-channel attack was first proposed in [1] to deduce the cipher key used in the cryptographic chip of data encryption standard (DES). New attack scenario was later proposed to break advanced encryption standard (AES) [2]. These two attack scenarios both assumed that the scan chain contains only scan cells related to the encryption unit. Unfortunately, this assumption is invalid in real chips because registers used by other components such as memories, processor, and control logic are also included in the scan chain. Some attack scenarios were proposed towards the NTRUEncrypt ctypto-system [3] and the linear feedback shift register (LFSR) based stream ciphers [4]. Signature attacks were proposed to attack the encryption algorithms of AES [5], DES [6], ECC [7] and RSA [8]. In these attacks, the target signature is divided into several segments and then the trial-and-error scheme is applied to break each segment.

To protect cryptographic chips from the above scan-based attacks, many countermeasures were proposed [2], [9], [14–17]. They can be classified into three categories [12]: inherent countermeasures, countermeasures against micro probing and protocol countermeasures. The inherent.

countermeasures use advanced DFT structures or the build-in self-test (BIST) design as a protective shield against scan-based attacks. Advanced DFT structures, such as response compactor and input decompressor, are considered to resist scan-based attacks inherently [9]. The work in [10–13] proposed enhanced signature attacks with differential analysis of the intermediate encrypted results, which the advanced DFT structures fail to protect [13]. BIST can guarantee high security [12], but at the cost of lower diagnostic resolution and higher area overhead, which limits its application in real design.

In micro probing attacks, the malicious attackers will use physical means, such as probe and wire, to control the working manner of scan design to gain the observability of its states. Countermeasure against micro probing was first proposed in [14]. It will ring an alarm when unauthorized access of scan enable signal is detected physically.

Protocol countermeasures have received a lot of research attention. Among them, methods based on scrambling and access restriction are believed to be the most effective and have been implemented in various forms. Many methods under this scenario adopt the “lock and key”

[☆] This work was supported in part by the National Natural Science Foundation of China under Grant 61672182, Guangdong Natural Science Foundation under Grant 2016A030313662, Shenzhen Overseas High-Level Talent Innovation Foundation under Grant KQJSCX20160226202510 and Shenzhen Fundamental Science Research Foundation under Grant JCYJ20150403161923508. Huawei Li was supported by the National Natural Science Foundation of China under Grant 61432017 and 61532017. Gang Qu was partially supported by AFOSR MURI under award number FA9550-14-1-0351.

[*] Corresponding author.

http://dx.doi.org/10.1016/j.vlsi.2016.10.011

Available online 20 October 2016

scheme, where a "lock" is inserted in the scan chain so that only the authenticated user with the correct "key" can unlock it and use the scan chain to perform the normal testing. On the other hand, without the authenticated key, any user or malicious attacker cannot use the scan design for testing or attacking. To implement the "lock" scheme, [15] and [16] proposed to divide the original scan chain into several sub chains and the order to connect these sub chains is determined by the "key". When the "key" is authenticated, sub chains will be connected in a correct order and ready for test. Otherwise, these sub chains' connection order becomes unpredictable and the scan out data will be random, not revealing directly any information to the attackers.

In this paper, we analyze the existing countermeasures for scan-based attacks and then focus on those based on obfuscating scan chain order. We discover their vulnerabilities and propose a new countermeasure that blocks the cipher key to enter encryption. We find that access to the complete states of scan chain is sufficient for an attack to be successful. None of the attacks relies on the information of specific scan chain order and thus existing countermeasures based on obfuscating scan chain order will not be effective. To overcome this, we propose a new method which clears the contents related to encryption in the sensitive scan cells when the DUT enters test mode and the cipher key will then be blocked throughout the entire testing process. Hence, none of the information related to cipher key will be observable to anyone during testing through scan chain and all scan-based attacks will fail.

The rest of the paper is organized as follows. In Section 2, we review the related work on countermeasures. In Section 3, we analyze the relation between scan-based attacks and scan chain order and derive the key condition for successful scan-based attacks. Two countermeasures based on obfuscating scan chain order are then analyzed. In Section 4, we propose a new secure scan design based on blocking cipher key and analyze its security and performance with the experimental results. The paper is concluded in Section 5.

## 2. Countermeasures against scan-based attacks

Existing countermeasures against scan-based attacks include those based on blocking cipher key [2] and those based on obfuscating the scan output [10], [11], [13], [15], [16], [18].

In [2], a secure scan design scheme based on mirror key registers (MKRs) was proposed to block cipher key to resist attacks. This scheme introduces two working modes as insecure mode and secure mode. Under insecure mode, the cipher key is isolated from entering cryptographic module by the inserted MKRs. It can be released for encryption only under secure mode. This design method can resist both mode-switching attacks [5] and test-mode-only attacks [6] by blocking cipher key under insecure mode. However, a power-reset operation is needed once the circuit is switched from secure mode to insecure mode. This disables the online testing and hence limits the application of the scheme.

All existing scan-based attacks rely on observing scan chain to obtain the intermediate encrypted result and retrieve the cipher key consequently. Several countermeasures are hence proposed to obfuscate the scan output to impede the retrieval of cipher key.

In paper [18], state dependent scan flip-flops (SDSFFs) are introduced to defend against scan-based attacks and its structure is illustrated in Fig. 1. The inserted latches between two scan cells are used to memorize a historic state of its former SFF. These latches can also change the current scan output response by performing exclusive-or operation between the current response value and the historic value. The '*load*' signal is used to control whether the latches are updated by the state of its former SFF. The generation of '*load*' signal is completed by using the states of some specific SFFs which will change dynamically during the shifting phase under test mode. Both the structure of control circuitry for '*load*' and the values of such specific SFFs are unknown to attackers. Therefore, they cannot obtain the original scan output to deduce the cipher key and scan-based attacks can hence be defeated.

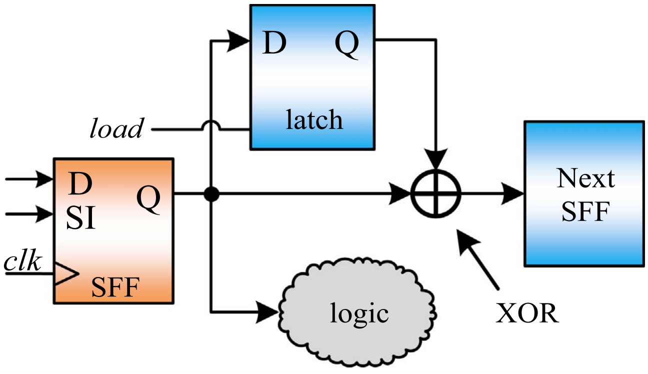

**Fig. 1.** State dependent scan flip-flops.

However, the testability of DUT is reduced. Since the control circuity for '*load*' signal is determined by the states of some specific SFFs, a possible fault in such SFFs will result in a wrong control signal for '*load*'. This results in unexpected change in output responses. Normal testing could be affected and the original fault coverage cannot be guaranteed. Also, it was not elaborated in [18] how a test engineer can deduce the original output response from the obfuscated scan output data after the introduction of SDSFFs.

The work in [19] proposed a countermeasure based on static obfuscation of scan data. A shift register is introduced to control the working mode of some specific scan cells and only when a correct user key is entered, can the scan chain be configured in the correct way and work normally. Otherwise, the scan chain simply unloads erratic data and the cipher key cannot be deduced successfully. However, it has been reported that specific signature attack can tick out the incorrectly configured scan cells one by one in order to deduce the cipher key.

The work in [25] proposed to implement a secure scan chain design, which introduced inverters at random positions in the scan chain. The security relies on that an attacker cannot guess the positions of the inverters with a probability significantly greater than 1/2. However, it was explored [26] that these positions can be easily identified by a reset operation on the scan chain under test mode. Hence, the scheme is vulnerable. It was proposed in [27] to improve the compactor for scan tree design to prevent the attacker to access the normal output responses from scan design. The compactor is designed to output the comparision result between each part of one output response and the correct one instead of the real output response. However, it was neglected that besides the referred counters will be introduced for the operation of comparison, lots of memeory are required to store all the correct responses, which will be accessed during test for comparison. For a scan design with 1000 scan cells and 500 test vectors, 0.5 M bits memory is needed. As a real design usually has a longer scan chain which needs more test data for fault detection, lots of memories are needed. This will results in a great overhead for the proposed secure scan design. The work in [26] proposed to insert XOR gates in scan design to obfuscate the scanout data. However, it was explored in [28] that by carefully observing the scanned out vector, an attacker can find out the positions where the XOR gates have been inserted. Thereafter, a double-feedback XOR-chain scheme [28] was proposed. The cryptanalysis work in [29] showed that by performing a sequence of operations as reset, insertion and de-assert of scan-enable signals in scan chain, an attacker can also deduce the locations of inserted XOR gates. To address these security concerns, the authors in [29] proposed to add a multiplexer at the end of every double-feedback XOR gate [28] so that the data to the scan cell appended after double-feedback XOR can be either the XOR result or the previous scan cell. This is determined by a control bit of an internal key. Physically unclonable function (PUF) was proposed to be used for the generation of such random key. One concern about this approach is that the PUF response can only be obtained after manufacturing, and cannot be accessed directly after chip is taped out (otherwise an attacker can also access the PUF response). Therefore the designers have to rely on the manufacturer to detect this PUF response and provide to them (the designers). This increases the design cost and brings the nature trust problem where the manufacturer may leak such confidential information intentionally or accidentally. Without a satisfactory solution to this problem, the random XOR-chain structure proposed in [29] cannot secure the scan chain design.

In all existing scan-based attack scenarios, attackers are assumed not to have any knowledge about the details of the structure of scan chain. This is a reasonable assumption. To analyze the encrypted result, some attack scenarios [1–4] assume that the attackers can identify the correspondence between encryption registers and the scan chain. Several countermeasures have hence been proposed to interfere the identification of such correspondence, for example, by obfuscating the scan chain order [15], [16]. We now study this in more details.

## 3. Scan-based attacks and scan chain order

In this section, we first use an example to illustrate the relation between the attacks [1–4] and scan chain order. We conclude that these attacks do not rely on any specific scan chain order, which can be verified by an example of attack on AES encryption. Then, we discuss signature attacks to show that such attacks even do not need any information on scan chain order. Finally, the vulnerability of two related methods are discussed.

### 3.1. Scan-based attacks and obfuscation of scan order

Suppose that a cryptographic chip has an originally ordered scan chain of 12 scan registers $S = \{s_i\}_{i=1}^{12}$ where six of them $R = \{r_i\}_{i=1}^{6}$ are used for the storage of encryption result. As shown in Fig. 2, assuming that the six registers in $R$ form the following subset of $S$, $SR=\{s_2, s_7, s_3, s_8, s_5, s_{10}\}$. We denote this bijection between $R$ and $SR$ by $SR=f(R)$. To countermeasure possible attacks, the scan chain $S$ is obfuscated to $OS = \{os_i\}_{i=1}^{12}$ by a permutation and the registers in $SR$ become $OR=\{os_1, os_5, os_4, os_7, os_3, os_{10}\}$, respectively. Denote this permutation (also a bijection) between $SR$ and $OR$ by $OR = g\ (SR)$. Thus, the relation between $R$ and $OR$ can be expressed as $OR=h(R)=g\ (f\ (R))$. Finding $h$ is a necessary condition to launch scan-based attacks [1–4].

When the scan chain is not obfuscated, an attacker can find the relation $f$, even without the knowledge of the original scan chain order. This can be done by multiple trials with different plaintext inputs. After obfuscation, due to the exponential number of permutations, it becomes difficult to determine $g$ and $f$. Countermeasures to secure the scan chain based on this were proposed in [15] and [16]. However, they did not consider that when the complete encrypted result is obtained from a scan chain, the attacks in [1–4] can reveal function $h$, which is the composition of $g$ and $f$, directly by multiple trials with input of different plaintext. This process is similar to finding $f$ without obfuscation. Next, we use the attack on AES encryption in [2] as an example to demonstrate this.

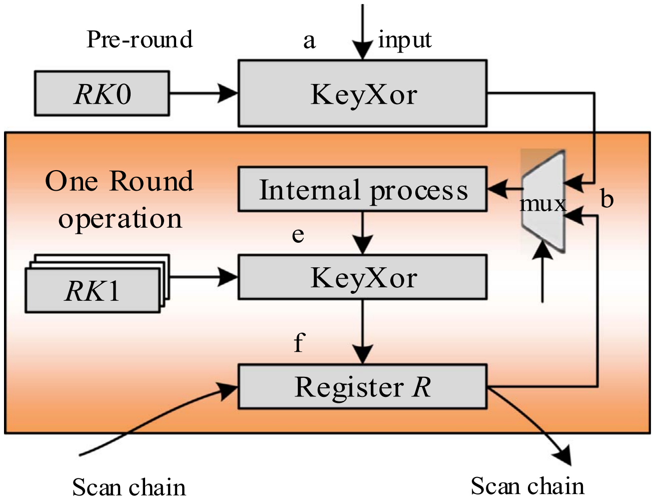

**Fig. 3.** Round operations in iterative AES encryption.

### 3.2. An illustrative example

Fig. 3 shows the hardware implementation of one round in an iterative AES architecture. In the first clock cycle, the plaintext is input from the point ‘a’ and the result after pre-round using the key of *RK*0 and round 1 with the key of *RK*1 is stored in Register *R*. A 128-bit exclusive-or operation is performed on plaintext in pre-round. The one-round operation includes several internal processes and the final exclusive-or operation by the KeyXor box. The output of Register *R* is the input to the next round operation and is fed back to the round operation through a multiplexer at point ‘b’. The AES round operation is repeated ten times and each time a different round key *RK*i is used to generate the cipher text. Round keys or the user secret key are the targets of an attacker.

To facilitate the discussion [2], a cipher key of 128-bit zero is assumed to be used in AES encryption. The 128-bit plaintext, which can be represented in hexadecimal as $(0000, 0000, 00F2, 0000, 0000, 0000, 0000, 0000)_{16}$ is input to the AES cryptographic module. Then, the hexadecimal value is obtained as $(24CF, EAEA, 0100, 0000, 0100, 0000, 0100, 0000)_{16}$ at Register *R* from the observed state of the scan chain as $S_1=(269F0, 10000, 00FE5, 20ABC, 01000, 00027, 68924, CFEAE, A72EF, D9010, 00000, A5412, F76E2, 87)_{16}$. Next, the plaintext is changed with one bit flipped to obtain $(0000, 0000, 00F3, 0000, 0000, 0000, 0000, 0000)_{16}$. With the new plaintext input, the value of $(B3DC, 6E6E, 0100, 0000, 0100, 0000, 0100, 0000)_{16}$ is obtained in register *R* from the value of $S_2$=(269F0, 10000, 00FE5,

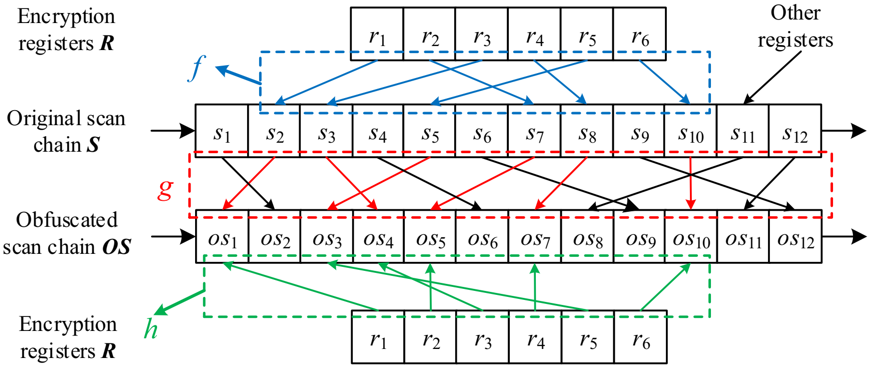

**Fig. 2.** Scan-based attack and scan chain order.

20ABC, 01000, 00027, 689B3, DC6E6, E72EF, D9010, 00000, A5412, F76E2, 87)$_{16}$ observed from the scan chain. The exclusive-or result of the two obtained states of the Register $R$ is (00000, 00000, 00000, 00000, 00000, 00000, 00**97, 13848, 4**0000, 00000, 00000, 00000, 00000, 00)$_{16}$ and the non-zero part of (97138484)$_{16}$ can be translated into a binary value of (1001, 0111, 0001, 0011, 1000, 0100, 1000, 0100)$_2$, which shows the Hamming distance between $S_1$ and $S_2$ is 12.

A 128-bit input plaintext is arranged as a 4×4 matrix of bytes with each $a_{i,j}$ ($0 \le i, j \le 3$) representing a byte. Key Xor operation is a bitwise exclusive-or of 128-bit round key with the data $b_{i,j}=a_{i,j} \oplus RK0_{i,j}$, $0\le i, j\le 3$. Then $b^1_{1,1}$ and $b^2_{1,1}$ are obtained as $(F2)_{16}$ and $(F3)_{16}$ according to the property of the AES encryption [2]. With $b^1_{1,1}$ and $b^2_{1,1}$, the $RK0_{1,1}$ can be recovered as $a^1_{1,1} \oplus b^1_{1,1} = a^2_{1,1} \oplus b^2_{1,1} = 0$. Similarly, all other $RK0_{i,j}$ can be recovered as 0. This is same as $RK0$ being used, which shows that the attack on AES is successful.

Now we discuss how to defend against this attack. Without loss of generality, we divide $S_1$ and $S_2$ into 14 excerpts (separated by comma as shown below) and reorder them with the following rule: (1→11), (2→1), (3→14), (4→4), (5→3), (6→7), (7→13), (8→10), (9→2), (10→5), (11→12), (12→6), (13→8), (14→9), where ($i$→$j$) dentoes that the $i$-th excerpt in $S_1$ ($S_2$) is assigned to the $j$-th excerpt in the obfuscated scan outputs $S_1'$ ($S_2'$). After obfuscation, $S_1'$ and $S_2'$ are obtained as (269F0, 10000, 00FE5, 20ABC, 01000, 00027, 68924, CFEAE, A72EF, D9010, 00000, A5412, F76E2, 87)$_{16}$ and (10000, E72EF, 01000, 20ABC, D9010, A5412, 00027, F76E2, 87, DC6E6, 269F0, 00000, 689B3, 00FE5)$_{16}$, respectively. The exclusive-or result of $S_1'$ and $S_2'$ is (00000, **4**0000, 00000, 00000, 00000, 00000, 00000, 00000, 00**138, 48**000, 00000, 00000, **97**000, 00)$_{16}$ and it can be translated as (0000, ..., 0100,..., 00010, 01110, 00010, 01000...10010, 1110...0)$_2$. It is found that the Hamming distance is also 12 and $b^1_{1,1}$ and $b^2_{1,1}$ can be similarly obtained as $(F2)_{16}$ and $(F3)_{16}$. Finally, $RK0_{1,1}$ can be recovered as 0. This example shows that the obfuscation of scan chain will not affect whether the round key used in AES encryption can be broken.

### 3.3. Signature attack and scan chain order

Scan-based attacks do not necessarily need the correspondence between encryption registers and scan chain. Signature attacks [5–8], [13] are examples of such attacks and they can be implemented without any information of the scan chain order.

Signature attacks use a simulator to simulate the function of the cryptographic chip by performing encryption with different input key at the software level. In Fig. 4, the left part shows the simulator and the right part shows a real cryptographic module. Since the encryption algorithm is known to the public, an attacker is assumed to have knowledge of the structure of the cipher key. To reveal the key, he will divide the cipher key into several sections according to the property of encryption algorithm. Such division should guarantee that the variation of each key section can result in corresponding variation of the encrypted result in some specific scan registers. Then, the trial-and-error scheme is used to deduce each key section. Such divide-and-conquer scheme can significantly reduce the difficulty in cracking.

The signature attack method in [6] can successfully retrieve the secret key in DES using less than 30 plaintexts on average and 32 plaintexts in the worst case. However, the attack scenario in [1] retrieves the secret key in DES with a scan chain of length 192 requires 67 plaintexts and 6 scan-in data in addition to several assumptions that need to be satisfied. This shows that the signature attack is more powerful than the method in [1] in terms of the requirement to launch a successful attack.

Suppose that the cipher key $K$ is divided into $k$ sections: $ks_1$, $ks_2$ ... $ks_k$. To determine $ks_i$ ($1\le i\le k$), the attacker will select $n$ specific plaintexts, which can invoke the role of $ks_i$ in encryption to feature in the related $r$-bit intermediate encryption result. Both the $n$ specific plaintexts and the key $KG$, which contains the figured key section, $fk_i$, are input to the simulator while the $n$ plaintexts are input to the real cryptographic chip simultaneously. Under $n$ plaintexts, the output from each encryption scan cell in simulator (the encryption scan cells are known to the attacker as it is designed by him) will form an $n$-bit stream and it is called a signature* corresponding to $fk_i$. Suppose there are $m$ ($1\le m$, and smaller $m$ corresponds a larger $n$) encryption scan cells related to $fk_i$ (called sensitive scan cells) and $m$ signatures* are hence generated. Correspondingly, the output from each scan cell in cryptographic module (obviously the encryption scan cells are unknown to the attacker) will form an $n$-bit signature. All the signatures from the cryptographic module will be checked to find whether there are $m$ signatures that can match $m$ signatures* from the simulator. If yes, the figured key section is correct, i.e., $fk_i$=$ks_i$. Otherwise, the attacker should alter $fk_i$ and repeat the above process until $fk_i$=$ks_i$. He will execute the trial-and-error process on each key section until all $k$ key sections are figured out.

Similar to the signature attack, the trace buffer attack in [30] is also based on the comparison of the simulation results between a real design and a test chip to determine the signals in trace buffers and use these signals to reconstruct other more relevant signals in order to retrieve the secret key. Since the signature attack and trace buffer attack do not need the knowledge of scan chain structure [6] or micro-architecture [30], they pose higher risk to cryptographic chips.

We can see that the signature attack can be performed without the need of any scan chain order information. The countermeasures based on obfuscation scan chain thus cannot be used to defend signature attacks. Also, an attacker can retrieve the key as long as all the sensitive scan cells, rather than the complete scan design, can be accessed.

Based on the above analysis, we conclude that the access to the complete state of scan chain is a necessary and sufficient condition for the attacks in [1–4] but it only serves as a sufficient condition for a successful signature attack. Also, the complete state of scan chain does not necessarily include the information on scan chain order. Obfuscation of the scan chain order hence cannot impede the attacks as granted.

### 3.4. Countermeasures based on obfuscating scan chain order

Based on the analysis above, we introduce two popular countermeasures based on obfuscating scan chain order in [15], [16] and analyze their respective vulnerabilities.

#### 3.4.1. Lock and key technique

A lock and key technique was proposed to secure a scan chain design as shown in Fig. 5. In this scheme, the original scan chain will be divided into $m=2^q-1$ sub chains with equal length of $l$ bits per sub chain. An extra test security controller (TSC) is introduced to manage the test sequence of these sub chains. It consists of four main parts: a finite state machine (FSM), a linear feedback shift register (LFSR), a test key comparator and a $q$-$m$ bit one-hot decoder.

The test procedure is performed as follows. First, a test key is fed to the test key comparator and will be compared with the stored correct key. Only when the key passes this comparison, can a user input a correct seed to initialize the LFSR to start the normal shifting. Otherwise, the LFSR will be randomly initialized and $r$ random redundant bits will be involved in the shifting process to enhance the unpredictability of the LFSR outputs. The one-hot decoder is used to translate the $q$-bit LFSR output into an $m$-bit one-hot code to enable one sub chain to work in one time slice. Only when the decoder receives normal output from the $q$-bit LFSR, can it manage all sub chains in the correct order.

The author of [15] claimed that such countermeasure can resist the scan-based attacks as the order of the sub chains is obfuscated without a correct key input. As we have analyzed above, a successful attack may not rely on any specific order, regardless whether the order is correct or not. In other words, the countermeasure in [15] cannot achieve its claimed security based on the obfuscated scan chain order.

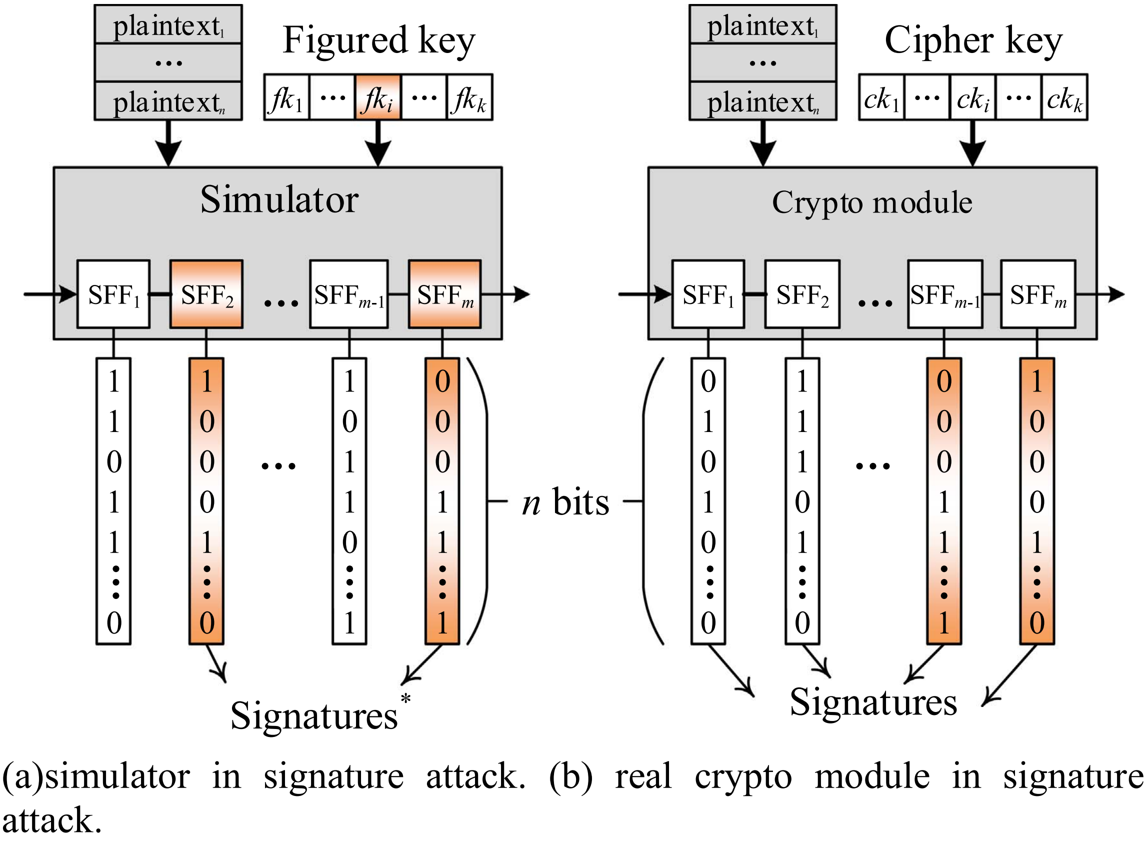

**Fig. 4.** Signature attack. (a) simulator in signature attack. (b) real cryptographic module in signature attack.

It is noted that the $q$-bit LFSR used in [15] is a primitive polynomial LFSR. The author of [15] noticed that such design is not secure due to the predictability of the LFSR's behavior. To avoid such predictability, the original $q$-bit modifiable LFSR is extended to a $(q+r)$-bit LFSR and the primitive polynomial LFSR becomes a non-primitive polynomial LFSR. The extended LFSR has been designed to make some sub chains need not to be accessed during the LFSR shifting process. Thus, a complete state of scan chain cannot be guaranteed to be always obtained. Based on the analysis in Section 3, the necessary and sufficient condition for the attacks in [1–4] will not hold because the complete output response cannot be accessed with the extended LFSR [15]. The countermeasure is hence secure against the attacks [1–4]. This is the main and only reason that the countermeasure can resist this type of scan-based attacks, not the fact that the scan chain order

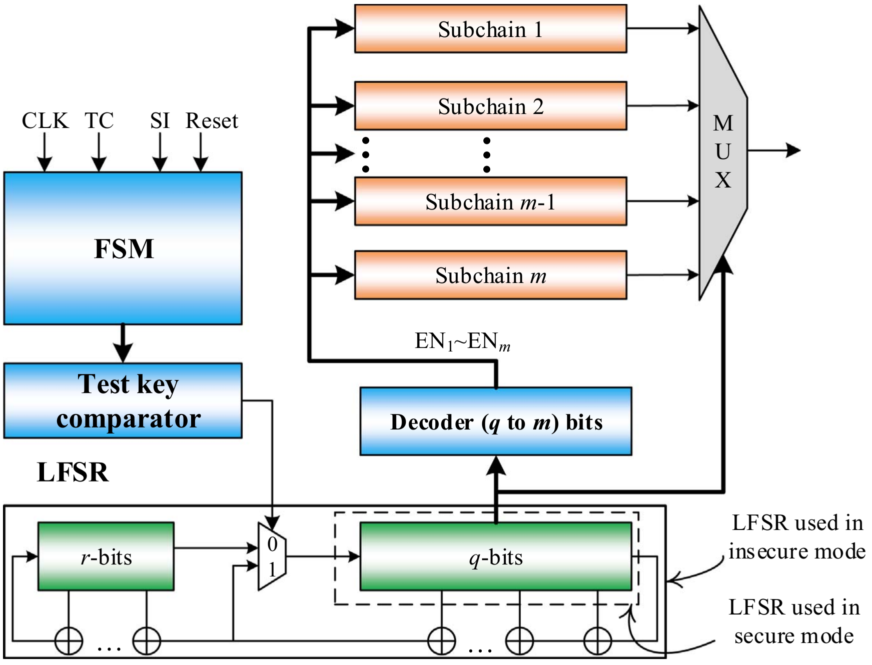

**Fig. 5.** Lock and Key technique.

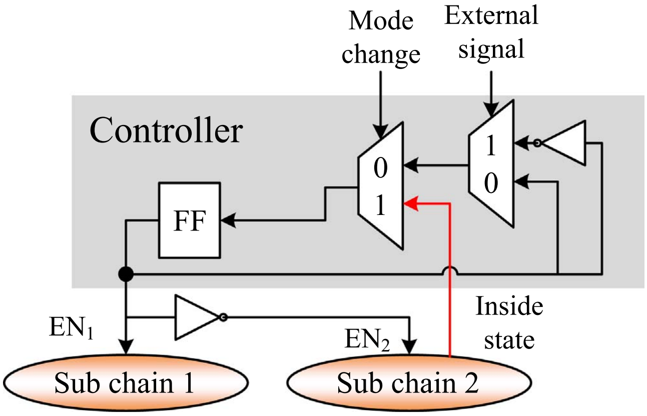

**Fig. 6.** Random order scan (ROS) controller.

obfuscated as suggested in [15].

As for the signature attack, if there are sensitive scan cells contained in the inaccessible sub chains, some key sections are destined to be unidentified. Thus, the key information cannot be obtained and the signature attack will fail. However, if all sensitive scan cells are accessible after the LFSR is extended, the key will still be retrieved by signature attack [6] as complete intermediate encryption states can be achieved through the accessible sensitive scan cells. Thus, for a DES module, the signature attack [6] can successfully retrieve the secret key in DES using just fewer than 30 plaintexts on average and no more than 32 plaintexts. Obviously, the more sub chains that are accessible to the attacker, the easier it is to retrieve the secret key. In this sense, the lock and key technique in [15] is not secure enough against the signature attacks.

### 3.4.2. Dynamically configurable connection technique

A secure scan design based on dynamically configurable connection technique, random order scan was proposed in [16]. Similar to the Lock & Key technique, the original scan chain is first divided into $n$ sub scan chains. Next, as shown in Fig. 6, two sub chains form a chain-pair and each sub chain is gated by the original clock and an enabling signal. A controller is used to provide the enabling signals. In this controller, a flip-flop, denoted by FF, is introduced to generate the enabling signals for each chain-pair. Its initial state is determined by the status of a specific SFF in the chain-pair. We can see that one sub chain will be enabled to perform testing. When it is done, the external signal will present “1” to transit the inverted FF history value to FF. As the FF state alters, the other sub chain is stimulated for testing.

The same controller design will be applied to determine the order between two chain-pairs, i.e., the order between two controllers. Likewise, the order among all sub chains can be determined by a hierarchical binary tree of controllers. For example, a scan design with six sub chains can be manipulated by three levels of controllers, as shown in Fig. 7. The label in each controller indicates its level and an index. For instance, 1–2 means the second controller at level 1.

When an unauthorized user, or an attacker, does not have the knowledge about how the order is determined by the internal SFF, he is thought not to be able to input the correct external signal or run testing normally [16]. However, it is also noted that the design scheme in [16] guarantees that each sub chain can be accessed by different control of the external signals although the accessing order among the sub chains is obfuscated, which enables the complete encrypted result to be accessed from the scan design. Based on the analysis in Section III, the attacks in [1–4] and the signature attacks could succeed against this proposed scan design because the sufficient condition (that the attacker has access to the complete state of scan chain) is satisfied. The countermeasure in [16] is hence deemed not secure. We explore an attack scenario in next section which can defeat this countermeasure.

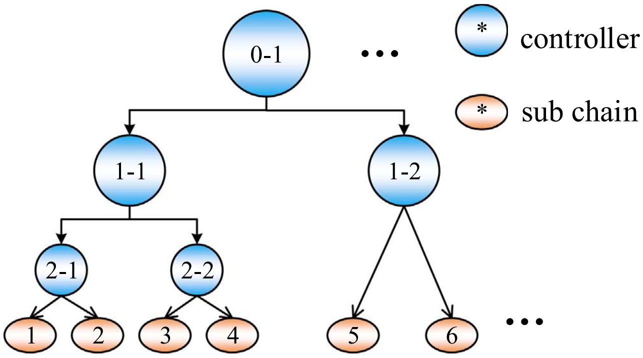

**Fig. 7.** Hierarchical tree of controllers in ROS.

### 3.4.3. An attack scheme on ROS countermeasure

We first outline the overall approach of the proposed attack and then elaborate its most challenging steps.

*3.4.3.1. Overview of the proposed attack.* We propose an attack against the countermeasure in [16] with the following four steps:

**STEP 1**. Identify the length of sub chain.

Although an attacker has no knowledge on the scan design detail of a cryptographic chip, he can coarsely estimate the number of scan cells of the chip, $n$ and the minimum number of the sub chains, $p$ with his experience. The length of the sub chain, $l$ is reasonably assumed to be unknown for an attacker. So the first challenge is to identify this unknown. The solution is non-trivial and we will discuss in detail in the next subsection.

**STEP 2**. Collect encrypted result from each sub chain.

When the length of sub chain is determined, attacker can properly manage the external control signals to collect the intermediate encryption results from each sub chain. He can use all the combinations of the external control signals to gain access to the encrypted result from each sub chain.

**STEP 3.** Remove the redundant information.

As discussed above, the ROS design scheme introduces a hierarchical binary tree of controllers. Suppose the tree has $r$ levels. The tree can control up to $2^r$ sub chains. In real scan design, the number of sub chains, $p$ is not necessarily to be $2^r$. It can be any number between $(2^{r-1}, 2^r)$. However, to access the state of each sub chain, one needs to cover all the $2^r$ combinations of the external control signals. If $p$=$2^r$, then each sub chain will be accessed just once and there will be no redundant information loaded from the scan chain. Otherwise, when $p < 2^r$, some sub chains will be accessed multiple times, which results in redundant excerpts. Such redundancy can be easily identified and then removed so that there will not be any duplicate excerpts remained.

**STEP 4.** Retrieve the secret key.

After the above three steps, an attacker will have collected all the irredundant information from the scan chain. He can then use this to retrieve the key with the attack schemes in [1–4], [6].

*3.4.3.2. Identify the length of sub chain.* Recall that the number of scan cells of the chip is $n$, the minimum number of the sub chains is $p$ and the attacker wants to determine the length of the sub chain is $l$.

Note that after an $l$-bit output response is completely loaded out from one sub chain, the last bit of the output response will continue to be loaded out until the external control signal inverts and the sub chain is no longer active. Therefore, we will observe a tail of consecutive bits with the same value. To facilitate our discussion below, we call the bit stream output from a sub chain when the sub chain is active an ***excerpt***. As we have just discussed, an excerpt will always have a tail of bit with the same value, as shown in Fig. 8.

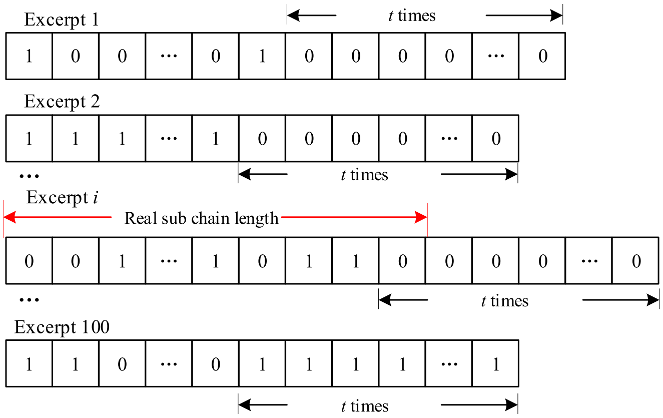

**Fig. 8.** Identify the length of scan chain in ROS design.

The following procedure illustrates how the value of sub chain length $l$ can be identified by constructing and analyzing multiple excerpts.

1) Set the external control signals and select a sub chain *sc*.
2) Collect the bit stream scanned out from the scan out pin. Maintain the external control signals to enable *sc* to be active until the output bit "1" or "0" is repeated for $t$ consecutive times. This gives one excerpt. ($t$ is a pre-determined number normally larger than $n/p$).
3) Reset the external control signals, change the test vector and repeat Steps 1) and 2) until we collect $q$ excerpts for sub chain *sc*. ($q$ is a pre-determined number).
4) Repeat steps 1)–3) to collect $q$ excerpts for each sub chains.
5) Among all the excerpts, find the one with maximum length $m$.
6) The length of sub chain is $l=m-t+1$.

The rationale behind this approach is that the $t$ consecutive 1's or 0's at the tail of each excerpt consists of none or more bits from the output response and the repeats of the last bit in the output response. To identify $l$, the length of sub chain, the attacker needs to distinguish the last bit in output response and its repeats. However, this is not a trivial problem. For example, consider an 8-bit excerpt with 5 0's as its tail, '10100000', the length of the sub chain may be 4 and the last 4 0's are the repeats. It is also possible that the length of sub chain is 7 and only the last 0 is a repeat. Even worse, the sub chain may be longer than 8 and it just happens to produce 5 consecutive 0's.

We take a probabilistic approach to solve this problem, as shown in the above procedure. Instead of one excerpt, we generate multiple of them, $q$ excerpts to be more specific. If one of these excerpts has two different values in its last two bits of the output response, that is '01' or '10', the last $t-1$ bits in the excerpt will be the repeat of the last bit of output response. If the length of this excerpt is $m$ bits, then it is easy to see that the length of the sub chain is $m-(t-1)$. Also note that in this case, the excerpt will be one of the longest.

So we only need to find the excerpt with the maximal length as specified in steps 5) and 6) in the proposed procedure. Take the 100 excerpts in Fig. 7 as an example. As the $i^{th}$ excerpt has the maximum length, the length of a sub chain can then be determined by the length of this excerpt.

Step 2) shows how to generate an excerpt. When $t$ consecutive bits of the same value appear in the output from a sub chain, the attacker should invert the external control signal to halt the operation of the current sub chain.

We conclude this section by a brief discussion on how to set the values for parameters $t$ and $q$. $t$ should be set to be larger than $n/p$ to avoid cases like $t$ consecutive 1's or 0's in the output response. When all the $q$ excerpts have '00' or '11' as the last two bits in the output response (not the last two bits in the excerpt), the length of the sub chain we compute from step 6) will be incorrect. However, this only happens with probability $1/2^q$, which decreases exponentially as $q$ increases. When $q > 20$, for instance, this probability becomes 9.54E-07. Hence, although large $q$ value will give more confidence, in practice, setting $q$=20 will be sufficient.

Following the above steps, an attacker will have collected all the irredundant information from the scan chain. He can then retrieve the key based on this information by applying the attack schemes in [1–4] or [6]. The difficulty of retrieving a key can hence be uniquely determined by the attack schemes. For example, using signature attack method in [6] to retrieve the secret key in DES needs fewer than 30 plaintexts on average while the attack scenario in [1] requires 67 plaintexts and 6 scan-in data to do the same.

## 4. Proposed secure design based on blocking cipher key

As some existing countermeasures based on obfuscating scan output are vulnerable to some known scan-based attacks, we propose a new countermeasure based on blocking cipher key which can resist all known scan-based attacks while incurring low overhead.

### 4.1. Secure design based on blocking cipher key

All existing scan-based attacks use the intermediate encryption result observed from scan chain to analyze and deduce the cipher key used in cryptographic module. If the observed scan response bares no relation with the cipher key or in other words, the cipher key is never involved in the scan output information, then scan chain will make no sense in deducing the cipher key. We hence propose a new secure scan design method to isolate the cipher key from encryption operation under test mode so as to resist the scan-based attack. Meanwhile, the intermediate encryption result captured by the scan chain when DUT is switched from normal mode to test mode will be cleared.

The proposed secure design is illustrated in Fig. 9. It mainly contains a scan chain and a controller. The controller works as a mode discriminator, which can identify the working mode of DUT between normal mode and testing mode. When DUT is working under normal mode, the '*load*' signal, which is used to monitor the data path of the cipher key, is set high (effective). The cipher key is enabled to pass normally to the cryptographic module to join in the encryption operation. Otherwise, DUT is working under testing mode and the path from the cipher key to the cryptographic module is cut off. The cipher key is hence insulated from the cryptographic module under testing. Meanwhile, upon TC=0 and DUT enters testing mode, the flip-flop FF in the controller will drive the '*clr*' signal to be high. This effective clear signal, which is unanimously connected to the *clr* port of each sensitive scan cell, will force the states of all sensitive scan cells to be zero. Thus, the confidential information loaded into scan chain

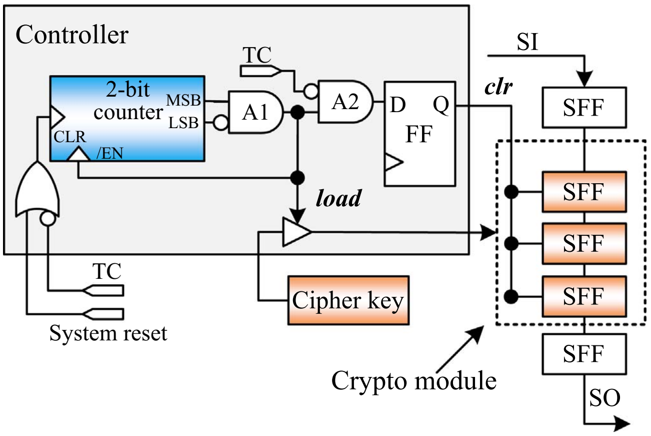

**Fig. 9.** Secure design based on blocking cipher key.

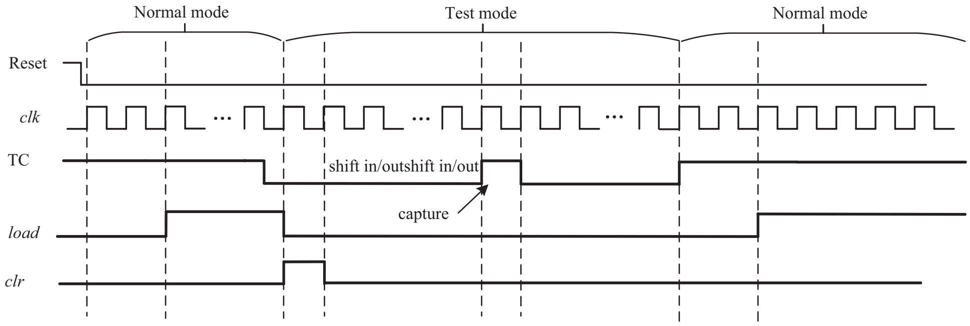

**Fig. 10.** Working procedure of the blocking key design.

under normal mode, if any, is cleared when the testing commences and the cipher key is kept insulated from the scan chain throughout testing. An attacker can never obtain any useful information related to cipher key by the side channel of scan chain (Fig. 9).

### 4.2. Working flow of the proposed design

The timing diagram of the controller with the internal signals transiting between normal mode and testing mode is shown in Fig. 10. When *TC* signal keeps high, the DUT works under the normal mode. The 'CLR' port of the 2-bit counter becomes low and counting starts from zero. After two clocks, the most significant bit (MSB) turns to '1' and the least significant bit (LSB) becomes '0'. Then, the AND gate A1 appended with the counter outputs '1'. The counting is halted as the '/EN' port of 2-bit counter becomes high. Meanwhile, the '*load*' signal is invoked to be high and the cipher key is released to be able to join in the normal encryption. As *TC* is high, the AND gate A2 outputs zero and the '*clr*' signal keeps low, which will not affect the state of the scan chain.

To perform a secure testing, the *TC* input is set low to switch the DUT into the testing mode. As *TC*=0, once the first rising edge of the clock arrives, the 2-bit counter is cleared with both its MSB and LSB forced to be zero. Then, the '*load*' signal presents low and the channel for cipher key entering encryption module is closed. Also, at the first clock after *TC*=0, the flip-flop FF connected to gate A2 will snapshot the state of '*load*' signal during the last clock period, '1'. Thus, the '*clr*' signal turns high for one clock and the content in all sensitive scan cells will be cleared. Thereafter, normal testing starts. During the capture phase with *TC* switched to high for one clock, the 2-bit counter will restart counting from zero and terminate after one clock as *TC* returns low. MSB and LSB turns to '0' and '1', respectively. The output of A1 gate will maintain low. Hence, both the '*clr*' and '*load*' signals keep low throughout the normal testing process. This guarantees the states of scan cells not to be cleared and the cipher key not to be accessed during scan testing.

As we can see from the working flow of the proposed blocking key design, the normal testing procedure is not affected as the cipher key is not necessarily to be invoked for encryption in test mode and all the states of the scan cells are kept well without being cleared. In addition, the cipher key is accessible under normal mode and the design function is hence guaranteed.

### 4.3. Analysis of security

The existing scan-based attacks can be classified as mode-switching attacks and test-mode-only attacks. The security of the proposed scheme can be analyzed under these two different attack scenarios.

In mode-switching attacks [5–8], [13], the DUT is first reset to an initial state. A known plaintext is fed into the cryptographic module for encryption and the result is captured into the scan registers in the functional mode. The DUT is then switched to the test mode and the encrypted results are shifted out for analysis. This forms one cracking cycle. Such attacks rely on the switching between normal working mode and test mode and it is unanimously assumed that the states of sensitive scan cells would maintain during the switching from normal mode to testing mode. The proposed secure scan design makes this assumption unsustainable since the contents in sensitive scan cells are all cleared upon the switching from normal mode to testing mode. Thus, the proposed design can resist the mode-switching attacks successfully.

Test-mode-only attacks [21–24] are attacks that can be conducted solely under the test mode. The known plaintext is shifted into the scan chain using the boundary scan cells in the test mode. During the capture phase, encryption is performed and the encrypted result is shifted directly out of the scan chain. After each cracking cycle, the DUT is reset to the initial state to scan in the next known plaintext under the test mode. It can be seen that such attacks rely.

on that the cipher key can be accessed in the capture phase under testing mode. In the proposed secure design, the cipher key is isolated from encryption operation throughout the test process. Thus, even an attacker can perform the encryption during the capture phase, he cannot obtain any meaningful information for retrieving the cipher key. The test-mode-only attacks can hence be defeated by the proposed design.

### 4.4. Analysis of testability

Both design-for-testability (DFT) and automatic test patterns generation (ATPG) are performed prior to the addition of the extra circuitry around the scan design. Such circuitry is transparent to the original test patterns. The effectiveness and efficiency of normal DFT and ATPG processes are not affected as the original test patterns are still applicable to the protected design with the same fault coverage achieved as the original unprotected design. The testability of the original design, i.e. the DUT without extra design for security, is not sacrificed at all. It is noted that the faults occurred in the extra circuitry are not testable. As only simple primitive logic gates, counters and registers are used to secure the scan design, they can be tested by BIST [31] with very high fault coverage and low overhead.

### 4.5. Analysis of overhead and comparison

In our experiment, the proposed secure design is implemented on the scan design for a pipelined AES design from opencores [20]. The original design was synthesized by Synopsys Design Compiler using the 'typical.lib' library to obtain the netlist. The constraint is set to optimize the design area. Scan chain is inserted into the netlist using Synopsys DFT Compiler. The proposed secure design is then inserted into the netlist of DUT with scan design and synthesized by Synopsys Design Compiler.

The areas of original circuit before and after scan insertion, which are represented by the number of equivalent two-input NAND gates

**Table 1**
Performance comparison among countermeasures.

| Design | $\Delta A$ (%) | Security | Testability | Other |
|---|---|---|---|---|
| SBK | 0.07 | NONE | Nil | 1 clock before test |
| SOSD-128[19] | 0.34 | Signature attack | Nil | 128 clocks before test |
| MKR[4] | 0.19 | NONE | Nil | online test is inapplicable |
| Mode reset[14] | ~10 | Test-mode-only attack | Nil | NA |
| TSC-8b [15] | 0.31 | Relies on accessibility of subchains | Nil | More clocks before test |
| ROS-16c [16] | 0.15 | Signature attack | Nil | More clocks before test |
| SDSFF-100[18] | 0.25 | NONE | compromised | NA |

with area as 13.3 μm$^2$, are 205934 and 212280, respectively. After the insertion of the proposed secure design, the area of the complete design equals to 212432 NAND gates. Hence, the area.

overhead due to the introduction of the proposed secure design is equivalent to 212432-212280=152 NAND gates and it just accounts for the percentage of 152/212280=0.07% of the original design.

We also compare the proposed secure scan design (abbreviated as SBK) with other countermeasures, SOSD-128 [19], MKR [2], mode-reset [14], TSC [15], ROS [16] and SDSFF [18] in Table 1. SOSD-128 refers to the secure design based on static obfuscation of scan data [18] with 128-bit shift registers for loading user key. TSC-8bits refers to the TSC design in [15] with 8-bit LFSR and a 64-bit test key. ROS-16c refers to the ROS design in [16] with 16 subchains. SDSFF-100 refers to the SDSFF design in [18] with 100 state-dependent SFFs, 1054 SFFs and a 5-bit test key.

These countermeasures are evaluated in terms of area overheads, security, testability and other possible affected performance. The area overhead (i.e., $\Delta A$) is represented by the percentage of extra gate counts due to secure design to that of the pipelined AES with scan design. Such percentages for the implementation of other countermeasures are excerpted directly from the related literature. It was noted that the mode-reset countermeasure [14] reported $\Delta A$ to be 10% without giving the exact gate counts or area of original scan design. Security is the most important merit and it is evaluated by demonstrating the scan-based attacks to which the countermeasures are vulnerable, if any. The maintenance of testability is another important merit to evaluate a countermeasure. If the testability of the protected design is affected much by the introduction of secure design, a countermeasure will lose its value no matter how secure it is. Besides, other downsides, such as test latency, or inability of online testing may also be possibly introduced by the countermeasures although they are not desired.

Compared with other secure designs, the area overheads due to the proposed SBK is negligibly low and minimum among all the referred countermeasures. As the cipher key is obstructed under the test mode, it cannot be tested by the scan design as other normal part of DUT. From the security viewpoint, the cipher key should be stored in a secure way so that none can access it easily without compromising the original design. In this sense, the SBK provides a shield for the cipher key from the scan-based side channel. The testability of the cipher key related circuitry should be guaranteed by the designer at the design phase by using specific memory or other specific test method after design. The original testability of other design parts is unaffected by the countermeasures and only 1 additional clock cycle is added to the total test latency for clearing the states in scan cells.

The SOSD-128 [19] is insecure against signature attack provided that the plaintext is loaded through primary inputs (PIs). The MKR scheme in [4] is secure, but online testing is inhibited in the secure scan design. The testability of the cipher key part can be similarly analyzed as the proposed SBK. Countermeasure based on mode reset [14] not only incurs high area overhead, but also is vulnerable to test-mode-only attack. The security of TSC design [15] depends largely on the accessibility of scan cells in the subchains and it is not always secure. Besides, the decoder size increases very rapidly with the length of its LFSR. ROS designs [16] were vulnerable to the signature attack. Both countermeasures of [15] and [16] will delay normal testing for more clock cycles to reorder the scan cells among multiple subchains. Although SDSFF [18] is secure, the original fault coverage cannot be maintained when there is any fault in the scan cells. This makes the scheme impractical as defects are unavoidable in manufacturing processes.

Overall, the scan design secured by our proposed SBK scheme outperforms other existing designs in one or more figures of merit with a negligibly low overhead on the design area and test time, and no compromise to the testability of original scan design.

## 5. Conclusion

In this paper, we analyze the security of the popular countermeasures against scan-based side-channel attack based on obfuscating scan chain order. We show that obfuscating scan chain order cannot secure the scan chain against scan-based attacks as they are expected. Instead, access to the complete states of scan chain is sufficient for an attack to be successful. Besides, signature attacks do not rely on any information of scan chain order. We propose a new secure design scheme based on blocking cipher key. By insulating the cipher key from encryption operation under test mode and clearing the states in scan chain at the start point of test, the countermeasure can resist all existing scan-based side-channel attacks as the scan chain cannot fetch any information related to cipher key. Also, the proposed secure design only incurs 0.07% area overhead, which is minimum among existing countermeasures. The testability of original design is not compromised and the normal testing is just delayed one clock cycle before commence.

## References

[1] Y. Bo, W. Kaijie, R. Karri, Scan-based side-channel attack on dedicated hardware implementations of data encryption standard, in: Proceedings of the International Test Conference (ITC), Washington DC, USA, October 2004, pp. 339–344.

[2] Y. Bo, W. Kaijie, R. Karri, Secure scan: a design-for-test architecture for crypto chips, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 25 (10) (2006) 2287–2293.

[3] A.A. Kamal, A.M. Youssef, A scan-based side channel attack on the NTRUEncrypt ctyptosystem, in: Proceedings of the IEEE 7th International Conference on Availability, Reliability and Security (ARES), Prague, Czech Republic, Aug. 2012, pp. 402–409.

[4] L. Yu, W. Kaijie, K. Ramesh, Scan-based attacks on linear feedback shift register based stream ciphers, ACM Trans. Design Auto. Elec. Syst. I 16 (2) (2013) 20:1–20:15.

[5] R. Nara, N. Togawa, M. Yanagisawa, T. Ohtsuki, A scan-based attack based on discriminators for AES cryptosystems, IEICE Trans. Fundam. Electron. Comm. Comput. Sci. E92-A (12) (2009) 3229–3237.

[6] H. Kodera, M. Yanagisawa, N. Togawa, Scan-based attack against DES cryptosystems using scan signatures, in: Proceedings of the Asia Pacific Conference Cir. Syst., Kaohsiung, Taiwan, December 2012, pp. 2–5.

[7] R. Nara, N. Togawa, M. Yanagisawa, T. Ohtsuki, Scan-based attack against elliptic curve cryptosystems, in: Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC), Taipei, Taiwan, January 2010, pp. 407–412.

[8] R. Nara, K. Satoh, M. Yanagisawa, N. Togawa, Scan-based side-channel attack against RSA cryptosystems using scan signatures, IEICE Trans. Fundam. Electron. Comm. Comput. Sci. E93-A (12) (2010) 2481–2489.

[9] C. Liu, Y. Huang, Effects of embedded decompression and compaction architectures on side-channel attack resistance, in: Proceedings of the IEEE VLSI Test Symposium (VTS), Berkeley, California, USA, May 2007, pp. 461–468.

[10] J.D. Rolt, G.D. Natale, M-L. Flottes, B. Rouzeyre, Scan attacks and countermeasures in presence of scan response compactors, in: Proceedings of the 16th IEEE European Test Symposium (ETS), Annecy, France, May 2011, pp. 19–24.

[11] J.D. Rolt, G.D. Natale, M. Flottes, B. Rouzeyre, Are advanced DFT structures sufficient for preventing scan-attacks? in: Proceedings of the IEEE VLSI Test Symposium (VTS), Maui, Hawaii, USA, April 2012, pp. 246–251.

[12] J.D. Rolt, A. Das, G.D. Natale, M.-L. Flottes, B. Rouzeyre, I. Verbauwhede, Test

versus security: past and present, IEEE Trans. Emerg. Top. Comput. 2 (1) (2014) 50–62.

[13] J.D. Rolt, G.D. Natale, M. Flottes, B. Rouzeyre, A novel differential scan attack on advanced DFT structures, ACM Trans. Des. Autom. Electron. Syst. 18 (4) (2013), 58.

[14] D. Hely, F. Bancel, M.-L. Flottes, B. Rouzeyre, Test control for secure scan designs, in: Proceedings of the European Test Symposium (ETS), Tallinn, Estonia, May 2005, pp. 190–195.

[15] J. Lee, M. Tehranipoor, C. Patel, J. Plusquellic, Securing designs against scan-based side-channel attacks, IEEE Trans. Dependable Secur. Comput. 4 (4) (2007) 325–336.

[16] Y. Atobe, Y. Shi, M. Yanagisawa, N. Togawa, Secure scan design with dynamically configurable connection, in: Proceedings of the 19th Pacific Rim International Symposium Dependable Computing (PRDC), Vancouver, Canada, December 2013, pp. 256–262.

[17] G.D. Natale, M. Doulcier, M.-L. Flottes, B. Rouzeyre, Self-test techniques for crypto-devices, IEEE Trans. Very Large Scale Integr. Syst. 18 (2) (2010) 329–333.

[18] Y. Atobe, Y. Shi, M. Yanagisawa, N. Togawa, State dependent scan flip-flop with key-based configuration against scan-based side-channel attack on RSA circuit, in: Proceedings of the Asia Pacific Conference on Cir. and Syst. (APCCAS), Kaohsiung, Taiwan, Dec. 2012, pp. 607–610.

[19] Y. Luo, A. Cui, G. Qu, H. Li, A new countermeasure against scan-based side-channel attacks, in: Proceedings of the IEEE International Symposium on Cir. and Syst. (ISCAS), Montreal, Canada, May 2016, pp. 1722–1725.

[20] AES: overview, [Online] Auguest 2014, Available: ⟨http://opencores.org/project, tiny_aes⟩

[21] S.S. Ali, S.M. Saeed, O. Sinanoglu, R. Karri, Scan attack in presence of mode-reset countermeasure, in: Proceedings of the International On-Line Testing Symposium (IOLTS), Santa Clara, CA, USA, July 2013, pp. 230–231.

[22] S.S. Ali, O. Sinanoglu, R. Karri, Test-mode-only scan attack using the boundary scan chain, in: Proceedings of the European Test Symposium (ETC), Paderborn, Germany, May 2014, pp. 1–6.

[23] S.S. Ali, O. Sinanoglu, S.M. Saeed, R. Karri, New scan attacks against state-of-the-art countermeasures and DFT, in: Proceedings of the IEEE International Workshop Hardware-Oriented Security Trust (HOST), Arlington, VA, USA, May 2014, pp. 142–147.

[24] S.S. Ali, S.M. Saeed, O. Sinanoglu, R. Karri, Novel Test-Mode-Only scan attack and countermeasure for compression-based scan architectures, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 34 (2015) 808–821.

[25] G. Sengar, D. Mukhopadhyay, D.R. Chowdhury, Secured flipped scan-chain model for crypto-architecture, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 26 (11) (2007) 2080–2084.

[26] M. Agrawal, S. Karmakar, D. Saha, D. Mukhopadhyay, Scan based side channel attacks on stream ciphers and their counter-measures, in: Proceedings of the 9th annual International Conference on Cryptology in India, Kharagpu, India, Dec. 2009, pp. 226–238.

[27] D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury, B. B. Bhattacharya, CryptoScan: a secured scan chain architecture, in: Proceedings of the 14th Asian Test Symposium (ATS'05), 2005, pp. 348–353.

[28] S. Banik, A. Choudhury, Improved scan-chain based attacks and related counter-measures, INDOCRYPT 2013, LNCS, Springer, Heidelberg, vol. 8250, pp. 78–97.

[29] S. Banik, A. Chattopadhyay, A. Chowdhury, Cryptanalysis of the double-feedback XOR-chain scheme proposed in Indocrypt 2013, in: Proceedings of the 15th International Conference on Cryptology in India, New Delhi, Indis, Dec. 2014, pp. 179–196.

[30] Y. Huang, A. Chattopadhyay, P. Mishra, Trace buffer attack: security versus observability study in post-silicon debug, in: Proceedings of the 2015 IFIP/IEEE International conference on Very Large Scale Integration (VLSI-SoC), Daejeon, South Korea, Oct. 2015, pp. 355–360.

[31] G. Hetherington, T. Fryars, N. Tamarapalli, Logic BIST for large industrial designs: Real issues and case studies, in: Proceedings of the IEEE International Test Conference, Atlantic City, N. J., USA, Sept. 1999, pp. 358–367.