Protection Method for AES IP Core from Scan-based Attack

YIFAN WU

Graduate School of Information,
Production and System, Waseda
University Kitakyushu, Japan 808-0135
e-mail : wuyifan@moegi.waseda.jp

SHINJI KIMURA

Graduate School of Information,
Production and System, Waseda
University Kitakyushu, Japan 808-0135
e-mail : shinji_kimura@waseda.jp

Abstract - **As Advanced Encryption Standard (AES) algorithm is widely adopted in VLSI as a hardware module called an IP (Intellectual Property) core to secure the data transfer among mobile products and internet. Scan path is indispensable for users to test such circuit. However, scan path has several problems on security.**

In the research, scan-based two bit difference attack method has been studied and complete the method by further analysis and additional tables and test patterns. Then a protection method for such scan-based attack is also proposed.

The proposed method cause less area overhead compared with the original AES IP core, higher security level and fault coverage compared to previous methods.

Keywords—Advanced Encryption Standard (AES), scan chain, secure scan design, bit difference attack, JTAG

I. INTRODUCTION

Advanced Encryption Standard (AES) is widely adopted to secure the data transfer among mobile products and internet [1]. A typical 128-AES encrypts 128-bit data blocks under the control of a 128-bit secret user key through ten rounds, in each rounds, there are SubBytes, ShiftRows, MixColumns and AddRoundKey. Except that an initial AddRoundKey is included in the first round and in the last round, there is no MixColumns.

Scan chain is a technique used in design for testing, which to make testing easier by providing a simple way to set and observe every flip-flop in an IC. Scan chain is indispensable for the users to test the IP core. At the same time, it causes several problems on security. The scan chains can be used as a side channel by attackers to retrieve the secret keys from the hardware implementations of AES algorithm.

Several attack methods through scan chains are proposed in the previous research. The main principles are based on the observation and analysis of differential scan-in vectors and scan outputs. A two-step attack method using the intermediate value between pre-round and first round is first proposed [2], totally 544 plaintexts on average are required to retrieve the secret key. Then a method of controlling round counter registers is proposed [3], which monitor the last round encryption data by controlling the round counter registers, the total vectors including those needed in distinguishing round counter registers are 146. Moreover, methods of signature attacks against advanced design for test (DFT) structures such as test response compaction, associated with X-masking structure, partial scan, etc. has been developed [4]. A low complexity bit difference attack then has been proposed, which retrieves the secrete user key without the knowledge of the order of registers in the scan chain. It also starts with scanning in one-bit differential input vectors, controlling the AES algorithm in specific round, however, unlike other methods, in this method, plaintexts are analyzed instead of the scan outputs. The algorithm is effective but missed few cases. In this research, the algorithm is improved by considering the dropped cases.

The protection of AES secret key becomes an important issue. Then secure scan techniques developed. There have been proposed secure scan techniques by modifying the scan chain structure to change intermediate value [3], which using specific structures to hide the intermediate value of scan flip-flop in scan chain. However this method is hard to apply IP core since all information should be opened to users to remain full controllability and observability. Other methods to protect AES secret key are to attach security circuit to the IP. To solve the problem of being attacked by the unauthorized testers, M numbers of different matching keys have been defined to protest AES user key in scan test mode which length is N [5]. However the engineer who knows the matching keys could activate the scan output to retrieve secret keys. In [2], a power-off operation has to be done when mode changed from insecure mode to secure mode. Mirror key registers (MKRs) are inserted to the AES core to avoid loading the user key during scan test mode and also to insert test keys instead. However, this method is not compatible with JTAG standards, scan flip-flops have to be implemented with reset mode, and it is hard to reuse the predesigned AES core since the MKR is added. Method of using fake key in scan mode and user key in normal mode has been proposed in [6]. Fake key method not only hide the user key information in test mode, but also provides high fault coverage. However, there are several security issues in this method.

The contributions of this research compared with other approaches are: this technique preserves compatibility with IEEE 1149.1 (JTAG) standards. No modification is needed to not only the AES core but also the TAP controller, thus the AES IP core and TAP controller IP core can be reused in designing an System on Chip (SoC), only external circuit is added. The technique can provide fully protection of key select signal and the scan output gating signal by triplicated

signals. The implementation requires less design penalties such as area overhead, and gets higher security level and fault coverage.

The thesis is organized as follows: chapter 1 briefly introduce AES algorithm and JTAG boundary scan test; chapter 2 summarizes the attack methods of AES secret key, which includes the discussion of the dropped case in bit difference attack; chapter 3 concludes secure scan structures and provide a new technique of protecting AES secret key; chapter 4 will show the comparison and analysis of proposed method with previous ones; chapter 5 will conclude the thesis.

II. Attack Methods of AES Secrete Key

A. Differential Attack of AES Secret Key

In the differential attack, when a pair of one-bit differential scan-in vector is applied, after one round encryption operation, some effect could be found in the intermediate value. The intermediate ciphertext is stored in the Register R. In an iterative AES architecture, the output of Register R is the input to the next round operation. Although R is included in the scan chain, the attacker does not know which bits in the bit stream come from R. By switching the AES circuit between normal mode and test mode, this correspondence can be established. First we apply plaintexts which are different in one byte, according to avalanche effect, 32 bit correspondence flip-flops in Register R are determined. Then pairs of one bit difference plaintexts are applied, unique hamming distance of scan-out determines the round key. Overall, 544 plaintexts on average are required to discover the user key.

B. Scan-based Attack on AES Through Round Registers

Due to the fact that in AES encryption process, when different inputs are applied, there are some special registers whose contents don't vary with the input in register R, one example of which is the round registers. This method makes the use of the round registers to control the AES encryption running in the first round, and then compare the intermediate value. The reason for doing so is that in the last round, only SubByte, ShiftRow, AddRoundKey is implemented, and the MixColumn operation is bypassed. Thus if one bit changes in the scan input, when the circuit under test (CUT) runs for one clock cycle with the same round key, it will lead at most 8 bit changes. However, if the one bit difference is injected in the first round, 32 bit change would happen because of the MixColumn operation. By using this method, the total vectors to discover the user key are 146.

C. Bit Differece Attack on AES with Scan Values and Output Ciphertext

In AES algorithm, it is easy to know the round key by executing only the last round with the scan-in values and monitoring the scan-out values. However, the registers in the scan chain are arranged randomly depending on layout, so it is important to identify the bit order of round registers in the scan chain for retrieving the secret key. To configure the bit positions of the registers, unique hamming distance are used to distinguish them. However the output ciphertexts are sequentially respected to the byte block and its bit order, thus, it could be utilized to distinguish bit order [7].

The steps of attack are shown as below:

Step 1: Identifying the round counter registers (RCRs)

The ideal vectors are the output ciphertexts, so the encrypt chip should runs in the last round, only by control the round counter registers (RCRs), encrypt chip could be set to the last round. The algorithm is the same as [3].

Step 2: Identifying the position for each bit of the round registers (RRs) in scan chain

In order to identify the bit position in round registers (RRs), in other words, scan-in vectors, which is input with, the sequentially predefined output ciphertext is used.

Before the introduction, several symbols are defined.

Scan-in vector V for round registers

$V = a_{00}a_{01}a_{02}a_{03}a_{10}a_{11}a_{12}a_{13}a_{20}a_{21}a_{22}a_{23}a_{30}a_{31}a_{32}a_{33}$

a_{ij} (0<=i, j<=3) is the byte with 8 bits.

Output ciphertext O (in order)

$O = d_{00}d_{01}d_{02}d_{03}d_{10}d_{11}d_{12}d_{13}d_{20}d_{21}d_{22}d_{23}d_{30}d_{31}d_{32}d_{33}$

d_{mn} (0<=m, n<=3) is the byte with 8 bits.

Step 2 could be concluded as: firstly, set all-0 scan-in vector as the initial input of the last round and run the last round in normal mode to get the initial output ciphertext. Then a new scan-in vector which has one bit difference is applied and a new output ciphertext if obtained, by comparing these two ciphertexts, according to Table 2, each unique changed bit positions in d_{mn} correspond to changed bit position in a_{ij}, then from Table 1 and 2, the bit position in round registers in scan chain is identified.

Table 1: Relationship between the input byte in round registers and output byte in output ciphertext

$a_{00} \to d_{00}$	$a_{01} \to d_{01}$	$a_{02} \to d_{02}$	$a_{03} \to d_{03}$
$a_{10} \to d_{11}$	$a_{11} \to d_{12}$	$a_{12} \to d_{13}$	$a_{13} \to d_{10}$
$a_{20} \to d_{22}$	$a_{21} \to d_{23}$	$a_{22} \to d_{20}$	$a_{23} \to d_{21}$
$a_{30} \to d_{33}$	$a_{31} \to d_{30}$	$a_{32} \to d_{31}$	$a_{33} \to d_{32}$

Table 2: Relationship between the bit position in a_{ij} and changed bit positions in d_{mn} of output ciphertext

Bit position in a_{ij}	Changed bit positions in d_{mn}
1	1,3,5,6,7
2	2,3,5,7
3	1,2,4,6
4	1,3,5,8
5	2,4,7,8
6	1,4,8
7	4,6
8	4,5,6,7,8

Step 3: Retrieving the secret key

Since the position for each bit of the round registers in scan chain is identified, the last round key could be derived from scan in and output ciphertext. Then the round key could be derived from the AES algorithm.

D. Two Bit Differece Attack on AES with Scan Values and Output Ciphertext

In one bit difference attack, only one bit of each scan-in vector changes compared with the initial one, which means each time, only one bit position of round registers in the scan chain is identified. 128 scan-in vectors are needed to identify all round registers. To reduce more scan-in vectors, two bit difference attack is introduced. Two situations are considered in two bit difference attack: in Situation I, the two changed bits are in different bytes. In Situation II, the two changed bits are in the same byte. If the changed bit positions appear in Table 2, the scan-in vector belongs to Situation I, otherwise, a new table should be introduced. Table 3 can be derived with the same method.

TABLE 3: RELATIONSHIP BETWEEN THE POSITIONS OF TWO BITS IN a_{ij} AND THE CHANGED BIT POSITIONS IN d_{mn} OF OUTPUT CHPHERTEXT

Positions of two bits in a_{ij}	Changed bit positions in d_{mn}	Positions of two bits in a_{ij}	Changed bit positions in d_{mn}
1,8	2,3,5,6,7,8	2,6	2,3,4,5
2,8	1,2,3	3,6	2,4,6,8
3,8	1,4,5,6,7	4,6	1,4,5,8
4,8	1,2,3,8	5,6	1,4,5,6,8
5,8	2,3,7	1,5	1,3,6,7,8
6,8	5	2,5	3,4,8
7,8	4,5	3,5	2,4,6,7,8
1,7	2,3,4	4,5	1,2,5,6,7
2,7	2,5,6,7,8	1,4	7,8
3,7	1,2,3,4	2,4	3,4
4,7	1,3,5,7	3,4	2,3,6,7,8
5,7	6	1,3	1,7,8
6,7	5,6	2,3	1,3,4,7,8
1,6	3,4,5,6	1,2	1,2,4,5,8

With the two bit difference attack, only 64 scan-in vectors are needed for round registers when all two changed bits are in different bytes, but if two changed bits are in the same bytes, other scan-in vectors are needed. So the attack is divided into two cases. The previous research ignored some cases which are mentioned in the following discussion.

The best case: All the scan-in vectors are included in Situation II. In this case, after 64 scan-in vector applied, by checking up Table 1 and Table 3, the positions and which byte it belongs of scan-in vectors will be uncovered. For example, the two bits belong to byte a_{00} and one of them is the 1st bit and the other is the 8th. However, which one is the 1st and which is the 8th still cannot be distinguished. Because one bit change could identify one bit position of the two-bit change scan-in vector, the other bit is then determined. For each byte, only four scan-in vector is enough to identify all the bit position, and also the 16 byte scan-in vector will not affect each other. Above all, 64+4=68 scan-in vectors are needed in the best case.

The worst case: All the scan-in vectors are included in Situation I. In this case, after 64 scan-in vector applied, by checking up Table 1 and Table 2, the bit position and which two bytes it belongs of scan-in vectors will be uncovered. For example, the two bits belong to byte a_{00} and a_{01} , one of them is the 1st bit and the other is the 8th. The possible combination is the 1st bit in a_{00} & the 8th bit in a_{00} or the 1st bit in a_{01} & the 8th bit in a_{01} . To distinguish them, additional scan-in vectors are needed. The changed bit should come from different byte, in order to avoid two or more bit changing in one byte (which is not included in Table 2 and Table 3), so at most 8 bits changing is allowed. Scan-in vector V shown above represents one situation. If each vector applied has 8 bits differential, 64/8=8 additional scan-in vectors are needed to identify the position. Totally 64+8=72 scan-in vectors are needed in the worst case.

The proposed bit difference attacks with scan values and output ciphertexts method is compared with the attack in [3] as shown in Table 4. For all the attacks, 7 plaintexts are used to identify the round counter registers in scan chain in Step 1 and one scan-in vector is used to retrieve the secret key in Step 3.two separate operations in previous attack can be combined to one step (step 20) of the proposed bit difference attacks In this paper and attack complexity can be reduced. In Step 2, by using one bit difference attack, 8 scan-in vectors can be reduced. By using two bit difference attack, 136 scan-in vectors can be reduced to 68 scan-in vectors in the best case while can be reduced to 72 scan-in vectors in the worst case.

TABLE 4: COMPARISON RESULTS WITH PREVIOUS ATTACK

Step number	**Contents in each step**	**Previous attack [3]**	**One bit difference attack**	**Two bit difference attack**	
Step 1	Identify RCRs	7 plaintexts	7 plaintexts	7 plaintexts	
Step 2	Identify each byte block in RRs	128 scan-in vectors	128 scan-in vectors	Best case: 68 scan-in vectors	Worst case: 72 scan-in vectors
	Identify the bit order in each block	8 scan-in vectors			
Step 3	Retrieve the secret key	1 scan-in vector	1 scan-in vector	1 scan-in vector	

III. MOTHODS OF PROTECTING AES KEY

From the mechanism of attacking, the protection method could be conclude into two types, the first one is modifying scan chain itself, the other one is adding external circuits at outside of the AES IP core.

A. Previous Methods of Protecting AES Key

In scan chain modification methods, MKR (Mirror Key Register) based architecture is proposed. On the secure-scan DFT architecture, two new modes of operation: insecure and secure modes are defined in [2]. When a crypto chip is in the insecure mode, it can be switched between the test mode and the normal mode similar to the general scan-based DFT. On the other hand, when a crypto chip is in the secure mode, it can only stay in the normal mode, and cannot change to the test mode. A crypto chip can be switched from insecure mode to secure mode at any time, but switching back from secure mode to insecure mode can only be done through a power-OFF reset.

In order to ensure the testability to every user, at the same time to protect the AES key information, fake key in test mode method is proposed [6], which is similar with [2] in mechanism. In this method, the data are encrypted with the fake key instead of the user key by executing new instruction in scan test mode.

B. Proposed Secure Scan Architecture

A secure scan design technique is proposed in this subsection to protect the secret key of an AES core embedded in an SoC. The features preserve the compatibility with JTAG standard for using to reuse predesigned AES IP core and TAP controller for IP cores. Compared with the original fake key method, the proposed method provides completely protection of the secret key is with less design penalties such as area overhead and testability.

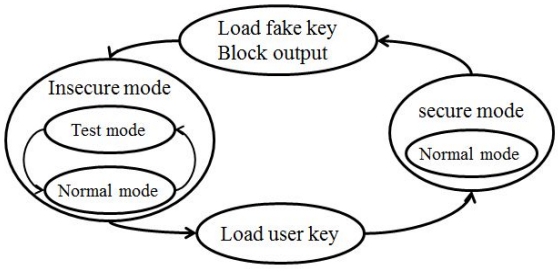


Figure 1: State diagram of proposed method

The state transition diagram of the controller is shown in Figure1.The main transition is briefly summarized as follows.

1) In normal mode, secret user key is loaded instead of a fake key with deactivating scan output port; the circuit runs in normal function.

2) When TMS (Test Mode Select) receives mode change signal, TAP controller will also change outputs to conduct scan test. At the same time, TAP controller will activate combinational circuit which generate Load Key signal to load the fake key.

3) While scanning in a test vector, internal values of IP cores will also be shifted out, but the counter is activated to gating the scan output port and cannot be monitored.

4) After scan in, the gating signal is deactivated and the scan output could be observed directly, the remaining work will using Fake key for encryption and can be monitored via scan chain.

The control block generating the Load Key and Scan Output Gating signals is designed using the TAP controller output. The architecture is shown in Figure 2.

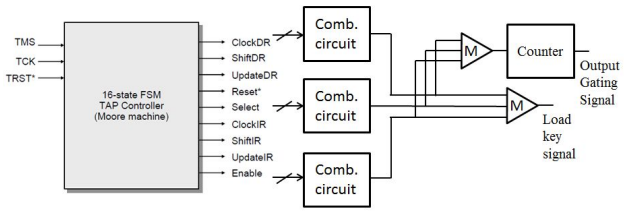


Figure 2: Modified Load Key signal generating circuit

Figure 3 shows our design of the combinational circuit. In the circuit, Shift DR, Enable and Select are used. By the simple combination of these 3 signals, Load Key signal is generated with this manner. Note that, the majority gate is also integrated in the multiplexer of Key Input to prevent manipulation of the Load Key signal by attacker.

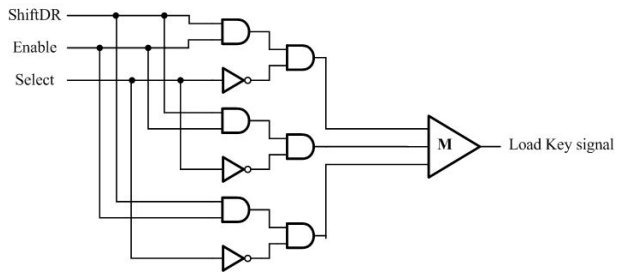


Figure 3: Circuit to generate Load Key signal

The circuit increases the security of scan architecture against attack, since it decreases the potential risk of changing control signals by an unauthorized person. The Load Key signal is generated by output of TAP controller, which activate Load Key signal only during scan test mode. Since the Load Key signal is triplicated, if one of three signals is controlled by an attacker, the output of majority gate will not change. On the other hand, since the attacker is not aware of the internal structure, the possibility of controlling two or more signals simultaneously is very low.

The security of the proposed secure-scan architecture depends in the integrity of the control signals. If the Output Gating signal is not activated after scan mode changes from normal mode to test mode, or the Enable signal of the TAP controller is activated during normal mode, then the secret key can be scanned out or retrieved by analyzing the temporal results via the scan chains. For all crypto devices, deliberate faults injected by the attackers to break the key are a potential threat to these critical control signals [8]. To enhance the security, the control signals should be concurrently checked.

(Load Key signal = 0, ShiftDR = 0) refers to the value combination of normal mode. (Load Key signal = 1, Output Gating signal = 0, ShiftDR = 1) refers to another value combination in test mode after changing from scan mode. After the key related intermediate value is flushed out, the combination becomes (Load Key signal = 1, Output Gating signal = 1, ShiftDR = 1). The Output Gating signal's correctness depends on the counter. When the circuit is flushing out the key related intermediate value, the Output Gating signal is activated (0), after that, the value can be 1.

The dependency of counter makes it difficult to identify the integrity of Output Gating signal at different time. At the same time, the counter is vulnerable against glitch attack, so the fault checking circuit is necessary in the implementation. Figure 4 shows the structure of concurrent fault checking circuit, one more counter which is activated by another signal is inserted. One of the counters is activated by Load Key signal, the other counter is activated by ShiftDR. Since glitch attack has some randomness on changing the intermediate value, and the activate signal is different, the possibility of inserting the same fault value to the two counters simultaneously is extremely low, in the most cases, once the fault is inserted into the counter, the values in each counter are different from each outer, so it will be detected by bitwise exclusive OR.

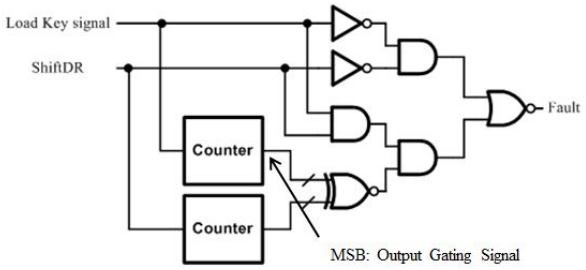


Figure 4: Concurrent Fault Checking circuit

This fault signal is connected to the reset signal of AES IP core and TAP controller to generate the reset signal of the whole chip. With this manner, the fault insertion attack which could decrease the security of the chip can be detected and protected by resetting the core.

IV. Implementation and Experimental Result Analysis

A. Implementation

The proposed method makes a few changes to the original structure of AES IP core with JTAG boundary scan chain, the load key signal is triplicated and majority detection monitors are adopted, then a concurrent fault checking circuit is inserted to enhance the security. The structure is shown in Figure 5. Compared with the previous method in [2], there is no modification on AES scan chain, which means the design can reduce the application cost and increase the reusability and compatibility. The power-off procedure is also not needed when mode changes from insecure mode to secure mode, thus there is no need to add reset pin to the scan flip-flops, which also reduce the area overhead. Compared with method [6], the proposed design makes no modification to the TAP controller, and no new JTAG states need to be added, which means less compromise to the reusability and area overhead.

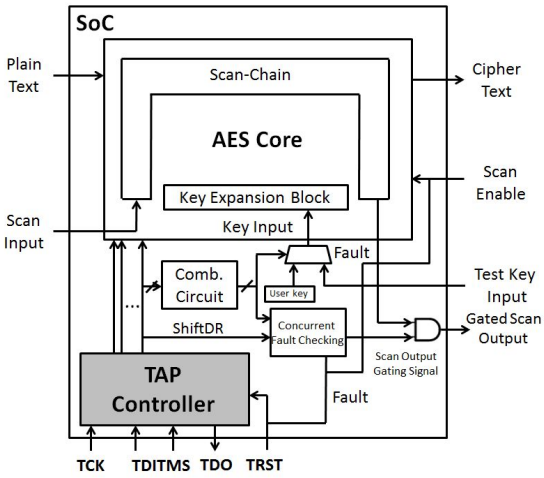
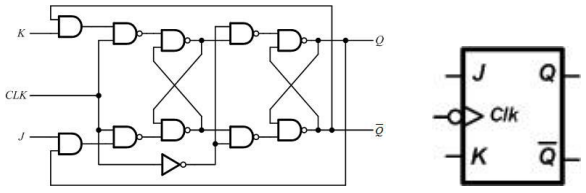


Figure 5: The structure of proposed method

The main part of area overhead lies in the control signal of Load Key signal and Output Gating signal. The Load Key signal is also connected with majority gate. In the implementation, the counter is constructed by master-slave JK flip-flops as shown in Figure 6. As can be seen in (a), the master-slave flip-flop is constructed by 11gates. The logic symbol is shown in (b). By serially connecting N previous output Q and the J pins in the next stage, we can get synchronous N bit binary counter (Figure 7). All the K pins are connected to constant 1, and the first J pins are set as reset. Initially the counter is all 0, then after 2^{n-1} clock cycles, the most significant bit becomes 1, which is used as Output Gating signal disables. The counter counts up when Load Key signal=1, Before the next term scan test, a reset is needed to make sure a valid gating signal during next term scan test. When the Load Key signal is disabled, the counter behaves as a shift register. After N clock cycles (N refers to the number of JK F.F.'s), the counter becomes all 0, which equals to be reset.



(a) (b)

Figure 6: structure of a master-slave JK flip-flop (a) Gate level structure (b) logic symbol of JK flip-flop

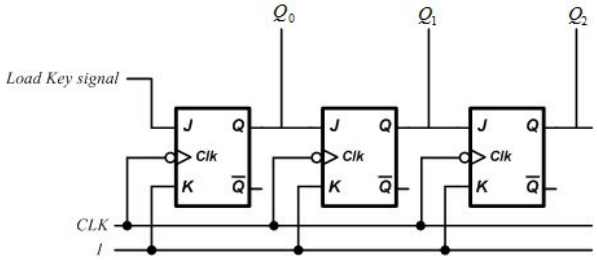


Figure 7: Counter structure

B. Overhead Analysis

The number of JK flip-flops is depending on the length of AES scan chain, which is vary due to the DFT design of AES IP core. We assume the total length of AES scan chain is N, then the number of JK flip-flop is $\log_2 N + 1$, which is correspond to $11 \log_2 N + 11$ gates. In the Load Key signal generation part, 18 gates overhead should be considered (including the key select multiplexer). When implement the concurrent fault checking circuits, result of bit-wise exclusive OR transmission will cost 19 2-input gates (when 10 JK flip-flops), the counter costs $22 \log_2 N + 22$ gates, so the total area overhead of concurrent fault checking circuit is $72+ \log_2 N + 22$ gates. The area overhead comparison of proposed method and previous methods is shown as Table 5. We use an AES core in [1]. The AES implementation is the iterative with key scheduling without pipeline. In the Fake key in test mode method, no discussion about the instruction decoder is introduced, so the structure is unavailable.

TABLE 5: COMPARISON OF AREA OVERHEAD

Architectures	Area (gates)	Area overhead	
		Area (gates)	Percentage (%)
Mirror key method [2]	31,234	412	1.32
Proposed method	31,132	$90+22 \log_2 N$	≤ 1.00

In the table, we set the length of scan chain 'N' to 1000, which is much larger than the number of Round registers 128. From the table, we can conclude that, since the Mirror key method adds multiplexer to every round registers flip-flops, the area overhead is much larger than the proposed one, even two counters in the proposed method are added.

C. Security and Fault Coverage Analysis

The concurrent fault detection circuit is added to the proposed secure design which is similar with [2] can greatly improving the security level. As the counter issue, in previous method [6], it is easy to conduct glitch attack on counter, and Output Gating signal is invalid before the intermediate value is completely shifted out. In the proposed method, it has low possibility to conduct both counters inserted with fault values, at that time, abnormal could be detected. The Fault signal is directly connected to the reset pin of TAP controller and AES IP reset pin, which will suspend scan test. If faults are inserted into two counters, however, since the randomness of glitch attack, it is hard to set two counters into the same value. The differential will also detect the fault integrally.

The fault coverage is depending on the structure of scan chain structure itself and the choosing of scan-in vector. It is claimed in [6], in order to obtain high fault coverage, the test key is chosen with meticulousness. 100 different Fake key values has been tested by Synopsys TetraMax automatic test pattern generation tool, finally the result shows all 0s provide the highest fault coverage. In our design, in order to obtain higher fault coverage, test key is set as a parallel input, thus different test key could be applied to the AES.

D. Overall Analysis

Table 6 compares the general technical aspects with previous approaches [2] and [6]. From the table, though the mirror key method offers complete protection on AES user key, the reusability of hard AES IP, TAP controller and compatibility with JTAG is not available; in fake key method, the vulnerable structure of load key signal and gating signal makes it weak under attack. However, the proposed method provides highest security level among the previous methods.

TABLE 6: COMPARISON OF GENERAL ASPECTS

	Mirror key method [2]	Fake key in test mode method [6]	Proposed method
Reusability of hard AES IP	N	Y	Y
Compatibility with JTAG	N	Y	Y
Complete protection of AES user key	Y	N	Y
Reusability of TAP controller	N	N	Y

V. CONCLUSION

This thesis focus on side channel attack methods on AES and protection method for such side channel attacks. A two bit difference attack is studied by discussing the dropped cases, which shows the applicable of this method. Then a new secure scan architecture has been proposed, which obtain higher security level with less area overhead.

References

[1] Mangard, M. Aigner and S. Dominikus, "A Highly Regular and Scalable AES Hardware Architecture", IEEE Transactions on Computer, vol. 52, no. 1, pp. 483-491, April, 2004.

[2] Yang, K. Wu and R. Karri, "Secure Scan : A Design-for-Test Architecture for Crypto Chips", IEEE Transaction Computer-Aided Design of Integrated Circuits and systems, Vol. 25, No. 10, pp. 2287-2293, Oct. 2006.

[3] Y. Shi, N. TOGAWA, M. YANAGISAWA, "Scan-Based Attack on AES through Round Registers and Its Countermeasure," IEICE TRASACTIONS on Fundamentals of Electronics, Communications and Computer Sciences Vol. E95-A No. 12, DEC. 2012, pp. 2338-2346

[4] Da Rolt, J.; Di Natale, G.; Flottes, M.-L.; Rouzeyre, B.; , "Are advanced DfT structures sufficient for preventing scan-attacks?," VLSI Test Symposium (VTS), 2011 IEEE, pp.246-251, June 2012.

[5] S. Paul, R. S. Chakraborty and S. Bhunia, "VIm- Scan : A Low Overhead Scan Design Approach for Protection of Secret Key in Scan-Based Secure Chips", VLSI Test Symposium, pp. 455-460, Berkeley, CA, USA, May 6-10, 2007.

[6] Jaehoon Song, Taejin Jung, Jihun Jung, Sungju Park, "An Efficient Technique to Protect AES Secret Key from Scan Test Channel Attacks," Journal of semiconductor technology and science, vol.12, NO.3, pp. 286-292, September, 2012

[7] Pei GAO, Shinji KIMURA, Graduate School of Information, Production and System, Waseda University Kitakyushu, Japan

[8] E. Biham and A. Shamir, "Differential fault analysis of secret key cryptosystems," in Proc. CRYPTO, Santa Barbara, CA, 1991, pp. 156–171.