AES Design Space Exploration New Line for Scan Attack Resiliency

Sk Subidh Ali, Ozgur Sinanoglu
New York University Abu Dhabi (NYUAD)

Ramesh Karri
New York University Polytechnic School of Engineering

***Abstract*—Crypto-chips are vulnerable to side-channel attacks. Scan attack is one such side-channel attack which uses the scan-based DFT test infrastructure to leak the secret information of the crypto-chip. In the presence of scan, an attacker can run the chip in normal mode, and then by switching to the test mode, retrieve the intermediate results of the crypto-chip. Using only a few input-output pairs one can retrieve the entire secret key. Almost all the scan attacks on AES crypto-chip use the same iterative 128-bit AES design where the round register is placed exactly after the round operation. However, the attack potency may vary depending on the design of AES. In this work, we consider various designs of AES. We shed light on the impact of design style on the scan attack. We also consider response compaction in our analysis. We show that certain design decisions deliver inherent resistance to scan attack.**

***Index Terms*—AES, Scan Chain, Scan Attack, Scan-based DFT, Testability, Security.**

I. Introduction

While electronic devices improve the communication power, provide computing power to common man, and enhance business in every sectors, they have also increased the threat to information security by many folds. Now the attacker can have direct access to security sensitive devices such as smart cards, RFID tags, PDAs and other hand-held devices, and launch various physical attacks. One such physical attack is the scan attack, where the attacker exploits the scan-based DFT test infrastructure to break a crypto-system.

Scan-based DFT is a widely used test methodology where the internal registers are connected into chains. In test mode a tester uses these chains to stimulate the internal nodes of a chip and to observe the response. While scan improves access and thus testability of manufacturing defects, it can also be used as a back door to steal the secret information from a chip. The attack which uses this implementation-based weakness is called *scan-based side-channel attack*. The first such attack was reported in [1]. The attack used scan architecture of Data Encryption Standard (DES) chip. It was shown that the intermediate results of DES cipher can be retrieved by running the cipher in normal mode for a few cycles and then by switching the device to test mode, where the attacker can shift out the content of internal registers. Then a simple analysis on the input-output pairs can retrieve the secret key.

Subsequently, the attack is proposed on many ciphers such as Advanced Encryption Standard (AES) [2], RSA [3], ECC [4] etc. The attack potency is also evaluated in the presence of different DFT structures such as partial scan [5], X-masking and X-tolerant architecture [6], [7]. These results show that the potency of the scan attacks varies with the underlying scan architecture being used.

In this work, we consider the AES design, as it is the most widely used cipher. AES designs vary in implementation in terms of:

1) Iterative vs pipelined,
2) 128-bit vs 32-bit vs 8-bit data width,
3) Location of the round register, and
4) Number of rounds.

Although all scan attacks have assumed the 128-bit iterative AES implementation, we show here that the implementation decisions have different implications on scan attack resiliency.

978-1-4799-6016-3/14/$31.00 © 2014 IEEE

II. Preliminaries

A. AES

AES is a 128-bit symmetric key block cipher [8], provides three different security levels with key lengths 128, 192 and 256 bits, respectively. The AES algorithm consists of identical round operations. The number of rounds in the three different versions of AES are 10 (128-bit key), 12 (192-bit key) and 14 (256-bit key) respectively. Each round comprises the following four basic transformations:

- `SubBytes` is a non-linear substitution operation.
- `ShiftRows` is the byte-wise permutation.
- `MixColumns` is the four-byte mixing operation
- `AddRoundKeys` is the X-ORing the state with the round key.

We will refer to these operations as *S*B, *S*R, *M*C and *A*RK, respectively. The basic round operation of AES is shown in Figure 1. The first round of AES contains an extra key X-ORing at the beginning, which is referred to as key whitening.

B. Design Decisions for AES

The AES algorithm is designed in such a way that it can be implemented as per the user requirements. If the objective of the user is to get high throughput, AES can be implemented in a pipelined fashion with 128-bit data width (Figure 2(b)). In a pipelined design, multiple 128-bit registers (R_0 to R_{10} in Figure 2(b)) are used to store the intermediate results. In this

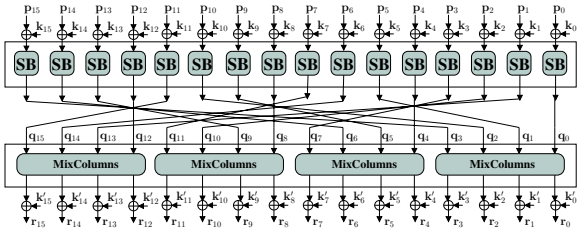
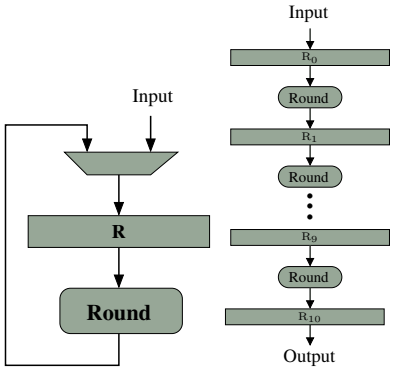


Fig. 1. First round of AES: p_i is the plaintext byte, k_i is the initial key byte, q_i is the SR output byte, k'_i is the round key byte, and r_i is the round output byte

case a ciphertext is generated in each clock cycle. Therefore, the design provides the highest throughput. On the other hand, if the user objective is low area cost, he/she should choose the iterative implementation (Figure 2(a)), where one round operation is repeated for ten times (AES with 128-bit key) and the intermediate round output is stored in the round register R. Iterative implementation consumes roughly one tenth of the area of the pipelined implementation, albeit with low throughput (an encryption takes ten clock cycles).



(a) Iterative (b) Pipelined

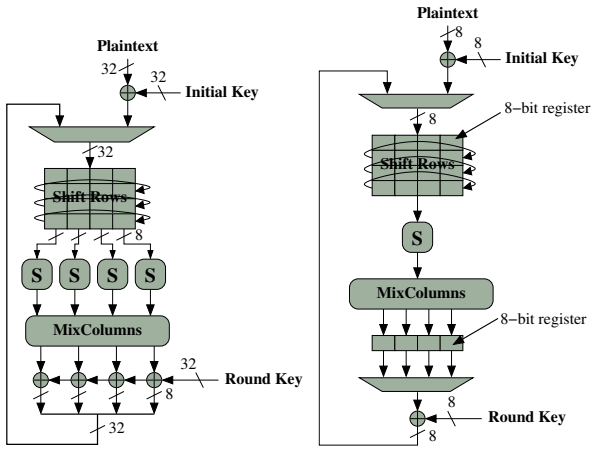
Fig. 2. Iterative and pipelined implementation of a 10-round cipher

The area overhead can be further reduced by considering the 32-bit design, where the number of *S*B and *M*C blocks reduces four times (Figure 3(a)). One can also use an 8-bit design (Figure 3(b)), where only one *S*B and *M*C block is used in a round operation. The 8-bit design of AES consumes the least area, which is 70% less than that of the 128-bit design of AES.

Another design parameter is the location of the round register. Generally, the round register is placed either in between the key whitening operation and the round operation, or at the end of the round operation. In the third case, the round register is placed exactly after the S-Box operation to share the S-Box between encryption and decryption blocks.

III. Basic Scan Attack on AES

Regardless of the implementation, the intermediate round outputs are stored in the round registers (R in Figure 2). In the presence of scan infrastructure, the round register of the implemented cipher is connected into scan chains. An attacker



(a) 32-bit (b) 8-bit

Fig. 3. AES implementation with 32-bit and 8-bit data width

can run the cipher in normal mode with the desired plaintext, and then by switching to the test mode, retrieve the contents of these round registers by unloading the scan chains. For retrieving the key, the attacker uses the differential property of the cipher. Generally, the attack is performed on the first round of the cipher.

The attack targets the whitening key of the first round. As per the differential properties of AES, if we apply all possible one-bit input differences to the least significant bit of a byte, only a few hamming distances (9, 12 23, 24) at the round output difference will correspond to unique input pairs (Figure 4). Therefore, by observing the unique hamming

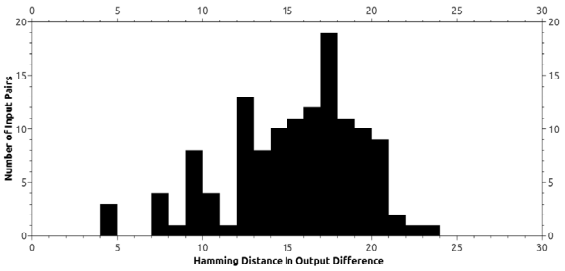


Fig. 4. Distribution of hamming distances for all possible 128 input pairs

distance at the output response pairs, one can determine the S-box input pair, which in turn reveals the secret key byte. This technique is applied across all the bytes to retrieve the entire key. Attack on AES with the basic scan architecture requires around 544 plaintexts and a brute-force search of 16 bits to retrieve the secret key [2]. Most of the scan attacks on AES have targeted the basic 128-bit iterative designs of the cipher [2], [5]–[7]. However, the potency of the attack may vary with the design decisions, which are analyzed next.

IV. Scan Attack on 128-bit iterative designs

In this work we do a detailed scan attack analysis by considering variation in terms of the location of the round

register in 128-bit iterative designs. In Section VI, we briefly discuss scan attack implications of other design aspects as well.

A. Round Register Placed After the Round operation

Almost all the existing scan attacks have targeted 128-bit iterative designs, where the round register, which is the only register in the encryption module, is placed after the round operation. In this case, the round register will hold the round output after each iteration. Therefore, after the first encryption-cycle the attacker can switch to the test mode and shift out the round output. The attack has to include the round operation, identification of the input to scan cell mapping, and a differential (hamming distance) analysis on the cipher. The detailed attack procedure can be found in [2]. The attack reduces the search space of the key to 16 bits, which can be easily brute-forced to get the actual key.

B. Round Register Placed Before the Round Operation

In the majority of AES implementations [9]–[11], the round register comes in between the pre-round and the round operations (Figure 2(a)). In this a case more effective form of scan attack can be developed.

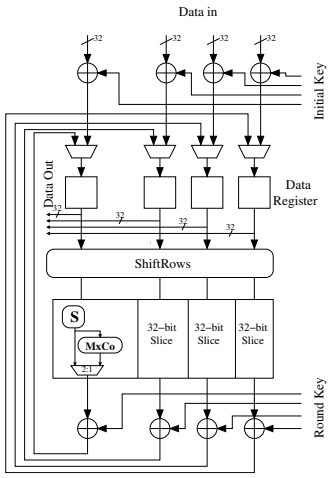


Fig. 5. General 128-bit architecture of AES encryption [12]

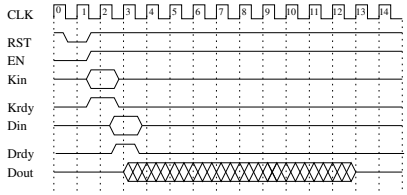


Fig. 6. Timing diagram for encryption module

In order to evaluate our analysis, we consider the open source AES verilog code provided in [11]. The timing diagram corresponding to this design is shown in Figure 6. As per the timing diagram, after the reset and enable signals (RST and EN) are asserted, the initial key is applied and stored in the key register (which is not shown in the figure). Krdy signal indicates that the key is ready to be used. Subsequently, the plaintext is loaded in the input register (not shown in the figure) and the Drdy signal goes high. On the third clock cycle the X-OR of plaintext and the key (pre-round operation) is loaded in the *Data Register* as shown in Figure 5.

At the end of the third clock cycle the round register (Data Register in Figure 5) holds the result of the pre-round operation. This type of AES design, which is widely used in the security industry[1], is more vulnerable to the scan-based attack as the attack does not need to involve the round operation and the challenges associated (differential analysis). The attacker can run the device for only three consecutive clock cycles and then switch to the test mode. In the test mode, he/she can shift out the content of the round register. In this case also, the attacker does not know the scan cells corresponding to the round register similar to the attack in the previous subsection. On the other hand, unlike the existing attacks, the attacker is not using the round output. Therefore, the attacker has to precisely know the exact mapping between the scan cells and the round register. However, we show next that this mapping can be easily identified.

We assume that the attacker has the access to the 128-bit input lines and can apply any desired plaintext to the input lines. We apply two plaintexts p and p_i with one-bit difference in the i-th bit. We can consider p to have all zero bits and p_i to have all zero bits except for the i-th bit. In each case we run the cipher for three consecutive clock cycles and then switch to the test mode. In the test mode we shift out the scan chain content. Say the one round response of p is $R(p)$. The output difference $D_i = R(p) \oplus R(p_i) = p \oplus k \oplus p_i \oplus k \Rightarrow D_i = p \oplus p_i$.

Therefore, the difference D_i will have one-hot data and the position of 1 will be i, revealing the scan cell corresponding to the i-th bit of the round register. By varying all possible 128 positions of i, we can determine the exact mapping between the scan cells and the round register. Once we have the mapping, we consider the response $R(p) = p \oplus k = 0 \oplus k = k$ (where k is the 128-bit round key). We can now recover the value of k from the scan outputs.

The complete attack on the pre-round operation requires only $128 + 1 = 129$ plaintexts. The attack is so simple that it does not require any off-line calculation or brute-force operation.

C. The Round Register Placed After the S-Box Operation

In this section we are going to assess the difficulty of scan attack on AES designs, where the round register is placed just after the S-Box operation, which is a resource constraint driven decision [10, §10.7.1] to share the S-Box between encryption and decryption modules (Figure 7).

It may be noted that in the first clock cycle, the round register will hold only the S-Box output. Therefore, after the first clock cycle, one can shift out the S-Box output by switching to the test mode. Figure 8 shows the distributions

[1]Side-channel attack standard evaluation board is one such example.

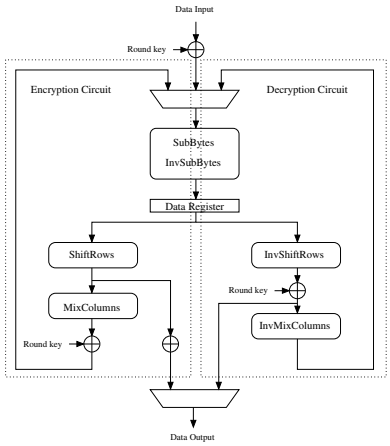
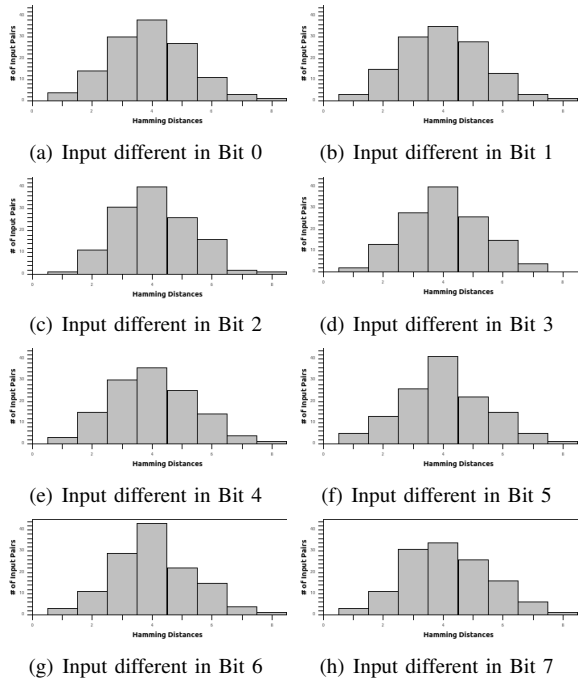


Fig. 7. 128-bit architecture with S-box shared between encryption and decryption modules



(a) Input different in Bit 0 (b) Input different in Bit 1

(c) Input different in Bit 2 (d) Input different in Bit 3

(e) Input different in Bit 4 (f) Input different in Bit 5

(g) Input different in Bit 6 (h) Input different in Bit 7

Fig. 8. Distribution of Hamming distances when the one-bit input difference is applied in different bit positions (0 to 7-th bit)

of hamming distances when the one-bit input differences are applied on bit 0 to bit 7. It shows that in seven out of the eight cases we get only one unique hamming distance of eight, which corresponds to the unique input pair. Therefore, in this case we can apply all possible 128 one-bit differences to any one of the input bytes. The one-bit difference should be applied to any bit other than bit 3. We calculate the output hamming distance corresponding to each input pair. When we encounter a hamming distance of eight, we directly get the corresponding unique S-Box input pair. In order to get the corresponding whitening key byte, we X-OR the input plaintext byte with the calculated S-Box input byte. We get two values corresponding to each key byte. We apply this technique across all the bytes and determine the key byte values. The final key space will reduce to 2^{16}, which can be brute-forced easily. In the worst case we need to apply $128 \cdot 16 = 2048$ plaintexts to determine the key. The attack complexity is the same as in the attack on 128-bit iterative design of AES, where the round register is placed at the end of the round operation [2]. However, the expected number of required plaintexts is twice the one in [2]. This is because in [2], the attacker gets four unique hamming distances, whereas in this design style we can get two such hamming distances.

V. Scan Attack on 128-bit iterative designs of AES in the Presence of Response compactor

Recently, it was claimed in [13] that on-chip compression/compaction can provide inherent resistance to scan attack. However, an attack was proposed on AES with X-OR response compactor [7], which can reveal the secret key using twice the number of plaintexts used in [2]. In this section we will present our attack analysis on AES implementation variations in the presence of an on-chip response compactor.

A. Round Register Placed After the Round operation

The attack on this type of design in the presence of a response compactor is shown in [7], which considered that the 32 key cells of an AES word are distributed from 1 to 32 scan slices, where each slice is compacted by an X-OR tree compactor. In the best case, each slice holds only one key cell and the attack is identical to the one in [2]. In the worst case, all the 32 key cells reside in the same slice and are compacted to a single bit by the X-OR compactor. In the existing attacks on AES, there are four unique hamming distances (9, 12, 23, 24). In the presence of a response compactor, it is assumed that the actual hamming distance is reduced by the response compactor. Therefore, whenever the attacker observes any none-unique hamming distance (say 8, even parity), he/she chooses the closest unique hamming distance (12) with the same parity as the corresponding actual hamming distance.

As there are four such unique hamming distances $(9, 12, 23, 24)$, in the final hypotheses, the actual key byte will have the highest frequency of occurrence (at least 4). The values with the highest frequency of occurrence is retrieved. There could be more than one such value. In order to uniquely determine the key byte value, the attacker applies input pairs with one-bit difference in the next bit of the byte and gets another set of possible key byte values. The intersection of the two sets should uniquely identify the key byte. The attack reveals the secret key using $256 \cdot 16 = 4096$ plaintexts and a brute-force search of 16 bits. The attack complexity is the same as in [2], while the required number of plaintexts increased twice.

B. Round Register Placed Before the Round Operation

The advantage of attacking the pre-round operation is that the attacker can easily identify the scan architecture details. Suppose two scan cells SC_i and SC_j are compacted, producing a single bit response corresponding to these two scan cells.

We can figure out if there is any such compaction by applying two plaintexts p_i with a single one in the i-th bit, and p_j with a single one in the j-th bit. If the i-th and the j-th scan cells are compacted, the output difference will be a one-hot pattern. Otherwise, we will get both the i-th and j-th bits to be one in the output difference. By varying i and j we can figure out the group of bits that are compacted together.

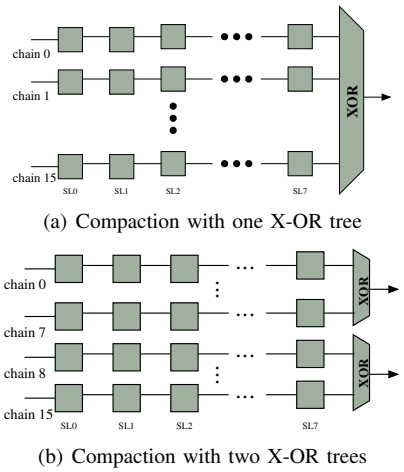
The presence of round register before the round operation helps to retrieve information one step ahead of the execution of the round operation. However, the amount of information leakage in the pre-round operation depends on the distribution of the round register flip-flops in the scan chains. We consider the general distribution of scan cells as proposed in [14]. We extend the distribution for up to 32 key cells in 24 active slices[2], as the effect of one byte input difference in an AES round will always confine within 32 bits (four bytes) of the output. Here, the key cells are the scan cells corresponding to the round register and are affected by the input difference. The number of scan chains will be 9 or more.

TABLE I
DISTRIBUTION OF 32 KEY CELLS IN 24 ACTIVE SLICES/SCAN CHAINS

#	Distribution number	#	Distribution number
1	$\{2, 2, 2, 2, 2, 2, 2, 2, 1, ..., 1\}$	12	$\{5, 3, 2, 2, 1, ..., 1\}$
2	$\{3, 2, 2, 2, 2, 2, 2, 1, ..., 1\}$	13	$\{5, 3, 3, 1, ..., 1\}$
3	$\{3, 3, 2, 2, 2, 2, 1, ..., 1\}$	14	$\{5, 4, 2, 1, ..., 1\}$
4	$\{3, 3, 3, 2, 2, 1, ..., 1\}$	15	$\{5, 5, 1, ..., 1\}$
5	$\{3, 3, 3, 3, 1, ..., 1\}$	16	$\{6, 2, 2, 2, 1, ..., 1\}$
6	$\{4, 2, 2, 2, 2, 2, 1, ..., 1\}$	17	$\{6, 3, 2, 1, ..., 1\}$
7	$\{4, 3, 2, 2, 2, 1, ..., 1\}$	18	$\{6, 4, 1, ..., 1\}$
8	$\{4, 3, 3, 2, 1, ..., 1\}$	19	$\{7, 2, 2, 1, ..., 1\}$
9	$\{4, 4, 2, 2, 1, ..., 1\}$	20	$\{7, 3, 1, ..., 1\}$
10	$\{4, 4, 3, 1, ..., 1\}$	21	$\{8, 2, 1, ..., 1\}$
11	$\{5, 2, 2, 2, 2, 1, ..., 1\}$	22	$\{9, 1, ..., 1\}$

The most likely distribution of key cells are shown in Table I. In the first distribution, 8 slices will have 2 key cells each and the rest of the 16 slices will have one key cell each. Similarly, in the 22-nd distribution, one slice will have 9 key cells and the remaining slices will hold one key cell each. The information leakage from the pre-round operation is at least one bit per slice, as in the worst case, the entire slice is compacted to one bit. So, each distribution will leak 24 bits of information for the 32-bit key word. By using the pre-round operation, we can retrieve 96 bits out of the 128-bit key. The exact information leakage depends on the number of slices. In order to get the entire key, we need to brute-force only 32 bits of the key, which can be done quickly.

We also experimented with our technique on a sample scan architecture of 16 scan chains and 8 slices, with one and two X-OR trees forming the compactor, respectively (Figure 9). In this case, we consider that the scan cells consist of only the round register flip-flops. We randomly distribute the 128 key cells of the round register to the 16 scan chains and 8 slices. With each random distribution, we calculate the information leakage. We repeat the experiment on $16777216 = 2^{24}$ random distributions of the key cells. The simulation results show that on average the two scan architectures (Figure 9(a) and Figure 9(b)) reveal 85 bits and 100 bits respectively of the 128-bit AES key. Therefore, a brute-force search on 43 bits, and 28 bits is sufficient to reveal the key, respectively. This takes six hours and ten seconds in our experiments, respectively.



(a) Compaction with one X-OR tree

(b) Compaction with two X-OR trees

Fig. 9. Linear response compaction with 16 chains and 8 slices

The results show, even in the presence of response compactor, that we can reveal the entire AES key without even running the AES round operation.

C. Round Register Placed After the S-Box Operation

When the round register is placed after the S-Box operation, we observe that there are only two unique hamming distances (1 and 8). In the presence of a response compactor there is a chance that the actual hamming distances are reduced by the response compactor. In this case the existing attack procedures [7] and [6] will fail. In Section IV-C, we use only the unique hamming distance (8) to figure out the key byte. If compaction reduces the hamming distance 8 to 7, we can not differentiate this case from the actual hamming distance of 8.

Further, in the existing attack on response compactor [7], there are four unique hamming distances. However, in our case, we get up to two such hamming distances, when the one-bit difference is applied on bit 2 (Figure 8). Therefore, we choose bit 2, apply all possible 128 input pairs with one-bit difference, and get the corresponding output hamming distances. If we encounter a hamming distance of 1, we use the corresponding precomputed S-Box input, X-OR it with the input plaintext byte, and get the key byte value. On the other hand if we encounter a hamming distance greater than 1, we consider it as the compacted form of hamming distance 8 and get the corresponding key byte value. For one key byte we will get a set of values, and some values will occur twice. We will

[2]Active slice or scan chain refers to those slices or scan chains that hold at least one key cell. A slice refers to a group of scan cells that are compacted in the same shift cycle; usually, they can be conceived to be aligned vertically.

choose only those values as the possible key byte candidates. Unlike the existing attack [7] no multiple list is there to take the intersection and to get the unique value. Therefore, we will get a list of possible values corresponding to each key byte.

We perform our simulation on the two scan architectures in Figure 9(a) and Figure 9(b). In each trial we consider a random distribution of the 128 key cells in the scan chains. The results are shown in Table II. The table shows the maximum, minimum and average number of key byte values among 16777216 trials. We can see that the result is the same for both architectures. This is because the unique hamming distance is either 1 or 8, which are the two extreme cases. On average we get $40 \approx 2^{5.32}$ possible values of each key byte. Therefore, on average, the final key hypotheses is $(2^{5.32})^{16} = 2^{85.15}$, **which can not be brute-forced easily**.

TABLE II
ATTACK RESULTS ON 16 SCAN CHAINS AND 8 SLICES

No. of X-OR Tree	Maximum	Minimum	Average
1	144	4	40
2	144	4	40

VI. IMPACT OF OTHER DESIGN PARAMETERS

In this section, we briefly analyze the impact of other design decisions on scan attack.

1) **Iterative vs pipelined:** As scan attack targets only one round of AES, the only difference potentially is the input register, which must exist in the pipelined design, but may or may not exist in the iterative design. Therefore, the pipelined implementation is easier to attack by targeting the pre-round operation as shown in Section IV-B.
2) **128-bit vs. 32-bit vs. 8-bit:** In the 32-bit design, the number of internal registers in a round is more than that in the 128-bit design. This is because, in the 32-bit design, a MC module is repeated four times for each round operation, with the result being loaded in a intermediate register each time. The attacker can use this register to leak information about the individual MC operations. Similarly, the 8-bit design contains more internal registers to hold the results of SB and SR operations. These internal registers could leak more information, and thus ease the attack procedure.
3) 10 **or** 12 **or** 14 **rounds:** The number of rounds does not have much affect on the scan attack. For 12-round and 14-round designs (AES with 192-bit and 256-bit key), the attacker needs to retrieve two consecutive round keys. He/she can apply the attack on the first round and retrieve the whitening key. Using the whitening key value, he/she can repeat the attack for the second round. However, for each possible key hypotheses of the whitening key, he/she has to repeat the attack in the second round and get the 2^{16} possible first round keys. Therefore, in total, the final key hypotheses is $2^{16} \times 2^{16} = 2^{32}$.

VII. CONCLUSIONS

In this work we perform scan attack analysis on different designs of AES. We show that the attack feasibility and complexity vary between different implementations. We perform our case study on various 128-bit iterative designs of AES, which vary in the round register position. We compare our attack results with the existing attacks. The results show that placing the round register in between the pre-computation and the round operation is more vulnerable to scan attack. On the other hand placing the round register exactly after the S-Box operation provides inherent resilience to scan attack when a contemporary scan architecture is incorporated in the chip.

While we also touch upon the impact of other design parameters such as iterative versus pipelined, datapath width, and number of rounds, in the future we would like to extend our analysis to the other two designs (32-bit and 8-bit) of AES. We will also extend our work to other symmetric key ciphers to see how the cipher primitives affect the analysis. This will help the designers evaluate their designs against scan attacks and enable them to make informed decisions.

REFERENCES

[1] B. Yang, K. Wu, and R. Karri, "Scan Based Side Channel Attack on Dedicated Hardware Implementations of Data Encryption Standard," in *ITC 2004*. IEEE, 2004, pp. 339–344.

[2] B. Yang, K. Wu, and R. Karri, "Secure scan: a design-for-test architecture for crypto chips," in *DAC 2005*. ACM, 2005, pp. 135–140.

[3] R. Nara, K. Satoh, M. Yanagisawa, T. Ohtsuki, and N. Togawa, "Scan-Based Side-Channel Attack against RSA Cryptosystems Using Scan Signatures," *IEICE Transactions*, vol. 93-A, no. 12, pp. 2481–2489, 2010.

[4] J. DaRolt, A. Das, G. D. Natale, M.-L. Flottes, B. Rouzeyre, and I. Verbauwhede, "A scan-based attack on Elliptic Curve Cryptosystems in presence of industrial Design-for-Testability structures," in *DFT 2012*. IEEE, 2012, pp. 43–48.

[5] R. Kapur, "Security vs. Test Quality: Are they mutually exclusive?" in *ITC 2004*. IEEE, 2004, p. 1414.

[6] B. Ege, A. Das, S. Ghosh, and I. Verbauwhede, "Differential Scan Attack on AES with X-tolerant and X-masked Test Response Compactor," in *DSD 2012*. IEEE, 2012, pp. 545–552.

[7] J. DaRolt, G. D. Natale, M.-L. Flottes, and B. Rouzeyre, "Scan Attacks and Countermeasures in Presence of Scan Response Compactors," in *ETS 2011*. IEEE, 2011, pp. 19–24.

[8] "Specification for the Advanced Encryption Standard (AES)," FIPS 197, 2001. [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[9] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A Compact Rijndael Hardware Architecture with S-Box Optimization," in *ASIACRYPT*, vol. 2248. Springer, 2001, pp. 239–254.

[10] K. Gaj and P. Chodowiec, "FPGA and ASIC Implementations of AES," in *Cryptographic Engineering*, Çetin Kaya Koç, Ed. Springer, 2009, pp. 235–294.

[11] "Cryptographic Hardware Project, Aoki Laboratory," http://www.aoki.ecei.tohoku.ac.jp/crypto/web/cores.html.

[12] "AES Hardware Macro Specification," http://www.aoki.ecei.tohoku.ac.jp/crypto/items/AESSpec2007Sep25.pdf.

[13] C. Liu and Y. Huang, "Effects of embedded decompression and compaction architectures on side-channel attack resistance," in *VTS 2007*. IEEE, 2007, pp. 461–468.

[14] J. DaRolt, A. Das, G. D. Natale, M.-L. Flottes, B. Rouzeyre, and I. Verbauwhede, "A New Scan Attack on RSA in Presence of Industrial Countermeasures," in *COSADE 2012*, vol. 7275. Springer, 2012, pp. 89–104.

[15] *Proceedings 2004 International Test Conference (ITC 2004), October 26-28, 2004, Charlotte, NC, USA*. IEEE, 2003.