New Security Threats Against Chips Containing Scan Chain Structures

Jean da Rolt, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre

New security threats against chips containing scan chain structures

Jean Da Rolt, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre
LIRMM (Université Montpellier II /CNRS UMR 5506)
Montpellier, France
{darolt, dinatale, flottes, rouzeyre}@lirmm.fr

***Abstract*—Insertion of scan chains is the most common technique to ensure observability and controllability of sequential elements in an IC. However, when the chip deals with secret information, the scan chain can be used as back door for accessing secret (or hidden) information, and thus jeopardize the overall security. Several scan-based attacks on cryptographic functions have been described and showed the need for secure scan implementations. These attacks assume a single scan chain. However the conception of large designs and restrictions in terms of test costs may require the implementation of many scan chains and additional test infrastructures for test response compaction. In this paper, we present a new generic scan attack that covers a wide range of industrial test infrastructures, including spatial response compressors.**

Keywords – security, testability, scan-based attack

I. Introduction

While scan insertion is one of the most popular Design for Testability (DfT) methods, its use for secure devices, smart cards for instance, opens a backdoor for security threats. “Scan attacks” (e.g. [1], [2]) exploit facilities offered by scan chains to retrieve embedded secret data, e.g. secret encryption keys. These attacks rely on the possibility for hackers to shift out the scan chain content while the circuit contains data correlated with the secret. More precisely, they rely on the possibility to switch the device from mission mode to test mode in order to observe intermediate states of the circuit by means of scan-out operations.

Several counter-measures have been proposed to face out these scan attacks. A common industrial practice consists in letting the scan pins unbound, or adding anti-fuses on scan pins and blowing them after manufacturing test. The DfT flow is not affected in this case. However, these solutions present several drawbacks, the maintenance in the field is compromised and the scan chain can still be controlled and observed with probing. Other counter-measures aim to provide secure scan-based DfT flows, they can be classified into methods that secure the control of the chip [1][3], they involve power-off or reset of scan flip-flops when switching from mission to test mode, methods that detect unauthorized scan shifts by mean of scan pattern watermarking [4][5][6], scan-enable tree monitoring [7], or insertion of spy FFs into the scan chains [8], and methods that provide confusion in the stream shifted out from the scan chain [9][10][11][12].

While these techniques initially address single scan chain circuits, other test architectures must be considered as well, for instance, multiple scan chains with decompression of test vectors and spatial compaction of test responses, which are proven to not affect fault detection and diagnosis [13][14]. Since the compaction reduces the observation of scan-out responses, it could be thought that it sufficiently increases the complexity of the scan attack for preventing such practice. In [13], the authors claim that embedded vector decompression and response compaction lead to security improvements, and that the attack proposed in [1] is not valid anymore. However the increase in the security is based on two assumptions. First, the attacker does not know the vector decompression structure and thus he has to work around. Secondly, it is not possible to retrieve the secret after the response compaction. The first assumption is weak from a security point of view, being based on the obfuscation principle. Moreover, scan attacks do not necessarily rely on controlling the internal state of the circuit by means of the scan chain, but conversely, rely on observing the internal state. In other words, the control through the scan chain is not a must for attacking the circuit.

This paper aims at showing that, conversely to the second assumption, it is possible to retrieve the secret from compacted responses, even if the amount of information related to the secret key is extremely decreased. For instance this is the case when the responses are compressed into a single parity bit before observation, as in most of the current industrial test solutions. Still, the attack proposed in Section III also proves to be useful in the case where almost any FF containing information related to the secret data is observable. The proposed method for attacking the circuit is of low complexity and the attack time may be negligible.

This paper is organized as follows: Section II reviews the Advanced Encryption Standard (AES) implementations on which experiments will be presented and lays the foundations of scan attacks. Section III presents the proposed attack and discusses optimizations in terms of number of input messages to be loaded on the target circuit. Several attack scenarios are presented in the Section IV covering up most of the industrial test schemes. Finally, we draw some conclusions in section V.

II. Scan-based Attacks on the Advanced Encryption Standard

As a support secure circuit example thorough this paper, we consider the case of a circuit including a crypto-core implementing the Advanced Encryption Standard (AES).

A. Advanced Ecryption Standard

The AES was adopted by the US government (FIPS-197, 2001) as a standard for symmetric encryption. The AES

This work is supported by Région Languedoc-Roussillon under the project "Prosecure"

978-1-4577-1057-5/11/$26.00 ©2011 IEEE

algorithm is a symmetric block cipher that can encrypt and decrypt information by 128-bits data blocks. Encryption converts a plaintext to an unintelligible form called cipher text; while decryption converts the cipher text back to its original form. Encryption and decryption use the same cryptographic secret key of 128, 192, or 256 bits. We focus here on the encryption algorithm with 128-bits cryptographic keys (details are fully described in [16]).

The basic unit for processing in the AES algorithm is the byte. Input, output and secret key bit sequences are internally processed on a two-dimensional array of bytes called the State. The State consists of four rows of four bytes.

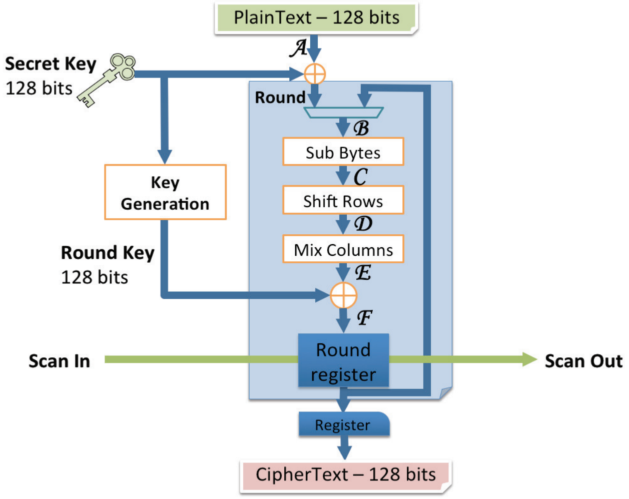

Figure 1. AES Algorithm

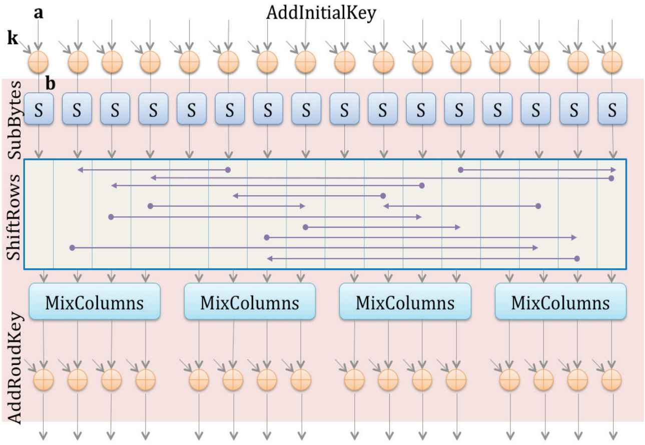

Figure 2. AES' first round

Fig.1 summarizes the AES algorithm. The plaintext is first copied to the State array then xored with the secret key. Then, the State array is transformed by implementing a round function that is repeated 10 times, with the final round slightly differing from the first 9 rounds. The final State is then copied to the output. The round function is parameterized using a Key Expansion function that generates a variation of the original secret key for each round. Round function is composed of 4 operations: SubBytes, ShiftRows, MixColumns, and AddRoundKey. These functions operate and modify the value of the State. The round-keys are processed in similar way. A schematic view of the round function is shown in Fig.2 in which each wire is one byte wide.

Pipelined and non-pipelined, as well as 8, 32 or 128 bits datapath implementations have been proposed. In any case, the result of a round is stored within a register, called round-register hereafter, which is the target of scan-based attacks.

B. *Scan-Based Attack*

The general principle of the scan attacks consists in observing the data stored in the round-register after the execution of the first round for several known plaintexts by means of scan-out operations, and then, from these observations, to derive the secret key. For this purpose we recall hereafter the principle of the attack proposed in [1] since it will serve as a basis for the proposed attacks.

The attack relies on three requirements:

1) the possibility to switch from mission mode to test mode, which allows the attacker to "stop" the cipher operation at any moment (typically after the first round);
2) the possibility to control somehow the input plaintext (e.g. as a primary input of the circuit) and to observe intermediate states by means of scan out;
3) reseting the circuit and performing one AES round with a first plaintext and doing the same operations with another plaintext only affect the FFs belonging to the round-register. In other words only the round-register FFs depends on he plaintext value after the first round.

If these requirements are fulfilled, then the procedure to retrieve the information is performed by the following steps: 1) reset the circuit and process a plaintext for a single round, 2) switch to test mode and scan out the response $\mathcal{F}$, 3) reset the circuit and encrypt another plaintext, 4/ switch to test mode and scan out the response $\mathcal{F}'$. Since only FFs belonging to the round-register may have their value changed between step 2 and 4 (due to third requirement), one can determine for instance some of the FFs of the round-register amongst all the FFs of the design by comparing the two scan responses. For example, if the first scan-out sequence is 01001100011001... and the second one is 00011101011000..., one can infer that the 2nd, 4th, 8th and the 14th FFs belong to the round-register.

Actually, for the attack described in [1] and the ones presented in this paper, there is no need to locate the FFs belonging to the round-register, because key retrieval is performed by examining the difference (Hamming distance) between the two scan-out contents $\mathcal{F}$ and $\mathcal{F}'$. Since the FFs that do not belong to the round-register are not supposed to flip between two plaintexts, they do not affect the Hamming distance value. This feature permits the attacker to calculate the Hamming distance without caring about which FFs correspond to the AES ones.

It must be noticed that the Hamming distance is the same as before the AddRoundKey operation (shown in Fig. 2), i.e. the dependence on the round-key is thus eliminated from the observation when the attack is differential. Besides of removing the effects of the round-key, working in differential mode also permits the division of the AES round in 16 independent datapaths, since when changing an AES input corresponding to one MixColumns and keeping the inputs corresponding to the other MixColumns constant will lead to changes only in the 32 output bits of the MixColumns which input has changed, thus the 96 bits from the other MixColumns outputs keep constant and are eliminated by the Hamming distance. Indeed each byte of the main key may be retrieved while analyzing the output of its respective MixColumns block, and thus for the 16 bytes of the key the attack may be performed independently. This is the main aspect responsible for the reduction of the complexity of the AES round, allowing the scan attacker to quickly retrieve the secret.

The attack described in [1] uses the same base previously described, however it relies on an analysis of the distribution of the Hamming distances between $\mathcal{F}$ and $\mathcal{F}'$. In order to have the Hamming distances the input pairs are chosen as a and $a' = a \oplus 1$ (one pair element differs from the other just at the least significant bit). Since the attack is performed for each byte of the key the length of a is a byte, and so the total number of pairs is 128 (256 input vectors). Then the attacker must find one Hamming distance between 9, 12, 23 or 24, which are generated by only one input pair. Each one of these special hamming distances has a constant pair of b (shown in Fig. 2), thus the subkey is $\mathcal{K}_1 s = a \oplus b$ or $\mathcal{K}_2 s = a \oplus b \oplus 1$. The same process is performed on the other 15 bytes of the input for retrieving the other 15 bytes of K. The final choice between the $\mathcal{K}_1 s$ and $\mathcal{K}_2 s$ can be done by checking an encryption result with respect to the 2^{16} possibilities.

The target of this attack is the single or multiple chain schemes, however other industrial test schemes exist (e.g. response compaction) where this attack is not valid anymore. For this purpose the attack proposed in the following section may be used instead, since it may be applied to a wide range of scenarios as single and multiple chains (Subsection IV.A), partially protected circuits (Subsection IV.B) and many spatial compaction schemes (Subsection IV.C).

III. Signature Attack

The previously described attack imposes the need to access the whole round-register in order to calculate the hamming distance between two values. However, it is possible that the attacker is not able to observe directly all 128 bits of the round-register. For instance, in presence of test response compaction, only a digest of the round-register is available out of the chip bounds and the previous attack can no longer be carried out. The goal of this paper is to present a new attack that aims at recovering the secret key while observing a digest or part of the information contained in the FFs related to the secret,. By repeating this procedure the whole secret key is recovered. Another improvement provided by this new attack is that the number of input messages required for successfully retrieving the key is reduced in comparison to the other attack.

The attack is divided in three parts: the pre-attack phase where the circuit is modeled and the observable responses are stored in a table; the practical part where the input messages are actually loaded in the chip and the observed output information is collected; and finally the signatures of the collected information are compared to the pre-phase tables revealing thus the secret key.

In the first phase the device implementing the crypto algorithm is modeled and logic simulation is used to predict its behavior. In this pre-attack phase the attacker must know specifically which information can be observed, for instance it may usually be the whole round-register value, as the attack described in [1], or it may be the parity of the round-register, which is the case of response compaction schemes. In the Section III, we show that the observed value is not necessarily obtained from the scan chain, Other side-channels (as chip probing) may be used as well with this attack.

Fig. 3 shows a generic crypto block model and the signature table built in the pre-attack phase. For each secret key value K, all the M possible input data D are simulated and the observed signal S is stored in the table at the right side, creating signatures for all N keys. This procedure is complete when all possible values of both key and input data are covered. It must be noticed that the width of the S elements is exactly the number of observed bits.

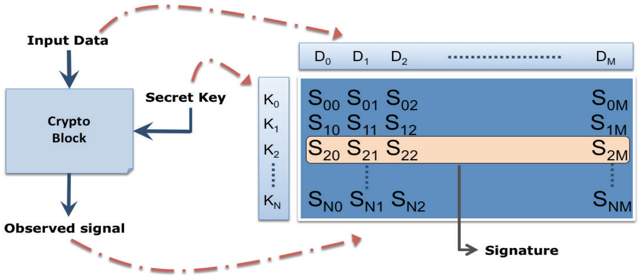
Input Data
Crypto Block
Secret Key
Observed signal
D_0 D_1 D_2 .... D_M
K_0
K_1
K_2
K_N
S_{00} S_{01} S_{02} S_{0M}
S_{10} S_{11} S_{12} S_{1M}
S_{20} S_{21} S_{22} S_{2M}
S_{N0} S_{N1} S_{N2} S_{NM}
Signature

Figure 3. Pre-attack phase

Considering the AES as the targeted crypto block, it is known that, when differential attacks are used (when the used output is actually the difference between two measured outputs), the first round can be decomposed in 16 independent datapaths of 8-bit inputs, where each 8 bits of key affects 32 bits of the round-register. In this case, the pre-attack phase consists on generating 16 signature tables, where for all the 256 possibilities of sub-key there is one signature. Concerning the input data range, one can choose all the possible pairs that differ just at the less significant bit, so there will be 128 pairs. It should be noticed that as in the attack described in Section II, this attack considers that the signature is composed of differential information (e.g. hamming code), once this mode allows us to eliminate the dependency of round key added by the AddKey layer. For the DES, the situation is even easier since each round-register bit depends on 6 key bits only and 7 plaintext bits after one round.

After the simulation is over and the signature table is complete, the attacker may start to load the vectors D at the input of the real circuit. This procedure is the very same of the attack described in the Section II: first, the circuit is reset, secondly a message is loaded at the input of the crypto chip, then the cipher encrypts (just for one round) the message using the secret key (while in normal mode), and finally the attacker force the circuit to enter in test mode and scan out all the data stored in the scan chains. Since we suppose that the circuit is reset after each step and only the AES input message is changed, it implies that only the round-register bits may change. Thus calculating the hamming distance between two scan chains leads to the hamming distance of the desired signal.

Finally, the unauthorized user will proceed by loading at the input of the crypto circuit the messages corresponding to the first row in the signature table. If the collected signal does not correspond to the value stored in one line at the first row, the key respective to that line cannot be the secret key. In doing so, after all pairs are finished there will be only one key left, which is the correct one.

A. Finding the minimum number of input vectors

One important aspect of the proposed signature analysis is that it is not necessary to simulate all the possible values for the input message. With a reduced number of input vectors it is possible to find a unique signature for each key. By unique, it means that there are no two keys in the table with the same signature. This characteristic may be better illustrated on the example of the AES in the case where only one bit is observable. The signature is therefore composed by the concatenation of one bit values (for each input vector there is a bit of output). In this case one may imagine that it would be possible to represent 256 different signatures (each signature composed by 8 bits) with only 8 pairs. In practice is not easy to find these 8 pairs, however randomly choosing tens of pairs normally results in unique signatures (at the cost of negligible simulation time).

If the design implementing the crypto algorithm is not protected at all, loading tens of pairs at its input is completely feasible. However, some counter-measures that restrict the number of scan-out operations related to one secret key (and thus the quantity of observable information) may be implemented in the chip. For instance, we can imagine that the circuit contains a block that reset all the information related to the secret key if the scan-out operation has been used for some rounds. In this case, having a reduced number of input vectors is a must for the attacker and so alternative methods to find the required input vectors may be used.

In order to find the minimum number of needed vectors for a particular observed signal, we used two different state space search algorithms: random and simulated annealing. The first one chooses randomly an initial number P of input pairs and then verifies whether these pairs generate unique signatures. If so, the number P is decremented by one unit and we restart the algorithm. If not, we repeat the random choice and verification T (trials) times. If after T trials there is no set of input vectors able to generate a unique signature then the minimum number of needed vectors is the current value of P.

Simulations show that randomly searching the vectors usually results in tens input vectors. However in order to reduce the search time and reduce even more the set of pairs, simulated annealing may be used. This algorithm initially chooses a random set of pairs, with P pairs. Then the signature table is generated and the number of repeated signatures is counted to perform an evaluation of the current set of pairs. If there are no repeated signatures then the P value is reduced and we restart the search with P-1. If not, one pair between the set will be modified randomly and the new signature table is evaluated. If the evaluation gets better (the number of repeated signatures decrease) then this is the new best state. This step is repeated N times for each pair position (for example, the first pair will be changed 10 times and then the best candidate remains), and the whole procedure is repeated T trials.

In addition to using simulated annealing to perform the state space search, the form of the pairs is also chosen in such a way that the number of total vectors is the minimum for that number of pairs. For instance we choose the vectors in the form (D1, D2), (D2, D3), (D3, D4) and etc, so one vector is used in two pairs, except for the first and last pairs that contain one unique vector. In other words, for N pairs there are N+1 vectors.

IV. SCENARI

The attack model presented in the previous section may be applied to several different scenarios, depending on which information the attacker is able to observe. It is important to notice that all the most common industrial test schemes are covered by this attack: single chain, multiple chains and especially different response compaction schemes. In Subsection A, the signature attack is used in the same scenari as [1], meaning that the attacker has access to all 128 bits of the round-register. Then in Subsection B, circuits which are partially protected or where the attacker has access to a reduced number of round-register bits are proved to be attackable by the proposed technique. In addition to these cases, in Subsection C it is shown how to use the signature attack against response compaction schemes.

A. Observing the 32 bits

In the usual single chain scenario the whole round register is inserted in the scan chain (as shown in Fig. 1), meaning that the attacker may access all the 128 bits. So in the pre-attack phase, the signature table is built using the hamming distance over all the FFs in the scan chain (similarly to the existent attack of Section II). Since the attacker normally changes a reduced set of bits of the AES input message, only the 32 round-register bits affected by the correspondent MixColumns could change between two different input messages. So the hamming distance over all flip-flops is exactly the distance over the targeted 32 bits.

As remarked, the AES attack may be split in 16 parts where the data length is 8 bits and the sub-key length is also 8 bits. In this case, the signature table is composed of 256 keys, and the signature is represented by a series of hamming distances (from 0 to 32). Using the algorithm for finding the least number of input vectors results that with only 4 input vectors we can determine the value of a sub-key (8 bits of the secret key), by means of generating 256 different signatures for all the keys. For instance, the input pairs of vectors used for the first byte of the secret key are (105, 223), (223, 143) and (143, 112). It must be noticed that for each byte there is a different set of 4 input vectors and each table contains different signatures. At least, repeating the procedure 16 times lead the attacker to the 128 bits of the secret key.

This instance of the attack is specially indicated for single and multiple chain scenarios, where the attacker has access to the round-register, and then he/she can retrieve the secret key with 4 (complexity for each subkey) times 16 (number of subkeys) input vectors at worst case.

B. *Observing a particular FF*

There are many scenarios where the user may have access only to partial information on the round-register:

1) It is possible that some of the FFs from the round-register are not inserted in the scan chain (partial scan design);
2) Some FFs may be masked with the intention to protect the circuit against the attack described in [1] (while some are not masked);
3) The attacker has no access via test and then he/she may use a side-channel other than test to probe the signal from one round-register FF;

In all these cases the attack described in [1] may not be appropriated because it requires the access to the whole round-register. Considering the attack model where at least 4 bits of the round-register are observable, one per block of 32 bits (MixColumns), the signature attack may be used. Each one of these FFs depends on 32 input bits (due to the MixColumn layer, see Fig. 2) and 32 key bits of key. It must be noticed that the AddRoundKey layer is not considered because it is eliminated when in differential mode.

Unlike the case shown in Subsection A, the signature in these scenarios will contain one bit per pair (the observed bit) instead of 32 bits. However the principle remains the same, for each subkey (byte) the attacker must create a signature table in the pre-attack simulation. Each table has 256 lines (corresponding to the sub key possible value). For each simulated pair, the Hamming distance of the observed bit is stored in the table.

Table 1 shows the vectors that generate 256 unique signatures for the bit number 0 of the round-register, it may be noticed that it contains 13 vectors (12 pairs). Simulations showed that for all the 128 round-register bits there is always a set of 13 vectors able to ensure the uniqueness of the signatures.

In summary, this instance of the signature attack demonstrates that observing only one bit per MixColums, 4 bits at all, allow the attacker to retrieve all the 128 bits of the secret key. However, it imposes the need of knowing exactly which register bit is being observed. Actually, this is not tricky since the same attack may be used to retrieve which observed bit corresponds to each round-register bit: since only 32 bits may be affected by a change in one of the sub keys corresponding input, the attacker knows which are the 32 candidates for the unknown bit, in other words he/she knows to which MixColumns block the observed output correspond. At worst, he/she may generate 32 signature tables for each of the candidates. In this case, the number of pairs should be increased because the goal is to find unique signatures between different tables, in such a way that all lines in the 32 tables have no repeated signature. This increases the complexity of the attack, however, it allows the identification of the round-register bits in the scan-chain.

If the attacker already knows the position of the 4 bits he/ she is observing, than this attack has a complexity of 16 times 13 (number of subkeys times complexity of retrieving one sub key). Regarding this low complexity, we conclude that all bits in the round-register must be protected, otherwise the signature attack may be used by an attacker to retrieve the key.

C. *Attacking response compaction schemes: observing the parity*

In the current state-of-art, there is no attack that considers a very common test practice: the response compaction. In presence of a XOR-tree compactor as shown in Fig. 4, the hamming distance between two output bitstreams is not anymore the same as the hamming distance between two states of the round register. The work proposed in [15] tends to demonstrate that a decompression/compaction structure naturally protects the circuit against scan-attacks. Anyway, the signature attack described above is able to reveal the secret key even in the presence of those response compactors.

This paper concentrates on the 1-bit output compactors such as the XOR-tree depicted in Fig. 4, this case being the worst situation for an attacker. The extension to other kinds of spatial compactors is straightforward.

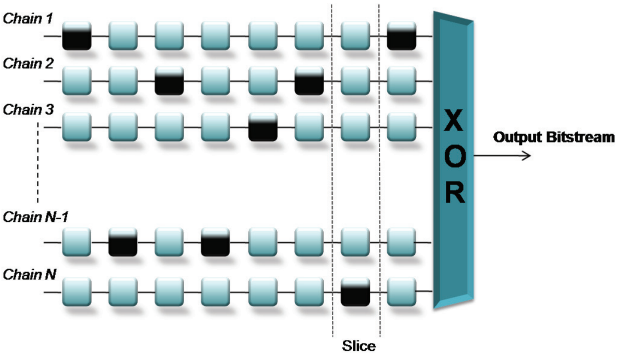

Figure 4. Response Compaction Scheme

As said in the Section III, the signature attack requires the observation of a signal that is related to a reduced number of secret key bits. The parity of the round-register may be used for this purpose. It is straightforward to measure if the parity of the round-register has changed or not (equivalent to the hamming bit over only one bit), once the output bitstream is completely unloaded, its parity is calculated and then if the resulting parity has changed from the previous bitstream, then the parity of the round-register has changed.

The pre-attack phase consists of the simulation of the AES round, where at each step one byte at the input message is changed and the parity over all round-register is stored in the signature table. This procedure generates 16 tables with 256 lines each. In the practical phase, the attacker reads two output bitstreams and calculates the difference of the parity, then he/

she searches for the measured signature in the simulation table till the good signature is found and thus the sub key.

In the same way that this instance of the signature attack works for a generic compressor, it is susceptible to be used against other proposed test structures whose parity is observable. For example, the solution proposed in [17] as well as the one proposed in [18] provides the parity as test response. Simulations using the algorithm described in Section III show that 13 vectors are enough for assuring the uniqueness of the signatures, as shown in Table 2.

Moreover, this attack also offers the advantage of being applicable for both single and multiple chain(s) scenarios. The attacker only needs to calculate the parity of the state scanned out from the scan chains. In other words, if the attacker does not know which test structure is implemented in the chip, he/ she may try this technique, as long as the parity difference can be observed.

V. Conclusion

This paper proposes a generic scan attack that is applicable to several industrial DfT scenarios: single and multiple chains, with or without response compaction structures. Besides, it is shown that state space search algorithms may be used together with this attack to find the minimum number of vectors needed to retrieve the secret key, reducing the time spent when interacting with the targeted circuit.

Considering the single or multiple chains without compaction, this attack on AES is optimized so that only 4 vectors have to be applied on the circuit to retrieve one byte of the secret key. The whole key is thus obtained with 2^6 inputs. Aside from the fact that observing only 4 bits of the round-register results in revealing the secret key, it is shown that all bits of the round-register have to be protected. Conversely, only one bit unprotected leads to divide the brute force attack complexity by a factor 2^{32}.

We have shown that the supposed protection offered by test response compaction structures has been overestimated in the literature.

Moreover the attack proposed in the presence of response compaction can be used as well for circuits without compaction. Furthermore, no special knowledge about the implemented DfT (number of scan chains, presence of response compactor) is mandatory to an attacker to succeed.

Future works will analyze the security of other kinds of compressors such as signature analyzers or convolutional compactors.

References

[1] Bo Y.; Kaijie W.; Karri R., "Secure Scan: A Design-for-Test Architecture for Crypto Chips", *IEEE Transactions on* CAD, vol.25, no.10, pp. 2287-2293, Oct. 2006

[2] Bo Y.; Kaijie W.; Karri R.; "Scan based side channel attack on dedicated hardware implementations of Data Encryption Standard," *Proc. ITC 2004*, pp. 339- 344.

[3] Hély D., Bancel F., Flottes M.-L., Rouzeyre B.. "Securing Scan Control in Crypto Chips". *J. Electron. Test.* 23, 5 (October 2007), pp. 457-464

[4] Hély D.et al., "Scan Pattern Watermarking", LATW 2006

[5] Lee J., Tebranipoor M., Plusquellic, J. "A low-cost solution for protecting IPs against scan-based side-channel attacks," *Proc. VTS 2006,* pp.93-99

[6] Paul S., Chakraborty R.S., Bhunia, S., "VIm-Scan: A Low Overhead Scan Design Approach for Protection of Secret Key in Scan-Based Secure Chips," *Proc VTS 2007.* pp.455-460.

[7] Hely D., Bancel F.; Flottes M.L., Rouzeyre, B., "Test control for secure scan designs," *Proc. ETS 2005*, pp. 190- 195.

[8] Hély D., "Testability of Secure ICs", PhD report University of Montpellier 2, 2005

[9] Mukhopadhyay D., Banerjee S., RoyChowdhury D.; Bhattacharya, B.B.; , "CryptoScan: A Secured Scan Chain Architecture", *Proc. 14th ATS 2005*, pp. 348- 353.

[10] Hely D., Flottes M.-L., Bancel, F., Rouzeyre B., Berard, N., Renovell M., "Scan design and secure chip [secure IC testing]," *Proc. IOLTS 2004.* pp. 219- 224.

[11] Lee, J.; Tehranipoor, M.; Patel, C.; Plusquellic, J.; "Securing Designs against Scan-Based Side-Channel Attacks," *IEEE Trans. on Dependable and Secure Computing,* vol.4, no.4, pp.325-336.

[12] Sengar G., Mukhopadhyay D., Chowdhury D.R., "Secured Flipped Scan-Chain Model for Crypto-Architecture," *IEEE Trans. on CAD* , vol. 26, no.11, pp.2080-2084, Nov. 2007

[13] Elm M., Wunderlich H.-J.; "Scan Chain Organization for Embedded Diagnosis", *Proc. DATE 2008,* pp.468-473.

[14] Vranken H., Kumar Goel S., Glowatz A., Schloeffel J., Hapke, F.; "Fault detection and diagnosis with parity trees for space compaction of test responses", *Proc. DAC 2006*, pp.1095-1098.

[15] Liu C., Huang Y.; "Effects of Embedded Decompression and Compaction Architectures on Side-Channel Attack Resistance," *Proc. VTS, 2007.* pp. 461-468.

[16] http://csrc.nist.gov/publications/PubsFIPS.html

[17] J. Rajski, et al., Embedded Deterministic Test for Low Cost Manufacturing Test. Proc. Int. Test Conf., pp. 301-310, 2002.

[18] S. Mitra and K. S. Kim, X-Compact: An Efficient Response Compaction Technique. IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 23, pp. 421-432, 2004

(238,198)	(198,44)	(44,86)	(86,222)	(222,60)	(60,231)	(231,4)	(4,3)	(3,175)	(175,73)	(73,144)	(144,187)

Table 1: Input vectors which create an unique signature table when a single FF is observable

(94,79)	(79,227)	(227,166)	(166,21)	(21,141)	(141,233)	(233,252)	(252,171)	(171,41)	(41,117)	(117,82)	(82,218)

Table 2: Input vectors which create an unique signature table when observing the response compression output