Security of Countermeasures Against State-of-the-Art Differential Scan Attacks

Bariş Ege*, Amitabh Das†, Lejla Batina*, Ingrid Verbauwhede†
* ICIS / Digital Security Group
Radboud University Nijmegen
Nijmegen, The Netherlands
email: B.Ege@cs.ru.nl, lejla@cs.ru.nl
† KU Leuven, ESAT/COSIC and iMinds
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
email: firstname.lastname@esat.kuleuven.be

***Abstract*—Test compression schemes have been claimed to provide a certain level of security against scan-based side-channel attacks. To mitigate these attacks, a number of scan attack countermeasures are proposed in the literature. Recently, a new differential scan attack (DSA) is proposed which focuses on the S-box outputs rather than the S-box inputs as in previous attacks. In this paper, a systematic security analysis of the most popular scan attack countermeasures against this differential scan attack is given. The countermeasures are evaluated when they are used together with industrial test compression schemes on a straightforward AES design. Security of the countermeasures is evaluated by emulating their behaviour in software, and the gain in security is experimentally investigated. Our experiments show that when the new DSA (focusing on the S-box output) is considered, both scan chain scrambling and partial scan countermeasures fail to provide sufficient security.**

I. Introduction

Design-for-test (DFT) is the test infrastructure added to a circuit to improve the controllability and observability of the internal flip-flops and nodes. It is employed in the efficient generation and application of manufacturing tests to complex circuits. Scan chains are the most efficient DFT structures used widely in the semiconductor industry nowadays. A test mode is added to the circuit in a way that when the circuit is in this mode, all flip-flops are connected in one or more shift registers. The inputs and outputs of these shift registers (also known as scan registers) are made into primary inputs and primary outputs [1].

Test compression is widely deployed in the semiconductor industry for testing complex circuits in a short time and lower costs without compromising test quality. When test vectors are generated for a circuit by an automatic test pattern generator (ATPG), most of the bit positions are unspecified, or don't care (X) states, which are randomly filled with 0s or 1s, to enable its use on an Automatic Test Equipment (ATE). These X-states can ruin the test output if not handled with care. Testing industry has two main solutions to this problem: X-masking and X-tolerant logic.

Scan chains may be permanently disabled after testing of the chip (by blowing some fuses, for instance) before being used in a product, but then the in-field testability of the chip is lost. In some applications, such as set-top box decoders, the firmware updates happens in most cases through the JTAG port internally connected to the scan chains. Hence, scan chains must be left intact.

Cryptographic circuits need a special testing strategy due to the constraints on security. Though scan-chain Design-for-Test (DFT) offers the highest testability, it is prone to scan-based side channel leakages which may enable a non-invasive attack on secure chips to extract secret information. There are scan-based attacks on symmetric-key algorithms through the test interface published in the literature [2]. In 2012, two attacks([3], [4]) on X-masking and X-tolerant logic are published pursuing two different methods. While Ege et al.[3] takes a more traditional approach and work on differences given to the plaintext, Da Rolt et al.[4] proposes to look for differences after the S-box layer in AES therefore making it possible to attack even when as little as one bit of information is leaked to the scan outputs.

There are a number of scan attack countermeasures proposed in the literature. One of the approaches is based on randomizing the scan sequence. A pseudo-random selection of scan chains is made and loaded with scan data at a time. Instead of serially transmitting the bit stream through the scan registers, the process is randomized. This scheme is also known as scan chain scrambling [5]. The 'Flipped Scan Tree' architecture [6] introduces inverters at the scan-in inputs of some of the scan flip-flops. The location of the flipped scan flip-flops in the scan tree architecture is known only to the designer and the SoC Tester, and completely unknown to an attacker. Embedded Deterministic Test (EDT) used in the popular MentorGraphics test compression tool, Tessent TestKompress, compresses the scan chains and imposes a dynamic mask on the scan outputs, and has been claimed to be secure against scan based attacks[7]. Other countermeasures include the 'Lock and Key Technique' [8], the design for secure test [9] employing an ad-hoc approach for pipelined AES, and the technique involving reset of the security chip and removing all traces of any secret information or cryptographic algorithm execution in test mode [5].

In this paper, we analyze the security of the most popular

scan attack countermeasures when they are combined with the industrial test compression schemes. We implement the attack proposed in [4] to evaluate the security provided by each scheme and provide results for each combination of "countermeasure"-"compression scheme" couples.

Structure of the paper is as follows. Previous work on scan attacks is summarized in Section II. The basic scan attack strategy is explained in Section III. The attack as applied to industrial test compression schemes, specifically Adaptive Scan from Synopsys, OPMISR from Cadence, and EDT from Mentor Graphics is also presented in the same Section. The main part of this paper is Section IV where we present the differential scan attack on industrial test compression schemes combined with three popular scan attack countermeasures, specifically scan chain scrambling, partial scan and Lock and Key Technique. Discussion on the applicability of other scan attack countermeasures is also included. A summary of the effectiveness of combined test compression and scan attack countermeasures is also given in that Section. We conclude the paper with ideas for future work in Section V.

II. Previous Work

The first attempt of analysing the security of scan testable circuits is presented by Yang et al. in 2006 [2]. The attack exploits the possibility of scanning out the contents of the round register after execution of one round of encryption or decryption. Later in 2007, Liu and Huang published an analysis [10], which also considers the response compactor of a test compression scheme. In that work, the authors focus on the Embedded Deterministic Test by Mentor Graphics, and evaluate the security of the scheme by identifying the flip-flops (FFs) which can be used for inferring the encryption key. The authors refer to these registers as key registers, and similarly the term key dependent Flip-Flops (KFFs) is used for those registers in the rest of this work. In that work, the authors claim that identification of these KFFs in the scan design is crucial for successful recovery of the encryption key. However, this has been proved wrong in later works by Da Rolt et al. [11], [12].

In [12], Da Rolt et al. present a scan based attack on an AES design with a scan response compactor, in which the identification of KFFs is not necessary for mounting a successful attack. They show that the attack proposed in [2] is directly applicable to designs which use an XOR tree structure for scan response compaction. They also provide different attack strategies for different distributions of KFFs over the scan chains. However, the scan-attack assumes a simple XOR compactor structure, without considering X-masking or X-tolerant architectures which can affect the success of the attack.

In [3], the attack proposed in [13] and improved in [12] are further extended to work against testing circuits with X-masking and X-tolerant logic (as they are used in most test compression schemes in the industry). Later in [4], a new method is proposed to perform DSA on AES circuits exploiting the linear structure of the `MixColumns` operation in AES. Here, the authors proposed to look for 1 bit differences after the `SubBytes` operation rather than providing two plaintexts with a certain Hamming difference in between. This attack is shown as effective against X-Masking schemes, Partial Scan and MISR based time compaction. Although theoretical analysis is given for the attack success in that paper, no experimental evidence is provided until now.

III. Background

In this work, scan attacks are demonstrated on AES as it is a widely used standardized block cipher and it also enables the reader to compare the work with previous works available in the literature. Since AES is a well-known block cipher we leave out the explanation of the details of it, and we refer the interested reader to [14].

A. Differential Scan Attacks on AES

This attack[2] basically exploits the fact that two particular inputs to the round function of AES can transform into output vectors with a unique Hamming distance in between after one round of encryption. For instance, if two plaintexts with an XOR difference of `0x01` in their least significant byte (LSB), are encrypted using only one round of AES, the Hamming distance between the one round output vectors can only have a handful of values. A one byte difference in the plaintext will transform into a four byte difference due to the structure of the `MixColumn` operation. Analysing the distribution of the Hamming distances for all 2^7 pairs generated with the byte difference `0x01` in their LSB, one can easily verify that there are four Hamming distance values (9, 12, 23 and 24) which can only be generated by a unique pair of inputs. Therefore, whenever such a Hamming distance is observed between the output vectors, one can XOR the corresponding plaintext byte with the pre-computed value to recover a byte of the encryption key.

Another approach to perform differential scan attack is to focus on the S-box outputs [4]. Rather than encrypting two plaintexts with a certain byte difference in between, one can also generate plaintext pairs which give a certain difference after the `SubBytes` operation, for a given(or guessed) key. If a fixed difference can be achieved after the S-box, the linear structure of the `MixColumns` operation will distribute the difference over the state always the same way. In other words, XOR of the first round outputs corresponding to the given plaintext pair is always a fixed value if the key guess is correct. Therefore, as long as the same test parameters can be set, XOR difference of the test outputs should be exactly the same, leading to much more powerful attacks than the earlier efforts.

These attacks are both based on evaluating the Hamming distances between the pairs of outputs after one round AES for different number of input pairs of plaintexts. Figure 1 shows the distribution of KFFs in the scan chains of a hardware design. As illustrated in the figure, a column of scan flip-flops containing one corresponding flip-flop for each scan-chain represents a slice. The flip-flops denoted by F_{ij} are ordinary scan flip-flops, whereas the flip-flops containing a

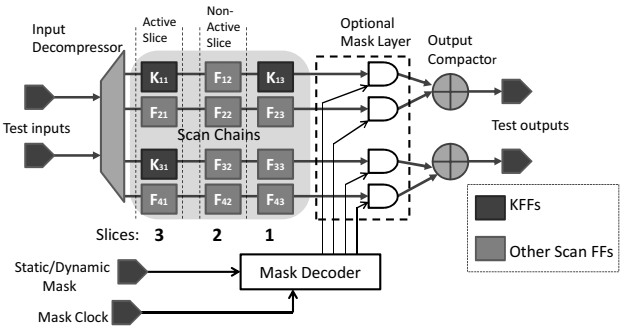

Fig. 1. Slices and active slices.

key bit is denoted by K_{ij}, where i stands for the scan-chain number and j indicates the position of the respective flip-flop in the scan-chain. Any slice containing one or more KFFs is called an active slice, while the others are called non-active slices. For instance in the figure, the slice containing KFFs K_{11} and K_{31} represents an active slice.

B. Industrial Test Compression Schemes

Different EDA vendors use different strategies for test compression but the most vendors including Mentor Graphics, Cadence and Synopsys, agree on using either X-tolerant logic or X-masking to deal with X-states in scan chains. Details of different industrial test compression schemes can be found in [1]. This is particularly important when analyzing the security of these systems when they are used in crypto chips since these systems affect the attack approach and eventually the success rate of the attack.

X-tolerant logic generally has multiple test outputs to ensure that there is always at least one test output which is not corrupted by X-states. These test outputs are basically the XOR of a collection of scan chain outputs and usually the same scan chain output is used in generating multiple test outputs.

Different from X-tolerant logic, X-masking is implemented with the aid of a mask register and AND gates at each scan chain output (see Fig. 1). The value in the mask register determines which scan chains are going to be included in the test output and this value can also be updated with a certain frequency. If the mask register is not updated during test (as in Cadence OPMISR) we refer to it as static X-masking. If the mask register is updated during test, then we refer to the system as dynamic X-masking (as in Mentor Graphics EDT).

IV. Combined Scan attack on AES with test compression and scan attack countermeasures in place

As almost all complex circuits contain some degree of test compression nowadays, it is worthwhile to evaluate their interaction with scan attack countermeasures. In this section, we investigate the effectiveness of scan attack countermeasures when they are used together with the leading test compression schemes. For this, we simulated the behaviour of the compaction algorithms of the test compression schemes, and also the effect of countermeasures are emulated in software. This approach enables us to get a reliable evaluation on the effectiveness of the countermeasures in reasonable time. The details of countermeasures and the evaluation in terms of security and cost are included in the following sub-sections. The DSA technique([4]) used in this paper is applied on the respective software emulation of these structures.

A. Partial Scan

We first consider the countermeasure proposed by Inoue et al. in ETS'09. This scheme (named as balanced secure scan) aims to protect non-scan registers by employing a test controller that enables the test mode only when an authentication succeeds [15]. Only a few flip-flops belonging to the secret registers are included in the scan chains. Further confusion is added to the kernel wherever a secret register is inserted in the scan chain. The partial scan methodology is represented graphically in Figure 2.

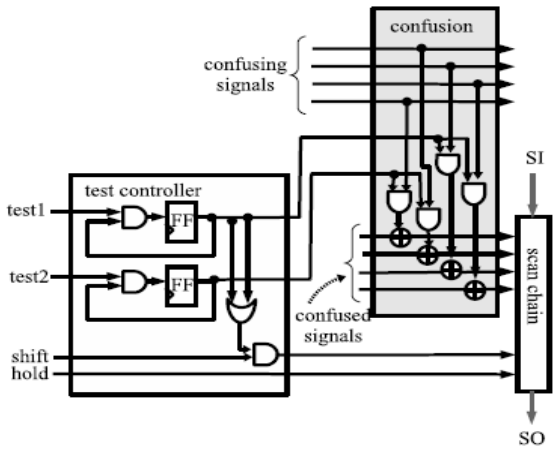

Fig. 2. Partial Scan [15].

The outline of the proposed method is as follows. First, scan registers are selected so that the kernel becomes a balanced structure and the number of FFs in secret registers selected as scan registers is minimized. Then, if some secret registers are selected as scan registers, confusion circuits are added into the kernel to randomise the values of the secret registers in test mode while preserving balanced structure.

As shown in the figure, 'test1' and 'test2' control the functioning of the test controller which in turn decides which mode the circuit will be. If either 'test1' or 'test2' is high, the circuit is in test mode; if both are low, the circuit is in normal functional mode. In test mode, the shift operation of the scan chain is enabled. Moreover during test mode, a dynamically changing mask is XORed to the key-dependent secret FFs to add confusion to them. Here 'confused signals' actually mean the secret crypto FFs which needs to be protected from an attacker. Once the circuit is in normal mode, it cannot be shifted to test mode. This is due to the FFs which maintain their states and their outputs are ANDed with 'test1' and 'test2'.

Using this method, 100% fault efficiency is demonstrated to be achieved for all the cases considered in [15] (Partial Scan DFT structures implemented on open-source RSA decryption core, 100%, 50% and 25% confusion added) with reasonable test generation time (for instance, 72.66 sec for the case 50% confusion, instead of 30.47 sec for full scan). However, the method identifies more redundant faults than full scan. The area overhead for incorporating this scheme is restricted between 6% and 9% depending on the amount of confusion circuits added.

To emulate the effect of this countermeasure in our software implementation of the attack, a random selection of KFFs, according to the chosen parameter, are excluded from the scan design. Results are given for 75%, 50%, 25% masking and with only one unmasked KFF.

The new differential scan attack presented in [4] is quite effective against partial scan combined with X-Masking and X-tolerant logic as indicated by the high success rates in Table I.

The attack is applied following the exact same methodology proposed in [4]. First, a suitable test input value which leaks information about KFFs is searched. Then, the actual attack is performed by making a key guess, and forming the input set. Later, the input set is processed through the testing circuit in pairs and the resulting test outputs are XORed together. If the output XORs of all pairs from the input set are the same, then the key byte is regarded as the most probable key byte. Whenever the guessed key matches with the actual key of the system, we regard the attack as successful.

The results given in Table I show the ratio of successful attacks over all 10 000 experiments. Repeating the attack 10 000 times took less than 24 seconds in all cases, with the set-up we have used to run simulations.

B. Scan Chain Scrambling

In this approach, the order of the scan chain elements is altered by a scrambler [5]. When the scan mode has been reached securely, the scan chain elements are ordered in a predetermined manner. However, in insecure mode, the order of the scan chains elements keeps changing at a certain frequency. Each scan chain is divided into multiple scan elements and the order of connections of the scan elements is controlled through the scan chain scrambler. The scan chain scrambling methodology is represented graphically in Figure 3.

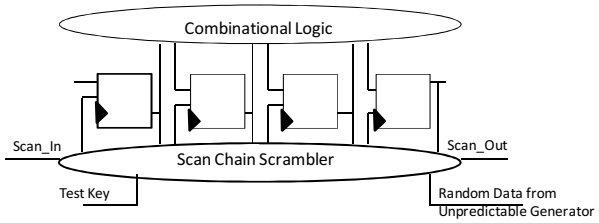

Fig. 3. Scan Chain Scrambling.

The extra area requirement of this scheme is quite small. Though test time can reduce using this method due to reduced size of scan chain segments, routing of the complete design can become more difficult due to modifications in the standard scan path.

This countermeasure can be emulated by changing the order in which the scan chains are connected using a pseudo-random generator at a frequency which is a fraction of that of the scan frequency. However, simulating the behavior of scan chain scrambling countermeasure is a bit challenging as there are no clear explanations in the paper as to how it should be implemented. Therefore, we chose to simulate the effect of this countermeasure by starting from a KFF distribution with 32 active scan chains and 32 active slices, and randomly re-ordering rows and columns of this distribution. In the end, we get a KFF distribution with different number of active slices and active scan chains. For this work, we generated 1000 random distributions to analyse the effect of this countermeasure.

Table I shows that scan chain scrambling method is transparent to the attack outlined in this work. This is because the structure of the scan chain scrambling is assumed to depend on the test inputs. Therefore, as long as the same test input is used for all the elements of the input set, the attack is expected to be successful.

Repeating the attack 1000 times took less than 19 seconds in all cases, with the set-up we have used to run simulations.

C. Lock and Key Technique

The Lock and Key technique [8] is intended at preventing malicious attackers from revealing secret information stored in the chip. The scan chains are divided into smaller sub-chains of equal length and a random selection of the sub-chain is made when an unauthorized user attempts to access the scan chains by switching to the insecure test mode. Thus, malicious users cannot predict where in the scan chain the stimuli on the scan inputs (SIs) goes and where the response from the scan outputs (SOs) comes from. Test vectors are not sequentially shifted into each sub-chain but instead a LFSR selects a random sub-chain to be filled. The general structure is represented in Figure 4.

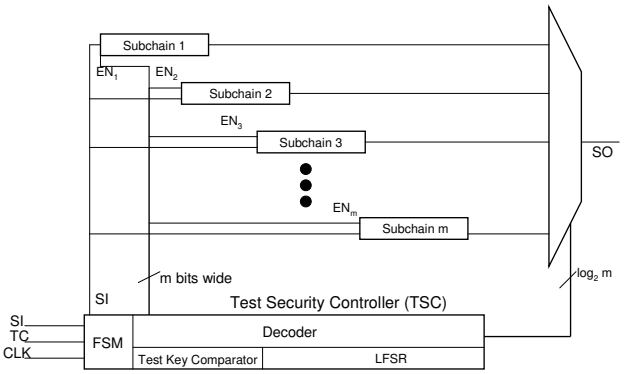

Fig. 4. Architecture of the Lock and key technique.

TABLE I
SUCCESS RATES FOR THE ATTACK[4] ON TEST COMPRESSION SCHEMES WITH COUNTERMEASURES

	Partial Scan - 75%	Partial Scan - 50%	Partial Scan - 25%	Scrambling
X-tolerant logic	100%	100%	100%	100%
Static X-masking	100%	100%	99.63%	100%
Dynamic X-masking	100%	100%	99.59%	100%

When the circuit under test (CUT) is initially reset, a Finite State Machine (FSM) sets the Test Security Controller (TSC) into insecure mode and will remain in this insecure state until TC is enabled. It is only after TC has been enabled for the first time and a test key has been entered that the TSC may exit the insecure state. When a test key is entered and a user has been ensured to be a trusted user, the FSM allows the TSC to enter secure mode allowing predictable operation of the scan chains and will remain in this state until the CUT is reset. Otherwise, the TSC will remain in insecure mode and the behaviour of the scan chain will not longer be predictable. If the entered key fails, the TSC remains in insecure mode and will seed the LFSR with an unpredictable random seed, essentially locking the scan chains from being used correctly. Since the choice of sub-chain is pseudo-random due to the LFSR, it is difficult to predict the response on SO if both the seed and the configuration of the LFSR are unknown. In insecure mode, the scan configuration keeps changing with each test clock and is unpredictable. The predictable behaviour of the LFSR primitive polynomial is removed when functioning in insecure mode for sufficiently long time. This is done by changing the LFSR configuration in insecure mode by using some additional bits (active only in insecure mode through multiplexers) which changes the feedback to the LFSR. Moreover, the FSM can be configured to generate a new random LFSR seed every round to further make scan attacks difficult.

Here, it should be noted that, in insecure mode, no two test inputs are processed in the same way, and thereby destroying any chance of mounting a successful differential attack.

D. Other Countermeasures

There are also other scan attack countermeasures such as insertion of inverters [6], Design for Secure Test [9], Resetting crypto chip in test mode [5], and masking schemes [12] that were not evaluated experimentally in this work. These schemes are briefly discussed qualitatively in this sub-section for the sake of completeness.

The insertion of inverters [6] is completely transparent to differential scan attacks, even though the locations of the secret inverters in the scan chains are kept secret. The reason behind this is that the position of the inverters is kept fixed in the scan path, and when the XOR difference of the scan data from two scan outputs is taken (as is done in differential scan attacks), their effect is neutralised.

The Design for Secure Test [9] which checks the parity of consecutive AES rounds is an ad-hoc solution for AES designs with a completely unrolled structure (having high area requirements), limiting its applicability to other designs. Though the scheme involving resetting the crypto chip and removing all traces of cryptographic execution in test mode [5] provides a high level of security. The scan attack employed in this paper is not effective against these schemes as all there is no secret stored in the round register. However, this scheme has the limitation that it is not applicable for implementations where there is a requirement to store some kind of secret data on-chip.

The Masking schemes proposed in [12] which mask each bit of the round register are effective against scan attacks. Therefore, the scan attack employed in this paper does not work against these schemes as all the round register flip flops are masked. However, these schemes have somewhat higher area overhead.

V. CONCLUSIONS AND FUTURE WORK

In this work we give the first comparative study of the two most popular scan attack countermeasures when they are used together with popular test compression strategies: XOR space compaction with X-masking and X-tolerant logic. Results suggest that the new differential scan attack which focuses on the S-box outputs is effective against all test compression schemes.

Since no particular information about the X-tolerant compactor is exploited in this work, a possible future work can be to study the design of the compactor and see if it is possible to increase the success rates by exploiting its structure. This is very important to investigate as it makes it possible to reverse engineer the structure of the compactor.

REFERENCES

[1] Laung-Terng Wang, Cheng-Wen Wu, and Xiaoqing Wen. *VLSI Test Principles and Architectures: Design for Testability (Systems on Silicon)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[2] Bo Yang, Kaijie Wu, and R. Karri. Secure scan: A design-for-test architecture for crypto chips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,*, 25(10):2287 –2293, oct. 2006.

[3] B. Ege, A. Das, S. Ghosh, and I. Verbauwhede. Differential scan attack on AES with X-tolerant and X-masked test response compactor. *Proceedings of the 15th IEEE Euromicro Conference on Digital System Design(DSD)*, pages 545 – 552, 2012.

[4] J. Da Rolt, G. Di Natale, and B. Flottes, M-L. Rouzeyre. Are advanced DfT structures sufficient for preventing scan-attacks? *Proceedings of the 30th IEEE VLSI Test Symposium (VTS12)*, pages 325 – 336, 2012.

[5] D. Hely, M.-L. Flottes, F. Bancel, B. Rouzeyre, N. Berard, and M. Renovell. Scan design and secure chip [secure IC testing]. In *On-Line Testing Symposium, 2004. IOLTS 2004. Proceedings. 10th IEEE International*, pages 219 – 224, july 2004.

[6] G. Sengar, D. Mukhopadhyay, and D.R. Chowdhury. Secured flipped scan-chain model for crypto-architecture. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 26(11):2080 – 2084, nov. 2007.

[7] Silicon test and yield analysis whitepaper - high quality test solutions for secure applications. Technical report, Mentor Graphics, 2010.

[8] Jeremy Lee, Mohammed Tehranipoor, Chintan Patel, and Jim Plusquellic. Securing scan design using lock and key technique. In *in IEEE Intl. Symposium on Defect and Fault Tolerance (DFT'05*, pages 51–62, 2005.

[9] Y. Shi, N. Togawa, M. Yanagisawa, and T. Ohtsuki. Design for secure test - a case study on pipelined advanced encryption standard. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pages 149 –152, may 2007.

[10] Chunsheng Liu and Yu Huang. Effects of embedded decompression and compaction architectures on side-channel attack resistance. In *VLSI Test Symposium, 2007. 25th IEEE*, pages 461 –468, may 2007.

[11] Jean Da Rolt, Giorgio Di Natale, Marie-Lise Flottes, and Bruno Rouzeyre. New Security Threats Against Chips Containing Scan Chain Structures. In *HOST'11: IEEE International Symposium on Hardware-Oriented Security and Trust*, June 2011.

[12] J. DaRolt, G. Di Natale, M.-L. Flottes, and B. Rouzeyre. Scan attacks and countermeasures in presence of scan response compactors. In *European Test Symposium (ETS), 2011 16th IEEE*, pages 19 –24, may 2011.

[13] B. Yang, K. Wu, and R. Karri. Secure scan: A design-for-test architecture for crypto chips. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, pages 135 – 140, June 2005.

[14] Joan Daemen and Vincent Rijmen. *The Design of Rijndael*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[15] M. Inoue, T. Yoneda, M. Hasegawa, and H. Fujiwara. Partial scan approach for secret information protection. In *Test Symposium, 2009 14th IEEE European*, pages 143 –148, may 2009.