Scan chain encryption for the test, diagnosis and debug of secure circuits

Mathieu da Silva, Marie-Lise Flottes, Giorgio Di Natale, Bruno Rouzeyre, Paolo Prinetto, Marco Restifo

Scan Chain Encryption for the Test, Diagnosis and Debug of Secure Circuits

Mathieu Da Silva, Marie-lise Flottes,
Giorgio Di Natale, Bruno Rouzeyre
LIRMM (Université Montpellier/CNRS)
161 rue Ada, Montpellier, France
{mathieu.da-silva,flottes,dinatale,rouzeyre}@lirmm.fr

Paolo Prinetto, Marco Restifo
Politecnico di Torino, Dip. di Automatica e Informatica
& CINI Cyber Security National Lab
Torino, Italy
paolo.prinetto@polito.it

***Abstract*—Scan attacks exploit facilities offered by scan chains to retrieve embedded secret data, in particular secret keys used in crypto-processors for encoding information in such a way that only knowledge of the secret key allows to access it. This paper presents a scan attack countermeasure based on the encryption of the scan chain content. The goal is to counteract the security threats and, at the same time, to preserve test efficiency, diagnosis and debugging abilities. We propose to use the secret-key management policy embedded in the device under test in order to encrypt both control and observed data at test time. This solution does not require additional key management, provides same test/diagnostic and debug facilities as under classical scan design with marginal impacts on area and test time.**

Keywords—Test and Security; Scan Attacks Countermeasure; Light Encryption

I. Introduction

Testing is used for weeding out bad products before they reach end-users and is considered as a necessary task in the IC production process to ensure quality. When it comes to the area of digital testing, many research works have been dedicated to this task for ending up today in an acceptable tradeoff between efforts dedicated to test pattern generation and Design for Testability (DfT). Most popular DfT methods rely on scan design. This structured approach consists in replacing original registers by scan registers, connecting these scan registers into one or several scan chains, where extra serial input/output provide a mechanism for serially controlling and observing registers' states at test time. Scan design greatly reduces the complexity of test pattern generation from sequential to combinatorial tests, much easier to generate.

While testability is positively impacted by full control and observation on internal states, the confidentiality of processed data is negatively affected by this design technique for exactly the same reasons. Free control and observation of IC internal states related to a secret have therefore to be avoided.

Scan attacks indeed exploit facilities offered by the scan chain to retrieve the secret data [1][4]. These attacks target secure circuits implementing a cryptographic algorithm and storing a secret key. They rely on the possibility for hackers to shift out the scan chain's content while the circuit state is correlated with the secret, i.e. the key. As a first line of defense, it is commonly assumed that the secret key must not be used at test time. The key register is thus not part of the scan chain for preventing direct observation of the secret. For the same reasons, a key-for-test is used at test time when internal states can be fully observed. This second constraint prevents observation of circuit states while processed information are related to the secret. These two conditions are however not sufficient for preventing scan attacks, which rely on the possibility to switch the device from mission mode, where the data are related to the secret, to test mode, where the data can be observed through shift out operations. The attack is possible because the scan chain allows observing the intermediate states of the crypto-processor. Observing the final state, i.e. the cypher text, would not provide useful information on the secret key even with the knowledge or control of the plain text on crypto-processor inputs. Cryptanalysis is indeed assumed not computationally tractable on current encryption standards. However, observation of intermediate states provides useful information on the key. Crypto-algorithms rely on two fundamental properties, obfuscation and diffusion. The first one reflects a desire to make cypher texts and the keys relation as complex as possible, the second aims to prevent statistical analysis between cypher and plain texts. These properties are carried out by dedicated operations between data-to-encrypt and key bits, and, iterations over these operations. Test facilities are used to break these expected iterations, and thus decrease obfuscation and diffusion. Thanks to the scan chain, the intermediate state of the crypto-processor can be observed, for instance right after first operations involving the secret key. At that time, plain text, key and cipher text relations are not too complex and key bits can be retrieved from plain-text control and observation of the intermediate state stored in the scan chain.

The main difficulty in scan-attack prevention stems from the need to maintain both data security and hardware quality provided by test, diagnostic and debug activities. The cost of the scan attack prevention in terms of circuit characteristics (performances, area), insertion in the design flow, test quality and test time are important issues. While several solutions have been proposed to prevent these scan attacks, no proposal is available for providing at low cost both high security and test / diagnostic / debug facilities to a trusted user.

The goal of this paper is to present a secure and cost efficient mechanism for full control and observation of the scan chain content. The main idea is to exploit the knowledge of the secret key originally embedded in a security-dedicated circuit in order to encrypt the scan chain itself. This way, only trusted users with the appropriate access rights, i.e., ones having access to the secret key, are able to encrypt (respectively decrypt) the bit stream shifted-in (respectively -out from) the scan chain. The knowledge of the secret key is therefore sufficient for performing all required activities, without sacrificing test efficiency, diagnostic, or debug facilities.

The remainder of this paper is organized as follow. Section II summarizes the known attacks exploiting the scan chains and related countermeasures. Section III presents the proposed secure test scheme. Section IV details related algorithms and reports on optimized implementation of the solution. Finally, Section VI concludes the paper.

II. Known Scan-Attacks and Countermeasures

In order to retrieve secret information (e.g., a secret key) from processed data, the attacker targets registers storing internal states of the embedded crypto-core. For instance, the round-register of a core implementing the Advanced Encryption Standard (AES [2]) is used when performing scan attacks on that crypto algorithm. A 128-bits AES involves a 128-bits key and proceeds on 128-bits data block, executing 10 iterations called rounds before to deliver the cipher text. The attack principle consists in observing the data stored in the round-register thanks to a scan out operation after the execution of the first AES round. It is assumed that the details of the encryption algorithm are known to the attacker since the security only relies on the secret key. It is also assumed that test access mechanisms are available (JTAG port, scan chains, test control), the attacker can choose the plaintext and is able to run exactly one round before to switch to scan out operation.

Such attacks have been first presented in [1] for DES and in [3] for AES. They are based on differential analysis on observed states and because the algorithm, the input plaintext, and the result of the 1st round operation are known to the user, it is possible to retrieve the secret key added to the plaintext during the first round. These attacks require to first understand the scan chain architecture, i.e., determine all the data that was the result of the round operation among all the data scanned out of the device. Improved scan attacks have been proposed to deal with more advanced DfT approaches such as scan response compactors, X-masking, partial scan [4], [5] where round operation data are not necessarily entirely observed on scan-out pin.

Several counter-measures have been proposed to face these scan attacks. An industrial practice consists in letting the test access mechanism unbound after manufacturing test. This solution does not affect production testing, DfT flow, nor the design itself but prevents maintenance in the field. Moreover, probing techniques can be easily set to re-connect to scan-in and scan-out pins by an attacker.

Resetting the round register when the circuit switches from mission mode to test mode has also been explored as countermeasure. The scan attacks presented above indeed do not stand if the switch implies a reset of the round register, since there is no meaningful data to observe. However, the attacker can either prevent the reset operation by acting on the scan-enable signal for shifting out the data without setting up the test mode (e.g., using probing), or she/he may implement a test-mode-only attack. The attack presented in [6] for instance circumvents the reset countermeasure since it proceeds in test mode only, without requiring any switch from the mission mode. The plaintext used all along the attack is set thanks to the scan in operation. However, it is important to note that the procedure assumes that the secret key used in mission mode is also used in test mode.

Built-In Self-Test (BIST) [7] seems to be a very attractive solution since scan chains are no longer accessible for external control and observation. However, besides possible loss in terms of fault coverage, this DfT approach compromises diagnostics and debugging.

A further solution proposed in literature is the use of secure test wrappers [8] where the access to the test mechanism is protected by a locking mechanism. Only authorized users with the test session key are granted test access. In addition to the area penalty introduced by this solution, issues on manageability of test-session keys must still be solved.

The solution proposed in [9] consists in fixing the scan chain structure (FFs' ordering) only during the test mode. In mission mode, the FFs are dynamically and randomly assigned to different positions in the scan chain. This scrambling operation prevents analysis of the data observed on the scan-out pin since there is no possibility to know which data is observed at any moment. This solution provides a high level of security and is compliant with usual design flow, but the mechanism for scrambling the scan chain data seriously impacts the device area and increases power consumption in mission mode.

Other architectures have been explored for preventing scan based attacks by implementing a 'secret' combinational or sequential function within the scan chain to obfuscate its content [10]. The tester has to be aware of the specific hidden procedure and test data has to be processed before being compared to expected data. This solution is based on the assumption that the attacker has no way to get the information on the scan chain's implementation. However, such 'security-by-obscurity' approach goes against Kerckhoffs' doctrine and is not considered as strong.

A secure embedded comparator is proposed in [11] for preventing states observation. The basic idea is to compare the actual circuit test response with the expected one within the chip boundaries instead of scanning-out the response for external comparison. Test data fed to the circuit include both test vectors and expected test responses. The comparison result is not bitwise delivered to the ATE, but stored inside the chip until the last bit of the whole response vector has been compared, making thus any attack based on the observation of test responses inefficient. With this solution, there is no impact on test coverage or test time. Diagnostic can still be done thanks to a fault dictionary where test responses are stored for modeled faults. However, this procedure requires much longer processes since all pre-computed faulty responses must be uploaded in the circuit in order to be on-chip compared with the actual, but unknown, ones. Because there are potentially as many faulty responses than faults detected by the pattern under evaluation, the diagnostic process requires #detected-fault-per-pattern iterations for each unexpected response. Debug is also affected by the lack of observability.

In [12] the authors propose the use of a stream cipher in order to encrypt the content of the JTAG communication. The authors decided not to use a block cipher due to expected large area overhead and potential supplementary test time for testing the extra bloc cipher. However, they resort also to a hash function and a message authentication code in order to set-up a secure protocol to exchange the secret key used for the stream cipher.

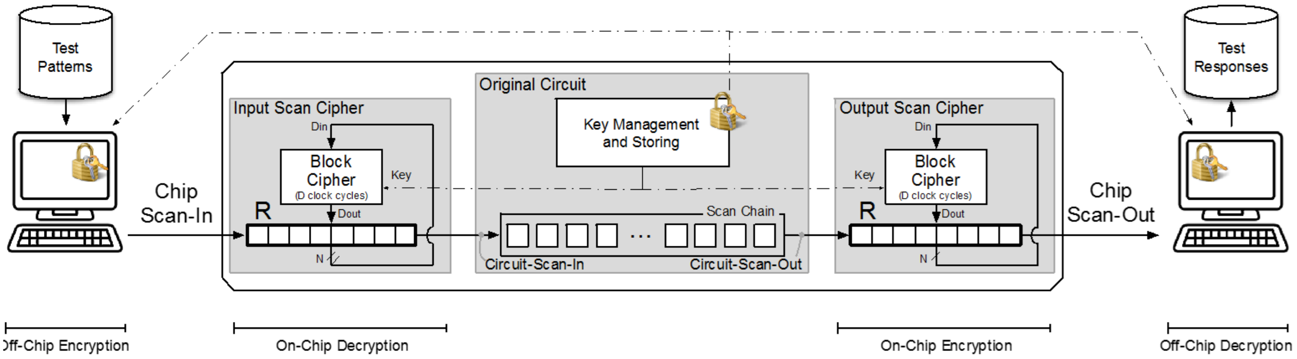

Fig. 1. Basic proposed scheme

In this paper we will show that the use of a light block cipher can reduce the additional area overhead. Moreover, we propose a scheme where the block cipher introduces a constant delay in the overall test scheme, without requiring non-standard test protocols. In addition, this extra delay may be offset by introduction of low cost observation points.

III. Proposed Solution

The proposed solution applies for integrated circuits embedding at least one crypto-core, a secret key stored in a non-volatile memory, and a secret key management policy. We also consider that the circuit implements scan-based DfT and that some FFs of the scan chains belong to the crypto-core, thus being the target of a possible scan attack. Moreover, we consider that the circuit will be used for applications where there is the need of a debugging facility implemented through the access to the scan chain. For instance, the circuit could be a microprocessor with a crypto co-processor, and the developer of the final application might need to access the content of the registers of these processors to debug the application.

The main idea is to encrypt the content of the scan chain with a block cipher algorithm. Assuming a key management policy already embedded in the secure device, the system integrator chooses to provide one initial key, or ideally distinct initial keys, for mission, debug and test modes on system's cores. Key extension circuitry must be designed according to the chosen scan cipher, i.e. PRESENT in this paper. The test data is ciphered/deciphered with the key currently known and used by the person accessing the circuit, the developer in charge of a debug procedure for instance.

The proposed test procedure (as shown in Fig. 1) consists of the following steps:

1. Generate test patterns for the circuit under test and compute expected 'fault-free' test responses;
2. Off-chip encrypt the test patterns with the chosen cryptographic algorithm and the secret key related to the current activity;
3. Scan-in an encrypted test pattern, which is first on-the-fly decrypted using the additional Input Scan Cipher, then scanned in the circuit under test;
4. On-the-fly encrypt the test responses using the additional Output Scan Cipher before shifting-out the encrypted circuit response;
5. Off-chip decrypt the encrypted test responses to obtain the actual responses of the circuit and compare with expected ones.

As shown in Fig. 1, two block-ciphers are added to the original circuit. These two ciphers have an N-bit round register (R) with two operating modes, parallel load and shift. The *Input Scan Cipher* decrypts data provided by the ATE, while the *Output Scan Cipher* encrypts the test response before transmission to ATE.

Every N clock cycles, the shift operation must be interrupted in order to allow the scan ciphers to encrypt/decrypt the N-bit data. By assuming that the encryption operation lasts D clock cycles, this solution would require D additional clock cycles for every N bits, and would result in an excessive test time overhead. To reduce this overhead, we propose the use of two registers in each scan cipher (see Fig. 2). These extra registers allow interleaving the shift operation and the encryption process. While one of the two registers (e.g., R1) is serially loaded with new data, the other one (i.e., R2) is used in the meantime to encrypt the data stored during the N previous clock cycles.

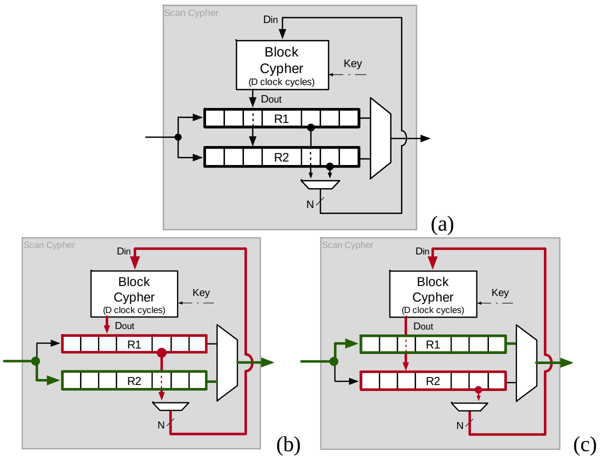

Fig. 2. Optimized Scan Cipher: (a) generic scheme; (b) R1 is used for encryption while R2 shifts test data to the device under test; (c) R2 is used for encryption while R1 is used to shift test data

Fig. 3 shows the detailed timing of the shift and ciphering operations for the input scan cipher. In this example, the bitstream feeding the scan chain is split in 4 segments of N bits each: S1, S2, S3 and S4. During the first N clock cycles, register R1 is fed by encrypted data S1. During the next N clock cycles, while S1 is decrypted (D clock cycles), the register R2 is fed by new encrypted data S2. Then, decrypted S1 is shifted into the scan chain of the device under test while new data S3 are shifted into R1, and S2 is decrypted in R2. Eventually, S3 is decrypted into R1 while last segment S4 is shifted into R2 and decrypted S2 is shifted from R2 to the circuit scan chain.

The test time is thus impacted by 2xN extra clock cycles used at the beginning of the test procedure for loading R1 and R2 registers before to feed the original scan chain. A similar procedure is executed at the output of the circuit and the same additional offset of 2xN extra clock cycles is required at the end of the test procedure in order to read-out last data stored in R1 and R2 of the Output Scan Cipher.

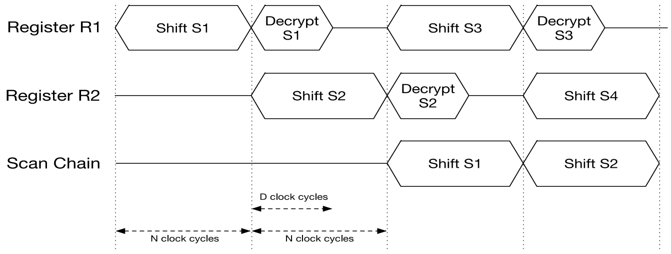

Fig. 3. Timing of scan operations

A controller is in charge of enabling the correct sequence of operations based on the value of the scan enable signal. Whenever a scan operation is required, the controller enables the two scan ciphers to prevent any clear bitstream to be inserted or observed.

Finally, managing these operations with scan chains whose length is not a multiple of N is not an issue. Following the same regular N-shifts scheme the controller can correctly feed the scan chain at the cost of additional clock cycles used to complete the shift operation on the smaller-than-N segment. Fig. 4 shows the complete time diagram of shift operations in the case of a circuit having F=*S•N+R* flip flops, where:

- F the total number of FF in the original circuit
- S the number of N-bit segments
- R = F modulo N
- I_j^k (respectively O_j^k) the jth segment (with 0≤j≤S) of N bits of the kth test pattern (respectively response) provided to (and obtained from) the circuit;
- E(x) the encrypted value of a segment x;
- *decrypt()* and *shift()* the two operations inside a block cipher.

As shown in Fig. 4, at the step where the first segment of the second pattern I_0^2 is decrypted (row 7), the scan chain is actually set to the first test pattern value: $I_x^1{}_{(\text{N bits})} \dots I_1^1{}_{(\text{N bits})}$ $I_0^1{}_{(\text{R bits})}$, while N-R first bits of the first scanned-in segment I_0^1 have been shifted in the Output Scan Cipher register. Row 12 in the time diagram shows that the N-R bits of the second pattern-first segment I_0^2, "ddd", are encrypted and shifted out as part of the last R-bits test response.

Concerning the test time overhead, we have 2xN additional shift operations at the very beginning of the test procedure in order to feed the pipeline and, in the case of R>0, we need N-R additional shift operations for each test pattern. More formally, by defining T the number of clock cycles for the original circuit to be tested without scan attack countermeasure, and K the overall number of test patterns, the number of clock cycles T_f required to test the circuit with the encryption of the scan chain is shown in equation 1:

$$T_f = \begin{cases} T + [2 \cdot 2N] & if\ R = 0 \\ T + [2 \cdot 2N + (N - R)(K + 1)] & if\ R > 0 \end{cases} \quad (1)$$

In section IV.B we propose an optimization that allows reducing the overall test time at the cost of R additional FFs.

IV. Experiments

In this section we present an implementation of the proposed solution, where the PRESENT block cipher [13] is used for test data encryption and decryption. PRESENT has been chosen because of its low cost implementation. It encrypts and decrypts data blocks of 64 bits (i.e., N=64) in 32 clock cycles (i.e., D=32).

A. Implementation

We implemented the finite state machine controlling the two scan ciphers and optimized the design by sharing common parts in both scan ciphers, i.e., the key expansion and the PRESENT control unit.

Both PRESENT ciphers and their control represent a total of 2081 combinational cells and 396 FFs (10,760 equivalent-cell area). This overhead is to compare with the original circuit under test. According to the number of extra gates/FFs for implementing the PRESENT ciphers, this secure scan chain solution is clearly dedicated to large designs. However, as we assumed in Section III, for circuits embedding a microprocessor and one (or more) cryptographic cores, this solution has a very small impact since the same PRESENT block ciphers can be used to encrypt the scan chain content of the whole original circuit.

We applied our solution to test a pipelined AES core with one scan chain composed of 7873 FFs (total area = 367,926 equivalent-cell area). The area overhead related to the proposed secure scan infrastructure is of 2.92%. According to the block size handled by the PRESENT algorithm, the 64-bit segment decomposition of the AES scan chain leads to F=7873=123x64+1. The scan chain is thus composed of S=123 segments of N=64 bits each, plus one extra segment of length R=1. This example corresponds to the worst-case scenario where the smaller-than-N segment requires the longest relative test time overhead. Indeed, N-R=63 extra clock cycles are added to the shift operation of each test pattern in order to implement the constant 64-bits shift operation on every segments. As an example, the Automatic Test Pattern Generation Tool we used provided us with a test set of K =1148 test patterns to achieve 100% of Fault Coverage. This test set results in a test time of T = (7873+1)x1,148+7873=9,047,225 clock cycles for the original AES. The final test time for the proposed architecture represents an overhead of (eq. 1): 2x2x64 + (64-1) (1148+1) = 72,643 clock cycles, i.e. only 0.8% of the original test time.

B. Optimization

The additional clock cycles that are wasted in order to synchronize the encryption scheme using constant segment-length N can be exploited to increase the testability of the original circuit.

On every pattern, N-R extra clock-cycles are used to shift N-bits data on a regular manner. By adding N-R dummy flip flops to the original scan chain, we can use these extra FFs for observing internal signals, without any impact on the test time. The insertion of these observation points increases the circuit testability and can reduce the number K of test patterns to achieve a given fault coverage.

This optimization on the pipelined AES example allows saving 29 test patterns, thus a time saving of 1.74% compared to the test time of the original circuit without protection. The proposed scheme including the two PRESENT input/output ciphers, the controller, and 63 extra scan FFs for observation represents an area overhead of 3.26% compared to the unprotected scanned AES.

This optimization is therefore attractive because with a small additional cost in terms of area (in the worst case 63 FFs are added) it can reduce even the original test time, by still protecting the scan chain against scan attacks.

C. Test of the test infrastructure

The proposed technique allows testing the original circuit without any loss in terms of test coverage. However, the test infrastructure must be tested as well. The input and output scan ciphers require a different test approach. Indeed, it is necessary to test the scan ciphers without transforming them into a classical scan design, otherwise the overall security would be jeopardized (the scan attack could be applied on these scan ciphers). Therefore, the solution is to test the scan ciphers with functional test patterns.

As previously shown in [14] and [15], the test of circuits implementing cryptographic algorithms is effective even by using random test patterns. Indeed, random data and possible errors are easily propagated through typical operations involved in such encryption algorithms thanks again to their obfuscation and diffusion properties. Experiments perform on PRESENT show that 100% fault coverage of stuck-at faults can be achieved with 1000 random patterns of N=64 bits.

Testing the original circuit provides us with patterns (and responses) processed through the PRESENT ciphers. These patterns can be seen as random patterns with respect to these PRESENT ciphers and should therefore be able to test them without requiring any additional patterns. We have validated this assertion using the S35932 benchmark. We used this circuit instead of the pipelined AES because this circuit is much smaller and it requires a lower number of test vectors to be fully tested. If, by using the vectors for the S35932 benchmark, the proposed architecture is also tested, then it will be also tested with bigger circuits (requiring more test patterns to be tested), The test sequence for the S35932 is composed of 55 patterns of 1728 bits. These data are equivalent to 1485 random patterns of 64 bits applied to the PRESENT Scan input cipher, while 1485 test responses from S35932 are applied to the Scan output cipher. As expected, while the sequence is applied in the first instance for detection of the S35932 faults, the sequence also covers 100% stuck-at faults in the proposed test infrastructure. No additional test patterns are needed to test the scan chain encryption.

D. Security

With scan chain encryption, scan-based attacks are not possible. Indeed, input decryption prevents control-based scan attacks since it is not possible to set desired values in the scan chain of the device under test without knowledge of the key. For the same reason, output encryption prevents observation-based attacks since internal states of the device under test cannot be analyzed without decryption.

If the scan-enable is disabled during an encryption (i.e., in the middle of a shift operation), all the registers of the proposed controller and scan ciphers are frozen. The FSM controlling block ciphers goes to a hold state and resume to the previous state when the scan-enable is asserted again. The encryption resumes where it lefts off and no unencrypted data is shifted-in or shifted-out. Concerning FSM vulnerabilities, no signals connected to FSM are directly accessible because its test is ensured without scan chains. Therefore, it is not possible to bypass the encryption by altering FSM control signals.

A possible attack imagined on the proposed test infrastructure could be to reset the FSM before the end of R1 filling. This reset operation would lead the FSM to start again shift operations on R1 and unencrypted data would be shifted out. For this reason, registers R1 and R2 are reset whenever the FSM is reset.

Regarding the security of PRESENT block cipher, security analysis is performed in [13]. It is proven that this block cipher is resistant to differential cryptanalysis. Concerning side-channels attacks, a resistant version of PRESENT exists in [16]. Therefore, PRESENT have a good security level and is one of the smallest block cipher.

To our knowledge, other proposed block ciphers have either a larger implementation than PRESENT or a prohibitive latency or do not guarantee a sufficient security level. Nevertheless, it is clear that if a better block cipher is proposed, it can replace PRESENT in the proposed test infrastructure.

V. Conclusions

Scan attacks exploit facilities offered by scan chains to retrieve embedded secret data. In this paper we proposed a solution based on the encryption of the scan chain content. This scheme applies to integrated circuits that embed at least one cryptographic core and thus a secret key management policy. The main idea is to cipher the content of the scan chain with a secret key developed for the current activity. In debug mode for instance, the developer must be aware of the secret key used in mission mode and thus, the same key could be used for deciphering/ciphering operations on scan chain input and output. The solution allows using the scan chains for both manufacturing test purposes as well as debugging the circuit at mission time, without the need of different test access protocols. Experimental results showed a marginal impact on both area and test time.

References

[1] Bo Yang, Kaijie Wu, and Ramesh Karri. Scan Based Side Channel Attack on Dedicated Hardware Implementations of Data Encryption Standard. In ITC, pp 339-344. IEEE, 2004.

[2] J. Daemen and V. Rijmen. The Design of Rijndael. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[3] Bo Yang, Kaijie Wu, and Ramesh Karri. Secure scan: a design-for-test architecture for crypto chips. In DAC, pp 135-140. ACM, 2005.

[4] Jean DaRolt, Giorgio Di Natale, Marie-Lise Flottes, and Bruno Rouzeyre. Scan Attacks and Countermeasures in Presence of Scan Response Compactors. In European Test Symposium, pp 19-24. IEEE Computer Society, 2011.

[5] Jean DaRolt, Giorgio Di Natale, Marie-Lise Flottes, and Bruno Rouzeyre. Are advanced DfT structures sufficient for preventing scan-attacks? In VTS, pp 246-251. IEEE, 2012.

[6] Sk Subidh Ali, Ozgur Sinanoglu, Samah Mohamed Saeed, and Ramesh Karri. New scan-based attack using only the test mode. In VLSI-SoC, pp 234-239. IEEE, 2013.

[7] Marion Doulcier, Marie-Lise Flottes, Bruno Rouzeyre. AES-based BIST: Self-test, Test Pattern Generation and Signature Analysis. In 4th IEEE International Symposium on Electronic Design, Test & Applications, Hong-Kong, IEEE, pp.314-321, 2008.

[8] Chiu G.-M.; Li J.; C.-M. A Secure Test Wrapper Design Against Internal and Boundary Scan Attacks for Embedded Cores, IEEE Trans. on Very Large Scale Integration (VLSI) System, vol 20, issue 1, p 126-134, 2010

[9] D. Hely, M.-L. Flottes, F. Bancel, B. Rouzeyere, N. Bernard, Scan design and secure chip. In Proc. IEEE Int. On-Line Test. Symp., pp. 219-224, 2004.

[10] Hideo Fujiwara and Katsuya Fujiwara, "Strongly Secure Scan Design Using Generalized Feed Forward Shift Registers," IEICE Trans. on Inf. and Syst., Vol. E98-D, No. 10, pp. 1852-1855, Oct. 2015

[11] Da Rolt J.; Di Natale G.; Flottes M.-L.; Rouzeyre B. Thwarting Scan-Based Attacks on Secure-ICs With On-Chip Comparison. In Proc. IEEE Trans. on Very Large Scale Integration (VLSI) System, no. 22 pp. 947-951, 2013.

[12] Kurt Rosenfeld, Ramesh Karri, "Attacks and Defenses for JTAG", IEEE Design & Test of Computers, vol.27, no. 1, pp. 36-47, January/February 2010, doi:10.1109/MDT.2010.9

[13] PRESENT: An Ultra-Lightweight Block Cipher, A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe, P. Paillier and I. Verbauwhede (Eds.): CHES 2007, LNCS 4727, pp. 450–466, Springer-Verlag Berlin Heidelberg 2007

[14] G. Di Natale, M. Doulcier, M. L. Flottes, B. Rouzeyre, "Self-Test Techniques for Crypto-Devices", IEEE Transaction on VLSI Systems, pp. 1-5, February 2010, Volume:18, Issue: 2, DOI: 10.1109/TVLSI.2008.2010045

[15] A. Schubert and W. Anheier, "On random pattern testability of cryptographic VLSI cores," J. Electron. Test.: Theory Appl., vol. 16, no. 3, pp. 185–192, Jun. 2000.

[16] A. Poschmann, A. Moradi, K. Khoo, C. Lim, H. Wang, S. Ling, "Side-Channel Resistant Crypto for less than 2,300 GE", J. Cryptology, vol. 24, pp 322-345, 2011

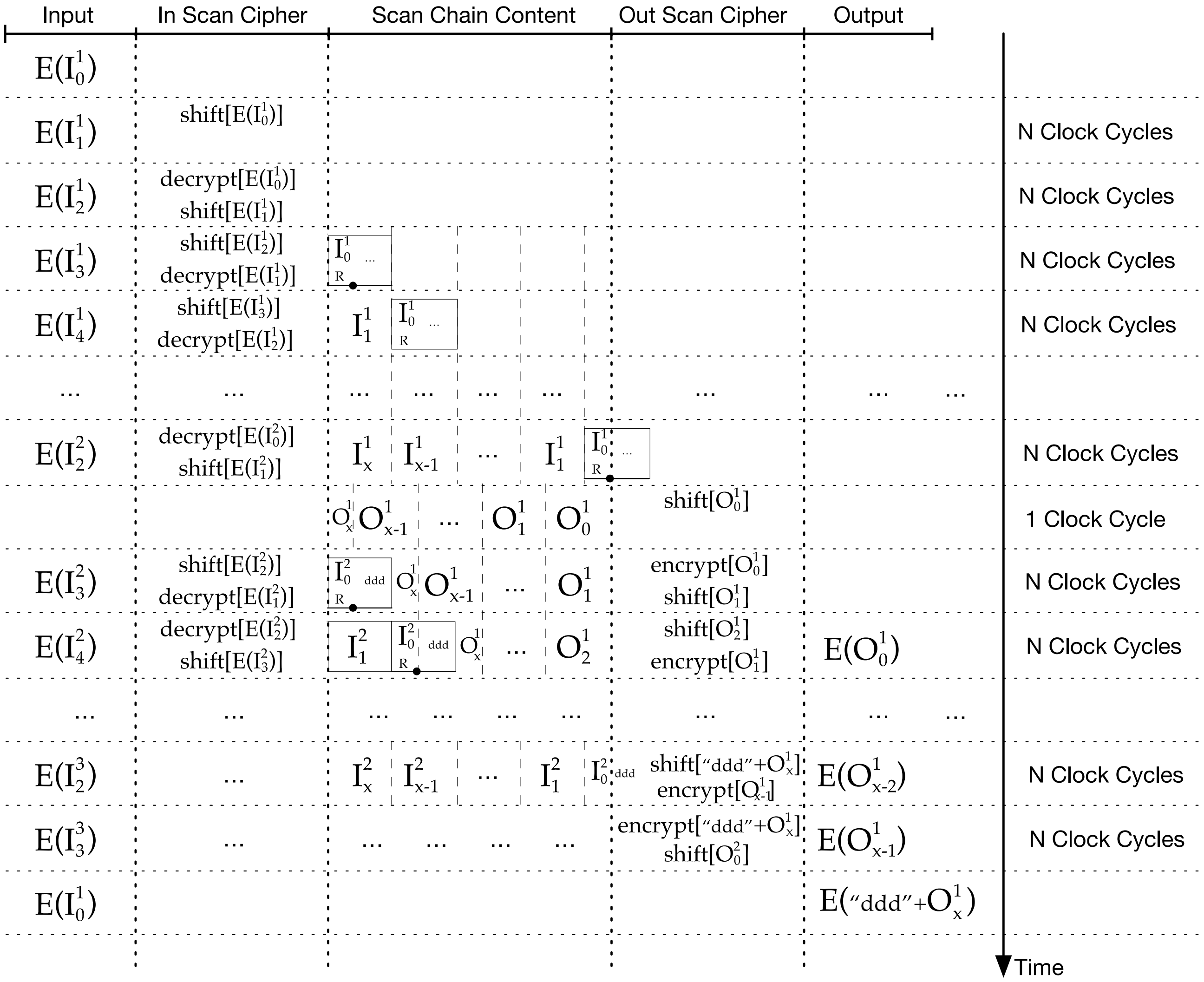

Fig. 4. Time Diagram of shift operations