Encryption of test data: which cipher is better?

Mathieu da Silva, Emanuele Valea, Marie-Lise Flottes, Sophie Dupuis, Giorgio Di Natale, Bruno Rouzeyre

Encryption of test data: which cipher is better?

Mathieu Da Silva, Emanuele Valea, Marie-Lise Flottes, Sophie Dupuis, Giorgio Di Natale, Bruno Rouzeyre
LIRMM (Université Montpellier - CNRS), Montpellier, France
{mathieu.da-silva,emanuele.valea,flottes,dupuis,dinatale,rouzeyre}@lirmm.fr

***Abstract*—Testing is a mandatory step in the Integrated Circuit (IC) production because it ensures the required quality of the devices. The most common solution for easing IC testing is the scan chain insertion. This way, a tester can control and observe the internal states of the circuit through dedicated pins. However, a malicious user can exploit this infrastructure in order to extract secret information stored inside the chip. This is the case for cryptographic circuits where partially encrypted results can be observed by shifting out the scan chain content and exploited to retrieve secret keys. Existing countermeasures consist in encrypting the scan content, ensuring the confidentiality of the exchanged messages between the circuit and the tester. The encryption techniques that have been proposed so far rely on the use of two different ciphers: stream ciphers and block ciphers. In this paper, we present pros and cons of both solutions in terms of security and performance. The purpose is to provide an overview of the state-of-the-art in test data encryption and to give elements of comparison between the two ciphers.**

Keywords—Test and Security; Test data encryption; Block cipher; Stream cipher

I. INTRODUCTION

Steady advances in the semiconductor technology have resulted in devices with hundreds of millions of transistors. The consequence is an increasing probability of physical defects in manufactured Integrated Circuits (ICs), each possibly leading to the failure of the system. Typical defects are shorts or opens involuntarily created during IC manufacturing. All ICs are thus tested after production in order to sort out faulty devices. The circuits that pass the manufacturing test are then packaged. A second test is performed on the packaged devices to eliminate those that may have been damaged during the packaging process or assembled into faulty packages. Finally, other tests are performed after the assembly of ICs on boards. They are used to ensure the final quality of the IC before going to market. Testing is therefore an important step along the IC production, representing half of the cost of the final product. Test costs include development costs (test sequence computation), implementation costs (design practices for high testability) and application costs (time needed to test every single IC), the latter representing a recurrent cost since every single IC must be tested before shipping.

Design-for-Testability (DfT) is a domain of paramount importance. Its goal is to maximize the capability in detecting faults at test time, possibly to perform diagnosis, while minimizing the test time and the required number of additional pins. The most popular DfT technique for dealing with sequential circuits is the scan chain insertion. It consists in replacing the Flip-Flops (FFs) of the IC by Scan Flip-Flops (SFFs). These SFFs are serially connected to form one or several shift-registers, the so-called scan chain(s). In mission mode, the SFFs behave as regular FFs while in test mode they can be serially written or read through scan-in and scan-out pins. Doing so, a tester can control the internal states of the IC by shifting-in test vectors into the scan chain(s), and it can observe internal states stored into the scan chain(s) by shifting-out test responses. An Automatic Test Pattern Generator (ATPG) is used to produce test vectors depending on the target fault models, expected fault-free responses are computed as well. The test procedure consists in serially shifting the test vectors inside the scan chain, and collecting the corresponding responses. The tester compares then the actual test responses with the expected ones in order to identify the presence of faults within the circuit.

Note that the scan-based test procedure introduces numerous cycles for each test data since test patterns and test responses must be serially propagated in the scan chain(s). Fortunately, while a test response is shifted out for observation, a new test vector can be concurrently scanned-in the same scan chain. Test time is thus financially affordable thanks to the simultaneous scan-in and scan out operations.

Unfortunately, scan chains jeopardize the security of the data processed by the IC. The observability feature provided by the test infrastructures can indeed be a source of information leakage, useful to retrieve secret keys of devices implementing cryptographic primitives, such as the Advanced Encryption Standard (AES) [1]. The target of the attack in that case is the AES round register that store partial encrypted results. The scan attack consists in scanning out the content of this round register after the execution of the first AES round. It has been shown in [2] that the scan attack to retrieve the whole 128-bit secret key of an AES crypto-processor can be completed applying on average 512 plaintexts.

Attacks performed on the scan chains are called “scan-based attacks” [2]–[5]. These hardware attacks do not require any invasive handling nor sophisticated equipment. On the other hand, a countermeasure consisting in disconnecting test IOs after manufacturing is not entirely satisfactory as it restrains debug and diagnostic during the IC life cycle.

Beside the security threat involving the scan chains, standard test interfaces can also be maliciously exploited. The IEEE 1149.1 [6] standard, named JTAG, was originally used for testing printed circuit boards. With the increasing number of cores implemented inside System-on-Chips (SoCs), the IEEE 1500 [7] standard was proposed to facilitate their testing. Nowadays, complex ICs integrate a great variety of instruments to ease test and diagnosis. Interfacing this large number of embedded instruments with the user is a challenge that has been addressed by introducing Reconfigurable Scan

Networks (RSNs). The RSNs have been standardized in the IEEE Std. 1687 [8], named also IJTAG. They provide a flexible and scalable access to the instruments. These test infrastructures usually allow the access to the scan chains after IC packaging, since the dedicated pads on the die are only accessible during manufacturing test. Moreover, standardized infrastructures are not strictly limited to test purposes, they also allow the access to the circuit for debugging. An attacker can use these structures to steal the contents of on-chip memories or to modify the firmware. The test infrastructure usually requires also the connection of the devices to a network, organized in a daisy chain structure. This represents another threat inside the chip. Indeed, if a malicious device is connected to the test daisy chain, this device can read and/or modify the test data shifted through it, in order to steal confidential data, or to force the device into an illegal behavior. In order to prevent misuse of these test infrastructures, several countermeasures have been proposed in the literature, protecting the access to the scan chains as well as the debugging features.

One of these countermeasures consists in the encryption of test data shifted to and from the test interface. An authorized user encrypts the test vectors off-chip using the established secret key. The encrypted test vectors are shifted in the circuit through the test interface. On-chip decryption is performed before the test vectors are applied to the circuit. The resulting test responses are then encrypted before scanning them out of the circuit. The authorized tester collects the encrypted test responses, and decrypts these data in order to compare them with the expected ones. This solution has the advantage to preserve testing and debugging facilities, while preventing malicious users from accessing the test infrastructure. Since the test communication is encrypted, a user with no knowledge of the secret key is not able to set the circuit in an undesired state, nor to read its internal states. Another advantage of the test data encryption is to not affect the fault coverage achieved with classical scan design, since the same test data are applied and collected once the encryption/decryption is processed.

The test communication encryption is performed by a symmetric cipher. A cipher transforms a plain message into a ciphered version using a secret key. In the same manner, the inverse transformation is performed in order to retrieve the plaintext from the ciphertext, by using the same secret key.

Two ciphers can be used in the test infrastructures, the stream cipher and the block cipher. The stream cipher performs a bit-to-bit encryption of a serial bitstream, while the block cipher encrypts an n-bit block of plaintext into a ciphertext block of n bits.

In this paper, we give an overview of the existing countermeasures based on test data encryption. We also compare the solutions based on stream ciphers with the ones based on block ciphers.

The remainder of this paper is organized as follows. In Section II we provide a background on block and stream ciphers, as well as the state-of-the-art on the encryption for securing the test infrastructures. In Section III we compare the test data encryption based on stream cipher with the one based on block cipher. Section IV finally draws some conclusions.

II. Background

We give in this Section a brief overview on both stream and block ciphers. We introduce then the existing countermeasures based on these ciphers, in order to compare them in the next Section.

A. Ciphers

A cipher ensures the confidentiality of a communication, executing an encryption function E on a message m, to produce a ciphertext c using a secret key k, such that $E(m,k) = c$. Only a receiver knowing the key k can properly apply the inverse function D to retrieve the original message, i.e. $D(c,k) = m$.

1) Stream ciphers

The stream cipher performs a bitwise XOR operation between the plaintext and a pseudo-random bitstream, called keystream, generated from a seed. In some stream ciphers (e.g. TRIVIUM [10] cipher), the seed is composed of the key k and the initial value IV. The key needs to be secret, while the IV can be public, but it is supposed to be different for each encryption session. The generated keystream is denoted as $S(k, IV)$. The encryption and decryption functions (E, D) of the stream cipher are thus respectively defined as $E(m,k) = m \oplus S(k, IV) = c$, and $D(c,k) = c \oplus S(k, IV) = m$.

The first requirement that must be fulfilled in order to consider a stream cipher secure is to produce an unpredictable keystream. This way, it is impossible to retrieve the plaintext from the ciphertext without knowing the keystream. The second requirement is to never use the same keystream more than once. In the case where two different plaintexts m_1 and m_2 are encrypted with the same keystream $S(k, IV)$, an attacker can exploit the XOR of the two respective ciphered messages c_1 and c_2. Indeed, this operation leads to remove the encryption: $c_1 \oplus c_2 = (S(k, IV) \oplus m_1) \oplus (S(k, IV) \oplus m_2) = m_1 \oplus m_2$. The XOR between two messages can then be exploited in a differential attack, such as it is the case for scan attacks [1-4]. That's why it is important to use a different seed, i.e. a different IV and/or secret key, to initialize the stream cipher between different encryption sessions.

2) Block ciphers

The block cipher executes iterative transformations based on substitutions and permutations on fixed-length groups of bits, called blocks. The transformation function depends on a secret key. The block encryption results in the diffusion and confusion of the plaintext on the ciphertext at each iteration of the execution. The iterations performed by the block cipher are called rounds.

The most used block cipher is defined by a standard, named AES [1]. Nevertheless, AES may induce a large area overhead to the device under test (see Tab. 1). For this reason, lightweight block ciphers implying a lower area cost have been studied, such as PRESENT [9]. This block cipher

guarantees a lower security level than AES, but cryptoanalysis studies show that it is enough for most applications. The encryption is performed on block size of 64 bits in 31 rounds with two possible lengths for the secret key, 80 bits or 128 bits.

B. Test data encryption

Several countermeasures to the scan attacks have been reported in the literature [10]–[15] to ensure the confidentiality of the exchanged test data between the tester and the circuit, while preserving the use of the test interface for authorized users. These solutions are applied to the existing test infrastructure by inserting two ciphers in the circuit. One is placed at the serial input of the test interface in order to decrypt the encrypted test data sent by the user. The other is placed at the serial output in order to encrypt the test responses before being shifted out of the circuit.

The decryption performed at the scan-in of the test interface takes the controllability of the circuit away from an unauthorized user, who is unable to apply chosen data. The encryption performed at the scan-out of the test interface prevents him/her from observing the plain scan content. An attacker is thus not able to perform scan attacks, nor illegally debugging the circuit. The confidentiality established between the protected circuit and the tester ensures also a protection from threats placed inside the chip, such as malicious devices connected to the test daisy chain. The encryption prevents these devices from making sense of the sniffed encrypted data, or from modifying them in a controlled way.

The encryption of test data proposed in the literature is based on stream ciphers as well as on block ciphers. The stream cipher used to encrypt the test communication is the TRIVIUM [10], because of its low silicon footprint. The pseudo-random sequence, used as keystream, is generated with a Non-Linear Feedback Shift Register (NLFSR) from an 80-bit secret key and an 80-bit IV. The TRIVIUM stream cipher encrypts the test interfaces of JTAG in [10], IEEE 1500 in [12] and IJTAG in [13], while the PRESENT block cipher is used to encrypt the scan chain in [14][15]. We will see the pros and cons of each encryption method in the next section.

III. COMPARISON: STREAM VS BLOCK ENCRYPTION

We evaluate the stream-based countermeasures and the block-based ones according to several cost functions: the area and power overheads, the impact on the testing cost and the provided security level.

A. Area and power consumption overhead

Tab. 1 shows the area and power consumption for the AES [1], the PRESENT [9] block cipher with 128-bit secret key, and for the TRIVIUM [10] stream cipher.

Stream ciphering is the technique that has been preferred so far in the literature [10]–[13]. The choice of the stream cipher is motivated by the lower impact on area and power costs. Block ciphers imply the usage of a larger area footprint than stream ciphers, as is the case for the AES block cipher. However, some modified versions of the AES have been designed to be lightweight, such as PRESENT. PRESENT block cipher and TRIVIUM stream cipher have similar costs in terms of area and power consumption, as shown in Tab. 1.

Ciphers	Area (Gate Equivalent)	Power consumption @ 10 MHz (µW)
Block ciphers		
AES-128	22 535	134.2
PRESENT-128	2 139	26.26
Stream cipher		
TRIVIUM	2 016	36.35

Tab. 1 Area and power consumption for block and stream ciphers

A more realistic estimation of area and power costs takes into account the number of ciphers that have to be implemented. In fact, depending on whether the encryption is performed with block or stream ciphers, a different number of ciphers must be placed inside the circuit. The block-based solution [14][15] requires two ciphers, one dedicated to the decryption of the test patterns, the other dedicated to the encryption of the test responses (both test vector in and test response out are concurrent operations).

Conversely, the stream-based solutions [10]–[13] can use only one stream cipher to generate both the decryption and encryption keystream. Therefore, if a lightweight block cipher is used, which has a cost comparable to the stream cipher, the block-based solutions implies twice more area and power overhead, due to the duplication.

B. Testing cost

The impact on test coverage is also important to evaluate the countermeasures applied to the test infrastructure. The set of faults that are detected by the test sequences, originally generated by the ATPG, must not decrease because of the insertion of the security countermeasure. The encryption of test data assures this condition. The content of the applied test vectors and the produced responses is not disrupted by the additional encryption/decryption steps. Concerning the test of the ciphers themselves, authors in [14][15] showed that the extra logic introduced for encryption is tested in the same time as the test data are processed. Therefore, there is no impact on the original test coverage even with the implementation of the ciphers.

Nevertheless, the decryption/encryption of the test data shifted through the test interface adds a cost in terms of test time. It is important to increase as less as possible the test time, since this overhead has an impact on the cost of each sample of the circuit. If this is not taken into account, the time to test an entire product chain can increase significantly.

Concerning the stream cipher, an additional initialization time is required. This overhead is 1152 clock cycles for the TRIVIUM, representing a marginal cost compared to the millions of clock cycles needed to test an entire SoC. Moreover, since both the testing interface and the stream cipher have a serial access, no additional timing overhead is required.

Contrariwise, the parallel interface of the block cipher requires padding the test data acquired serially into a multiple of the block size. The padding of test data results in additional clock cycles to complete the shifting operations, implying a test time overhead on each pattern. This results in higher overhead than the stream-based solutions. However, an optimization is proposed in [14][15], based on an alternative DfT approach that makes the scan chain length multiple of the block size. Block ciphers have thus to be adapted to cope with the serial interface of the testing infrastructures.

C. Security

As shown in Section II.B, the state-of-the-art countermeasures based on the test communication encryption protect against the aforementioned threats. However, the stream-based solutions [10]–[13] show a vulnerability due to the mismanagement of the seed generating the keystream. In this case, the attacker has the possibility to provoke the generation of the same keystream to encrypt different test data. The requirement on the use of the stream cipher, stated in Section II.A.1), is therefore not respected, circumventing the encryption in the case of differential scan attacks [1-4].

This security flaw is not present on the block-based countermeasures, representing therefore a more secure encryption solution than the stream-based ones. Tab. 2 resumes the pros and cons of both solutions.

Test data encryption	Stream cipher	Block cipher
Security	-	+
Area	+	-
Power	+	-
Test time	+	-

Tab. 2 Comparison overview

The security of the stream-based countermeasures can be improved, making sure that the stream cipher does not generate the same keystream for multiple encryptions. To produce a different keystream, the stream cipher has to change its seed between each cipher initialization, i.e. the secret key and/or the initial value *IV*. A possible solution is to generate a random *IV* at each circuit reset. As a result, the stream cipher is initialized with a different seed between each encryption. The differential scan attacks [2]–[5] are thus no longer feasible. However, the issue with this solution is to share the random *IV* to the authorized users in order to perform the encryption/decryption of the test data.

IV. Conclusion

Granting access to the internal states of the ICs is fundamental for testing during production, as well as for debugging and diagnosis in the field. Test infrastructures composed of scan networks meet these needs, but compromise the security at the same time. To prevent attacks that exploit the scan side-channel, several countermeasures exist. Some of them are based on the encryption of the test data. Two types of ciphers can be used to encrypt the test communication: the stream cipher and the block cipher. From our study, it comes out that stream ciphers can be preferred due to their smaller overhead and their easy adaptation to the serial test interface. Nevertheless, as implemented in [10]–[13], the stream-based solutions present a vulnerability, due to a misuse of the stream cipher, while the block-based solutions prove to be secure in all cases.

Acknowledgment

This project has been funded by the French Government (BPI-OSEO) under grant FUI#20 TEEVA (Trusted Execution EVAluation).

References

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

[2] B. Yang, K. Wu and R. Karri. “Secure scan: a design-for-test architecture for crypto chips”. In Design Automation Conference (DAC), pp. 135-140, 2005.

[3] 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), pp. 19-24, 2011.

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

[5] Sk Subidh Ali, Ozgur Sinanoglu, Samah Mohamed Saeed, and Ramesh Karri. “New scan-based attack using only the test mode”. In International Conference on Very Large Scale Integration (VLSI-SoC), pp. 234-239, 2013.

[6] Committee, I. S. (1990). IEEE Standard Test Access Port and Boundary-Scan Architecture. IEEE Std (Vol. 2001).

[7] IEEE Standard Testability Method for Embedded Core-based Integrated Circuits. (2012). IEEE Std 1500-2005.

[8] The IEEE Standards Association. (2014). IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device.

[9] 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. PRESENT: An Ultra-Lightweight Block Cipher. CHES 2007, LNCS 4727, pp. 450–466, Springer-Verlag Berlin Heidelberg 2007

[10] C. De Canniere and B. Preneel. “TRIVIUM Specifications”. ECRYPT Stream Cipher Project, Report, 30, 2005.

[11] K. Rosenfeld and R. Karri. . “Attacks and defenses for JTAG”. In IEEE Design and Test of Computers, 27(1), 36–47, 2010.

[12] K. Rosenfeld and R. Karri “Security-aware SoC test access mechanisms”. In IEEE VLSI Test Symposium (VTS), pp. 100–104, 2011.

[13] S. Kan, J. Dworak and J. G. Dunham. “Echeloned IJTAG data protection”. In IEEE Asian Hardware Oriented Security and Trust Symposium (AsianHOST), 2016.

[14] M. Da Silva, M.-L. Flottes, G. Di Natale, B. Rouzeyre, P. Prinetto and M. Restifo. “Scan chain encryption for the test, diagnosis and debug of secure circuits”. In IEEE European Test Symposium (ETS) pp. 1–6, 2017.

[15] M. Da Silva, M.-L. Flottes, G. Di Natale and B. Rouzeyre. “Experimentations on scan chain encryption with PRESENT”. In International Verification and Security Workshop (IVSW), pp. 45–50, 2017