Encryption Techniques for Test Infrastructures

Emanuele Valea, Marie-Lise Flottes, Giorgio Di Natale, Bruno Rouzeyre

Encryption Techniques for Test Infrastructures

Emanuele Valea[1], Marie-Lise Flottes[1], Giorgio Di Natale[2], Bruno Rouzeyre[1]
[1]*LIRMM (Université de Montpellier - CNRS)*, Montpellier, France
[2]*TIMA (Univ. Grenoble Alpes - CNRS - Grenoble INP*)*, Grenoble, France

***Abstract*—Test infrastructures are widely deployed in modern Systems-on-Chip (SoC). They allow the tester to control and observe the internal state of the SoC. However, a malicious user can exploit these infrastructures in order to extract secret information stored inside the SoC. In this paper, we review and compare security countermeasures based on the encryption of test data. The present techniques ensure the confidentiality of the exchanged messages between the circuit and the tester. Moreover, they provide lightweight user authentication in order to prevent malicious users from accessing the test infrastructure.**

***Index Terms*—Test and Security; Test data encryption; Block cipher; Stream cipher**

I. Introduction

The increase in complexity of modern Systems-on-Chip (SoC) led to the development of standard test infrastructures. They allow all the actors along the life cycle of the product to perform testing, device programming, debug and diagnostics. The successive releases of the IEEE Std. 1149.1 (JTAG), the IEEE Std. 1500 and the IEEE Std. 1687 (IJTAG) stimulated the design of hierarchical test infrastructures. The underlying principle behind these test infrastructures is the Reconfigurable Scan Network (RSN). RSN is a configurable-length scan chain, which can be accessed through the TDI and TDO pins of the SoC. The external user can access, via proper configuration of the RSN, a large number of SoC's internal resources. The test infrastructure provides a serial communication in which data are shifted through all the resources that are connected to the RSN. Many different types of resources can be connected to the test infrastructure: (i) embedded instruments for SoC configuration and monitoring; (ii) register-based BIST interfaces; (iii) debug ports. In addition, the IEEE 1500 standard offers third-party designers the possibility to wrap a standard test interface around their own IP cores. Hence, these can be easily interfaced with the SoC RSN. For this reason, internal resources of IP cores, such as internal scan chains, find themselves exposed to the external world.

Because of these characteristics, test infrastructures can be maliciously used for jeopardizing the security of the whole SoC. Access to internal resources of specific IP cores, poses a serious security threat. For instance, secret information can be obtained by an external unauthorized user. In addition, malicious IP cores can exploit their serial connection with other IP cores connected to the same scan network. This can result in sniffing and/or tampering with sensitive data [1]. For this reason, it is important to provide confidentiality of test data and user authentication capabilities at both SoC level and core level.

Our work aims to thwart the possibility of exploiting the test infrastructures for malicious purposes. We propose the encryption of test data as a powerful and lightweight countermeasure, whose purpose is twofold. In the first place, test data encryption makes them unreadable by unauthorized users and malicious IP cores. In the second place, the controllability of the test infrastructures is drastically reduced. This prevents both unauthorized users and malicious IP cores from inserting corrupted data and tampering with the encrypted data flow.

In the following sections, we present the encryption techniques that we have developed for test infrastructures. They are based on two different cryptographic primitives: the stream cipher and the block cipher. Finally, we propose a comparison between them.

II. Test Data Encryption

Both block and stream ciphers are symmetric ciphers. They rely on a shared secret key between the sender and the receiver in order to provide confidentiality to their communication. In test infrastructures the communication takes place between the external user and a target SoC (or an IP core integrated inside the SoC). Test data encryption can be applied at both SoC level (i.e. protecting the target SoC only from external threats) and IP core level (i.e. protecting a target IP core from threats also coming from untrusted third-parties). For the sake of readability, we refer the target SoC or IP core with the generic term of *target device*.

If test data encryption is implemented on a target device, the user must know the secret key, which the device keeps stored inside a secure memory. The communication starts with the user encrpyting test data using the secret key. Encrpyted data are shifted thorugh the TDI pin and decrypted inside the device by a specific hardware module. The plaintext message is exclusively present inside the target device. All upstream entities receive encrpyted data, which is unintelligible. The responses generated by the target device, are encrypted before being shifted out the TDO pin. All downstream entities receive encrpyted responses and only the authorized user is able to decrypt them.

A. Block Cipher Encryption

A block cipher encrypts n-bit blocks from a plaintext message and generates n-bit blocks of the corresponding ciphertext. The encryption process takes a fixed number of clock cycles. The same key is used for all the encryptions

*Institute of Engineering Univ. Grenoble Alpes

performed along the life cycle of the device. Lightweight block ciphers are preferred for the encryption of test infrastructures, due to their reduced implementation cost. An example of lightweight block cipher that has been used for test data encrpytion is the PRESENT block cipher.

B. Stream Cipher Encryption

A stream cipher performs a bit-wise XOR operation between the plaintext and a pseudo-random bit stream, called *keystream*. The keystream is generated by a pseudo-random generator, which is the core of the stream cipher. The TRIVIUM stream cipher is used for test data encryption, due to its lightweight hardware implementation. In the TRIVIUM stream cipher the keystream generation is initialized by an 80-bit secret key and an 80-bit Initialization Vector (IV). While the key must be secret and it never changes, the IV is a never-repeating value that is publicly known. The first requirement that must be fulfilled in order to consider a stream cipher secure is the generation of 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 order to avoid *two-times pad attacks*.

III. Comparison

We evaluate the stream-based and the block-based test data encryption techniques according to divers cost functions: area and power consumption overhead, impact on test time and security level.

A. Area and power consumption

Tab. I shows the area and power consumption of the PRESENT block cipher with 128-bit secret key and the TRIVIUM stream cipher.

TABLE I
Area and Power Consumption

Ciphers	Area (Gate Equivalent)	Power Consumption @10MHz (μW)
PRESENT-128	2139	26.26
TRIVIUM	2016	36.35

The PRESENT block cipher and the TRIVIUM stream cipher have similar costs in terms of area and power consumption. However, a more realistic estimation is performed taking into account the number of ciphers that have to be implemented in the target device. In fact, the block-based solution requires two ciphers: one for the decryption of input data; the other for the encryption of the responses. Conversely, the stream-based solution requires only one stream cipher to generate both decryption and encryption keystreams. Therefore, the block-based solution implies twice the area and power overhead.

B. Test Time

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 test infrastructure and the stream cipher have a serial interface, no additional timing overhead is required. On the other hand, the parallel interface of the block cipher requires padding test data into a multiple of the block size. The padding of test data results in additional clock cycles needed to complete the shift operations, implying a test time overhead on each pattern. This results in higher overhead than stream-based solutions. However, the application of alternative DfT approaches can help making the scan chain length multiple of the block size [2]. Therefore, block ciphers have to be adapted, in order to cope with the serial interface of the testing infrastructures.

C. Security Level

Stream-based solutions can be vulnerable to *two-times pad attacks* if the keystream generation is not properly managed. In this case, the attacker has the possibility to force the generation of the same keystream to encrypt more than one plaintext. This security flaw is not present in block-based solutions, thus representing a more secure encryption solution than the stream-based one.

Tab. II resumes pros and cons of both solutions.

TABLE II
Comparison Overview

Cost Function	Stream Cipher	Block Cipher
Area	+	-
Power	+	-
Test time	+	-
Security	-	+

IV. Conclusions

Providing the confidentiality of data involved in test infrastructures is crucial in order to grant the security of modern SoCs. The encryption of test infrastructures is a promising security technique that provides confidentiality of test data and lightweight user authentication at the cost of acceptable overhead on area, power consumption and test time degradation. The proposed comparison between stream- and block-based solutions shows that both techniques can be taken into account by designers, according to the required security level and the amount of resources that are available on the target SoC.

References

[1] E. Valea, M. Da Silva, G. Di Natale, M. Flottes, and B. Rouzeyre, "A survey on security threats and countermeasures in IEEE test standards," *IEEE Design & Test*, pp. 1–1, 2019.

[2] E. Valea, M. D. Silva, M.-L. Flottes, G. D. Natale, and B. Rouzeyre, "Stream vs block ciphers for scan encryption," *Microelectronics Journal*, vol. 86, pp. 65 – 76, 2019.