A Comprehensive Approach to a Trusted Test Infrastructure

Marc Merandat, Vincent Reynaud, Emanuele Valea, Jerome Quevremont, Nicolas Valette, Paolo Maistri, Régis Leveugle, Marie-Lise Flottes, Sophie Dupuis, Bruno Rouzeyre, et al.

To cite this version:

Marc Merandat, Vincent Reynaud, Emanuele Valea, Jerome Quevremont, Nicolas Valette, et al.. A Comprehensive Approach to a Trusted Test Infrastructure. IVSW 2019 - 4th IEEE International Verification and Security Workshop, Jul 2019, Rhodes, Greece. pp.43-48, 10.1109/IVSW.2019.8854428. lirmm-02306980

HAL Id: lirmm-02306980
https://hal-lirmm.ccsd.cnrs.fr/lirmm-02306980v1

Submitted on 7 Oct 2019

A Comprehensive Approach to a Trusted Test Infrastructure

Marc Merandat[1,4], Vincent Reynaud[2], Emanuele Valea[3], Jerome Quevremont[4], Nicolas Valette[1,4], Paolo Maistri[2], Regis Leveugle[2], Marie-Lise Flottes[3], Sophie Dupuis[3], Bruno Rouzeyre[3], Giorgio Di Natale[2]
[1]INVIA, Meyreuil, France
[2]Univ. Grenoble Alpes, CNRS, Grenoble INP*, TIMA, Grenoble, France
[3]LIRMM (Université de Montpellier – CNRS), Montpellier, France
[4]Thales, Gennevilliers, France

***Abstract*—The testability of electronic devices is of critical importance and it is often supported by IEEE standards. The available methods, on the other hand, can be an entry point to a malicious attacker, if no proper countermeasure is adopted. In this paper, we report the latest results from the HADES project, presenting a portfolio of solution towards a secure test infrastructure.**

Keywords—Test, Scan chains, IEEE 1687, BIST, Authentication, Scan chain encryption.

I. Introduction

Increased safety, reliability and cost-control for Internet of Things (IoT) devices are of critical importance nowadays. Vast numbers of devices are becoming connected via the IoT, from smart phones and smart home systems to safety-critical systems in airplanes and road vehicles. This increasing complexity calls for a new generation of miniature electronics test instruments that can be built into devices and systems to keep them operating safely, dependably and with optimal performance.

There is a great need for a hierarchy-aware smart and secure embedded test infrastructure for dependability and performance enhancement of integrated systems. In order to achieve this goal, we have to move forward from the classic design and post-silicon fabrication test approach to a new, efficient, scalable and low-cost on-line paradigm. Several markets can be potentially impacted: i) machine to machine and connected systems, ii) remote-controlled systems, iii) smart home and mobile phone, iv) safety-critical systems – typically found in the automotive and avionics domains, v) mission-critical systems -such as in space and security applications. In the coming years European Semiconductor companies will bring many new applications to the market to improve the way of living. Examples are linked to road safety, personal health care, secured wireless communications, and lighting and consumer electronics. These applications concern very complex semiconductor systems with highly integrated technologies where digital, memories and analogue are funneled in one piece of silicon.

In these kinds of applications, reliability and trustability is a key factor which cannot be guaranteed without extensive test solutions. It is therefore of utter importance to address these issues, from two complementary points of view:

1. Security in test: Access to embedded test instruments and infrastructure can lead to security flaws in several ways, including malicious control of chip features, firmware stealing or modification, and unauthorized sensitive information gathering. Avoiding these threats requires some level of encryption, both for authentication and for keeping sensitive data secret. It is important to develop secure test solutions to guarantee circuit integrity against test-based attacks while providing enough diagnosis capability to perform defect analysis, as only scan-based external testing is able to. At the same time, implementation cost and power consumption are strong constraints, especially in the context of IoT objects.
2. Reliable test of secure ICs: Industrial secure ICs contain many countermeasures to detect attack attempts and react to safeguard the security of the application, for instance by erasing sensitive key store and therefore ceasing to provide the expected service. Minor defects or out-of-range conditions embedded in such ICs can trigger false alarms that can lead to unexpected denials of service which can have serious consequences for critical applications. BIST solutions for both digital and analogue modules can provide appropriate health checks and are expected to be secure solutions as they require fewer external interfaces and therefore reduce risks of intrusion into ICs.

These aspects are one of the main focuses of the HADES project, a RD&I project consortium involving 14 partners from France and The Netherlands. Table 1 lists the requirements for secure testing identified within the consortium in order to prevent steal of end-user confidential data stored inside IC.

This paper is an initial answer to those requirements and presents the latest advances in the project related to security. In the next section, industrial solutions for sensors contribution and secure testability with reliability of IC are presented. In Section III, we detail an authentication framework to access the test infrastructure, compatible with the existing standards. Section IV will cover test data encryption, including comparative results.

Table 1. HADES requirements for secure testing

Property	Requirement	Comments
Secure access	Mutual authentication protocol	Possibly Challenge-response
Confidentiality	Symmetric encryption	1 bit per cycle (avg) ; Random IV against replay attacks
Integrity	Hashing algorithm	Collision resistant, possibly shared with other processes
Interfaces	Standard compliance	Wrappers if needed (AMBA, I2C, CAN, …)
Key management	Updatable Multiple keys	No fixed single global key Group management
Attacker model	Skilled attacker	Internal or external threats; complete access to device

* Institute of Engineering Univ. Grenoble Alpes

II. Industrial Solution

At the level of an IC, reliability and trustability are directly linked to low (zero) defects at customer side once a chip is delivered, in ensuring test paths on which we can trust whatever the external event the IC will face.

A. Test securization enhanced by sensors in ICs

Test infrastructures in secure integrated circuits are effective threats for security. It is a must to protect the test paths accessibility and integrity also against physical and environmental attacks.

In addition to their roles for product security in user mode in the field, environmental sensors are also present to prevent security weaknesses of the test infrastructure in IC.

For instance, one can modify the Test Access Port (TAP) configuration, extract keys to unlock the TAP or send or force unexpected data on the scan chains, in order to extract sensitive information.

One primary protection is the Hardware physical shield. It is required to prevent scan chain or secure registers (or nets) monitoring or forcing by needles in microprobing for instance. One could inject specific data sequence on the chain, to extract secure information and bypass or destroy the security level of the commercial product.

Frequency monitors are one guard band to prevent accelerating or decelerating the internal or external circuit clock frequency. This clock frequency modulation could be used to better control the attack sequence and extract sensitive data.

Forcing supply voltages (large wires) can be tried as well to decrease or increase the supply voltage to possibly enter the chip in an unsecure mode or find a grey voltage area not covered by voltage monitors. Internal and reliable supply monitors are an excellent complement to limit such physical intrusive threats.

Supply glitches may modify the TAP behavior, might impact the chains content and partially access to the functional circuit part. Cumulated with side channel analysis, test paths may be modified or modulated to access to some memory content for instance or extract critical registers.

Temperature sensors are commonly present on secure ICs but risk of attacks through the tests structures in playing with temperature itself remains low to medium. First, because the circuit weakness (expected by the attacker) at high or low temperature can be evaluated at test house during characterization. We can ensure that the test paths integrity is kept before reaching the temperature sensor limits or the first controlled functional fail. However, this assumes that the temperature sensor spread in temperature remains low to enable a trustable and accurate evaluation by characterization. Secondly, on the attack side, it is hard to control what happens within circuit test paths only in playing with temperature. Attack may need to be coupled with another.

Laser shooting is a relatively easy way to access and modify the test chains content or registers. For critical registers or keys opening access to the test chains, registers duplication or integrity checks can be implemented. However, as attack schemes can be more complex than flipping one or a few bits, on-chip laser sensors, when numerous and positioned very close to the critical DFT controls, can be very effective to flag unexpected (out of security target) behavior.

As highlighted above, to prevent environmental and physical threats through the test paths, the hardware circuit must have reliable sensors in terms of variability and functional trust ability. This means that they have to be characterized accurately (possibly thanks to dedicated DFT add-ons then) AND their capabilities verified in the field.

This second requirement means that the analog IP (sensors) must be verified through dedicated BIST (Analog BIST controllable by the IC TAP) or similar circuitry at power up for instance. As a consequence, sensors require an access protocol compliant with BIST and an embedded DFT layer in sensor that goes as close as possible to the intrinsic analog sensing structure or transistors.

At test, in controlled environment (for measurements and accurate voltage references drive), in addition to usual functional characterization (temperature, corners, preconditioning), test infrastructures must naturally be evaluated and their auto-test characterized.

B. Security in test of ICs

In order to guarantee reliability of ICs in the field, providers have to detect structural defects in production sites to reach a high coverage rate.

Currently, SoCs become more and more complex, embedding IPs from third parties. ICs providers have to use efficient, trustable and scalable test infrastructures. For advanced technology, “at speed” structural testing is the industry state of the art, using IEEE 1149, IEEE 1500 and IEEE 1687 standards. It allows:

- ICs testing with high coverage rate,
- Design partitioning, thanks to reconfigurable scan network and
- On-chip instruments access.

This advanced test infrastructure could open security breaches if not correctly handled. ICs providers face many kinds of attacks such as external/internal threats, including data sniffing or corruption by on-chip untrusted IPs.

Figure 1 represents a generic overview of a Secure Test Access Interface which embeds some security add-ons (light grey) to fulfill requirements of secure testing.

The standard IEEE 1149.1 TAP FSM allows accessing the test infrastructure through standard registers (TDR, IR, IDCODE, BYPASS, …) plus some additional custom ones. Secure registers are used to interface with the Controller. To improve security level, the SECURE TAP:

- controls the access to TAP FSM when user is authenticated (ACCESS CONTROL),

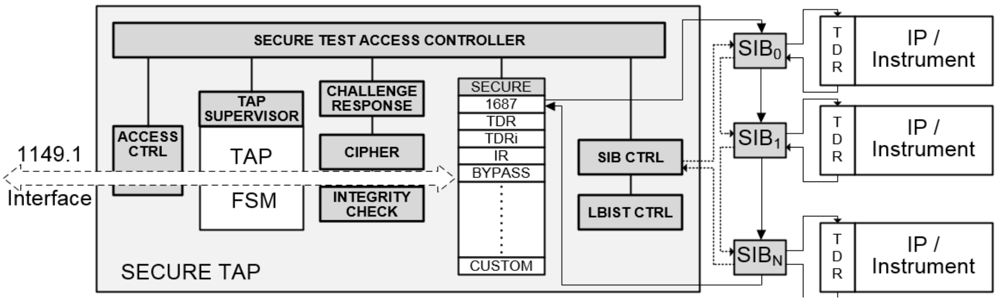

Figure 1. Secure Test Access Interface overview

Table 2. LBIST vs. functional self-test

	Functional self-test	LBIST
Fault coverage (stuck-at faults)	Not practical to assess	LBIST CAD tools allow to easily compute it.
Execution time	Slow (tens of seconds) Software execution time	Fast (less than 1 second) All scan chains are tested simultaneously
Security	Hypothetically, do not bring more vulnerability than functional code	Potential vulnerabilities to assess
Power/IR drop	Nominal Same as functional mode	Higher than functional mode, comparable to scan mode, needs to be addressed
Coexistence with functional mode	Partial self-test is possible on currently unused functions	The coexistence between parts of the design in functional mode and in self-test modes is tricky.

- monitors the correct behavior of the TAP FSM (SUPERVISOR),
- checks the integrity and ciphers/deciphers scan chain input stream,
- authenticates the user through a CHALLENGE/RESPONSE engine,
- supports access to secure SIB, allowing SOC partitioning, and
- BIST modules (Memory, Logic and Analog).

Those concepts will be presented in next sections.

C. *Logic BIST for Secure ICs*

BIST (Built-In Self-Test) has been used in most digital ICs for decades for the industrial test of embedded RAM and ROM. This technique, also referred to as MBIST (Memory BIST), allows to sort out devices that contain defective memory points after the foundry steps.

Logic BIST, illustrated on Figure 2, also referred to as LBIST, is a more recent technique that has emerged in the automotive sector [1]. It is not used only once after the foundry process but repeatedly to ensure that the vehicle operation is correct.

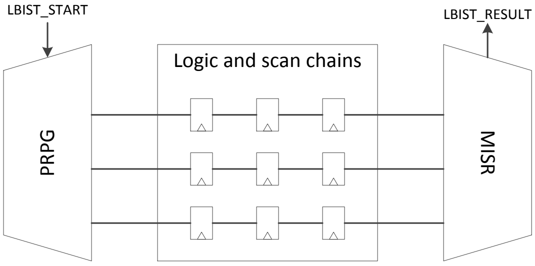

Figure 2. LBIST synoptic

LBIST is based on scan chains. But unlike full scan, it does not require an industrial tester to feed the IC with reference vectors and compare the outputs with known good patterns. Instead, the input vectors are generated by a PRPG (pseudo-random pattern generator) and the outputs are “compressed” into a signature by a MISR (Multiple-Input Signature Register). At the end of the test, the signature is compared against a known good reference and a pass/fail result is asserted.

For the mission-critical secure ICs that are the targeted use cases for LBIST in the HADES project, LBIST comes as an alternative to functional self-test. A functional self-test is most often realized by a dedicated software running on the processor cores that are embedded in the IC. Table 2 compares LBIST with functional self-test for our use case.

To assess the reliability of the device at its power-up or full reset, the following steps, combining MBIST and LBIST are conducted by the IC as shown in Figure 3:

Combining MBIST and LBIST covers most of the digital resources of the IC. The main area that is not covered is the pads and the external connections. A hardware reset following the LBIST is necessary as the LBIST puts the IC in a pseudo-random state.

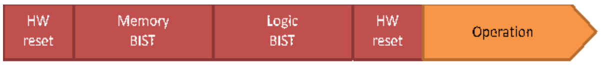

Figure 3. LBIST synoptic

We do not expect with LBIST to reach the 99% stuck-at coverage required for industrial test as pseudo-random vectors generated by the PRPG cannot compare with directed patterns computed by an ATPG (automatic test pattern generator) to address faults that are difficult to control or observe. Therefore LBIST – for recurrent self-test – will still coexist with full scan controlled by an external tester – for post-foundry test. One notable point is that full scan and LBIST use the same scan chains and the extra silicon cost should be limited.

Finally, the specific requirements we want to cover in the HADES project are shown in Table 3.

III. Authentication-based solution for test infrastructure

The high level of controllability and observability of the circuit brined by the test infrastructure raises security problems. It can be a mighty tool in the hand of malicious users for spying or tempering the data compute by the circuit. The objective of this section is to propose an optimized mechanism to ensure the security of the test infrastructure by authenticating any user wanting to access a restricted part. In addition, authentication allows providing a personalized access authorization for each user.

This section will firstly present some of the already existing solutions found in the literature, then present a new strategy that answers the problematic, and finally the evaluation and some comparisons with the state of the art.

A. Previous solutions

During the last decade, studies have been made to control the access inside the test architectures. Two interesting strategies will be presented in this part.

The Locking Segment Insertion Bit (LSIB) [3] is a distributed mechanism that makes a user able to lock or unlock the state of a SIB with the knowledge of the appropriate key. The user has to insert the key trough the scan chain, inside a key register. This key register is distributed in the scan chain: its position being unknown to the attacker, it is harder to find the key. Strategies to improve the security level of this method are described in [4]. This approach has however a non-negligible weakness: the user must plainly shift-in the key, so a replay attack would be successful.

Avoiding this weakness makes necessary using an authorization protocol where secret keys are not plainly exchanged. Most of the corresponding strategies use a challenge/response protocol. Many of these protocols use cryptographic functions, such as hashing [5], symmetrical encryption [6] or asymmetrical encryption [7]. Due to the size of the cryptographic functions, it is not appropriate to distribute the authorization system, so one of the remaining possibilities is to centralize the authorization in a unique controller and then distribute its authorization. The technique presented in [8] relies on using a second chain to distribute the opening authorization. The protected SIBs, called Secure SIBs (S²IB), are connected to a second network: the secure scan chain (SCC). Those SIBs can only include subparts of the network and are unlocked if an authorization is delivered through the secure scan chain. A controller drives the secure chain, it makes use of a challenge/response protocol needing a specific key for each targeted instrument (i.e., each protected network sub-chain).

B. Proposed Segment Set Authorization Key strategy

Our new approach must guarantee the same features than the S²IB based strategy but aims at adding features while reducing costs.

The main feature addresses the key management; the previous version, the S²IB, uses many secret keys for the authentication, one for each secured SIB. It is not a major issue in its context because the design does not aim for reconfigurable keys, so the storage used a cheap ROM. Due to our specifications requiring reconfigurable keys, it becomes a problem. Indeed the reconfigurable memories such as EEPROM or FLASH are expensive.

To be able to use a reconfigurable memory without a large increase of the implementation costs, a sub-key generation system is proposed. Instead of storing many keys inside the memory, only a single key (the circuit key) is stored, whereas the keys used for each authorization are procedurally generated. To decrease computation time, the system uses configuration strings combined with the circuit key instead of several instrument keys; therefore, only one cryptographic operation is needed to unlock any combination of instruments once the specific key is generated. The proposed approach is called "Segment Set Authorization Key" (SSAK) strategy, and this section briefly describes its operation and its architecture. The secured SIBs are similar to [8]; the novelty is in the authentication procedure.

1) Key and configuration generation

The SSAK is obtained from its related configuration value and the unique circuit key, using a symmetrical encryption function or a hash function. The cryptographic function used should not allow the owner of the SSAK to recover the circuit key nor the configuration.

Table 3. LBIST HADES requirements

Requirement	Remark
Keep LBIST execution time under 1 second	Short boot sequence (user benefit)
Stuck-at coverage ≥ 90% Transition fault coverage ≥ 60%	Comparable to automotive domain ISO 26262 ASIL B
Reduce the attack surface (external access to test interface and to configurable values)	Specific requirement for our application domain.
Prevent side-channel attacks by avoiding operation on sensitive data during LBIST shift and capture	Specific requirement for our application domain. A prior reset looks like a simple yet elegant solution.
Power envelope: comparable to functional mode	For battery-operated devices. A solution is to lower the LBIST clock frequency.
Explore low-switching rate PRPG to reduce the IR drop	It is necessary to reduce internal voltage drops in the internal power grid to remain in the CMOS technology specifications.
Avoid adding clock structures to existing On-Chip-Clocking (OCC) used by full scan	In order not to add complexity for CTS (clock tree synthesis) and STA (static timing analysis)

The segment set code, contained in the configuration vector, represents the secure scan chain; each bit of the code refers to a secure SIB. A bit of the code set at one stands for asking access to the sub-chain controlled by the related S²IB and a zero is used for locking it.

2) *Authentication protocol*

The new connection protocol is summarized hereafter, where each bullet represents one Capture Shift Update (CSU) cycle.

- The user opens the controller's SIB.
- Exchange of the challenge and the configuration vector. The user sends the configuration and receives the controller's challenge.
- The user already has the configuration related SSAK, both are combined to get the challenge response. Since this specific SSAK is not stored in the circuit, the controller needs to generate it from the circuit key, and then uses it to solve the challenge.
- The user sends the response to the controller.
- The controller checks the response; if correct, the targeted S²IBs are unlocked.

If a user, owing a SSAK and the corresponding configuration vector, wants to access an instrument that is not allowed, the Segment Set Code has to be changed, to include the new instrument. If this user tries to answer the challenge response protocol with the original key, the controller will notice noncompliance between the SSAK and the new configuration and will reject the connection.

C. Performance

In order to evaluate the performances and the costs, the new strategy is compared to the previously presented strategies in terms of connection latency and area. The BASTION Benchmark library [9] offers in free access a set of various internal RSN circuits that we use for evaluation. For the implementation reported here, the controller uses a SHA-256 hash function to generate the SSAK and resolve the challenge.

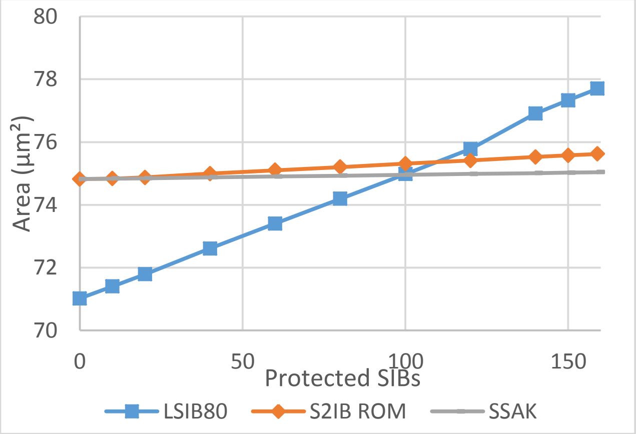

Figure 4 Implementation area on T512505, with up to 159 protected SIBs

1) *Area*

The T512505 circuit is a hierarchical scan chain composed of 192 scan segments and 159 SIBs. Its full length is 77,005 bits, and it has been chosen because its large number of SIBs permits a large range of study (from 1 to 159 protected SIBs).

Figure 4 is the result of the implementation of the strategies [3] (LSIB) with keys of 80 bits, [8] (S²IB) and the SSAK, for 0 to 159 protected SIBs on this circuit. The security systems have a small impact on the overall area of the circuit, around 6% overhead. Aside from security issues, the LSIB is the only one to have a significant dependency on the number of protected SIBs that makes it competitive for few SIBs to protect, but more expensive when their number increases.

2) *Test time overhead*

The second objective is to provide a faster connection than the other existing solutions. To evaluate the performance in this domain a simulation model was created, based on a behavioural model of the test. With this model, it is possible to know the length of the chain for any configuration of its protected SIBs, and compute the time needed to open all the SIBs. It is also possible to add the LSIB structures inside the chain and directly compute the opening time. For the two other structures, our simulation model uses the original circuit but adds the authentication time.

As expected, both S²IB and LSIB strategies observe an opening time highly dependent on the number of protected SIBs, because each LSIB adds key registers in the scan chain, and the S²IB strategy needs one additional hash operation for every extra protected S²IB. On the contrary, the SSAK approach offers a constant and small opening time due to its constant number of cryptographic operations.

IV. 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 encrypting test data using the secret key. Encrypted data are shifted through 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 encrypted data, which is unintelligible. The responses generated by the target device, are encrypted before being shifted out the TDO pin. All downstream entities receive encrypted 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 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 encryption 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*.

C. Test Data Encryption 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.

1) Area and Power Consumption

Table 4 shows the area and power consumption of the PRESENT block cipher with 128-bit secret key and the TRIVIUM stream cipher.

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.

2) 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 [10]. Therefore, block ciphers have to be adapted, in order to cope with the serial interface of the testing infrastructures.

Table 4. Area and Power Consumption

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

Table 5. Comparison Overview

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

3) 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. Table 5 summarizes pros and cons of both solutions.

V. Conclusions

This paper presents both current industrial solutions and complementary new proposals to secure an IC with respect to attacks on the test infrastructure. The impact of each proposal is briefly discussed and illustrated. More complete evaluations are on-going and a full demonstrator on an industrial test case is being developed.

Acknowledgment

This work has been funded by the French Government under the framework of the PENTA HADES ("Hierarchy-Aware and secure embedded test infrastructure for Dependability and performance Enhancement of integrated Systems") European project.

References

[1] Nikhil Garg, Sagar Kataria, Abhishek Mahajan, Anurag Jindal (Freescale Semiconductor), *LBIST – A technique for infield safety*, Design&Reuse, January 2015, https://www.design-reuse.com/articles/38290/lbist-a-technique-for-infield-safety.html

[2] IEEE, 1687-2014, "IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device," 2014.

[3] J. Dworak, A. Crouch, J. Potter, A. Zygmontowicz and M. Thornton, "Don't Forget to Lock your SIB: Hiding Instruments using P1687," in IEEE International Test Conference, 2013.

[4] A. Zygmontowicz, J. Dworak, A. Crouch and J. Potter, "Making it harder to unlock an LSIB: Honeytraps and misdirection in a P1687 network," in Design, Automation and Test in Europe Conference and Exhibition, 2014.

[5] E. Koopahi and S. E. Borujeni, "Secure scan-based design using Blum Blum Shub algorithm," in East-West Design & Test Symposium (EWDTS), 2016.

[6] M. Aigner and M. Feldhofer, "Secure Symmetric Authentication for RFID Tags," in Telecommunication and Mobile Computing, 2005.

[7] J. Da Rolt, S. Ghosh, S. Seys, S. Dupuis, G. Di Natale, L. Marie-Flottes, B. Rouzeyre and I. Verbauwhede, "Secure JTAG implementation using Schnorr protocol," Journal of Electronic Testing, pp. 193-209, 2013.

[8] B. Rafal, K. Michael A and H.-J. Wunderlich, "Fine-Grained Access Management in Reconfigurable Scan Networks," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, pp. 934-947, 2015.

[9] A. Tsertov, A. Jutman, S. Devadze, M. Sonza-Reorda, E. Larsson, F. Ghani Zadegan and R. Krenz-Baath, "A Suite of IEEE 1687 Benchmark Networks," in 47th IEEE International Test Conference, 2016.

[10] Emanuele Valea, Mathieu Da Silva, Marie-Lise Flottes, Giorgio Di Natale, Bruno Rouzeyre, "Stream vs block ciphers for scan encryption", *Microelectronics Journal*, Volume 86, 2019, Pages 65-76