Enhancing DFT Security in Chiplet-Based Systems with Encryption and Integrity Checking

Juan Suzano, Anthony Philippe, Fady Abouzeid, Giorgio Di Natale, Philippe Roche

Enhancing DFT Security in Chiplet-Based Systems with Encryption and Integrity Checking

Juan Suzano[†‡§], Anthony Philippe[‡], Fady Abouzeid[†], Giorgio Di Natale[§], Philippe Roche[†]
[†]STMicroelectronics, Crolles, 38920, France {juan.suzano, fady.abouzeid, philippe.roche}@st.com
[‡]Univ. Grenoble Alpes, CEA, List, F-38000 Grenoble, France {anthony.philippe}@cea.fr
[§]Univ. Grenoble Alpes, CNRS, Grenoble INP, TIMA, 38000 Grenoble, France
{giorgio.di-natale}@univ-grenoble-alpes.fr

***Abstract*—Chiplet-based chips are the natural evolution of traditional 2D SoCs. In the future, off-the-shelf chiplets are expected to represent an important component of the semiconductor industry. The IEEE Std 1838(TM)-2019 design-for-testability (DFT) standard enable testing of stacked chiplets from multiple vendors. However, the shared DFT network threatens the confidentiality and integrity of test data and other sensitive information. This paper addresses the security concerns associated with DFT infrastructures in chiplet-based systems. We discuss the necessity of securing DFT infrastructures to prevent unauthorized access and malicious activities. Furthermore, we propose a hardware countermeasure that combines encryption and encoding to secure communication over the DFT network. Results show that the DFT can be protected from misbehavior from malicious chiplets on the stack, scan-based attacks, and brute force attacks with minimal overhead in terms of area and test time. The proposed solution causes less than 1% area overhead on designs composed of more than 5 million gates and less than 1% test time overhead for typical DFT implementations.**

***Index Terms*—2.5D, 3DIC, Chiplet, Design for Testability (DFT), Hardware Security, IEEE 1838, IEEE 1687, IJTAG, Root of Trust, Scan Encryption**

I. Introduction

The semiconductor industry is expected to witness a paradigm shift towards the adoption of chiplet-based systems, which are the natural evolution of traditional 2D System-on-Chip (SoC) [1]. Historically, SoCs have had only one layer of transistors. In contrast, 3D Integrated Circuits (3DICs) expand vertically by integrating multiple layers of computing logic within the same package. This is achieved by stacking multiple pre-manufactured dies, known as chiplets. Naturally, each die must be tested before assembly to ensure that only known-good-dies are stacked. Additionally, a post-bond test process is required to detect defects caused by the stacking process.

The post-bond testing of chiplets is complicated by the fact that designers do not know in advance the stack architecture into which the chiplets will be integrated, requiring the standardization of test infrastructure access. While the IEEE 1687 Standard (IJTAG) [2] standard defines the infrastructure for accessing design-for-testability (DFT) elements within a die, the IEEE Std 1838(TM)-2019 (IEEE 1838) DFT standard [3] defines mandatory and optional DFT infrastructures for both pre- and post-bond tests of chiplets on a stack.

IEEE 1838 compliant chiplets form a 3D DFT network where DFT elements are accessible from the bottom die. This means that test data transmitted by the Automatic Test Equipment (ATE) may pass through multiple dies before arriving at the target chiplet. In the context of off-the-shelf chiplets, the test data is exposed to potential misbehavior by untrusted chiplets. Untrusted chiplets can spy on and sabotage the communication of test patterns, test responses, cryptographic keys, activation bitstreams, and other sensitive data. Therefore, the secure adoption of off-the-shelf chiplets requires, among other things, the development of techniques that secure the 3D DFT infrastructure [4].

Many works have demonstrated the use of the scan chain to leak secret information, such as cryptographic keys [5], [6]. Although the literature still lacks works demonstrating this type of attack on 3DICs implementing the IEEE 1838 standard, it is reasonable to assume that 3DICs are at least equally vulnerable to scan-based attacks. Scan encryption has been proposed to protect the confidentiality of test data transmitted over the scan chain [7]. However, scan encryption techniques do not prevent an attacker from writing random data to the internal flip-flops of the device, which is sufficient to mount an attack.

In this work, we expand on the work first introduced in [8]. We present a scan encryption technique combined with a lightweight message integrity checking to ensure that only authorized users can access the scan chain of an IEEE 1838 compliant chiplet. The proposed countermeasure ensures that an adversary unaware of the secret key is not able to write meaningful data to the scan chain, while the integrity checking mechanism ensures that tampered test patterns and brute force attacks are easily detectable. Along the way, we provide an in-depth justification for putting effort on securing the DFT of chiplet-based systems. We also elaborate on multiple attack vectors and capabilities, as well as showcase some solutions available in the literature. Additionally, we study the typical DFT infrastructure of a chiplet-based chip and provide guidance on how to integrate our solution into the DFT while remaining compliant with the IEEE 1838 and IJTAG standards. This work also provides an extensive discussion on the security aspects and cost of the proposed countermeasure.

The remainder of this paper is organized as follows.

Section II argue for the importance of DFT security for chiplet-based system. Section III presents the threat model considered for this work. Section IV provides a non-exhaustive state-of-the-art study. Section V elaborates on the DFT of

chiplet-based chips. Section VI presents the proposed countermeasure. Section VII elaborates on the integration of the proposed countermeasure on the DFT. Section VIII presents the experimental evaluation of area and test time overhead. Section IX discusses security aspects and overhead. Finally, Section X presents our conclusion.

II. DFT Security - Necessity and Opportunity

This section highlights the necessity of securing the DFT infrastructure under the light of the move towards an open chiplet ecosystem. Moreover, it also highlights the DFT infrastructure as the first viable platform for security features for chiplet-based systems.

The emergence of chiplets as a central component in semiconductor manufacturing has been anticipated for several years [9]. Researchers and industry leaders continue to debate and develop the standards that will enable this disruptive change in the industry. Initiatives like Universal Chiplet Interconnect Express (UCIe) [10] aim not only to provide solutions for vertically integrated chiplet-based systems but also to lay the foundation for an open chiplet ecosystem.

However, the standards that will enable true plug-and-play functionality for chiplets are not yet fully developed and adopted. This gap in standardization poses challenges for the integration of chiplets from different vendors and for the implementation of security features. Despite this, IEEE 1838 design-for-testability (DFT) standard [3] has been published and is already supported by Electronic Design Automation (EDA) tools [11]. The IEEE 1838 defines mandatory and optional structures for accessing DFT functions on the chiplet independently of the multi-die stack architecture.

The importance of the IEEE 1838 standard on the upcoming chiplet ecosystem makes securing it against old threats — inherent to DFT insertion — and new threats — posed by off-the-shelf chiplets — a crucial step towards adoption.

Closely related to the necessity of protecting the DFT infrastructure is the need to secure the post-stacking test process itself. The post-stacking test represents the first critical moment in terms of security in the lifetime of the chiplet-based system. During this process, several sensitive operations may be carried out, making it a prime target for adversaries:

- The test itself may involve the handling of confidential data, which can be stolen and exploited to reveal secret information about the chiplet design.
- Cryptographic keys may need to be loaded onto the chiplet as part of a broad security scheme.
- Unlocking keys may need to be loaded on logic-locked chiplets
- Confidential set-up metadata may need to be loaded onto blank chiplets.

Here, a clear convergence of different needs can be observed. First, there is the straightforward requirement of ensuring the confidentiality and integrity of test data. Second, there is the necessity of preventing the exploitation of the DFT infrastructure for malicious activities. Third, there is the need for a Root of Trust (RoT) to enable plug-and-play operability of security features—such as the unlocking of logic-locked chiplets—independently of the stack architecture.

III. Threats and Attacker Model

This section explores the threats against the confidentiality, integrity, and intellectual property of chiplet-based systems, highlighting the chiplet paradigm and the exploitation of DFT infrastructure.

A. Attack Vectors

DFT security is a well-known topic in the literature, with numerous works defining attacker models in the context of traditional 2D SoC [6], [12]–[14]. In summary, without proper security measures, the DFT infrastructure can be exploited by any actor with uncontrolled access to it. The chiplet paradigm exacerbates the situation by introducing additional points of entry for attackers, as adversaries can now infiltrate the chiplet-based system in the form of a compromised chiplet. This work highlights the following attack vectors:

- **Malicious Chiplet:** During the supply chain process, chiplets can be replaced by malicious versions. These malicious chiplets may contain hidden backdoors or other modifications designed to compromise the security of the system.
- **Counterfeit Chiplet:** Counterfeit chiplets can pollute the production chain, leading to security breaches. These counterfeit versions may be the product of overmanufacturing, recycling, or stolen designs and may not comply with expected test practices.
- **Insecure Design Practices:** Chiplets designed with inadequate security measures can serve as entry points for attackers. These vulnerabilities can be exploited to gain unauthorized access to the entire chiplet-based system.
- **Attacker at the Test Facility:** An attacker with access to the test facility can tamper with the testing process, stealing secret information, introducing vulnerabilities, bypassing security enrollments or performing complex attacks. This can lead to compromised chiplets being passed as secure.
- **Attacker in the Field:** Once deployed, chiplet-based systems can be targeted by attackers in the field. These attackers can exploit vulnerabilities found and/or introduced during the test process. They can also leverage the DFT to perform attacks.

B. Attacker Capabilities

In an IEEE 1838 DFT architecture, chiplets share the same DFT network, which introduces significant security vulnerabilities. The data transmitted over the DFT data path is susceptible to misbehavior by any chiplet on the network. To access an element of a chiplet's DFT infrastructure, the user must send a specific instruction to the target chiplet and the 'bypass' instruction to the other chiplets in the stack. A chiplet could be easily designed to store the data passing through its bypass register for later use. This enables an attacker to steal test patterns, test results, internal states, unlock keys, and cryptographic keys.

Furthermore, a malicious chiplet can actively modify the data transmitted to other chiplets, leading to incorrect test

results, masking the presence of defects, introducing new faults, or enabling more complex attacks.

C. Known Attacks

Several attacks targeting DFT structures have been proposed in literature [6]. Among the various DFT elements used to test ICs, scan chains are one of the most studied as they are widely used as DFT solution. Scan chains improve the observability and controllability of the system by allowing external access to flip-flops (FF) within the IC [15]. Nevertheless, an attacker can exploit the scan chain to reverse engineer the design, leak secret information or bypass security mechanisms.

Scan-based netlist extraction is a non-invasive reverse engineering technique where the attacker attempts to retrieve the netlist of portions of the design. The basic premise involves treating scan flip-flops as inputs and outputs of a function F. The attacker then shifts known test vectors into the scan chain to approximate or discover F. Although this method has its limitations, its feasibility has been demonstrated on a variety of benchmarks, including circuits from the ISCAS'89 benchmark suite, a Bitcoin SHA-256 accelerator, and cryptographic ciphers [16]–[19].

Scan-based attack is a type of attack where the adversary leverages the scan chain to leak security-critical assets from the system. In this type of attack, the attacker gains access to the scan chain, which is used for testing and debugging purposes. By manipulating the scan chain, the attacker can shift out sensitive internal states and data, effectively extracting confidential information that can be used to retrieve the secret keys of cryptographic algorithms, for example [6].

Attacks against logic-locked designs are more effective when the scan chain is accessible [20]. Logic locking is a technique used to protect intellectual property by inserting additional key-controlled gates into the circuit, making the correct functionality dependent on a secret key. When the scan chain is accessible, Boolean satisfiability (SAT) based attacks can be particularly effective in extracting the logic locking secret key. The oracle provided by scan chain access allows the attacker to quickly verify the correctness of the guessed keys, significantly reducing the complexity and time required for the attack [21].

The chiplet paradigm strengthens the scan-based and SAT attacker models by providing an entry point for attackers. A malicious chiplet, a chiplet running malware, or a chiplet with exploitable design flaws can compromise the security of all chiplets in the 3DIC.

IV. Related Works / State-Of-The-Art

This section explores some of the typical solutions for protecting DFT structures against the attacks mentioned previously in section III. Since the paradigm of chiplets and 2.5D and 3D chips is relatively recent, the literature on DFT security in this context is still very sparse. To the best of our knowledge, the work first introduced in [8] - and further explored in section VI of this article - is the first to provide a security solution specifically focused on protecting an implementation of the IEEE 1838 multi-die DFT standard. It can be said that the solutions present in the literature dedicated to protecting the DFT of traditional SoCs can be adapted to the context of chiplets. Therefore, to provide additional reading for interested readers, we spend the remainder of this section shedding light on some techniques and solutions present in the "traditional" 2D SoC literature.

A logical approach to preventing attacks that exploit the DFT infrastructure is to limit access to it to only authorized users. This can be achieved through the use of access passwords, where the user must load a password into the Device Under Test (DUT) to gain access to its DFT, or through more complex protocols such as a challenge-response protocol, where the DUT generates a challenge and the user must compute a response based on a shared secret. Additionally, the system can be configured to provide different levels of access to different users. Another approach involves real-time monitoring of user behavior through detectors that analyze the sequence of test instructions and test patterns. A detailed exposition of these techniques, as well as additional reading material, can be found in [12].

Next, we highlight solutions that use cryptography to protect the confidentiality of test data, as well as to prevent an unauthorized user—specifically, a user who does not know the secret key—from writing meaningful data into the test structures. The most commonly used technique in the literature for encrypting test data is the insertion of stream ciphers in the serial path at the input and output of the DFT elements to be protected. Block ciphers are generally considered inadequate for this type of task because they typically require multiple clock cycles to encrypt the plaintext. A detailed study on the use of stream ciphers and block ciphers for scan encryption can be found in [22].

In [23], [24], the Trivium stream cipher was used to encrypt the serial path of an IEEE 1149 (JTAG) infrastructure, preventing malicious chips from "sniffing" the test data related to another chip on the PCB. In [25], [26], the same stream cipher was similarly used to protect communication with a core against misbehavior from third-party IPs within a SoC. In [27], the authors utilized lightweight block ciphers (PRESENT and SKINNY-64) to encrypt the serial path of an IP within a SoC. In this case, the authors employed buffers to circumvent the issue that the PRESENT and SKINNY-64 block ciphers take 36 and 32 clock cycles to encrypt their input block, respectively. Due to the fact that all the mentioned countermeasures utilize simple ciphers for the encryption, they all cause a small overhead of 1-5% when inserted into their respective benchmarks. In terms of test time overhead, all the cited solutions present a negligible overhead of around 1% in the worst cases.

Lastly, we are interested in solutions that protect the integrity of test data, thereby preventing attacks based on altering the legitimate communication between the tester and the DUT. Again, the literature is sparse regarding this type of solution. Most countermeasures focus on the inclusion of shadow test data registers in parallel with third-party IPs in the serial path [28]–[30]. In this way, if a malicious IP modifies the test data, this misbehavior is easily detected by the countermeasure. It is necessary to point out that this type of solution is inadequate

for chiplet-based systems because the chiplet integrator does not have the capability to replicate this solution in a chiplet stack. In [23], the authors protect the integrity of test data by generating a Message Authentication Code (MAC) for each message exchanged between the tester and the DUT. This is achieved through the use of an incremental MAC function, which is a type of MAC function that allows the calculation of the authentication code to be performed incrementally, that is, as the bits of the message are received serially. Incremental MAC functions are particularly useful in this scenario because they allow the function to be inserted into the serial path of the DFT in a straightforward manner. The authors reported an area overhead of 56k to 114k gates for the entire implemented security system, depending on the configuration used. The article lacks a detailed presentation of the cost of each component of the system. However, we can presume that most of the cost is related to the implementation of the MAC function and the Trivium stream cipher.

V. The typical DFT of a Chiplet-Based Chip

Over the years, many DFT standards have been developed to enable efficient testing of integrated circuits. In the context of 3DIC testing, EDA tools leverage mainly the IJTAG and IEEE 1838 standards to provide designers with a complete access infrastructure to DFT elements such as scan chains and instruments [31]. The first, IJTAG, is a well-established standard introduced to enhance the accessibility of embedded instruments within ICs. The primary goal of IJTAG is to enable more complex and dynamic DFT architectures, in contrast to the single serial path provided by the JTAG standard. The second, IEEE 1838, is a multi-die-specific DFT standard that solves the problem of providing access to the DFT elements of all chiplets in a stack, independently of the stack architecture.

A. IEEE 1838 Standard

In the production of 3DICs, pre-fabricated chiplets are stacked in a complex process that can lead to damage or faulty connections between the chiplets. Therefore, a post-bond test is required in addition to the conventional test of each die. The testing of chiplet-based 3DICs is complicated by the fact that chiplets can come from different sources. When developing a chiplet, the designer may not know the architecture into which the chiplet will be assembled. For various reasons, it is not feasible for each chiplet in a 3DIC to have its own access to the ATE. Therefore, the post-bond test requires a standard that enables chiplets from different suppliers to be tested.

The IEEE 1838 standard defines both mandatory and optional on-chip circuitry for 3DIC testing. The standard is die-centric, meaning the DFT features are added individually to each die and not to the stack. However, when compliant dies are stacked, they form a comprehensive DFT architecture that enables stack-level testing. Additionally, the IEEE 1838 standard does not require any specific assembly scheme and supports 2.5D, 3D, and 5.5D configurations.

Access to the on-die DFT features is achieved through a Test Access Port (TAP). A compliant die must have a Primary TAP (PTAP) so that the ATE or other dies can access its DFT structure. The TAP is the same as that of the JTAG standard and consists of five terminals: TCK, TMS, TDI, TDO, and TRSTN.

The PTAP signals drive the PTAP Controller, which is an JTAG compatible finite state machine (FSM) and a mandatory element of the standard. The PTAP must implement, at a minimum, the following elements:

A bypass register that bypasses all DFT elements present on the die and essentially excludes them from the serial path. A die-wrapper register that goes on the boundary of the die, enabling testing within and between dies. An instruction register that stores the PTAP instruction and controls the logic that supports the other registers and DFT elements. The Bypass, Die-wrapper, and Three-Dimensional Configuration Register (3DCR) registers are categorized as data registers. Other data registers can be implemented as needed. In fact, the scan chain is considered a data register in the IEEE 1838 standard.

The interface between dies is realized by the Secondary TAP (STAP), which consists of a TAP and control logic. The control logic is driven by the 3DCR and is used to insert the next die into the serial path or to bypass it. A compliant die must have a STAP for each die to which it is connected. Figure 1 shows a stack of two IEEE 1838-compliant dies. The PTAP controller of the first die controls the DFT functions on the die and is connected to the PTAP of the second die via its STAP port.

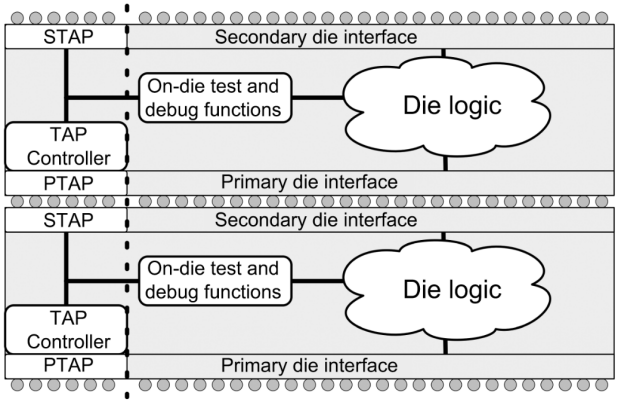

Fig. 1. Generic representation of an IEEE 1838 DFT architecture for a two-die 3DIC. (Image from [3])

B. IEEE 1687 Standard (IJTAG)

The IJTAG standard, also known as IJTAG (Internal JTAG), enables dynamic access to test elements within the chiplet without the need to traverse the entire scan chain. This is achieved by using Segment Insertion Bits (SIBs), which allow reconfiguration of the scan path. Figure 2 shows a hierarchical DFT architecture where embedded instruments can be individually accessed depending on the SIB configurations.

The IJTAG standard also defines Test Data Registers (TDRs) that are used to store and shift data to and from the embedded instruments. TDRs enable easy integration of instruments with different data requirements on the scan path. Additionally,

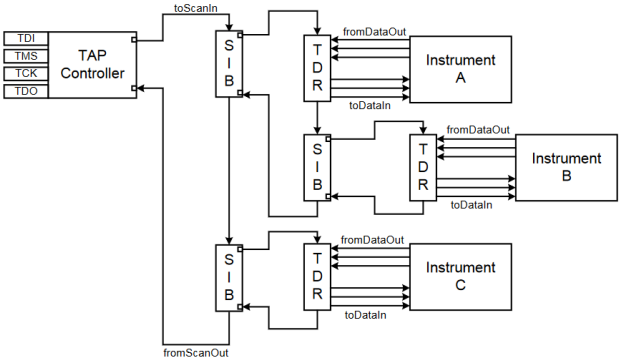

Fig. 2. Diagram of an IJTAG-based hierarchical architecture. (Image from [2]

the IJTAG standard introduced the Instrument Connectivity Language (ICL) and Procedure Description Language (PDL). ICL describes the hierarchical structure and connectivity of embedded instruments within the DFT network. It defines the relationships between instruments and the scan paths that provide access to them. PDL is used to specify the sequences of operations required to interact with instruments. It defines the actions needed to configure, control, and retrieve data from these instruments. [2] provides a detailed description of all IJTAG elements.

VI. Proposed Solution

A hardware countermeasure for the threats presented in Section III must prevent the adversary from reading useful data from the scan chain of the chiplet, as well as prevent the adversary from writing the scan chain. Blocking physical access to the DFT I/O could prevent in-field tampering at the expense of preventing in-field debugging and configuration. However, attacks at the test facility would still be possible. Additionally, the threat of untrusted chiplets in the 3D stack would remain unaddressed.

The countermeasure proposed in this work secures communication with the protected chiplet through a combination of encryption and encoding. The security principle is that any data transmitted to the chiplet through the serial path is encoded with a public encoding algorithm and encrypted. By doing so, we ensure that only entities with knowledge of the secret key can generate a compliant message, i.e., a message that can be successfully decoded after decryption. Therefore, an attacker cannot retrieve the meaning of the data transmitted over the scan chain nor apply unauthorized test patterns to the scan chain.

The general principle is illustrated in Figure 3. First, test data is encoded with a low-cost method to enable the verification of the integrity of the data. Later, the test data is encrypted using standard encryption techniques. The test data is then loaded onto the ATE, which performs the test according to standard test procedures. The test patterns are decrypted on the chiplet, applied to the scan chain, and the test results are encrypted on the chiplet before being sent back to the ATE. This scheme ensures that the test is obfuscated to the ATE, tester, and untrusted chiplets in the stack.

A. Architecture overview

An overview of the proposed countermeasure integrated into a simple IEEE 1838 DFT architecture is shown in Figure 4. For simplicity, only data paths are illustrated, and control signals and the clock are omitted. The chiplet implements the infrastructure needed to decrypt and decode the test patterns, as well as encode and encrypt the test responses. Encryption and Decryption (E/D) are performed on the die by two symmetric ciphers. The decryption module is inserted at the input of the serial path to decrypt the test data. Similarly, the encryption module is inserted at the output of the serial path to encrypt the test results.

The secret key must be securely stored on the chiplet. Therefore, a secure key management unit (SKMU) must be provided, along with a scheme for communicating the key between the chiplet and the ATE or administrator. Secure storage and secret key enrollment are out of the scope of this work. A true random number generator (TRNG) is used to generate an initial value (IV). Each of the E/D modules includes a mechanism to generate a new E/D key based on a combination of the IV and the secret key for every E/D operation. This is necessary to prevent replay attacks. By changing the E/D key for each E/D operation, an adversary attempting to replay a message would perform the E/D operations with the wrong key and produce random bits as output. However, the IV value must be known by the tester to correctly derive the encryption key. Thus, a scheme for communicating this value must be implemented. A simple PTAP Control instruction that puts the TRNG registers on the TDI to TDO serial path is sufficient. The fact that this value is shared with the external world does not jeopardize the countermeasure's efficacy, as the cryptographic key is still kept secret.

B. Encryption and Decryption Modules

The E/D modules are presented in Figure 5. Each module implements a symmetric block cipher. As the symmetric ciphers are deployed on the serial path, some logic is required to interface the two. Therefore, two shift registers (R1 and R2) are implemented for buffering. They allow the system to receive the data being shifted through the serial path and load data into/from the cipher and integrity modules in parallel. In other words, while the data is decrypted and integrity-checked on R1, for example, R2 simultaneously loads the scan chain with the previously decrypted and integrity-checked data and receives the next E/D blocks from the serial path. The control of the switch between both shift registers to ensure uninterrupted shift capabilities is performed by a FSM.

Test data integrity verification is performed by the integrity module connected to R1 and R2. The basic idea consists of encoding the plaintext message with a parity bit before encryption. After the decryption operation — and before shifting the data to the scan chain — the parity bit is checked, and the test is aborted if the message does not meet the integrity requirement, i.e., if the computed parity is different from the value of the parity bit. Similarly, the encryption block at the end of the scan chain contains an integrity module

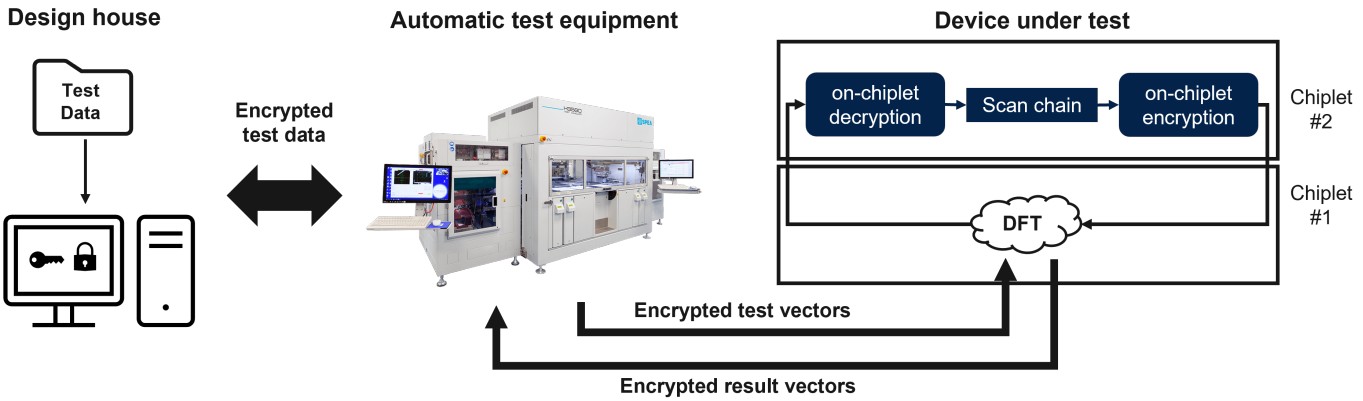

Fig. 3. Overview of the communication scheme.

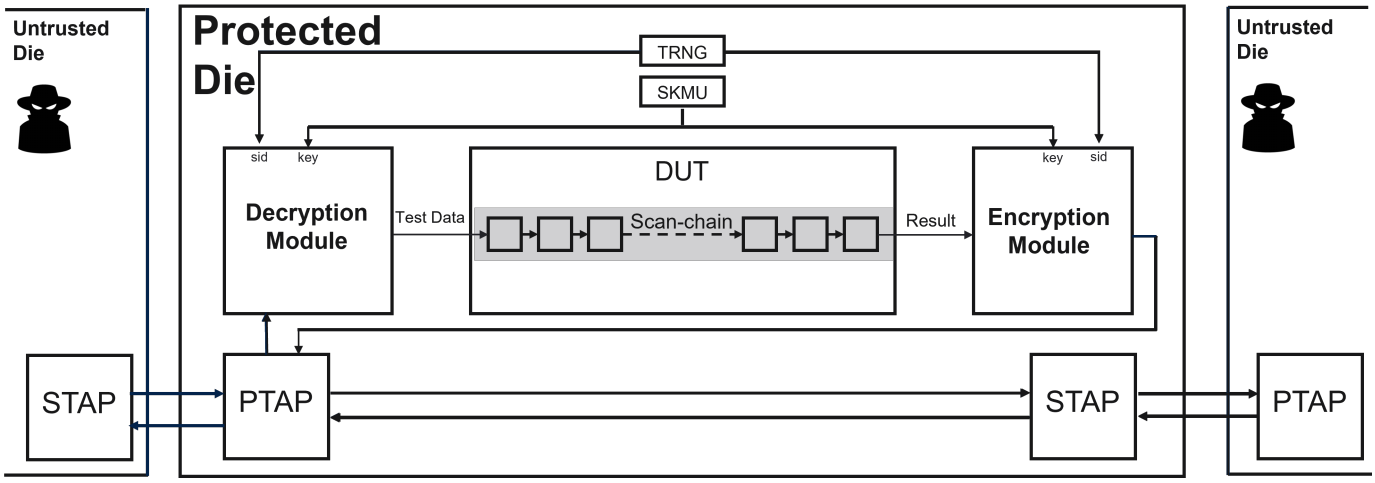

Fig. 4. Architecture overview for a protected die with the proposed scan encryption technique, stacked with others untrusted dies (Modified from [4]).

that produces a parity bit to ensure the integrity of the test responses.

The IV produced by the TRNG (see Figure 4) serves as input to a Pseudo Random Number Generator (PRNG). The PRNG produces a new value for each E/D operation. This value is XORed with the key received from the SKMU, ensuring a different cryptographic key for each E/D block.

Integrity Check

The basic idea of the integrity check scheme consists of encoding the plaintext with a publicly known encoding algorithm. The message, along with the encoding information, is then encrypted. The receiving device decrypts the message and checks its compliance with the encoding algorithm before applying it to the scan chain. The security principle of this approach is based on the following assumption: an unauthorized user is not able to forge a ciphertext in such a way that the resulting plaintext matches the desired format after decryption. An attacker who does not know the secret key is therefore unable to generate valid encrypted test patterns that successfully pass the integrity check.

The proposed countermeasure uses a parity algorithm as the encoding method. The parity bit is encoded in the test patterns before encryption. In the implemented scheme, the last bit of each encryption block is a parity bit. With a sufficiently large scan chain, the probability of the attacker sending a message with correct parity bits at the end of each encryption block is negligible. In fact, if L is the length of the scan chain and b is the block length, the number N of parity bits that must be added to the test patterns is equal to:

$$N = \frac{L}{b-1} \tag{1}$$

If N parity bits are added, the probability for the attacker to guess a valid ciphertext (i.e. a ciphertext that has valid parity bits after decryption) is 2^{-N}.

For this work, a 64-bit and a 128-bit block ciphers were tested. Thus, for every 128-bit encryption block, the 128th bit is a parity bit. It is computed by XORing the other 127 bits. Similarly, for every 64-bit encryption block, the 64th bit is a parity bit, computed by XORing the other 63 bits. The parity bits are computed off-chip during the encryption of the message. For every 128-bit or 64-bit on-chip decryption round, the decryption block calculates and compares the parity bit to ensure the integrity of the data. If any decryption block fails

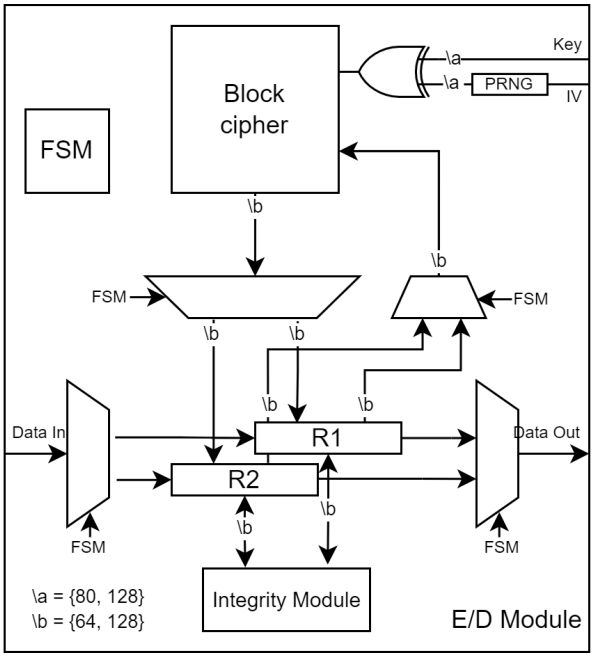

Fig. 5. Diagram of the architecture of the Encryption/Decryption modules (modified from [8]); Unlabeled data-paths are 1-bit wide; Data-paths labeled a are 80 or 128-bit wide and b are 64 or 128-bit wide for the PRESENT-80 or AES-128 implementations respectively.

the parity bit check, a flag is raised, and the shift operation is halted to prevent potential attacks.

The last cycle of the shift operation of the decrypted message is performed with the scan chain disabled. In this way, the parity bit is not inserted into the scan chain. This inserts an ”empty” bit into the test response that is shifted from the scan chain to R1 or R2 of the encryption module. Before the encryption of the test responses, the integrity module of the encryption block adds the parity bit to the ”empty” bit. In this way, the integrity of the test response can be checked by the tester.

VII. DFT Integration

The solution presented in Section VI is intended to be deployed seamlessly on the scan path without altering the intended functioning of a compliant die during the test procedure. Accordingly, the E/D modules are placed at the boundaries of the protected scan elements. When an unprotected scan element is put on the TDI-TDO serial path, system operation will not be affected by our design. This approach allows the implementation of multiple protected and unprotected scan elements. Considering a DFT infrastructure compliant with the IEEE 1838 or IJTAG standards, a protected scan element can be deployed as a protected IEEE 1838 Data Register or as a protected ICL-described scan chain.

A. Integration With IEEE 1838

The integration with the IEEE 1838 standard is straightforward as scan chains are interpreted as Data Registers. When the appropriate instruction is loaded on the PTAP controller, the scan chain is placed on the DFT serial path. When the TMS signal is used to drive the PTAP state machine, the appropriate scan, shift, and update signals are sent to the scan elements. Thus, designers are tasked with two main steps:

- Place the E/D modules at the boundaries of the Data Register.
- Account for the additional delay caused by the E/D modules on the test patterns.

This can be done prior to test pattern generation by adding shadow shift registers to the boundaries of the Data Register and replacing them with the E/D modules afterward. Alternatively, this can be done post-test pattern generation by adding ”don’t care” bits to the test patterns directly. A manual step is necessary because the IEEE 1838 standard provides access to DFT elements in the context of a stack and not in the context of an individual chiplet. This is why it is argued in Section V that a complete DFT implementation of a chiplet-based system should also implement an IJTAG infrastructure.

Other schemes, such as placing the E/D modules before the PTAP and after the STAP, would result in all data passing through the protected die being necessarily encrypted and decrypted. This would interfere with the tests of the other chiplets in the stack, whose test procedures would not account for the additional latency caused by the E/D modules. The same applies to schemes in which the E/D modules are implemented in an active interposer, for example.

B. Integration With IJTAG

Integration with the IJTAG standard is facilitated by the ICL and PDL, eliminating the need for the manual steps necessary for IEEE 1838-only DFT architectures. Using ICL, the designer can describe the scan path as it best suits the test requirements. Later, an ICL description of the E/D shift register modules must be added to describe the placement of the E/D modules at the boundaries of the protected scan segments. In this manner, the additional latency to write to and read from the scan chain is automatically taken into account by the Automatic Test Pattern Generation (ATPG) tool.

C. Integration on Advanced DFT Schemes

Previously, the proposed solution was illustrated using a single scan chain as an example. However, real-world designs often require more complex DFT architectures to efficiently test chiplets. Advanced DFT architectures typically include multiple scan chains and parallelized test data buses. In a 1838-compliant design, parallel test transmission is achieved through a Flexible Parallel Port (FPP) [3]. The FPP is a multi-bit data bus capable of driving multiple scan chains. Figure 6 depicts a DFT architecture consisting of an IJTAG network, an FPP, an FPP configuration register, FPP control logic, unencrypted scan chains, and a protected scan chain. In this scenario, the proposed solution can be utilized in two ways. First, E/D modules can be positioned at the boundaries of each protected scan chain, as discussed in Section VI. In this setup, different test modes to access the protected and unprotected scan chains are required, and test patterns must be generated and encrypted accordingly. Second, the proposed

countermeasure can be implemented at the boundaries of the FPP configuration register itself. This approach ensures that only entities with knowledge of the secret key can transmit data to the scan chains via the FPP.

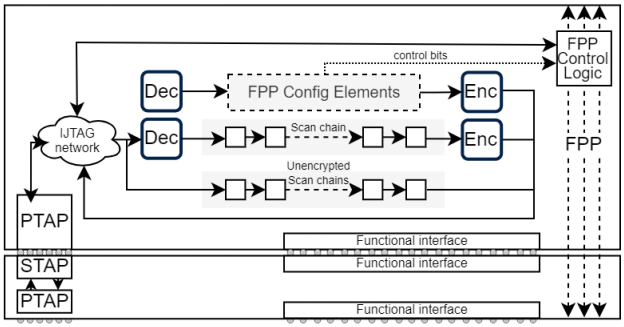

Fig. 6. The proposed countermeasure inserted in an advanced DFT architecture with a flexible parallel port.

Another crucial technique in advanced DFT architectures is the use of compressors and decompressors, which allow the ATPG tool to reduce the size of test vectors. In this scenario, the proposed countermeasure remains applicable. Figure 7 illustrates a DFT infrastructure comprising an IJTAG network, an FPP, FPP control logic, multiple scan chains, a compressor, and a decompressor. In this setup, all scan chains can be encrypted by positioning the E/D modules before the decompressor and after the compressor, as demonstrated in [22].

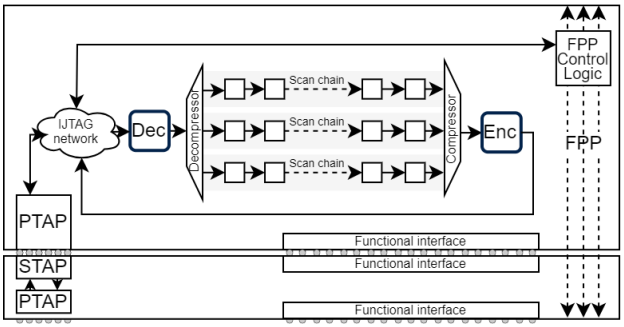

Fig. 7. The proposed countermeasure inserted in an advanced DFT architecture with compressor and decompressor.

For both scenarios mentioned above, an important consideration when implementing the proposed countermeasure is how many E/D modules are required. For example, an N-bit parallel test access could drive N different scan chains or feed an N-bit decompressor. In this case, using the same E/D module to protect every scan lane is preferable over multiplying the number of E/D modules by N. The number of scan inputs and scan outputs supported by the block ciphers is determined by the ratio between the encryption block size and the number of clock cycles required to encrypt a block. The AES-128 block cipher used in this study encrypts 128 bits over 10 clock cycles. This means that it produces 12.8 encryption bits per clock cycle, which can be used to encrypt/decrypt up to 12 data lanes. In turn, the PRESENT-80 cipher generates 64 encryption bits every 32 clock cycles, making it suitable to E/D up to 2 data lanes. Figure 8 shows the architecture of the E/D modules with support for two scan lanes.

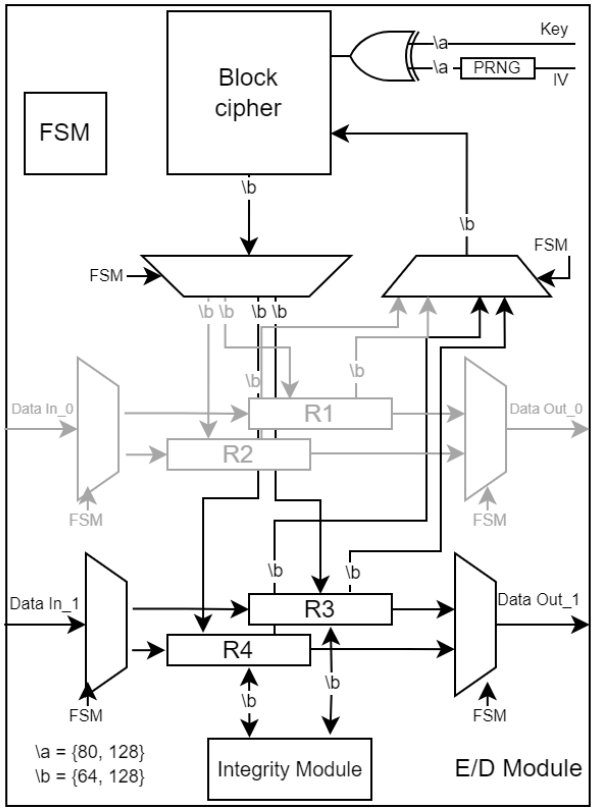

Fig. 8. Diagram of the architecture of the Encryption/Decryption modules with support for two scan lanes; Unlabeled data-paths are 1-bit wide; Data-paths labeled a are 80 or 128-bit wide and b are 64 or 128-bit wide for the PRESENT-80 or AES-128 implementations respectively; The data-path between data_in_0 and data_in_1 is grayed out and connection between R1 & R2 omitted to facilitate visualization.

D. Test Procedure

The test procedure could leverage our scheme in different ways. In this paper, we are interested in providing the platform, but we do not restrict the way it can be used. However, we briefly describe two possible test procedures.

In the first procedure, the IV scheme is disabled, and the same secret key is used for every E/D operation. In this way, the design house can provide pre-encrypted test patterns to the test facility, which does not know the secret key. The IV scheme can then be activated after testing to protect the chiplet from replay attacks in the field. This test scheme is compatible with standard test procedures. However, chiplets are not protected from replay attacks during the test.

The second possible scheme is more disruptive and requires a secure cloud interface between the ATE and the design house. This scenario is consistent with other zero-trust secure testing schemes from the literature [32]. In this scenario, the IV is transmitted to the design house, which encrypts the test patterns using a combination of the secret key and the IV. The design house then transmits the encrypted test patterns to the test facility, which does not know the key. This scheme

protects the chiplet from replay attacks as it avoids the reuse of the same encryption key more than once.

In both schemes described above, the Automated Test Equipment (ATE) can verify the test results by comparing the encrypted expected result vectors with the encrypted actual result vectors, provided there are no X (don't care bits) in the expected results. For more complex analysis of the test results, a secure cloud interface is necessary if the ATE is untrusted.

VIII. Experimental Results

A demonstrator of the proposed countermeasure was synthesized using the Synopsys Design Compiler Suite. The synthesis was performed with a 28nm FD-SOI standard-cell library. Two symmetric encryption algorithms were tested: AES-128 and PRESENT-80. The first works on blocks of 128 bits, and the second on blocks of 64 bits. The two ciphers were chosen to represent different trade-offs between cost and security. The PRNG present in each E/D module is implemented in the form of a Linear-Feedback Shift Register (LFSR). An LFSR is a shift register that uses a linear feedback function to generate a sequence of binary numbers. The register consists of a series of flip-flops connected in a feedback loop. The output of the register is determined by the feedback function. The LFSR works as a random counter, generating a pseudo-random value for each interaction. The randomness of the generated value is not required to ensure security. In fact, a simple counter would be sufficient to generate an ever-changing value that can be used to prevent the repetition of keys. However, the LFSR is a simpler circuit compared to an adder and thus causes less area overhead. The SKMU and the TRNG are out of the scope of this work since these elements are commonly present in secure devices. Therefore, their cost cannot be considered an overhead specific to the proposed countermeasure.

Area Overhead

Table I details the cost distribution between the different components of the proposed countermeasure implementing the AES cipher. Area values are from synthesis using the 28nm FD-SOI library; Gate Equivalent (GE) values are calculated by dividing the design's area by the area of the library's NAND gate (0.4352 $\mu\mathrm{m}^2$). The two E/D modules are responsible for 98% of the area of the proposed scheme. It is noticeable that the decryption module costs 62% more than the encryption module. This is expected and it happens because the decryption process in the AES cipher involves more complex operations, such as inverse operations, which require more circuitry to be implemented. The block ciphers implemented in this work are responsible for 73% and 83% of the cost of the E/D modules, respectively. The rest of the area is taken by the control FSM, registers, the integrity verification system, the PRNG, and glue logic.

Although Table I does not include the cost of the TRNG, it is important to note that it should not significantly impact the overall cost of the proposed solution. For instance, in [33], the authors implemented a ring-oscillator-based TRNG in the 28 FD-SOI technology node. Their implementation achieved a throughput of 23 Mb/s while occupying only 375 $\mu\mathrm{m}^2$.

TABLE I
Overhead of the proposed DFT architecture in terms of area

	Total		Combinational		Noncombinational	
	μm^2	GE	μm^2	GE	μm^2	GE
Decryption Module	13433	30866	7125	16372	6308	14494
Encryption Module	8287	19042	5341	12273	2946	6769
PTAP	255	586	97	223	157	361
STAP	4	9	1	2	3	7
Total	21979	50503	12564	29076	9414	21631

TABLE II
Comparison between the proposed solution against a 16-core MIPS32v1 chiplet

	Area	
	μm^2	GE
Proposed Countermeasure AES	21,979	50,503
Proposed Countermeasure PRESENT-80	1,938	4,453
16-core MIPS32v1 [34]	22,000,000	50,551,470

Next, we benchmark both versions of the proposed solution against the 16-core MIPS32v1 chiplet from [34]. It has been synthesized on the same 28nm FD-SOI technology. As shown in Table II, the security mechanism proposed in this work would represent only ˜0.1% of the total chiplet area when using the AES cipher, and ˜0.01% of the area when using the PRESENT-80 cipher. We generalize this comparison by stating that our solution would represent an overhead of 0.1% or 1% on any of more than approximately 5 million gates, when using the proposed ciphers. Notice that other block ciphers are suitable with few to none modifications on the rest of the system.

Test time overhead

The execution time for the unsecured test procedure in terms of clock cycles depends on the size of the scan chain (L) and the number of test vectors (T).

$$t_{test} = L(T+1) + T \tag{2}$$

Adding our countermeasure, the test time becomes:

$$t_{test}^{sec} = (L+N)(T+1) + T + 4b \tag{3}$$

Where N is the number of parity bits added and b is the size of the encryption block. The term $4b$ derives from the four registers in the E/D modules. As described in Equation 1, N depends on the size of the scan chain and the size of the encryption block. The overhead (%) in test time can be found by the ratio between t_{test}^{sec} and t_{test}.

A typical DFT implementation can have scan chains of thousands of SFFs and hundreds of test vectors. In this case, the terms $L(T+1)$ and $(L+N)(T+1)$ become much more

important than the terms T and $T + 4b$. Consequently, the ratio between t_{test}^{sec} and t_{test} converges to $128/127 = 1.00787$, which represents an overhead of 0.787%.

The test time overhead for the 16-core MIPS32v1 chiplet from [34], which contains scan chains of size 4068 and is tested with 1790 test vectors, would be 0.818% when using the AES cipher with a block size of 128 bits.

Additionally, we have evaluated a solution using the PRESENT-80 cipher, which utilizes blocks of 64 bits. For the same 16-core MIPS32v1 chiplet, the test time overhead was calculated to be 1.6%.

The increase in test time overhead is due to the smaller block sizes. In our scheme, one parity bit is added to each encryption block. Thus, the solution using the PRESENT-80 cipher has double the number of parity bits. The tradeoff between area, test time, and security will be discussed on section IX.

IX. Discussion

The general goal of the proposed countermeasure is to secure communication with the chiplet's DFT elements. In this paper, we have chosen to secure the communication with the scan chain as a demonstrator, which can be described as either an IEEE 1838 Data Register or an IJTAG element. Additionally, by choosing other DFT elements, we can protect any sensitive information such as secret keys or activation bitstreams transmitted during the test. First, we analyze the security aspects of the proposed solution and later the overhead in terms of area and test time.

A. Security Analysis

The threat of malicious chiplets arises naturally from the chiplet paradigm. Chiplets equipped with hidden malicious functions or running malware can sniff or modify data transmitted over the shared DFT network. The proposed solution is based on two security primitives: encryption and data integrity checking. By encrypting data, we obfuscate the information for the chiplets on the stack that do not know the secret key.

Encryption: The strength of encryption-based systems depends on several factors, including robustness to attacks that expose or circumvent the secret key, and the inherent security of the cipher itself. The latter is often measured using the k-bit security metric, which essentially measures the amount of computational effort required to break a cryptographic system. The k-bit security metric indicates that breaking the cryptographic system is as hard as performing 2^k brute-force operations. This study examines two ciphers: AES-128 and PRESENT-80. AES-128 is a state-of-the-art cipher offering 128-bit security, while PRESENT-80 is lightweight, providing 80-bit security, adequate for applications with lower security demands. However, 80-bit security is generally viewed as minimal, and AES is recommended when security is crucial.

The proposed solution addresses the threat of replay attacks, which intend to circumvent the secret key. In a chiplet-based production chain, the overproduction of commodity chiplets can become a major problem for fabless design houses. Untrusted foundries may overproduce a chiplet design and sell the chiplets on the gray market. Additionally, a chiplet can be produced with multiple functional modes that are sold to the 3DIC integrator upon demand. Therefore, off-the-shelf chiplets may require a logic locking key or a secret configuration bitstream during post-bond testing. An attacker in the test facility or within the 3DIC can intercept the encrypted data and use it to unlock features on other chiplets without authorization and knowledge of the secret key. Our solution prevents this type of attack by dynamically changing the E/D keys.

Integrity: As presented in Section VI, the chance of crafting a malicious ciphertext that bypasses the integrity check mechanism is 2^{-N}, where N is the number of parity bits added. However, an attacker could, in theory, tamper with only one 128-bit block. In this case, there is a 50% chance that the attacker could write 127 bits of random data to the scan chain. This capability is very limited and has a negligible chance of resulting in a successful attack. Brute-force attacks typically require a large number of attempts, and the probability of triggering the integrity system grows exponentially with the number of tampered blocks.

Equation 1 defines how many parity bits are added to the test patterns. This quantity depends only on the size of the test patterns and the size of the E/D block. Thus, the two ciphers explored in this work provide different levels of integrity robustness due to the different block sizes. In fact, the less secure PRESENT-80 provides twice the number of parity bits, making the integrity system significantly more resistant to brute-force attacks. However, the integrity mechanism itself is not subject to brute-force attacks because this type of attack depends on being able to produce multiple failed attempts, which is exactly what the module intends to prevent. Thus, during testing, an attacker would have only one attempt to craft a compliant malicious test vector.

Outside the test environment, a brute-force attack on the integrity checking system would be feasible in principle, as the attacker would be able to restart the test ignoring the security flag. However, for designs containing a non-volatile memory (NVM), the security flag could be stored in the memory and checked at startup.

The integrity check also mitigates scan chain insertion threats, such as scan-based and SAT attacks, which rely on reading and writing scan chain information. By undermining this ability, scan attacks are prevented by design. SAT attacks remain theoretically possible, but without flip-flop control, solving SAT equations becomes significantly harder [21]. 3DICs inherit these threats from 2D SoCs, but chiplets introduce new attack vectors. Attackers might exploit less secure chiplets to attack others within the 3DIC stack.

Next, we justify the decision of using a checksum (parity bit) as the encoding mechanism for integrity checking instead of more complex methods like MACs, cryptographic hashes, or authenticated cryptography in general. This decision is justified on several grounds. Firstly, these solutions are costly to implement. For instance, AES-GCM-128 offers state-of-the-art message integrity but incurs a significantly higher area cost. When synthesized under the same conditions as the proposed countermeasure, the AES-GCM-128 implementation available on [35] takes approximately 54000 $\mu\mathrm{m}^2$ for the encryption

and decryption blocks. It is a cost of approximately 2.4 times that of the AES version and 28 times that of the PRESENT-80 version of our countermeasure. In the context of secure testing for chiplets, such robustness is unnecessary. Scan encryption ensures that unauthorized users cannot write or read meaningful data to the scan chain. Therefore, by leveraging the simplicity and efficiency of checksums, we can add a layer of security by making message tampering detectable without incurring the overhead associated with more complex cryptographic operations. Moreover, while using encryption and checksums for integrity is not a conventional approach, it has been shown to be both efficient and secure for simple symmetric encryption, as evidenced by the findings in [36].

Finally, the system's reliability depends on encryption method robustness. Testing block ciphers raises concerns about scan chain attacks revealing secret keys. Inserting a scan chain into ciphers creates vulnerabilities our system aims to solve. However, cryptographic algorithms' diffusion properties ensure faults at any cipher stage cause noticeable E/D operation errors [37]. Thus, E/D blocks are tested by performing encryption, followed by decryption, and comparing plaintexts.

B. Overhead

Area: As presented in Section VIII, the proposed solution incurs minimal area overhead. The PRESENT-80 cipher is particularly advantageous, occupying one-tenth the area of AES. However, AES remains a compelling choice due to its superior security at a reasonable cost. We suggest opting for less secure options only when the alternative is no security at all.

Comparing our solutions with existing literature referenced in Section IV, we find that scan encryption countermeasures typically result in similar overheads, as the cipher largely determines area cost (see Table I). Few lightweight ciphers exist, and those using Trivium, PRESENT, and SKINNY often require only a few thousand GE for cryptographic operations [22], [24]–[27]. Thus, our countermeasure aligns with state-of-the-art scan encryption solutions concerning E/D functionality.

The countermeasure proposed in this work stands out from the state-of-the-art when considering the novelty of implementing integrity check using a lightweight encoding mechanism. For reference, [23] implements integrity-robust scan encryption by combining the Trivium cipher with a MAC scheme. The authors reported an overhead from 56k to 114k GE, where we can assume that the majority is due to the MAC implementation. In contrast, the parity bit scheme proposed in this work represents a negligible cost in terms of area.

Test time: As shown in section VIII, the test time overhead is defined by the ratio between t_{test}^{sec} and t_{test}. In turn, t_{test}^{sec} differs from t_{test} by the inclusion of the terms N and $4b$. Here, N represents the number of parity bits added to the test patterns, and b is the size of the buffering registers. Looking at equation 3, we can see that the overhead mainly depends on the number of parity bits added. N, in turn, depends on the size of the scan chain—assumed to be fixed—and the size of the encryption block (see Equation 1). As pointed out in this section, the number of parity bits added affects the robustness of the integrity verification scheme. Therefore, for a given value of N—defined by the security requirement of a system—and the size of the scan chain—defined by the test coverage—one may choose between a 64 or 128-bit cipher. In the results presented in this work, for a real-world DFT architecture, the 64-bit cipher caused a 1.6% overhead, while the 128-bit cipher caused a 0.8% overhead, which is a twofold increase. We recommend the approach of using smaller encryption blocks only for small scan chains to ensure a minimum number of parity bits, as discussed previously in this section.

C. Future Work

Lastly, we discuss the possibility of adapting the countermeasure presented in this work to enable other security features in the absence of standards other than the IEEE 1838 that enable plug-and-play integration of chiplets. A good example of this potential is the enrollment of unlocking keys for obfuscated or feature-locked designs. In this type of scheme, locking gates are scattered across the design where one input is a functional wire of the design and the other comes from a key register. In the absence of the key, the functional wire is locked to logic zero. The register key can be easily implemented using an IEEE 1838 Data Register or an IJTAG element. In this case, our solution could be used to obfuscate the unlocking vector from the tester, unauthorized users, or untrusted chiplets on the stack. The same applies for loading confidential metadata into any type of secure memory that can be interfaced with the DFT serial path.

X. Conclusion

The transition to chiplet-based systems in semiconductor manufacturing presents both opportunities and security challenges. The DFT infrastructure, while crucial for testing and debugging, introduces potential vulnerabilities that must be addressed. Our proposed hardware countermeasure effectively secures the DFT communication in chiplet-based systems by combining encryption and message encoding for integrity verification. The scan encryption method ensures that only those knowledgeable of the secret key can write meaningful data on the scan chain. The integrity verification system ensures that messages from unauthorized senders are detected and do not enter the scan chain. The experimental results demonstrate that our solution provides robust security while remaining lightweight in terms of area and test time overhead. By ensuring the integrity and confidentiality of test data, our approach enhances the security of chiplet-based systems, paving the way for their broader adoption in the industry. Additionally, we argue that the proposed countermeasure can be used as Root of Trust for other security features integrated directly on the DFT infrastructure.

References

[1] F. Sheikh, R. Nagisetty, T. Karnik, and D. Kehlet, "2.5 d and 3d heterogeneous integration: emerging applications," *IEEE Solid-State Circuits Magazine*, vol. 13, no. 4, pp. 77–87, 2021.

[2] "IEEE standard for access and control of instrumentation embedded within a semiconductor device," *IEEE Std 1687-2014*, pp. 1–283, 2014.

[3] "IEEE Standard for Test Access Architecture for Three-Dimensional Stacked Integrated Circuits," *IEEE Std 1838-2019*, pp. 1–73, Mar. 2020.

[4] J. Suzano, F. Abouzeid, P. Anthony, G. Di Natale, and P. Roche, "On hardware security and trust for chiplet-based 2.5 D and 3D ICs: Challenges and Innovations," *IEEE Access*, 2024.

[5] B. Yang, K. Wu, and R. Karri, "Scan based side channel attack on dedicated hardware implementations of Data Encryption Standard," in *2004 International Conference on Test*, pp. 339–344, Oct. 2004.

[6] J. Da Rolt, A. Das, G. Di Natale, M.-L. Flottes, B. Rouzeyre, and I. Verbauwhede, "Test Versus Security: Past and Present," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, pp. 50–62, Mar. 2014.

[7] M. Da Silva, M.-L. Flottes, G. Di Natale, and B. Rouzeyre, "Preventing Scan Attacks on Secure Circuits Through Scan Chain Encryption," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, pp. 538–550, mar 2019.

[8] J. Suzano, A. Chastand, E. Valea, G. Di Natale, A. Philippe, F. Abouzeid, and P. Roche, "IEEE 1838 compliant scan encryption and integrity for 2.5/3D ICs," in *IEEE European Test Symposium* (IEEE, ed.), (La Haye, Netherlands), May 2024.

[9] B. Vinnakota, "Building An Open Chiplet Economy." https://www.opencompute.org/blog/building-an-open-chiplet-economy.

[10] "Universal chiplet interconnect express (ucie) website." https://www.uciexpress.org/. [Accessed 10-02-2025].

[11] V. Hiremath, "Introduction to Tessent Multi-Die - EDA Support Blogs — blogs.sw.siemens.com." https://blogs.sw.siemens.com/eda-support/2024/06/01/introduction-to-tessent-multi-die/, 2024. [Accessed 10-02-2025].

[12] E. Valea, M. da Silva, G. Di Natale, M.-L. Flottes, and B. Rouzeyre, "A Survey on Security Threats and Countermeasures in IEEE Test Standards," *IEEE Design & Test*, vol. 36, pp. 95–116, June 2019.

[13] J. DaRolt, G. Di Natale, M.-L. Flottes, and B. Rouzeyre, "Scan Attacks and Countermeasures in Presence of Scan Response Compactors," in *2011 Sixteenth IEEE European Test Symposium*, pp. 19–24, May 2011.

[14] X. Li, W. Li, J. Ye, H. Li, and Y. Hu, "Scan Chain Based Attacks and Countermeasures: A Survey," *IEEE Access*, vol. 7, pp. 85055–85065, 2019.

[15] L.-T. L. T. . Wang, X. Wen, and K. S. Abdel-Hafez, "Chapter 2 - Design for Testability," in *VLSI Test Principles and Architectures* (L.-T. Wang, C.-W. Wu, and X. Wen, eds.), pp. 37–103, San Francisco: Morgan Kaufmann, Jan. 2006.

[16] D. G. Saab, V. Nagubadi, F. Kocan, and J. Abraham, "Extraction based verification method for off the shelf integrated circuits," in *2009 1st Asia Symposium on Quality Electronic Design*, pp. 396–400, July 2009.

[17] L. Azriel, R. Ginosar, S. Gueron, and A. Mendelson, "Using Scan Side Channel to Detect IP Theft," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, pp. 3268–3280, Dec. 2017.

[18] L. Azriel, R. Ginosar, and A. Mendelson, "Revealing On-chip Proprietary Security Functions with Scan Side Channel Based Reverse Engineering," in *Proceedings of the Great Lakes Symposium on VLSI 2017*, GLSVLSI '17, (New York, NY, USA), pp. 233–238, Association for Computing Machinery, May 2017.

[19] L. Azriel, J. Speith, N. Albartus, R. Ginosar, A. Mendelson, and C. Paar, "A survey of algorithmic methods in IC reverse engineering," *Journal of Cryptographic Engineering*, vol. 11, pp. 299–315, Sept. 2021.

[20] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 137–143, 2015.

[21] L. Alrahis, M. Yasin, N. Limaye, H. Saleh, B. Mohammad, M. Al-Qutayri, and O. Sinanoglu, "ScanSAT: Unlocking Static and Dynamic Scan Obfuscation," Sept. 2019. arXiv:1909.04428 [cs].

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

[23] K. Rosenfeld and R. Karri, "Attacks and Defenses for JTAG," *IEEE Design & Test of Computers*, vol. 27, pp. 36–47, Jan. 2010.

[24] E. Valea, M. D. Silva, M.-L. Flottes, G. D. Natale, and B. Rouzeyre, "Encryption-Based Secure JTAG," in *2019 IEEE 22nd International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, pp. 1–6, Apr. 2019.

[25] K. Rosenfeld and R. Karri, "Security-aware SoC test access mechanisms," in *29th VLSI Test Symposium*, pp. 100–104, May 2011.

[26] B. Thiemann, L. Feiten, P. Raiola, B. Becker, and M. Sauer, "On Integrating Lightweight Encryption in Reconfigurable Scan Networks," in *2019 IEEE European Test Symposium (ETS)*, pp. 1–6, May 2019.

[27] M. Da Silva, M.-L. Flottes, G. Di Natale, and B. Rouzeyre, "Preventing Scan Attacks on Secure Circuits Through Scan Chain Encryption," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, pp. 538–550, Mar. 2019.

[28] R. Elnaggar, R. Karri, and K. Chakrabarty, "Securing IJTAG against data-integrity attacks," in *2018 IEEE 36th VLSI Test Symposium (VTS)*, pp. 1–6, Apr. 2018.

[29] R. Elnaggar, R. Karri, and K. Chakrabarty, "Security Against Data-Sniffing and Alteration Attacks in IJTAG," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, pp. 1301–1314, July 2021.

[30] S.-J. Wang, Y.-C. Shih, K. S.-M. Li, C.-Y. Lin, and S.-K. Chong, "Improving IJTAG Test Efficiency and Security," in *2022 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, pp. 1–4, Apr. 2022.

[31] J.-F. Côté, J. Fan, S. Shen, G. Danialy, M. Lipinski, M. Garbers, W. Yang, M. Keim, A. Glowatz, and J. Reynick, "Affordable and Comprehensive Testing of 3-D Stacked Die Devices," *IEEE Design & Test*, vol. 39, no. 5, pp. 17–25, 2022.

[32] P. Slpsk, S. Ray, and S. Bhunia, "Treehouse: A secure asset management infrastructure for protecting 3dic designs," *IEEE Transactions on Computers*, 2023.

[33] K. Yang, D. Fick, M. B. Henry, Y. Lee, D. Blaauw, and D. Sylvester, "16.3 A 23Mb/s 23pJ/b fully synthesized true-random-number generator in 28nm and 65nm CMOS," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 280–281, Feb. 2014.

[34] P. Vivet, E. Guthmuller, Y. Thonnart, G. Pillonnet, C. Fuguet, I. Miro-Panades, G. Moritz, J. Durupt, C. Bernard, D. Varreau, *et al.*, "Intact: A 96-core processor with six chiplets 3d-stacked on an active interposer with distributed interconnects and integrated power management," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 1, pp. 79–97, 2020.

[35] Luca, *BLu85/AES-GCM-128-192-256-bits*. 7 2024.

[36] B. Lampson, M. Abadi, M. Burrows, and E. Wobber, "Authentication in distributed systems: Theory and practice," *ACM Transactions on Computer Systems (TOCS)*, vol. 10, no. 4, pp. 265–310, 1992.

[37] G. D. Natale, M. Doulcier, M.-L. Flottes, and B. Rouzeyre, "Self-test techniques for crypto-devices," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 2, pp. 329–333, 2010.

Suzano Juan received his B.S in computer engineering from the Universidade Federal do Rio Grande do Sul and M.S from the École Supérieure de Chímie Physique Électonique de Lyon through the excellence double degree program BRAFITEC in 2022. In the same year, he started to pursue his Ph.D. in micro and nano electronics at the Université Grenoble Alpes in partnership with STMicroelectronics and the Commissariat A L'Energie Atomique Et Aux Energies Alternatives (CEA).

During his graduation, he worked on research projects on hardware reliability in harsh environments for aerospace applications and participated as a co-author in two conference papers. Now, he is invested in the research and development of security solutions for chiplet-based 2.5D and 3D integrated circuits.

ABOUZEID Fady received the M.S. (2007) and Ph.D. (2010) in Micro and Nano Electronics from Grenoble University, France. Since 2007 he has been with STMicroelectronics, Central R&D, Crolles, France in a research and development group in charge of hardening and qualifying IPs for space and terrestrial environments, and ultra-low voltage and high energy efficiency circuit design. He was in charge of the design activities, enabling research and implementation of CPU embedded hardening and low power mechanisms, advance embedded radiation effects capture systems, and test vehicles for radiation qualification. Since 2020, his research activities are now focused on the enablement of 2.5D / 3D Chiplets solutions, and autonomous microcontrollers.

Di Natale Giorgio received the PhD in Computer Engineering from the Politecnico di Torino in 2003. He works as Director of Research for the French National Research Center (CNRS), and he is the director of the TIMA laboratory in Grenoble since January 2021. His research interests include hardware security and trust, secure circuits design and test, reliability evaluation and fault tolerance, and VLSI testing. He has published 2 books and 9 book chapters, 50 journal papers, and more than 150 conference and symposium papers in these domains. He has been involved in projects funded by the EU, Italy and France. He has been the action chair of the COST Action TRUDEVICE (Trustworthy Manufacturing and Utilization of Secure Devices), the biggest European research network on hardware security and trust.

He also actively contributed in the organization of the main international conferences in his domain (general chair of DATE20, program chair of DATE17, program chair of ETS16, member of the executive committee of DATE since 2012, member of organizing committee of ETS and VTS since 2010). He belongs to the program committees of many conferences (DATE, ETS, IOLTS, DSD, DTIS, FDTC, GLSVLSI, HOST, CS2) and he serves as associate editor for IEEE Transactions on CAD and IEEE Transactions on Computers.

He served as chair of the IEEE Computer Society TTTC, he is Golden Core member of the Computer Society and Senior member of the IEEE.

PHILIPPE Anthony is graduated from Central-Supelec engineer school in France. After 17 years in STMicrolectronics as system architect defining complex system on chip in different technology nodes including 28FDSOI. Anthony joined the CEA as research engineer in December 2014. He is currently involved in system-on-chip architecture for High performance IP, computing and 3D Integrated Circuit projects. He has been the lead architect of several designs all combining low power, high speed communication links, many core architecture and Network-on-Chip. He strongly participates to the roadmap, definition and specification of the next generation of communication and computing components and actively participated to the architecture definition of the computing silicon developed within the H2020 FETHPC ExaNoDe project.

Roche Philippe received the M.S. (1995) and Ph.D. (1999) in semiconductor physics. His primary activities are Single Event Effects and Total Ionizing Dose, as well as Ultra Low Voltage IPs, from sub-0.25µm technologies down to FinFET 3nm. He has been serving in conferences since 1997 as session chair and short course instructor. Philippe has coauthored +300 papers and filed +75 patents and 3 trade marks in radiation hardening. He was appointed Regional Fellow and Technical Director R&D in 2013, then elected by the ST Technology Advisory Board as Corporate Fellow in 2020. After 5 years in a product organization designing ASICs, Philippe is back to ST Central R&D (FTM/TDP), in charge of new R&D explorations such as 3D/GaN/Safety/RF/batteries/... with a team of senior experts, also acting as Head of Labs & Ecosystems management, with LETI, CNRS, ANRT and CIME-P as key partners.