IEEE 1838 compliant scan encryption and integrity for 2.5/3D ICs

Juan Suzano, Antoine Chastand, Emanuele Valea, Giorgio Di Natale, Anthony Philippe, Fady Abouzeid, Philippe Roche

IEEE 1838 compliant scan encryption and integrity for 2.5/3D ICs

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

***Abstract*—2.5D and 3D integrated circuits (IC) are the natural evolution of traditional 2D SoCs. 2.5D and 3D integration is the process of assembling pre-manufactured chiplets in an interposer or in a stack. This process can damage the chiplets or lead to faulty connections. Thus, the importance of post-bond test of chiplets. The IEEE Std 1838(TM)-2019 (IEEE 1838) design-for-testability (DFT) standard defines mandatory and optional structures for accessing DFT functions on the chiplet. Compliant chiplets form a DFT network that can be exploited by attackers to violate the confidentiality or integrity of the message transmitted over the serial path. In this work, we combine a message integrity verification system with a scan encryption mechanism to protect the scan chain of an IEEE 1838-compliant DFT implementation. The scan encryption prevents unauthorized actors from writing meaningful data into the scan chain. Message integrity verification makes messages from untrustworthy sources detectable. In conjunction, both security primitives protect the scan chain from malicious chiplets on the stack, scan-based attacks, and brute force attacks. 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*—3DIC, Chiplets, Design for Testability (DFT), Hardware Security, Root of Trust**

I. Introduction

3DICs[1] are the natural evolution of traditional 2D SoC [1]. Historically, SoCs have had only one layer of transistors. 3DICs however expand vertically integrating multiple layers of computing logic in the same package. This is achieved by stacking multiple pre-manufactured dies, called chiplets. Individual dies must be tested before assembly to ensure only Know-Good-Dies are stacked. Additionally, the chiplets must also be tested post-bond to detect defects caused by the manufacturing process. The post-bond test of chiplets is made more difficult by the fact that designers do not know in advance the stack architecture the chiplets integrate and how to access the design-for-testability (DFT) structures of the chiplet. The IEEE Std 1838(TM)-2019 (IEEE 1838) DFT standard [2] defines mandatory and optional DFT infrastructure for pre- and post-bond tests of chiplets. IEEE 1838-compliant chiplets have the infrastructure necessary for individual pre-bond tests. Moreover, when stacked, chiplets form a DFT network that allows testing data to seamlessly circulate through the stack.

The cohesive DFT network formed by IEEE 1838-compliant dies introduces a risk to the confidentiality of test data. As the tester and chiplet transmit data through the DFT network, this is exposed to the other chiplets in the stack. Untrusted chiplets can spy and sabotage the communication of test patterns, test responses, cryptographic keys, and activation bitstreams. This highlights the lack of a communication root of trust (RoT) mechanism for post-stacked chiplets. 3DICs also inherit vulnerabilities from 2D SoC in regard to the DFT structures. Many works have demonstrated the use of the scan chain to leak secret information, such as cryptographic keys, for instance [3]. The assumption that more complex DFT architectures can stop attackers has also been disproven [4]. 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 [5]. However, scan encryption techniques do not prevent an attacker from writing random data on the internal flip-flops of the device, which is sufficient to mount an attack. In this work, we combine an integrity checking mechanism with a scan encryption technique to build a communication RoT. The data transmitted over the scan chain is encoded before encryption in such a way that messages from unauthorized sources can be easily detected. In this way, we protect communications over the DFT network from espionage, sabotage and scan-based attacks. Our solution is integrated into an IEEE 1838 DFT implementation, as we understand that a zero-trust approach to chiplets development is necessary to enable secure and trustworthy chiplet-based 3DICs.

The remainder of this paper is organized as follows. Section II presents the IEEE 1838 DFT standard. Section III presents the threats model that our countermeasure intends to protect against. Section IV presents the proposed countermeasure. Section V presents the experimental evaluation of area and test time overhead. Section VI presents the security evaluation. Finally, Section VII presents our conclusion.

II. The IEEE 1838 Standard

In the production of 3DICs, pre-fabricated chiplets are stacked in a complex process that can lead to damage to the

[*]Institute of Engineering Univ. Grenoble Alpes

[1]In this work we use the umbrella term "3DIC" to refer to every type of 2.5D and 3D IC.

chiplets 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 is to be assembled. For various reasons, it is not possible for each chiplet in a 3DIC to have its own access to the automatic test equipment (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, i.e. the DFT features are added individually on each die and not on the stack. However, when compliant dies are stacked, they form a comprehensive DFT architecture that enables stack-level testing. In addition, the IEEE 1838 does not require any assembly scheme and supports the 2.5D, 3D, and 5.5D [6].

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 IEEE 1149.1 standard [7] and consists of five terminals: TCK, TMS, TDI, TDO, TRSTN.

The PTAP signals drive the PTAP Controller, which is an IEEE 1149.1 compatible 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. It enables 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 on 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.

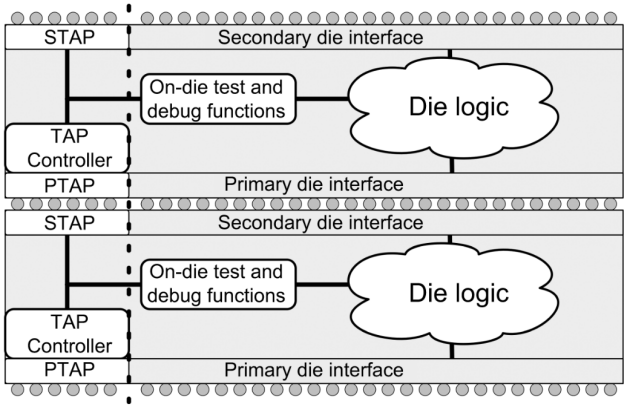

Fig. 1. Generic representation of an IEEE 1838 DFT architecture for a two-die 3DIC [2].

III. Threat and Attacker Model

The industrial transition from 2D SoCs to chiplet-based 3DICs creates new vulnerabilities in addition to those inherited from traditional 2D SoCs. The use of off-the-shelf chiplets provides an entry point for attackers, as malicious chiplets can pollute the market. A malicious chiplet is a chiplet that contains undisclosed logic and can act as an attacker within the stack [8]. In an IEEE 1838 DFT architecture, the chiplets share the same DFT network. Therefore, the data transmitted over the DFT data path is vulnerable to the misbehavior of a malicious chiplet. The chiplets can passively spy on or modify the data sent to other chiplets through the DFT network.

Several attacks targeting DFT structures have been proposed in literature [9]. 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 [10]. Nevertheless, an attacker can exploit the scan chain to leak secret information or bypass security mechanisms. In a scan-based attack, the attacker can use the scan chain to steal security-critical assets, such as the secret keys of cryptographic algorithms. Attacks against logic-locked designs are more effective when the scan chain is accessible [11]. In particular, Boolean satisfiability (SAT) based attacks can be used to extract the logic locking secret key, exploiting the oracle offered by having scan chain access on the target IC [12]. 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.

Scan encryption has been proposed as a solution to prevent scan-based attacks by undermining the attacker's ability to write and read meaningful data in the scan chain [13]. Testing a circuit protected by scan encryption requires dealing with encrypted test data. A secret key, that is safely stored inside the device, must be used by the tester to encrypt test data. These data are then decrypted inside the circuit and test responses are encrypted before returning the results back to the tester. This technique provides confidentiality of test data outside the boundaries of the device (or chiplet) under test. However, scan encryption techniques do not prevent the attacker from writing random bits to the SFFs. This is a security weakness as it enables brute-force attacks in which the attacker simply tries in a structured way all possible combinations, until he finds the secret key. The attacker may only need to guess a small set of bits that are responsible for putting the device into an exploitable state [14]. Therefore, it is necessary to ensure the integrity of the message transmitted on the scan chain, i.e., to

ensure that the message originates from a sender who knows the secret key. Moreover, scan encryption is also vulnerable against replay attacks, i.e., a type of attack based on the fact that the encryption operation for a given key always results in the same plaintext/ciphertext pair. Therefore, even if the message is encrypted and the adversary cannot convey the meaning of the message, he can still reuse it to reproduce its purpose.

In this paper, we focus on the protection of data communication performed via the scan chain. Our threat model assumes that the attacker can be in the test facility, in the field, or inside the 3DIC in the form of an untrusted chiplet. We assume that the ability to read meaningful data transmitted via the scan chain or the ability to write random data to the scan chain is sufficient to mount an attack.

IV. Scan Encryption With Integrity Check

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 any data on the scan chain. Blocking physical access to the DFT I/O could prevent on-field tampering at the expense of preventing in-field debugging and configuration. However, attacks at the test facility would remain possible. Additionally, the threat of untrusted chiplets on the 3D stack would remain unaddressed.

The countermeasure proposed in this work takes the path of securing the communication with the protected chiplet through a combination of encryption and encoding. The security principle is that any data transmitted to the chiplet on the serial path is encoded with a public encoding algorithm and encrypted. By doing so, we grant that only the entities having 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.

Architecture overview

The proposed scheme is shown in Figure 2. The user treats the test data off-chip using the encoding and encryption techniques. 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) is performed on the die by two symmetric ciphers. The decryption module is inserted at the input of the serial path for decrypting the test data. Likewise, 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, together with the scheme for communicating the key between the chiplet and the test equipment, or administrator.

A true random number generator (TNRG) needs to be used to generate an initial value (IV). Each of the E/D modules include 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 would produce random bits as output. However, the IV value must be known by the tester for correctly deriving the encryption key. Thus, a scheme for communicating this value must be implemented. A simple PTAP Control instruction that puts the TNRG 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.

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, 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 it 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 [15]. 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 transmits the encrypted test patterns to the test facility, which does not know the key. This scheme protects the chiplet from replay attacks as we avoid the reuse of the same encryption key more than once.

Integration with IEEE 1838

Our solution is intended to be an add-on to the IEEE 1838 test infrastructure 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 data registers. When an unprotected data register is put on the TDI-TDO serial path by the instruction stored in the instruction register, system operation will not be affected by our design. This approach allows the implementation of multiple protected and unprotected data registers. That way, the only disruption caused by our design is that when writing and reading in a protected data register, the additional delay caused by the block ciphers must be taken into account. Other schemes, such as placing the E/D modules before the PTAP and after the STAP, would result in all data passing 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 take into account 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.

Encryption and Decryption Modules

The E&D modules are presented in Figure 3. They implement a symmetric block cipher. As the DFT works in a serial manner and we are using block ciphers, some control logic is needed. Two registers (R1 and R2) are implemented for

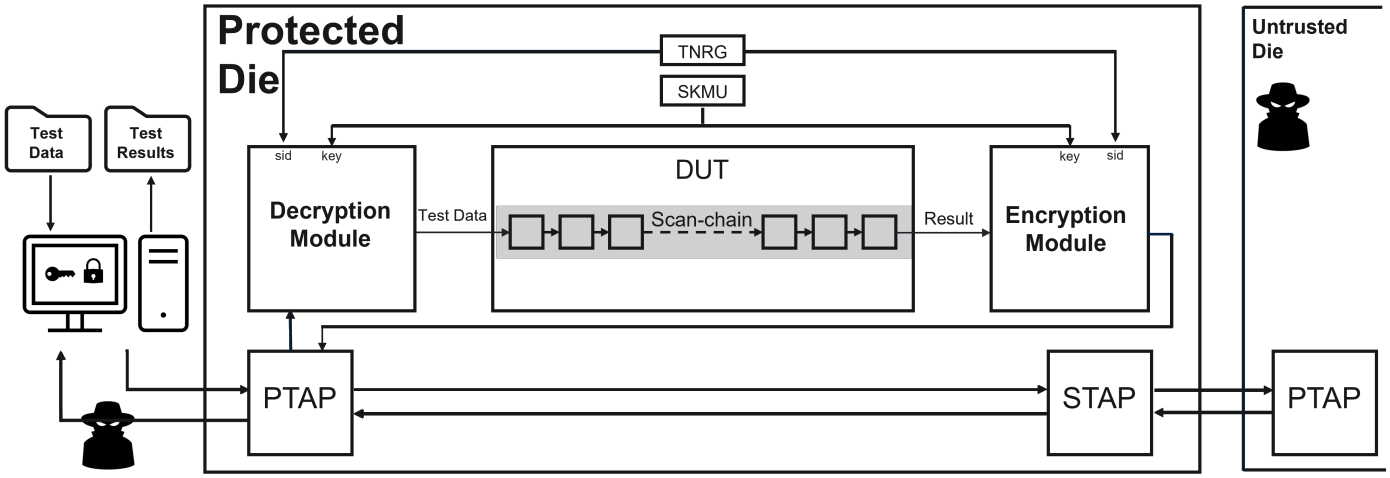

Fig. 2. Architecture overview for a protected die with the proposed scan encryption technique, stacked with other(s) untrusted die(s), in a test environment.

buffering. They allow the system to receive the data being shifted through the serial path and load data into/from the cipher in parallel.

The control is performed by a Finite State Machine (FSM). This is accomplished through a lightweight parity bit verification. The basic idea consists of encoding, before the encryption, the plaintext message with a parity bit. Registers R1 and R2 of the decryption module are connected to the integrity module. After the decryption operation, the parity bit is checked and the test is aborted if the message does not meet the integrity requirement, i.e., 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 that produces a parity bit that can be used to ensure the integrity of the test responses.

A Pseudo Random Number Generator (PRNG) is inserted on the IV signal, in order to produce a different encryption key at each encryption block. It receives the input IV from the TRNG. The output of the PRNG is XORed with the secret key received from the SKMU. This results in a different E/D key for each encryption block.

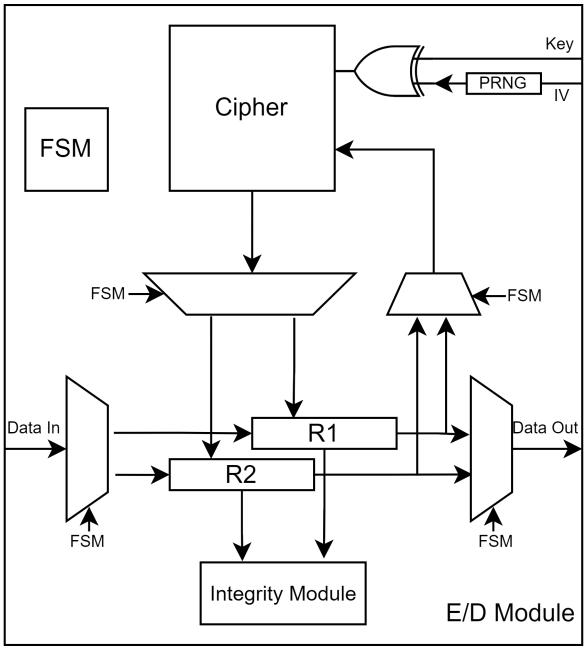

Fig. 3. Diagram of the architecture of the Encryption/Decryption modules.

Integrity Check

The basic idea of the integrity check scheme consists of encoding the plaintext with a publicly known encoding algorithm. The message 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 device. 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.

We have decided to use a parity algorithm as the encoding method. The parity bit is encoded in the test patterns before encryption. In the implemented scheme, the 128th bit of each 128-bit 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}.

In our scheme, for every 128-bit encryption block, the 128th bit is a parity bit. It is computed by XORing the other 127 bits.

The parity bits are computed off-chip during the encryption of the message. For every 128-bit on-chip decryption round, the decryption block calculates and compares the parity bit to ensure the integrity of the data. If the checking fails, the system can interrupt the testing process to avoid 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 the 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.

V. Experimental Results

A demonstrator of the proposed countermeasure was synthesized using Synopses Design Compiler Suite. The synthesis has been performed on a 28nm FD-SOI standard-cell library. The symmetric encryption algorithm used is the AES, thus the E/D modules work with blocks of 128 bits. 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 that are 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 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 as an overhead specific to the proposed countermeasure.

Area Overhead

Table I presents the cost of our countermeasure in terms of silicon area. 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 μm^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.

Next, we benchmark our solution against the 16-core MIPS32v1 chiplet from [16]. 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. We generalize this comparison by stating that our solution would represent an overhead of less than 1% on any design composed of more than approximately 5 million gates.

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	21,979	50,503
16-core MIPS32v1 [16]	22,000,000	50,551,470

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. N is the number of parity bits added. 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 [16] which contains scan chains of size 4068 and is tested with 1790 test vectors would be 0.818%.

VI. Security Analysis

The general goal of the proposed countermeasure is to secure the communication with the chiplet over the IEEE 1838 serial data path. In this paper, we have chosen to secure the communication with the scan chain, which is one of the IEEE 1838 data registers. Thereby, we secure the scan test process of chiplets that implement the IEEE 1838 standard. In addition, by choosing other data registers, we can protect any sensitive information such as secret keys or activation

bitstreams transmitted during the test. Our solution is based on two security primitives: encryption and data integrity checking.

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. By encrypting data, we obfuscate the information for the chiplets on the stack that do not know the secret key. By encoding the information with a publicly known algorithm prior to encryption, we make data tampering easily detectable. In Section IV, we showed that an attacker is unable to generate ciphertext that conforms to the implemented integrity mechanism after decryption. A natural concern is that the integrity system acts as an oracle for the attacker, facilitating brute-force attacks. As far as we know, there is no work in the literature that uses the parity bit method to perform such attacks on block ciphers. However, to prevent this threat, it is sufficient to hide the failed check from the attacker and mask the bitstream with random bits or zeros.

Our solution also addresses the threat of replay attacks. 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. Also, 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. Our solution prevents this type of attack by dynamically changing the E/D keys.

Our solution mitigates two threats associated with scan chain insertion, namely scan-based attacks and SAT attacks. Both attacks rely on the ability to read and write information in the scan chain. By undermining this ability, we prevent scan attacks by design. SAT attacks are still theoretically possible, but without control over the internal flip-flops, the mathematical task of solving the SAT equations becomes significantly more difficult [12]. Although 3DICs inherit these threats from 2D SoCs, chiplets introduce new points of entry for attacks. Attackers can use less secure chiplets to carry out these attacks on the other chiplets on the stack from within the 3DIC.

The reliability of the proposed system depends on the robustness of the implemented encryption method. A natural concern would be about how to test the block ciphers. In Section III we discussed the use of scan chain attacks to obtain the secret keys of cryptographic systems. To insert a scan chain on block ciphers would create the same vulnerability that our system is trying to solve. However, it has been shown that the diffusion properties of cryptographic algorithms mean that a fault at any stage of the cipher will create a noticeable error in the E/D operation [17]. Therefore, the E/D blocks are tested for free by performing an encryption, followed by a decryption and comparing the plaintexts.

VII. Conclusion

In this work, we paired message integrity verification and scan encryption to secure the scan chain of an IEEE 1838 DFT implementation. 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. Although we applied the countermeasure to protect the scan chain, our solution can be applied to protect any other sensible data register of the IEEE 1838 standard. We showed that our solution would represent a negligible overhead in terms of the area when compared with state-of-the-art chiplets. The test time overhead is kept at less than 1% for typical DFT configurations.

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, "Ieee standard for test access architecture for three-dimensional stacked integrated circuits," *IEEE Std 1838-2019*, pp. 1–73, 2020.

[3] 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.

[4] J. D. Rolt, G. D. Natale, M.-L. Flottes, and B. Rouzeyre, "A novel differential scan attack on advanced DFT structures," *ACM Transactions on Design Automation of Electronic Systems*, vol. 18, pp. 58:1–58:22, Oct. 2013.

[5] 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.

[6] E. J. Marinissen, "Challenges and emerging solutions in testing tsv-based 2 1 over 2d- and 3d-stacked ics," in *2012 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 1277–1282, 2012.

[7] "Ieee standard for test access port and boundary-scan architecture," *IEEE Std 1149.1-2013 (Revision of IEEE Std 1149.1-2001)*, pp. 1–444, 2013.

[8] M. Nabeel, M. Ashraf, S. Patnaik, V. Soteriou, O. Sinanoglu, and J. Knechtel, "2.5 d root of trust: Secure system-level integration of untrusted chiplets," *IEEE Transactions on Computers*, vol. 69, no. 11, pp. 1611–1625, 2020.

[9] 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, no. 1, pp. 50–62, 2014.

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

[11] 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.

[12] 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].

[13] 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.

[14] F. Strenzke, "An analysis of openssl's random number generator," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 644–669, Springer, 2016.

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

[16] 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.

[17] 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.