IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 February 2024, accepted 4 March 2024, date of publication 14 March 2024, date of current version 5 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3375874

RESEARCH ARTICLE

Advancing Trustworthiness in System-in-Package: A Novel Root-of-Trust Hardware Security Module for Heterogeneous Integration

MD SAMI UL ISLAM SAMI, TAO ZHANG, AMIT MAZUMDER SHUVO, MD SAAD UL HAQUE, PAUL E. CALZADA, KIMIA ZAMIRI AZAR, HADI MARDANI KAMALI, FAHIM RAHMAN, FARIMAH FARAHMANDI, (Member, IEEE), AND MARK TEHRANIPOOR, (Fellow, IEEE)
Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA
Corresponding author: Md Sami Ul Islam Sami (md.sami@ufl.edu)

ABSTRACT The semiconductor industry has adopted heterogeneous integration (HI), incorporating modular intellectual property (IP) blocks (chiplets) into a unified system-in-package (SiP) to overcome the slowdown in Moore's Law and Dennard scaling and to respond to the increasing demand for advanced integrated circuits (ICs). Despite the manifold benefits of HI, such as enhanced performance, reduced area overhead, and improved yield, this transformation has also led to security vulnerabilities in the SiP supply chain and in-field operations, ranging from chiplet piracy and SiP reverse engineering (RE) to information leakage. Although conventional countermeasures provide the desired robustness for monolithic ICs, they are insufficient for addressing these challenges in the context of HI. To address these concerns, this paper presents a novel root-of-trust architecture, augmenting the process of integration using a centralized chiplet hardware security module (CHSM), aiming to provide comprehensive and robust protection throughout the SiP supply chain and in-field operations. Also, the proposed architecture equipped with the CHSM effectively addresses potential security breaches while providing robust protection against zero-day attacks through its reconfigurable capabilities. Throughout *five* detailed case studies, this paper performs a comprehensive security analysis to illustrate the resilience of CHSM against contemporary attack scenarios in the HI domain.

INDEX TERMS Heterogeneous integration, packaging technology, system-in-package, chiplet, hardware security module, SiP security, supply chain security, vulnerability mitigation.

I. INTRODUCTION

With the ever-increasing demand for advanced ICs addressing complex applications, the semiconductor industry is adopting HI against Moore's law, and Dennard scaling [1], adopting modular and reusable IP blocks (chiplets) integrated into systems through emerging packaging technologies such as interposer layers, through-silicon via (TSV), embedded multi-die interconnects (EMIB), etc. [2]. This approach substantially improves functionality, yields, time-to-market, and cost reduction. However, the packaging technology, coupled with the complex SiP supply chain followed by the horizontal (globalized) business model, introduces new security vulnerabilities augmented with existing ones inherent in system-on-chips (SoCs) [3].

In contrast to the SoC supply chain for monolithic ICs with two main stages, as shown in Figure 1, the HI supply chain shown in Figure 2 consists of three phases [4]. In the SoC supply chain, the design house (trusted or untrusted) integrates IPs sourced from third-party IP vendors (untrusted) or developed in-house to design the SoC. The design then moves to offshore untrusted foundries for fabrication in the form of GDSII, subsequently undergoing assembly, packaging, and testing at untrusted offshore OSAT facilities before reaching the end-user, and eventually, it enters its end-of-life [5], [6] (see Figure 1). In HI, the chiplets

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

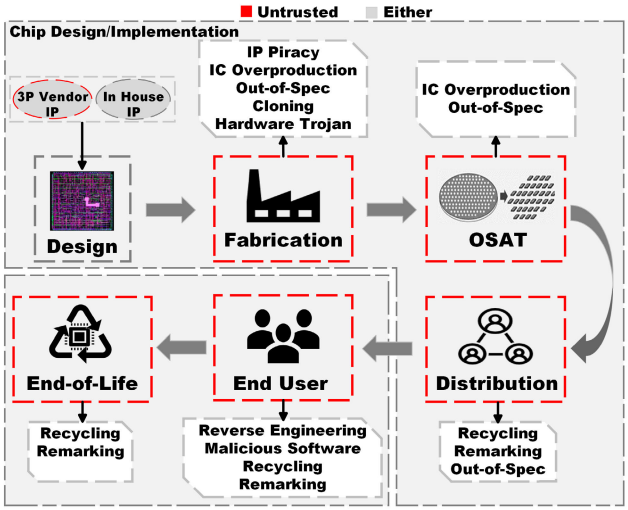


FIGURE 1. (Monolithic) SoC supply chain with associated vulnerabilities.

proceed through the same stages as in the (monolithic) SoC supply chain. Consequently, the chiplets are susceptible to the identical security vulnerabilities observed in SoCs, i.e., overproduction, piracy, cloning, recycling, etc. [7], [8] (see Figure 1). Moreover, the interposer acquired from an offshore interposer foundry (presumed untrusted) may introduce new security threats, like malicious (Trojan) insertion into the active interposer and overproduction of SiP (wherein the interconnection GDSII is accessible to the interposer foundry). As a result, integrating such chiplets obtained from untrusted sources (e.g., the open market) with such interposers will propagate the above security vulnerabilities into the SiP. In addition to existing attacks (e.g., malware [9], ransomware [10], etc.), which are common to both SiP and SoC, the SiP packaging technology also facilitates more accessible ways for attackers in the field to conduct probing attacks, leading to information leakage and RE of the SiP wherein the recovery of the interconnection alone suffices for the RE attack [4].

The existing literature primarily centers around identifying and mitigating security threats associated with SoCs, including obfuscation [11], camouflaging [12], split manufacturing [13], etc., to prevent overproduction, cloning, IP piracy, and RE of SoC. Moreover, SST [14] and CSST [15] strategies not only address these concerns but also prevent the distribution of out-of-spec and defective chips into the market. Also, security policies drafted for SoCs at the pre-silicon stage are synthesized into run-time security monitors, which are embedded within the SoCs to monitor suspicious activities and identify potential security risks during in-field [16], [17]. However, these methods may not directly address the security concerns associated with HI. For example, the activation keys (not encrypted) used in obfuscation techniques, SST and CSST, need to be transferred through the interposer layer of the SiP, rendering them vulnerable to probing-based attacks in a more straightforward way [4]. While camouflaging and split manufacturing are effective at the chiplet level, they do not adequately prevent RE at the SiP level. This is because potential adversaries among end users could RE [18] the interconnections and integrate identical chiplets, thereby enabling the cloning of the SiP. Moreover, formulating security policies (monitoring) must be tailored to address the specific security vulnerabilities related to HI.

Meanwhile, the existing root-of-trust mechanisms, like Intel SGX [19], ARM TrustZone [20], and AMD SEV [21], have been designed to isolate hardware elements and shield security assets from threats at the software level. For instance, ARM TrustZone partitions a system's resources into secure and normal worlds, preventing software in the normal world from accessing resources in the secure world. However, TrustZone is susceptible to fault injection attacks (voltage manipulation), which compromises its protective features [22]. Furthermore, when memory contents are transferred between chiplets while TrustZone securely executes in an SiP system, it becomes exposed to probing-based information leakage attacks, similar to the earlier scenario. Consequently, SiP systems remain vulnerable to attacks that exploit the unique vulnerabilities inherent in HI.

Establishing a secure HI necessitates implementing multi-faceted security measures, including (i) *encrypted* transmission of security-critical information through the interposer layer, (ii) SiP level *obfuscation*, (ii) *detection of potential fault injection and physical attacks* on the SiP and lastly, (iv) protection of SiP *against in-field security vulnerabilities and zero-day attacks*. To meet these requirements, this paper introduces an SiP architecture equipped with a chiplet hardware security module (CHSM) specifically designed to mitigate the security vulnerabilities associated with SiP systems. We present mitigation strategies and corresponding architectures featuring the CHSM to address these requirements. This integrated approach ensures holistic protection from the SiP supply chain stage to in-field operations. Leveraging the modularity and reusability of SiP packaging technology, the CHSM of the proposed architecture is developed as a centralized chiplet compatible with diverse chiplets within the SiP. Moreover, the CHSM offers a reconfigurable capability, allowing SiP designers and system integrators to customize the design according to specific security requirements and effectively countering zero-day attacks. The contributions of this paper are summarized in the following:

1) We first provide a detailed analysis of the SiP architecture, assessing a set of security vulnerabilities considering the HI supply chain and packaging technology.
2) We introduce our innovative centralized CHSM design, aiming to safeguard the SiP from both the supply chain and in-field security vulnerabilities.
3) Throughout five threat cases in HI, we comprehensively analyze the techniques employed for mitigating security vulnerabilities in our proposed architecture. Moreover, the efficacy of these techniques is validated through a detailed security and performance analysis.

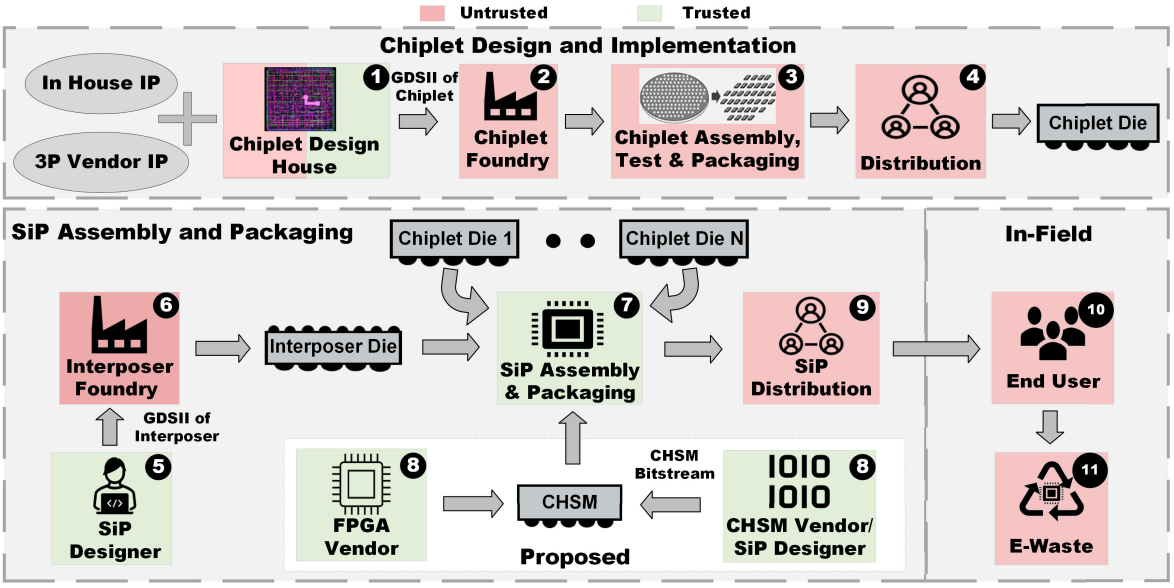


FIGURE 2. The Main Stages of SiP supply chain and the trustworthiness of entities throughout supply chain (With proposed chsm integration).

4) We explore future possibilities for advancing the CHSM and its potential to address a wider range of attack scenarios.

II. THREAT MODEL IN SIP SUPPLY CHAIN

The supply chain of HI is comprised of an intertwined network of stakeholders distributed globally, divided into three phases, as depicted in Figure 2. In the initial phase of chiplet design and implementation, various chiplet design houses follow a similar supply chain with SoC to manufacture chiplet dies (steps 1 to 4). However, unlike the supply chain for SoCs, where packaged SoCs are delivered directly to end users, chiplet dies are manufactured for procurement by SiP integrators during the SiP assembly and packaging phase. This phase is a distinct addition compared to the supply chain for SoCs. The SiP designer is assumed to be responsible for designing the SiP, and in the subsequent phase of SiP assembly and packaging, the SiP designer transfers the GDSII of the interconnections to an offshore interposer foundry (steps 5 and 6). The SiP integrators (the SiP assembly and packaging entity) utilize in-house developed chiplets or acquire chiplets from external vendors and integrate them with the interposer. The SiPs then undergo assembly, testing, and packaging (step 7) before market distribution (step 9) and finally at the end user (step 10). Eventually, upon reaching the end of their life, the SiPs are considered e-waste (step 11). While designing the proposed security protocols and framework, we have considered the following threat model:

- **Trusted Chiplet Design, Untrusted Chiplet Foundry:** In this model, we assume the chiplet design house is fully trusted (green part of step 1). However, these entities rely on (mostly) offshore fabrications (steps 2-4), which introduces security concerns such as cloning, overproduction, and piracy because of the involvement of untrusted parties. Considering trust in the chiplet design house, a set of mitigation strategies may be applied, e.g., obfuscation, watermarking, etc. We also can assume that trusted chiplets are treated as a white-box model (more realistically, gray-box), with known internal connections to the SiP designer and integrators.
- **Untrusted Chiplet Design, Untrusted Chiplet Foundry:** Chiplets from trusted chiplet design houses are secure and reliable, while chiplets from untrusted vendors (red part of step 1) or the open market are untrusted, carrying security vulnerabilities like malicious implants, defective or out-of-spec chiplets, recycled chiplets, etc. These chiplets also may have no security measures. We also assume that untrusted chiplets are treated as a black-box model with undisclosed internal connections.
- **Untrusted Interposer Foundry:** The offshore interposer foundry may covertly insert a Trojan within the active interposer layer after receiving the GDSII from the SiP designer (step 6). Although efforts have been made to encourage local handling of interposer fabrication (e.g., CHIPS), we assume this step could be untrusted [3]. Moreover, the untrusted interposer foundry can access the interconnection layer and has the ability to obtain identical chiplets from various sources: the open market, reverse-engineered and cloned chiplets from the field, as well as recycled or remarked chiplets to overproduce the SiP.
- **Untrusted End User:** Upon deployment in the field (step 9), SiPs encounter a range of potential threats, spanning from attacks during system boot to firmware and software-level breaches [10], [23]. In addition, adversaries at the end user level might exploit weaknesses in the chip's operating system, leading to compromised

security and functionality of the devices [24]. Furthermore, an in-field system is vulnerable to several fault injection attacks. Skilled attackers can exploit different techniques (e.g., underpowering [25], [26], overclocking [27], [28], electromagnetic (EM) radiation [29], laser illumination [30], etc.) to inject bit-flip or bit-set-reset faults into the security-critical components of a system and compromise the system security. Furthermore, skilled end users have the potential to reverse engineer the SiPs, which could result in the creation of cloned SiPs.

- **Untrusted E-waste Facilities:** During the disposal and recycling phase (step 10), e-waste facilities have the potential to recycle SiPs for reuse in the supply chain without adequate inspection, thereby reintroducing compromised or faulty components [31].

III. OVERVIEW OF SIP ARCHITECTURE

Packaging technology and the intended application or target device greatly influence the architectures of SiP. SiP architectures can vary, ranging from derivatives of traditional printed circuit boards (PCBs) to more complex systems resembling large SoCs. The evolution of SiP technology has seen the emergence of various advanced packaging methods, each designed to address specific design challenges and enable higher levels of integration and performance. Among the prominent packaging technologies are (see Figure 3):

A. 2D PACKAGING

This approach involves directly integrating multiple chips on a packaging substrate, resembling a miniature PCB. Utilizing methods such as fan-out/fan-in wafer-level packaging and narrow pitch wire bonds, 2D SiP [32] offers increased integration capabilities and a smaller form factor, making it well-suited for portable devices like smartphones, tablets, and smartwatches.

B. 2.1D PACKAGING

2.1D packaging [33] employs an ultra-high-density redistribution layer (RDL) situated between thin-film layers, characterized by precise metal line width and spacing. Organic interposers can also be employed in this category, providing a cost-efficient means to enhance input and output density for advanced IC packaging. The adoption of organic materials possessing high elastic modulus yields reduced internal stress, thereby enhancing overall reliability.

C. 2.5D PACKAGING

This packaging method integrates an additional interposer layer between the chiplets and the packaging substrate. For instance, Chip-on-Wafer-on-Substrate (CoWoS) [34] stacks multiple chiplets on a silicon interposer, enabling high-speed data buses between high-performance logic and memory devices. Alternatively, bridge-based 2.5D packaging uses 'bridges' to connect adjacent chips such as EMIB from intel [35], offering an alternative to traditional interposers.

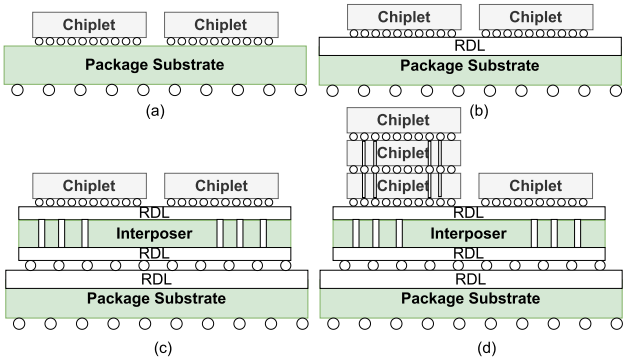


FIGURE 3. Different advanced packaging technologies for chiplet stacking and interconnection: (a) 2D (b) 2.1D (c) 2.5D (d) 3D packaging.

D. 3D PACKAGING

In 3D packaging, semiconductor dies are stacked vertically, and through silicon vias (TSVs) are used for interconnections. This approach is commonly employed for stacking memory on top of processors or integrating analog and digital circuits. Intel's Foveros [36] is a noteworthy example of 3D packaging, where different functional dies are stacked using TSVs and micro-bumps.

Among these technologies, 2.5D packaging has gained more popularity as it balances various factors, including enhanced performance via shorter interconnect lengths, efficient high-bandwidth communication via TSVs, improved power efficiency, form factor optimization, and capacity for a wider range of applications and design complexity than other packaging technologies [32], and accordingly, in this study, we consider the more generic 2.5D SiP structure for our implementation and evaluation.

Apart from the packaging technology, the system-level SiP integrator must choose a communication architecture and determine the appropriate physical layer communication technologies. Based on the whole SiP target functionality and performance, it could involve options like serializer/deserializer (SerDes) [37], peripheral component interconnect express (PCIe), advanced interface bus (AIB) [38], or universal chiplet interconnect express (UCIe) [39]. Amongst these interconnection technologies, network on chip (NoC) [40], [41], [42] is also gaining popularity as it allows for seamless integration and efficient communication between heterogeneous chips and IP blocks within the SiP. By reducing design complexity and increasing scalability, NoC helps overcome communication challenges for large systems. The NoC fabric can be realized on a chiplet or integrated inside the active interposer layer (see Figure 4), providing a structured and organized inter-chiplet (In this paper, the latter case is used.).

IV. STATE-OF-THE-ART SECURITY ASSESSMENT ON SIP

Recent investigations indicate a limited exploration of the security aspects of HI systems, with some exposure to academic researchers. Previous studies have primarily centered on developing taxonomies for security vulnerabilities related

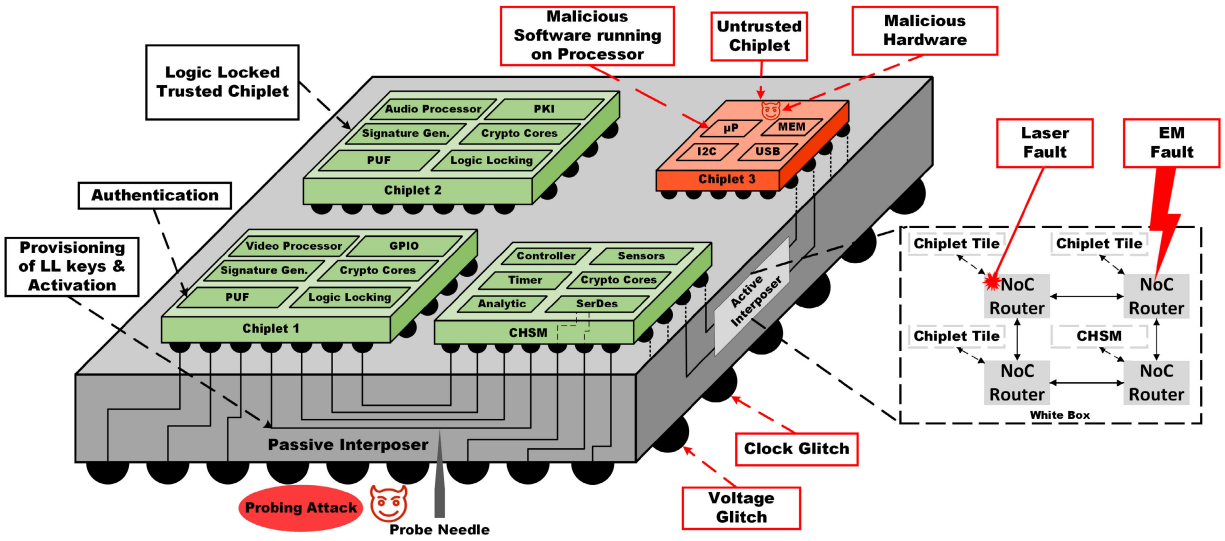


FIGURE 4. High-level diagram of an SiP architecture, its threat possibilities, and integration of chiplet hardware security module (CHSM).

to interposer-based approaches [43] and examining the physical assurance challenges for HI [44]. Some studies suggest developing security measures at the active-interposer level. For instance, a study employed 2.5D interposer technology to establish system-level security against hardware and software threats by integrating chiplets through a security-enforcing interposer [45]. Similarly, another study introduced a secure Network-on-Chip (NoC) by integrating security monitors into the NoC to defend against adversarial traffic [40]. However, these security measures lack applicability across various SiP architectures and fail to ensure the system's trustworthiness through a unified root-of-trust.

Consequently, these approaches are vulnerable to emerging security threats, such as the potential for malicious implants within the root-of-trust, overproduction, and cloning. It is crucial to note that these security solutions are also at risk of probing-based attacks, which may lead to the leakage of sensitive information through the interposer layer. Moreover, remote attackers can introduce software-induced hardware fault injection attacks to make these security solutions inapplicable in mission-critical applications [46]. Furthermore, commercial Electronic Design Automation (EDA) tools predominantly focus on 2D monolithic SoC design or verification methodologies. While some EDA tool vendors [47] offer additional features for designing and verifying 2.5D or 3D systems, the majority of these features are inaccessible for academic use. Consequently, there is a lack of EDA tool-based assessments for heterogeneously integrated SiP architectures, hindering the identification of potential threats, quantitative or qualitative evaluation of vulnerabilities, and the proposal of a unified security solution. In summary, given the increasing demand for advanced packaging technology, the hardware security research community is urged to devise unified security solutions to address emerging threats linked to the complex structure and supply chain vulnerabilities of 2.5D or 3D heterogeneous system design.

V. TARGETED SUPPLY CHAIN AND IN-FIELD THREAT CASES

The shift towards advanced packaging-based system architecture has rendered the SiP designs vulnerable to various threats. Given the SiP architecture and the packaging technology discussed earlier, this paper focuses on studying and mitigating some of the most important security threats and risks, as described in detail in the following sub-sections.

A. CASE C1: PROBING-BASED INFORMATION LEAKAGE

Modern chips utilize a variety of security assets, including session keys, digital certificates, public/private keys, logic locking keys, physical unclonable function (PUF) responses, etc. These assets contain sensitive information proprietary to the chip designer and are vital to security operations, such as secure data transfers, chiplet activation at boot, client-server authentication, etc. [48], [49]. Typically, these assets reside in secure tamper-proof memory (TPM) (e.g., non-volatile memory (NVM)) within the SiP and are passed through (unencrypted) the chiplets during data communication at the interposer layer. In this case, adversaries (in-field) can insert semi/non-invasive micro/nano probe needles into the interposer layer [50], [51], thus gaining access to the transmitted data [4], [43]. Consequently, they gain unauthorized access to extract the security assets, as shown in Figure 4. This exposure of assets leads to financial losses for the SiP designer and compromises the security protocols reliant on the mentioned security assets.

B. CASE C2: MALICIOUS HARDWARE MODIFICATION

As shown in Figure 2, although integration serves as the primary trust anchor, the SiP integrator (with ownership of the chip) must depend on the design and manufacturing

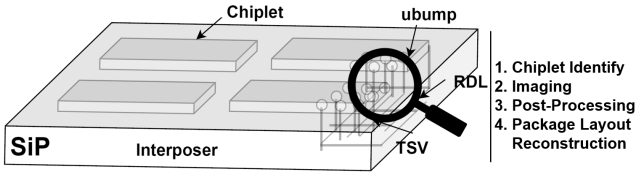


FIGURE 5. SiP RE through imaging and interposer layout reconstruction.

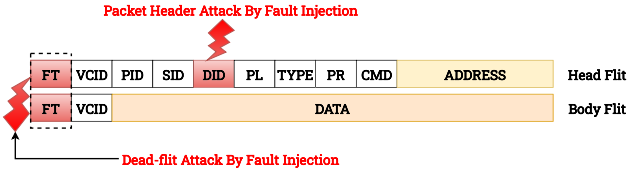


FIGURE 6. Packet header and dead-flit attack on NoC buffer in-port.

capabilities of upstream parties, i.e., chiplet designers (untrusted) and offshore foundries, thereby enabling the possibility of malicious functionality being implanted in individual chiplets. For instance, chiplet design/fabrication or/and interposer fabrication team may have intentions to incorporate hardware Trojans to intercept sensitive communication among chiplets [52], resulting in unauthorized physical disruptions to the global on-chip infrastructure, such as the power distribution network (PDN), leading to the induction of faults in other chiplets [53]. It also could be substantial performance degradation that may result from injecting a large volume of fake traffic into the on-chip communication interface [54]. Note that while the current HI solutions may primarily be implemented internally by prominent semiconductor companies such as AMD and Intel [55], more malicious implants will be witnessed over time due to the ever-increasing emergence of third parties.

In the field, SiPs can be vulnerable to software-level attacks, including unauthorized firmware manipulation by end-users for privilege escalation and remote network attacks exploiting application programming interfaces (API) or OS vulnerabilities to execute malware or ransomware on devices [10]. These actions enable control manipulation or system takeover for ransom [56]. In this study, we aim to focus on malware variants disrupting control-flow integrity and on-chip power network switching patterns.

C. CASE C3: REVERSE ENGINEERING OF SIP

Commercial availability of SiP chiplets eases SiP RE compared to monolithic SoCs. Adversaries can disassemble SiP packages to understand chiplet types and interconnections. Non-destructive methods like X-ray tomography provide multi-layer SiP images for interconnect analysis [57], while focused-ion beam (FIB) and scanning electron microscope (SEM) capture chiplet layer details [58]. Passive interposer RE involves continuity checks with nano-probing and logic analyzers. By locating these details, adversaries can reconstruct and clone the entire package, as shown in Figures 5.

Furthermore, the growing demand for SiP designs has led to increased chiplet usage, posing IP protection challenges for chiplet owners and SiP designers. Traditional methods like logic locking are effective for monolithic SoCs [11], [59], [60], [61], but chiplets, being standalone entities, present unique challenges. Chiplets are used across different SiP designs, making individual obfuscation economically impractical for chiplet owners. The challenge is to find a secure and efficient protection mechanism that lets SiP designers safeguard their system-level designs using obfuscated chiplets while allowing chiplet owners to control unlocking keys and prevent unauthorized access to IPs. Also, embedding unlocking keys in chiplet TPM/VNM is vulnerable to RE, making security reliant on key storage integrity.

D. CASE C4: SIP COUNTERFEITING THREATS

Unlike counterfeit SoCs with a single die attack surface, counterfeit SiPs can manifest at three levels:

(1) *At the SiP-level*, untrusted e-waste facilities can recycle SiPs at end-of-life, refurbish them, and resell them as new (with degraded performance due to aging) [8]. Deviations from expected behavior can cause system-wide impacts, especially problematic in critical applications.[1] Despite internal architectural differences between SiPs and SoCs, the recycling threat for SiPs mirrors that of the SoC domain [8].

(2) *At the chiplet level*, vulnerabilities resemble those of SoC counterfeiting, and if left unaddressed, they can escalate to SiP-level concerns. As many chiplets are incorporated in an SiP, the impact of such threats is compounded. Counterfeit chiplets can introduce parametric or functional issues that negatively affect chiplet and SiP designers. Out-of-spec and defective chiplets, yet approved by chiplet foundries, may result in improper SiP operation. Chiplet distributors might also remark that chiplets appear in higher grades for increased profits, even if they cannot meet the necessary conditions.

(3) *At the interposer level*, vulnerabilities can stem from out-of-spec or defective SiPs. Interconnects may exhibit incorrect parametrics, such as capacitance or path delays. Interposers approved by the foundry may harbor latent defects that can lead to SiP malfunctions/failures over time. Also, interposer foundries might overproduce interposer dies, making them available to adversaries for use in replicas/similar designs.

Considering these counterfeit threats, either at SiP-level, chiplet-level, or interposer-level, a high-level definition of them is summarized in Table 1. It is worth noting that some of the threats listed here may overlap with other threats defined in other threats (e.g., RE in case C3).

E. CASE C5: FAULT INJECTION ATTACKS

This threat focuses on system-level fault injection (FI) impacting inter-chiplet communication within SiPs. This

[1]Recently, Apple sued a former e-waste facility in Canada as they found more than 100,000 devices, $22.7 million worth of product, sent for disposal were still operating and accessing the internet [62].

TABLE 1. SiP counterfeit threats: definition, source of threats, and level of manifestation.

Threat	Definition	Level	Source of Threat (Figure 2)
Recycling	SiP is not disposed, but refurbished and sold as new.	SiP	11
Remarking	SiP package is remarked to yield higher profits.	SiP	9
Out-of-spec	Interposer or chiplets do not meet functional or parametric specs.	All	2 and 6
Defective	Interposer or chiplets contain defects that can cause system failure or reliability issues.	All	2 and 6
Cloning	Replicas of SiP are created for financial gain	All	Any untrusted entity
Overproduced Interposer	Interposer is overproduced for use in a replica or other designs	Interposer	6
Forged Documentation	Certificates and documents are altered to misrepresent the SiP for higher profits	SiP	Any untrusted entity

threat applies to multi-die chiplet-based designs with NoC routers with the tiled-chip multi-core systems (See Figure 4). This model is adopted by leading semiconductor companies (e.g., Intel's Ponte Vecchio SiP using the X^{e}HPC with X^{e}Link for GPU-to-GPU communication [63], [64], [65]. For this study, we target dead flit attacks [66] and packet header attacks [67]. Controlled FI attacks induce bit-flips in NoC router input-port buffer registers, corrupting flit types (FT) or destination addresses (DID), as depicted in Figure 6. Consequences include network traffic stalling (e.g., denial of service) or packet drops due to incorrect destination addresses, leading to availability violations. It is worth noting that these (bitwise) attack scenarios may be the target of hardware Trojans, but attackers can also execute them by injecting bit-flip faults.

Considering the SiP supply chain structure demonstrated in Figure 2, our assumption is that the attacker has limited knowledge of the chiplets' functionalities and design details (e.g., black box), except for specific networking components (e.g., switch-box protocol, router components, packet generator, etc.) where fault injection attacks are viable. For a fault injection attack to be successful, the attacker must have both expertise and adequate resources, enabling them to exert precise control over the timing and location of the injected faults. Therefore, we assume the attacker is sufficiently equipped and skilled to execute fault injection attacks, exercising meticulous control over when and where the faults are introduced.

VI. DISTINCTIONS BETWEEN SOC AND SIP IN EACH CASE

In this section, we demonstrate the primary distinctions in the supply chain and security vulnerabilities between SoC and SiP, examining each case individually.

A. CASE C1: PROBING-BASED ATTACKS

The primary architectural distinction between an SoC and an SiP lies in their packaging technologies. SoCs consolidate all IPs onto a monolithic chip, where data transmission occurs through stacked metal layers above the base layer. This stacking makes it extremely challenging to intercept any transmitted assets through these layers using micro- or nanoprobes without causing potential damage to the SoC, rendering it nonfunctional. SiPs integrate multiple SoC-like chiplets onto a passive or active interposer (as discussed in Section III), and data exchange happens between chiplets through the interposer layer. This setup introduces a unique vulnerability compared to SoC, as it allows attackers to potentially intercept security assets more easily through probing attacks (see Figure 4). As a result, security protocols such as encryption become vital for protecting assets transmitted across the interposer within SiP, while they may not be as essential for the SoC.

B. CASE C2: MALICIOUS HARDWARE MODIFICATION

When it comes to malicious hardware modifications, one can see a clear difference between the contexts of SoC and SiP because of the changes in supply chain models. As for conventional SoCs, the hardware modifications mainly stem from two aspects, i.e., malicious IPs and rogue foundries (see Figure 1). These untrusted entities may stealthily implant adversarial functionality at the behavioral or silicon level. As for SiPs, the supply chain has become even more convoluted, as illustrated in Figure 2 since most actors except for the SiP integrators cannot be trusted completely. For example, a chiplet design house itself is responsible for defining the entire functionality and specification. It might be a victim of malicious third-party vendor IP or hide malicious circuitry in the original design to compromise the security of other chiplets in the same SiP later. Similarly, foundries and facilities for chiplet fabrication and packaging may tamper with the GDSII implementation or silicon. Furthermore, chiplets rely on the interposer as the communication infrastructure to talk with each other. The interposer foundry might tend to manufacture falsified silicon for communication interception/spoofing during run-time, which is not applicable in the SoC devices and supply chain. Such threats are unique and threatening, calling for dedicated solutions to guarantee SiP security.

C. CASE C3: REVERSE ENGINEERING OF SIP

The evolution of advanced packaging technology necessitates the development of specialized obfuscation solutions, as existing logic locking techniques designed for integrated circuits or SoCs face limitations in this domain. Traditional methods, which aim to obscure IC functionality through key gates or control FSMs with unique input patterns, cannot seamlessly transition to the SiP landscape. This disconnect is attributed to the distinct architectural and manufacturing

steps of SiPs compared to SoCs or ASICs. The SiP supply chain involves multiple stakeholders, from chiplet designers to assembly facilities where SiP designers have limited to no access to the design for security features of the chiplets. Moreover, it creates a trust issue for chiplet designers to share design critical security info (e.g., logic locking keys) with system-level designers for forward trust. In the chiplet ecosystem, if the chiplets can be sourced from third-party vendors, attackers can reproduce counterfeit SiPs by reverse engineering the interposer layer [18]. This extended supply chain increases the risk of IP piracy and tampering. SiP-specific obfuscation solutions must address these supply chain vulnerabilities to ensure end-to-end protection.

D. CASE C4: SIP COUNTERFEITING THREATS

SiP counterfeiting differs from SoC counterfeits due to the potential inclusion of untrusted chiplets that might be remarked, out-of-spec, overproduced, or defective. Additionally, the interposers, fabricated in separate foundries, could face risks of being overproduced, out-of-spec, or defective. Unlike the cloning threat to SoCs, SiP cloning creates a replica in three possible fashions whereby 1) the entire SiP is reverse engineered, including the interposer and all chiplets, 2) the interposer and some chiplets are reverse engineered, and other chiplets are purchased on the open market, and 3) only the interposer is reverse engineered, and all chiplets are purchased. This process contrasts with SoC cloning, which typically involves reverse engineering a single die to create a replica. As hinted at earlier, out-of-spec and defective SiPs create a larger challenge than their SoC counterparts in that any chiplet and the interposer may not function or not meet parametric specifications. Any of the dies can contain defects that degrade the reliability of the entire SiP.

E. CASE C5: FAULT INJECTION ATTACK

Regarding fault injection attacks, there is a clear distinction between the SoC and SiP concerning threat models because of the structure and supply chain changes. As for conventional SoCs, an attacker mainly targets the functional block of an SoC to inject timing faults to extract the secret keys (confidentiality violation) or modify the secure memory contents or configuration bits (integrity violation). In this case, the attacker requires a complete knowledge of the device's functionality to pinpoint the location and timing of the attacks (e.g., white-box attacks). It is feasible to expose the structure of the monolithic SoC by destructive reverse engineering and learn the circuit's layout. In addition, an attacker can analyze pre-silicon soft IP (gate-level netlist) or firm IP (physical layout) to guide a white-box attack on an SoC. In contrast, a heterogeneous system consists of several fabricated chiplet dies in different technology nodes within a single SiP. Since the SiP owner usually purchases the chiplet IPs from different vendors, it is impractical for an attacker to learn the functional behavior of each fabricated chiplet individually. Due to the variety of technology nodes involved, the device timing changes significantly [68], [69], causing fault injection vulnerabilities to vary from chiplet to chiplet. Therefore, conducting any random attack from a chiplet without knowing its functional details or without dealing with different process nodes is extremely unlikely to compromise the system-level security of an SiP. In this case, the more viable option for an attacker is to reverse engineer only the inter-chiplet communication layer to know the functionality and perform a successful attack to compromise secure communication within an SiP. It implies that a heterogeneous integration shifts the white-box attack models to gray-box attack models where the functional chiplets are entirely back-boxes and only the inter-chiplet communication layer is a white box. Moreover, unlike a conventional SoC, die stacking techniques in 2.5D or 3D heterogeneous SiP can automatically shield optical illumination or electromagnetic radiation to reach a specific chiplet location. However, the active interposer layer (absent in an SoC) embedding the inter-chiplet communication in an SiP is more vulnerable to optical, electromagnetic, or probing attacks. Eventually, unlike an SoC, a heterogeneous SiP introduces emerging fault injection threats on the interposer layer.

VII. PROPOSED ARCHITECTURE: CHSM-ENABLED SIP

Considering the unique SiP-oriented security vulnerabilities outlined in cases C1 to C5, to address such threats, we introduce an enhanced SiP architecture equipped with an FPGA-based chiplet hardware root-of-trust security module, CHSM as depicted in Figure 7.[2] The CHSM is designed in alignment with the standard hardware security module (HSM) definition to fulfill the security requisites essential for mitigating SiP-oriented threats. During the design phase, the SiP designer/integrator specifies the security requirements for the SiP and implements the CHSM (See step 8 of Figure 2). As the SiP designer/integrator acquires chiplets as hard IP and lacks direct access to all of the internal signals of chiplets (in a gray-box model), CHSM is designed as a distinct centralized chiplet containing critical security measures. The SiP integrator acquires the CHSM chiplet along with other chiplets and integrates them into the SiP. While CHSM is designed to protect the SiP from all potential security vulnerabilities throughout its lifespan, this paper specifically emphasizes the elements required for the CHSM to combat threats described in Section V (i.e., C1-C5).

A. CHSM ARCHITECTURE: ARCHITECTURE AND FLOWS

The CHSM architecture, as illustrated in Figure 7, consists of four main components: (i) *processing/controller unit*, (ii) *Cryptographic modules and hardware primitives*, (iii) *Sensors* and lastly, (iv) *analytical/evaluative components*. The processing/controller unit encompasses a processor core (e.g., ARM, RISC-V) with a memory system, in which the bootloader and firmware are securely loaded.[3]

[2]Note that the CHSM as the root of trust in the whole SiP with protection techniques (e.g., C1/C3 mitigation) are against the specific attack vectors, such as FPGA bitstream reverse engineering [70], [71] or tampering [72].

[3]As this paper focuses on the security protocols aimed at alleviating hardware-based security vulnerabilities of SiP architecture, we defer the discourse on the boot process of the CHSM to future works.

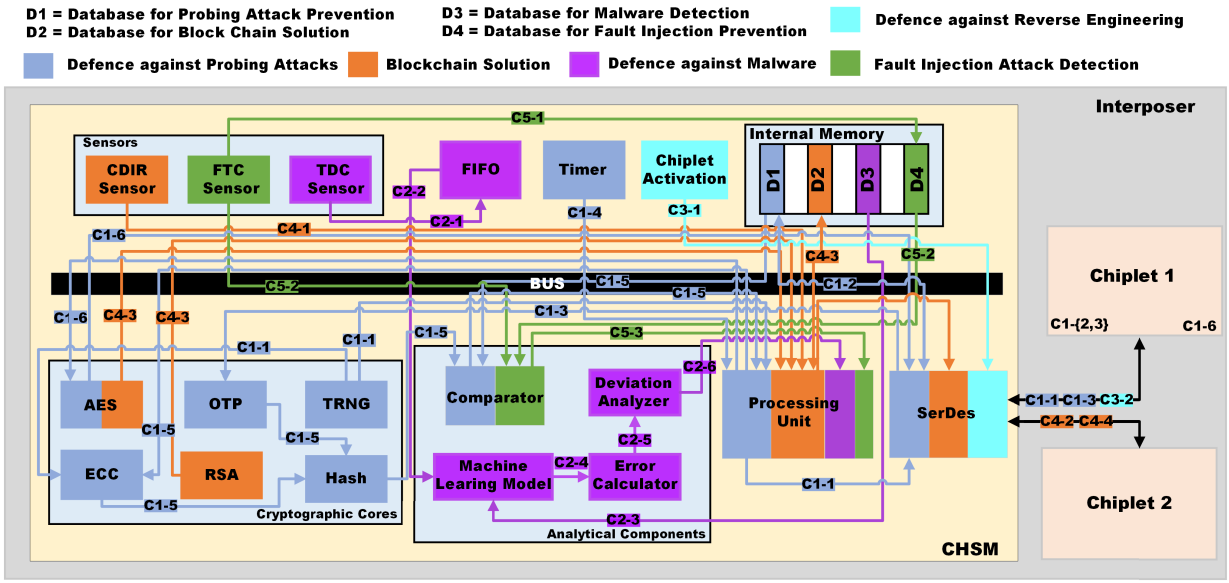


FIGURE 7. High-Level Overview of CHSM-enabled SiP Architecture against C1-5. Ci-j represents the j^{th} action step of case Ci mitigation flow. Color coding determines how each module is employed per each case Ci (Some of the modules are common between different cases for mitigation).

The CHSM incorporates diverse cryptographic IPs [73], including symmetric encryption cores (e.g., one-time pad (OTP), AES, hash functions (e.g., SHA256), asymmetric crypto accelerators (e.g., RSA and ECC modules), and hardware primitives (e.g., TRNG and PUF) for executing security operations.

To target and pinpoint physical-oriented attacks on hardware (e.g., FI through voltage/clock glitching or laser injection), the CHSM also encompasses various sensors (e.g., time-to-digital converter (TDC) sensor, fault-to-time converter (FTC) sensor, combating die and IC recycling (CDIR) sensor) and analytical components for security analysis and verification. Upon detecting a physical attack, the proposed CHSM triggers preventive measures (e.g., rendering tamper-proof/tamper-resistant to secure memory -assets- integrity). Moreover, the CHSM provides a secure cryptographic boundary that prevents access to the SiP's security assets by unauthorized chiplets. Alongside these features, the CHSM includes security application-specific components, such as a hardware-based timer, a FIFO, and a chiplet activation module. Furthermore, the CHSM includes a SerDes and supports the UCIe protocol, facilitating seamless communication with other chiplets while optimizing data transfer rates through a reduced number of micro bumps. Considering the architecture and flows represented in Figure 7, the subsequent sections explore leveraging the capabilities of CHSM for mitigation techniques of each C1-C5 threat.

B. C1 MITIGATION: AGAINST PROBING ATTACKS

To be against C1, the proposed architecture establishes trust between the CHSM (as verifier) and the chiplets (as prover). To achieve this, we employ a secure storage of these assets (within the CHSM memory). Alternatively, if external memory is used, this architecture provides the support of storing them in encrypted form, with only the CHSM possessing the decryption key. To prevent probing attacks, the CHSM authenticates the chiplets and establishes a shared secret key with the chiplets by using the elliptic curve Diffie-Hellman key exchange (DHKE) protocol [74], [75], [76]. Using this shared secret, the CHSM enables encrypted transmission of assets between chiplets.[4]

Depending on the chosen authentication and key exchange protocol, the CHSM (verifier) and trusted chiplets (prover) must incorporate essential components and maintain a series of authentication and key exchange steps. Due to the potential diversity in these methods among trusted chiplet design houses, the SiP integrator must equip the CHSM with the required hardware and firmware components to support these protocols. In our proposed architecture for addressing case C1, the CHSM employs challenge-response pairs (CRPs) to authenticate each trusted chiplet. The authenticity of chiplets is established by evaluating their responses using a predefined series of steps, outlined in Figure 7 (C1-1 through C1-6).

1) DETAILED FLOW OF C1 MITIGATION IN CHSM-ENABLED SIP

Relying on Figure 8, following is the detailed step-by-step description of the authentication and key exchange protocol:

Step C1-0: CHSM communicates securely with a trusted server (e.g., Transport Layer Security (TLS) 1.3 [77]), obtains

[4]In the proposed CHSM-enabled SiP architecture, we assume that only trusted chiplets are responsible for carrying out security operations, and the CHSM, when necessary, transmits security assets to these chiplets.

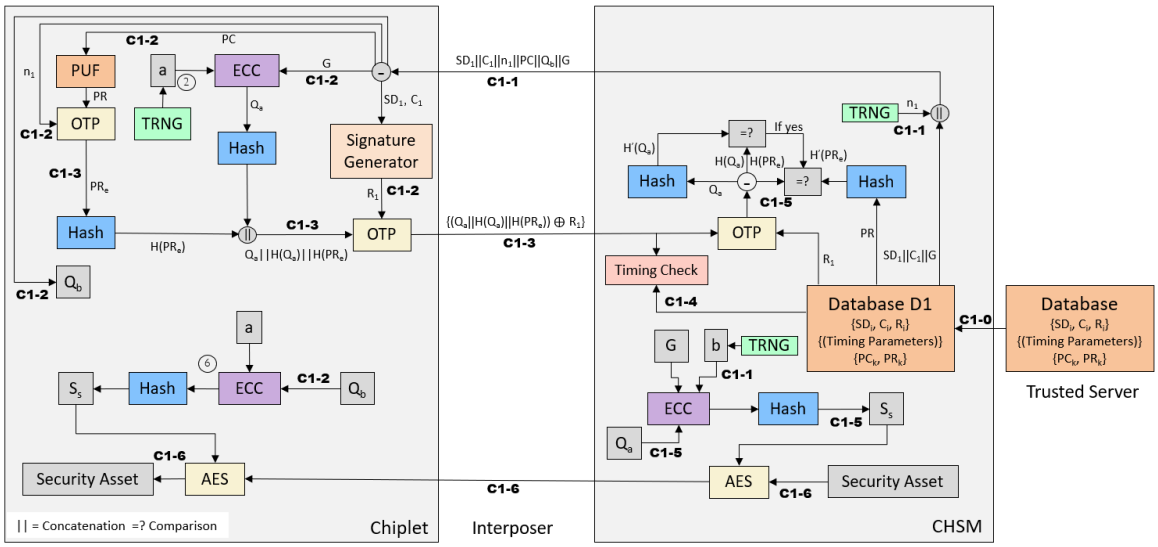


FIGURE 8. Detailed block diagram of the authentication and key exchange protocol against C1 (Probing Attack).

the CRP database along with the parameters and stores it inside CHSM memory (D1 database). Here, we assume that during the chiplet design and implementation stage, this database was successfully enrolled in a trusted server.

Step C1-1: The CHSM generates a random number (n_1) using the TRNG. Then, it picks a PUF challenge (PC) and signature generator inputs (SD_1 & C_1) from D1. The CHSM also generates its private key (b) and uses ECC to multiply it by the base point of the elliptic curve G from memory. This operation yields the CHSM's public key, Q_b. Then, by concatenating all (SD_1, C_1, n_1, PC, Q_b, and G), CHSM sends it as the challenge to the chiplet and initiates its timer.

Step C1-2: Upon receiving the challenge, the chiplet extracts the individual elements of the challenge. The chiplet applies SD_1 and C_1 to the signature generator, PC to the PUF, n_1 to the one-time pad (OTP), and G to the ECC module. Within the chiplet, TRNG generates the chiplet's private key (a), while the ECC module generates its public key (Q_a). It is assumed that the ECC module is designed based on the recommended parameters of the domain of the elliptic curve [76]. Within the chiplet, a shared secret key S_s is generated by multiplying the chiplet private key a by Q_b and subsequently hashing the result ($S_s = H(aQ_b)$). Lastly, the signature generator within the chiplet generates the signature R_1.

Step C1-3: The generated PUF response PR is XORed (PR_e) using the OTP with n_1 and subsequently hashed, $H(PR_e)$. This resulting value, $H(PR_e)$, represents the derived PUF signature of the chiplet. Furthermore, the chiplet forms a message by concatenating its public key (Q_a) with its hash ($Q_a||H(Q_a)$). This message is then combined with $H(PR_e)$, resulting in a concatenated message ($Q_a||H(Q_a)||H(PR_e)$). The concatenated message is XORed using the R_1 to create the chiplet's response. Consequently, this XORed message, $(Q_a||H(Q_a)||H(PR_e) \oplus R_1$, represents the chiplet's response.

Step C1-4: Upon receiving the response, the CHSM halts its timer and verifies if the response was received within the threshold T_1. When the timer exceeds T_1, the CHSM flags the chiplet as unauthentic and refrains from transferring the security assets. The threshold time is determined by the SiP integrator, which needs to be sufficiently short to prevent attackers from executing impersonation attacks.

Step C1-5: The CHSM XORs the received response using R_1 and separates its elements. It then generates the hash ($H'(Q_a)$) of the Q_a received from the chiplet and compares it with $H(Q_a)$. If the hashes match, the CHSM confirms that the response originated from the intended chiplet. Furthermore, the CHSM encrypts PR using the OTP with n_1, generates the hash ($H'(PR_e)$), and compares it with the received $H(PR_e)$. If these hashes are also identical, the CHSM verifies the authenticity of the chiplet. If any verification steps fail, the CHSM refrains from transferring the security assets into the chiplet. Likewise, the CHSM generates its shared secret key by multiplying its private key b with the chiplet's public key Q_a and hashing the outcome ($S_s = H(bQ_a)$).

Step C1-6: After generating the shared secret key, the CHSM utilizes it to encrypt the security assets. Afterward, these encrypted assets are transferred into the chiplet. The CHSM then proceeds to authenticate the next chiplet.

2) PUF AND SIGNATURE GENERATOR ARCHITECTURE

A weak PUF [78], as opposed to a strong PUF [79], is used due to its capability to consistently produce reliable responses over time [80], [81], while supporting a reduced number of CRPs.[5] To safeguard the PUF response, it is obfuscated using a derivation function. The derivation function operates in combination of the signature generator and a hash function [83] within the chiplet. The signature generator comprises four 32-bit Nonlinear Feedback Shift Registers

[5]Any weak PUF can be used as they offer the required reliability [82].

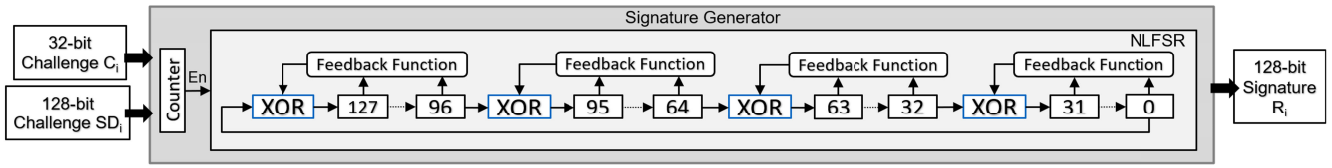


FIGURE 9. Signature generator architecture (function used in chsm-enabled SiP: $x_0 + x_2 + x_6 + x_7 + x_{12} + x_{17} + x_{20} + x_{27} + x_{30} + x_3x_9 + x_{12}x_{15} + x_4x_5x_{16}$).

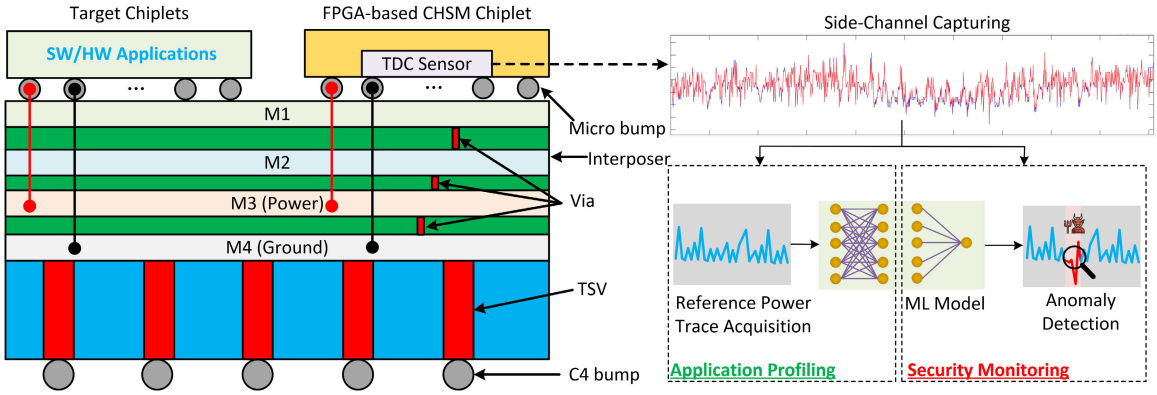


FIGURE 10. Overview of power noise variation-based mitigation on malicious functionality detection using TDC sensor in CSM-enabled SiP.

(NLFSRs) and a 32-bit counter (see Figure 9). The counter in this architecture determines the NLFSRs shift count, known as the capture-cycle count. Once the initial seed and capture-cycle count are received from the CHSM, the signature generator shifts the feedback bits until the counter completes counting, resulting in a distinct digital signature for the chiplet. Our feedback function (see Figure 9) is adopted from [84]. However, chiplet designers may opt for a custom feedback function, modify tapping locations, or introduce additional nonlinearity in the signature generator based on their security requirements and design specifications. The advantage of the signature generator lies in its ability to generate a unique signature for every session between the CHSM and the chiplet. This eliminates the need for a strong PUF when dealing with numerous CRPs for authentication.

C. C2 MITIGATION: AGAINST MALWARE ATTACKS

As shown in Figure 7 (C2-1 through C2-6), our malware/ransomware mitigation solution requires run-time monitoring capabilities from the corresponding sensor(s) and run-time computation from the on-chip analytical components, processing units, and internal memory [85]. Figure 10 shows a different view of the CHSM, positioned atop the silicon interposer, while the target chiplet (e.g., chiplet 2 in Figure 7) running software and/or hardware applications may face potential compromise. In SiP architectures, with the limited controllability and observability that SiP integrators have over the target chiplet die(s), coupled with the unpredictable behaviors in the field due to threats such as zero-day vulnerabilities [86], [87], our C2 mitigation strategy operates under the assumption that CHSM solely shares the power supply with the target chiplets (e.g., metal layers M3/M4 in Figure 10), without requiring signal connectivity (metal layers M1/M2 in Figure 10). To monitor system-level switching activities by the CHSM, we incorporate a TDC sensor for power measurements, as shown in Figure 11, which digitizes the variations in time delays within the buffer path (i.e., "initial" delay line and the "tapped" delay line). Due to the relationship between the voltage drop and the delay amount in each buffer unit, the digitized time delay can serve as an indicator of the voltage supply. Hence, the TDC sensor functions as a lightweight oscilloscope integrated into the SiP [88].

We would like to highlight that security monitoring against hardware Trojans and software malware remains an open challenge even when golden references (e.g., design layout or software code) are available. Our C2 solution here cannot serve as a silver bullet to completely address the concern. However, we aim to provide the community with a foothold to mitigate the issues in the era of heterogeneous integration. The underlying reason is that the supply chain and device architecture of heterogeneous integration-based SiPs are becoming even more complicated than their SoC counterparts. Conventional golden information is less likely to be procured by the trusted SiP integrator. Typically, only black-box silicon dies are expected to be purchased along with high-level product specifications and manuals, rendering most conventional golden information-based Trojan detection methods useless. Therefore, our methodology assumes hardware/software applications on chiplets can be golden because the chiplets are offline while hardware Trojans are

mostly dormant at the integration stage, as explained in Section VIII-B. With the benign power signatures extracted by SiP integrators, run-time security monitoring can be enabled effectively, as demonstrated in Section VIII-B.

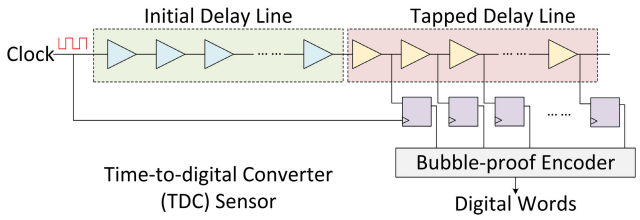


FIGURE 11. Top view of TDC sensor in CHSM for power measurements.

The C2 framework consists of two primary stages, as depicted in Figure 10: (1) application profiling, and (2) in-field security monitoring, whose details are as follows.

1) APPLICATION PROFILING

In this stage, we assume software apps have no control flow integrity violations, and hardware apps are either Trojan-free or Trojan-dormant. Thus, we can profile power fluctuations caused by their activity as *reference traces* using the TDC sensor, as transistors switch on/off during execution, generating distinct current spectra from the power supply.

Given the complexity of these patterns and CHSM's resource constraints, conventional methods like look-up tables [89] are not suitable for modeling and storage. Instead, we use machine learning (ML) models to capture these patterns within reference traces. These ML models serve for application profiling and in-field inference. Figure 12 illustrates the complete application profiling process, comprising four stages: (i) deep learning training, (ii) model parameter profiling, (iii) high-level synthesis (HLS) model conversion, and (iv) HLS and FPGA design compilation,[6] as follows:

(i) Deep Learning Training: It constructs a training dataset, which consists of pre-processed reference traces of the application (to fit the target ML model). For the ML model, we utilize a multi-layer perceptron (MLP) that offers a relatively straightforward structure, resulting in a smaller overhead [90]. Throughout this work, we have employed the rectified linear unit (ReLU) [90] as the activation function in our MLP model, which provides faster computation and reduced likelihood of encountering vanishing gradient problems. For iterative error measurement, we utilize the mean squared error (MSE), which is a commonly employed metric in training and timing series anomaly detection settings.

(ii) Model Parameter Profiling: With the trained model in floating-point, which cannot efficiently map to FPGA fabric, quantization to fixed-point is crucial. This involves intelligently selecting fixed-point data types for each layer, balancing accuracy preservation and resource efficiency.

[6]As CHSM is ultimately deployed on FPGA fabric, we employ FPGA-based HLS implementation for the ML model implementation.

(iii) HLS Model Conversion: To transfer the ML model into an HLS entry, We use the open-source HLS4ML framework [91], allowing us to achieve automatic ML-to-HLS translation with fine-grained optimization and eliminate the need for extensive expertise, thereby removing implementation barriers.

(iv) FPGA-based HLS Compilation: The C/C++ HLS model generated by the HLS4ML framework can be further processed by HLS tools to produce the corresponding RTL, then into the FPGA bitstream for the CHSM integration.

2) SECURITY MONITORING

As shown in Figure 13, the security monitoring unit, alongside the TDC sensor and ML engine, includes vital components: a FIFO buffer, interface module, error calculator, and deviation analyzer. The procedure of this security monitoring unit against SiP architecture malware/ransomware threats, as depicted in Figure 7 (C2-1 through C2-6), is described below:

Step C2-1: Once the target application is initiated, the TDC sensor can be triggered by a flag originating from the chiplet under monitoring (or by analyzing the captured waveform to achieve trace-behavior synchronization). Then, the output of the TDC sensor will be stored in the FIFO buffer.

Step C2-2: When the FIFO buffer is full, it starts sending the elements to the ML interface. This interface manages control signals and status updates between the FIFO buffer and the ML engine. The interface also handles pre-processing for incoming TDC outputs. Afterward, we activate the ML engine by de-asserting its reset signal, allowing it to generate predictions using trained parameters.

Step C2-3: The activated ML inference processes FIFO buffer data continuously until the buffer becomes empty.

Step C2-4: We use the prediction to calculate errors by comparing it to reference data stored in the FIFO buffer (Figure 13(a)). These errors are stored in a FIFO buffer within the 'deviation analyzer' module, accumulating individually until full, at which point we update the total accumulated error.

Step C2-5: When the accumulated error exceeds a user-defined threshold, we flag the corresponding timestamp as an anomaly. Figure 13(b) outlines our threshold determination strategy. We employ the RTL model from the application profiling phase in functional simulation to generate predictions for unseen benign testing data. By quantifying errors and assuming they follow a Gaussian distribution, we set the threshold using the 3-σ rule [92] to achieve a 99.7% confidence level and reduce false positives. This threshold is subsequently used in the security monitoring process.

Step C2-6: Upon successful malicious anomalies detection, a set of security measures (tampering) by the SiP architecture must be executed (such as erasing sensitive on-chip security assets or resetting the entire systems).

D. C3 MITIGATION AGAINST SIP REVERSE ENGINEERING

To enhance the protection of SiPs against reverse engineering and IP piracy, we propose a dual-tiered approach that

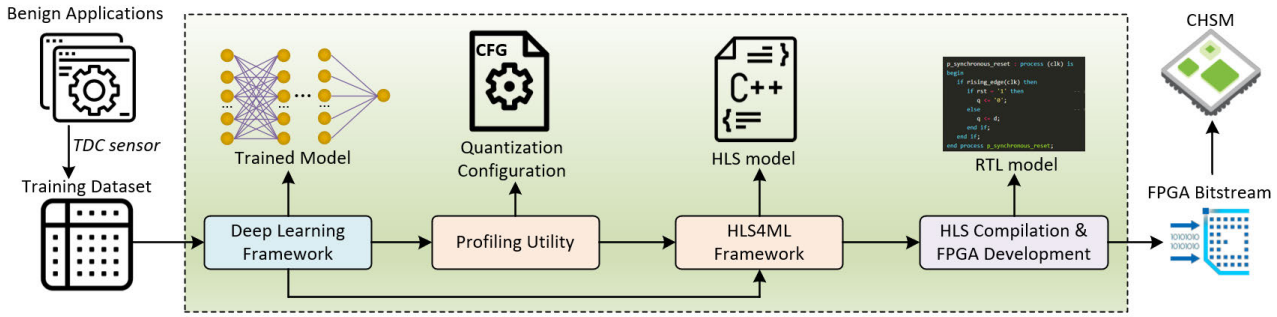


FIGURE 12. Application profiling phase encoding the reference behaviors (reference traces) of benign applications in ML models for malware mitigation.

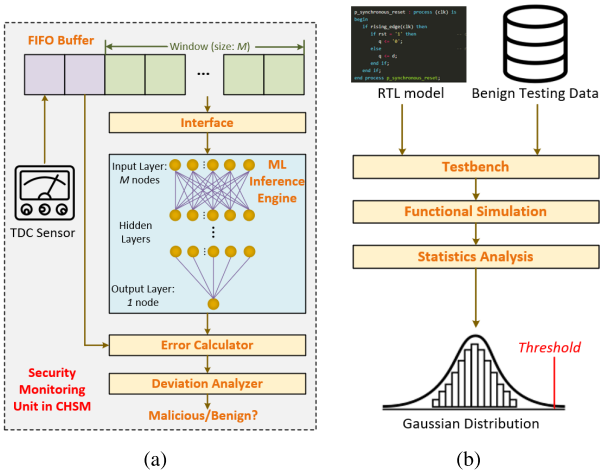


FIGURE 13. The security monitoring phase: (a) Security Monitoring Unit, (b) Threshold Determination (The differentiation of anomalies can be determined by utilizing rtl analysis to set the appropriate threshold).

leverages CHSM. This approach strategically combines chiplet-level obfuscation with overarching system-level security measures, particularly addressing supply chain vulnerabilities as highlighted in [93]. Figure 14 outlines the revised SiP supply chain protocol incorporating these dual protection mechanisms. The following details the important phases of this process and the integration of the dual-tiered security strategy:

(1&2&3) Design and Obfuscation of Chiplet: This phase encompasses the entire development cycle of chiplets, from conceptualization to their integration, including design, implementation, and obfuscation. Obfuscation entails analyzing the critical parts of the design and pinpointing specific design segments that require protection. This framework applies a hybrid obfuscation approach, employing both key-based and key-less (for the initial stage, the chiplet designer is responsible for provisioning the primary key, while in the subsequent stage, the SiP designer handles the provisioning of the secondary activation, which is keyless) as elucidated in [61] and [93]. Upon completing this step, the focus shifts to the physical design, which involves finalizing the obfuscated GDSII for fabrication, typically carried out at an off-shore foundry.

(4&5&6) First-stage Activation: This step occurs in a trusted facility of the chiplet design house after fabrication. Post-fabrication activation of the IC involves a unique registration process employing electronic chip IDs (ECIDs). The chiplet design house undertakes security evaluations, such as PUF enrollment, and loads the initial activation keys into the secure TPM. The subsequent activation key or input sequence is derived from these primary keys in accordance with the requirements of the secondary activation functions. This step sets the configuration parameters for each chiplet's activation IP (to be integrated into CHSM for run-time activation), emphasizing critical security aspects like timing and specific activation input patterns.

(7&8&9) Implementation of SiP:In this phase, the SiP designer proceeds to integrate the enrolled chiplets into the SiP architecture. They acquire obfuscated chiplets, along with all other chiplets, and retrieve the secondary keys needed for the CHSM configuration. Additionally, considering the requirements of inter-die communication, the interposer layer is constructed utilizing heterogeneous packaging, such as 2.5D or 3D integration technologies, followed by thorough electrical and timing-based verification. It's important to note that the production of the interposer may take place at an untrusted facility. After Integration, the SiP isn't completely operational since some chiplets remain locked and require activation in the field, which is based on the distinct values associated with their secondary keys (referred to as '(*Step C3-1*' in Figure 7).

(10) Second-stage Runtime Activation: During the in-field phase, the obfuscated chiplets require the provisioning of a secondary activation key. This process is distinctive and follows a cycle-specific approach based on the input sequence, specifically the license activation sequence. Within the CHSM, this activation secret is stored alongside various static security assets, such as a device-specific ID and private keys. These assets are utilized for cryptographic operations and secure communication between the CHSM and the chiplets. It is important to emphasize that, in this process, the CHSM initially performs authentication and establishes a secure communication channel to facilitate asset transfer (referred to as (*Step C3-2* in Figure 7)).

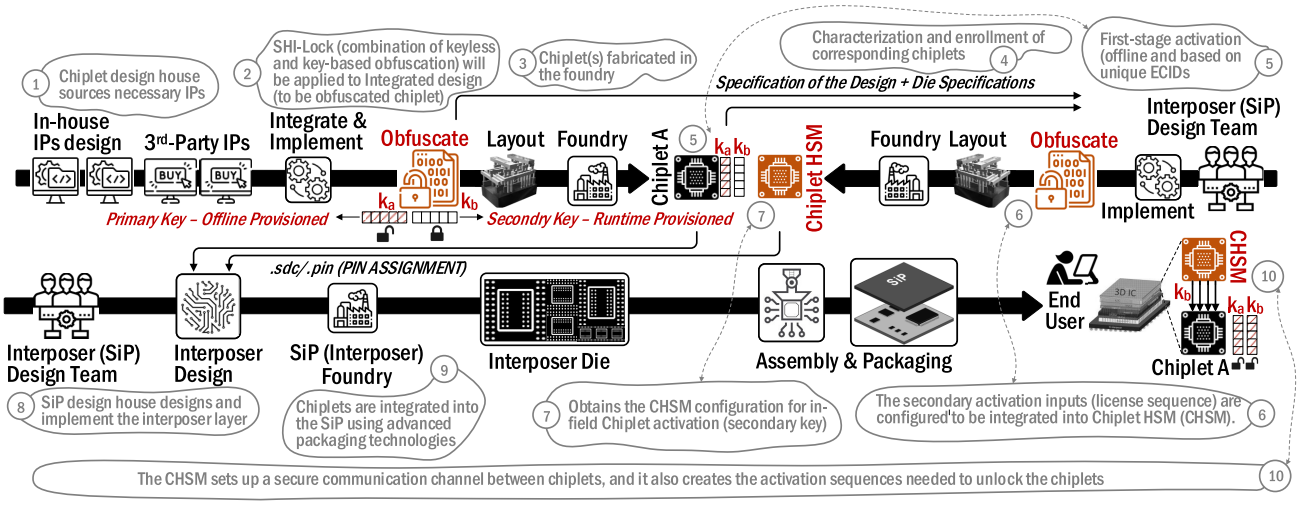


FIGURE 14. Crucial steps of co-obfuscation to enable two-stage activation in SiP supply chain.

Following this process, our proposed obfuscation method provides both the chiplet designer and SiP integrator with a secure means of integration and activation that effectively safeguards against threats in both the supply chain and in-field environments. Through the implementation of a compound obfuscation technique across the chiplet and CHSM during integration, the chiplet and SiP designs necessitate a two-factor activation approach.

1) CHSM-ORIENTED HYBRID CO-OBFUSCATION IN SIP

We have implemented a hybrid approach (drawing inspiration from ReTrustFSM [61]) in our CHSM-equipped SiP architecture that enables two-factor obfuscation and activation. This approach combines both explicit and implicit secrecy through sequential locking mechanisms.[7] Our strategy involves the provisioning of a portion of the keys, with the primary key representing explicit secrecy, applied post-fabrication to the chiplet in a trusted facility. In this obfuscation model, we employ a state encoding approach similar to the external explicit secrecy method utilized in ReTrustFSM [61]. However, we introduce a different approach for expanding the state space using implicit secrecy. This enhancement not only ensures resilience against functional I/O query-based attacks but also enables the designer to associate state transitions with various sets of input patterns (serving as secondary keys). Consequently, the chiplet designer gains the capability to offer unique instantiation for each contract, permitting distinct activation patterns based on whether the chiplet is being used in SiP design 'A' or SiP design 'B'. With this mechanism in place, even though chiplets are provisioned with the primary key, they still require a specific pattern of inputs in order to gain full functionality. Furthermore, different SiP designs employing the same chiplet should possess their own distinct sets of input patterns, mitigating the risk of confidentiality breaches. This approach allows chiplet designers to extend chiplet-level protection schemes applicable for multiple SiP designs, achieving system-level obfuscation.

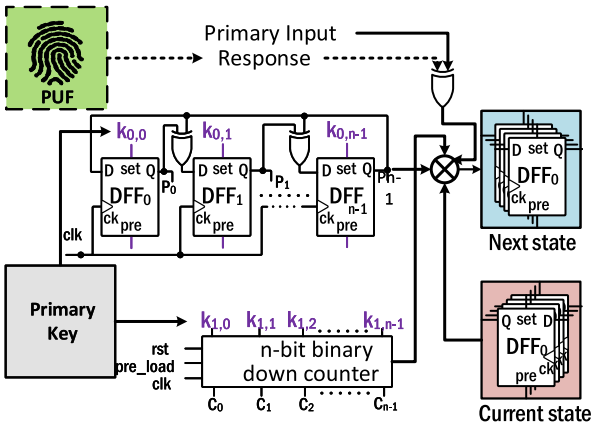


FIGURE 15. PUF-based Obfuscation Methodology at Chiplet-level.

Figure 15 illustrates the necessary architecture additions within a chiplet to facilitate the activation of the second key using our obfuscation model. This modification introduces a dependency of the chiplet's FSM, whether it's the FSM of one module or multiple modules, on a counter and an LFSR. By employing a specific and unique input pattern, which serves as the second key, the state of the counter and LFSR can be effectively utilized to initiate the targeted FSM state within the chiplet. Much like the approach in ReTrustFSM, the newly integrated obfuscated components, represented by the obfuscated FSM, seamlessly merge with the original FSM. This integration results in a strongly connected FSM structure that offers robust resistance against various forms

[7] It is important to note that the application of such a technique within the SiP domain necessitates the development of a new definition model that will enable the distribution of this process across multiple stakeholders within the SiP supply chain.

of structural attacks, including removal attacks and their derivatives [94].

Figure 16 illustrates a typical outcome of the obfuscation process applied in our model. In essence, to enable or properly initiate the controller of the obfuscated chiplet, a unique secondary key, functioning as the primary input, must be applied to the FSM in a cycle-accurate manner. The responsibility for constructing such a cycle-accurate activation rests with the CHSM. This specific input sequence is designed to place both the LFSR and the counter into a predetermined state, which serves as the activation license for the second-factor form of activation (as depicted in Fig. 16). It's important to note that the primary key (representing explicit secrecy) serves as the primary initialization for both the LFSR and the counter. Consequently, these two keys, the primary and secondary, establish a strong interconnection, and the activation process cannot proceed without both keys Much like the FSM-oriented obfuscation techniques, our implicit secrecy (represented by the secondary key) encompasses a comprehensive sequence of input patterns required for the successful completion of a full round within the FSM. In simpler terms, it includes the specific sequence of input patterns necessary not only to traverse the encFSM (depicted in Figure 16) but also to return to the initial state, completing a full cycle. To introduce uniqueness across various target System-in-Package (SiP) designs, we enhance the traversal of encFSM using a device-specific PUF fingerprint. This allows SiP integrators to activate the chiplet directly by leveraging activation challenges provided by the chiplet designer. To generate a chiplet-specific distinct input sequence, we employ an XOR cipher along with an n-bit PUF response (truncating or concatenating as needed). This process transforms the required primary input pattern into a set of values such as $i_1 \oplus R$,$i_3 \oplus R$, …, where i_1, i_2, i_3,… represents the original input sequence, and R denotes the PUF response (adjusted to match the required bit width). Assuming the PUF responses are unique, with an average inter-chiplet HD of 50%, we can reasonably expect the required input patterns to be unique across all manufactured chiplets (as illustrated in Figure 15).

To address potential PUF instability resulting from varying environmental conditions, one approach is to utilize error correcting codes (ECC) [95]. However, implementing ECC introduces additional overhead (area, power, timing). An alternative strategy involves the careful selection of more reliable CRPs from a pool of previously assessed CRP-space. By making judicious choices, it becomes possible to reduce reliance on ECC. Furthermore, incorporating multiple redundant challenges can bolster reliability and entirely eliminate the need for ECC. It's important to note that our approach focuses on a narrower selection of responses while preserving the PUF's entropy.

E. C4 MITIGATION AGAINST SIP COUNTERFEIT

By integrating CHSM with blockchain, a distributed network, we can ensure the integrity of System-in-Package (SiP) devices from manufacturing to end-of-life. This involves

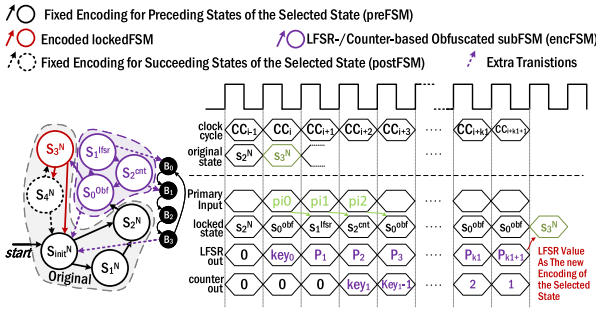


FIGURE 16. Obfuscated FSM and Cycle-accurate Activation (Traversal of encFSM) using the Secondary Key.

enrolling and securely storing SiP data in the blockchain, allowing trusted supply chain entities to identify and address counterfeit SiP threats as discussed in Section V-D.

Blockchain technology, known for its transparency, resistance to tampering, and scalability, has found applications in multiple sectors like healthcare, art [96], and currency [97]. Consortium blockchains, specifically, offer several advantages in supply chain interactions. Various frameworks and implementations [98], [99], [100], [101], [102], [103], [104], [105] have demonstrated their ability to provide assurance to ICs by enabling traceability and tracking during production and field deployment.

Depending on the threat model and overhead constraints, trusted entities use various blockchain techniques to register verified chip data across all nodes' ledgers. Blockchain's tamper-resistant nature allows any entity to verify an IC by comparing an IC's information with the true data stored in the blockchain. If discrepancies exist, the IC is likely counterfeit and requires inspection or disposal. Also, consortium-style blockchains are advantageous for their ability to grant permissions to numerous supply chain participants, not all of whom may be inherently trusted. These permissions determine who can register or verify information in the blockchain.

CHSM, used alongside blockchain, ensures SiP integrity and provenance. As shown in Figure 7, each CHSM includes a CDIR for identifying recycled ICs when paired with blockchain [103]. The CHSM's age serves as a proxy for the chip's age since they integrate at the same time as the CDIR sensor activation. Firstly, the SiP designer creates the CHSM design with the CDIR sensor architecture [31], [106], choosing sensor types based on factors like overhead, sensitivity, technology node, and CHSM size. After SiP assembly, objective and threshold values, along with a starting usage count, are determined through statistical modeling and testing, and these can be recorded in the blockchain. The CHSM's unique ECID is also essential to identify each SiP's data in the blockchain, and it can also securely communicate with chiplets and blockchain nodes for verification.

Integration of the SiP blockchain involves several steps in establishment. Initially, the SiP designer configures the

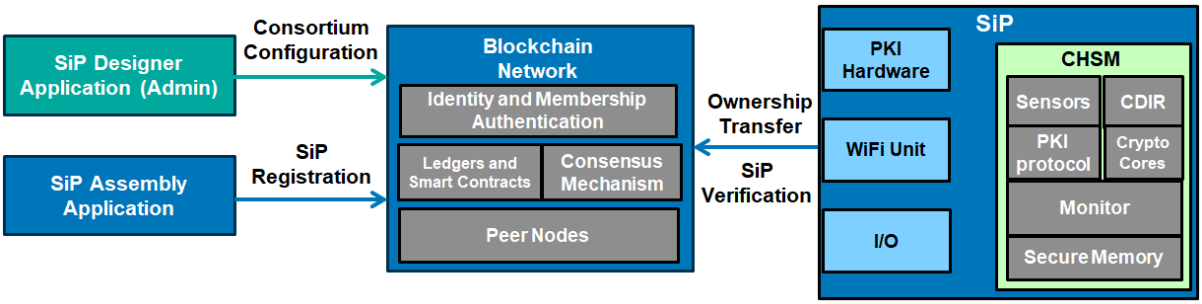


FIGURE 17. High-level view of blockchain framework utilizing CHSM against counterfeiting.

blockchain network, setting up peer nodes and permissions based on the threat model. After the SiP assembly, both the SiP designer and assembly collaborate to register the SiP in the blockchain. They use the CHSM's CDIR sensor values to detect recycled SiPs and cross-reference them for verification. Additionally, expected grade codes, part numbers, and documentation are registered to identify remarked SiPs and forged documentation. Also, the resistance of CHSM to RE prevents unauthorized cloning/replication (C3 mitigation). With these steps of establishment in place, SiP verification proceeds through the following stages outlined in Figure 7:

Step C4-1: To verify an SiP throughout its life, the CHSM reads the SiP's current state, which is later securely communicated via TLS to the blockchain; the CHSM operates as the trust anchor to verify the SiP. In this step, the CHSM gathers the usage time count from CDIR to send in a request for verification in the third step. The CDIR sensor's current usage time is read to the bus and into the processing unit.

Step C4-2: CHSM gathers more information to use in the verification request. Here, supply chain parties input information via IO ports for data fields that the CHSM cannot establish, e.g., electrical measurements or grade code. The CHSM communicates with the IO ports through another chiplet, procuring the inputted values for use in the verification request.

Step C4-3: The CHSM securely communicates with blockchain nodes using RSA encryption, whose keys are stored securely internally. It sends a verification request with its ECID, CDIR sensor usage time, and values from IO ports (e.g., grade code, part number, electrical measurements, and documentation). The blockchain processes the request, executing a smart contract that reads the blockchain ledgers for the requested SiP and compares the information.

Step C4-4: The blockchain nodes respond to the CHSM. If the chip is counterfeit, it's flagged for disposal, and both the CHSM and SiP designers are notified.

Apart from SiP registration and the aforementioned verification using CHSM, other techniques and benefits can be identified with this proposed solution. Ownership transfer, seen in Cui's work [99], is utilized by the blockchain to maintain the current owner of the chip and can aid in effective tracking to prevent human error, thefts, losses, etc. This logs a two-step chip owner change during shipping to ensure a seamless and secure asset transfer. A high-level view of the framework, including smart contracts and CHSM, is provided in Figure 17. As the CHSM and blockchain can both be configured to meet the SiP designer's specifications, the proposed approach is catered to the application. For example, if the packaging entity is trusted, then package marking information can be utilized for more robust assurance. This aids in the resiliency of the system to unforeseen threats.

F. C5 MITIGATION AGAINST FAULT INJECTION

Among existing research studies for the identification and mitigation of FI, where sensors are used primarily to monitor changes in electrical parameters [107], [108], [109], [110], [111] in 2D ASIC designs [112], [113], there is a notable lack of countermeasures in the context of SiP. To address this shortcoming, we integrate a two-stage comprehensive framework into the CHSM to detect FI and tampering attempts within an SiP. The first stage involves simulation-based sensor placement and EDA tool validation at the pre-silicon level, while the second stage defines CHSM FI detection capabilities at the post-silicon level. Relying on Figure 18, following describes the details of pre-silicon stage:

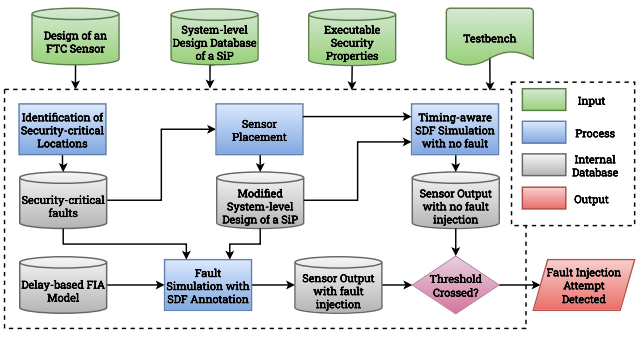


FIGURE 18. EDA-based Framework to Detect FI Attacks in an SiP (at Pre-silicon).

1) IDENTIFICATION OF SECURITY-CRITICAL LOCATIONS

This step is based on potential system-level security threats. To do that, we adopt the criticality analysis (fan-in circuit extraction and gate-level fault simulation) used in [114].

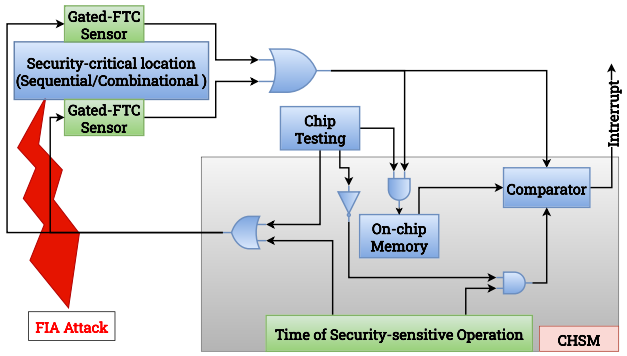


FIGURE 19. The Functionality of CHSM against FI at Post-silicon.

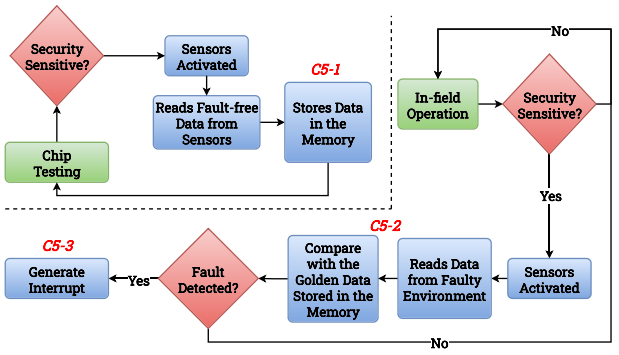


FIGURE 20. Flow of FI Attack Detection by CHSM.

It is noteworthy that such a criticality analysis by the SiP integrator is applicable at the interposer level. At the chiplet level, a high level of trust is imperative between the chiplet designer and the SiP integrator (white-box modeling - see Section V-E)).

2) SENSOR PLACEMENT

Based on the identified locations, we place Fault-to-Time-Converter (FTC) sensor due to its ability to detect various FI attempts (e.g., clock and voltage glitches, EM faults, and laser faults) with minimal overhead [88].

3) VERIFICATION OF SENSOR DETECTION

In the modified SiP design with sensors, we create a reference 'golden database' of sensor outputs using standard delay format (SDF) for timing analysis. We then model various faults (based on alterations in the propagation delays of standard cells)and perform SDF-based timing analysis. By comparing the outputs of faulty with that of golden database using a predefined threshold, we verify the framework fault detection capability.

After pre-silicon verification, at post-silicon, we integrate our proposed framework, demonstrated for a single security-critical circuitry in Figure 19, with the functionality of CHSM. It is noteworthy that we enhance the FTC sensors by incorporating clock-gating sub-circuitry, optimizing power consumption by activating the sensors only during security-sensitive operations. Given this architecture, the overall flow of FI detection by CHSM is illustrated in Figure 20, which elaborates on the following sequence of steps illustrated in Figure 7.

Step C5-1: CHSM reads fault-free data from the sensors during SiP testing after fabrication and stores them in TPM (any unauthorized access is restricted) as a *Golden Database* (e.g., D4 of Figure 7). CHSM also safeguards the integrity of the *Golden Database* throughout in-field operations by preventing any modifications. Sensors are only activated if any security-critical operation starts at SiP testing.

Step C5-2: CHSM reads data from the SiP (in case of security-critical operation) via sensors, and a comparison with the corresponding *Golden Database* will be executed.

Step C5-3: Finally, if an anomalous sensor reading is detected, CHSM generates an interrupt and transfers it to the processing unit. Upon receiving this interrupt, the processing unit halts any ongoing security-sensitive operations or safeguards security-sensitive data from potential compromise.

VIII. EXPERIMENTAL RESULTS AND EVALUATION

Since our solution is tailored for chiplet-based SiP architectures, we performed experiments on an ARM MPS3 FPGA platform [115], which uses a Xilinx Kintex UltraScale 115 (KU115) FPGA, as illustrated in Figure 21. The KU115 FPGA comprises two super logic regions (SLRs), essentially representing two distinct silicon chiplets residing on the same interposer. The following section provides implementation details per each case on this platform, along with a thorough examination of the security-related outcomes.

A. C1 MITIGATION IMPLEMENTATION AND EVALUATION

Since C1 aims to build a secure communication channel by authenticating trusted chiplets and generating a shared secret key for each session between these chiplets and the CHSM, to assess the effectiveness of our proposed protocol (against probing attacks), we considered three key properties:

(i) *Information Concealment:* Data transferred in plain text should not divulge any information about the shared secret.

(ii) *Attack Resistance:* The protocol should exhibit resilience against various types of attacks.

(iii) *Response Secrecy:* The attacker must not be able to gain any meaningful information from the responses.

To realize the implementation of our protocol on an ARM MPS3 FPGA platform, we employed the K-283 elliptic curve for the ECC module [76], utilized SHA256 for hashing, used a 128-bit ring oscillator (RO) based TRNG, and integrated a 256-bit SRAM PUF [116]. Relying on the fundamentals of these components, while integrated with the designed signature generator and corresponding controller, the following assesses the achievement of the three properties listed above.

1) INFORMATION CONCEALMENT

Random challenges (SD_1, C_1, and n_1 in Figure 8) are generated using a TRNG and differ per each communication

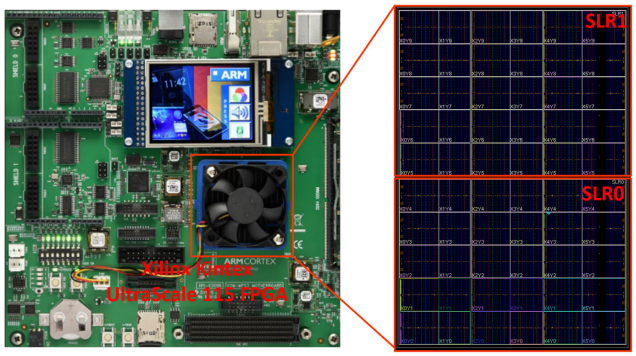


FIGURE 21. ARM MPS3 FPGA Board Featuring a Xilinx Kintex UltraScale 115 FPGA with Two Individual Chiplets (Super Logic Regions (SLRs)).

session. The n_1 value is XORed with the PUF response PR and is subsequently hashed. This process ensures that even if an attacker knows both n_1 and the PUF challenge PC, they cannot deduce any information about the PUF response due to the one-way nature of the hash function. Concurrently, SD_1 and C_1 are applied to the signature generator, generating R_1, which is then XORed with the chiplet's message. As SD_1 and C_1 are both random and unique for each session, the attacker can only access them after intercepting chiplet-CHSM communication by probing during its initial encounter. For an attacker to disrupt the authentication protocol, they need to execute impersonation attacks after recovering the challenges to deceive the CHSM and establish a shared secret key. However, the following depicts the proposed protocol is resistant to such attacks. Furthermore, G and Q_b are public information and do not pose any threat if disclosed [76].

2) SECURITY ANALYSIS OF THE PROTOCOL

The following sections present a detailed analysis of the security measures implemented to counter main attacks.

a: RESILIENCE AGAINST MAN-IN-THE-MIDDLE ATTACKS

This attack intercepts the CHSM and chiplet communication by probing and observing/modifying the transferred data to recover the secret key. In our architecture, the challenges do not reveal any information regarding the secret key by simply observing them. Modifying SD_1 and C_1 results in generating a distinct R_1, which is detectable at the verification stage (reference check at C1-5). Similarly, tampering with the PUF challenge PC can also be identified at C1-5, leading to a failed authentication. Attempting to modify the chiplet's response without knowing R_1 will inevitably result in authentication failure.

b: RESILIENCE AGAINST IMPERSONATION ATTACKS

This attack disconnects the CHSM from the chiplet and establishes a physical connection with an impersonator. The attacker has three options for impersonators: (i) a software-based simulation program, (ii) an FPGA-based emulation, or (iii) an overproduced or newly purchased chiplet. Assuming the attacker manages to recover the signature generator architecture (distinct for each chiplet requiring RE to extract the netlist), for the first two cases, they must cycle/time-accurately simulate or emulate it to derive R_1 after capturing SD_1 and C_1 during transmission.[8] Our gate-level simulation using Synopsys VCS, with varying C_i while keeping SD_i constant, reveals a significant time requirement for generating R_1 (see Table 2), making it challenging to maintain cycle/time accuracy[9],[10] Also, chiplets, often manufactured using advanced process nodes, outperform FPGA-based emulation, complicating cycle/time-accurate impersonation [117]. In the third scenario, if the attacker connects an overproduced or newly acquired chiplet to the CHSM, the PUF response crafted by the attacker will diverge from the response computed within the CHSM using the stored PUF CRP.

TABLE 2. Simulation runtime for various capture cycle count.

# of Cycle*	15	45	75	105	135	165	195	225	255	285	315
Sim Time+	0.99	2.13	3.29	4.48	5.62	6.87	7.97	9.14	10.34	11.51	12.63

*: $(C_i \times 10^3)$ Simulation Runtime in (*second*)

c: RESILIENCE AGAINST REPLAY ATTACKS

This attack leverages prior CHSM-chiplet sessions to gain unauthorized access or authenticate rogue chiplets. Our protocol guarantees a new session for each security asset transfer, with CHSM generating unique values (SD_1, C_1, and n_1). This ensures variability in chiplet response, PUF response, and shared secret key, making replay attacks impractical.

d: RESILIENCE AGAINST PRE-COMPUTATION OF DATABASE

This attack attempts to exhaustively simulate the signature generator using all input combinations, build a signature database, and then search for the matching R_1 when obtaining challenges through probing. However, our signature generator uses a 128-bit random SD_1 and a 32-bit random C_1, resulting in a vast key space of 2^{160}. This makes database recreation practically infeasible, and the chance of finding a match within the threshold time T_i is minimal.

3) RESPONSE SECRECY

For this study, we generated 1000 responses by utilizing randomly generated values for SD_1, C_1, and n_1. Subsequently, we computed the hamming distance (HD) between signatures and responses, as shown in Figure 22. Our analysis

[8]The attacker generates R_1 first, decrypts the chiplet's response, replaces Q_a and $H(Q_a)$ with their values, combines them with $H(PR_e)$ (as the PUF response is unknown), and sends this manipulated response to the CHSM.

[9]CHSM can distinguish between the genuine chiplet and the imposter by setting a threshold time T_1 (Based on the timing of the genuine chiplet).

[10]In our protocol, generating the response in 553 cycles, assuming a chiplet operates at 1-2GHz, takes only 0.277-0.554 μ s, significantly less time than simulating the standalone signature generator.

demonstrates that, in each session, both signatures and responses exhibit distinct and random characteristics (with an approximate HD of 50%). As a result, the response does not divulge any insights into the chiplet's signature or secret key.

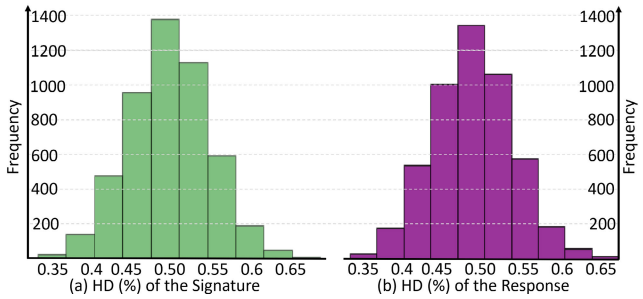


FIGURE 22. Hamming Distance (%) Ratio of (a) Signatures, (b) Responses.

TABLE 3. Model training and security monitoring statistics of mibench software applications against ransomware intrusion.

Benchmark	ML Model	ML Train	Security Monitoring		
	M-L-1)	Accuracy	Precision	Recall	Th_{err}
basicmath	16-64-32-16-1	$8.45e^{-4}$	1.00	1.00	$4.11e^{-3}$
	32-64-32-16-1	$6.23e^{-4}$	1.00	1.00	$2.83e^{-3}$
	16-128-64-32-16-8-1	$4.73e^{-4}$	1.00	1.00	$2.25e^{-3}$
bitcount	16-64-32-16-1	$8.31e^{-4}$	0.98	1.00	$1.01e^{-2}$
	32-64-32-16-1	$5.15e^{-4}$	1.00	1.00	$6.23e^{-3}$
	16-128-64-32-16-8-1	$2.18e^{-4}$	1.00	1.00	$3.58e^{-3}$
qsort	16-32-16-8-1	$6.87e^{-4}$	1.00	1.00	$2.86e^{-3}$
	32-32-16-8-1	$5.62e^{-4}$	1.00	1.00	$2.81e^{-3}$
	16-64-32-16-8-4-1	$3.77e^{-4}$	1.00	1.00	$2.73e^{-3}$
SHA	16-32-16-8-1	$2.88e^{-4}$	1.00	1.00	$4.11e^{-3}$
	32-32-16-8-1	$2.79e^{-4}$	1.00	1.00	$4.08e^{-3}$
	16-64-32-16-8-4-1	$2.16e^{-4}$	1.00	1.00	$3.81e^{-3}$

B. C2 MITIGATION IMPLEMENTATION AND RESULTS

With the use of ARM MPS3 FPGA platform for our prototyping, featuring KU115 FPGA, two building chiplets (SLRs) are interconnected using the 2.5D Xilinx stacked silicon interconnect technology (SSIT) [118]. In C2 mitigation requiring an ML model, for the ML training phase, we utilize Tensorflow in Python to train our generic ML model on an Nvidia GTX 1660 ti GPU. HLS4ML [91] is used to (i) convert the trained floating-point model to its fixed-point counterpart (input of Xilinx Vivado 2019.1 HLS) and (ii) to compile the resulting RTL implementation of the ML model, along with other units depicted in Figure 13, into bitstreams. For our experiments, the target designs (chiplets) encompass a Microblaze microprocessor and an AES-GF accelerator mapped on SLRs, verifying the integrity of software/hardware applications [119].

1) SOFTWARE APPLICATION COMPROMISE

In our experiments, to reenact ransomware attacks [10], we consider the Microblaze microprocessor to be operating in bare-metal mode. We designate Mibench's four embedded applications as benign programs [120], namely *basicmath*, *bitcount*, *qsort*, and *SHA*, while a software AES-128 implementation is employed as a potential ransomware variant.[11] Table 3 reflects the detection[12] results related to model training and security monitoring in the ML inference engine to profile all four Mibench programs in terms of their susceptibility to ransomware intrusion. With two sets of reference profiles for each benchmark (100 ransomware traces and 100 testing benign application traces), we calculate *precision* and *recall* rates, denoted as $\frac{TP}{TP+FP}$ and $\frac{TP}{TP+FN}$, respectively, where TP is the number of true positive cases, FP refers to false positive cases, and FN is false negative cases. As shown, the proposed architecture can accurately identify ransomware intrusions by distinguishing them from benign applications without any *false positives* (both *precision* and *recall* rates are 1.00.).

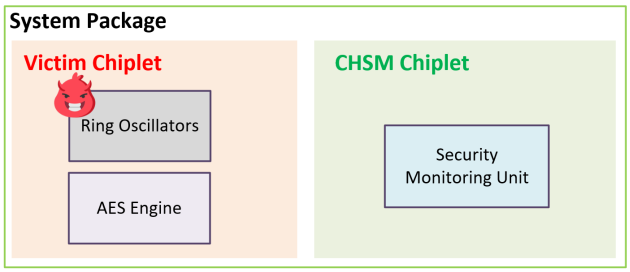
To achieve a balance between overhead and precision, we provide experimental results for three ML model options, which are (i) baseline, (ii) wide, and (iii) deep. As shown, per each application, e.g., *basicmath*, three ML model structures are used, i.e., 16-64-32-16-1 (baseline), 32-64-32-16-1 (wide), and 16-128-64-32-16-8-1 (deep). For ML training, we apply 10-fold cross-validation and use the validation mean squared error (MSE) for accuracy measurement. For instance, the average error for each prediction sample of the 16-64-32-16-1 model of *basicmath* is calculated as $\sqrt{accuracy} \times TDC_{max} = \sqrt{8.45 \times 10^{-4}} \times 63 \approx 1.83$. The error can be reduced to 1.57 and 1.37 via the wider 32-64-32-16-1 and deeper 16-128-64-32-16-8-1 models, respectively.

2) HARDWARE APPLICATION COMPROMISE

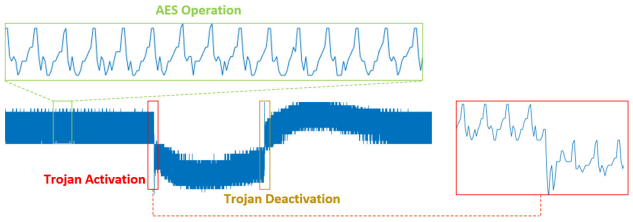
For hardware comprise, the case study aims to identify activated hardware Trojans, as inactive Trojans usually generate negligible power traces [121], [122]. Here, we concentrate on a particular malicious ring oscillator (RO) array, which consists of an odd number of inverters, creating an unstable circuit configuration that leads to self-oscillation. The oscillation frequency of a ring oscillator can be extremely fast as it relies on the delay of chained inverters, resulting in a higher driven current requirement. Such circuits can intentionally be inserted by Rogue foundries during the engineering change order (ECO) phase prior to chiplet fabrication. Although decoupled, the RO array activation would cause a significant time-derivative of the current $L\frac{dI}{dt}$, where L represents the device-level inductance. Activation is observed as an undershoot in voltage at the power

[11] Ransomware attacks typically employ encryption/decryption algorithms with a secret key known only to the adversaries

[12] The successful detection of such an attack demonstrates the capability of our sensor and the deployed ML inference engine to accurately identify deviations in cross-chiplet fluctuations caused by the potentially malicious AES program in contrast to the benign Mibench applications.



(a) The Floorplan of Components in the victim and CHSM chiplets.



(b) AES Cross-chiplet fluctuations before/after.

FIGURE 23. Trojan Detection by Power Noise Variations in CHSM.

supply, corresponding to the Trojan Activation moment (see Figure 23).

Also, this incident follows a drop in TDC outputs in the CHSM, making the IR drop induced by the RO array dominant during the interval between Trojan activation and deactivation. If the voltage drop is sufficient and timed correctly, it can result in faults in the AES circuitry by increasing the path delay and violating the design's timing constraints. During our experiment, the RO array is activated for 3000 clock cycles, and 100 traces of repetitive AES operations are collected as the Trojan-positive set. The trained 16-32-16-8-1 MLP model can effectively distinguish these patterns from a separate collection of 100 AES traces that are free from activated Trojans without any false positives or negatives.

C. C3 MITIGATION IMPLEMENTATION AND RESULTS

The evaluation of our obfuscation methodology was aimed at withstanding sophisticated oracle-guided BMC (Bounded Model Checking) attacks, assuming that attackers have access to both the locked netlist and the corresponding unlocked chiplet. Given the sequential characteristics of our obfuscation approach, traditional SAT-based oracle-guided deobfuscation methods prove ineffective. We subjected our strategy to rigorous BMC attack scenarios, incorporating sequential loop unrolling, to assess its defense capabilities against such security threats. Moreover, the unique design of our hybrid obfuscation technique, which intricately interweaves the state transition and state encoding of the functional FSM with the locked FSM, demonstrates a strong inherent defense against oracle-less removal attacks (which are effective on gate-level obfuscations), as outlined in [93]. To assess the effectiveness of our CHSM-enabled architecture against RE and overproduction, we conducted an evaluation by mapping a selection of established benchmark circuits from ITC'99 [123] and an SoC onto the ARM MPS3 FPGA platform. Subsequently, we performed a series of performance and security analyses. The left section of Table 4 offers detailed specifications of these benchmark circuits, encompassing primary inputs/outputs, key information, and gate counts. Additionally, the table provides details about the candidate state present in the original FSM designs, which were the focus of our co-obfuscation strategy. All experiments are carried out on a dual AMD EPYC 7662 64-core CPU with 512GB of RAM and a maximum runtime of 24 hours. Throughout the experiments, the co-obfuscation process made use of various tools, including ABC [124], Cadence JasperGold, Synopsys Design Compiler, nuXMV [125], and Python 3.9.

Table 4 demonstrates the robustness of our co-obfuscation solution against oracle-guided BMC attacks across all benchmark scenarios.[13] Cadence JasperGold was utilized as our model checking engine [126] for BMC attacks. It's important to note that even if a BMC attack were successful, it wouldn't be sufficient to compromise our architecture entirely. If a BMC attack were to succeed, the attacker might be able to deduce a set of dises that could expose the explicit secret. However, the implicit input sequence required for correct traversal of the encFSM would remain undisclosed. As a result, an additional structure + function attack would be required for the attacker to reconstruct the state transition graphs of the co-obfuscated circuit. This added complexity presents a significant challenge to potential adversaries attempting to breach the security measures provided by our CHSM-enabled architecture.

In the event that an end user gains access to the SiP, there may be attempts to retrieve the activation input sequence, which passes from the CHSM to the chiplet through the interposer layer. Alternatively, physical attacks may be launched in an effort to access memory key values. To counter these threats, we have distributed the activation key/secret across both the chiplet and the CHSM. This means that even if an attacker manages to read the chiplet's key registers, they would still require the correct timing and secondary activation input sequence for SiP functionality. Moreover, through the implementation of measure C1, we fortify the security of communication channels within the SiP, effectively preventing unauthorized access and possible attacks on the chiplet's security mechanisms.

When transitioning from SoC to SiP architectures, it Is important to consider that multiple SiP designers may use the same chiplet in their systems. In the event that one SiP's secondary activation process is compromised, it could potentially affect others using the same chiplet. Our architecture employs unique input patterns through PUF. The randomness of these activation patterns conforms to NIST's statistical test suite [127], demonstrating a high degree of randomness. We have measured the uniqueness of input

[13] This evaluation was conducted under two scenarios: (a) when neither primary nor secondary keys were available; (b) when only secondary keys were unavailable.

TABLE 4. Resilience against oracle-guided query-based attacks (BMC) & Associated overhead on benchmark designs.

Design	Key Length	Candidate FSM %	PI/PO	Run Time	Original Gate Count	Obfuscated Design Gate Count	Overhead w/o PUF(%)
b05	16	80	3/36	timeout	~0.61K	~0.75K + PUF	25
b07	16	66.7	3/8	timeout	~0.40k	~0.56K+ PUF	40
b11	16	30	9/6	timeout	~0.36K	~0.49K + PUF	36
fib	16	60	12/8	timeout	~0.51K	~0.74K + PUF	48
RISC	16	98	252/260	timeout	~20K	~20.4K+ PUF	2

patterns masked with PUF, consistently approaching the ideal value of 50% for all cycles. While the successful activation of the architecture heavily depends on the reliability of PUF-generated responses, experimental findings demonstrate robust performance if ECC is employed. The fixed overhead linked to both PUF and the supplementary ECC (as shown in Table 5) may appear somewhat substantial for smaller designs with fewer gate counts. However, when taking into account chiplets as large designs with significantly higher gate counts compared to the benchmark designs used for proof of concept, the relative increase in overhead becomes quite insignificant.

TABLE 5. PUF area overhead details.

PUF Type	Key Length	Gate Count	Inter-Chip HD
ARO PUF [128]	128 bit	3.7K	49.67%
SRAM PUF [73]	128 bit	3.6k	48.42%
ECC	128 bit	1.7k	-

D. C4 MITIGATION IMPLEMENTATION AND EVALUATION

A prototype blockchain was implemented utilizing Hyperledger Fabric [129], a platform for producing consortium and private blockchains. The prototype is a catered implementation of Calzada et al.'s framework for blockchain to provide integrity to the SiP supply chain [130]. For prototyping our threat model, we assume the SiP assembly is trusted, while the SiP distributor and end-user are untrusted entities (see Figure 2). The architecture of this prototype contains three organizations: the SiP assembly, the SiP distributor, and the end user. Each party contains a certificate authority block responsible for maintaining the identity of each organization. Smart contracts, programmed with Go language [131], were developed, which allow organizations to interact with the SiP data stored in the blockchain ledgers. Depending on each organization's permissions, they can interface with certain smart contracts. For example, only the trusted SiP assembly can access the *createSiP* smart contract for registering new SiPs in the blockchain. All parties have access to the verification procedure; however, the CHSM supplies the necessary information to the blockchain for verification, so the untrusted entities cannot view the blockchain information or security assets. Access control of resources in network systems has been shown to augment when utilizing blockchain in tandem with an attribute-based access control (ABAC) scheme [132], [133]. In a similar fashion, Hyperledger Fabric leverages access control lists (ACLs). Policies are leveraged, allowing the identities associated with a request to be verified against the policy associated with the resource to fulfill the request. The access control can be configured solely by the trusted network admin, the SiP designer, via the configtx.yaml file affecting new channel configurations or updating access control of an active channel [134]. Hence, untrusted participants in the supply chain having only view access cannot manipulate the assets within the blockchain and are unable to alter the policies.

To evaluate the capabilities of the network to detect the counterfeit threats consistent with this example threat model, a custom script was developed which invokes the SiP assembly's smart contract *createSiP* registering 1000 SiP assets where 150 of them are or will be counterfeits throughout their lives. The verification smart contract applied both authentic and counterfeit queries to the blockchain. This simulates the CHSM sending the network queries of SiP information, which may or may not be authentic. The network successfully identified all counterfeits, which fall into the following categories based on the threat model:

1) RECYCLED SIPS

These generated counterfeit queries are applied via the verification contract, which contains high CDIR count values. As this current count surpasses the acceptable range, it creates a float value less than or equal to the threshold value enrolled under that SiP stored in the blockchain, triggering the logic within the smart contract. This implies the SiP under verification is either suspect and should be further tested or confirmed recycled and marked for disposal. Also, if an SiP marked for disposal attempts to communicate with the blockchain, the smart contract will respond with a corresponding error. The blockchain correctly identified all 50 of these cases.

2) REMARKED SIPS AND FORGED DOCUMENTATION

Through the verification smart contract, generated SiPs are applied, which contain altered grade code or part numbers that are analogous to distributors attempting to misrepresent

TABLE 6. Transaction, throughput, and latency measurements from caliper for the blockchain prototype for trusted SiP Supply chain.

Transaction Type	Successes	Failures	Send Rate (*TPS*)	Max. Latency (*S*)	Min. Latency (*S*)	Avg. Latency (*S*)	Throughput (*TPS*)
Create an SiP	5000	0	23.3	2.28	0.06	0.26	23.0
Change SiP Owner	735	0	25.2	2.07	0.06	0.26	23.6
Query an SiP	13621	0	469.2	0.06	0.00	0.01	469.1

the SiP. As they attempt to sell a different part through different part numbers or grades, this will be identified in the blockchain. The smart contract has logic that checks the equality of these values and those stored in the blockchain asset. Again, the blockchain correctly identifies all 50 of these cases.

3) UNREGISTERED SIPS AND OTHER THREATS

We anticipate certain requests may be made by unverified SiPs to access the blockchain. For example, SiPs querying with ECIDs are not in the blockchain. These are handled in the prototype. We also note that depending on the application of the SiP designer, different SiP information may be utilized to mitigate those threats. All 50 of these cases were properly identified.

Hyperledger Caliper is a benchmarking/testing platform developed by the Hyperledger Foundation to benchmark developed blockchain networks. Caliper was utilized to test the proposed blockchain prototype, and its measurements can be seen in Table 6. Through Caliper, various rounds of testing were performed with 0 failures, measuring the transaction rate, latency, and throughput of each of the different types of transactions. For enrolling a new SiP into the blockchain, the send rate (transactions per second or *TPS*) is 23.3, which would approximate 41,940 registered SiPs in 30 minutes. The ownership transfer transaction measured similar rates and latency but with a slightly increased performance. Querying an SiP had a much higher transaction rate of 469.2 tps with an average latency of 0.01s and throughput of 469.1 tps. The verification smart contract greatly leverages the querying of elements of the blockchain to perform its cross-referencing, so efficient reading is significant. The verification smart contract must read the SiP asset stored in the blockchain into a temporary data structure to compare with the data supplied by the CHSM. With an efficient querying transaction, a high-performing verification process will follow, allowing for quick and effective authentication throughout an SiP's life.

E. C5 MITIGATION IMPLEMENTATION AND EVALUATION

The security evaluation of C5 has been done through a simulation environment by utilizing the *3DIC integrity Platform* tool developed by Cadence [47] to carry out floorplanning and implementation *2.5D* heterogeneous system. The system comprises one *ASIC* die with a processor core and two *HBM* (high-bandwidth memory) dies as the chiplets that are placed on a silicon interposer interconnected through a 2×2 mesh of NoC routers. We use open-source benchmarks [135], [136] and RAK (Rapid adoption Kits) [137] provided by *Cadence* to develop the overall systems. Note that the functionality of a NoC router is implemented within the interposer layer, which serves as the white box component.

For the criticality analysis, we first define security properties related to the *Dead-flit* and *Packet header* attacks. These properties are a set of rules which, if violated, enable an attacker to compromise the security of communication (e.g., availability violation) between chiplet tiles. These security properties are: (i) The bits representing the type of a flit (e.g., header or body) must not be flipped by any unauthorized entity until the destination tile receives it; and (ii) Any unauthorized entity must not alter the destination address in a flit until the destination tile receives it.

We analyze the design's functionality, focusing on the input-port buffer of an NoC router for potential *Dead-flit* or *Packet header* attacks. We extract the circuit from the fan-in cone (FIC) [114] of these buffers (from the system-level netlist). Afterward, we define the fault targets (e.g., sequential and combinational cells) within the extracted FIC and generate a fault list. Next, We conduct fault-free machine simulations (we use *Xcelium Fault Simulator* tool developed by *Cadence* for both fault-free and fault simulations.), incorporating System-Verilog assertions representing security properties in the testbench for simulation. We also set the input stimuli initiating transactions between chiplet tiles via the NoC router. After identifying attack times during the registration of flit packets in the input-port buffers from the fault-free simulation, we perform fault simulations using the generated fault list. Assertion failures indicate security property violations, flagging associated faults as security-critical.

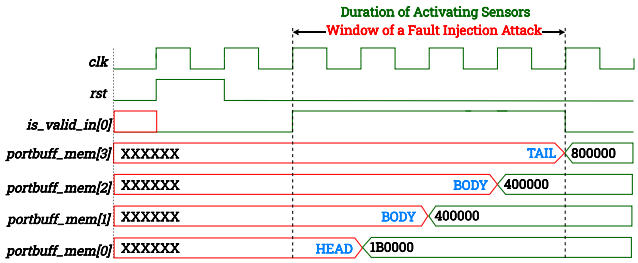


FIGURE 24. Attack Timing Window for a 4-flit Packet Transaction.

Our analysis obtains 6 sequential cells (registers) at the fan-in cone of each input-port buffer's memory of a single NoC router as the fault targets. Considering a buffer of size $8 \times 24bit$ and all possible combinations, we get $8 \times (2^6 - 1) = 504$ faults as the fault list. Failures of assertions from fault simulation results suggest that according to the 2 security

properties, 256 faults are security-critical per each input port. These faults correspond to $5 \times 8 = 40$ registers of a single input-port buffer. Therefore, for a 2×2 mesh and 5 input ports (e.g., local tile, east tile, west tile, north tile, and south tile) per each router, overall $40 \times 5 \times 4 = 800$ registers are security-critical with $256 \times 5 \times 4 = 5120$ security-critical faults. Assertion of the *valid_in* signal for each port signifies the duration of a flit-based transaction between the chiplet tiles, which is the potential timing window of a FI attack. Figure 24 illustrates this timing window of a 4-packet (e.g., 1 HEAD, 2 BODY, and 1 TAIL) transaction for *input_port*[0] of a router. According to our proposed framework, sensors need to be placed intelligently around the security-critical input-port buffer's memory locations, and CHSM needs to activate the sensors only during the duration of a transaction. However, the smart placement of the sensors and the verification of the functionality of CHSM in activating the sensors with FI attack detection are left for our future research.

IX. CONCLUSION

While heterogeneous integration (HI) brings significant benefits in power, area, and performance, it concurrently introduces a range of security vulnerabilities, both emerging and inherited from conventional monolithic SoCs. Conventional security mitigation techniques, designed primarily for SoCs, are insufficient for addressing the distinct challenges introduced by the HI supply chain and packaging technology. This paper analyzes a set of five security vulnerabilities arising from the SiP supply chain, for which the countermeasures need to be revisited. In response to these threats, we propose a novel root-of-trust chiplet called Chiplet Hardware Security Module (CHSM) for SiP architecture and explain its architecture in detail in relation to the identified attack vectors. Our work demonstrates how the CHSM effectively implements the proposed security measures to safeguard the SiP and its security assets from potential attack vectors associated with this newer technology. We also show how CHSM provides traceability throughout the SiP's lifetime and incorporates tamper-proof features to protect against various physical attacks. In our future endeavors, we aim to enhance its protective capabilities by addressing software-based attacks and detecting malicious circuits within untrusted chiplets. This will be accomplished by deploying distributed sensors and controllers to monitor critical memory locations and enforce strict access control over debug ports. Moreover, when encountering unexpected vulnerabilities like zero-day attacks, CHSM will promptly adapt by adjusting its security protocols and leveraging its reconfigurable design to retrieve updated bitstreams from trusted servers. Furthermore, we are exploring a distributed architecture approach, distributing CHSM functions across multiple chiplets to diminish the risks associated with single-point attacks.

REFERENCES

[1] IEEE EPS. (2019). *Heterogeneous Integration Roadmap–Chapter 1*. [Online]. Available: https://eps.ieee.org/technology/heterogeneous-integration-roadmap/2019-edition.html

[2] T. Li, J. Hou, J. Yan, R. Liu, H. Yang, and Z. Sun, "Chiplet heterogeneous integration technology-status and challenges," *Electronics*, vol. 9, no. 4, pp. 1–12, 2020.

[3] N. Vashistha, M. L. Rahman, M. S. U. Haque, A. Uddin, M. S. U. I. Sami, A. M. Shuo, P. Calzada, F. Farahmandi, N. Asadizanjani, F. Rahman, and M. Tehranipoor, "Toshi—Towards secure heterogeneous integration: Security risks, threat assessment, and assurance," Cryptol. ePrint Arch., Tech. Paper 2022/984, 2022. [Online]. Available: https://eprint.iacr.org/2022/984

[4] M. S. U. I. Sami, H. M. Kamali, F. Farahmandi, F. Rahman, and M. Tehranipoor, "Enabling security of heterogeneous integration: From supply chain to in-field operations," *IEEE Des. Test.*, vol. 40, no. 5, pp. 86–95, Oct. 2012.

[5] S. Bhunia and M. M. Tehranipoor, *Hardware Security: A Hands-on Learning Approach*. San Mateo, CA, USA: Morgan Kaufmann, 2018.

[6] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor, "Hardware trojans: Lessons learned after one decade of research," *ACM Trans. Design Autom. Electron. Syst.*, vol. 22, no. 1, pp. 1–23, Jan. 2017.

[7] M. Rostami, F. Koushanfar, and R. Karri, "A primer on hardware security: Models, methods, and metrics," *Proc. IEEE*, vol. 102, no. 8, pp. 1283–1295, Aug. 2014.

[8] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," *Proc. IEEE*, vol. 102, no. 8, pp. 1207–1228, Aug. 2014.

[9] Ö. A. Aslan and R. Samet, "A comprehensive review on malware detection approaches," *IEEE Access*, vol. 8, pp. 6249–6271, 2020.

[10] N. Pundir, M. Tehranipoor, and F. Rahman, "RanStop: A hardware-assisted runtime crypto-ransomware detection technique," 2020, *arXiv:2011.12248*.

[11] H. M. Kamali, K. Z. Azar, F. Farahmandi, and M. Tehranipoor, "Advances in logic locking: Past, present, and prospects," Cryptol. ePrint Arch., Tech. Rep. 2022/260, 2022. [Online]. Available: https://eprint.iacr.org/2022/260

[12] M. Li, K. Shamsi, T. Meade, Z. Zhao, B. Yu, Y. Jin, and D. Z. Pan, "Provably secure camouflaging strategy for IC protection," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 8, pp. 1399–1412, Aug. 2019.

[13] T. D. Perez and S. Pagliarini, "A survey on split manufacturing: Attacks, defenses, and challenges," *IEEE Access*, vol. 8, pp. 184013–184035, 2020.

[14] G. K. Contreras, Md. T. Rahman, and M. Tehranipoor, "Secure split-test for preventing IC piracy by untrusted foundry and assembly," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFTS)*, Oct. 2013, pp. 196–203.

[15] Md. T. Rahman, D. Forte, Q. Shi, G. K. Contreras, and M. Tehranipoor, "CSST: An efficient secure split-test for preventing IC piracy," in *Proc. IEEE 23rd North Atlantic Test Workshop*, May 2014, pp. 43–47.

[16] K. Alatoun, B. Shankaranarayanan, S. M. Achyutha, and R. Vemuri, "SoC trust validation using assertion-based security monitors," in *Proc. 22nd Int. Symp. Quality Electron. Design (ISQED)*, Apr. 2021, pp. 496–503.

[17] A. Basak, S. Bhunia, T. Tkacik, and S. Ray, "Security assurance for system-on-chip designs with untrusted IPs," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 7, pp. 1515–1528, Jul. 2017.

[18] M. S. M. Khan, C. Xi, M. S. U. Haque, M. M. Tehranipoor, and N. Asadizanjani, "Exploring advanced packaging technologies for reverse engineering a system-in-package (SiP)," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 13, no. 9, pp. 1360–1370, Sep. 2023.

[19] W. Zheng, Y. Wu, X. Wu, C. Feng, Y. Sui, X. Luo, and Y. Zhou, "A survey of Intel SGX and its applications," *Frontiers Comput. Sci.*, vol. 15, no. 3, pp. 1–15, Jun. 2021.

[20] J. Amacher and V. Schiavoni, "On the performance of arm trustzone: (Practical experience report)," in *Distributed Applications and Interoperable Systems*. Cham, Switzerland: Springer, 2019, pp. 133–151.

[21] AMD. *Secure Encrypted Virtualization (SEV)*. Accessed: Sep. 10, 2023. [Online]. Available: https://www.amd.com/en/developer/sev.html

[22] P. Qiu, D. Wang, Y. Lyu, and G. Qu, "VoltJockey: Breaching TrustZone by software-controlled voltage manipulation over multi-core frequencies," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 195–209.

[23] C. Gao, L. Luo, Y. Zhang, B. Pearson, and X. Fu, "Microcontroller based IoT system firmware security: Case studies," in *Proc. IEEE Int. Conf. Ind. Internet (ICII)*, Nov. 2019, pp. 200–209.

[24] M. Santana, "Eliminating the security weakness of Linux and Unix OSS," in *Network and System Security*. Amsterdam, The Netherlands: Elsevier, 2014, pp. 155–178.
[25] K. Murdock, D. Oswald, F. D. Garcia, J. Van Bulck, D. Gruss, and F. Piessens, "Plundervolt: software-based fault injection attacks against Intel SGX," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2020, pp. 1466–1482.
[26] L. Zussa, J.-M. Dutertre, J. Clédière, and A. Tria, "Power supply glitch induced faults on FPGA: An in-depth analysis of the injection mechanism," in *Proc. IEEE 19th Int. On-Line Test. Symp. (IOLTS)*, Jul. 2013, pp. 110–115.
[27] A. Tang, S. Sethumadhavan, and S. Stolfo, "CLKSCREW: Exposing the perils of security-oblivious energy management," in *Proc. 26th USENIX Secur. Symp. (USENIX Secur.)*, 2017, pp. 1057–1074.
[28] M. Agoyan, J.-M. Dutertre, D. Naccache, B. Robisson, and A. Tria, "When clocks fail: On critical paths and clock faults," in *Proc. Int. Conf. Smart Card Res. Adv. Appl.*, 2010, pp. 182–193.
[29] M. Dumont, M. Lisart, and P. Maurine, "Electromagnetic fault injection: How faults occur," in *Proc. Workshop Fault Diagnosis Tolerance Cryptography (FDTC)*, Aug. 2019, pp. 9–16.
[30] J. Dutertre, "Laser fault injection at the CMOS 28 nm technology node: An analysis of the fault model," in *Proc. Workshop Fault Diagnosis Tolerance Cryptogr. (FDTC)*, Sep. 2018, pp. 1–6.
[31] X. Zhang and M. Tehranipoor, "Design of on-chip lightweight sensors for effective detection of recycled ICs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 5, pp. 1016–1029, May 2014.
[32] J. H. Lau, *Chiplet Design and Heterogeneous Integration Packaging*. Berlin, Germany: Springer Nature, 2023.
[33] W.-C. Chen, C.-W. Lee, H.-C. Kuo, M.-H. Chung, C.-C. Wang, S.-K. Huang, Y.-S. Liao, C.-C. Wang, and D. Tarng, "Development of novel fine line 2.1 d package with organic interposer using advanced substrate-based process," in *Proc. IEEE 68th Electron. Compon. Technol. Conf. (ECTC)*, May 2018, pp. 601–606.
[34] S. Y. Hou, W. C. Chen, C. Hu, C. Chiu, K. C. Ting, T. S. Lin, W. H. Wei, W. C. Chiou, V. J. C. Lin, V. C. Y. Chang, C. T. Wang, C. H. Wu, and D. Yu, "Wafer-level integration of an advanced logic-memory system through the second-generation CoWoS technology," *IEEE Trans. Electron Devices*, vol. 64, no. 10, pp. 4071–4077, Oct. 2017.
[35] R. Mahajan, R. Sankman, N. Patel, D.-W. Kim, K. Aygun, Z. Qian, Y. Mekonnen, I. Salama, S. Sharan, D. Iyengar, and D. Mallik, "Embedded multi-die interconnect bridge (EMIB)—A high density, high bandwidth packaging interconnect," in *Proc. IEEE 66th Electron. Compon. Technol. Conf. (ECTC)*, May 2016, pp. 557–565.
[36] D. B. Ingerly, "Foveros: 3D integration and the use of face-to-face chip stacking for logic devices," in *IEDM Tech. Dig.*, Dec. 2019, pp. 19.6.1–19.6.4.
[37] D. R. Stauffer, J. T. Mechler, M. A. Sorna, K. Dramstad, C. R. Ogilvie, A. Mohammad, and J. D. Rockrohr, *High Speed Serdes Devices and Applications*. Berlin, Germany: Springer, 2008.
[38] D. Kehlet, "Accelerating innovation through a standard chiplet interface: The advanced interface bus (AIB)," Intel, Santa Clara, CA, United States, White Paper WP-01285-1.1, 2017.
[39] D. Das Sharma, G. Pasdast, Z. Qian, and K. Aygun, "Universal chiplet interconnect express (UCIe): An open industry standard for innovations with chiplets at package level," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 12, no. 9, pp. 1423–1431, Sep. 2022.
[40] H. Park, J. Kim, V. C. K. Chekuri, M. A. Dolatsara, M. Nabeel, A. Bojesomo, S. Patnaik, O. Sinanoglu, M. Swaminathan, S. Mukhopadhyay, J. Knechtel, and S. K. Lim, "Design flow for active interposer-based 2.5-D ICs and study of RISC-V architecture with secure NoC," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 10, no. 12, pp. 2047–2060, Dec. 2020.
[41] M. Ebrahimi, A. Y. Weldezion, and M. Daneshtalab, "NoD: Network-on-die as a standalone NoC for heterogeneous many-core systems in 2.5D ICs," in *Proc. 19th Int. Symp. Comput. Archit. Digit. Syst. (CADS)*, Dec. 2017, pp. 1–6.
[42] V. Pano, R. Kuttappa, and B. Taskin, "3D NoCs with active interposer for multi-die systems," in *Proc. 13th IEEE/ACM Int. Symp. Networks-on-Chip*, Oct. 2019, pp. 1–8.
[43] M. S. M. Khan, C. Xi, A. A. Khan, M. T. Rahman, M. M. Tehranipoor, and N. Asadizanjani, "Secure interposer-based heterogeneous integration," *IEEE Des. Test. IEEE Des. Test. Comput.*, vol. 39, no. 6, pp. 156–164, Dec. 2022.
[44] C. Xi, A. A. Khan, N. Jessurun, N. Vashisthan, M. M. Tehranipoor, and N. Asadizanjani, "Physical assurance for heterogeneous integration: Challenges and opportunities," in *Proc. IEEE Int. Symp. Phys. Failure Anal. Integr. Circuits (IPFA)*, Jul. 2022, pp. 1–6.
[45] M. Nabeel, M. Ashraf, S. Patnaik, V. Soteriou, O. Sinanoglu, and J. Knechtel, "2.5D root of trust: Secure system-level integration of untrusted chiplets," *IEEE Trans. Comput.*, vol. 69, no. 11, pp. 1611–1625, Nov. 2020.
[46] A. M. Shuvo, T. Zhang, F. Farahmandi, and M. Tehranipoor, "A comprehensive survey on non-invasive fault injection attacks," Cryptol. ePrint Arch., Paper 2023/1769, 2023.
[47] *Integrity 3D-IC Platform: 3D Design and Signoff for System-level Optimization*. Accessed: Sep. 10, 2023. [Online]. Available: https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/soc-implementation-and-floorplanning/integrity-3dic-platform.html
[48] M. S. U. I. Sami, F. Rahman, A. Cron, D. Donchin, M. Borza, F. Farahmandi, and M. Tehranipoor, "POCA: First power-on chip authentication in untrusted foundry and assembly," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, 2021, pp. 124–135.
[49] M. S. U. I. Sami, F. Rahman, F. Farahmandi, A. Cron, M. Borza, and M. Tehranipoor, "End-to-end secure SoC lifecycle management," in *Proc. 58th ACM/IEEE Design Autom. Conf. (DAC)*, Dec. 2021, pp. 1295–1298.
[50] M. Gao, M. S. Rahman, N. Varshney, M. Tehranipoor, and D. Forte, "IPROBE: Internal shielding approach for protecting against front-side and back-side probing attacks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 12, pp. 4541–4554, Dec. 2023.
[51] H. Wang, D. Forte, M. M. Tehranipoor, and Q. Shi, "Probing attacks on integrated circuits: Challenges and research opportunities," *IEEE Des. Test. IEEE Des. Test. Comput.*, vol. 34, no. 5, pp. 63–71, Oct. 2017.
[52] N. Fern, I. San, Ç. K. Koç, and K. T. Cheng, "Hiding hardware trojan communication channels in partially specified SoC bus functionality," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 9, pp. 1435–1444, Sep. 2017.
[53] T. Zhang, M. L. Rahman, H. M. Kamali, K. Z. Azar, M. Tehranipoor, and F. Farahmandi, "FISHI: Fault injection detection in secure heterogeneous integration via power noise variation," in *Proc. IEEE 73rd Electron. Compon. Technol. Conf. (ECTC)*, May 2023, pp. 2188–2195.
[54] J. Wang, S. Guo, Z. Chen, and T. Zhang, "A benchmark suite of hardware trojans for on-chip networks," *IEEE Access*, vol. 7, pp. 102002–102009, 2019.
[55] S. Naffziger, K. Lepak, M. Paraschou, and M. Subramony, "2.2 AMD chiplet architecture for high-performance server and desktop products," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 44–45.
[56] I. Yaqoob, E. Ahmed, M. H. U. Rehman, A. I. A. Ahmed, M. A. Al-Garadi, M. Imran, and M. Guizani, "The rise of ransomware and emerging security challenges in the Internet of Things," *Comput. Netw.*, vol. 129, pp. 444–458, Dec. 2017.
[57] N. Asadizanjani, M. Tehranipoor, and D. Forte, "PCB reverse engineering using nondestructive X-ray tomography and advanced image processing," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 7, no. 2, pp. 292–299, Feb. 2017.
[58] R. Torrance and D. James, "The state-of-the-art in IC reverse engineering," in *Cryptographic Hardware and Embedded Systems—CHES 2009*, C. Clavier and K. Gaj, Eds. Berlin, Germany: Springer, 2009, pp. 363–381.
[59] D. Halder, M. Merugu, and S. Ray, "ObNoCs: Protecting network-on-chip fabrics against reverse-engineering attacks," *ACM Trans. Embedded Comput. Syst.*, vol. 22, no. 5s, pp. 1–21, Oct. 2023.
[60] H. M. Kamali, K. Z. Azar, H. Homayoun, and A. Sasan, "InterLock: An intercorrelated logic and routing locking," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*. New York, NY, USA: Association for Computing Machinery, Nov. 2020, pp. 1–9, doi: 10.1145/3400302.3415667.
[61] M. S. Rahman, R. Guo, H. M. Kamali, F. Rahman, F. Farahmandi, and M. Tehranipoor, "ReTrustFSM: Toward RTL hardware obfuscation—A hybrid FSM approach," *IEEE Access*, vol. 11, pp. 19741–19761, 2023.
[62] (2020). *Apple Sues Recycling Partner for Reselling More Than 100,000 IPhones, IPads, and Watches It Was Hired To Dismantle*. Accessed: Jan. 11, 2023. [Online]. Available: https://www.theverge.com/apple/2020/10/4/21499422/apple-sues-recycling-company-reselling-ipods-ipads-watches

[63] *Intel® data center GPU max series overview*. Accessed: Sep. 10, 2023. [Online]. Available: https://www.intel.com/content/www/us/en/products/details/discrete-gpus/data-center-gpu/max-series.html

[64] (2021). *Architecture Day 2021 Presentation*. [Online]. Available: https://download.intel.com/newsroom/2021/client-computing/intel-architecture-day-2021-presentation.pdf.

[65] W. Gomes, A. Koker, P. Stover, D. Ingerly, S. Siers, S. Venkataraman, C. Pelto, T. Shah, A. Rao, F. O'Mahony, E. Karl, L. Cheney, I. Rajwani, H. Jain, R. Cortez, A. Chandrasekhar, B. Kanthi, and R. Koduri, "Ponte vecchio: A multi-tile 3D stacked processor for exascale computing," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 42–44.

[66] M. H. Khan, R. Gupta, J. Jose, and S. Nandi, "Dead flit attack on NoC by hardware trojan and its impact analysis," in *Proc. 14th Int. workshop Netw. Chip architectures*, 2021, pp. 10–15.

[67] V. J. Kulkarni, R. Manju, R. Gupta, J. Jose, and S. Nandi, "Packet header attack by hardware trojan in NoC based TCMP and its impact analysis," in *Proc. 15th IEEE/ACM Int. Symp. Networks Chip (NOCS)*, Oct. 2021, pp. 21–28.

[68] A. B. Sachid, P. Paliwal, S. Joshi, M. Shojaei, D. Sharma, and V. Rao, "Circuit optimization at 22 nm technology node," in *Proc. 25th Int. Conf. VLSI Design*, Jan. 2012, pp. 322–327.

[69] N. Mendiratta and S. L. Tripathi, "A review on performance comparison of advanced MOSFET structures below 45 nm technology node," *J. Semiconductors*, vol. 41, no. 6, Jun. 2020, Art. no. 061401.

[70] T. Zhang, J. Wang, S. Guo, and Z. Chen, "A comprehensive FPGA reverse engineering tool-chain: From bitstream to RTL code," *IEEE Access*, vol. 7, pp. 38379–38389, 2019.

[71] T. Zhang, M. Tehranipoor, and F. Farahmandi, "BitFREE: On significant speedup and security applications of FPGA bitstream format reverse engineering," in *Proc. IEEE Eur. Test Symp. (ETS)*, May 2023, pp. 1–6.

[72] D. Volya, T. Zhang, N. Alam, M. Tehranipoor, and P. Mishra, "Towards secure classical-quantum systems," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, May 2023, pp. 283–292.

[73] T. Rahman, M. K. Bepary, M. S. Ul Haque, M. Tehranipoor, and F. Rahman, "Design and security-mitigation of custom and configurable hardware cryptosystems," in *Proc. IEEE 16th Dallas Circuits Syst. Conf. (DCAS)*, Apr. 2023, pp. 1–6.

[74] N. Mehibel and M. Hamadouche, "A new approach of elliptic curve Diffie–Hellman key exchange," in *Proc. 5th Int. Conf. Electr. Eng. Boumerdes (ICEE-B)*, Oct. 2017, pp. 1–6.

[75] R. R. Ahirwal and M. Ahke, "Elliptic curve Diffie–Hellman key exchange algorithm for securing hypertext information on wide area network," *Int. J. Comput. Sci. Inf. Technol.*, vol. 4, no. 2, pp. 363–368, 2013.

[76] L. Chen, D. Moody, A. Regenscheid, and K. Randall, "Recommendations for discrete logarithm-based cryptography: Elliptic curve domain parameters," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. NIST SP 800-186, 2019.

[77] E. Rescorla, "The transport layer security (TLS) protocol version 1.3," Internet Eng. Task Force (IETF), Tech. Rep. RFC 8446, 2018.

[78] R. Della Sala, D. Bellizia, and G. Scotti, "A lightweight FPGA compatible weak-PUF primitive based on XOR gates," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 6, pp. 2972–2976, Jun. 2022.

[79] Y. Gao, H. Ma, S. F. Al-Sarawi, D. Abbott, and D. C. Ranasinghe, "PUF-FSM: A controlled strong PUF," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 5, pp. 1104–1108, May 2018.

[80] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proc. IEEE*, vol. 102, no. 8, pp. 1126–1141, Aug. 2014.

[81] A. Shamsoshoara, A. Korenda, F. Afghah, and S. Zeadally, "A survey on physical unclonable function (PUF)-based security solutions for Internet of Things," *Comput. Netw.*, vol. 183, Dec. 2020, Art. no. 107593.

[82] F. Zerrouki, S. Ouchani, and H. Bouarfa, "PUF-based mutual authentication and session key establishment protocol for IoT devices," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 9, pp. 12575–12593, Sep. 2023.

[83] H. Yoshida and A. Biryukov, "Analysis of a SHA-256 variant," in *Proc. Int. Workshop Sel. Areas Cryptography*. Cham, Switzerland: Springer, 2005, pp. 245–260.

[84] B. M. Gammel, R. Gottfert, and O. Kniffler, "An NLFSR-based stream cipher," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2006, p. 4.

[85] T. Zhang, M. L. Rahman, H. M. Kamali, K. Z. Azar, and F. Farahmandi, "SiPGuard: Run-time system-in-package security monitoring via power noise variation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 32, no. 2, pp. 305–318, Feb. 2024.

[86] S. Bezzateev, S. Fomicheva, and G. Zhemelev, "Agent-based zerologon vulnerability detection," in *Proc. Wave Electron. Appl. Inf. Telecommun. Syst. (WECONF)*, 2021, pp. 1–5.

[87] T. Zhang, M. Tehranipoor, and F. Farahmandi, "TrustGuard: Standalone FPGA-based security monitoring through power side-channel," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 32, no. 2, pp. 319–332, Feb. 2024.

[88] M. R. Muttaki, T. Zhang, M. Tehranipoor, and F. Farahmandi, "FTC: A universal sensor for fault injection attack detection," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, Jun. 2022, pp. 117–120.

[89] S.-C. Kim and E.-S. Kim, "Fast computation of hologram patterns of a 3D object using run-length encoding and novel look-up table methods," *Appl. Opt.*, vol. 48, no. 6, p. 1030, 2009.

[90] J. Brownlee. *A Gentle Introduction to the Rectified Linear Unit (ReLU)*. Accessed: Sep. 10, 2023. [Online]. Available: https://machinelearningmastery.com/

[91] F. Fahim, B. Hawks, C. Herwig, J. Hirschauer, S. Jindariani, N. Tran, L. P. Carloni, G. Di Guglielmo, P. Harris, and J. Krupa, "Hls4ml: An open-source codesign workflow to empower scientific low-power machine learning devices," 2021, *arXiv:2103.05579*.

[92] *3-Sigma Rule*. Accessed: Mar. 7, 2023. [Online]. Available: https://www.indeed.com/career-advice/career-development/3-sigma

[93] M. S. U. Haque, R. Guo, M. S. Rahman, H. M. Kamali, F. Farahmandi, and M. Tehranipoor, "SHI-lock: Enabling co-obfuscation for secure heterogeneous integration against RE and cloning," in *Proc. IEEE Phys. Assurance Inspection Electron. (PAINE)*, Oct. 2023, pp. 1–7.

[94] M. Yasin, B. Mazumdar, O. Sinanoglu, and J. Rajendran, "Removal attacks on logic locking and camouflaging techniques," *IEEE Trans. Emerg. Topics Comput.*, vol. 8, no. 2, pp. 517–532, Apr. 2020.

[95] M.-D. Yu, R. Sowell, A. Singh, D. M'Raïhi, and S. Devadas, "Performance metrics and empirical results of a PUF cryptographic key generation ASIC," in *Proc. IEEE Int. Symp. Hardware-Oriented Secur. Trust*, Jun. 2012, pp. 108–115.

[96] Z. Wang, L. Yang, Q. Wang, D. Liu, Z. Xu, and S. Liu, "ArtChain: Blockchain-enabled platform for art marketplace," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Jul. 2019, pp. 447–454.

[97] S. Nakamoto. (2008). *Bitcoin: A Peer-to-peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[98] M. N. Islam, V. C. Patii, and S. Kundu, "On IC traceability via blockchain," in *Proc. Int. Symp. VLSI Design, Autom. Test (VLSI-DAT)*, Apr. 2018, pp. 1–4.

[99] P. Cui, J. Dixon, U. Guin, and D. Dimase, "A blockchain-based framework for supply chain provenance," *IEEE Access*, vol. 7, pp. 157113–157125, 2019.

[100] N. Vashistha, M. M. Hossain, M. R. Shahriar, F. Farahmandi, F. Rahman, and M. M. Tehranipoor, "EChain: A blockchain-enabled ecosystem for electronic device authenticity verification," *IEEE Trans. Consum. Electron.*, vol. 68, no. 1, pp. 23–37, Feb. 2022.

[101] L. Aniello, B. Halak, P. Chai, R. Dhall, M. Mihalea, and A. Wilczynski, "Anti-BIUFf: Towards counterfeit mitigation in IC supply chains using blockchain and PUF," *Int. J. Inf. Secur.*, vol. 20, no. 3, pp. 445–460, Jun. 2021.

[102] C. K. Chaudhary, U. Chatterjee, and D. Mukhopadhayay, "Auto-PUFChain: An automated interaction tool for PUFs and blockchain in electronic supply chain," in *Proc. Asian Hardw. Oriented Secur. Trust Symp. (AsianHOST)*, Dec. 2021, pp. 1–4.

[103] J. Vosatka, "Confidence modeling and tracking of recycled integrated circuits, enabled by blockchain," in *Proc. IEEE Res. Appl. Photon. Defense Conf. (RAPID)*, 2020, pp. 1–3.

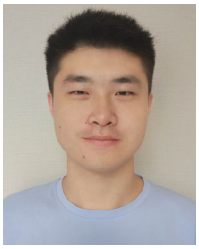
[104] T. Zhang, F. Rahman, M. Tehranipoor, and F. Farahmandi, "FPGA-chain: Enabling holistic protection of FPGA supply chain with blockchain technology," *IEEE Des. Test. IEEE Des. Test. Comput.*, vol. 40, no. 2, pp. 127–136, Apr. 2023.

[105] M. N. Islam and S. Kundu, "Enabling IC traceability via blockchain pegged to embedded PUF," *ACM Trans. Design Autom. Electron. Syst.*, vol. 24, no. 3, pp. 1–23, Apr. 2019, doi: 10.1145/3315669.

[106] U. Guin, X. Zhang, D. Forte, and M. Tehranipoor, "Low-cost on-chip structures for combating die and IC recycling," in *Proc. 51st ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2014, pp. 1–6.

[107] C.-Y. Kim, S.-J. Jun, and E.-S. Kim, "Voltage-glitch detection device and method for securing integrated circuit device from voltage glitch attack," U.S. Patent 7 085 979, Aug. 1, 2006.
[108] H. Igarashi, Y. Shi, M. Yanagisawa, and N. Togawa, "Concurrent faulty clock detection for crypto circuits against clock glitch based DFA," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2013, pp. 1432–1435.
[109] N. Homma, Y.-I. Hayashi, N. Miura, D. Fujimoto, D. Tanaka, M. Nagata, and T. Aoki, "EM attack is non-invasive?—Design methodology and validity verification of EM attack sensor," in *Proc. Cryptograph. Hardw. Embedded Syst. (CHES)*, 2014, pp. 1–16.
[110] W. He, J. Breier, and S. Bhasin, "Cheap and cheerful: A low-cost digital sensor for detecting laser fault injection attacks," in *Proc. 6th Int. Conf. Secur., Privacy, Appl. Cryptogr. Eng. (SPACE)*. Hyderabad, India: Springer, 2016, pp. 27–46.
[111] F. Schellenberg, D. R. E. Gnad, A. Moradi, and M. B. Tahoori, "An inside job: Remote power analysis attacks on FPGAs," *IEEE Des. Test. IEEE Des. Test. Comput.*, vol. 38, no. 3, pp. 58–66, Jun. 2021.
[112] A. M. Shuvo, N. Pundir, J. Park, F. Farahmandi, and M. Tehranipoor, "LDTFI: Layout-aware timing fault-injection attack assessment against differential fault analysis," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, 2022, pp. 1–12.
[113] N. Pundir, H. Li, L. Lin, N. Chang, F. Farahmandi, and M. Tehranipoor, "SPILL—Security properties and machine-learning assisted pre-silicon laser fault injection assessment," in *Proc. Int. Symp. for Test. Failure Anal.*, Oct. 2022, pp. 225–236.
[114] H. Wang, H. Li, F. Rahman, M. M. Tehranipoor, and F. Farahmandi, "SoFI: Security property-driven vulnerability assessments of ICs against fault-injection attacks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 3, pp. 452–465, Mar. 2022.
[115] *ARM MPS3 FPGA Prototyping Board*. Accessed: Sep. 10, 2023. [Online]. Available: https://www.arm.com/products/development-tools/development-boards/mps3
[116] A. Garg and T. T. Kim, "Design of SRAM PUF with improved uniformity and reliability utilizing device aging effect," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Jun. 2014, pp. 1941–1944.
[117] I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 2, pp. 203–215, Feb. 2007.
[118] X. Wu, "3D-IC technologies and 3D FPGA," in *Proc. Int. 3D Syst. Integr. Conf. (3DIC)*, Aug. 2015, pp. KN1.1–KN1.4.
[119] T. Zhang, J. Park, M. Tehranipoor, and F. Farahmandi, "PSC-TG: RTL power side-channel leakage assessment with test pattern generation," in *Proc. Design Autom. Conf. (DAC)*, 2021, pp. 709–714.
[120] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown, "MiBench: A free, commercially representative embedded benchmark suite," in *Proc. 4th Annu. IEEE Int. Workshop Workload Characterization. WWC-4*, 2001, pp. 3–14.
[121] T. Zhang, J. Wang, and Z. Chen, "A reverse engineering-based framework assisting hardware trojan detection for encrypted IPs," in *Proc. 8th Int. Conf. Instrum. Meas., Comput., Commun. Control (IMCCC)*, Jul. 2018, pp. 1649–1652.
[122] N. Wen, J. Wang, and T. Zhang, "Hardware trojan detection technique based on SOM neural network," in *Proc. 8th Int. Conf. Instrum. Meas., Comput., Commun. Control (IMCCC)*, Jul. 2018, pp. 1645–1648.
[123] F. Corno, M. S. Reorda, and G. Squillero, "RT-level ITC'99 benchmarks and first ATPG results," *IEEE Design Test Comput.*, vol. 17, no. 3, pp. 44–53, Jul. 2000.
[124] R. Brayton and A. Mishchenko, "ABC: An academic industrial-strength verification tool," in *Computer Aided Verification: 22nd International Conference, CAV 2010, Edinburgh, UK, July 15–19, 2010*. Springer, 2010, pp. 24–40.
[125] R. Cavada, A. Cimatti, M. Dorigatti, A. Griggio, A. Mariotti, A. Micheli, S. Mover, M. Roveri, and S. Tonetta, "The NUXMV symbolic model checker," in *Proc. Comput. Aided Verification, 26th Int. Conf., CAV Held Part Vienna Summer Log., VSL*, Vienna, Austria. Cham, Switzerland: Springer, Jul. 2014, pp. 334–342.
[126] S. Roshanisefat, H. Mardani Kamali, H. Homayoun, and A. Sasan, "RANE: An open-source formal de-obfuscation attack for reverse engineering of logic encrypted circuits," in *Proc. Great Lakes Symp. VLSI*, Jun. 2021, pp. 221–228.
[127] ITL Computer Security Division. (2023). *NIST Sp 800-22: Documentation and Software—Random Bit Generation: Csrc*. [Online]. Available: https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software
[128] M. T. Rahman, D. Forte, J. Fahrny, and M. Tehranipoor, "ARO-PUF: An aging-resistant ring oscillator PUF design," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2014, pp. 1–6.
[129] (2023). *Hyperledger Fabric*. [Online]. Available: https://www.hyperledger.org/use/fabric
[130] P. E. Calzada, M. S. U. I. Sami, K. Z. Azar, F. Rahman, F. Farahmandi, and M. Tehranipoor, "Heterogeneous integration supply chain integrity through blockchain and CHSM," *ACM Trans. Design Autom. Electron. Syst.*, vol. 29, no. 1, pp. 1–25, Nov. 2023, doi: 10.1145/3625823.
[131] (2023). *Go Documentation*. [Online]. Available: https://go.dev/doc/
[132] S. Ullah, V. Oleshchuk, and H. Pussewalage, "A lightweight access control scheme with attribute policy for blockchain-enabled Internet of Things," in *Proc. 20th Int. Conf. Secur. Cryptography*. SciTePress, 2023, pp. 528–539.
[133] S. S. Ullah, V. Oleshchuk, and H. S. G. Pussewalage, "A survey on blockchain envisioned attribute based access control for Internet of Things: Overview, comparative analysis, and open research challenges," *Comput. Netw.*, vol. 235, Nov. 2023, Art. no. 109994. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128623004395
[134] (2023). *Access Control Lists*. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/access_control.html
[135] *Riscv/Core/Riscv At Master Ultraembedded/Riscv*. Accessed: Sep. 10, 2023. [Online]. Available: https://github.com/ultraembedded/riscv/tree/master/core/riscv
[136] *Agalimberti/Nocrouter: Rtl Network-on-chip Router Design in Systemverilog By Andrea Galimberti, Filippo Testa and Alberto Zeni*. Accessed: Sep. 10, 2023. [Online]. Available: https://github.com/agalimberti/NoCRouter/tree/master
[137] *2.5D Rdl Interposer System Codesign Using Integrity 3D-IC Platform and Allegro Pd Plus: Rak*. Accessed: Sep. 10, 2023. [Online]. Available: https://support.cadence.com/

MD SAMI UL ISLAM SAMI received the B.S. degree in electrical and electronic engineering from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh, and the M.S. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, where he is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department. His research is focused on hardware security and trust, security-aware SoC design and test, security-aware system-in-package, and VLSI.



TAO ZHANG received the B.S. degree from Northwest University, in 2016, and the M.S. degree from the University of Electronic Science and Technology of China, in 2019. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Florida. He has published more than ten peer-reviewed publications in premier venues, including Design Automation Conference (DAC), IEEE Electronic Components and Technology Conference (ECTC), and European Test Symposium (ETS). His research focuses on side-channel security, FPGA security, and heterogeneous integration security. He was a recipient of the DAC Young Fellowship, in 2020 and 2021. He serves as a reviewer for multiple renowned IEEE/ACM journals and conferences.

AMIT MAZUMDER SHUVO received the B.Sc. degree in electrical and electronics engineering from Bangladesh University of Engineering and Technology, in 2017. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering (ECE) Department, Florida Institute for Cybersecurity Research (FICS), University of Florida. He is also a Graduate Research Assistant with the University of Florida. His research focuses on fault injection attack assessment, property-driven security assurance, tamper detection, and secure heterogeneous integration.

MD SAAD UL HAQUE received the B.S. degree in electrical and electronic engineering from Bangladesh University of Engineering and Technology. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering (ECE) Department, Florida Institute for Cybersecurity Research (FICS), University of Florida, USA. His research focuses on secure electronic system design and trust validation.

PAUL E. CALZADA received the B.S. degree in computer engineering from the University of Florida (UF), where he is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering (ECE) Department, Florida Institute for Cybersecurity Research (FICS). His research focuses on hardware security and trust, secure heterogeneous integration, and PCB-level Trojans.

KIMIA ZAMIRI AZAR received the B.S. degree from the Department of Electrical and Computer Engineering (ECE), K. N. T. University, in 2013, the M.S. degree from the Department of ECE, Shahid Beheshti University, in 2015, and the Ph.D. degree from the Department of ECE, George Mason University, in 2021. She is a Research Assistant Professor with the Department of ECE, University of Florida. She has numerous publications in high-prestigious journals and conferences, including IEEE TRANSACTIONS journals, IACR Transactions on CHES, RAID, DAC, DATE, ICCAD, and VTS, with awards including nominations/recipients for the Best Paper Award in ISVLSI'20, ICCAD'20, HOST'22, and DATE'23. Her research interests span hardware security and trust, supply chain security, system-on-chips security validation and verification, and IoT security.

HADI MARDANI KAMALI received the B.S. degree from the Department of Electrical and Computer Engineering (ECE), K. N. T. University, in 2011, the M.S. degree from the Department of ECE, Sharif University of Technology, in 2013, and the Ph.D. degree from the Department of ECE, George Mason University, in 2021. He is a Research Assistant Professor with the Department of ECE, University of Florida. His research delves into hardware security, with a particular focus on exploiting IP protection techniques, design-for-trust for VLSI circuits, and CAD frameworks for security (design-for-security), in which he has numerous publications in top journals and conferences. He is the coathour of two books and multiple patents. His reserach received awards including nominations for Best Paper Award in ISVLSI'20, ICCAD'19, ICCAD'20, DCAS 2020, HOST'22, and DATE'23.

FAHIM RAHMAN received the B.S. degree in electrical and electronic engineering from Bangladesh University of Engineering and Technology, Bangladesh, the M.S. degree in electrical and computer engineering from the University of Connecticut, USA, in 2015, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2018. He is currently a Research Assistant Professor with the Electrical and Computer Engineering Department, University of Florida. His research has been sponsored by SRC, AFOSR, AFRL, DARPA, Cisco, TI, and NIST. His current research interests are in the domain of hardware and cybersecurity and trust, including electronic supply-chain security, CAD for security and automatic assessment, and hardware-assisted cybersecurity. He is a member of ACM.

FARIMAH FARAHMANDI (Member, IEEE) received the B.S. and M.S. degrees from the Department of ECE, University of Tehran, Iran, in 2010 and 2013, respectively, and the Ph.D. degree from the Department of CISE, University of Florida, in 2018. She is an Assistant Professor with the Department of ECE, University of Florida. Her research interests include design automation of system-on-chips and energy-efficient systems, formal verification, hardware security validation, and post-silicon validation and debug. Her research has resulted in five books, nine book chapters, and several publications in premier ACM/IEEE journals and conferences including DAC and DATE, with awards including the IEEE System Validation and Debug Technology Committee Student Research Award, the Gartner Group Info-Tech Scholarship, a nomination for the Best Paper Award in ASPDAC 2017, and the DAC Richard Newton Young Student Fellowship. She is a member of ACM.

MARK TEHRANIPOOR (Fellow, IEEE) is currently the Intel Charles E. Young Preeminence Endowed Chair Professor of cybersecurity and the Chair of the Department of Electrical and Computer Engineering (ECE), University of Florida. He served as the Founding Director for Florida Institute for Cybersecurity (FICS) Research, from 2015 to 2022. His current research projects include hardware security and trust, supply chain security, IoT security, VLSI design, and test and reliability. He has published over 400 journal articles and refereed conference papers and has delivered more than 220 invited talks and keynote addresses. He has 21 patents issued, 25 pending invention disclosures, and has published 16 books. He is a Fellow of ACM and the National Academy of Inventors (NAI), a Golden Core Member of IEEE Computer Society, and a member of ACM SIGDA. He was a recipient of a dozen best paper awards and nominations. He has co-founded the IEEE International Symposium on Hardware-Oriented Security and Trust (HOST) and the IEEE International Conference on Physical Assurance and Inspection of Electronics (PAINE).