TREEHOUSE: A Secure Asset Management Infrastructure For Protecting 3DIC Designs

Patanjali SLPSK, Sandip Ray *Senior Member, IEEE*, and Swarup Bhunia *Senior Member, IEEE*

Abstract—The push to meet growing user requirements and manufacturing challenges at lower technology nodes have motivated chip designers to adopt non-traditional design techniques. 2.5D/3DIC stacking has gained popularity in recent years since it enables chip manufacturers to integrate complex IPs to meet user demands without incurring design penalties. However, the non-traditional nature of the supply chain also means that additional challenges exist for verification and testing of the manufactured design, making the trust assurance of these designs an extremely challenging proposition. While there have been works focussing on securing 3DIC designs, very few address a completely untrusted supply chain. A robust security countermeasure must address the diverse trust requirements of the IPs in the design and the distributed supply chain requirements while ensuring that the functionality and performance overheads of the IC are not violated. We present TREEHOUSE , a trust assurance solution to counter piracy, reverse-engineering, and counterfeiting attacks. TREEHOUSE uses scan authentication to detect piracy and counterfeiting, scan-and functional-locking to prevent reverse-engineering. We evaluate the efficiency of our proposed scheme on an example 3DIC design. We show that TREEHOUSE incurs less than 1% area and power overheads while incurring less than 1% increase in overall gate count for each layer.

Index Terms—3DIC, IP Authentication, IP Piracy, Reverse-engineering, Logic Locking.

1 INTRODUCTION

THE need to integrate complex functionality while satisfying the area, power, and delay constraints at lower technology nodes have motivated chip designers to look beyond conventional system design and integration techniques. One promising strategy is the 3DIC architecture, which involves stacking different functional components vertically. Unlike traditional SoC architectures, 3DIC designs consist of several IPs spread across different layers and communicate via specially designed interconnects. Figure 1(a) shows a typical 3DIC design implementation. 3DICs offer many benefits to chip manufacturers, including reduced area consumption and the ability to design different functional components with different process technologies.

3DICs are similar to traditional System-on-Chip (SoC) architectures from a functional standpoint. Both SoCs and 3DICs comprise multiple IPs integrated to achieve a common objective. Current 3DIC designs contain several dies fabricated individually and then combined at the assembly stage, as shown in Figure 2. The unconventional approach adopted for 3DICs poses an interesting challenge from a trust assurance perspective. The globally distributed IC supply chain introduces various trust issues such as piracy, counterfeiting, reverse engineering (RE), and tampering. While approaches like watermarking, Physically Unclonable Functions (PUFs), and Logic Locking mitigate these issues at an IP level, the problem of trust assurance in a 3DIC is rather complex.

Patanjali SLPSK, Sandip Ray, and Swarup Bhunia are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, 32603 USA. E-mail:patanjal.sristil@ufl.edu, sandip@ece.ufl.edu, swarup@ece.ufl.edu.

Over the years, various researchers have proposed attacks and countermeasures for securing a 3DIC design. The attacks on 3DICs have focused predominantly on side-channels [1], [2] and Trojans [3], [4]. The countermeasures focus on split manufacturing [5], Root-of-trust modules [6], [7], logic-locking [8], PUFs [9]and parameter analysis techniques such as delay analysis [10] and power analysis [11] for improving the Trojan and side-channel resistance of the design respectively. The problem of trust assurance in 3DICs thus remains an active, and interesting area of research.

The complexity in 3DIC trust assurance arises due to the following factors, the diverse trust requirements imply that each IP in the design could have a different security countermeasure integrated by the design house or the IP vendor. For example, the architecture shown in Figure 1(a) consists of a crypto core that is Logic Locked using sequential locking to prevent RE attacks, an FIR IP with a PUF module for authentication, and a GPS accelerator that is Logic Locked to prevent RE attacks and also contains a watermark for authentication. Thus, from the 3DIC designer's perspective, the challenge of securing the 3DIC design is to ensure the various security operations on these IPs can be performed seamlessly, irrespective of the underlying trust assurance protocol. We address this pertinent and crucial issue in this work.

Another factor that needs to be considered for trust assurance in 3DICs also arises in part due to the difference in the supply chain. Figure 2 shows the various steps involved in the manufacture of a 3DIC design. Unlike a traditional SoC flow which considers a single foundry and a testing facility for the entire SoC, a 3DIC design requires multiple foundries for manufacturing each layer, and multiple testing facilities involved at layer-level (pre-bond), die-level (post-bond), and post-packaging (final IC) [12]. Thus, any trust assurance framework for securing 3DIC designs against supply chain attacks must consider these differences in the supply chain. The inclusion of multiple foundries, and testing facilities makes it harder to directly adapt traditional SoC solutions.

In this work, we refer to the security countermeasures such as PUFs, watermarks, etc., as Hardware Security countermeasures (HSCs). These HSCs typically have associated metadata such as Challenge-Response vectors (PUFs) and Unlocking Keys (Logic Locking). We refer to this metadata as Hardware Security Metadata (HSMs). Any protocol for securing 3DICs should achieve the following objectives: 1) that the overall functionality and testability of the IC are not affected; for example, the test facility should be able to test the 3DIC design seamlessly, 2) ensure that the design house can safely transfer the HSM data to the design, and 3) provide a safe and secure interface for transferring HSM data to the HSC. We refer to steps (2) and (3) as provisioning.

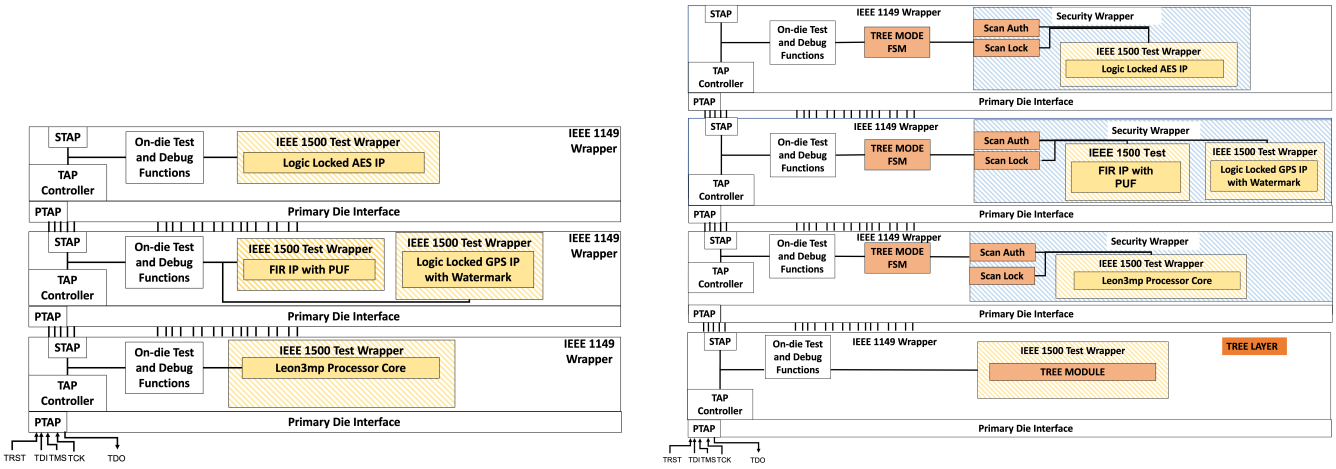

Fig. 1: (a) The architecture of a typical 3DIC design containing a locked AES IP, an FIR IP with a PUF, a Logic Locked GPS IP with a watermark, and processor core. The various layers contain a IEEE 1838 Test Wrapper which contains a Primary Test Access Port and a IEEE1149 wrapper for layer-level testing and a secondary test access port for ensuring layer-to-layer communication. (b) shows the TREEHOUSE protocol with the TREE Module, TREE MODE FSM, Scan protect, and security wrapper modules.

During the course of the testing phases, provisioning can occur several times and is thus vital to ensure the overall safety of the 3DIC design.

To address the above issue, we propose TREEHOUSE, a flexible trust assurance solution for 3DIC designs. TREEHOUSE consists of three components: a *TRust Enforcing Entity* (TREE) module, an augmented bus wrapper to support the security operations, and a Scan protection module. The TREE module enables the designer to implement flexible security policies based on the trust requirements of the individual IPs and the entire 3DIC. To ensure that the TREE Module is not tampered with during the integration process, the TREE Module contains a in-built fingerprint such as PUF that can be verified by the design house during the provisioning process. The security wrapper ensures that the TREE module can communicate with the various IPs to efficiently implement the security protocols, test, and functional mode operations. The purpose of the Scan protection module is to ensure the design's security during layer-level and IC-level test phases. Figure 1(b) shows the TREEHOUSE framework integrated into the baseline design shown in Figure 1c. In this work, we illustrate that these three modules help the 3DIC designer implement a trust assurance mechanism to prevent various supply chain attacks without incurring significant overheads. We sum up the contributions of our work below:

1) We propose TREEHOUSE, a flexible trust assurance architecture for 3DICs that requires minimal trust assumptions on the supply chain.
2) We illustrate the role of TREEHOUSE in securely provisioning and testing the various IPs in the design. We show that TREEHOUSE is agnostic to the underlying HSC by showing the interaction with various HSC architectures.
3) We show that the proposed secure provisioning architecture and protocol can mitigate reverse-engineering, counterfeiting, and piracy threats due to an untrusted foundry, untrusted testing facility, or both.
4) We demonstrate an example implementation of TREEHOUSE using a RISCV-based TREE controller integrated onto a 3DIC design and show that TREEHOUSE incurs less than 1% area and power overheads while incurring less than 1% increase in overall gate count for each layer.
5) We evaluate the security of the TREEHOUSE protocol both empirically and mathematically to demonstrate the complexity of breaking the security protocols employed by TREEHOUSE.

We organize the rest of the paper as follows: We describe the 3DIC manufacturing process, the security threats, and the existing countermeasures in Section 2 along with a short discussion on the IEEE1838 architecture, a critical component of the TREEHOUSE framework. We discuss our proposed threat model and the resulting potential supply-chain attacks in Section 3. We describe the TREEHOUSE architecture in Section 4 and the TREEHOUSE protocol in Section 5. We detail the experimental setup used for evaluating the robustness of TREEHOUSE in Section 6 and present the results regarding overheads in Section 7 respectively and the security guarantees in Section 8. We discuss the interoperability and scalability of TREEHOUSE in Section 9. We offer concluding remarks and identify potential future directions in Section 10.

2 Background and Related Work

In this Section, we discuss the current state-of-the-art in 3DIC security and provide a brief background on the IEEE1838 and IEEE1500 Test Wrapper standards used in our design. The diminishing area and power benefits of the traditional 2D stacking has led researchers to explore different stacking technologies. 2.5D/3D architectures offer better area and performance benefits compared to naive 2D stacking. 2.5DIC architectures use a common integration platform (intersposer) that allows multiple 2D architectures to be integrated next to each other. 3DIC architectures either consist of stacked architectures that use layers or planes fabricated in different foundries integrated together and communicating via specially designed interconnects known as Through-Silicon-Vias (TSVs) or monolithic 3DICs integrated using inter-layer vias. In this work, we focus on the 3DIC architectures using TSVs and the corresponding supply-chain. Figure 2 highlights the various entities involved in the 3DIC manufacturing process, the attacker

TABLE 1: Related Works on Security Countermeasures in 3DIC research.

Work	Supply Chain Threat	Nature of Countermeasure	Threat Model	3DIC Architecture	Attacker Access
Imeson et al. [5]	Trojans	Isolation between Trusted and Untrusted Layers	Untrusted Foundry	3D	GDSII
Valamehr et al. [6]	Trojans, Counterfeiting	Split Manufacturing; Runtime Monitoring	Untrusted Foundry	3D	GDSII
Nabeel et al. [7]	IP piracy, Trojans, Reverse-Engineering	Secure Intersposer Root-of-Trust Module	Untrusted Testing Facility & Untrusted Foundry	2.5D	GDSII and manufactured IC
Dofe et al. [8]	Reverse-Engineering	Split Manufacturing; Runtime Monitoring	Untrusted Foundry	3D	GDSII
Wang et al. [9]	IP Piracy	Physically Unclonable Functions	Untrusted Foundry	3D	GDSII
Alhelaly et al. [10]	Trojans	Delay Analysis	Untrusted Foundry	3D	GDSII
Dofe et al. [11]	Side-channel Attacks	Secure Power Delivery Network	Field	3D	Power Delivery Network
Bilzor et al. [13]	Trojans	Using EM-signature for detecting malicious Logic	Untrusted Foundry	3D	GDSII
Xie et al. [14]	IP Piracy	Secure Intersposer	Untrusted Foundry	2.5D	GDSII
Knetchel et al. [15]	Side-channel attacks	Floorplanning for reducing information leakage	Field	3D	Manufactured IC
Huffmire et al. [16]	Trojans, Side-channel attacks	Control Layer for reducing information leakage	Untrusted Foundry	3D	GDSII
Salman et al. [17]	Trojans	Shielding plane between layers	Untrusted Foundry	3D	GDSII
Yan et al. [18]	Trojans	Fine-grained locking for reducing information leakage	Untrusted Foundry	3D	GDSII
Ours	Reverse-Engineering, IP Piracy	Plug-and-play Trust Enforcing Layer, Scan protection Module	Untrusted Testing Facility & Untrusted Foundry	3D	GDSII and manufactured IC

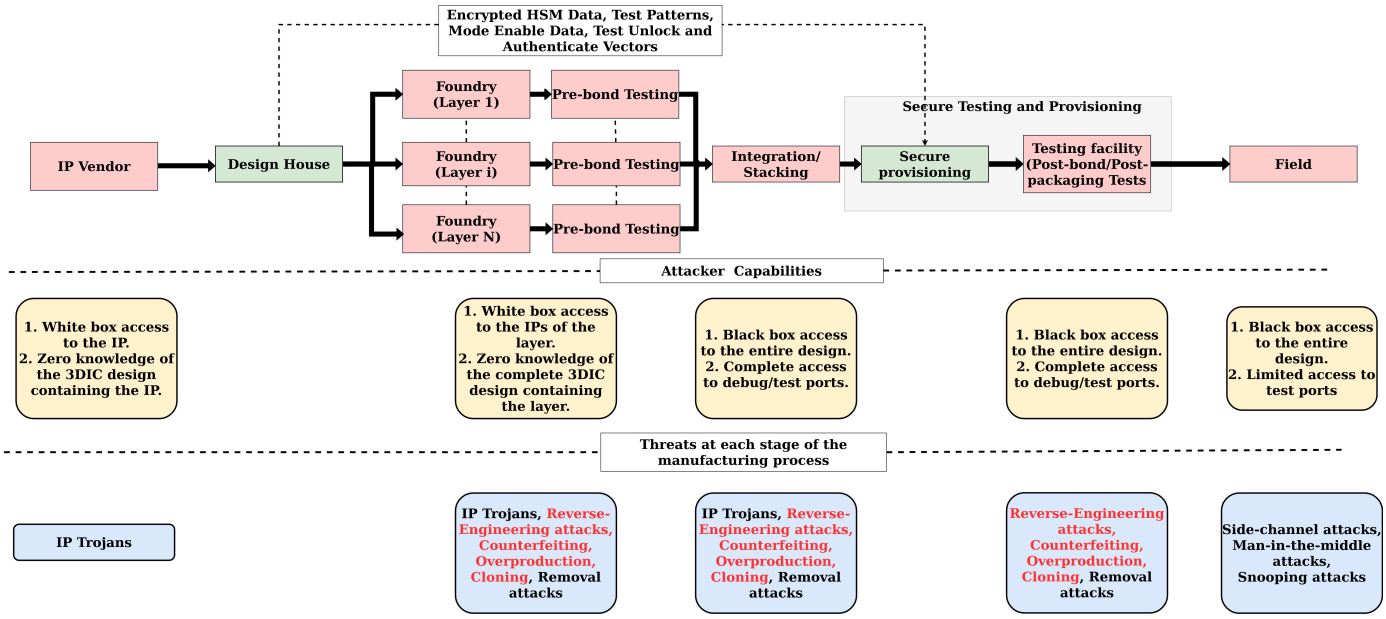

Fig. 2: The IPs in the design are sourced from multiple IP vendors and are fabricated in different layers using separate foundries. The layers are then undergo several testing phases before and after integration before being sent to the field. The attacker's capabilities and the corresponding threats to the IPs at each stage are also highlighted.

capabilities at each stage, and the possible attacks. Over the years, various research works have focused on attacks and countermeasures. A detailed survey of the various works can be found in [19], [20]. We detail some of the prominent attacks and countermeasures in this Section.

2.1 Attacks on 3DIC Designs

The distributed supply chain poses a myriad of security challenges to system designers. The attacks have primarily focussed on inserting Trojans and exploiting the side-channel leakages. We detail the recent results focussing on attacks below:

2.1.1 *Trojan Attacks on 3DICs*

The authors in [21] provide a comprehesive survey of different hardware Trojan structures for 3DICs such as thermal-triggered [4], cross-layer [22] along with various threat models. [4] studies the impact of Trojan circuits triggered due

to high-temperature effects. The authors use the state transitions triggered under high-temperature effects to activate the Trojans. Since the trigger conditions are not met during normal temperature operations, the Trojan is undetected. [3] studies the effects of Trojans due to process variations. [22] propose a cross-layer Trojan architecture where the Trojan in one layer can be activated by the Triggers present in other layers.

2.1.2 Side-channel Attacks on 3DICs

The authors in [1] study the effect of thermal leakages on the side-channel activity of the design. The authors profile the temperature of the design to leak information. The work in [2] exploits power distribution network for power side-channel attacks. The authors utilize the 3D-power distribution network to infer the switching activity of the logic in the neighboring layers.

2.2 3DIC Security Countermeasures

We now discuss the various countermeasures aimed at securing the 3DIC supply chain.

2.2.1 Side-channel Countermeasures

The authors in [11] propose a randomization-based side-channel countermeasure for thwarting power side-channel leakages. The proposed technique leverages the Power Distribution Network in the neighboring layers to introduce additional noise, making it harder for the attacker to carry out power side-channel attacks. On the other hand, the authors in [1], [15] propose an EDA-based technique that attempts to partition the design to reduce the side-channel leakages.

2.2.2 Reverse-Engineering Countermeasures

These techniques prevent the attacker from gaining complete knowledge of the entire design. The most popular technique is split manufacturing [5], [6], [13]. Split manufacturing distributes the various design components across multiple foundries. The advantage is that the attacker does not gain complete knowledge of the design and cannot compromise the design. Split manufacturing is employed at the coarse-grained level wherein different layers are manufactured at separate foundries [6], or a fine-grained level wherein individual IPs are partitioned and manufactured separately. Split manufacturing techniques prevent reverse-engineering attacks since the attacker does not know the design. However, they are susceptible to counterfeiting attacks. Other Reverse-Engineering Countermeasures have been explored at different abstractions. The authors in [8] propose a transistor-level Logic Locking technique. The proposed approach locks the design by inserting locking transistors and camouflaged contacts to prevent the attacker from understanding the design. The authors in [23] present a combined split-manufacturing and camouflaging technique to counter reverse engineering attacks. The authors in [7], [14] propose a countermeasure using the intersposer layer for securing the design architecture.

2.2.3 Trojan Countermeasures

Trojan Countermeasure techniques rely on monitoring the inter-and intra-layer communication to detect and prevent malicious activity. Techniques such as [16] rely on additional control layers to avoid side-channel and Trojan attacks. The authors in [6] propose adding a plane for securing inter-layer communication. Compartmentalization techniques have also been explored at the design level. The designer could integrate a shielding plane that prevents information leakage across layers [17] or by inserting fine-grained locking techniques that prevent the attacker from gaining unauthorized access to the various layers of the design [18].

Table 1 summarizes the various 3DIC security countermeasures and their capabilities.

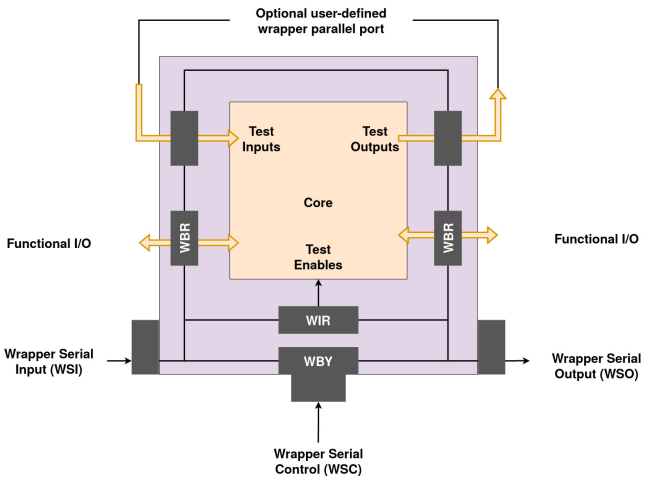

Fig. 3: An example implementation of the IEEE 1500 Core Test Access protocol.

2.3 3DIC Test Access Standard

Our proposed TREEHOUSE solution assumes that the 3DIC design contains the IEEE 1838 Core Test Access protocol for facilitating the testing process. Figure 1(a) shows the various components of the IEEE 1838 Test Access protocol. Due to the distributed nature of the supply chain, the 3DICs undergo multiple testing processes. The first phase, also known as Known-Good-Die Testing [24], occurs at the foundries where individual layers are manufactured, and the second phase occurs after the layers are integrated. The final test phase is done after packaging and assembly. The IEEE 1838 test wrapper is designed to facilitate multiple test operations.

The IEEE 1838 wrapper [25], [26] consists of a baseline IEEE1149 test wrapper coupled with an IEEE 1500 wrapper for testing the IPs. To facilitate intra-layer and inter-layer communication, the IEEE1838 wrapper consists of two test access ports: A Primary Test Access Port (PTAP) and a Secondary Test Access Port (STAP). The PTAP contains the signals used to test the IPs on the corresponding layer. The PTAP logic consists of five signals TCK, TMS, TDI, TDO, and TSRSTN, that are analogous to the IEEE1149 counterparts. The STAP logic is used to communicate the test signals to the PTAP modules of the neighboring layers. The signals in the STAP module correspond to the signals of the PTAP logic. Additionally, there could be a Flexible Parallel Port (FPP) logic to facilitate user data transfer between the layers.

Our proposed TREEHOUSE implementation assumes the presence of the IEEE1838 test wrapper for layer-level testing and the IEEE1500 test wrapper for IP-level testing. Our TREEHOUSE implementation modifies the IEEE1500 wrapper. We give a short background on the IEEE1500 to clarify our system design. The IEEE 1500 [27] test architecture was proposed to standardize test and debug interfaces of hardware IP cores with various functionality, interfaces, and control mechanisms. It contains two components a

Core Test Language (CTL) [28] and a test wrapper architecture [29]. Figure 3 shows the example of a core containing the IEEE1500 logic. The wrapper structure contains a Wrapper Serial Port (WSP) and an optional Wrapper Parallel Port. Control registers such as Wrapper Instruction Register (WIR), Wrapper Bypass Register (WBY), and Wrapper Boundary Register (WBR) are also included. The WBY register provides a bypass path for Wrapper Serial In (WSI)- Wrapper Serial Out (WSO) terminals. The WIR register enables various wrapper operations, and the WBR register is used to transfer the test patterns and collect multiple responses. Additional control registers such as WRCK, captured, ShiftDR, UpdateDR, and WRSTN trigger various register-level events.

3 Threat Model

We now detail our proposed threat model, our assumptions concerning the 3DIC design, the adversary capabilities, and discuss the possible attacks in this Section.

3.1 System Configuration

We assume that the individual layers in the 3DIC design are obtained from various IP vendors and organized in different layers at the design house. The separate layers of the design are manufactured at other foundries and then integrated. The individual layers are tested before integration. The integrated 3DIC chip is then tested being sent to the field. We assume that only the provisioning process is secure. We assume that the testing facility is untrusted; thus, the ATE could be compromised. We also assume that the interaction between the TREE Module and the IC vendor happens via a secure cloud interface using the Ethernet controller. We assume that the TREE Layer is manufactured and tested under trusted settings. Thus, the Ethernet controller is considered fully functional during the integration stage. This assumption is consistent with other Zero Trust solutions in literature [30], [31] that assume the presence of a secure cloud interface. To counter the existing threats in the supply chain, we assume that the individual IPs contain some security countermeasures. We assume that the IPs are locked and that the 3DIC design uses IEEE1838 test wrapper and the individual IPs contain the IEEE1500 test architecture for testing and debugging purposes in our proposed threat model.

Some complexity in the TREEHOUSE architecture stems from our goal to make the integration compatible with IEEE1838 standard, which precludes certain optimizations and enforces constraints on wrapper designs. Nevertheless, we feel that adherence with the standard is critical since it provides flexibility in integrating TREEHOUSE on a variety of 3DIC systems with minimal custom engineering. Furthermore, our results show that this does not result in significant cost of area of power.

3.2 Foundry-based Threat Model

The adversary at the foundry can access the layout-level representation of the design. In the case of a 3DIC system, the adversary could either have access to a single-layer, single IP on a layer, or the complete design. We assume that the attacker can perform structural analysis, apply input patterns of their choice and observe the outputs for analyzing design intent extracting design secrets, thus compromising the integrity of the original design. The attacker can also replace individual IPs or layers with counterfeit or pirated components. In the foundry-based threat model, we assume that the attacker has access to the layout-level representation of one or more design layers. The attacker can then reverse-engineer the layout-level representation to infer the design intent, extract design secrets, or perform malicious modifications to compromise the original design's integrity. The foundry-level attacker has unrestricted access to the entire design, including the internal state elements that are not observable otherwise. However, the layout-level abstraction limits the attackers' ability to successfully reverse complex circuits or circuits with inbuilt countermeasures. Additionally, the attacker is also restricted by the design layers fabricated at the foundry. For example, a foundry-level attacker with access to only one layer of a multi-layer 3DIC is limited to performing RE attacks only on that layer.

3.3 Testing Facility-based Threat Model

An adversary at the testing facility has access to the complete design but in a black-box fashion. The adversary's access is limited by the ability to control/observe only the design's input/output (functional and scan) ports. Despite the limited access to the internal structure of the design, the attacker in an untrusted testing facility can leak design secrets such as unlocking keys challenge-response pairs used for authentication. The attacker in the testing facility has complete access to the scan chain of the manufactured design. The attacker can apply specially crafted input patterns to infer design secrets. Additionally, the attacker can also leak the unlocking keys to unlock the design during testing.

3.4 Collusion-based Threat Model

In the collusion attack, we assume that the adversary in the testing facility and the foundry work in tandem by sharing information obtained at each stage to compromise the design. For example, the attacker in the untrusted foundry could leak the information regarding the location of the critical gates, which the attacker in the untrusted testing facility could then use to craft targeted structural attacks or oracle-guided attacks to recover the unlocking key.

The attacker in the untrusted foundry could utilize the layout-level access to the design and leak the structural information of the design to the attacker in the testing facility. This would enable the attacker in the testing facility to carry out attacks such as key sensitization attacks [32] or structural attacks [33] to recover the unlocking for the corresponding design.

The orthogonal attacker capabilities and the collusion-based threat model provide a unique challenge for securing the 3DIC design. Traditional solutions that prevent counterfeiting, such as PUFs, work for the foundry-based attacks but do not scale for the testing facility and the collusion attacks. However, techniques like scan-locking [34], [35] prevent unauthorized scan access but do not prevent counterfeiting attacks. Thus, there is a need for a comprehensive trust assurance solution that simultaneously averts the three attacks described above.

4 TREEHOUSE Architecture

In this Section, we outline the various components of the TREEHOUSE framework. TREEHOUSE consists of a TREE module that helps the designer securely provision the HSM across the HSCs in various IPs. The security wrapper module allows the TREE module to interact with the HSCs for performing security operations. Finally, the scan protection module helps protect the design during the various testing phases (pre-bond, post-bond, and post-packaging). We now describe the architecture of these modules in detail.

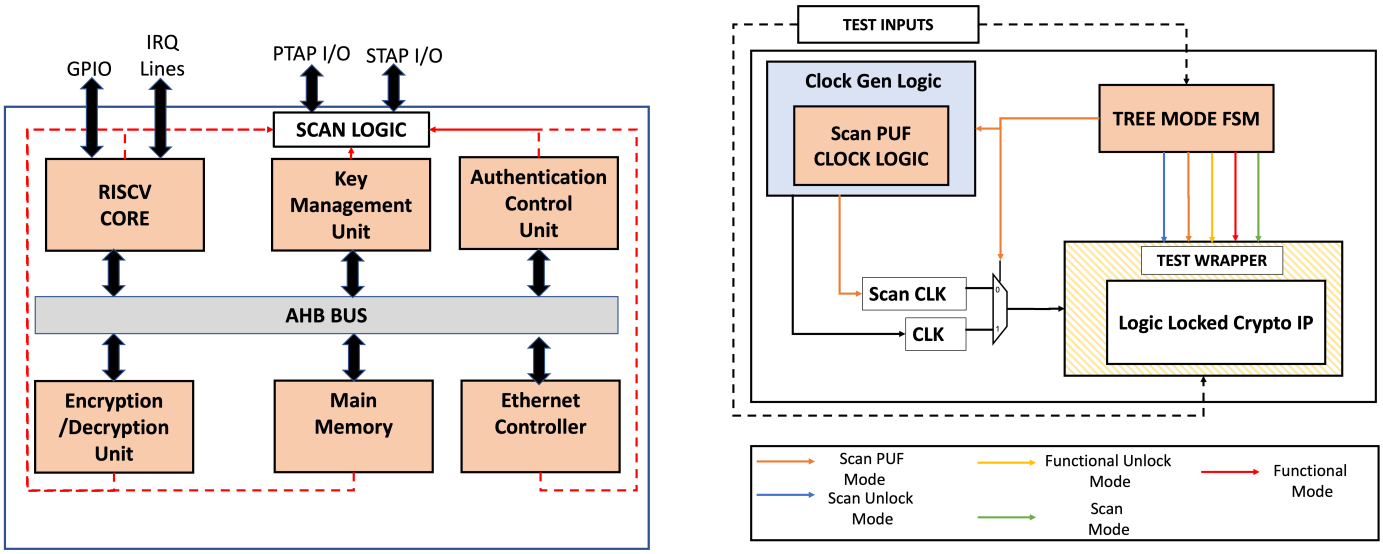

Fig. 4: (a) The architecture of a the TREE Module with different functional units for performing Scan protection, and HSM data management. (b) The architecture of Scan Protection Logic within Layer. The Test Inputs pins are used to feed the Mode Enable Vectors in the TREE MODE FSM. The TREE MODE FSM enables the appropriate modes in the IP based on these vectors.

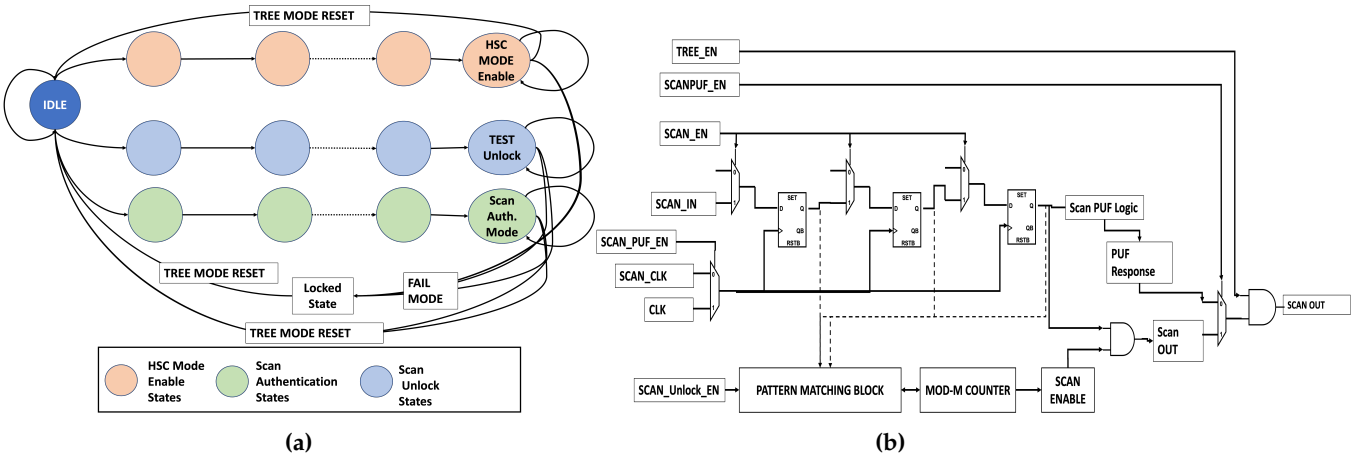

Fig. 5: (a) The various state elements of the TREE MODE FSM interacting with an IP containing two HSCs. The application of the appropriate Mode Select Vectors causes the TREE MODE FSM to transition to the appropriate stage and enable the corresponding operation. (b) The architecture of Scan Protect Logic within an IP. The Scan Authentication and Scan Unlocking mechanisms are highlighted.

4.1 Trust Enforcing Entity

The Trust Enforcing Entity (TREE) module is a lightweight microcontroller in the trusted layer. The TREE module consists of a RISCV core, a memory module for storing the HSM data, and dedicated controllers for enforcing the security policies concerning various HSCs. In this implementation, the 3DIC shown in Figure 1(a) contains a locked IP, an IP with a PUF, and an IP that is both Logic locked and has a watermark. The TREE module thus contains a Policy controller module to help unlock and authenticate the various IPs. The TREE module also includes a Scan Protection module to help unlock and authenticate the test modules. Figure 4(a) shows the TREE module's architecture. The HSM data is stored in an encrypted format to prevent unwarranted data leakage. The HSM data is decrypted before being transmitted to the corresponding layer. The TREE Module contains dedicated IPs for managing the various HSC protocols. The TREE Module contains a Key Management Units for Test and Functional Unlocking. These units store the Unlocking Keys, and other associated metadata such as nature of the unlocking protocol (Sequential or Combinational), nature of data transfer (burst or word) etc. The Authentication Control Unit comprises of a control element and Content Accessible Memory that stores the challenge vectors corresponding to each IP and the Scan Logic, and their golden responses in an encrypted format. The PCM module also contains a comparison logic for verifying the validity of the generated response. The Key Management Unit stores the encrypted keys for each IP. We implement a content accessible memory (CAM) structure within the unlocking module. The ID of each IP/Layer is used as the index and the corresponding unlocking key vectors are provided as the response. The Ethernet Controller is used to establish a secure communication channel between the design house

and the TREE Module. To ensure that the TREE Module is not tampered with during the integration process, the TREE Module contains a in-built fingerprint such as PUF that can be verified by the design house during the provisioning process.

4.2 Security Wrapper

We augment each IP in the untrusted layer with a modified test wrapper. The security wrapper aims to facilitate a) a secure testing process and b) help the TREE module communicate with the various HSCs present in the 3DIC. The security wrapper helps the TREE module communicate with the various IPs agnostic to the underlying security protocol. The security wrapper consists of a Finite State Machine called the TREE MODE FSM. The Security wrapper also contains additional registers for enabling authentication and other security operations. Figure 5(a) shows the various components of the Security Wrapper. The Security Wrapper communicates with the TREE MODE FSM to enable various security operations. The Security Wrapper also contains dedicated storage buffer for storing the HSM data such as unlocking keys, challenge-vectors and the responses. We modify The IEEE 1500 wrapper by adding an extra input port called the `TREE_MODE_RESET`. The `TREE_MODE_RESET` along with the WRSTN signal is used to enable different modes as follows:

`TREE_MODE_RESET`=0,WRSTN=0: Functional Mode.
`TREE_MODE_RESET`=0,WRSTN=1: Test Mode.
`TREE_MODE_RESET`=1,WRSTN=1: At Speed Test Mode.
`TREE_MODE_RESET`=1,WRSTN=0: TREE Mode.

The various security operations, such as Scan Authenticate, Scan Unlock, and other IP-level security protocols, are performed in TREE Mode. Figure 5(a) shows the various stages in the TREE MODE FSM. Once in TREE mode, the different security protocols can be enabled by applying specialized input vectors known as the Mode Enable vectors that cause state transitions in the TREE MODE FSM. We ensure no overlap between the input vectors for each security operation to ensure seamless state transition.

4.3 Scan Protection Module

As mentioned in Section 1, the distributed nature of the 3DIC supply chain involves multiple testing phases such as pre-bond (layer-level), post-bond (pre-packaging), and post-packaging. The Scan Protection Module helps ensure the overall design's security across the testing phases. The Scan Protection Module consists of the following components, a Scan Authentication Module, and a Scan Unlock Module.

The role of the authentication module is to perform device registration and authentication for detecting counterfeiting attacks. The signature produced by the authentication module has to achieve two objectives i) a) produce a unique response that can be used for device authentication, b) provide a cryptographic collision-resistant signature that can be used as the device encryption key. We specify the latter condition to prevent tampering attacks during the pre-bond testing. If an attacker tampers with the design during the pre-bond testing process, it would compromise the ScanPUF signature and thus cause the generated key to be incorrect. Thus the metadata transferred from the design house such as test patterns and the HSM data cannot be decrypted. This renders the device untestable and unusable. Since, existing watermark techniques only satisfy the first objective but not the second. Thus a watermark cannot be used to authenticate the device. Our implementation of `TREEHOUSE` uses the boundary-scan PUF implementations from [36] to implement a low-cost, low-overhead authentication mechanism. The PUF uses the path delays of the boundary scan elements as the entropy source. During the enrollment process, the path delays of the ICs are measured and recorded. The location of the scan flip flops serves as the challenge, while the delay between two transitions during successive scan operations serves as the response. `TREEHOUSE` prevents unauthorized access in test/debug mode by obfuscating the scan chain.

The Locking module restricts unauthorized access to the scan chain using a scan-lock mechanism. Figure 5(b) shows an example implementation of `TREEHOUSE` on a single layer of a 3DIC design. The authentication and locking elements are integrated into the individual IP blocks and communicate with the control logic. The control logic communicates with the Clock generation logic to produce the control signals necessary for authentication. We use the implementation from [34] to restrict access to the scan chain. Scan lock contains a key expansion module that takes in the debug key provided by the TREE module and unlocks/locks the various debug registers during the testing and debug phases. The test ports are not enabled by default. The unlocking protocol requires the application of an unlocking key sequence, i.e., multiple correct keys over several test cycles during the test initiation process, for enabling the debug/scan output registers.

5 TREEHOUSE Provisioning and Test Protocol

We now detail the various steps involved in the provisioning and test phases of the 3DIC. We describe the interaction between the components of `TREEHOUSE` during the provisioning and test phases. We first illustrate the various operations during the layer-level (pre-bond) test phase and then the steps during the post-bond and post-packaging test phases.

5.1 Pre-bond (Layer-level) Testing

During the pre-bond testing, the test process focuses on testing the integrity of the TSVs and limited testing of the functional IPs in the layer/die. This process, known as Known Good Die testing [24], is used to increase the overall yield of the 3DIC design by eliminating non-functional layers/dies. In the `TREEHOUSE` implementation, the scan protection modules help the design house protect the integrity of the 3DIC during this phase. Figure 6(a) shows the interaction between the design house, security wrapper, and the IPs during the layer-level testing process. During the layer-level testing, the design house first unlocks the scan ports to enable the testing process. This is achieved by setting the Mode bits to TREE Mode (step 2) and applying the Mode enable vectors for enabling Test Unlocking Mode (line 3). Once the Unlocking mode is enabled, the test unlocking key patterns for each IP is applied (line 4-5). A counter in the IPs counts the number of key bits checked during this phase. A count of M indicates that M $\mathcal{N}$-bit correct keys have been applied. The counter's output is used to enable the 'AND' gate connected to the scan-out port of the IP. The value of the scan-out port will be zero until the counter reaches a specified value indicated by the number of key sequences. A test vector containing the incorrect key or an incorrect number of mismatches does not allow any scan output and thus prevents valuable information from being leaked. Once the scan ports are unlocked, the design house performs Scan Authentication (Steps 6-9). The

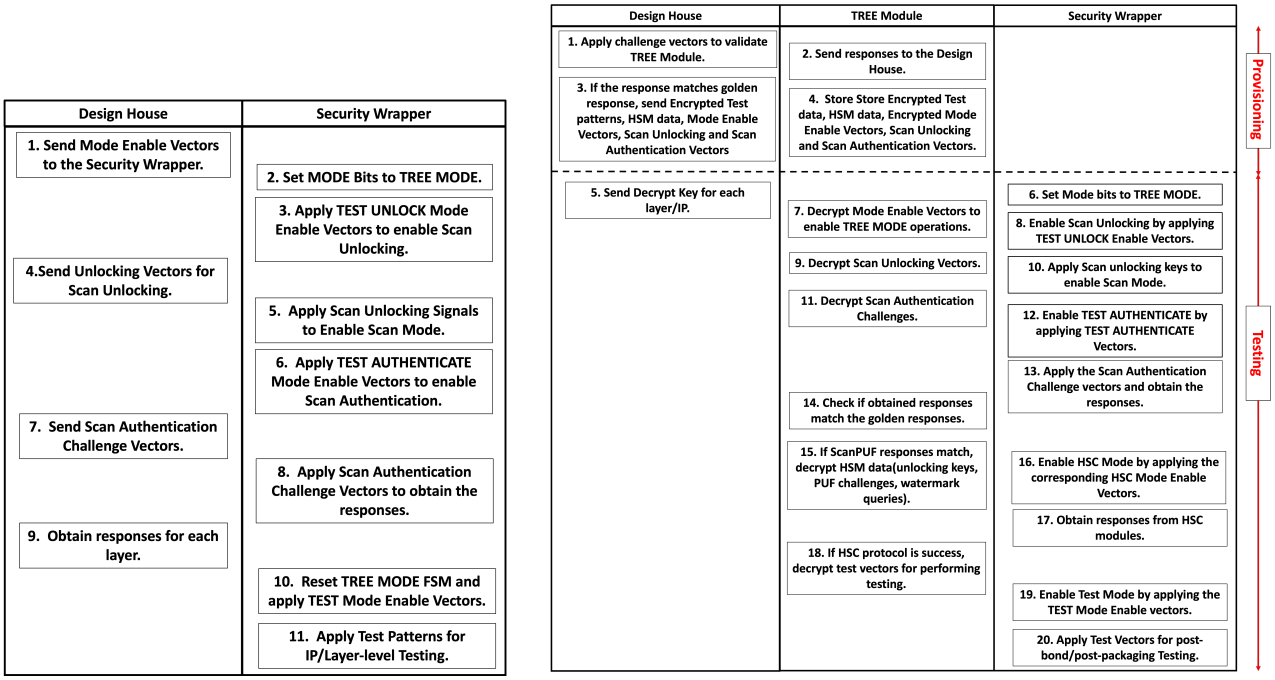

Fig. 6: (a) The interaction between the design house and the security wrapper during layer-level (pre-bond) testing. (b) The interaction between the design house, the TREE Module, and the security wrapper during post-bond and post-packaging phases.

design house then ensures that the IP is in authentication mode by applying the appropriate mode enable vectors, which would cause the TREE MODE FSM to transition to Test Authentication mode (Step 6). Once the PUF mode is enabled, the challenge vectors are applied for each IP (Step 7-8). The transition delays in select flops are observed, and the signature is generated using Equation 1. Thus, the location of the scan flops also forms a part of the input challenges.

$$Sig_i = \begin{cases} 0 & \text{if } t_{challenge} > t_{interval} \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

Once the Scan Authentication is complete, the TREE MODE FSM is set to TEST mode by setting the `TREE_MODE_RESET` and the WRSTN pins to the appropriate values. The design house then applies the test mode enable vectors and obtains the corresponding responses for layer-level testing (Steps 10-11). The design house uses the response obtained from the layers to create a composite device signature which is then used for encrypting the metadata such as Mode Enable Vectors, HSM data, Test patterns and golden responses which are then transferred to the TREE Layer for enabling Post-bond and Post-packaging testing.

5.2 Provisioning

Figure 6(b) shows the interaction between the various elements of the `TREEHOUSE` framework and the 3DIC during the provisioning and the post-bond and post-packaging testing phases. Once the pre-bond testing is done, the different layers are packaged at the integration facility. During this phase, the TREE layer is also integrated into the design. During the subsequent testing phases, the HSCs in each IP need their corresponding HSM data. For example, the locked IPs in layer three and layer one must be unlocked before the test equipment can apply the test patterns. Similarly, the IPs containing PUF and watermarks need to be authenticated. Since the testing process includes multiple phases (post-bond, post-packaging), the HSM data must be applied several times. This process of providing the HSM data for the HSCs to enable the security protocols such as unlocking and authentication is known as provisioning. To allow safe and secure provisioning, the design house transfers the HSM data for each layer from the design house in an encrypted fashion(Step 1-3). The design house encrypts the HSM data of each IP using a composite signature generated using the response obtained from Scan PUF modules before transferring it to the TREE module. This prevents eavesdropping attacks and ensures that the designer can securely transfer the HSM data from the design house to the TREE module. Before the provisioning/test process begins, the design house verifies if the TREE Module is authentic by obtaining the response of the PUF from the TREE Module (lines 1-2). If the response matches the golden response, the design house then transfers the encrypted metadata to the TREE Module for storage (lines 3-4).

5.3 Post-bond and Post-packaging Testing

To initiate the testing process, the design house sends the key for decrypting the metadata for each layer and IP(line 5). The TREE Module then sets the TREE MODE FSM for the corresponding layer to TREE Mode (Step 6). The TREE Module then obtains the Decrypt Key for each layer from the design house and decrypts the Mode Enable Vectors (Step 7). To enable Test unlocking, the corresponding mode select vectors are applied (Step 8) followed by the application of the decrypted Scan Unlock Keys (Step 9-10). Once the Scan Unlocking process is complete, the TREE Module sets the TREE MODE FSM to the Test Authentication Mode by applying the corresponding Mode Select Vectors (Step 11-12) and checks if the obtained signature matches the golden

TABLE 2: The baseline Area, Gate Count and Power values of the 3DIC design described in Figure 1(a).

Module	Area (sq.μm)	Gate Count (KGates)	Power (mW)
FIR Accelerator with PUF	187102.4	51.45	2.39
Locked GPS Accelerator with Watermark	584532.38	23.54	8.52
Locked AES	954700.25	410.37	13.09
Leon3mp Processor	2221368.63	691.77	86.16
Total	3947703.66	1389.04	110.16

response obtained during the layer-level testing (Steps 13-14). If the response matches, the HSM data is decrypted and applied. For example, in the case of the Logic Locked Crypto IP, the unlocking Keys are obtained from the Key Management Unit and then decrypted (Step 15). The appropriate Mode Select Vectors are applied before the decrypted HSM data is applied to the IP (Steps 16-17). In the case of HSC protocols like PUFs that produce a response, the response is obtained and then compared with the golden signature collected during the layer-level testing. Once the IPs are unlocked/authenticated, the mode select vectors for testing are applied to enable Test Mode along with setting the `TREE_MODE_RESET` and WRSTN pins to the appropriate values. If any of the security protocols in Steps 9, 13-14, or Steps 18 fail then TREE Module disables access to that corresponding layer until further audit can be performed.

6 Experimental Setup

In this Section, we detail the experimental setup used for evaluating the `TREEHOUSE` mechanism. We describe the various components used for constructing our 3DIC design, the Synthesis configurations and the system setup used for evaluating our experiments.

6.1 Hardware Setup

We describe the configuration of the various IPs used for our 3DIC design and the configuration of our TREE Module.

6.1.1 3DIC configuration

The baseline 3DIC configuration that we use for our analysis is detailed in Table 2. We assume that the 3DIC design consists of three layers. The layer-wise composition of the design is as follows: The bottom layer consists of the Leon3mp Processor. The middle layer contains a logic locked GPS IP with watermark and a FIR Module with a PUF. The topmost layer consists of a crypto-IP (AES) that is logic locked. The designs are sourced from the IWLS 2005 Benchmark suite [37] and the mit-cep benchmark designs [38]. We augment the IPs to integrate IEEE1500 wrapper functionality and the security wrapper interface for supporting the authentication, scan and functional unlocking capabilities. We also assume that the IPs are locked using the sequential locking algorithm described in [39]. During the locking operation, the alogrithm identifies 48 data path Flip-Flops and inserts welve additional Flops to lock the circuit. The GPS IP contains 66 unlocking key patterns applied over these 60 Unlocking Flops. For the AES IP, the locking algorithm modifies 325 existing Flops and adds 27 additional Flops. The Unlocking Key sequence contains 1334 input vectors applied over these 352 Flops. For Watermark insertion, we use the challenge-response based watermarking scheme detailed in [40] and use the PUF reported in [41].

6.1.2 `TREEHOUSE` Configuration

The TREE unit was generated using a RISC-V microcontroller [42]. The SCR1 core is a lightweight 32-bit microcontroller containing five pipeline stages. The microcontroller supports both AXI and AHB-bus protocols. In our implementation, we utilize the AHB interface. The core consists of 16 Interrupt lines and a JTAG interface. The core is augmented with additional memory maps, control registers, and peripheral features required to support the scan-unlocking, authentication, device-unlocking operations. We use a 128KB main memory module in the TREE layer. We use 32 registers in the security wrapper to ensure seamless key transfer. One of the registers is a Mode Register that can be configured by the TREE MODE FSM to control the nature of operation. The Mode Register also contains additional bits that can be used to control the nature of key transfer (burst-mode or serial mode) and the HSC mode (Authentication, Watermark verification,unlocking etc.). The remaining 31 registers are used to transfer the Functional Unlocking Keys, the challenge vectors for authentication. Our implementation of `TREEHOUSE` uses the scan PUF architecture proposed in [36]. For performing the scan authentication we assume that the measurements are made over 32 iterations across each Scan path over 8-phase shifted clock frequencies. The responses obtained are then compared to produce a 128-bit signature. The scan lock architecture from [34] is used for Scan locking operation. The scan lock configuration that we implement requires 16 32-bit unlocking test vectors each of which produce a 16-bit response vector.

6.2 Synthesis Settings

The designs were synthesized using Synopsys Design Compiler version Version R-2020.09-SP5 for Linux64. We optimized the design for optimal area and delay by using the set_max_delay and set_max_area constraints using the gscl45nm freepdk library using the standard PVT settings of (1.1V, $27^{\circ}C$, and typical process corner). The time period of the SoC was set at 10.43 ns to avoid timing violations. The TREE Module was also synthesized with the same time constratins. We did not observe any timing violations for the TREE module when synthesized with the same time period. For the power estimation, we use the Total power Values reported by the synthesis tool. The IPs are then augmented with the locking and authentication mechanisms and synthesized to estimate the overhead results. Table 2 contains the baseline area, gate-count, and power values for each layer in the 3DIC design.

6.3 System Setup

We performed our synthesis and simulation runs on an 8-core Intel Core i7 CPU running at 3GHz running Linux kernel version 5.4.0-81-generic. To simulate the brute-force attack on the Scan Unlocking Module we implemented a random pattern generator in C. The random patterns were then fed to the IP using a testbench setup written in Verilog.

7 Results

In this Section, we evaluate the overheads of the `TREEHOUSE` framework and the security guarantees.

7.1 Impact of `TREEHOUSE` on Area and Power

We first evaluate the hardware overheads of the various components of the TREE Module shown in Figure 4(a). We present the overall Area, Gate-count and Power overheads

TABLE 3: The Layer-level Area, Gate Count and Power overheads incurred by the components of treelock for the 3DIC design described in Figure 1(a).

Component	Baseline			with TREEHOUSE			Percentage Overheads		
	Area (sq.μm)	Gate Count (KGates)	Power (mW)	Area (sq.μm)	Gate Count (Gates)	Power (mW)	Area (sq.μm)	Gate Count (KGates)	Power (mW)
Locked GPS Accelerator with Watermark, Fir Module with PUF	597798.08	238.95	8.65	599105.17	239.13	8.66	0.22	0.07	0.15
Locked AES	954700.25	410.38	13.09	956007.34	410.55	13.10	0.13	0.04	0.10
Average							0.18	0.06	0.13

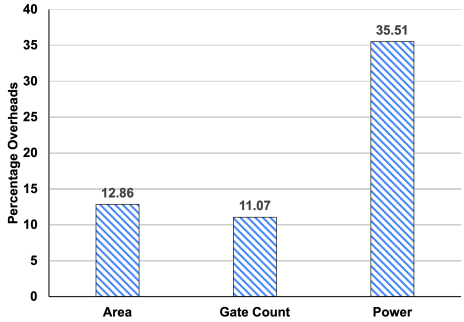

Fig. 7: Impact of TREEHOUSE on overall Area, Gate Count and Power of the 3DIC design.

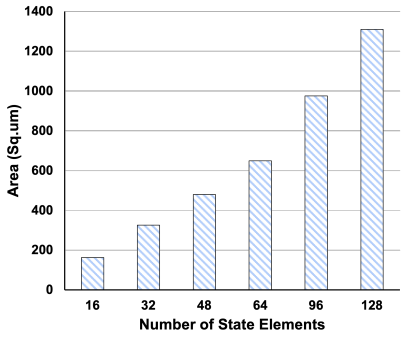

Fig. 8: Impact of increasing the state-elements in the TREE MODE FSM on the Area.

in Figure 7. We also present the Layer-level Area, Gate-count, and Power overheads in Table 3. The TREE Module accounts for 166K Gates and consumes 67.32 mW of power as shown in Table 4. We observe that the Modules for enforcing security policies, such as Authentication Control Unit, Key Management Unit, and Encryption Unit, account for only 35% of the total power and 12% of the entire Area. We also observe that the components of the TREEHOUSE framework only incur negligible impact on the layer-level power, Area, and gate count. We show that for the baseline architecture we consider, the TREEHOUSE framework incurs a layer-level penalty of less than 1% on Area, power, and gate count. We also present these values for each component in an IP in Table 7.

From the Tables it can be observed that these three modules incur negligible Area and power overheads compared to the layer-level power consumption as well as overall power consumption of the 3DIC. We now assess the impact of the design choices for individual elements of the TREEHOUSE framework.

TABLE 4: The baseline Area, Gate Count and Power values for each component in the TREE Module described Figure 4(a).

Component	Area (sq.μm)	Gate Count (KGates)	Power (mW)
Authentication Control Unit	4816.42	1.24	0.46
Key Management Unit	25494.72	7.37	0.48
Memory Module	40463.98	11.29	2.93
Encryption Unit	97105.68	29.45	0.7
Ethernet Controller	187102.34	51.45	2.39
RISCV Core	201786.32	66.08	60.36
PUF	127.32	0.13	0.002
Total	556896.78	167.02	67.32

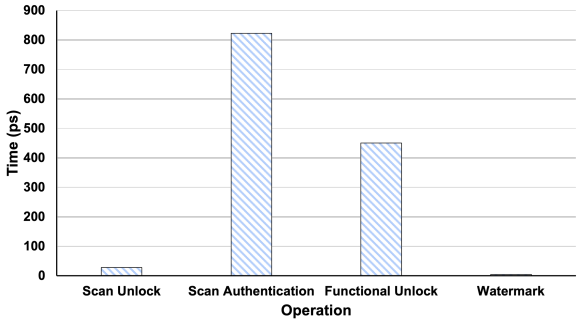

Fig. 9: The runtime overheads for various security protocols in the Locked GPS IP with a watermark.

7.1.1 Impact of the TREE MODE FSM

The TREE MODE FSM is vital to ensure the overall safety of the TREEHOUSE framework. The TREE MODE FSM provides the TREE Module with a degree of control in the whole testing and provisioning process and must be designed accordingly. The TREE MODE FSM is designed with two key objectives: a) to make it harder for the attacker to access the HSC modules, b) to achieve maximum security through non-overlapping Mode Enable Vectors. While a large number of state elements provide the designer with maximum security since the size of the Mode Enable Vectors increases along with the probability of finding non-overlapping sequences. A large FSM incurs penalties in both time and Area. It is crucial to ensure that the TREE MODE FSM does impact the testing process significantly. A 128-state FSM, while being harder for the attacker to compromise, incurs a significant delay penalty during the provisioning and testing phases, thus becoming an impediment for the designer. We evaluate the TREE Mode Configurations for the Area overhead in Figure 8. We select a 48-state FSM in our implementation since it offers the right balance between security and overheads.

TABLE 5: The area, gate count and Power values of each component in the TREE MODE FSM and the Scan Protection Modules.

Component	Area (Sq.μm)	Gate Count (Gates)	Power (μW)
TREE MODE FSM	320.32	32	2.18
Scan Unlocking	405.04	72	5.37
Scan Authentication	268.8	209	2.85

TABLE 6: The area, authentication time, and gate count overheads for different configurations of the Scan Authentication Module for the GPS design.

No.of Scan Chains	FFS per Scan Chain	Time For Authentication (ps)	Area (Sq.μm)	Gate Count (Gates)
1	7351	207004.16	89.6	81
4	1838	12957.59	179.2	145
16	460	822.64	268.8	209
32	230	202.41	354.4	273

7.1.2 Impact of the Scan Authentication IP

We use the Scan PUF implementation from [36] in our implementation. The Scan PUF module works by estimating the delay difference in transition delays at select flops. We estimate the impact of the number of Scan paths in a given IP on the Authentication process. We consider the layer 2 IP (locked GPS module with watermark). The baseline design has 7351 Scan-Flops. We vary the number of Scan paths in the design and observe the impact on Authentication time, Area, and Gate Count. We consider a 128-bit response size. We notice that a single Scan Flop design requires significantly high Authentication time. This is because each challenge vector has to be applied over the entire scan path. Multiple Scan paths incur lesser overheads for authentication, but the Area overhead increases due to the additional Logic required for comparing the delay differences obtained for the target Flops over different paths. We present the Area, Authentication Time, and Gate Count overheads in Table 6.

7.2 Impact of TREEHOUSE on Timing

We quantify the overall time required for performing the various steps of the TREEHOUSE algorithm in Figure 9 on the logic-locked GPS IP with a watermark. We observe that the Scan Authentication requires the most time. This is due to the size of the signature (128-bits) and the number of Scan chains (16). As mentioned earlier, varying the scan chains would decrease the Authentication time but increase the area and power overheads. For functional unlocking, a set of 66-input patterns are applied over the 60 Unlocking Flip-Flops to unlock the design. The watermark verification takes only one clock cycle (4.5ps) since the watermark inserted is a combinational logic and thus does not incur significant overheads.

8 Security Analysis

The overall security of the TREEHOUSE framework is due to the various modules interacting to ensure the security of the 3DIC. Thus compromising any specific module would not give the attacker a significant advantage. For example, the attacker could deploy attacks [43], [44] against the Scan Unlocking module to recover the keys. Such attacks require the chip to be in test mode and employ combinational locking, which is not the case in this scenario. Similarly, the attacker could employ reverse-engineering attacks to recover the structure of the ScanPUF modules. However, the attacker would still need to know the exact location of the flops used for generating the authentication signature. Thus the attacker can only employ brute-force attacks on the TREEHOUSE framework. We now analyze the resilience of TREEHOUSE against brute-force attacks.

TREEHOUSE has three components viz, the TREE Module, the Scan Protection elements, and the TREE MODE FSM operate in tandem to contribute to the overall security of the design. We analyze the security of the overall TREEHOUSE by characterizing the resistance of the proposed architecture elements against counterfeiting attacks, and key retrieval attacks.

TABLE 7: The area, gate count and power values for different components of TREEHOUSE for a single layer (Locked GPS with Watermark).

Component	Area (sq.μm)	Gate Count (Gates)	Power (μW)
Watermark	150.08	32	1.6
Scan Unlock	405.04	72	5.37
Scan Authentication	268.8	209	2.85
Functional Unlock	120.02	12	2.3
TREE MODE FSM	480.48	48	2.18
Baseline design	584262.28	235398	85270.1
Total	585839.47	235621	85287.4

8.1 Security of TREE MODE FSM and Scan Protection Modules

The Security of the TREE MODE FSM and the Scan protection modules ensures the overall safety of the design in both the pre-bond and post-bond testing phases. The TREE MODE FSM and the Scan unlocking modules play a vital role in the pre-bond testing phases. To compromise the layer's integrity in the pre-bond testing phase, the attacker can perform one of three attacks: A brute-force key recovery attack. A tampering attack to compromise the Scan signature during the authentication phase. A counterfeiting attack.

8.1.1 Brute-force key recovery attack

Since the attacker at the foundry does not know the Mode Enable Vectors, the attacker can try to either guess the Mode Enable Vectors or try a brute-force guessing attempt. For a TREE MODE FSM containing α-state elements, the Probability of a successful guess P_α is given by Equation 2. The attacker would also need to guess the unlock key sequence for the scan unlocking module. For a series of $\mathbf{M}$ state elements and a sequence of $\mathbf{N}$ unlocking vectors, the probability of a successful attack,P_β, is given by Equation 3.

$$P_{alpha} = \frac{1}{2^{\alpha}} \tag{2}$$

$$P_{beta} = \frac{1}{2^{\mathbf{M}\times\mathbf{N}}} \tag{3}$$

The attacker could try to bypass the TREE MODE FSM and attempt to unlock the Scan Unlock module. As shown in Figure 4(b), the Scan Unlock Module receives inputs via the Test ports, and thus the attacker could attempt to brute-force the Scan Unlock module by feeding in random test patterns. We simulate this scenario and present the results of the brute-force attack on the GPS IP in Figure 10. We

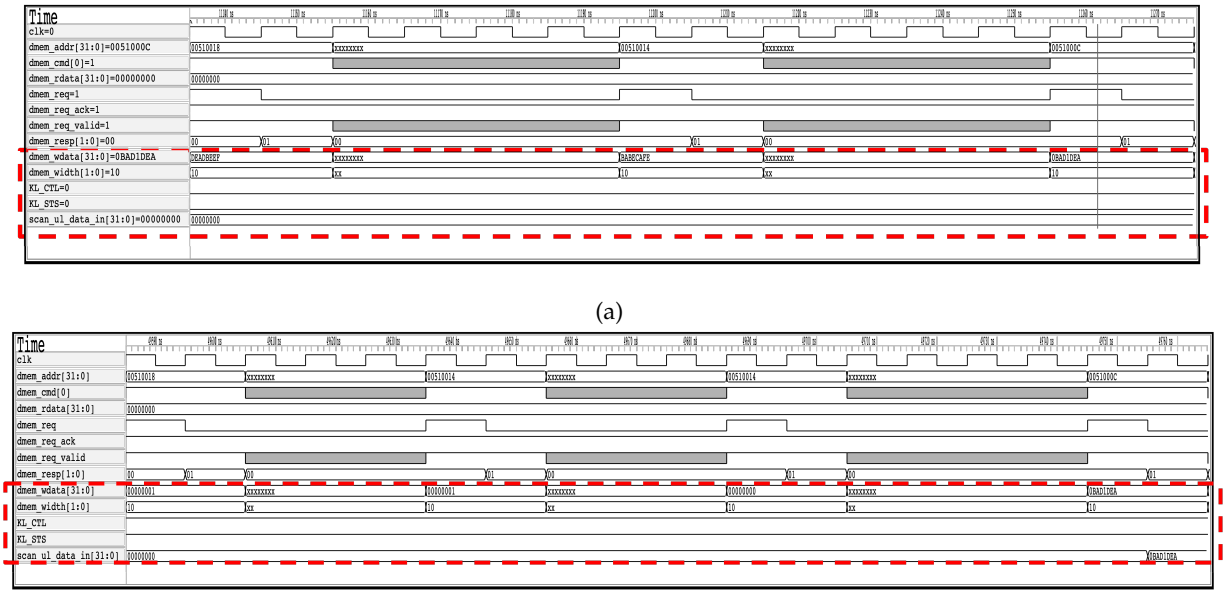

Fig. 10: A brute force attack on the Scan unlocking Module of the GPS IP during the provisioning phase. The attacker tries to write the unlocking key "0xbad1dea" into the scan unlocking register without enabling the scan unlocking mode in (a). The data is not written into the register. In (b) the scan unlocking mode is enabled by applying the appropriate Mode Enable Vectors to set the KL_CTL and KL_STS bits in the security wrapper and the unlocking key bits are written into the scan unlock register.

use a random pattern generator implemented in C connected to a testbench to mimic the attacker. The design contains two bits in the security wrapper (KL_CTL and KL_STS) for implementing the mode change. The GPS IP has four security operations (Scan Authentication, Scan Unlock, Functional Unlock, and Watermarking). Each process is triggered by setting the mode bits to their appropriate states. The Scan unlocking mode is enabled by setting the KL_CTL and KL_STS bits to 11. In Figure 10(a), the attacker attempts to feed in the unlocking Key "0xbad1dea" to the scan unlocking register in the security wrapper by applying random mode enable vectors. The Scan unlocking mode is not allowed; thus, the value in the scan unlocking register (scan_ul_data_in) remains unchanged. In Figure 10(b), the data is written to the scan unlocking register by enabling the Scan unlocking mode. Thus to successfully enact a brute-force attack on the Scan unlocking module, the attacker needs to guess the Mode Enable Vector and the unlocking key sequence for the Scan Unlock module. Thus the total probability of success P_{brute} is given by the Equation 5.

$$P_{brute} = P_{\alpha} \times P_{\beta} \tag{4}$$

$$P_{brute} = \frac{1}{2^{\alpha \times \mathbf{M} \times N}} \tag{5}$$

Thus for a 48-state TREE MODE FSM, and a Scan unlocking sequence of 16 patterns applied over 16-inputs, the overall probability P_{brute} comes to 1 in 2^{12288}, which is extremely small.

8.2 Tampering Attack on ScanPUF Module

To effectively compromise the ScanPUF signature, the attacker would need to know the exact Flip-Flops whose delay is used to generate the response. Without prior knowledge of the location, the attacker would resort to a brute-force attack. For a design with $\mathbf{F}$ Scan Flops containing $\mathbf{S}$ signature flops split into $\mathbf{P}$ scan paths, each containing $\mathbf{K}$ Scan Flops, the probability of an attacker tampering with the Scan signature P_{tamper} is given by Equation 6.

$$P_{tamper} = \frac{\mathbf{S}}{\mathbf{P} \times \mathbf{K}} \tag{6}$$

The GPS design contains 7315 Scan Flops with 16 Scan paths and 32-target flops and thus has a success probability of 0.0043. The attacker could try to tamper an entire scan path to bias the signature. However, the attacker does not know the challenge vectors and the responses to leverage this information into a successful attack.

8.3 Counterfeiting Attack

The attacker in an untrusted foundry can counterfeit the elements of the layer(s) they fabricate. To successfully do so, The attacker would need to know the exact scan-flops used for authentication and the shift-frequencies used for producing the signatures. The attacker can also attempt to replicate the signature of the Scan-PUF elements. However, the attacker cannot capture the source of entropy. The attacker must also ensure that delay variation is preserved across the locations. In addition to the location of scan flops, the attacker must also capture the test patterns used for generating the response. Consider a design with D with F scan flip-flops. Let T_N be the number of input patterns applied over T_{shift} clock frequencies to generate a $M-$ length response vector. The effort the attacker requires to carry out a successful attack $P_{response}$ is given by Equation 7.

$$P_{response} = \frac{1}{2^{T_N \times T_{shift}}} \tag{7}$$

8.4 Attack on the Functional Locking Module

The attacker could try to recover the Functional unlocking key for an IP/Layer. To do so, the attacker would need to guess the exact number of obfuscation flip-flops incorporated into the design and apply the correct input sequence to unlock the obfuscation FSM. Consider a circuit with N Flip-Flops of which N_{obf} are the obfuscation flip-flops. Assuming a brute-force approach, the attacker's effort is enumerated by the Equation 8, where ${}^{N}C_{N_{obf}}$ represents the number of ways of selecting N_{obf} Flip-Flops out of a given sequence of N Flops.

$$P_{Key} = \frac{1}{{}^{N}C_{N_{obf}}} \tag{8}$$

Considering an obfuscation FSM of length N_{obf} with K unlocking Key sequences, the number of tries required by the attacker to successfully break the functional locking scheme is given by Equation 9. Similar to the scan locking scheme, even relatively small values of N_{obf} and K give a very low probability of success for the attacker. For a example, an obfuscation FSM containing 16 obfuscation Flops with an unlocking sequence of length 16 would require 2^{256} tries by the attacker.

$$P_{success} = \frac{1}{2^{N_{obf} \times K}} \tag{9}$$

9 Scalability and Interoperability of TREEHOUSE

The need for energy-efficient designs is giving rise to custom stacking techniques to minimize overall power consumption, thermal effects, etc. Unlike the existing 3DIC security solutions, TREEHOUSE is not tied to a given 3DIC architecture. The TREE module developed for a particular architecture can be reused with little to no changes to another technique such as intersposer-based design. In this Section, we provide a short discussion on the interoperability of the various components of TREEHOUSE.

We now study the interoperability of the TREEHOUSE framework. Any security countermeasure in the 3DIC ecosystem must be flexible. A security countermeasure that is highly coupled to a specific design technique or design component might not be an efficient solution in the long run. Interoperability thus becomes an interesting metric for analyzing a given countermeasure's efficiency. We study the interoperability of each component in TREEHOUSE in detail.

9.1 Interoperability of the TREE Module

The current implementation of TREEHOUSE uses the scr1 RISC-V core. However, the TREEHOUSE protocol is both ISA and architecture agnostic. Thus, the TREE module could be implemented using any lightweight microcontroller as long as the interface of the newer controller matches that of the TREE module.

9.2 Interoperability of Scan Protection Modules

TREEHOUSE is agnostic to the underlying authentication and locking protocols implemented. The architecture of TREEHOUSE is quite flexible and can support various authentication protocols and locking techniques operating at the scan and functional abstractions.

Thus, we see that TREEHOUSE is not limited by the design components and the 3DIC architecture. TREEHOUSE is also not affected by the changes in technology nodes and can seamlessly scale to support both increases in the number of layers and the number of IPs.

10 Conclusion and Future Work

The distributed, non-standard nature of the 3DIC supply-chain has given rise to various attacks that threaten to compromise the integrity of a 3DIC design thus motivating the need for robust, secure 3DIC architectures. To that end, we proposed TREEHOUSE a lightweight flexible 3DIC provisioning architecture that can mitigate reverse-engineering, and IP counterfeiting attacks originating at the untrusted foundry and testing facilities. The latter threat model has not been considered before during security analysis of 3DIC systems. We showed the feasibility of such an attack as well as a possible collusion-based attack. We discussed the components of the TREEHOUSE framework along with the provisioning process. We showed that TREEHOUSE does not incur significant overheads, and analyze the security guarantees of the TREEHOUSE framework. We show that the proposed framework is robust, and can be seamlessly integrated into any design with minimal effort.

In future work, we plan on improving the performance of the TREEHOUSE by utilizing a hybrid implementation as opposed to a centralized implementation (low-overheads but is less secure) or distributed implementation (huge-overheads but is more secure). We also plan on modifying the TREEHOUSE implementation to integrate security constraints within a given thermal budget.

References

[1] P. Gu, D. Stow, R. Barnes, E. Kursun, and Y. Xie, "Thermal-aware 3d design for side-channel information leakage," in *2016 IEEE 34th International Conference on Computer Design (ICCD)*. IEEE, 2016, pp. 520–527.

[2] J. Dofe, Z. Zhang, Q. Yu, C. Yan, and E. Salman, "Impact of power distribution network on power analysis attacks in three-dimensional integrated circuits," in *Proceedings of the on Great Lakes Symposium on VLSI 2017*, 2017, pp. 327–332.

[3] S. F. Mossa, S. R. Hasan, and O. Elkeelany, "Hardware trojans in 3-d ics due to nbti effects and countermeasure," *Integration*, vol. 59, pp. 64–74, 2017.

[4] S. R. Hasan, S. F. Mossa, O. S. A. Elkeelany, and F. Awwad, "Tenacious hardware trojans due to high temperature in middle tiers of 3-d ics," in *2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2015, pp. 1–4.

[5] F. Imeson, A. Emtenan, S. Garg, and M. Tripunitara, "Securing computer hardware using 3d integrated circuit (IC) technology and split manufacturing for obfuscation," in *22nd USENIX Security Symposium (USENIX Security 13)*. USENIX Association, Aug. 2013, pp. 495–510.

[6] J. Valamehr, T. Sherwood, R. Kastner, D. Marangoni-Simonsen, T. Huffmire, C. Irvine, and T. Levin, "A 3-d split manufacturing approach to trustworthy system development," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 4, pp. 611–615, 2013.

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

[8] J. Dofe, C. Yan, S. Kontak, E. Salman, and Q. Yu, "Transistor-level camouflaged logic locking method for monolithic 3d ic security," in *2016 IEEE Asian Hardware-Oriented Security and Trust (AsianHOST)*. IEEE, 2016, pp. 1–6.

[9] M. Wang, A. Yates, and I. L. Markov, "Superpuf: Integrating heterogeneous physically unclonable functions," in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2014, pp. 454–461.

[10] S. Alhelaly, J. Dworak, T. Manikas, P. Gui, K. Nepal, and A. L. Crouch, "Detecting a trojan die in 3d stacked integrated circuits," in *2017 IEEE North Atlantic Test Workshop (NATW)*. IEEE, 2017, pp. 1–6.

[11] J. Dofe and Q. Yu, "Exploiting pdn noise to thwart correlation power analysis attacks in 3d ics," in *2018 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP)*. IEEE, 2018, pp. 1–6.

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

[13] M. Bilzor, "3d execution monitor (3d-em): Using 3d circuits to detect hardware malicious inclusions in general purpose processors," in *Proceedings of the 6th International Conference on Information Warfare and Security*, 2011, p. 288.

[14] Y. Xie, C. Bao, and A. Srivastava, "Security-aware 2.5d integrated circuit design flow against hardware ip piracy," *Computer*, vol. 50, no. 5, pp. 62–71, 2017.

[15] J. Knechtel and O. Sinanoglu, "On mitigation of side-channel attacks in 3d ics: Decorrelating thermal patterns from power and activity," in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE, 2017, pp. 1–6.

[16] T. Huffmire, T. Levin, M. Bilzor, C. E. Irvine, J. Valamehr, M. Tiwari, T. Sherwood, and R. Kastner, "Hardware trust implications of 3-d integration," in *Proceedings of the 5th Workshop on Embedded Systems Security*, 2010, pp. 1–10.

[17] J. Dofe, Q. Yu, H. Wang, and E. Salman, "Hardware security threats and potential countermeasures in emerging 3d ics," in *2016 International Great Lakes Symposium on VLSI (GLSVLSI)*, 2016, pp. 69–74.

[18] J. Dofe, C. Yan, S. Kontak, E. Salman, and Q. Yu, "Transistor-level camouflaged logic locking method for monolithic 3d ic security," in *2016 IEEE Asian Hardware-Oriented Security and Trust (AsianHOST)*, 2016, pp. 1–6.

[19] Y. Xie, C. Bao, C. Serafy, T. Lu, A. Srivastava, and M. Tehranipoor, "Security and vulnerability implications of 3d ics," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 2, pp. 108–122, 2016.

[20] J. Knechtel, "Hardware security for and beyond cmos technology: an overview on fundamentals, applications, and challenges," in *Proceedings of the 2020 International Symposium on Physical Design*, 2020, pp. 75–86.

[21] Z. Zhang and Q. Yu, "Modeling hardware trojans in 3d ics," in *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2019, pp. 483–488.

[22] J. Dofe, P. Gu, D. Stow, Q. Yu, E. Kursun, and Y. Xie, "Security threats and countermeasures in three-dimensional integrated circuits," in *Proceedings of the on Great Lakes Symposium on VLSI 2017*. Association for Computing Machinery, 2017, p. 321–326.

[23] P. Gu, D. Stow, P. Mukim, S. Li, and Y. Xie, "Cost-efficient 3d integration to hinder reverse engineering during and after manufacturing," in *2018 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*. IEEE, 2018, pp. 74–79.

[24] D. Appello, P. Bernardi, M. Grosso, and M. S. Reorda, "System-in-package testing: problems and solutions," *IEEE Design & Test of Computers*, vol. 23, no. 3, pp. 203–211, 2006.

[25] E. J. Marinissen, T. McLaurin, and H. Jiao, "Ieee std p1838: Dft standard-under-development for 2.5 d-, 3d-, and 5.5 d-sics," in *2016 21th IEEE european test symposium (ETS)*. IEEE, 2016, pp. 1–10.

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

[27] "Ieee standard testability method for embedded core-based integrated circuits," *IEEE Std 1500-2005*, pp. 1–136, 2005.

[28] R. Kapur, M. Lousberg, T. Taylor, B. Keller, P. Reuter, and D. Kay, "Ctl the language for describing core-based test," in *Proceedings International Test Conference 2001 (Cat. No.01CH37260)*, 2001, pp. 131–139.

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

[30] 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 *2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2021, pp. 124–135.

[31] A. Stern, H. Wang, F. Rahman, F. Farahmandi, and M. Tehranipoor, "Aced-it: Assuring confidential electronic design against insider threats in a zero trust environment," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021.

[32] M. Yasin, J. J. Rajendran, O. Sinanoglu, and R. Karri, "On improving the security of logic locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 9, pp. 1411–1424, 2016.

[33] P. Chakraborty, J. Cruz, A. Alaql, and S. Bhunia, "Sail: Analyzing structural artifacts of logic locking using machine learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3828–3842, 2021.

[34] S. Paul, R. S. Chakraborty, and S. Bhunia, "Vim-scan: A low overhead scan design approach for protection of secret key in scan-based secure chips," in *25th IEEE VLSI Test Symposium (VTS'07)*, 2007, pp. 455–460.

[35] S. Potluri, A. Aysu, and A. Kumar, "Seql: Secure scan-locking for ip protection," in *2020 21st International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2020, pp. 7–13.

[36] Y. Zheng, A. R. Krishna, and S. Bhunia, "Scanpuf: Robust ultralow-overhead puf using scan chain," in *2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2013, pp. 626–631.

[37] C. Albrecht, "Iwls 2005 benchmarks," Jun 2005. [Online]. Available: https://iwls.org/iwls2005/benchmarks.html

[38] MIT, "Common evaluation platform," https://github.com/mit-ll/CEP, 2018.

[39] R. S. Chakraborty and S. Bhunia, "Harpoon: An obfuscation-based soc design methodology for hardware protection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 10, pp. 1493–1502, 2009.

[40] A. Nair, P. SLPSK, C. Rebeiro, and S. Bhunia, "Signed: A challenge-response based interrogation scheme for simultaneous watermarking and trojan detection," *arXiv preprint arXiv:2010.05209*, 2020.

[41] C. Vega, P. SLPSK, S. D. Paul, and S. Bhunia, "Melpuf: Memory in logic puf," *arXiv preprint arXiv:2012.03162*, 2020.

[42] Syntacore, "Scr1," Jun 2021. [Online]. Available: https://github.com/syntacore/scr1

[43] L. Alrahis, M. Yasin, N. Limaye, H. Saleh, B. Mohammad, M. Al-Qutayri, and O. Sinanoglu, "Scansat: Unlocking static and dynamic scan obfuscation," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 4, pp. 1867–1882, 2021.

[44] N. Limaye and O. Sinanoglu, "Dynunlock: Unlocking scan chains obfuscated using dynamic keys," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2020, pp. 270–273.

Patanjali SLPSK received the master's degree and the Ph.D. degree with a focus on energy-efficient computing and hardware security from the IIT Madras, Chennai, India, in 2014 and 2019, respectively. He joined the University of Florida (UF), Gainesville, FL, USA, as a post-doctoral researcher, where he is currently working with Dr. Swarup Bhunia with the Warren B. Nelms Institute for the Connected World. His research interests include the Internet of Things, hardware security, and computer architecture.

Sandip Ray (SM'13) is a professor at the Warren B. Nelms Institute for the Connected World affiliated with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA, . Prior to that, he worked with NXP Semiconductors and Intel Strategic CAD Laboratories where, he led industrial research and R&D projects in pre-silicon and post-silicon validation of security and functional correctness of SoC designs, design-for-security and design-for-debug architectures, and security validation for automotive and the Internet-of-Things applications. His current research targets correct, dependable, secure, and trustworthy computing. He is the author of three books and over 100 publications in international journals and conferences. He has also served as a TPC Member of over 50 international conferences and as Guest Editor for several journals. He has a Ph.D. from University of Texas at Austin.

Swarup Bhunia (Senior Member, IEEE) received his B.E. (Hons.) from Jadavpur University, Kolkata, India, M.Tech. from the Indian Institute of Technology (IIT), Kharagpur, and Ph.D. from Purdue University, IN, USA. Currently, Dr. Bhunia is a professor and Semmoto Endowed Chair in the University of Florida, FL, USA. Earlier he was appointed as the T. and A. Schroeder associate professor of Electrical Engineering and Computer Science at Case Western Reserve University, Cleveland, OH, USA. He has over ten years of research and development experience with over 200 publica- tions in peer-reviewed journals and premier conferences. His research interests include hardware security and trust, adaptive nanocomputing and novel test methodologies