iPROBE: Internal Shielding Approach for Protecting Against Front-side and Back-side Probing Attacks

Minyan Gao, M Sazadur Rahman, Nitin Varshney,
Mark Tehranipoor, *Fellow, IEEE* and Domenic Forte, *Senior Member, IEEE*

***Abstract*—Focused ion beam (FIB) has emerged as one of the most prevalent integrated circuit (IC) editing techniques in the past decade, greatly assisting post-silicon debugging and failure analysis. However, the confidentiality of security assets on electronic devices is gravely threatened by FIB-based probing attacks because of the FIB's fine-grained milling and deposition capabilities on silicon die. Although numerous solutions such as active shields and analog sensors have been proposed, they either incur prohibitively high overhead or suffer from low reliability, failing to provide protection against such threats in a feasible manner. In this paper, we propose a FIB-aware framework, iPROBE, as a set of computer-aided design (CAD) utilities to quantify the threats of FIB attacks on the target layout from both *front-side* and *back-side* at the *pre-silicon* stage. The subsequent shield nets/layer place-and-route are completely automated by iPROBE to minimize the quantified FIB vulnerability metric, so-called exposed area, allowing users achieving the optimal security level at a cost of minimal performance degradation, extra design efforts, and time consumption. Our experimental results show that the exposed area of security-critical nets, i.e., vulnerable regions inside the physical layout, to front-side and back-side probing attacks can be fully eliminated at low FIB aspect ratios with only 3% timing and area overhead. Moreover, we validate the results through the FIB experiments on a fabricated test chip covering both baseline and iPROBE-protected AES implementations at 65nm technology node, further demonstrating the effectiveness of iPROBE.**

I. Introduction

Physical attacks have become one of the most prevalent and threatening attack vectors for compromising confidentiality, integrity, and accessibility of modern electronic devices [1]. As illustrated in Figure 1, these attacks can be generally classified into three categories, i.e., non-invasive, semi-invasive, and invasive attacks. Non-invasive attacks do not need any specific sample preparations of the target device nor do they tamper with the chip package during the attack phase [2]. For example, an adversary can launch side-channel attacks to break cryptographic implementations by *passively* monitoring the unintentional physical emissions from the running electronic device, such as power [3] and electromagnetic (EM) [4]. On the other hand, fault injection attacks *actively* alter the original behaviors of the target integrated circuit (IC) to bypass the built-in security routine, escalate privilege, and/or leak secrets. This can occur by inducing clock/power/EM glitches. Unlike non-invasive attacks, semi-invasive attacks typically require depackaging. However, their passivation layers remain intact since physical contact to internal metal wires/transistors is unnecessary [2]. For instance, [5] shows how to flip the transistors in a static random access memory (SRAM) through

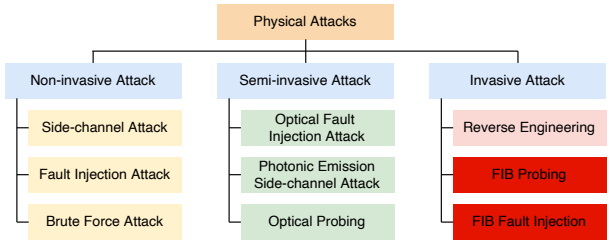


Fig. 1: The taxonomy of physical attacks.

optical fault injection. State-of-the-art optical probing techniques have also been demonstrated to steal on-chip FPGA bitstream decryption keys on 28nm Xilinx devices [6], [7].

Invasive physical attacks assume the strongest adversaries, i.e., those with not only the broad knowledge of the target devices but also the sophisticated equipment, time, and money for sample prep and other attack steps. The adversary can examine the depackaged chips using scanning electron microscope (SEM) to reverse engineer layouts for intellectual property (IP) piracy and make illegitimate copies [8]. FPGA reverse engineering aims to recover the readable netlist from the corresponding binary bitstream [9]. Recently, security and privacy concerns have been raised by the powerful focused ion beam (FIB) which is commonly used for semiconductor development and manufacturing [10]. FIBs enable fine-grained milling and deposition on silicon dies, allowing for *probing attacks* by building a bridge between the chip surface and internal metal nets carrying sensitive information [11]. Such attacks are extremely dangerous for the confidentiality of security-critical applications/assets and hard to prevent given the capabilities of adversaries; thereby, calling for effective and efficient solutions to address the dilemma. Although a variety of schemes have been proposed, unfortunately, none of them become a silver bullet to tackle this problem for the following reasons: (i) **Prohibitively high overhead.** Designers can incorporate top-layer *active shields* [12], [13] at the design stage by transferring a specific pattern (e.g., random ciphertext from a lightweight cipher implementation [12]) through the shield nets. A reference pattern is transmitted through a lower layer for comparison to detect any FIB-induced mismatch. The downsides are mainly the prohibitively high area and power consumption due to the additional metal wires and associated circuitry. (ii) **Low reliability.** *Analog sensors* such as the probe attempt detector (PAD) [14] can measure the parameters such as additional capacitance and timing delay introduced by the probe. While imposing smaller hardware overhead they suffers

from low reliability due to analog process variations [10]. In addition, they are themselves vulnerable to being disabled by FIB edits. (iii) **Poor compatibility with modern EDA environment.** The aforementioned solutions cannot be seamlessly integrated into the modern electronic design automation (EDA) flow.

With the aforementioned limitations in our mind, it is imperative to have an automatic EDA framework for quantifying and localizing probing vulnerabilities in a target physical design so that users can adopt optimal countermeasures. [15] presents the first physical design flow to mitigate FIB-based probing at a minimal cost by utilizing a subset of the existing functional nets as active shields in an intelligent and automatic manner. Unfortunately, [15] only focuses on eliminating the front-side threats by enhancing the top-level passivation, failing to take the susceptible back-side into consideration. However, the increasingly prevalent flip-chip packages exposes the back-side to the adversary, allowing the FIB probes to bypass the deliberately planned front-side protection layers and access their destinations through the silicon substrate easily [16]. To address the dilemma, we propose the iPROBE framework to offer a comprehensive protection for both front-side and back-side of the layout against FIB probing attacks. To be specific, *target nets* carrying security-critical assets will be sandwiched by *shield nets* above and below to detect potential intrusions caused by FIB-based probing. Our contributions are summarized as follows.

- We develop a new metric, *shield reliability*, to identify the optimal shield layers protect the target nets/security assets from both front-side and back-side probing attacks, according to the state-of-the-art FIB precision. Compared to *shield security* in [15], the new metric can account for non-ideal conditions of layers, such as routing congestion.
- We extend the definition of *exposed area* that quantifies vulnerable regions inside physical designs from [15] to cover the back-side vulnerabilities as well as addition protection from transistors in the silicon substrate. Since the adversary is expected to avoid shorting the circuitry by touching multiple nodes using FIB, such transistors can further decrease exposed area of target nets from the back-side by as much as 10%.
- We completely automate the vulnerability assessment and countermeasure deployment as a FIB-aware anti-probing flow which can be seamlessly integrated into today's EDA tool-chain, e.g., Synopsys IC Compiler II (ICC2) [17]. To this end, we introduce a parameter, *keepout region*, into the iPROBE framework that can be optimized in order to reduce routing congestion issues that cause certain shield configurations to be unrouteable.
- We evaluate the probing vulnerabilities of cryptographic implementations such as AES and DES [18] at different protection options, i.e., without protection, protection on front-side only, protection on back-side only, and protection on both sides. The experimental results state that our approach can shrink the vulnerable exposed area to zero at most at a cost of as small as 1% timing, 1.5% area, and 10% power overhead.

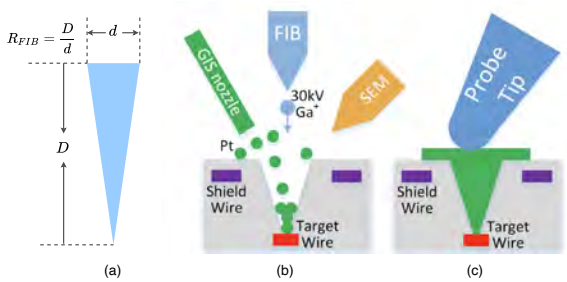


Fig. 2: (a) Definition of the FIB aspect ratio; (b) FIB deposits Platinum in the milling cavity to build conducting path from target wire; (c) The deposited conducting path serves as electrical probe contact and capacitance brought by FIB probe tip [15].

- We design and fabricate a test chip in the 65nm technology node that includes a baseline AES and iPROBE-protected AES. A FIB is used to access and probe the same nets in the two designs from the front-side. Based on SEM images and GDSII overlay, we determine the FIB aspect ratio and show that the iPROBE protected AES has less exposed area than the baseline.

The rest of this paper is organized as follows. Section II describes the background knowledge and threat model. Section III illustrates the anti-probing design flow, provides pseudocode algorithms, and explains for each step of the implementation. Section IV discusses new metrics for front-side and back-shield shield layer identification, iPROBE's keep-out region parameter, and back-side shield exposed area assessment. Sections V and VI discuss our experimental and silicon results, respectively. Finally, the paper is concluded in Section VII.

II. Background

A. FIB Probing Attack

A FIB is advanced equipment that is widely used in failure analysis (FA) because of its ability to use high beam ions (see Figure 2(b)) for site-specific milling and deposition, respectively [10]. Nevertheless, skilled attackers may misuse FIBs for physical attacks to steal confidential information from a physical IC, privilege escalation, and/or fault injection [10].

FIB aspect ratio or R_{FIB} is an important parameter for evaluating FIB probing capabilities. It is defined as the ratio between the depth and diameter of the FIB milling hole [19], as shown in Figure 2(a) and Equation (1), where D and d denotes the depth and diameter, respectively. A larger FIB indicates that the consequent milling hole would be narrower and allow higher probing resolution for the adversary.

$$R_{FIB} = \frac{D}{d} \tag{1}$$

Typical probing attacks [20] against ICs consist of the following steps:

- *Decapsulation:* The chip die needs to be exposed by removing its package. This is an invasive process, requiring sufficient practice for handling noxious chemicals, and

enabling chip die exposure without affecting circuit area or bond wires.

- *Reverse engineering of the chip under attack:* This step aims to figure out the structure and functionality of the chip through reverse engineering. This step can be skipped if the attacker (e.g., foundry) already knows the netlist and layout of the design. Location of asset net identification is the most important step since it determines where an adversary needs to probe.
- *Identification of target wire locations:* After reverse engineering, one-to-one correspondence between the design netlist and layout can determine target net locations.
- *Exposing the target wires to probes:* This step provides access to target wires and builds a conducting path connected to them for probing without damaging critical parts of the design circuitry. A cavity is milled on the chip to expose target wires first, and then, atoms of platinum (Pt) or tungsten (W) gas will be released from gas injection system and deposited in the cavity to build the conducting path.
- *Target information extraction:* In the last step, the chip will be powered and provided with input patterns to transmit signals and excite sensitive operations in the chip. A probe station will be used to probe the conducting paths connected to target wires to extract sensitive information.

Throughout the whole process, the attacker needs to ensure that the chip remains functional. Further, it is possible that an IC might contain one or more countermeasures such as probing sensors or active shields. In such cases, attackers must make sure that exposed asset wires are connected to the conducting path without triggering any probing alarms from the countermeasures. Note that the ability to perform these attacks successfully and without being detected depends on the layout [21]. In a vulnerable layout, critical information will be easy to be accessed by an attacker while it will be difficult in a protected layout.

B. Related Work

Existing countermeasures against probing attacks are reviewed in this section.

Active shields have been a popular countermeasure against front-side probing. They utilize a shield placed on the top-most metal layer or two of the IC, through which, a signal pattern from a pattern generator is transmitted. This pattern will be compared with its copy transmitted from lower metal layers. Once the probing hole cuts one or more shield wires, the signal mismatch will be detected at the comparator and an alarm will be triggered to take actions to protect the assets from being revealed such as erasing or stopping sensitive information generation. It is one of the widely used approaches and is relatively easy to implement. However, large design overhead and the vulnerability to advanced FIBs with high aspect ratio is its biggest issue [22], [15]. Given a FIB with high aspect ratio, smaller holes will be created for probing, and then it would be easy for attackers to utilize the wide space between metal wires in the upper layers to access target nets without leaving cuts on the shielded layer. Besides, an entire metal layer is required to establish the shield and large design overhead, such as area and routing, will be generated inevitably.

In addition, **analog sensors** were once considered as a promising approach. They detect probing attacks by monitoring the variations in parameters of asset-carrying interconnects, such as capacitance and RC delay values. Proper actions will be executed when the parametric variations of the targets go beyond a threshold value. Although it requires less overhead, compared to the active shields approach, its low reliability is one of its biggest problems, considering the effects of process and environmental variations on the assets. Yet another issue is that there is no protection from editing the sensor or its output. *In this paper, iPROBE protects the comparators that monitor its shield nets using shields.*

A **FIB-aware anti-probing physical design** flow proposed in [15] utilizes CAD tools and conventional ASIC design flow to reduce probing attack vulnerability. Essentially, constraints are added to the floor-plan and routing steps to constrain the target nets together and then route an active shield located above the target region using internal layers (rather than top layers only) of the IC. This localized approach introduces much less overhead than active shields while the use of internal shields significantly reduces the area that is exposed to FIB probing. Although [15] provides plentiful insights into the shielding solution against probing intrusions, our work excels in the following three aspects and thus addresses the security concern more comprehensively and practically. (i) *Protection and vulnerability assessment for front- and back-side FIB attacks.* In contrast to [15], our solution focuses on not only front-side intrusions but also protecting devices from back-side FIB attacks. This improvement presents a better fit in modern microelectronic devices given the proliferation of flip-chip packaging models. Also, we extend the vulnerability metric, *exposed area*, to thoroughly quantify defensive contributions from both shielding metal layers and the silicon substrate. (ii) *Optimal shield layer determination and deployment.* Our solution can analytically determine the optimal shield layer by utilizing a new metric, *shield reliability*, to maximize efficiency without sacrificing protective capabilities. Besides, [15] relies more on the theoretical technology parameters without considering practical routing congestion issues since, for example, the pitch size of shield nets may not be as small as the ideal (minimal) values in the context of specific P&R sessions. On the contrary, we can overcome the challenges to yield practical results by introducing two configurable parameters, i.e., the *keep-out region* and the *number of shield gates*. (iii) *Flow automation enhancement.* Our solution has been fully automated covering target/shield nets identification, netlist integration, shield nets deployment, and overhead analysis, making it a plug-and-play utility in modern EDA flow. Meanwhile, part of [15] still needs labor-intensive manual efforts.

C. Threat Model

In this paper, we assume a strong threat model where the adversary has physical access to the target electronic devices. The

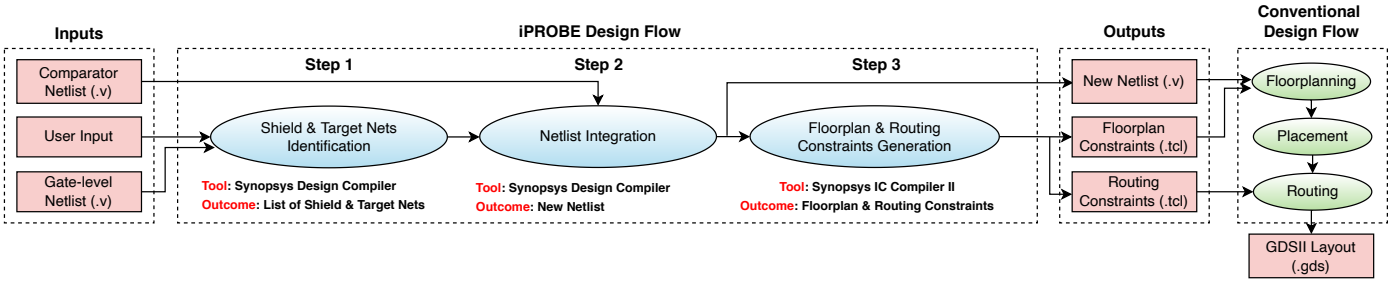


Fig. 3: FIB-aware anti-probing physical design flow, i.e., iPROBE. It takes the user input including user-specified constraints information, such as the ones regarding the shield nets selection, and gate-level design netlist as the inputs to identify the shield and target nets in the design. Then, comparator netlist will be inserted into the original netlist and floorplan and routing constraints are automatically generated to finish the physical design [23].

.

attackers also have knowledge of the entire layout either from an untrusted foundry or through hardware reverse engineering so they can physically locate the target nets. Besides, they are proficient at operating advanced equipment such as FIB and SEM for probing attacks from both the front- and back-sides of the device. Here, we assume that the adversary aims to compromise the confidentiality of security-critical assets. The representative examples of hardware security assets are [2]: private keys, firmware, protected data, and entropy.

The adversary will first mill a cavity to expose the sensitive nets using FIB and then deposit the conductor streaming the internal electrical signals out of the chip. In order to counter the attacks, we consider a conservative detection, i.e., the intrusion can be detected only if one or more shield nets are entirely cut. In addition, a minimum distance between the shield wire and the milling hole is added to avoid the process variation, as the changed parasitic capacitance due to the close distance will probably trigger the alarm by affecting the timing of the shield wires. The detection of the partial cut on the shield nets is out-of-scope of this paper, and left for future work.

Our iPROBE framework will analyze the pre-silicon layout under non-ideal conditions to optimize the floorplan of shield layers to enable maximum protection against threats from *both* front-side and back-side probing.

III. iPROBE Methodology and Implementation

We aim to develop an automated FIB-aware anti-probing physical design flow that incorporates cell placement, floorplanning, and routing into the conventional ASIC design flow for the protection of the assets in a design, e.g., security-critical nets, against *both* front-side and back-side probing attacks.

Similar to [15], our proposed automated FIB-aware anti-probing physical design flow is achieved by routing target nets (i.e., those nets carrying asset signals) between the shield nets (identified from functional nets) routed in an upper layer on top of the target nets and a copy of the shield signals in the lower layers. Once there is a complete cut on a shield net by probing attack, there would be a mismatch on the signal between the signal on that shield net and its corresponding copy, which would then be detected by the comparator and trigger an alarm to take appropriate actions, e.g., reset chip or erase all asset information.

Figure 3 shows the overall workflow of the anti-probing physical design flow. Three new steps are brought into conventional ASIC design flow. First, in order to achieve optimal protection against probing attacks, appropriate shield nets and target nets are automatically identified from the design netlist. User input here denotes information and threshold values that help identify target nets and shield nets. Then, we will insert comparators into the gate-level netlist of the original design that are used to detect the mismatch on the signal transmitted on shield nets on upper layer and its lower copy. Finally, internal shield is built with floor-planning and routing constraints being added to the design to provide protection against probing attacks. Regarding the protection aimed for both front-side and back-side, *two sets of shield nets* will be added to the design, with one being routed on an *upper layer* to prevent front-side probing attack and the other one on a *lower layer* to protect against back-side probing attack. Target assets nets will be routed on the layers *between these two sets of shield nets*.

A. Target Nets and Shield Nets Identification

The first stage, target nets and shield nets identification, aims to recognize the target nets from the entire netlist and identify appropriate nets that will used as shield nets. It takes the inputs of the comparator netlist, user input, and gate-level netlist. A comparator is used to compare the signal transmitted from the shield nets and their copy on the upper/lower layer, the length of which varies with the number the shield nets inserted. User input includes requirements for identifying shield and target nets and the layers where they will be distributed. In short, we adopt the technique from [15] where target nets are identified using the *target score* metric, which is used to describe the likelihood of a net in an IC to be considered as the target in a probing attack, and the shield nets are recognized using a set of following metrics:

- **Toggle frequency:** the shield nets with a relatively high toggling rate may not be easy to be replaced by the attackers.
- **Switching probability:** The probability of the shield nets to be 1 or 0 should be balanced so that it would be

difficult for the attackers to predict the signals on the shield nets.

- **Controllability:** The nets with relatively high SCOAP controllability value [1] will make it less possible for the attackers to have control over the shield nets.
- **Delay slack:** The internal nets that are used as the shield nets should not lie on critical paths as they should not impact the critical path delay and the design's performance.

Note that, in order to maximize the protection on target nets and minimize the vulnerabilities and impacts from shield nets, a threshold value of each aforementioned corresponding metric should be determined. The final set of shield candidate nets should be the intersection of the five net collections for each metric requirement which satisfy the threshold values.

B. Netlist Integration

The netlist integration step mainly consists of building up the comparator netlist and integrating the comparator, shield, and original design netlist together into one new netlist that will be placed and routed later.

The comparator gates are XNOR based that are designed to output a logic 1 when there are no attacks, and optimized by synthesis tool to result in less area overhead. In detail, two inputs to the comparator are from the same source nets, one of which is connected to the shield net from upper layer, and the other one connected to the copy form lower layer. Note that, the comparator is also included to be part of potential probing target assets to be protected against attacks, the size of which is determined by the number of shield nets required to be added to the design netlist.

C. Floor Planning and Routing Constraints Generation

In the conventional ASIC design flow, target nets and the blocks containing them are distributed randomly in the design layout. This form of placement is neither easy nor efficient to provide protection to target nets. Therefore, we hope to constrain target nets and gates in a regularly shaped region, e.g., rectangle. In addition, the comparator nets should be constrained and protected as target nets because the results of the comparator may also be tampered by the probing attack. Therefore, the comparator gates are constrained in a rectangular shaped floor-plan group adjacent to the target block. The placement of target and comparator nets and gates is shown in Figure 4.

To implement front-side and back-side protection, shield gates are divided into four groups: shield nets driver and load groups on lower metal layer, and shield nets driver and load groups on upper metal layer. The driver and load gates of the shield nets are constrained at opposite ends of the target and comparator gates. Therefore, routing of shield nets will cross the target and comparator gates and nets area and vertical protection can be provided with shield nets being routed above *and* below the target and comparator gates and nets area. A cross-sectional example where M8 and M2 are used for front- and back-side shields is shown in Figure 5.

Algorithm 1 and 2 demonstrates iPROBE physical design flow and floorplan and routing constraints generation. The red parts denote the three main steps: loading design and placement and routing. Brown and teal parts represent the similarities and differences compared to [15]'s flow. In generating placement constraints (lines 2 to line 9), as discussed before, shield, target, and comparator gates will be constrained in a regularly shaped region, which follows a similar procedure as [15]. It is finished with the command *create_rp_group* to create a placement group and *add_to_group* to add protected gates or shield gates to their corresponding group.

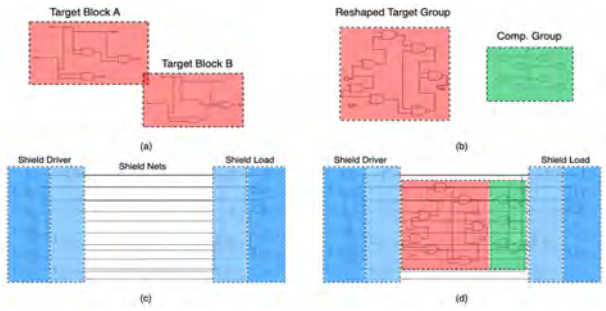


Fig. 4: Placement constraints. (a) Irregular location of target nets; (b) Reshape the target nets to fit one regular rectangle block (red) and comparator block (green); (c) Shield gates (blue) are divided into lower shield nets driver/load block (light blue) and upper shield nets driver/load block (dark blue); (d) Shield gates are placed surrounding the target and comparator blocks which will be covered by shield nets.

In the process of generating routing constraints, new parameters, *size of keepout region* and *number of shield gates* are introduced to reduce the exposed area while eliminating the routing congestion issue. In detail, if routing congestion issue appears, we will try to either reserve more space for shield nets routing or decrease the number of shield nets to be routed. These new parameters and the rest of Algorithm 2 will be discussed in Section IV-A.

With the identification of shield nets, the next problem to be settled is the optimal routing layer of the shield nets that need to be determined for the best protection. A milling scenario is assumed using FIB technology as shown in Figure 6. Colored bars are used to represent metal wires on different routing layers in the design layout. More about this will be discussed in the next section.

IV. iPROBE Metrics and Assessment Flow

In past work, the quality of shields was determined by two metrics: shield security (pre-layout) and exposed area (post-layout). Below, we briefly discuss their limitations followed by our improvements for the remainder of this section.

The *shield security* metric, $R_{FIB,max}$, was proposed in [15], which is defined as the FIB with maximum aspect ratio that a metal layer can protect against.

$$R_{FIB,\max} = \frac{D_{s2t}}{P_s - W_s - 2M_{pv} - 2S_{s2h}} \tag{2}$$

where all the notations are denoted as shown in 6. Given the library technology, Algorithm 1 will first calculate the

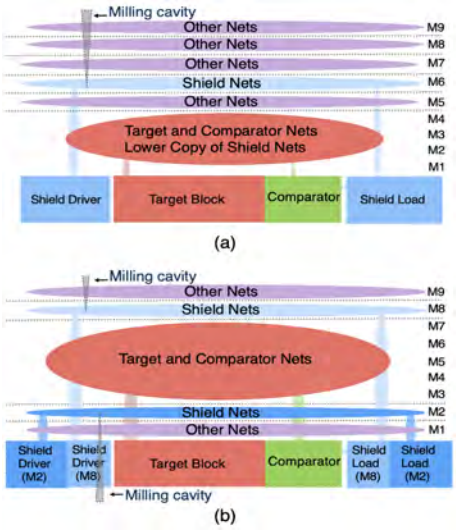


Fig. 5: The cross-sectional view of target and shield nets layer distributions. (a) the baseline shield structure against frontside-only probing threats as presented in [15] where shield nets residing in M6 provide a single-layer protection. (b) our shield structure provisioning a dual-layer protection against both frontside (shield nets in M8) and backside (shield nets in M2) probing intrusions.

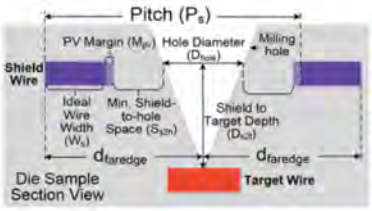


Fig. 6: Calculations for shield security and $d_{faredge}$ [15].

$R_{FIB,max}$ of each metal layer and then determine the optimal shield nets layer based on the user-defined R_{FIB}, i.e., the metal layer with the $R_{FIB,max}$ that is larger than the user-defined R_{FIB} for each protected layer will be chosen as the optimal layer to route the shield nets on.

The primary use of this metric was to evaluate how much protection that a shield on a certain layer can provide to assets nets. A higher value of the shield security metric would bring better protection. In theory, shield security is only decided by technology-dependent parameters, such as the width and thickness of wires. However, we've found that other parameters such as the design itself, number of shield nets, number of target nets, and selected shield layer impact its utility. Specifically, a major side effect of the introduced shield nets, comparators, etc. is the consequent *routing congestion* in the same layer. In practice, we've found that this can result in routing failures. To address such issues, a new parameter and metric are needed. To this end, we introduce the *keepout region* parameter in Section IV-A and the *shield reliability* metric in Section IV-B. Together, these allow pre-layout shield security to be estimated under non-ideal conditions and can be used to compare different shield layers with constraints.

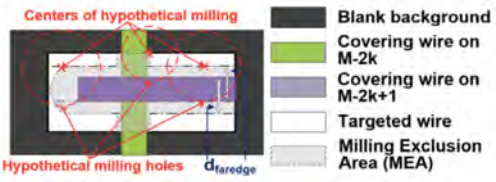


Fig. 7: Front-side exposed area (EA) calculation [21].

The *exposed area* metric was originally proposed to assess a design's post-layout vulnerability to probing attacks from the front-side [21]. Figure 7 shows the determination of exposed area (EA) for any given target wire and covering wires which are capable of providing protection to the milling exclusion area (MEA). MEA is the complementary part of the protected area. In Figure 7, the target wire is represented as white region, and the green and purple regions are the covering wires at upper layers. The shaded region is the MEA, indicating that if the FIB milling center falls in this area, a complete cut will happen to the covering wires. A larger exposed area represents a higher level of vulnerability to FIB probing attacks.

In this paper, we introduce back-side shields whose protection can only be partially quantified by the exposed area calculation from[21]. Hence, in Section IV-C, we augment the approach to more accurately calculate exposed area from the back-side.

A. Keepout Region Parameter

To mitigate the routing congestion caused by the iPROBE shield, we propose to reserve a *keepout* region between the target gates and shield gates. Specifically, we place the target gates whose connectives might carry sensitive information in the center region of the layout. The shield gates are placed in the regions which are isolated from the target gates at a certain distance. The space between them refers to the keepout region being filled with the non-target logic gates in the original circuitry. We can control the size of keepout regions through floorplanning constraints as shown in **Algorithm 2**, i.e., varying the *height* setting in the command *add_to_rp_group* as presented in lines 15 and 18 where the parameter **ko** here refers to the number of cell rows belonging to the keepout region. Empirically, a larger keepout region implies more slack for nets routing at the shield layers and thus less congestion.

In detail, initialization of **ko** is incremented from one if current **ko** incurs routing congestion issue and does not exceed chip area. Otherwise, iPROBE will remove one item from the current set of shield gates, **SG***.

B. Shield Reliability Metric

As described above, the value of shield security may not be as high as expected in practice, e.g., due to the limited routing space and resources available in the IC, which leads to routing failures. The *shield reliability* metric is defined for a given shield layer as the maximum shield security value for which the design is routable. To compute shield reliability, the keepout region size (s) and the number of shield nets (n) are

Algorithm 1: iPROBE Physical Design Flow

Input: **IN** - Integrated netlist
Input: **pc.tcl** - Placement constraints file
Input: **rc.tcl** - Routing constraints file
Input: $\mathbf{P}_{lib}$ - Library technology dependent parameters
Input: $R_{FIB,user}$ - User-defined FIB aspect ratio
Output: **GDSII** - GDSII Layout

```
1  R_FIB,max ← Determine_FIB(P_lib)
2  Optimal shield layer ← Determine_Shield(R_FIB,max, R_FIB,user)
3  read_verilog IN % load design netlist
4  source pc.tcl % input placement constraints file
5  place_opt
5  source rc.tcl % input routing constraints file
6  route_opt
```

Algorithm 2: Floorplan and Routing Constraints Generation

Input: **IN** - Integrated netlist
Input: **TN, TG** - List of target nets and gates
Input: **SG1, SG2, SN** - List of shield driver and load gates and shield nets
Input: **CN, CG** - List of comparator nets and gates
Output: **ko*** - Size of keepout region
Output: **SG*** - Number of shield gates

```
1  read_verilog IN % load design netlist
2  create_rp_group target_rp % create target gates placement group
3  add_to_rp_group target_rp TG % add target gates to placement group
4  create_rp_group shield_rp1 % create shield driver gates placement group
5  add_to_rp_group shield_rp1 SG1 % add shield driver gates to placement group
6  create_rp_group shield_rp2 % create shield load gates placement group
7  add_to_rp_group shield_rp2 SG2 % add shield load gates to placement group
8  create_rp_group comp_rp % create comparator gates placement group
9  add_to_rp_group comp_rp CG % add comparator gates to placement group
10 SG* ← SG
11 while (1) do
12   for ko ∈ KO do
13     create_rp_group -name rp -columns 1 -rows 6
14     add_to_rp_group rp -rp_group shield_rp1 -column 0 -row 0
15     add_to_rp_group rp -blockage gap -column 0 -row 1 -height ko
16     add_to_rp_group rp -rp_group target_rp -column 0 -row 2
17     add_to_rp_group rp -rp_group comp_rp -column 0 -row 3
18     add_to_rp_group rp -blockage gap2 -column 0 -row 4 -height ko
19     add_to_rp_group rp -rp_group shield_rp2 -column 0 -row 5
20     place_opt
21     create_shape -boundary llx lly urx ury
22     set_routing_rule -min_routing_layer -max_routing_layer
23     route_opt
24     if Routing Congestion Occurs then
25       if exceeds chip area then
26         SG* ← Remove one gate from SG*
27       else
28         increase ko
29       end
30     else
31       ko* ← ko
32       return ko*, SG*
33     end
34   end
35 end
```

swept while constraining by the design layout, technology-dependent parameters, etc, as shown in Equation (3).

$$\textit{Shield Reliability} = \max_{s,n}(R_{FIB,max}) \tag{3}$$

where $R_{FIB,max}$ is defined in equation 2.

The higher the shield reliability is, the more protection the shield layer can provide. Given a technology library, the calculation of shield reliability for each metal layer is demonstrated in **Algorithm 2**. In detail, it entails the determination of the size of the keepout region and number of shield gates that leads to the maximum shield security value. It starts with the minimum pitch size and determines whether there is routing congestion issue or not after the physical design flow (line 24), and the size of the keepout region will be adjusted if they do exist, as larger keepout region will leave more space for the routing to help mitigate the congestion problem. If the width/height of the protected region exceeds the chip area after the modification of the size of the keepout region, we will decrease the number of the shield gates (line 26). The loop will continue till no routing congestion is reported under current size of the keepout region and number of the shield nets. After determining the shield reliability of each metal layer, the layer with the largest metric value will be picked to route the shield nets. Below, we compare the two metrics for front-side and back-side cases assuming the technology parameters in the SAED32nm library.

TABLE I: Front-side shield security in SAED32nm library.

Shield Security	Shield Layer						
Target Layer	**8**	**7**	**6**	**5**	**4**	**3**	**2**
7	0.64	N/A					
6	1.28	0.64	N/A				
5	1.91	1.28	1.81	N/A			
4	2.55	1.91	3.61	1.81	N/A		
3	3.19	2.55	5.42	3.61	4.41	N/A	
2	3.83	3.19	7.23	5.42	8.82	4.41	N/A
1	4.47	3.83	9.04	7.23	13.24	8.82	INF

TABLE II: Front-side shield reliability in SAED32nm library.

Shield Reliability	Shield Layer						
Target Layer	**8**	**7**	**6**	**5**	**4**	**3**	**2**
7	0.64	N/A					
6	1.28	0.64	N/A				
5	1.91	1.11	1.81	N/A			
4	2.55	1.86	3.61	1.12	N/A		
3	3.19	2.10	5.42	2.53	4.27	N/A	
2	3.83	3.04	7.23	3.17	7.17	2.79	N/A
1	4.47	3.64	9.04	5.47	9.40	5.42	28.59

Front-side shield security vs. reliability. The front-side shield security and shield reliability for different choices of shield layer are shown in Tables I and II, respectively. It can be observed from the comparison between the shield security and shield reliability that shield security has a higher value than the shield reliability for each layer. It can be seen most clearly for layer M2, of which the shield security is INF, while the shield reliability is not. For the shield security, "INF" denotes that no FIB can bypass the shield on M2 layer, because the required shield wire to FIB hole space is larger than the pitch space between the adjacent metal wires on M2 layer. However, in reality, there could not be as many shield nets in layer M2 as one expects because of the routing congestion issue. Then, we have to either increase the size of keepout region to reserve more space for the shield nets routing or reduce the number of shield nets to reduce the shield nets density – *this loss in protection is reflected in the shield reliability.* There are other important differences when identifying the second-best shield layer as well. In Table I, the next best layer is probably M4 while it is probably M6 in Table II – note that the best one depends on how many targets appear on each layer below the shield.

Back-side shield security vs. reliability. The back-side shield security and shield reliability for different choices of shield layer are shown in Tables III and IV, respectively. Similar to the front-side case, we can see from the comparison that shield reliability is no larger than shield security due to the increase of keepout region or reduction of number of shield nets in order to avoid the routing congestion issue. For both tables, M2 is the best shield layer; however, its protection is not INF for all target layers in the case of shield reliability.

Front-side shield reliability vs. back-side shield reliability. We can see that the back-side shield is much better than the front-side shield, especially for the shield on M2 layer. This

TABLE III: Back-side shield security in SAED32nm library.

Shield Security	Shield Layer					
Target Layer	**7**	**6**	**5**	**4**	**3**	**2**
8	0.64	3.61	5.42	17.65	22.06	INF
7	N/A	1.81	3.61	13.24	17.65	INF
6	N/A		1.81	8.82	13.24	INF
5	N/A			4.41	8.82	INF
4	N/A				4.41	INF
3	N/A					INF

TABLE IV: Back-side shield reliability in SAED32nm library.

Shield Reliability	Shield Layer					
Target Layer	**7**	**6**	**5**	**4**	**3**	**2**
8	0.64	3.61	5.17	14.32	13.92	19.86
7	N/A	1.81	3.27	11.17	9.25	14.57
6	N/A		1.44	7.52	5.96	11.52
5	N/A			3.76	4.16	9.01
4	N/A				2.26	6.79
3	N/A					4.47

is mainly due to the smaller pitch size of lower layers, thus enabling more shield nets to be distributed and more protection to be provided.

Front-side shield reliability vs. both front-side and back-side shield reliability. Provisioning shielding protection on both front-side and back-side simultaneously is slightly different from merely covering either front-side or back-side individually because of the affected consequent placement and routing (congestion) of the entire physical layout. In order to fairly evaluate the effectiveness of shields on both sides, we tabulate the shield reliability for front-side and back-side attacks in Tables V and VI, respectively, for different combinations of shield layers. One can observe that the reliability metric targeting the metal layers from M3 to M5 is very close under both M2+M6 and M2+M8 shield structures. This is interesting because none of the individual tables (Tables I –IV) seemed to indicate strong protection from M8.

C. Back-side Exposed Area (EA) Assessment

When it comes to the back-side protection, electrical probing attacks from back-side of the IC need to go through the transistor level to reach metal lines of assets. Destroying some layers of a transistor will render the transistor useless and may facilitate probing attack detection. Most notably, the gate terminal[1] of a transistor controls its region of operation and subsequent current.

Figure 8 shows the schematic, top-view and cross sectional-view of a NAND2 gate. Suppose there is an asset region located above the NAND2 in the middle of the red circle shows in the top-view. Then, probing from the back-side of the IC will either destroy (through milling) or make contact (through deposition) to polysilicon of the cell, as shown in the cross-section view. In this scenario, since the components of a cell, e.g., n-well, p-well, poly-Si, etc., get in the way of probing from the back-side of ICs, attackers would try to avoid touching target and comparator cells, because the probing attack results might be affected if they get sabotaged. *Thus, the transistors in their gates will act as yet another set of back-side shields. Therefore, the contacts and poly-silicon gates of a cell should be taken into consideration when calculating EA for back-side protection.*

[1]Transistor gates may be made out of polysilicon or metal depending on the technology node. We refer to it as polysilicon for simplicity.

TABLE V: Front-side shield reliability in SAED32nm library when both front-side and back-side protection is provided.

Backside shield layer	Target Layer	Frontside shield layer			
		5	**6**	**7**	**8**
2	**3**	2.21	2.97	2.86	2.76
	4	0.97	1.65	1.91	2.56
	5	N/A	1.1	1.46	1.65
	6	N/A	N/A	1.12	1.09
	7	N/A	N/A	N/A	0.5
3	**4**	0.86	2.79	1.65	1.87
	5	N/A	1.26	1.35	1.6
	6	N/A	N/A	0.27	1.07
	7	N/A	N/A	N/A	0.3

TABLE VI: Back-side shield reliability in SAED32nm library when both front-side and back-side protection is provided.

Backside shield layer	Target Layer	Frontside shield layer			
		5	**6**	**7**	**8**
2	**3**	1.27	3.12	3.67	3.89
	4	2.05	3.34	3.9	4.32
	5	N/A	4.1	4.8	5.6
	6	N/A	N/A	6.1	8.7
	7	N/A	N/A	N/A	10.7
3	**4**	0.96	2.76	3.1	3.1
	5	N/A	3.14	3.68	4.49
	6	N/A	N/A	5.7	6.1
	7	N/A	N/A	N/A	9.7

To consider this in back-side exposed area calculation, components of polysilcion gates and contacts in a transistor are taken as an example since other units such as p-well and n-well can provide the protection in a similar manner. In detail, the transistor level is considered as an active layer and location of poly-Si and contacts of logical cells will be identified first based on the types of the cell. Then, the location of the polysilicon gate and contacts in all the cells are determined and summarized. They will provide protection if the target nets above them reside in their projecting area on the layer where target nets are located. In this way, the milling exclusion area (and exposed area) due to these layers can be calculated similarly to the metal interconnects as previously shown in Figure 7.

D. iPROBE Layout Assessment Flow

Figure 9 shows the overall iPROBE exposed area assessment flow. It takes as the inputs of design GDSII layout, list of target nets, given FIB aspect ratio, and technology-dependent parameters. As for FIB aspect ratio, state-of-art FIB systems can reach up to approximately 10 [11], and the exposed area will be assessed across FIB aspect ratio from 1 to 10. iPROBE assessment flow is computationally heavy considering large number of nets in an IC.

Then we use MATLAB to take over the computation work for efficient calculation considering the large amount of the target nets and shield nets on different metal layers. MATLAB accepts the inputs from IC Compiler II including the location of target nets and their constituent net wires, and their covering shield nets on each metal layer, and calculates the MEA on each target net. Then, the complementary part of the MEA

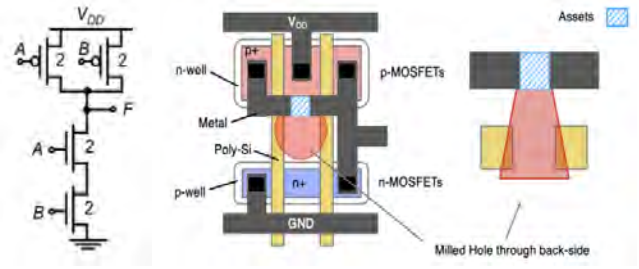


Fig. 8: Transistor-level extension for back-side exposed area assessment. (left to right) NAND2 gate schematic, top view, and cross-section view of the layout.

is the exposed area. Note that compared to [15], our flow calculates EA for both front-side and back-side as discussed in the previous subsection.

V. iPROBE Pre-silicon Evaluation

In this section, we will evaluate the iPROBE design flow in terms of pre-silicon design time, silicon overhead, and security from front-side and back-side probing. We also compare our assessment methodology with and without the inclusion of transistor shielding.

A. Experimental Setup

We select four benchmark cryptographic implementations for our experiments, i.e., AES, DES, Simon, and Present [18], [24], [25]. These register-transfer level (RTL) designs are compiled to physical layouts using Synopsys Design Compiler [26] and ICC2 [17]. The technology cells are from the SAED 32nm library [27]. We consider the hardcoded cryptographic *keys* as our assets for all designs. Also, we set a same threshold of the *target score* metric as the one of the front-side FIB-aware work [15]. Specifically, 0.125 for the threshold value of the target score, for a fair comparison, and as for the shield nets, they will be the intersection of the aforementioned five net collections that satisfy the threshold values for each shield requirement, which would vary for different designs.

To provide more insights into the experimental layout, we take the AES design as an example while the other three benchmarks follow a similar pattern. As one can observe in Figure 4, we separate the cells into different groups, i.e., target gates, comparator gates, M2 shield nets drive gates, and M8 shield nets drive gates, for clear illustrations. In detail, target gates along with the symmetric key storage cells are grouped together and reshaped into the red rectangular since they serve as the drivers of target nets carrying sensitive data. The shield nets are placed at both M2 and M8 metal layers to wrap the middle layers up against the potential intrusions while their drivers (dark blue and light blue cells) are placed at the peripheral. In addition, a set of comparators are inserted into the post-synthesis netlist to monitor whether the shield nets remain intact or not. The corresponding units (in green) locate in between the target gates and shield net drivers.

Figure 10 shows that the shield nets at M2 and M8 reserve a very limited space of probing attacks. Besides, the smaller pitch size of M2 layer allows more shield nets to achieve a higher density than M8. Further, Figure 11 presents the distributions of shield and target nets across all 8 layers which are consistent with our intention, i.e., the target nets are concentrated on middle layers (more than 80%) ranging from M3 to M7 while shield nets mainly reside in M2 and M8 for preventing the external probes from reaching the internal assets.

TABLE VII: Design types for quantitative comparison.

No.	Shield Type	Description
1	Original Design (No Shield)	Conventional physical design
2	Front-side Single Layer Shield	Single layer shield on M8
3	Front-side Single Layer Shield	Single layer shield on M6
4	Back-side Single Layer Shield	Single layer shield on M2
5	Back-side and Front-side Shield	Shield on M2 and M8

One may have concerns about the security of target nets within lower layers (e.g., M1 and M2) which seems not to be protected well with shield nets. However, as illustrated in 11, less than 20% of sensitive signals reside in the M1 and M2 layers whereas the remaining 80% (from M3 to M7) is covered by shield nets. Besides, thanks to solutions such as [16], [28], [29], it is feasible to effectively utilize silicon substrate to thwart backside probing attacks from compromising the security of target signals within low layers. Note that such backside probing attacks on target nets are not very trivial and affordable, requiring high-end equipment, e.g., high aspect ratio FIB, to avoid rendering transistors useless."

Table VII denotes four scenarios of protected implementations and the exposed area calculation. Both front-side and back-side exposed area assessment will be performed on all four designs. It is worth noting that, for the back-side exposed area assessment, it will include both shielding ability from metal layer and active layer, i.e., poly-Si and contacts, demonstrating the shielding abilities of the transistor layer unless otherwise specified.

B. Pre-silicon EDA Time

As the design implementations are written in RTL, our entire pre-silicon analysis involves the following four stages.

- **Stage 1: Synthesis and Netlist Processing** includes the synthesis process as well as shield gate and comparator insertion.
- **Stage 2: Place and Route** (self-explanatory).
- **Stage 3: Exposed Area Calculation** computes the exposed area as discussed in Section IV-D.
- **Stage 4: Design Overhead Data Collection** where power, area, timing, and routing overhead are measured.

We tabulate the time for each step as well as design overhead results in Table VIII. For each design and shield, the entire process takes less than 40 minutes.

C. Exposed Area

As discussed in Section IV-C, *exposed area* metric is used to evaluate the proposed internal shielding approach. Figure 12 illustrates the normalized front-side and back-side exposed area metrics corresponding to the FIB aspect ratio ranging

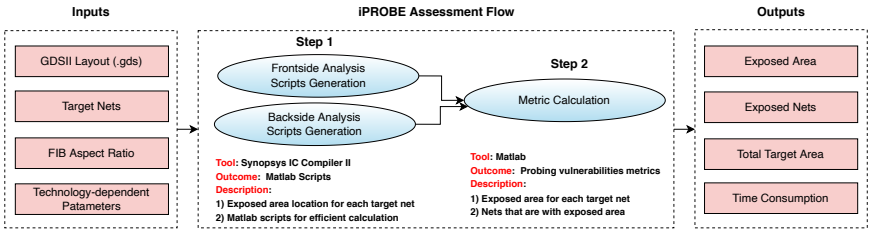


Fig. 9: iPROBE's exposed area evaluation flow. It takes the design layout, FIB and technology-dependent parameters as the inputs, and generates design layout vulnerabilities assessment results including exposed nets in the design and their exposed area.

TABLE VIII: Design overhead in different iPROBE structures.

Design	Target Nets	Target Gates	iPROBE Structure	Total Gates	Overhead				Time Consumption (min)			
					Timing	Power	Area	Routing	Stage 1	Stage 2	Stage 3	Stage 4
AES	256	384	Frontside	10547	0.44%	3.20%	0.74%	11.60%	4.9	7.0	14.9	2.3
			Backside	10551	0.69%	5.79%	0.90%	11.72%	5.2	7.4	15.0	2.7
			Both	10816	0.75%	7.28%	1.16%	13.17%	5.3	8.2	16.0	2.9
DES	496	880	Frontside	18577	0.66%	5.99%	1.24%	13.23%	6.2	7.2	19.1	2.9
			Backside	18590	0.91%	6.40%	1.39%	15.20%	6.3	8.0	19.1	3.2
			Both	19262	0.96%	7.89%	1.56%	19.25%	6.3	9.0	20.0	3.4
Simon	112	496	Frontside	7979	0.27%	1.99%	0.65%	10.61%	3.8	6.5	11.7	1.9
			Backside	8040	0.34%	2.20%	0.76%	11.40%	3.9	6.6	11.8	1.9
			Both	8176	0.36%	2.56%	0.80%	11.98%	3.9	7.0	12.0	1.9
Present	80	160	Frontside	6522	0.19%	1.50%	0.29%	8.02%	3.6	5.6	10.2	1.8
			Backside	6777	0.22%	1.92%	0.40%	8.66%	3.6	5.8	10.7	1.8
			Both	7024	0.22%	2.34%	0.61%	9.27%	3.6	6.0	11.1	1.8

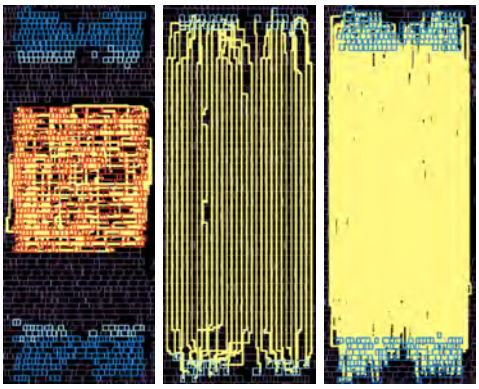


Fig. 10: AES shield gates (light blue at M8 and dark blue at M2), target gates (red), and highlighted nets (yellow).

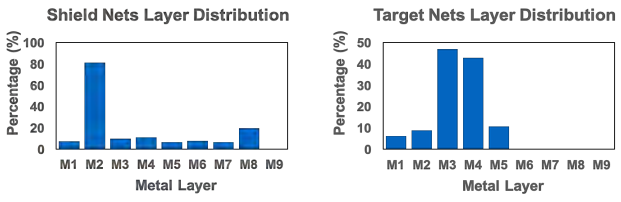


Fig. 11: Shield nets and target nets layer distribution for AES module.

from 1 to 10. We assess the FIB vulnerabilities under three scenarios, i.e., no shield nets, shield nets at M2 and M6, as well as shield nets at M2 and M8. As depicted in Fig. 12, when the FIB aspect ratio is small, the front-side exposed area can be reduced to as low as 0 in the protected implementation thanks

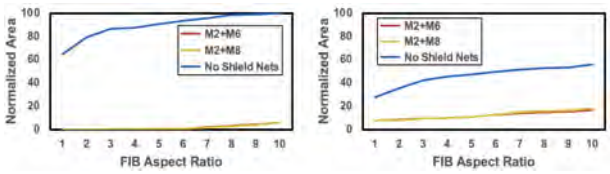


Fig. 12: Frontside (left) and backside (right) exposed area results for AES module.

to the shield nets at either M6 or M8. As the FIB aspect ratio increases, the probe is getting *sharper* being more capable of bypassing the alarm shield circuitry [15]. For that reason, the exposed area indicating the FIB vulnerabilities becomes larger for both front-side and back-side orientations, which is also consistent with our results in Fig. 12. It is worth noting that the results of M2+M6 and M2+M8 are very close whereas M6 has a smaller spacing and wire width specification compared to M8, thus owning a better theoretical resilience against FIB probing. Despite this, on the other hand, the number of signal and shield nets at M6 needs to be limited strictly to avoid routing congestion, thereby consequently diminishing its protective capabilities to some degree. This is consistent with what was shown in Table V.

Figures 13 and 14 shows the normalized back-side exposed area for all types of designs in Table VII for all four designs. Similarly, results are calculated across FIB aspect ratio from 1 to 10. It can be seen that, in Figure 13, when there is front-side shield only, i.e. at M6 layer, there is large exposed area left at the back-side of the IC and thus barely any protection is provided. While in Figure 14, by comparing the exposed area results between M2 only and M2 with poly-Si and contacts, we can see that transistor's level does provide protection from the

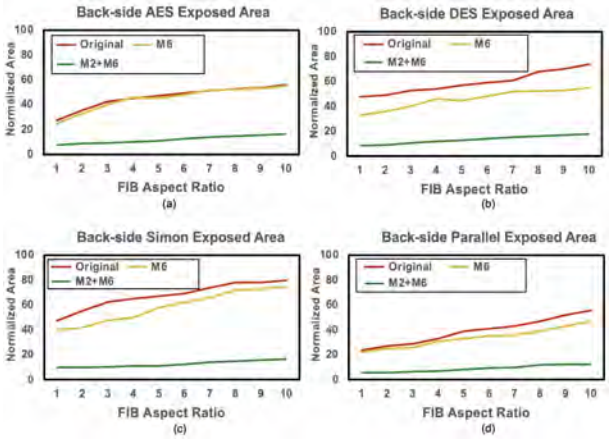


Fig. 13: The exposed area of physical layouts with and without back-side protection.

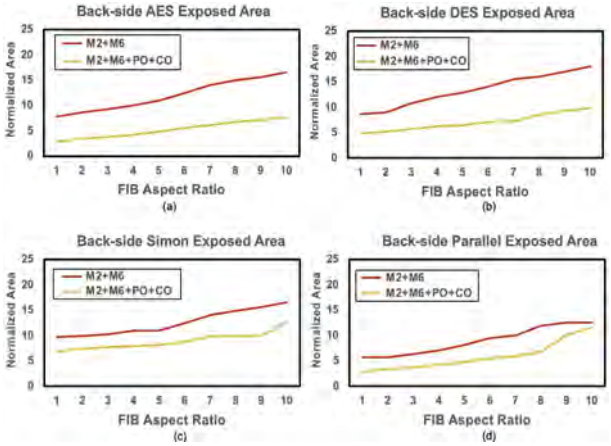


Fig. 14: The impacts of transistor-layer protection on the exposed area.

back-side and transistor's level can bring up to 10% reduction in the exposed area.

D. Overhead

We present the overhead and time consumption in Table VIII. Overhead includes timing, power, area and routing of all these designs with front-side and back-side shield compared to the original designs without any constraints. As we can see from the table, the timing overhead of these designs is less than 1%, the power overhead is less than 10%, and the area overhead is less than 2%. It is worth noting that by comparing the overhead of front-side and back-side for each design, overhead will only increase at most 2% for the back-side protection while significant protection can be provided to the back-side of the IC, which helps reduce 60% in the exposed area on the target nets, as discussed in Section V-C.

VI. Silicon Evaluation with 65nm Test Chip

A. Test Chip Overview

To investigate and evaluate the effectiveness of iPROBE, we designed and fabricated a test chip with the TSMC 65nm CLN65g+ process [30].

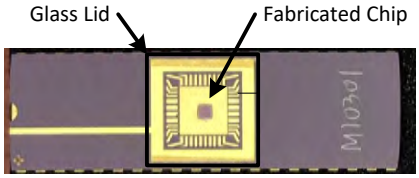


Fig. 15: Fabricated die and package. A removable glass lid is used for easier FIB and probe access.

1) Top-level Integration, Synthesis, and Verification: The top-level design consists of several sub-modules - (i) AES baseline, (ii) AES shielded by iPROBE, (iii) AES PRNG, (iv) PRNG 128, and (v) s38417 from ISCAS'89 benchmark circuits [31]. For the evaluation in this article, we focused our experiments on the first 2 sub-modules. The baseline AES HDL code was collected from opencores.org [32]. All the sub-modules are integrated into a TCL script and synthesized into a gate-level netlist. Synopsys Design Compiler tool was used to perform synthesis. A constraint specification file was used during the synthesis process to define the gate-level design's clock frequency, port delays, area, and power overhead. After the synthesis step, a gate-level design constraint file (SDC) was exported and later used in the place-and-route design flow. The gate-level netlist and associated testbench were simulated in Synopsys VCS MAX flow.

The synthesized and verified gate-level netlist along with the associated timing constraint file was the input design to the place-and-route step. Place-and-route was performed in Synopsys IC Compiler and physical verification of design-rule-check (DRC) and layout-versus-schematic (LVS) were performed in Mentor/Siemens EDA Calibre.

2) Chip Packaging: The size of the chip is $1.672 \times 1.672\ mm^2$. Figure 15 shows the fabricated and packaged chip in dual inline packaging (DIP). A DIP is a chip encased in hard plastic with pins running along the outside. To facilitate probing and evaluate the efficiency of iPROBE, the top of this chip is covered with a removable glass lid (marked in Figure 15) so that the die can be more easily accessed. In other words, we did not have to perform time-consuming sample preparation steps such as depackaging.

B. FIB Editing Approach

FIB circuit edit was performed using a finely focused gallium (Ga+) ion beam with nanoscale resolution using our TESCAN LYRA3 system. In order to perform circuit edits, the FIB is coupled with X-positioner and K-layout that make it possible to locate the area of interest. The designer's graphic display system (GDSII) files are used to navigate to a precise area. This provides a method to find sub-surface features and ensure that the correct edits are performed [33]. Furthermore, endpoint detection tool has been used to know when selected layers of interest have been successfully reached.

1) GDSII Layout Alignment: The online tool K-Layout is used to understand the layout of the chip and to determine and understand the different metal layers in the sample. By using K-Layout and the X-positioner tool, the different metal

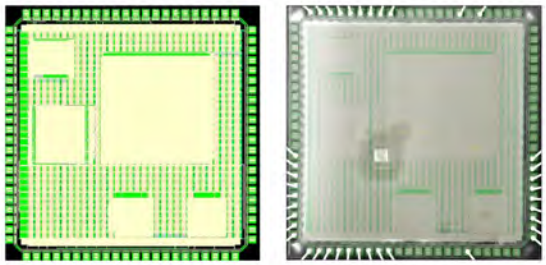


Fig. 16: Overlapped image of complete GDSII layout with the SEM image

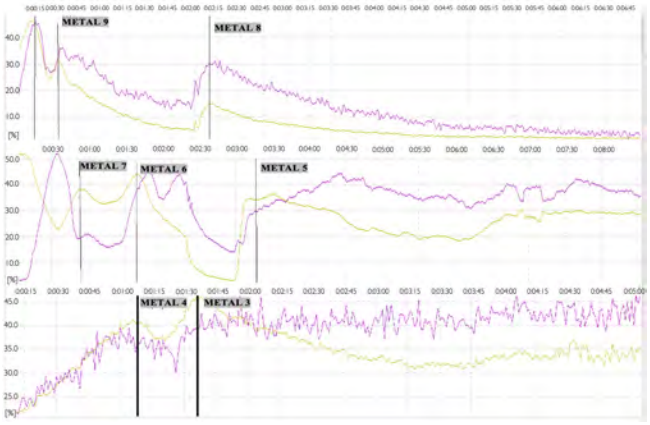
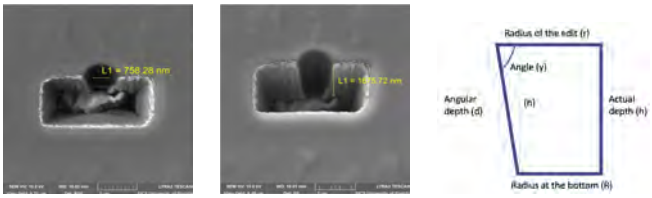


Fig. 17: EPD curves for monitoring and controlling the delayering process for a single milled hole. Curves represents signal from the all currently exposed area in percents on Y axis as time [s] on X-axis. Gold line represents average values while purple line shows standard deviation.

layers were calibrated and aligned very precisely with the SEM Image of the sample to determine the accurate edit locations on the test chip. In addition, an area of 100×100 μm^2 was also delayered for the proper alignment of the metal layers layouts with the SEM Images. Figure 16 shows an accurately aligned overlapped image of the layout with the SEM image of the chip.

2) FIB Circuit Edits on AES modules: To validate the effectiveness of our solution on the test chip, the FIB edits were performed on randomly selected 6 target nets located in metal layer 3 from both non-protected and protected AES implementations. Specifically, we chose metal wires with a small, exposed area in the protected layout and obtained their counterparts from the non-protected layout. Note that, due to the deployment of shield nets, the protected layout placement and routing are changed geometrically from the original one so we need to separately locate the targets to get the coordinates of these wire pairs according to the layout files. A finely focused Ga+ ion beam with a current of 300 pA was used to drill holes with a size of approximately 7504 nm diameter at specified coordinates.

3) End-Point Detection: End-point detection (EPD) is a useful technique to determine the end of FIB milling according



(a) (b) (c)

Fig. 18: (a) (b) SEM images of the circuit edits to metal 3 (M3) layer. Additional (square) cuts were made to measure the depth and angle of the conic edits. (c) Cross-section illustration of the editing hole for calculating FIB aspect ratio and edit radius at each metal layer.

to a detector signal from a milled area. The typical use is for milling multi-layered materials or ICs for circuit edit. It displays the charge absorbed in the specimen (specimen current) and enables monitoring these changes as the ion beam interacts with different materials in the sample. For example, the differences encountered when milling through some dielectric and striking another material are usually very subtle; however, once the beam begins milling into metal, a dramatic difference in absorbed charge can be observed on the EPD graph. In other words, end-point detection can be used to determine the dose per area required to complete the various mills, where a drop in secondary electron yield signaled the dose at which total removal of material in the patterned region is achieved.

Figure 17 shows the curves representing the signal from the all currently exposed objects in percent's on y-axis as a time (seconds) and depth (μm) or dose $\left(\frac{nC}{\mu m^2}\right)$ on x-axis for the milling through all 6 top different layers in the test chip before reaching the target nets in metal layer 3.

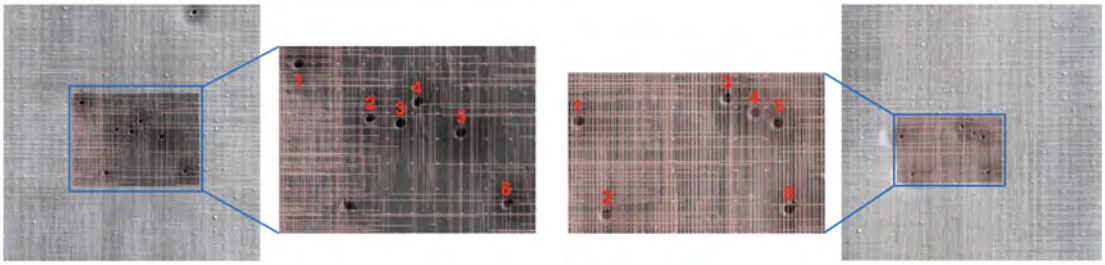
C. FIB Aspect Ratio Diagnosis

In this section, we show SEM images of a FIB circuit edit and calculate the radius at different metal layers of the edit in order to determine the FIB aspect ratio, R_{FIB}. Note that, radius value at different layers would be different since it uses technology-dependent parameters. Figure 18 shows the SEM image results of the circuit editing at M3 layer and we can get the radius of the milling hole at the top layer and its height, which refers to the r and h as shown in Figure 18(c). Besides, we also know the angle between the hypotenuse of the milling hole and the lower layer, which is 15 degrees and shown as angle y in Figure 18(c).

With these parameters, we can obtain that the radius at the lower layer, R in Figure 18(c), is 289.07 nm using

$$R = r - \frac{h}{\tan(y)} \tag{4}$$

Then, we can get the radius at different metal layers of the edit, as shown in Table IX. From these, we determine that the aspect ratio of our FIB is approximately 2.3.



(a) (b)

Fig. 19: FIB experiments on both non-protected and protected AES designs. (a) FIB milling holes on the non-protected AES implementation. (b) FIB milling holes on the protected AES implementation.

TABLE IX: Radius at different metal layers of the edit for SEAD32nm techonology library.

Metal layers	Approximate values of h for different metal layers (nm)	Value of R (nm)
M8	150	371.35
M7	910	331.89
M6	1110	321.50
M5	1310	311.12
M4	1510	300.73
M3	1710	290.35

D. Exposed Area and Shield Detection

Figure 19 shows the SEM images of FIB milling holes on the non-protected and protected AES implementations with images of upper metal layers overlaid. We choose 6 target nets and the holes of the same label denotes the same logical nets. We can see that there are more covering nets distributed in the milling holes on the iPROBE protected AES implementation, leading to less exposed area on the target nets, as shown in Figure 19. Further, it is worth noting that most of these locations are protected by the metal layer 6 shield in the iPROBE design. The edits shown clearly intersect with all of the holes in the protected design – thus, they should be detectable by iPROBE's comparators on the live chip.

VII. Conclusion

In this paper, we presented and demonstrated iPROBE, the first anti-probing physical design flow to protect ICs against probing attacks from both front-side and back-side. Evaluations on different implementations of four designs demonstrate that nearly 80% decrease on the exposed area can be achieved from both front-side and back-side compared to the original design, in which, the transistors' shielding ability consists of up to 10%. In addition, the timing and area overhead for the both front-side and back-side protection is less than 2%. Besides, SEM images of the circuit editing are demonstrated and we calculate the radius at different metal layers of edit. In the future, we hope to optimize the design overheads even further, allowing a designer to optimize the exposed area under overhead constraints or vice versa.

VIII. Acknowledgements

This work was supported in part by Semiconductor Research Corporation (SRC) under task ID 2769.001 and National Science Foundation (NSF) under project number 1717392. We are also thankful to Dr. Huanyu Wang and Dr. Qihang Shi for their great efforts in designing and implementing the test chip.

References

[1] M. T. Rahman, Q. Shi, S. Tajik, H. Shen, D. L. Woodard, M. Tehranipoor, and N. Asadizanjani, "Physical inspection & attacks: New frontier in hardware security," in *2018 IEEE 3rd International Verification and Security Workshop (IVSW)*. IEEE, 2018, pp. 93–102.

[2] S. Bhunia and M. Tehranipoor, *Hardware security: a hands-on learning approach*. Morgan Kaufmann, 2018.

[3] T. Zhang, J. Park, M. Tehranipoor, and F. Farahmandi, "Psc-tg: Rtl power side-channel leakage assessment with test pattern generation," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 709–714.

[4] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The em side—channel (s)," in *International workshop on cryptographic hardware and embedded systems*. Springer, 2002, pp. 29–45.

[5] S. P. Skorobogatov, "Semi-invasive attacks: a new approach to hardware security analysis," 2005.

[6] S. Tajik, H. Lohrke, J.-P. Seifert, and C. Boit, "On the power of optical contactless probing: Attacking bitstream encryption of fpgas," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1661–1674.

[7] H. Lohrke, S. Tajik, T. Krachenfels, C. Boit, and J.-P. Seifert, "Key extraction using thermal laser stimulation: A case study on xilinx ultrascale fpgas," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 573–595, 2018.

[8] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing ic piracy using reconfigurable logic barriers," *IEEE design & Test of computers*, vol. 27, no. 1, pp. 66–75, 2010.

[9] 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. 38 379–38 389, 2019.

[10] H. Wang, D. Forte, M. M. Tehranipoor, and Q. Shi, "Probing attacks on integrated circuits: Challenges and research opportunities," *IEEE Design & Test*, vol. 34, no. 5, pp. 63–71, 2017.

[11] V. Sidorkin, E. van Veldhoven, E. van der Drift, P. Alkemade, H. Salemink, and D. Maas, "Sub-10-nm nanolithography with a scanning helium beam," *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures Processing, Measurement, and Phenomena*, vol. 27, no. 4, pp. L18–L20, 2009.

[12] J.-M. Cioranesco, J.-L. Danger, T. Graba, S. Guilley, Y. Mathieu, D. Naccache, and X. T. Ngo, "Cryptographically secure shields," in *2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*. IEEE, 2014, pp. 25–31.

[13] M. Ling, L. Wu, X. Li, X. Zhang, J. Hou, and Y. Wang, "Design of monitor and protect circuits against fib attack on chip security," in *2012 Eighth International Conference on Computational Intelligence and Security*. IEEE, 2012, pp. 530–533.

[14] S. Manich, M. S. Wamser, and G. Sigl, "Detection of probing attempts in secure ics," in *2012 IEEE International Symposium on Hardware-Oriented Security and Trust*. IEEE, 2012, pp. 134–139.

[15] H. Wang, Q. Shi, A. Nahiyan, D. Forte, and M. M. Tehranipoor, "A physical design flow against front-side probing attacks by internal shielding," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2019.

[16] A. Covic, Q. Shi, H. Shen, and D. Forte, "Contact-to-silicide probing attacks on integrated circuits and countermeasures," in *2019 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*. IEEE, 2019, pp. 1–6.

[17] I. Synopsys, "Compiler user guide," 2013.

[18] "AES core," https://opencores.org/projects/aes-128_pipelined_encryption.

[19] H. Wu, L. Stern, D. Xia, D. Ferranti, B. Thompson, K. Klein, C. Gonzalez, and P. Rack, "Focused helium ion beam deposited low resistivity cobalt metal lines with 10 nm resolution: implications for advanced circuit editing," *Journal of Materials Science: Materials in Electronics*, vol. 25, no. 2, pp. 587–595, 2014.

[20] H. Wang, D. Forte, M. M. Tehranipoor, and Q. Shi, "Probing attacks on integrated circuits: Challenges and research opportunities," *IEEE Design Test*, vol. 34, no. 5, pp. 63–71, 2017.

[21] Q. Shi, N. Asadizanjani, D. Forte, and M. M. Tehranipoor, "A layout-driven framework to assess vulnerability of ics to microprobing attacks," in *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2016, pp. 155–160.

[22] H. Wang, Q. Shi, D. Forte, and M. M. Tehranipoor, "Probing assessment framework and evaluation of antiprobing solutions," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 6, pp. 1239–1252, 2019.

[23] M. Gao and D. Forte, "iprobe-o: Fib-aware place and route for probing protection using orthogonal shields."

[24] "Simon Block Cipher Implementation," [Online], https://github.com/inmcm/Simon_Speck_Ciphers, Accessed Nov. 19, 2020.

[25] "Present Hardware Core," [Online], https://github.com/saiedhk/PresentCryptoEngine, Accessed Nov. 19, 2020.

[26] P. Kurup and T. Abbasi, *Logic synthesis using Synopsys®*. Springer Science & Business Media, 2012.

[27] V. Naganathan, "A comparative analysis of parallel prefix adders in 32nm and 45nm static cmos technology," Ph.D. dissertation, 2015.

[28] "Practical silicon-backside-protection method for abnormally detection," *Journal of Semiconductor Technology and Science*, vol. 19, no. 6, pp. 577–584, Dec. 2019.

[29] S. Borel, L. Duperrex, E. Deschaseaux, J. Charbonnier, J. Cledière, R. Wacquez, J. Fournier, J.-C. Souriau, G. Simon, and A. Merle, "A novel structure for backside protection against physical attacks on secure chips or sip," in *2018 IEEE 68th Electronic Components and Technology Conference (ECTC)*, 2018, pp. 515–520.

[30] https://www.tsmc.com/english/dedicatedFoundry/.

[31] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," in *IEEE International Symposium on Circuits and Systems,*. IEEE, 1989, pp. 1929–1934.

[32] https://opencores.org/.

[33] E. Laboratories, "Focused ion beam circuit edit becomes increasingly valuable in high-stakes world of advanced node design," 2017.

Minyan Gao received the B.S. degree from Northwest University, Xi'an, China, in 2016, and the M.E. degree in Electrical Engineering from University of Virginia, Charlottesville, VA, USA, in 2018. She is currently working toward the Ph.D. degree in Electrical Engineering at Florida Institute for Cyber Security (FICS), University of Florida, Gainesville, FL, USA. Her current research interests include hardware security and trust, VLSI CAD, VLSI physical design.

M Sazadur Rahman is a final year Ph.D. student under the supervision of Prof. Mark Tehranipoor. He got his B.Sc. in Electrical and Electronic Engineering from the Bangladesh University of Engineering and Technology. He worked as a design engineer in different fabless semiconductor companies for four years, working in industrial scale 28nm and 14nm custom ICs. His research has resulted in one patent, one book, and several peer-reviewed publications in premier ACM/IEEE journals and conferences.

Nitin Varshney is currently working as a Lab Engineer at Florida Institute for Cyber Security (FICS) Research from last 3 years. He got his Masters Degree in Materials Science from University of California, San Diego. He has multiple Internship experience in Hardware Assurance during his bachelor's at Australian National University, National University of Singapore , A*STAR and North Carolina State University. His extensive experience with physical assurance and failure analysis tools like FIB-SEM, X-RAY, TEM, PHEMOS, LSI, Probing, Sample Preparation etc. has resulted in 2 patents and several peer-reviewed publications.

Mark Tehranipoor is currently the Intel Charles E. Young Preeminence Endowed Chair Professor in Cybersecurity at the University of Florida. His current research projects include: hardware security and trust, supply chain security, IoT security, VLSI design, test and reliability. He is a recipient of a dozen best paper awards and nominations, as well as the 2008 IEEE Computer Society (CS) Meritorious Service Award, the 2012 IEEE CS Outstanding Contribution, the 2009 NSF CAREER Award, and the 2014 AFOSR MURI award. He received the 2020 University of Florida Innovation of the year as well as teacher/scholar of the year awards. He co-founded the IEEE International Symposium on Hardware Oriented Security and Trust (HOST), IEEE International Conference on Physical Assurance and Inspection of Electronics (PAINE). He serves on the program committee of more than a dozen leading conferences and workshops. He has also served as Program and General Chair of a number of IEEE and ACM sponsored conferences and workshops (HOST, ITC, DFT, D3T, DBT, NATW, and more). He is currently serving as a founding EIC for Journal on Hardware and Systems Security (HaSS) and served as Associate Editor for TC, JETTA, JOLPE, TODAES, IEEE D & T, TVLSI. He is currently serving as a founding director for Florida Institute for Cybersecurity Research (FICS) and a number of other centers with focus on microelectronics security. Dr. Tehranipoor is a Fellow of the IEEE and ACM, a Golden Core Member of IEEE CS, and Member of ACM SIGDA.

Domenic Forte received the B.S. degree from the Manhattan College, Riverdale, NY, USA, in 2006, and the M.S. and Ph.D. degrees from the University of Maryland at College Park, College Park, MD, USA, in 2010 and 2013, respectively, all in electrical engineering. He is currently an Associate Professor with the Electrical and Computer Engineering Department, University of Florida, Gainesville, FL, USA. His research interests include the domain of hardware security, including the investigation of hardware security primitives, hardware Trojan detection and prevention, electronics supply chain security, and anti-reverse engineering. He was a recipient of the Presidential Early Career Award for Scientists and Engineers (PECASE), the Early Career Award for Scientists and Engineers (ECASE) by the Army Research Office (ARO), the NSF Faculty Early Career Development Program (CAREER) Award, and the ARO Young Investigator Award. His research has also been recognized through multiple best paper awards and nominations.