Covert Gates: Protecting Integrated Circuits with Undetectable Camouflaging

Bicky Shakya, Haoting Shen, Mark Tehranipoor and Domenic Forte

ECE Department, University of Florida
bshakya@ufl.edu,{htshen,tehranipoor,dforte}@ece.ufl.edu

Abstract. Integrated circuit (IC) camouflaging has emerged as a promising solution for protecting semiconductor intellectual property (IP) against reverse engineering. Existing methods of camouflaging are based on standard cells that can assume one of many Boolean functions, either through variation of transistor threshold voltage or contact configurations. Unfortunately, such methods lead to high area, delay and power overheads, and are vulnerable to invasive as well as non-invasive attacks based on Boolean satisfiability/VLSI testing. In this paper, we propose, fabricate, and demonstrate a new cell camouflaging strategy, termed as 'covert gate' that leverages doping and dummy contacts to create camouflaged cells that are indistinguishable from regular standard cells under modern imaging techniques. We perform a comprehensive security analysis of covert gate, and show that it achieves high resiliency against SAT and test-based attacks at very low overheads. We also derive models to characterize the covert cells, and develop measures to incorporate them into a gate-level design. Simulation results of overheads and attacks are presented on benchmark circuits.

Keywords: IP Protection · Camouflaging · Reverse Engineering · SEM Imaging · ATPG · SAT

1 Introduction

Reverse engineering of integrated circuits (ICs) is a common practice in the semiconductor industry. It is routinely used for (i) failure analysis, defect identification, and fault diagnosis, (ii) detection of counterfeit ICs and (iii) analysis of competitor IP (e.g., technology node or process analysis, and checking whether patents were infringed) [TJ07][TJ11]. While reverse engineering for such purposes is a legal and acceptable practice, it can also be done with malicious intent. For example, a reverse engineer may obtain the complete gate-level netlist of the circuit through reverse engineering of an IC's physical layout. By doing so, the reverse engineer could infringe on the owner's intellectual property (IP), by incorporating the extracted IP core into his or her own design or by selling it to third parties. The illicitly obtained IP could also be used to created cloned ICs and electronic systems [GHD+14][MBT17]. Further, an untrusted semiconductor foundry could also use reverse engineering to fully understand the functionality of a design and insert a targeted, stealthy hardware Trojan [TK10][BRPB13].

Traditional IC reverse engineering involves four distinct phases, namely:

1. **Decapsulation** involves removing the IC packaging using corrosive chemicals and/or abrasion (e.g., by polishing the package surface).

2. **Delayering** is the layer-by-layer removal of the metal interconnects between logic gates, polysilicon traces as well as the transistor layers. The insulating layer between different metal layers is also removed. Dry or wet etching (i.e., removing specific materials via chemical reactions) is employed for this purpose.

IACR Transactions on Cryptographic Hardware and Embedded Systems ISSN 2569-2925, Vol. 2019, No. 3, pp. 86–118
DOI:10.13154/tches.v2019.i3.86-118

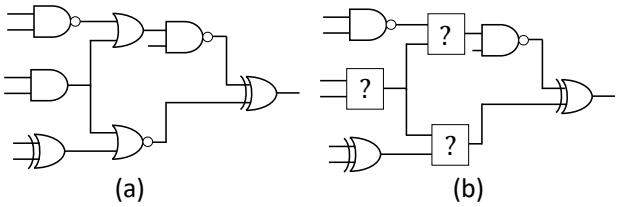


Figure 1: (a) Original gate-level netlist and (b) Netlist containing camouflaged cells obtained after reverse engineering.

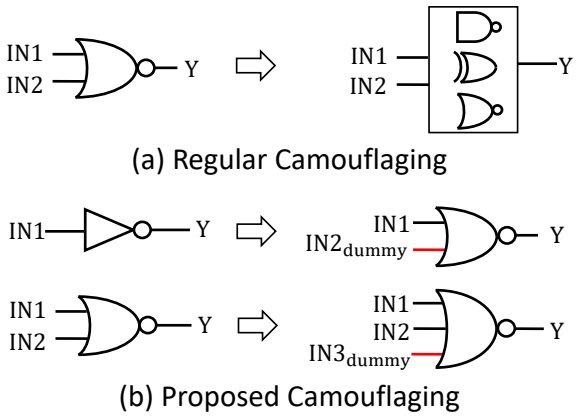


Figure 2: (a) Regular camouflaged gate configurable based on dummy contacts, threshold voltage, or dopant polarity; (b) Proposed camouflaging that introduces dummy inputs.

3. **Imaging** is used to capture the IC layout at each layer after delayering. Scanning electron microscopy (SEM) is used for this purpose to capture high resolution images of micro/nano-scale features.

4. **Post-processing** includes stitching of imaged layers after image processing, identifying primitives (e.g., logic gates) and obtaining a gate-level netlist, with associated connections between them.

IC camouflaging techniques focus on disrupting the ability of the attacker to identify logic gates from the images obtained after delayering and imaging. For instance, this may be achieved by creating special standard cells with 'dummy contacts' [RSSK13]. Based on the configuration of the contacts, a cell can implement one of many Boolean functions (e.g., AND, OR, NAND, NOR, XOR). When an attacker delayers and images the IC, he/she would be unable to resolve the contacts and reveal the true identity of the camouflaged gates. As a result, some of the gates in the netlist obtained through the reverse engineering process become ambiguous, as shown in Figure 1. However, it has been shown that camouflaging can be vulnerable to attacks adapted from VLSI testing [RSSK13] and Boolean satisfiability [EMGT15, SRM15]. For successfully performing these attacks, the adversary generates a set of input patterns and observes the outputs to decide the identity of the camouflaged gates. Test-based attacks generate these patterns by modeling the gate behavior with manufacturing faults and using common automatic test pattern generation (ATPG) techniques. On the other hand, satisfiability (SAT)-based attacks attempt to find proper logical assignments to the camouflaged gates by using various SAT formulations. It has been shown that creating secure designs with camouflaged gates that are resilient to these attacks incurs significant overhead in terms of area and performance.

In this paper, we take a different approach to IC camouflaging. In contrast to using readily identifiable camouflaged gates, we modify regular logic gates to introduce 'dummy input(s)'. When a reverse engineer tries to recover the netlist, the gate is identified as any regular cell (e.g., AND, OR, XOR). Hence, it is coined as a 'covert gate'. However, an extra pin (or pins) is introduced into the gate, due to which the recovered netlist becomes erroneous. A comparison between previous camouflaging and our proposed technique is shown in Figure 2. As seen in Figure 2(a), all previous camouflaging approaches configure a logic gate into any one of N possibilities (e.g., NAND, NOR or XOR). However, these cells are easily identifiable in the design, consume large area/power/delay overheads and are susceptible to attacks. In case of covert gates, a regular gate with N inputs is transformed to the same gate with $N+i$ inputs, where i is the number of dummy inputs used, as shown in Figure 2(b). For example, a 2-input NOR gate is transformed into a 3-input NOR gate. Similarly, an inverter with one input is transformed to a NOR gate with 2 inputs, where one of the inputs is a dummy. These gates still function as intended (i.e., a camouflaged 2 input NOR gate still behaves as an inverter). However, this is not obvious to the attacker, as one (or more) of the inputs is configured as dummy and the camouflaged gates look no different from regular gates. Compared to prior camouflaging approaches, our covert gates offer the following benefits:

- *Inexpensive:* A covert gate can be implemented by a minor change in the IC fabrication process, with only three additional masks required.

- *Low overhead:* The covert gate incurs minimal overhead compared to previously proposed camouflaged gates, which need to be configured for various Boolean functions in one single cell. This contributes to their high layout area and poor leakage power/delay performance. In contrast, covert gates with dummy inputs consume no more area than regular standard cells and have much lower delay/power overheads. For example, the delay and power characteristics of an inverter that has been converted to a 2 input covert NAND/NOR gate are similar to that of a regular 2 input NAND/NOR gate, with area increasing by less than $1.67X$. In contrast, the area, delay and power overheads of a dummy contact based camouflaged $NAND$ gate are 5.5X, 1.6X and 4X respectively.

- *Indistinguishable from standard cells under SEM:* From an attacker's perspective, all gates with more than one input are considered as suspect in the entire design, as the covert gates 'blend in' with other gates in the netlist. Therefore, any invasive or non-invasive attack has to consider all gates in the design, which greatly increases attack complexity without the need for additional countermeasures.

- *Tunable output corruptibility:* There is also the opportunity to choose nets in the design to connect to the dummy inputs, such that the functional/logical difference between the netlist recovered by the attacker and the original design is further increased.

1.1 Contributions

Our main contributions in this paper include the following:

- We propose covert gates based on doping modification and dummy contact that look no different from regular gates.

- We demonstrate how these gates can be created by minor changes in the foundry fabrication process, without incurring high costs.

- We present fabrication and SEM imaging results on structures with various doping and material stacks, demonstrating that even with state-of-the-art SEM used for reverse engineering, the modification used to create the covert gates cannot be resolved. To our knowledge, this is the first paper that provides a camouflaging strategy with definitive results on SEM imaging resistance.
- We propose circuit models for the covert gates that allow overhead assessment during the design phase.
- Finally, we perform SAT as well as test (ATPG)-based security analysis of designs camouflaged with our proposed technique under multiple scenarios, and quantitatively demonstrate that both attacks do not scale well for compromising designs that contain our covert gates.

The rest of the paper is organized as follows. Section 2 reviews the state-of-the-art in IC camouflaging. It also explains the adversarial model used and the attacks proposed so far in the context of camouflaging. Section 3 introduces covert gates (our proposed camouflaging technique), along with motivation, logic gate construction and overhead characterization. It also provides SEM imaging results to demonstrate that the doping scheme used to construct the proposed camouflaged gates is not detectable. Section 4 analyzes resiliency of the proposed scheme to SAT-based attacks. Section 5 describes how test generation techniques can be used to analyze camouflaging security. Section 6 provides results on SAT and ATPG attacks on the covert gate strategy. Finally, Section 7 concludes the paper and provides directions for future work.

2 IC Camouflaging

2.1 Scope and Attack Model

IC camouflaging intends to protect semiconductor IP once a manufactured chip enters the open market, and finds its way into the hands of an adversary. The design house or entity designing the chip as well as the foundry fabricating the chip are assumed to be trusted. The problem of untrusted foundries is usually tackled by techniques such as logic locking, and is beyond the scope of camouflaging, since the foundry has the mask information needed to fabricate the camouflaged cells. As in previously proposed camouflaging approaches, it is assumed that the reverse engineer has access to the following:

- **The design netlist** which is obtained after performing IC reverse engineering, and includes camouflaged cells with unknown functions.
- **A functional chip** which the attacker can use as an oracle (i.e., apply input patterns and observe known-good responses). This is because camouflaging does not affect the functionality of a manufactured chip; rather, it only prevents an adversary from obtaining a functionally correct netlist after imaging the chip.
- **Scan chain access** is also commonly assumed. A scan chain, shown in Figure 3, is made by chaining together all the flip-flops in a design to form a shift register. This allows a user to excite each combinational logic cone (CLC) in the design with their input vector of choice [BA04]. To do so, input vectors are loaded through the scan-in (SI) port instead of the primary input (PI). After several clock cycles, the vectors are loaded onto each individual CLC. Then, the responses can be readout by running the IC for one clock cycle and scanning out the results. This is vital because the presence of a scan chain *transforms a sequential circuit into several small combinational circuits*, each of which can be accessed individually. If the scan chain was not present, loading of desirable input vectors into each specific CLC would not

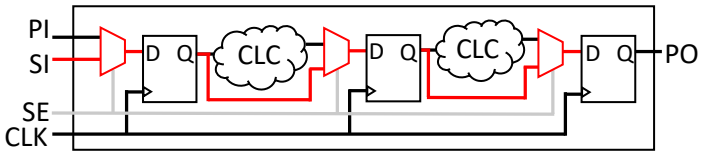


Figure 3: Scan chain-inserted design where PI, SI, SE, CLK, CLC, and PO stand for Primary Inputs, Scan In, Scan Enable, Clock, Combinational Logic Cone and Primary Outputs.

be feasible, as many CLC's would not be accessible directly via the primary inputs (PIs). As we will see in the following sections, an attacker cannot excite camouflaged cells with desired input vectors and observe their behavior without this capability.

2.2 SEM Imaging for Reverse Engineering

When the IC feature size went below the optical spatial resolution (<1 µm), SEM became the tool-of-choice during the imaging step of IC reverse engineering flow. SEM is a powerful magnification system that employs focused electron beams (e-beams) to capture information from a sample surface. Various properties of the sample surface, such as topography, conductivity, chemical component, and surface potential, can be revealed by SEM imaging. Although the feature size of modern ICs has scaled down to sub-10 nm, SEM imaging resolution capabilities have also grown. For certain materials, SEMs can achieve sub-1 nm spatial resolution. In addition, SEM is able to perform imaging over a large surface in an efficient manner, compared to other microscopy techniques such as transition electron microscope (TEM), atomic force microscope (AFM), etc. Thus, due to its high resolution and efficiency, SEM is the most popular imaging technique for IC reverse engineering. Hence, any secure camouflaging technique must be resistant against SEM imaging, as the images obtained from SEM are directly used for annotation and netlist extraction.

The most challenging part of IC reverse engineering is recognizing different doping regions and gates in the transistor layers. This is due to the weak contrast and reduced resolution resulting from material similarities [TJ07][CEMG16]. Beside surface materials and topography (which are the two most common contrast sources for SEM), passive voltage contrast (PVC) is also necessary for feasible imaging of the transistor features. Two factors are essential for PVC: surface potential and charging effect. Depending on the doping type and the doping concentration, the surface potential of silicon (Si) varies. Such variations can be captured by SEM [EBH02, Che16]. It is worth noting that although 1 nm wide doped features were claimed to be detected in [EBH02], the feature width in the final image was larger than 20 nm. In this case, if the two doping regions are closer than 20 nm, they cannot be differentiated. Besides doping, the CMOS structure itself also causes grounding effects, which decide the charge accumulation rate on the sample surface. Moreover, the accumulation of surface charge changes the surface potential, and thus, affects the SEM images. It is also reported that primary electron (i.e., e-beam) energy is important for doping-based SEM image contrast. Low energy (< 5 keV) is suggested in [EBH02, Che16, SSF+14]. For IC imaging, the e-beam energy is critical for yet another reason: the penetration depth (i.e., reaction volume). Higher energy e-beam goes deeper into the material, and provides more information from the deeper layers. Because some layers in an IC are as thin as a few nanometers and the signal-to-noise ratio (e.g., from doping-related information) can be weak, the contrast of such layers will be easily lost by using a high energy e-beam.

2.3 Invasive Attacks

After reverse-engineering, an attacker has access to a gate-level netlist. In the case of existing camouflaging techniques, he/she also knows which exact cells are camouflaged in the design. The limited number of camouflaged gates could allow an attacker to feasibly conduct various invasive attacks on each of the camouflaged cells and, thereby, recover their functionality. For example, a recently proposed de-processing technique [PAF+17] allows high resolution milling of circuit layers using plasma focused ion beam (FIB) tools. Using FIBs, an attacker could mill the metal contacts to observe the presence (or absence) of insulating layers. Optical probing techniques, such as photon emission probing, laser-induced fault injection, and laser voltage probing (LVP) [LTBS16], allow observation of gate-level switching activity, which can be used to help recognize the camouflaged gate functions. Compared to regular camouflaging, our approach severely limits the ability of the attacker to conduct invasive attacks to resolve the covert gates. This is because he/she is forced to apply FIB milling/imaging or LVP on every single gate in the design (as covert gates are identical in layout to regular standard cells). This is clearly infeasible, considering the time required to perform the invasive attacks and the number of gates in modern designs.

2.4 Non-invasive Attacks on Camouflaging

While an attacker can perform invasive attacks on each individual camouflaged cell, the presence of the netlist and a functional IC with scan chain access also makes it possible for an attacker to *non-invasively* reveal the identity of the camouflaged gates. Such attacks would be favored over invasive ones, in scenarios where the attacker does not have access to state-of-the-art probing and imaging tools. Two significant types of such non-invasive attacks have been proposed on camouflaging techniques, where a given cell can implement one of N different functions.

2.4.1 Test-based attacks:

In test-based attacks (also referred to as ATPG-based attacks in this paper), the attacker leverages common test generation techniques as well as the on-chip scan chain mechanism to excite individual camouflaged gate inputs. Once the camouflaged gate is stimulated, its response is propagated to an observable output point (for example, the primary outputs or scan flip-flops). This response can then be checked against the response from a functional IC to decide the Boolean function of the camouflaged gate [RSSK13]. For example, a 2-input camouflaged gate implementing either NAND, NOR or XOR can be stimulated with the vector $< 00 >$. If its response is observed to be 0, one can be certain that the gate implements XOR. This attack requires two actions to be performed:

- **Sensitize:** It should be possible to set the input pins of the camouflaged gates to the desired value through either the primary inputs or via the scan chain.

- **Propagate:** Once the camouflaged gate's pins are set to the desired value, the output of the gate should be propagated to an observable point such as the primary output or a flip-flop (from which it can be flushed out to the scan-out port via the scan chain). This requires that every other gate in the path between the camouflaged gate and the observe-point be set to a non-controlling value. Otherwise, the output of other logic gates would interfere or mask the camouflaged gate's logic value, and render it un-observable.

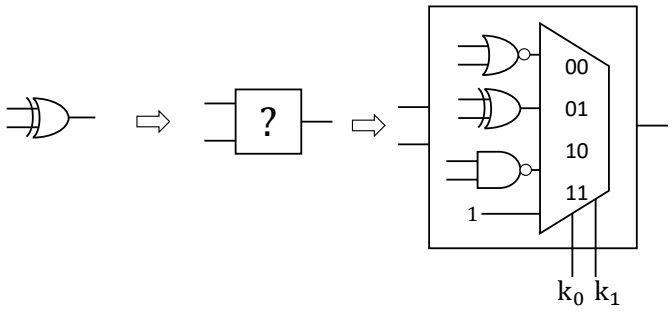


Figure 4: SAT formulation for resolving camouflaged gates. The original gate is unresolvable by imaging (→ camouflaged gate). The camouflaged gate is represented by multiple functionalities determined by a key variable (→ MUX network controlled by key bits).

2.4.2 SAT-based attacks:

In SAT-based attacks, it is also assumed that there is scan access so that the design (usually sequential) reduces to several combinational circuits. Once this is done, each camouflaged gate is replaced with a selection network as shown in Figure 4, controlled by key variables. Now, the task of the solver is to find the correct 'key values' so that the camouflaged netlist is functionally equivalent to the IC. Once the 'key' is found, the multiplexer network is replaced with the correct gate assignment, leading to a fully resolved netlist. For example, in Figure 4, the key values $k_0 = 0, k_1 = 1$ lead to a correct assignment, as the original gate that was camouflaged was an XOR. Recent results have shown that the attack only requires a few input-output observations from a functional IC to rule out incorrect camouflaged gate assignments and converge on the correct identity of the gates [EMGT15] [LYZH16].

Several circuit-level countermeasures have also been proposed to resist SAT-based attacks [LSM+16] [XS16]. These techniques mainly focus on limiting the information gained by the solver from the functional IC i.e., they reduce the number of incorrect key or camouflaged gate assignments that can be ruled out in a single iteration, thereby requiring the solver to take an exponential number of iterations to reveal the gate identities. This is usually achieved by the insertion of 'SAT-resistant logic' (e.g., AND trees [LSM+16] or point functions [YMSR16]) into the circuit. However, they come at the cost of either reduced circuit corruptibility (i.e., even with the wrong camouflaged gate assignments, the circuit is functionally still very similar to the original design), susceptibility to removal attacks [YMSR17], or susceptibility to bypass attacks [XSTF17]. Further, these techniques are also vulnerable to 'approximate SAT' or AppSAT attacks [SLM+17]. In contrast to regular SAT attack, AppSAT intermittently computes the error rate of the circuit (using logic simulation) and terminates the attack if the error rate drops below a pre-defined threshold. This is helpful as regular SAT attack only terminates if it finds a provably correct camouflaged gate assignment. Unfortunately, this prevents the algorithm from terminating early, even if its current key assignment has a sufficiently low error rate (i.e., most of the camouflaged gate assignments are correct). On the other hand, AppSAT guarantees an 'almost correct' key (i.e., most camouflaged gate identities are resolved) while quickly converging.

2.5 Survey of Current Techniques

Several techniques for creating and integrating camouflaged gates have been recently proposed. They can be grouped into the following categories.

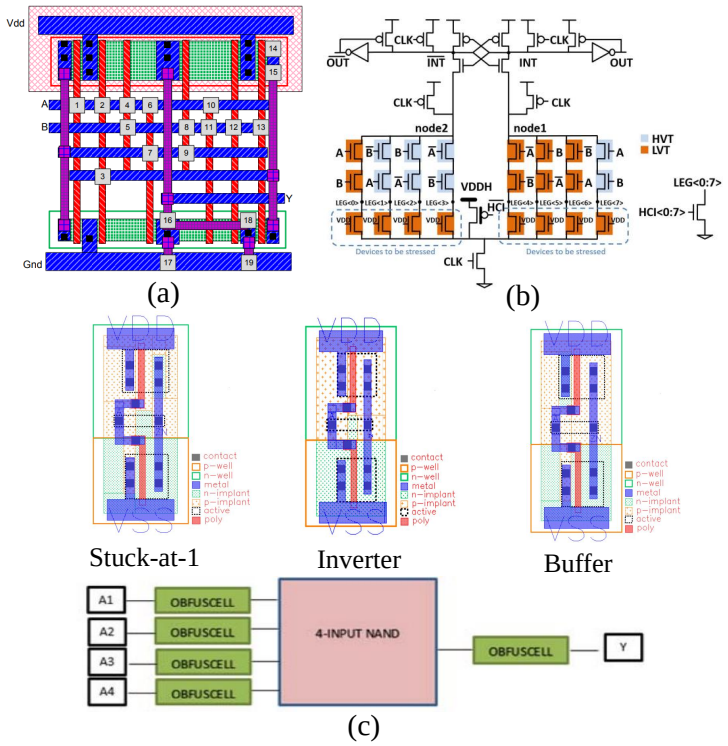


Figure 5: (a) Dummy contact-based camouflaged gate that can be configured as $NAND$, NOR or XOR [RSSK13] (b) Post-manufacturing programmed threshold voltage defined gate [AEM18] (c) Camouflaged gate based on dopant polarity - The 'obfuscell' can be configured as an inverter, buffer or be stuck-at-1/0. The cell is combined with a logic gate to create a multifunctional 'ObfusGate' [MBPB15]

2.5.1 Dummy contact-based camouflaging

The most widely known embodiment of camouflaging leverages dummy contacts to create a cell capable of implementing a variety of Boolean functions [RSSK13] (Figure 5a). In these cells, some contacts between the metal layers and the gate are actual tungsten contacts whereas others have a fine insulating layer of silicon dioxide SiO_2 separating the logic gate terminals and metal layers. When imaging is performed on these gates, it is believed to be difficult (if not impossible) to observe the validity of the contacts from the IC frontside or backside. Therefore, without valid contact information, the identity of the camouflaged gate is not revealed.

The drawback of such an approach is the high overhead incurred. As each camouflaged gate contains a large number of transistors (whose connections can be configured by the dummy contacts), the area, delay and power characteristics of these cells are much higher than regular standard cells. For example, a camouflaged $NAND$ gate that can be configured as a $NAND$, NOR or XOR gate is reported to have power, delay and area overheads of 5.5X, 1.6X and 4X respectively [RSSK13]. Thus, a designer needs to devise various techniques for insertion of a limited number of such cells into the design. Further, such techniques have also been shown to be vulnerable to both SAT and test-based attacks. Test-based attacks are countered by inserting camouflaged gates in such a way that their logic function cannot be resolved by sensitization and propagation [RSSK13]. However, SAT-resistant camouflaging is harder to achieve, as it may lead to low output corruptibility, more overhead or the inserted extra logic to counter SAT attacks may be prone to further

attacks, as discussed in Section 2.4.2. It should also be noted that the basis of both SAT and test-based attacks is that an attacker can tell which gates in the extracted netlist are regular standard cells and which are camouflaged cells. If all gates in a design were camouflaged, both SAT and test-based attacks would be thwarted, albeit at unreasonable overheads.

2.5.2 Threshold voltage-based camouflaging

Several techniques that configure a logic gate into various functions, depending on the threshold voltage configuration of the transistors, have also been proposed. The authors in [CEMG16] proposed a new circuit structure, composed of pass transistors with varying threshold voltages (V_t). Depending on the configuration of the threshold voltages, the same structure can function as an AND gate or an OR gate. Another threshold voltage-based approach using sense amplifier-based logic (SABL) is experimentally demonstrated in [AEM18], where the V_t of transistors in a camouflaged gates is programmed post-fabrication using intentional hot carrier injection (Figure 5b). In [IVR+18], threshold-defined pass transistors, which can either be programmed as open or closed based on their V_t configuration, are used in the design of multi-functional camouflaged gates.

Similar to dummy contact-based camouflaging, these techniques come at the cost of high area, delay and power overheads. For example, a threshold voltage-defined $NAND$ gate proposed in [AEM18] consumes 9.2X power, 6.6X delay and 7.3X area overheads. Further, from a security perspective, they have the same vulnerability to SAT and test-based attacks as dummy contact-based camouflaging. This is because the V_t-based camouflaged gates, regardless of their mode of implementation, can be replaced with a formulation as shown in Figure 4 and consequently, resolved by SAT attack.

2.5.3 Doping-based camouflaging

In [MBPB15], the authors manipulate dopant polarity in the layout of logic gates to perform camouflaging. Depending on how the doping is adjusted in the NMOS and PMOS transistor regions, the same cell (termed as an 'obfuscell') can behave as an inverter, buffer or a cell that always outputs logic 0 or logic 1, as shown in Figure 5c. These obfuscells can then be used to create complex 'obfusgates' that implement a large number of functions, depending on the obfuscell configuration. Similar to dummy contact and V_t-based camouflaging, obfusgates suffer from high overheads. For example, an average of 7.09X area, 6.45X power and 3.12X delay overhead was incurred while using the NAND-type obfuscation gate on an AES S-Box circuit. The approach has also been shown to be vulnerable to SAT attacks [YZL+17]. Further, the doping scheme used to implement obfuscell can be detected by imaging [SSF+14].

2.5.4 Interconnect camouflaging

The authors in [CCF+15] take the route of obfuscating the interconnects instead of modifying the logic gates. They propose the use of magnesium wires that instantly oxidize on exposure during reverse engineering and become indistinguishable from dummy wires that are made of magnesium and have been oxidized intentionally during the fabrication. Therefore, each gate has a large number of possible input/output pin choices.

The work in [YZL+17] showed that such transformable interconnects can also be attacked by replacing the unresolvable interconnects with a MUX or switch network, and using the SAT attack algorithm to find the key that configures the suspect interconnects correctly. Further, no imaging results were presented to see if there was any difference between the dummy magnesium oxide interconnects and the magnesium interconnects that oxidize on exposure to air. It is also unclear if there are fabrication, volatility and reliability challenges associated with magnesium interconnects.

Table 1: Comparison of various regular camouflaging techniques, where a camouflaged gate assumes one of N logical functions, and covert gates. Note that some techniques are SAT or test attack resistant after application of countermeasures (we mark these as ✗). If the technique is naturally resistant to such attacks without any countermeasure, we mark it as ✓. '?' implies that the particular feature of the approach is unverified.

Feature	Regular Camouflaging				Covert Gates
	Dummy Contact	Threshold Voltage	Doping	Transformable Interconnects	
SAT resistant at low overhead	✗	✗	✗	✗	✓
Test attack resistant	✗	✗	✗	✗	✓
Low overhead	✗	✗	✗	✗	✓
Configurable after fabrication	✗	✓	✗	✗	✗
Imaging resistant	✓	✓	✓	?	✓
Undetectable during netlist extraction	✗	✗	✗	✗	✓

A comparison of all these camouflaging techniques with our proposed covert gate is presented in Table 1. It shows that covert gate is the only low overhead approach that is resistant to various attacks, without the need for any additional countermeasures.

3 Covert Gates

3.1 Motivation

While a plethora of techniques have been introduced for IC camouflaging, they all revolve around the same idea of creating a 'special cell' that can implement a variety of functions depending on how they are fabricated or configured. Unfortunately, it has been shown that test and SAT-based attacks can compromise most of these techniques, i.e., they can reveal the camouflaged gate identities, even if their precise function cannot be resolved by inspection during reverse engineering/imaging. Countermeasures against these non-invasive/functional attacks have also proven to be costly in terms of overhead. Moreover, they have their own set of vulnerabilities (e.g., low output corruptibility and removal attack susceptibility). Towards the side of invasive attacks, a reverse engineer could use advanced probing tools (e.g., LVP) to make direct contact with the camouflaged gate pins and reveal their identity by exhaustive tests (e.g., test all four possible input combinations for a 2-input camouflaged gate).

With these limitations and vulnerabilities in mind, we try to bring back the literal meaning of 'camouflage' into IC camouflaging, by creating 'covert gates'. More specifically, we focus on creating logic gates *that appear no different from any other regular gate in the design under SEM imaging.* Yet, these cells are not a part of the original circuit functionality, and are intended to mislead the reverse engineer so that an incorrect netlist is extracted. This leads to a fundamental issue for any kind of attack on IC camouflaging: *the attacker has to first figure out which gates are being camouflaged before even proceeding with any type of attack.* This leads to a significant increase in complexity for the attacker, in contrast to the scenario where they are readily able to tell which cells are camouflaged and which aren't. In terms of invasive attacks, thinning/imaging and laser voltage probing would need to be carried out on practically every gate in the design, since covert gates are indistinguishable from normal ones. This would lead to infeasible cost and time for an attacker, as both FIB and LVP cannot be conducted on-the-fly on every gate in a large design. While non-invasive attacks based on SAT and ATPG are not as time-consuming or expensive as invasive ones, their scalability would also suffer, as every gate is suspect. As we will show in Sections 2.4.2 and 5, the SAT-attack would need to encode a very large number of gates to resolve the covert gates, and ATPG-based attacks would also be unable to generate/propagate test patterns/responses to distinguish the gates.

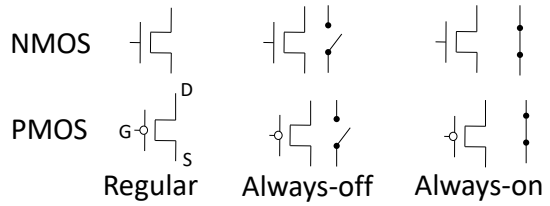


Figure 6: Regular NMOS/PMOS transistors and always-on/always-off transistors (where $S \rightarrow$ Source, $G \rightarrow$ Gate, $D \rightarrow$ Drain).

3.2 Covert Cell Fabrication and Operating Principle

In order to create camouflaged covert gates that look exactly like normal logic gates but with dummy inputs, we require variants of regular PMOS and NMOS transistors. While regular NMOS transistors conduct when a high voltage is applied to its gate (and PMOS transistors conduct when a low voltage is applied), we can create PMOS and NMOS transistors that are always open (or closed), regardless of the input applied at its gate. These variants are shown in Figure 6. Using these transistors, we can construct a 2 input NOR gate $Y = NOR(A, B)$, as shown in Figure 7. If we use an always-on transistor in the pull-up network (behaving as a shorted wire) and an always-off transistor in the pull-down network (behaving as an open wire), the input applied to pin B no longer affects the functionality of the NOR logic gate. Thus, the output Y will be high (low) whenever the input A is low (high). Thus, the NOR gate is transformed into an inverter, with a *dummy input* B. In a similar fashion, AND, OR, $NAND$, XOR and a variety of other gates can be camouflaged by using a combination of always-on/always-off transistors to create n dummy inputs.

3.2.1 Prior Work on Always-on/Always-off Transistors

Note that such type of camouflaged gates with always-on/always-off transistors have previously been proposed in [CCJB07] and briefly discussed in [LSM+16] as well. In [CCJB07], lightly-doped drain (LDD) structures with opposite doping polarity in the source/drain regions are used. The LDD regions act as insulators, creating an open defect in the transistors. In [LSM+16], such LDD-based structures were leveraged to create gates with dummy or 'stuck-at' inputs, through a combination of always-on/always-off transistors. However, it should be noted that such LDD-based structures are quite costly to fabricate. This is because regular LDD structures have the same polarity as that of the source/drain regions (i.e., n-type doping is applied on the LDD region and $n++$ in the source/drain region for NMOS). To create a stuck open fault, a $p+$ doping is applied in the LDD region along with $n++$ in the source/drain region (for NMOS). As self-alignment is not possible, doping would need to be performed precisely on the LDD regions with accurate alignment, which is extremely difficult for newer technology nodes. Further, it can be expected that the junction between the highly doped LDD and source/drain regions will create high leakage current. It is also unclear if such structures are invisible under SEM imaging. Further, both [CCJB07] and [LSM+16] do not propose a fabrication mechanism to create the always-on transistors, which are needed to complement the always-off transistors in our proposed approach.

3.3 Design of Device Structure and Fabrication

In this section, we describe how the transistors needed to construct the proposed covert gates can be fabricated in CMOS technologies.

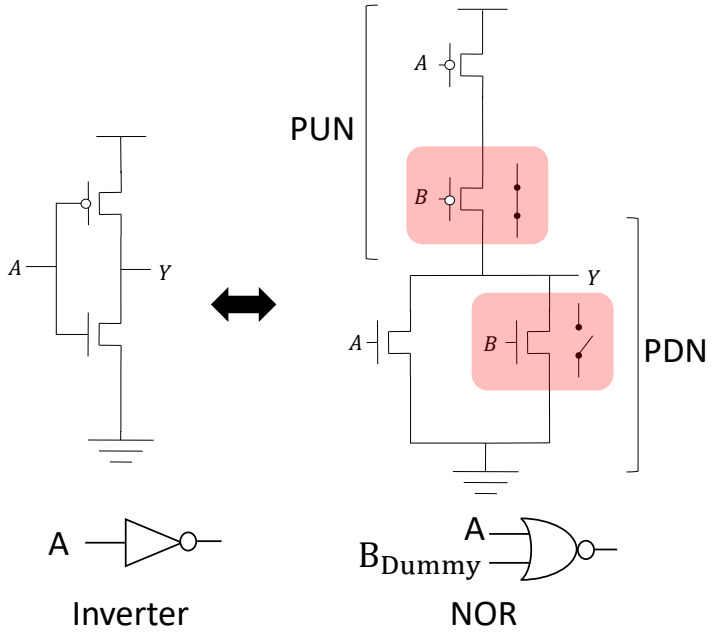


Figure 7: A 2 input NOR gate where B is a dummy input. PMOS transistor in the pull-up network (PUN) connected to B is always on (short circuit) and the NMOS transistor in the pull-down network (PDN) connected to B is always off (open circuit). As a result, the gate behaves like an inverter with input A.

3.3.1 Always-on transistor

To set a transistor as 'always-on', the transistor is fabricated with a heavily doped implanted channel, with the same doping type as the source and drain. The dopant concentration should be higher than the effective carrier concentration in a strongly inversed layer. Consequently, the heavily doped channel will not be significantly affected by gate bias and the contacts between source/channel and channel/drain are low-resistance Ohmic contact. A permanent "stuck-on" fault is thus formed intentionally. The fabrication steps needed to implement this structure is illustrated in Figure 8, with NMOS as an example. The fabrication starts from Si doping. Depending on which parts of the design need to be camouflaged, regions on a wafer to be used for the channels of 'always-on' FETs are first defined. Then, a heavily doped but shallow ($< 10\,\text{nm}$) region is formed by ion implantation. The doping is performed on a slightly large area, which should ensure that the whole channel is covered while the neighbor FETs are not affected. After this, all the processing steps follow the same flow as regular fabrication (Figure 8).

3.3.2 Always-off transistor

To set a transistor as always-off, 'dummy contacts', as proposed in [RSSK13], can be applied on the gate and source terminals. In this approach, the contacts connecting the transistor terminals to metal layer 1 are interrupted by insulating thin films. Note that in dummy contact-based camouflaging, the insulating layer causing an electrical disconnect is applied on several contacts in the cell. For covert gates, we only require an insulating layer on the gate and source terminals. Due to the insulating film, the bias that turns on the transistors cannot be effectively applied on the gate. Further, the insulating layer on the source terminal cuts the transistor connection to VDD (for $NAND$-type gates)

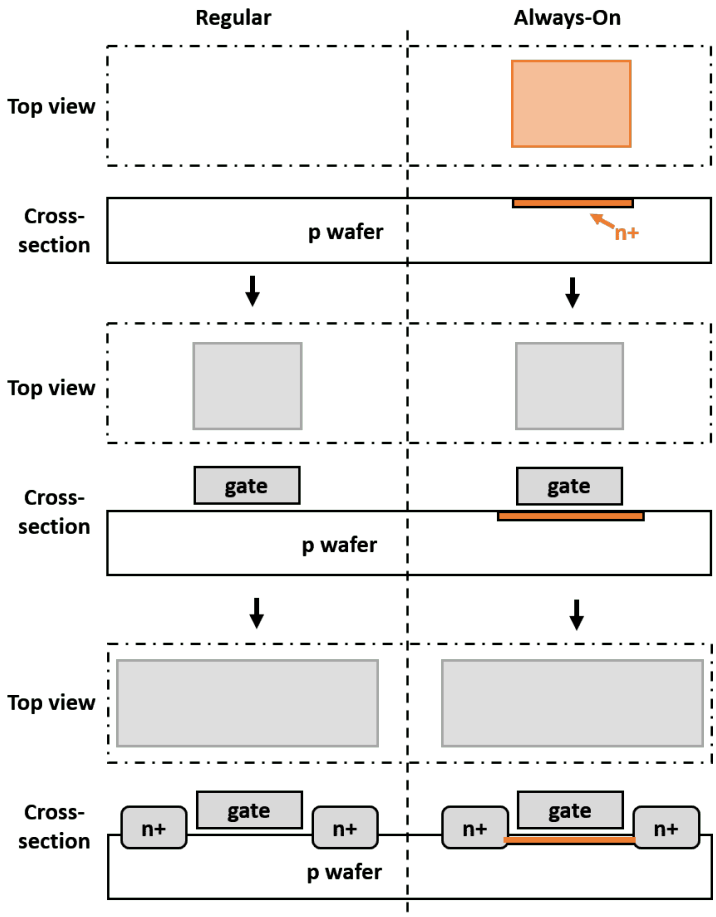


Figure 8: The fabrication of regular (left) and 'always-on' transistors (right) by modifying the doping in the channel regions. The mask used for each step is shown in top view, while the sample obtained after each step is shown in cross-section.

and GND for NOR-type gates. As a result, a strong inversion layer (i.e. the conducting channel) is prevented from forming, even with bias applied, and the transistors always stay as 'off'. The fabrication of such 'dummy-contacts' is similar to the forming of an electrical discontinuity in the interconnect, as proposed in [RFBL18]. As shown in Figure 9(b), regular contacts are first fabricated. Then, the dielectric material (e.g. SiO_2) of the 'always-off' transistor is etched to open an orifice for metal filling. Before the metal filling, a conformal thin layer of the dielectric material (SiO_2) is coated in the orifice. Given sufficient thickness, the dielectric thin film serves as a 'stopper' to prevent the switching-on of the 'always-off' transistor over the frequency range that the chip is supposed to work on.

3.4 SEM Imaging Attack Analysis

In covert gates, we always make one MOS transistor as 'always-on' and the complementary one as 'always-off' for one or more inputs. Thus, the adversary needs to find out at least one type of modification (heavily doped channel or dummy contact) to locate the covert gates. Considering that the materials exposed on the surface (typically the drain, source, and contacts) are similar to regular transistors, imaging-based recognition of the doped channel relies on passive voltage contrast (PVC). SEM sample preparation (i.e., removing

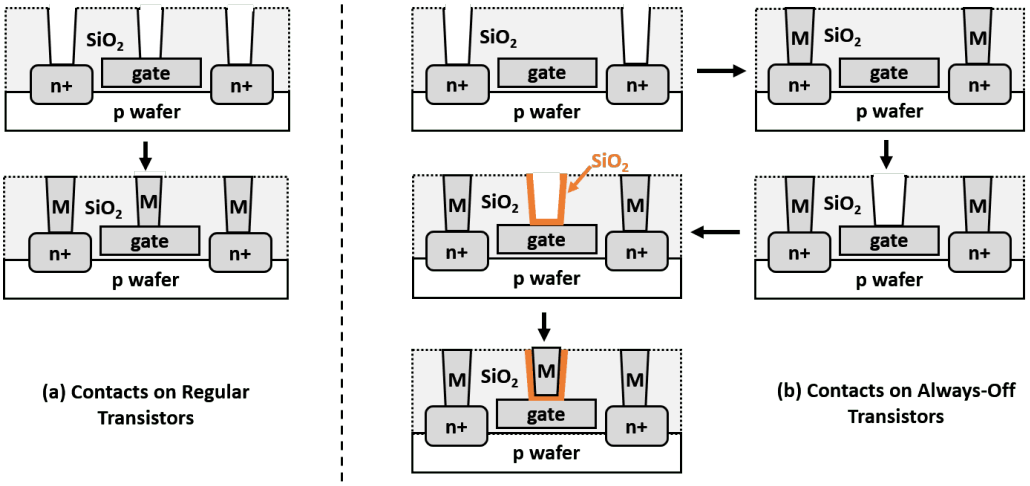


Figure 9: The fabrication of contacts on (a) regular and (b) "always-off" transistors.

materials to expose the active layer) can be performed in two ways: from the backside, or the front side of the IC. As discussed in Section 2.2, PVC imaging mainly depends on the surface charging effect. If the imaging is done from the backside, the contacts, metal layers, and vias, which make various connections among MOSFETs, are preserved. Therefore, the conducting volume corresponding to each region will be different and the charging of each region varies as well. It is thus very difficult to correctly extract modified doping information from the backside. Thus, front-side imaging would be preferred for detecting doping changes.

When imaging is performed from the front side, an ideal sample preparation procedure should expose the silicon surface (i.e., remove all the contacts, gates, and oxides) while leaving enough of the doped region, as shown in Figure 10. When the e-beam is scanning across a PMOS transistor, negative charges (electrons) accumulate in the p-type source and drain, and n-type well. On the other hand, for an NMOS transistor, the negative charges accumulate in the n-type source and drain [SSF+14]. Positive charges (electron escaping) can be analyzed in a similar way. When comparing regular and "always-on" transistors, the shorted channels should contribute to the conduction volume change. If such a change is significant (e.g. as in the devices designed in [BRPB13]), it should help an attacker to detect the modification. However, in the proposed "always-on" PMOS transistor, the shorted channel takes a volume from the n-well, without changing the total conduction volume (gray region in Figure 10a and c). For the proposed "always-on" NMOS, the shorted channel is an additional conduction volume and thus increases the total volume (gray regions in Figure 10b and d). However, compared to the thickness of source and drain (typically few tens of nanometers), the thickness of the shorted channel (controlled within a few nanometers by low energy ion implantation) is quite limited, resulting in the conduction volume increasing by less than 10%. This increase is comparable to processing variations and should be very difficult to be distinguished by PVC-based imaging.

3.5 SEM Imaging on Fabricated Devices

To demonstrate the indistinguishability of the proposed devices, we fabricated devices with different feature sizes. The SEM images are taken with Tescan LYRA-3 FESEM. As shown in Figure 11, the sources, drains, and channels of regular and 'always-on' PMOS transistors are fabricated on one wafer. In each set, there are transistors with channel lengths varying from 90 nm to 20 µm. The channel width is two times the length. All source and drain

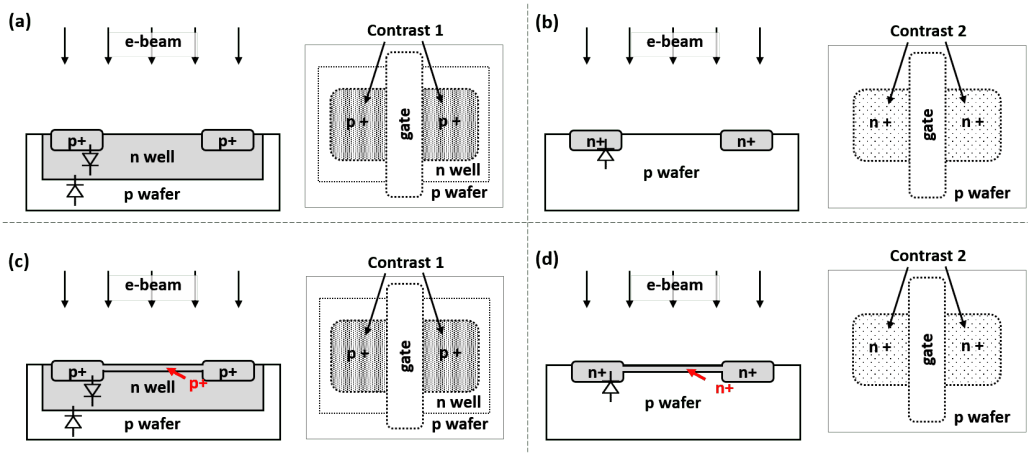


Figure 10: Due to the asymmetricity of the p/n interface (indicated by diodes on interfaces), the effective conduction volumes (grayed on the cross-section views) vary between PMOS and NMOS. However, the change between regular PMOS (a) and always-on PMOS (c) is insignificant. The same is the case with regular NMOS (b) and always-on NMOS (d).

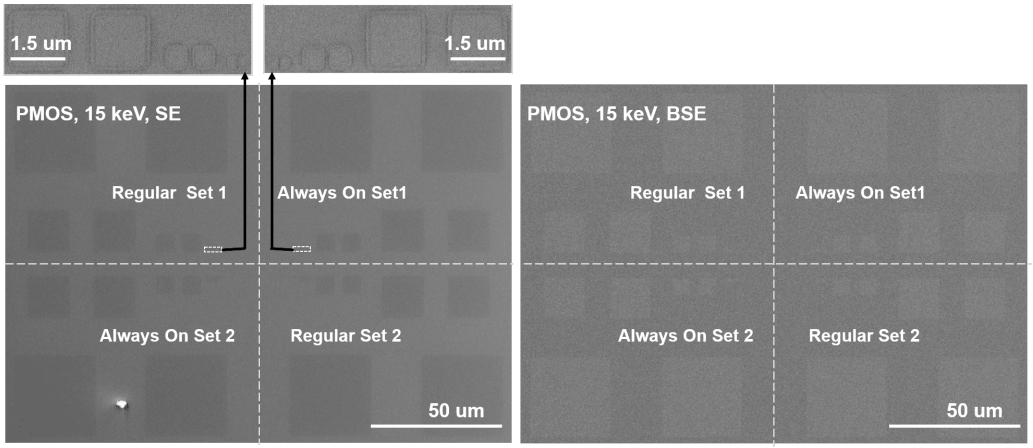
regions are defined in the square shapes. As described in Section 3.3, the channels of the 'always-on' transistors are first doped by ion implantation (boron, 2 keV, $5 \times 10^{13}\,\mathrm{cm}^{-2}$) to achieve a hole concentration of $1 \times 10^{19}\,\mathrm{cm}^{-3}$ within 5 nm surface layer, providing more effective charge carriers than a strong inversion layer in PMOS. Subsequently, the source and drain of both regular and 'always-on' transistors are doped by ion implantation (boron, 15 keV, $1 \times 10^{16}\,\mathrm{cm}^{-2}$), giving doping concentration about $1 \times 10^{20}\,\mathrm{cm}^{-3}$ to $1 \times 10^{21}\,\mathrm{cm}^{-3}$ within 100 nm surface layer. To identify the transistors, various e-beam energies (800 eV - 15 keV) are used. Different detectors, including secondary electron (SE) detector and back scattered electron (BSE) detector are also employed. As shown in Figure 11a - 11d, the SE images taken with lower e-beam energies give better doping-based contrast, which is consistent with previously reported studies [EBH02, Che16, SSF+14]. On the other hand, for BSE images, the effective signal is mainly from high energy electrons that are back-scattered by the nuclei of the atoms in the sample material [GNM+17]. Although the doping concentration is high, Si atoms still dominate the number of atoms in the sample. Therefore, the doped regions can barely be recognized on high keV (>10 keV) BSE images, and they are invisible on low keV BSE images. To obtain high quality SEM images with 800 eV (low energy), the sample has to be very close to the e-beam column (<6 mm). Since the BSE detector on our SEM system does not allow for this distance, 800 eV BSE images are not included. However, under all SE and BSE imaging conditions we tried, no difference between the regular and 'always-on' transistors was observed on the SEM images.

Regular NMOS and 'always-on' NMOS transistors are also fabricated in a similar fashion, by using phosphorus as the dopant (5 keV, $5 \times 10^{13}\,\mathrm{cm}^{-2}$ for modified channels; and 40 keV, $1 \times 10^{16}\,\mathrm{cm}^{-2}$ for sources and drains). The imaging results are the same as that for PMOS: the modified doped channels cannot be distinguished from the regular ones by visual inspection (Figure 12).

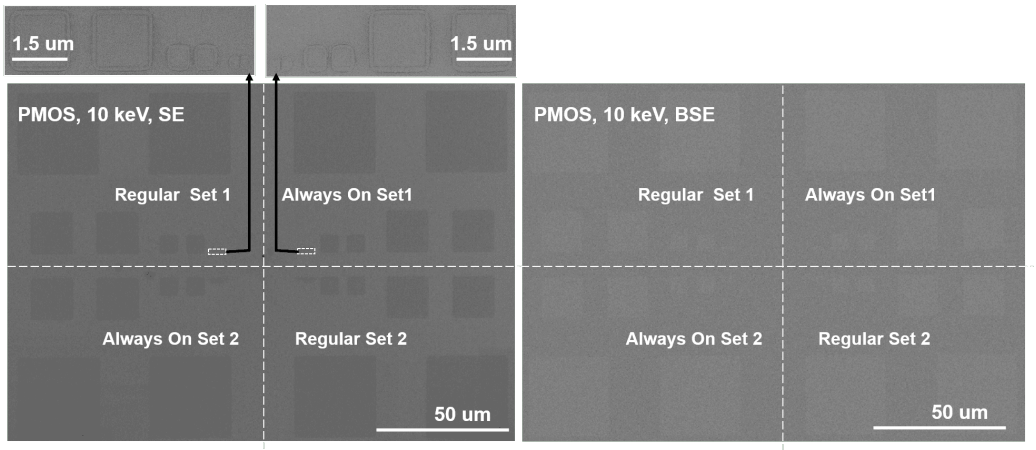
For the 'always-off' transistors, simplified metal structures with silicon oxide are fabricated to analyze the difference in imaging between regular contacts and dummy contacts. Due to fabrication facility limitations, we used gold as the metal material, which is easily deposited without getting oxidized during fabrication (which became an issue for copper). For the regular contacts, the thickness of the gold is 200 nm. For the dummy contacts, a stacked structure of gold-100 nm/SiO_2-10 nm/gold-100 nm was

prepared. The diameter of the contacts and the dummy contacts varies from 100 nm to 2 µm. Since low keV SE imaging is PVC sensitive while high keV BSE imaging is sensitive to materials difference, 800 eV SE image and 15 keV BSE images were taken from the contact sample, as shown in Figure 13. From the SEM images, we can see that no difference is observed between the structures representing regular contacts and dummy contacts. This is because the contrast is mainly dependent on material differences (metal vs SiO_2). The charging that may affect the contrast (and thus, imaging results) in both structures mostly comes from the surrounding SiO_2, instead of the metal and/or SiO_2 between the metal contacts. Therefore, any structural difference between regular and dummy contacts cannot be observed.

We also used FIB to detect the channel doping used for creating the always-on transistor. However, we noticed surface damage and lower resolution with the FIB, resulting in very poor imaging at nodes below 300 nm. This is also in line with the observations in [SSF+14], where the contacts on top of the dopant-varying regions were only partially distinguishable when observed with FIB i.e., results were worse than with SEM.

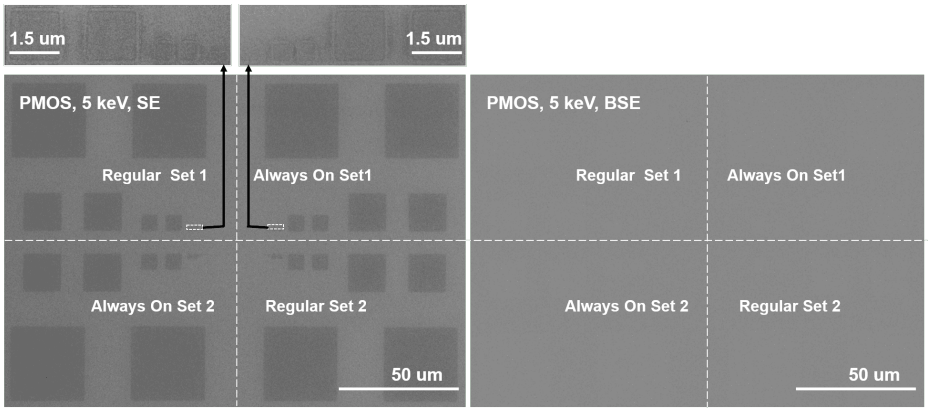


(a) 15 keV

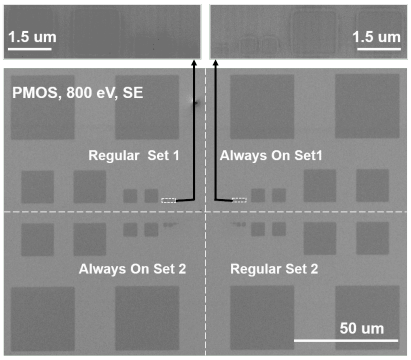


(b) 10 keV

Figure 11: The doped regions of PMOS transistors for regular and 'always on' cases are imaged by SE and BSE detectors. The e-beam energy varies from 15 keV to 800 eV. There are 2 sets of transistors with channel lengths varying from 90 nm to 20 µm for each case.

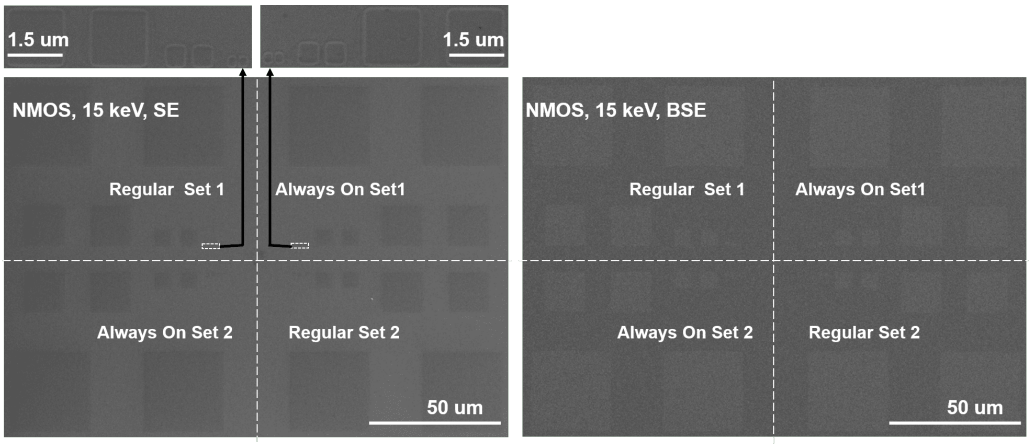


(c) 5 keV



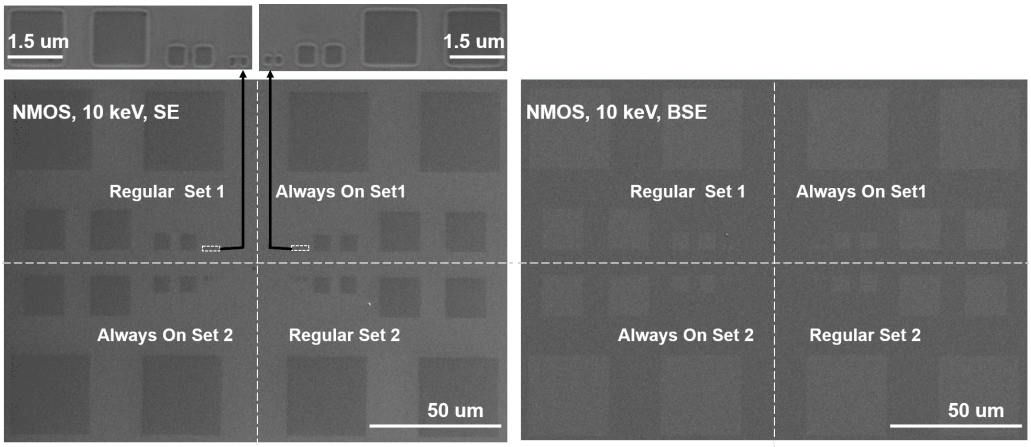
(d) 800 eV

Figure 11: The doped regions of PMOS transistors for regular and 'always on' cases are imaged by SE and BSE detectors. The e-beam energy varies from 15 keV to 800 eV. There are 2 sets of transistors with channel lengths varying from 90 nm to 20 µm for each case. (continued..)

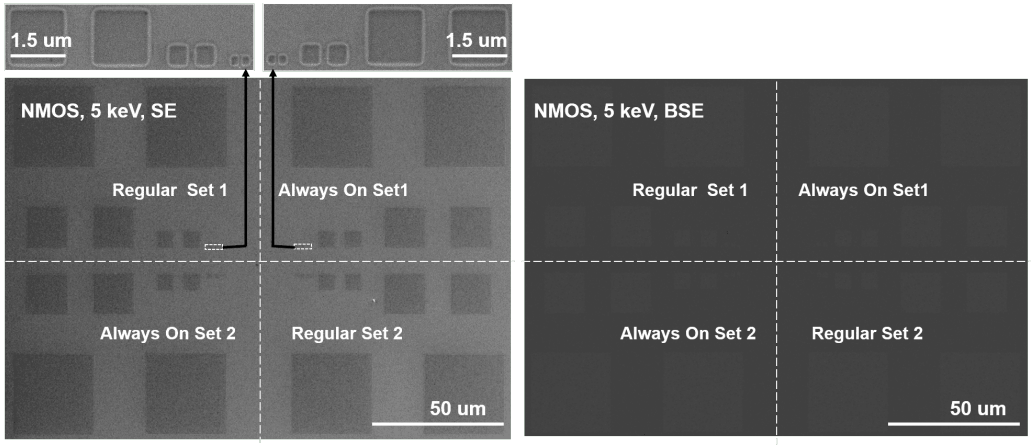


(a) 15 keV

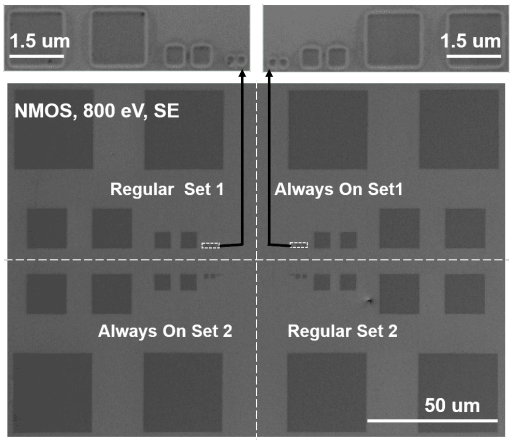
Figure 12: The doped regions of NMOS transistors for regular and 'always on' cases are imaged by SE and BSE detectors. The e-beam energy varies from 15 keV to 800 eV. There are 2 sets of transistors with channel lengths varying from 90 nm to 20 µm for each case.



(b) 10 keV



(c) 5 keV



(d) 800 eV

Figure 12: The doped regions of NMOS transistors for regular and 'always on' cases are imaged by SE and BSE detectors. The e-beam energy varies from 15 keV to 800 eV. There are 2 sets of transistors with channel lengths varying from 90 nm to 20 µm for each case. (continued..)

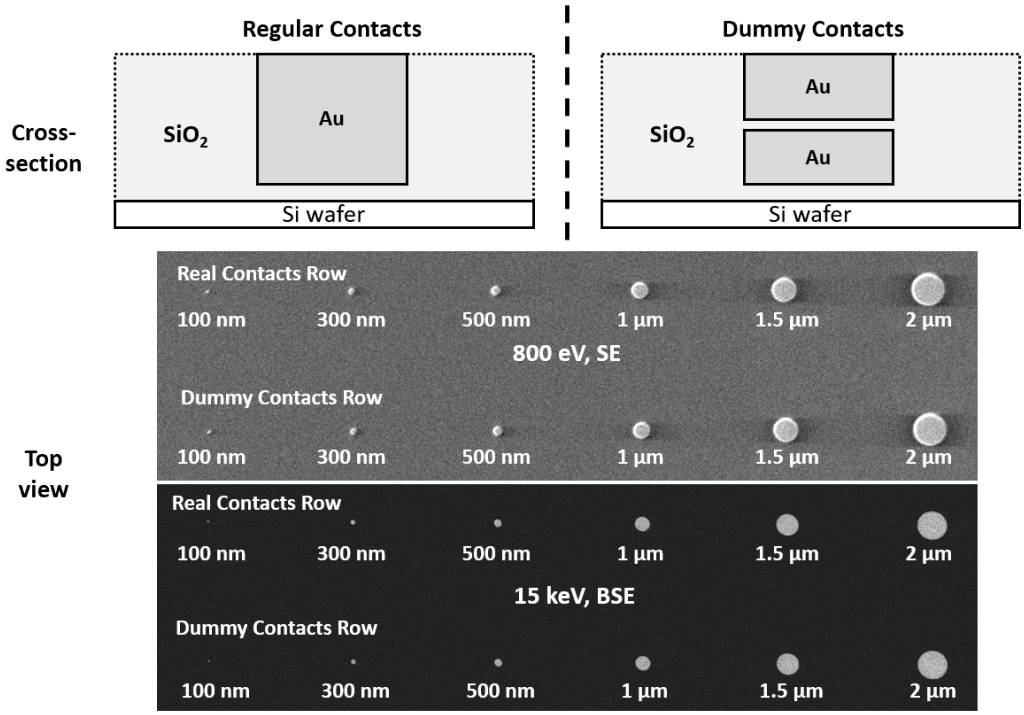


Figure 13: Contacts and dummy contacts for regular and 'always-off' transistors under SE and BSE imaging conditions.

3.6 Cell Modeling

In order to characterize the power and performance characteristics of the covert cells, we developed a simplified circuit model for the always-on and always-off transistors, which is explained below and also shown in Figure 14.

- **Always-on**: The always-on transistor's behavior is replicated with the help of depletion mode transistors. Depletion mode devices are identical to enhancement mode devices (i.e., regular NMOS/PMOS devices), except in one regard: in regular enhancement mode devices, the channel between the source and drain terminals is formed once an appropriate gate voltage is applied. In contrast, depletion mode devices come with 'pre-made' channels, created through deliberate ion implantation. When a negative gate bias is applied, the channel stops conduction (for NMOS). In our case, a negative bias is never applied on the always-on transistors (for NMOS). Therefore, regardless of the logic level applied to the input, the transistor is always on.

- **Always-off**: The always-off transistor is modeled with the combination of a regular NMOS/PMOS device and a parallel plate capacitor in series with the dummy input pin and the gate input, as well as between the source terminal and GND (for NOR-type gates, with the always-off transistor in the pull-down network). The capacitor models the metal-insulator-metal (i.e., Tungsten-SiO_2-Tungsten) stack, with the capacitance approximated by:

$$C = \frac{\varepsilon_{SiO_2} WL}{t_{SiO_2}} \tag{1}$$

Here, ε_{SiO_2} is the permittivity of SiO_2, W, L are the width and length of the contact, and t_{SiO_2} is the thickness of the insulating SiO_2 layer.

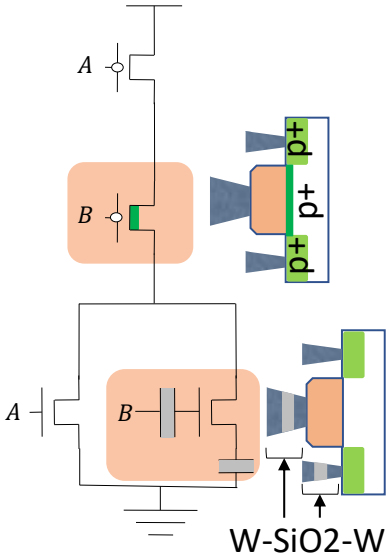


Figure 14: Modeling the always-on transistor (above) and always-off transistor (below). W represents the tungsten metal contact, $p+$ is the heavily doped diffusion region and SiO_2 is the insulating layer on the gate and source terminals, modeled by a parallel plate capacitor.

Note that we do not put an insulating layer between metal 1 and the drain terminal, as the two NMOS transistors share the drain terminal (and thus, the drain contact). If an insulating layer was put on the drain contact, the normal NMOS transistor's drain terminal would also be cut off, which is not desired. Also, it should be noted that this model is similar to that of a floating gate transistor, but without a separate voltage source and additional capacitance to model tunneling [HMD99]. This is because the applied voltage levels in a logic gate do not reach the high voltages required for tunneling (e.g., $> 9V$). However, in our simulations, we do include a very high resistance in parallel with the capacitors, as SPICE simulators do not handle floating nodes.

Using this simplified circuit model, we performed SPICE simulations on 2 input NAND and NOR gates (with one of the inputs being dummy), using $90nm$ device parameters from [GBW+09] and [ZC07]. Delay overheads were obtained by averaging $max(t_{phl}, t_{plh})$ over all possible input patterns. Dynamic power was estimated with an average of (fall power + rise power), and static power was measured by monitoring the current when the output of the gate was static over several $1ns$ windows. From the results in Table 2, we can see that the covert NAND and NOR gates have $\leq 1.82\times$ delay compared to a regular inverter. Since we do not rely on any special layout, the overheads are significantly better than camouflaged gates based on only dummy contacts or threshold voltage variations. For example, the power, delay and area overheads for a camouflaged $NAND$ gate that could either be a $NAND$, NOR or XOR were $5.5\times, 1.6\times$ and $4\times$ respectively [RSSK13]. It can also be seen that the power consumption (both dynamic and static) of the covert $NAND$ and NOR gates are much lower than that of an inverter. This is because the inverter (INVX1) considered from [GBW+09] has a much larger W/L ratio than the $NAND2X1$ and $NOR2X1$ gates in the same library.

We also performed SPICE simulations on the gates using sizing dimensions from $NAND2X2$, $NOR2X2$ and $INVX0$ gates in the SAED90nm library. This was done in order to compare the overhead of the covert $NAND$ and NOR gates to a minimum-sized inverter (the INVX0 cell in the library). The results in Table 3 show that the covert

$NAND2X2$ and $NOR2X2$ consume more static and dynamic power than a minimum-sized inverter, while being faster than the inverter. However, the overheads of the covert gates are still very similar to that of regular $NAND2X2$ and $NOR2X2$ gates.

Table 2: Comparison of area, delay and power characteristics of covert gates with unit strength standard cells. Note that we are mostly concerned with comparing 2-input covert gates to inverters which are being replaced in our experiments. However, similar experiments can be performed by comparing 3-input covert gates with 2-input standard cells, and so on.

	Regular Gates			Covert Gates			
				NAND2X1		NOR2X1	
	NAND2X1	NOR2X1	INVX1	Value	Comp. with INV	Value	Comp. with INV
Area (μm^2)	5.53	6.45	6.45	5.53	**0.86X**	6.45	**1.00X**
Delay (ps)	33.59	50.70	26.84	36.00	**1.34X**	48.75	**1.82X**
Dynamic Power (μW)	341.22	191.24	352.20	254.91	**0.72X**	244.24	**0.69X**
Static Power (nW)	80.75	172.62	143.72	31.11	**0.22X**	38.68	**0.27X**

Table 3: Comparison of area, delay and power of regular NAND2X2, NOR2X2 and INVX0 cells with camouflaged NAND2X2 and NOR2X2 cells.

	Regular Gates			Covert Gates			
				NAND2X2		NOR2X2	
	NAND2X2	NOR2X2	INVX0	Value	Comp. with INVX0	Value	Comp. with INVX0
Area (μm^2)	9.22	9.22	5.53	9.22	**1.67X**	9.22	**1.67X**
Delay (ps)	21.49	30.82	48.07	21.29	**0.44X**	32.85	**0.68X**
Dynamic Power (μW)	692.45	386.76	172.52	515.05	**2.99X**	499.28	**2.89X**
Static Power (nW)	403.35	785.21	3.69	117.95	**31.96X**	289.98	**78.58X**

3.7 Netlist Integration

The NOR and $NAND$ covert gates can be integrated into a gate-level netlist by replacing an N-input gate with an $(N+n)$ input covert gate of the same type (or by replacing inverters with 2-input covert $NAND$ or NOR gates). While performing the gate replacement, we can follow a variety of metrics such as insertion into locations which are likely to corrupt the output more [RSSK13, RZZ+15], insertion into locations with flipped signal probability [DBDN+14, SAFT16] and insertion into locations such that multiple gates interfere with each other [RSSK13]. Such metric-guided insertion techniques have been explored quite heavily and can be chosen by the designer based on their area/delay/power budget and threat model considered. However, note that the use of these metrics may also help an attacker to narrow down possible covert gate locations in the netlist. Thus, it should be ensured that a sufficient number of other regular gates in the design also have the same metric measure (e.g., static probability difference) as the covert gate.

In this paper, we follow the simple approach of random insertion but with one important constraint: the net to be connected to the dummy input must not form a combinational feedback loop. This is because such loops normally cause undesirable oscillatory behavior in digital circuits. Although these nets would not affect the circuit functionality (since the pins are dummy), the attacker would easily be able to identify them in the design, as they violate design rules. Valid (and invalid) net selection examples are illustrated in Figure 15.

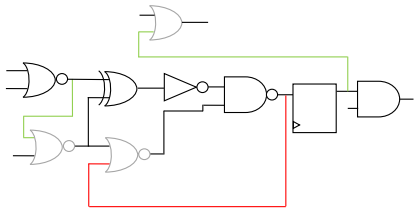


Figure 15: Valid nets (marked in green) for the dummy inputs of the covert gates. Invalid nets are shown in red. Note that sequential feedback loops are valid, as loops are common in sequential circuits with state.

4 SAT Attack Analysis

4.1 SAT formulation

In Section 2.4.2, we discussed how previously proposed camouflaged gates based on dummy contacts and varying functions could be resolved by a SAT-based formulation. Here, we analyze how an attacker might attempt to take a similar approach to resolve dummy pins on candidate covert gates and why it scales poorly. Using a MUX-based selection network, the solver would return 'key bits' to decide the correct permutation of pins on each candidate gate. For a candidate covert gate G with N pins, the total number of valid permutations P would be given by:

$$P = \sum_{i=1}^{n} \binom{N}{n} \text{ , where } 1 < n < N \tag{2}$$

A pin permutation network to transform a covert gate with function F and N pins (but an unknown number of dummy pins) is shown in Figure 16(a). Here, the total number of choices is denoted by P, and the length of the key K is given by

$$|K| = log_2(P + 1) \tag{3}$$

An example of the choices in the permutation network is shown in Figure 16(b), where a 3-input NAND gate can have 7 possible choices. The number of choices is calculated with Equation 2 as:

$$\binom{3}{1} + \binom{3}{2} + \binom{3}{3} = 7 \tag{4}$$

Further, the length of key K to resolve the 7 choices would be $log_2(7 + 1) = 3$. The pin permutation for NOR, AND, OR and XOR gates can also be determined in a similar manner.

Alternatively, each pin of a suspect gate can also be replaced by a MUX, whose select line is controlled by a key bit and chooses between either the pin or a non-controlling value of that gate. Thus, the SAT attack would choose the pin if it is an actual pin. Otherwise, a non-controlling constant value would be chosen for the pin in order to cancel it out (i.e., render it dummy). Both formulations in help in identifying the dummy pins in a potential covert gate. Section 6.1 compares these two formulations, with respect to SAT attack time and iterations needed to solve the identity of potential covert gates.

4.2 SAT Attack Difficulty

The main difficulty in using SAT attacks to resolve the camouflaged gates would arise from:

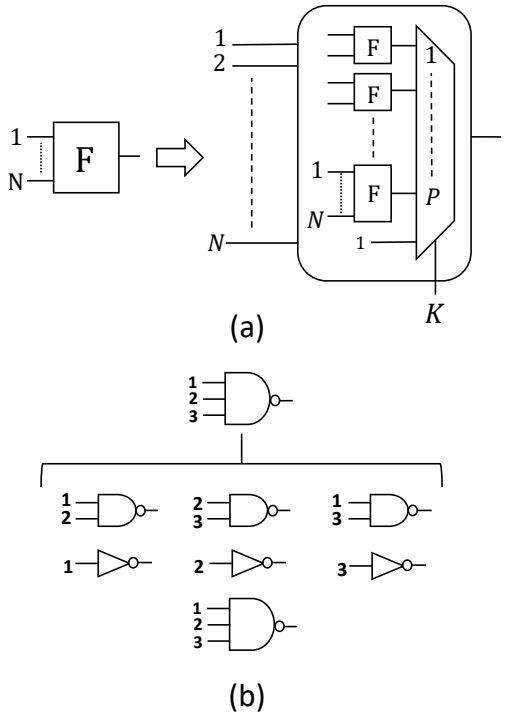


Figure 16: (a) Transforming a camouflaged gate with an unknown number of dummy inputs into a pin permutation network to be solved by SAT. (b) A 3 input NAND gate and its 7 possible candidates.

1. The number of suspect gates that need to be encoded by a pin permutation network,
2. The number of pins N on each suspect gate.

Both factors lead to an increase in the size of the overall key K that needs to be solved by the attack. Clearly, by Equation 2 and 3, we can see that the number of choices P (and thus, the key length $|K|$) to encode an individual gate has a proportional dependency on the number of pins N it possesses. However, a standard cell library used to synthesize a netlist seldom has gates with pins in excess of 5 (i.e., $N < 5$), since there is a heavy delay penalty associated with high fan-in gates. Thus, resistance to SAT attacks is mainly dependent on how many gates the attacker has to encode using a pin permutation network and feed into the solver. In this regard, the defender using the proposed covert gate has a clear advantage: since the attacker does not know which gates are covert gates and which aren't (this was demonstrated by imaging results in Section 3.5), *he or she needs to encode every single gate.* Even under simplified assumptions such as only $NAND$ gates being suspect, the attacker would have to encode every single NAND gate in the design with a pin permutation network.

5 Test-based Attack Analysis

As discussed in Section 2.4.1, standard test generation tools can be adapted to sensitize camouflaged cells to chosen input values and propagate their responses to the outputs. This allows an attacker to decide the Boolean function of the logic gate by comparing the obtained response against that of a functional chip. In a similar fashion, the pins

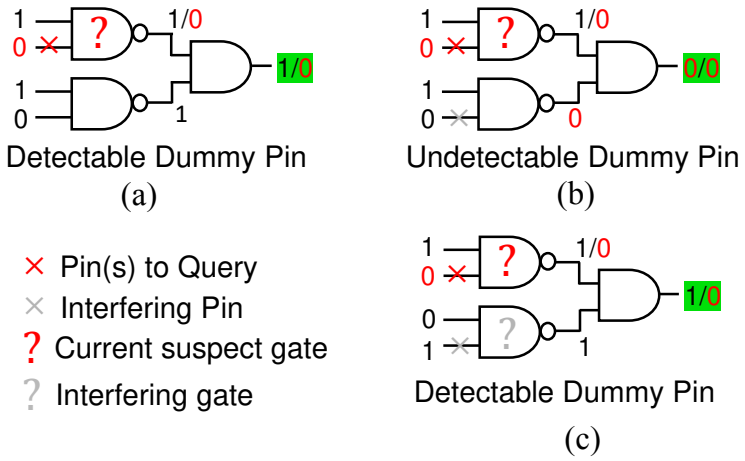


Figure 17: (a) A detectable dummy pin which can be sensitized and propagated to output. (b) Undetectable dummy pin whose output is contaminated by another dummy pin. (c) Detecting dummy pin in the presence of an interfering dummy pin.

of the proposed covert gates can also be queried with chosen inputs and their responses can be analyzed to decide which of the pin(s) (if any) are dummies. Similar to regular camouflaging, the attack can be performed in two steps:

1. **Sensitize**: For each suspect pin of a (potentially) camouflaged gate, sensitize the pin to a 'controlling value'. A controlling value forces the output of the logic gate to a known value, regardless of the values applied on the other pins. For example, the controlling value for AND/NAND gates is 0, since a logic 0 applied on any one of the pins of a AND/NAND gate forces the output of the gate to 0/1. Here, the rationale is that a controlling value applied on a dummy pin will not affect the output of the gate. If the pin is, in fact, real, the controlling value should force the output of the gate to a known value, which can then be observed and compared against the known-good response from the functional chip.

2. **Propagate**: Apply 'non-controlling values' to (i) all other pins of the gate currently being tested, and (ii) to all other pins in the path between the queried pin and an observe-point such as a primary output or scan flip-flop. This ensures that the effect of the controlling value on the pin being queried is observable (i.e., affects the output) and can be checked against the response of a functional IC.

These two concepts are illustrated in Figure 17 (a). In this example, we are concerned with generating a test to detect the (potentially) dummy pin marked by ×. on the NAND gate. The attack proceeds as follows.

- Set the × pin to a controlling value of 0.
- Set the other pin of the $NAND$ gate to a non-controlling value of 1.
- Set the net connected to the other pin of the AND gate to a non-controlling value of 1.
- If the suspect pin is dummy, the circuit output should be 0; else, the output should be 1, since the suspect $NAND$ gate behaves as an inverter.

- This generated pattern $< 1, 0, 1, 0 >$ can now be applied on the functional IC to confirm if the pin is indeed dummy i.e., if the obtained output is 0, the pin is indeed dummy (and vice-versa).

However, note that this type of attack has its limitations. Consider the case in Figure 17(b). Here, we can see that the attacker is trying to discern the pin marked by ✕. However, the pin marked by ✕ could also be dummy. In this case, we see that there is no difference between the output values if the pin ✕ is (or is not) dummy. Thus, the pin location ✕ cannot be discerned in the presence of another dummy pin ✕.

In other cases, it would be possible to generate an input vector such that pin ✕ is detected, even in the presence of ✕. This scenario is shown in Figure 17(c), where a non-controlling value of 1 is applied on the pin ✕. Here, we see that the output is 0 if pin ✕ is dummy and 1 if it isn't.

5.1 Adapting ATPG Tools for the Attack

In order to execute an attack, one can use a commercial automatic test pattern generation (ATPG) tool such as Tetramax, and adopt its stuck-at fault test generation mode for the attack. A stuck-at fault is a VLSI fault model in which it is assumed that a net is permanently stuck at a logic value of either 1 or 0 due to manufacturing defects. Once a stuck-at fault location is set, the tool generates an input vector that can simultaneously sensitize the net to the opposite logic value of the stuck-at fault [1] and propagate it to an observable point. Note that this is exactly the 'sensitize and propagate' technique that we outlined in our explanation of test-based attacks in Section 5. The only difference is that in order to sensitize a pin to its controlling value 0, we set a stuck-at-1 fault on the pin. Similarly, for sensitizing a pin to 1, we set a stuck-at-0 value on the pin.

While generating the input vectors, one also wants to make sure that scenarios such as the one in Figure 17(b), where two or more dummy pins interfere with each other to make the output change un-observable, do not happen. In order to do this, one might set a controlling value to a suspicious pin, and generate non-controlling values on all other suspect pins in its path. However, it might not always be possible to test a suspect pins while setting all other suspect pins to non-controlling values.

These two scenarios also help to explain why test-based attacks might not be able to defeat covert gates. Scenarios such as (b), where erroneous patterns are generated, would happen if one covert gate is in the path of another covert gate. When a sufficient number of covert gates are inserted, this scenario is likely to occur frequently, resulting in the undetectability of dummy pins. Patterns to correctly sensitize and propagate the value on the potential dummy pin without interference - as shown in (c) - would also become harder to generate. This is because an ATPG tool might not always be able to assign non-controlling values to every pin on the required path, leading to ATPG untestable faults (i.e., undetectable dummy inputs). Moreover, note that both these issues would be exacerbated if several or all types of gates, or multiple pins on multiple gates, are suspect. We discuss these various scenarios, where an attacker has varying amounts of information on covert gates, in Section 6 and Figure 18. Regardless, we find, through our results, that test-based attacks only work when the attacker has the most amount of information on covert gates; it is quite limited in realistic scenarios where limited information is able on covert gate/pin identities.

[1] This is done so that if the fault is indeed present in the manufactured chip, applying the generated input pattern will trigger the faulty output.

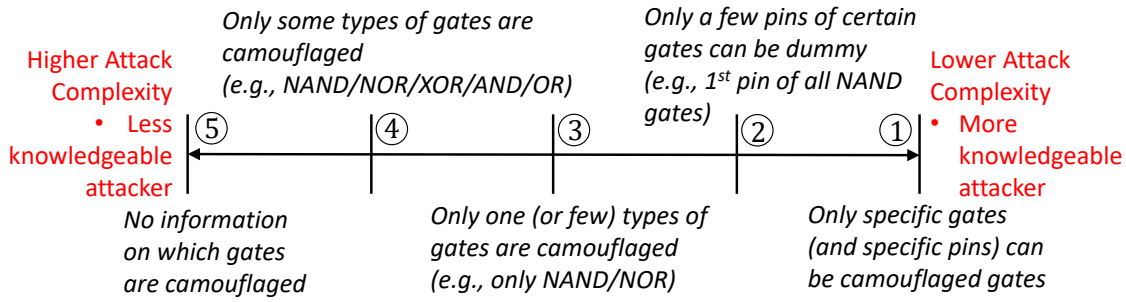


Figure 18: Varying range of available information to the reverse engineer changes de-camouflaging complexity

6 Experimental Results and Discussion

The basis of our proposed technique is that the camouflaged gates look no different from regular logic gates in the design. However, there might be scenarios where an adversary has more knowledge about the design or covert gate constraints. For example, they might know (through an insider in the design house) that only some types of gates can be camouflaged (e.g., only NAND and NORs). In other scenarios, they might realize that only gates in non-critical paths of the design could be covert (assuming they know the operating frequency of the design). Conversely, the design house might also want to perform security analysis under similar kinds of restrictions to get a sense of how various attacks scale. Figure 18 shows a spectrum of attack models (stronger to weaker), highlighting these different scenarios and how more (or less) information about the covert gate technique can lead to lower (or higher) attack complexity. With this spectrum in mind, we conducted both SAT and test-based attacks on the camouflaged netlist of various benchmark circuits.

6.1 SAT Attack Results

For SAT based attacks, we considered scenario ③ from Figure 18, where the reverse engineer knows that only a few types of gates can be camouflaged. For the benchmark circuits in Table 5, which are from the ISCAS'85 [Bry85] and EPFL suite [AGDM15], we considered 2 to 4 input $NAND$, AND, OR, NOR gates with their respective pin permutation networks, as shown in Figure 16. The distribution of these gates in the benchmark circuits is shown in Table 4. For regular camouflaging, we allocated 5% of the gates ($NAND/NOR/XOR$), which is line with prior work [RSSK13] and is also a realistic amount, given the high overheads of regular camouflaging approaches. After constructing the encoded circuits, we used the SAT attack platform from [SRM15] to collect results on the number of iterations and time required to find the correct camouflaged gate assignments. The number of key bits needed to encode all the gate choices as a pin permutation network is also listed in Table 5. Timeout for the attack was set to 12 hours, which is in line with similar work [SRM15, XS16]. Note that the size of the benchmarks we used for the SAT attack experiments are quite small. This is because the SAT attack platform, in its current form, can only handle combinational circuits. In reality, the attack would be launched on sequential circuits with scan access. This would allow the attacker to treat the large sequential circuit as a collection of much smaller combinational logic cones (CLC). The benchmarks are representative of these smaller CLCs.

Since our camouflaging technique makes it impossible for the attacker to discern the candidate gates, he or she needs to replace *all* gates in the design and encode them.

Table 4: Distribution of the number of 2, 3 and 4 input AND/NAND/OR/NOR gates in the combinational benchmark circuits used for SAT attack evaluation.

Benchmark	2 Inputs		3 Inputs		4 Inputs	
	AND/NAND	OR/NOR	AND/NAND	OR/NOR	AND/NAND	OR/NOR
C1908	377	1	13	0	4	0
C2670	457	63	112	2	11	22
C3540	684	60	93	83	17	1
C5315	773	114	359	56	27	63
C7552	1562	220	146	20	64	34
arbiter	11839	0	0	0	0	0
voter	13758	0	0	0	0	0

Table 5: Comparison of SAT attack resiliency between regular camouflaged gates and proposed covert gates. Timeout is set to 12 hours for regular SAT attack, and default parameters from [SLM+17] are used for AppSAT with a timeout of 2 hours. We did not apply AppSAT to designs that were already broken by regular SAT attack (marked by N/A). This is because AppSAT should be used only when regular SAT does not succeed. Attack time and attack iteration data given in between braces '[ ]' indicates the results obtained by encoding each pin of a suspect gate with a MUX that selects either the pin or the non-controlling value of the gate.

Benchmark	Gate / Node Count	Regular Camouflaging				Proposed Camouflaging			
		5% of NAND/NOR/XOR				NAND+NOR+AND+OR			
		|K|	Attack Time (s)	# Attack Iterations	AppSAT Time (Hrs)	|K|	Attack Time (Hrs)	# Attack Iterations	AppSAT Time (Hrs)
C1908	880	**34**	0.55	7	N/A	**811**	3.52 [5.91]	235 [191]	N/A
C2670	1193	**26**	0.65	11	N/A	**1514**	Timeout [Timeout]	2127 [4891]	Timeout
C3540	1669	**28**	0.68	11	N/A	**2088**	Timeout [Timeout]	28 [34]	Timeout
C5315	2307	**46**	3.58	25	N/A	**3379**	Timeout [4.27]	240 [459]	Timeout
C7552	3512	**106**	4.07	27	N/A	**4454**	Timeout [Timeout]	52 [91]	Timeout
arbiter	11839	**1182**	3815.00	855	N/A	**23678**	Timeout [Timeout]	82 [141]	Timeout
voter	13758	**1078**	Timeout	33	Timeout	**21560**	Timeout [Timeout]	51 [28]	Timeout

Therefore, the length of the key to solve our camouflaging technique is always going to be longer than that of regular camouflaging, where the gates are known in advance. Since the key length $|K|$ is longer, the overall search space for the solver also becomes larger, leading to higher attack run time. The results in Table 5 support this argument, as the SAT attack platform is unable to resolve most of the benchmark circuits in the allotted 12 hours. On the other hand, the SAT attack platform is able to resolve most of the circuits that have been camouflaged in a regular fashion (i.e., each camouflaged gate is either a $NAND$, NOR or XOR). These results show that as the attack is not successful even under a restricted notion (i.e., only four types of gates can be covert), it is not expected to scale under realistic scenarios where *any gate could be a potential covert gate*.

We also evaluated the benchmarks circuits using the recently proposed 'approximate SAT' or AppSAT attack algorithm [SLM+17]. For AppSAT experiment analysis, we set a timeout of 2 hours, as it should terminate early once it sees an improvement in error rate. Our results show that AppSAT fares no better than regular SAT when used on covert gates. This is expected because AppSAT is only suitable for hybrid obfuscation schemes, where a low corruptibility obfuscation technique is combined with a high corruptibility technique [SLM+17]. In the case of covert gates, AppSAT does not see a sufficient drop in the error rate to trigger early algorithm termination.

6.2 Test-based Attack Results

For conducting the test-based attack, we considered industrial-size sequential benchmark circuits with full-scan capabilities (i.e., all the flip-flops are configured to form a scan chain, as mentioned in Section 2.1). It should be noted that, similar to SAT attacks, the

Table 6: Results of test generation for detecting dummy inputs on various types of camouflaged gates.

Benchmark	Gate	Gate Count	Detectable		Undetectable		ATPG Untestable		Not Detected	
			#	%	#	%	#	%	#	%
b18 Primitive Count = 84,632 # Scan DFF = 3020 I/O = 40/24	$NOR2X$	2390	10	**0.42**	5	**0.21**	2373	**99.29**	2	**0.08**
	$NOR3X$	270	12	**4.44**	0	**0.00**	237	**87.78**	21	**7.78**
	$NOR4X$	195	17	**8.72**	0	**0.00**	114	**58.46**	64	**32.82**
	$NAND2X$	4194	7	**0.17**	30	**0.72**	4154	**99.05**	3	**0.07**
	$NAND3X$	2135	8	**0.37**	19	**0.89**	1849	**86.60**	259	**12.13**
	$NAND4X$	909	38	**4.18**	0	**0.00**	753	**82.84**	118	**12.98**

success of test-based attacks will vary based on the information available to the adversary (see Figure 18). In our experiments, we follow variants of scenario ② and ③, where we know that only a few types of gate (and their respective pins) can be camouflaged. More specifically, we considered different types of NOR gates present in the design (e.g., 3 input NOR gates $NOR3X$, 2 input NOR gates $NOR2X$) and assumed pin 1 of these gates could be dummy. For each suspect gate, we performed the following steps:

- **Force controlling value**: Set a stuck-at-0 fault on the first pin of the current NOR gate n. This forces the ATPG tool to generate input patterns that sensitize the pin to logic 1 (the controlling value for a NOR gate).
- **Constrain other pins**: Set a constraint of 0 on the first pin of all other NOR gates in the design. This forces the tool to only generate patterns that ensure 0 (non-controlling value) on these pins, since one or many of them could be dummies. This helps to avoid scenarios such as the ones shown in Figure 17, where the effect of the dummy pin becomes unobservable due to another unidentified dummy pin. In order to not over-constrain the pattern generation, we also make sure that the constraint is applied to only those NOR gates that fall in the fan-in or fan-out cone of NOR gate n.
- **Generate pattern**: Run ATPG and generate pattern for resolving the pin on gate n.
- **Repeat**: Repeat for all other pins.

For the $NAND$ gates, the steps are the same as above, except that we set a stuck-at-1 fault instead of stuck-at-0.

The results of the test generation procedure are shown in Table 6. In the table, 'Detectable' implies that the gate/pin in that location can be tested without any interference from other gates (i.e., it can be determined with certainty whether the pin on the gate is dummy or not). 'Undetectable' implies that a dummy pin placed on this location has no effect on the output i.e., it does not affect the circuit functionality. 'ATPG Untestable' implies that a test pattern cannot be generated to sensitize and propagate a controlling value on a potentially dummy pin (i.e., it cannot be determined whether the pin is dummy or not). Finally, 'Not Detected' implies that a test pattern to detect the pin could not be generated with the current ATPG tool effort level. Some of the observations from the table are as follows:

- Even when considering that only one type of gate is covert in the netlist, the number of gates for which a test vector to differentiate between a dummy and a real pin is generated is very low. This is shown under the column 'Detected'. Note that from a designer's perspective, these are the locations to avoid inserting a covert gate.
- From the table, we see that certain locations in the netlist result in undetectable faults. Therefore, an attacker misidentifying a camouflaged gate at this location gains an advantage, as the recovered netlist's output is not corrupted due to this

Table 7: Area, delay and power overhead estimates from integrating covert gates into benchmark circuits.

Benchmark	Area (μm^2)			Delay (ns)			Power (μW)		
	Covert	Original	%	Covert	Original	%	Covert	Original	%
aes	114098.90	113384.22	0.63	18.19	15.99	13.76	2689.2	2678.9	0.38
b12	9725.38	9646.59	0.81	2.98	2.88	3.47	154.9783	154.4319	0.35
b15	53432.06	53134.15	0.56	26.32	26.32	0.00	654.9308	657.4276	-0.38
b17	171193.62	170264.84	0.54	32.47	31.14	4.27	2015.7	2011.3	0.22
s35932	111402.38	111088.12	0.28	14.13	10.84	30.35	2290.2	2328.4	-1.67
s38417	107803.98	107349.70	0.42	20.84	16.69	24.87	1949	1949.6	-0.03
s38584	87647.35	87229.18	0.48	15.38	13.11	17.32	1572.1	1570.9	0.08

mis-identification. However, note that the percentage of undetectable gates is never above 1%. Further, they can be avoided during the netlist integration process.

- A large portion of the gates are either ATPG untestable or not detected i.e., even under the very restricted notion of *only one type of gate being camouflaged*, it is not possible to generate a test pattern that differentiates the gate's pin as dummy or real. This is either because the pattern cannot be generated due to the added constraints (which are required to obtain guaranteed-correct patterns - thereby causing the gate/pin fault to be ATPG untestable), or the ATPG tool is unable to generate a pattern with the current effort level set for the test generation algorithm (i.e., the gate/pin fault is not detected). Further, when multiple types of gates are camouflaged with multiple pins, the success probability (i.e., chance of generating an input pattern to detect the dummy pins) becomes even lower.

6.3 Netlist Integration Overhead

We also performed a set of experiments with various benchmark circuits to estimate the overhead from integrating covert gates. All of the designs, obtained through the ITC'99 benchmark set [CRS00] and opencores.org, were synthesized with an academic 90nm standard cell library [GBW+09] in full-scan mode. For all the designs, we allocated 20% of inverters (INVX0) - selected randomly - to be converted to covert gates (with half as $NOR2X2$ and half as $NAND2X2$). The covert gate timing is emulated by setting a false timing path on the dummy pin, and power estimation is done by setting the dummy pin to a non-controlling value for the gate. Area, delay and power overhead estimates, as well as fan-out analysis to avoid combinational feedback loops were all performed in Synopsys Design Compiler. The results are shown in Table 7. From the results, we can see that the overhead from the covert gates is minimal with respect to the area, delay and power characteristics of the original design. However, in some benchmarks, we noted a slightly large increase in delay, as the covert gates were placed on the critical path.

Note that in actual applications, we would not be limited to only inverter-to-NAND conversion. Any n-input gate could be converted into a n+m input gate of the same type (where $m > 1$). Further, the no. of gates that are converted to covert gates could also be increased, based on the available overhead budget. However, we noticed that when a very large number of gates are converted to covert gates, delay and area overheads get worse. For example, when one or two extra pins are introduced to roughly half of all logic gates in the s38584 benchmark, delay overhead of 53.24% and area overhead of 19% were noticed.

6.4 Other Possible Attacks

In the VLSI test community, always-open and always-closed defects have been studied exhaustively. Although we used a test-based attack using stuck-at fault modeling, a natural question to ask would be if regular always-open/always-closed tests can be used to detect

the camouflaged cells. Always-closed defects are usually considered in the context of short-circuit current i.e., if there is an always-closed transistor that causes a large current spike from VDD to GND, it can usually be detected by parametric tests under static conditions (i.e., no switching). However, our gates are constructed such that there is never a direct path from VDD to GND in any of the transistors in the covert gates. Therefore, it would not be possible to use leakage current measurements to detect the covert gates.

Always-open defects tend to behave as memory elements, as they cause floating nodes in the circuit. Therefore, a sequence of two input vectors could be used to first set the logic gate output to a known value, and then generate another vector to detect the fault [LH91]. However, in our covert gates, an 'open defect' in the pull-up network is always compensated by a 'closed defect' in the pull-down network (and vice-versa). Thus, a regular two-pattern test to detect open defects is not applicable in our case.

7 Conclusion and Future Work

In this paper, we proposed a new covert gate-based camouflaging strategy that is rooted in the true meaning of camouflaging i.e., creating gates that are truly indistinguishable from regular logic gates in a design. We proposed transistor structures to realize such gates and characterized their resistance to SEM imaging through experimental analysis. We also created models for the camouflaged logic gates and showed that they achieve much lower overhead than those based on regular camouflaging techniques (e.g., dummy contacts). SAT and test-based attack evaluations were also performed on the camouflaging technique. We showed that in the absence of the attacker's ability to pin-point the covert gates, attack success greatly diminishes for both SAT and test-based attacks.

In the future, we plan to fabricate covert logic gates and experimentally characterize their performance. This would also allow us to measure the accuracy of our circuit models from Section 3.6. Future work could also look at combining covert gates with logic locking techniques, so that the design becomes key-dependent and can be metered while also increasing SAT resistance against non-foundry attackers. Towards the microscope side, the effectiveness of imaging covert gates with helium/neon ion microscopes could also be investigated.

Acknowledgments

This work was supported in part by NSF under grant CNS 1651701 and AFOSR under award number FA9550-14-1-0351.

References

[AEM18] Nail Etkin Can Akkaya, Burak Erbagci, and Ken Mai. A secure camouflaged logic family using post-manufacturing programming with a 3.6 ghz adder prototype in 65nm cmos at 1v nominal v dd. In *Solid-State Circuits Conference-(ISSCC), 2018 IEEE International*, pages 128–130. IEEE, 2018.

[AGDM15] Luca Amarú, Pierre-Emmanuel Gaillardon, and Giovanni De Micheli. The epfl combinational benchmark suite. In *Proceedings of the 24th International Workshop on Logic & Synthesis (IWLS)*, number CONF, 2015.

[BA04] Michael Bushnell and Vishwani Agrawal. *Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits*, volume 17. Springer Science & Business Media, 2004.

[BRPB13] Georg T Becker, Francesco Regazzoni, Christof Paar, and Wayne P Burleson. Stealthy dopant-level hardware trojans. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 197–214. Springer, 2013.

[Bry85] David Bryan. The iscas'85 benchmark circuits and netlist format. *North Carolina State University*, 25, 1985.

[CCF+15] Shuai Chen, Junlin Chen, Domenic Forte, Jia Di, Mark Tehranipoor, and Lei Wang. Chip-level anti-reverse engineering using transformable interconnects. In *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), 2015 IEEE International Symposium on*, pages 109–114. IEEE, 2015.

[CCJB07] Lap-Wai Chow, William M Clark Jr, and James P Baukus. Covert transformation of transistor properties as a circuit protection method, May 15 2007. US Patent 7,217,977.

[CEMG16] Maria I Mera Collantes, Mohamed El Massad, and Siddharth Garg. Threshold-dependent camouflaged cells to secure circuits against reverse engineering attacks. In *VLSI (ISVLSI), 2016 IEEE Computer Society Annual Symposium on*, pages 443–448. IEEE, 2016.

[Che16] Augustus KW Chee. Quantitative dopant profiling by energy filtering in the scanning electron microscope. *IEEE Transactions on Device and Materials Reliability*, 16(2):138–148, 2016.

[CRS00] Fulvio Corno, Matteo Sonza Reorda, and Giovanni Squillero. Rt-level itc'99 benchmarks and first atpg results. *IEEE Design & Test of computers*, 17(3):44–53, 2000.

[DBDN+14] Sophie Dupuis, Papa-Sidi Ba, Giorgio Di Natale, Marie-Lise Flottes, and Bruno Rouzeyre. A novel hardware logic encryption technique for thwarting illegal overproduction and hardware trojans. In *On-Line Testing Symposium (IOLTS), 2014 IEEE 20th International*, pages 49–54. IEEE, 2014.

[EBH02] SL Elliott, RF Broom, and CJ Humphreys. Dopant profiling with the scanning electron microscope—a study of si. *Journal of applied physics*, 91(11):9116–9122, 2002.

[EMGT15] Mohamed El Massad, Siddharth Garg, and Mahesh V Tripunitara. Integrated circuit (ic) decamouflaging: Reverse engineering camouflaged ics within minutes. In *NDSS*, 2015.

[GBW+09] Richard Goldman, Karen Bartleson, Troy Wood, Kevin Kranen, C Cao, Vazgen Melikyan, and Gayane Markosyan. Synopsys' open educational design kit: capabilities, deployment and future. In *Microelectronic Systems Education, 2009. MSE'09. IEEE International Conference on*, pages 20–24. IEEE, 2009.

[GHD+14] Ujjwal Guin, Ke Huang, Daniel DiMase, John M Carulli, Mohammad Tehranipoor, and Yiorgos Makris. Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain. *Proceedings of the IEEE*, 102(8):1207–1228, 2014.

[GNM+17] Joseph I Goldstein, Dale E Newbury, Joseph R Michael, Nicholas WM Ritchie, John Henry J Scott, and David C Joy. *Scanning electron microscopy and X-ray microanalysis*. Springer, 2017.

[HMD99] Paul Hasler, Bradley A Minch, and Chris Diorio. Adaptive circuits using pfet floating-gate devices. In *Advanced Research in VLSI, 1999. Proceedings. 20th Anniversary Conference on*, pages 215–229. IEEE, 1999.

[IVR+18] Anirudh S. Iyengar, Deepak Vontela, Ithihasa Reddy, Swaroop Ghosh, Syedhamidreza Motaman, and Jae-won Jang. Threshold defined camouflaged gates in 65nm technology for reverse engineering protection. In *Proceedings of the International Symposium on Low Power Electronics and Design*, ISLPED '18, pages 6:1–6:6, New York, NY, USA, 2018. ACM.

[LH91] Hyung Ki Lee and Dong Sam Ha. Soprano: an efficient automatic test pattern generator for stuck-open faults in cmos combinational circuits. In *Proceedings of the 27th ACM/IEEE Design Automation Conference*, pages 660–666. ACM, 1991.

[LSM+16] Meng Li, Kaveh Shamsi, Travis Meade, Zheng Zhao, Bei Yu, Yier Jin, and David Z Pan. Provably secure camouflaging strategy for ic protection. In *Computer-Aided Design (ICCAD), 2016 IEEE/ACM International Conference on*, pages 1–8. IEEE, 2016.

[LTBS16] Heiko Lohrke, Shahin Tajik, Christian Boit, and Jean-Pierre Seifert. No place to hide: Contactless probing of secret data on fpgas. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 147–167. Springer, 2016.

[LYZH16] Duo Liu, Cunxi Yu, Xiangyu Zhang, and Daniel Holcomb. Oracle-guided incremental sat solving to reverse engineer camouflaged logic circuits. In *Proceedings of the 2016 Conference on Design, Automation & Test in Europe*, pages 433–438. EDA Consortium, 2016.

[MBPB15] Shweta Malik, Georg T Becker, Christof Paar, and Wayne P Burleson. Development of a layout-level hardware obfuscation tool. In *VLSI (ISVLSI), 2015 IEEE Computer Society Annual Symposium on*, pages 204–209. IEEE, 2015.

[MBT17] Prabhat Mishra, Swarup Bhunia, and Mark Tehranipoor. *Hardware IP security and trust*. Springer, 2017.

[PAF+17] EL Principe, Navid Asadizanjani, Domenic Forte, Mark Tehranipoor, Robert Chivas, Michael DiBattista, Scott Silverman, Mike Marsh, Nicolas Piche, and John Mastovich. Steps toward automated deprocessing of integrated circuits. In *International Symposium on Testing and Failure Analysis*, 2017.

[RFBL18] Christian Rivero, Pascal Fornara, Guilhem Bouton, and Mathieu Lisart. Method for forming at least one electrical discontinuity in an interconnection part of an integrated circuit without addition of additional material, and corresponding integrated circuit, May 24 2018. US Patent App. 15/596,877.

[RSSK13] Jeyavijayan Rajendran, Michael Sam, Ozgur Sinanoglu, and Ramesh Karri. Security analysis of integrated circuit camouflaging. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 709–720. ACM, 2013.

[RZZ+15] Jeyavijayan Rajendran, Huan Zhang, Chi Zhang, Garrett S Rose, Youngok Pino, Ozgur Sinanoglu, and Ramesh Karri. Fault analysis-based logic encryption. *IEEE Transactions on computers*, 64(2):410–424, 2015.

[SAFT16] Bicky Shakya, Navid Asadizanjani, Domenic Forte, and Mark Tehranipoor. Chip editor: leveraging circuit edit for logic obfuscation and trusted fabrication. In *Proceedings of the 35th International Conference on Computer-Aided Design*, page 30. ACM, 2016.

[SLM+17] Kaveh Shamsi, Meng Li, Travis Meade, Zheng Zhao, David Z Pan, and Yier Jin. Appsat: Approximately deobfuscating integrated circuits. In *Hardware Oriented Security and Trust (HOST), 2017 IEEE International Symposium on*, pages 95–100. IEEE, 2017.

[SRM15] Pramod Subramanyan, Sayak Ray, and Sharad Malik. Evaluating the security of logic encryption algorithms. In *Hardware Oriented Security and Trust (HOST), 2015 IEEE International Symposium on*, pages 137–143. IEEE, 2015.

[SSF+14] Takeshi Sugawara, Daisuke Suzuki, Ryoichi Fujii, Shigeaki Tawa, Ryohei Hori, Mitsuru Shiozaki, and Takeshi Fujino. Reversing stealthy dopant-level circuits. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 112–126. Springer, 2014.

[TJ07] Randy Torrance and Dick James. Reverse engineering in the semiconductor industry. In *Custom Integrated Circuits Conference, 2007. CICC'07. IEEE*, pages 429–436. IEEE, 2007.

[TJ11] Randy Torrance and Dick James. The state-of-the-art in semiconductor reverse engineering. In *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, pages 333–338. IEEE, 2011.

[TK10] Mohammad Tehranipoor and Farinaz Koushanfar. A survey of hardware trojan taxonomy and detection. *IEEE design & test of computers*, 27(1), 2010.

[XS16] Yang Xie and Ankur Srivastava. Mitigating sat attack on logic locking. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 127–146. Springer, 2016.

[XSTF17] Xiaolin Xu, Bicky Shakya, Mark M Tehranipoor, and Domenic Forte. Novel bypass attack and bdd-based tradeoff analysis against all known logic locking attacks. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 189–210. Springer, 2017.

[YMSR16] Muhammad Yasin, Bodhisatwa Mazumdar, Ozgur Sinanoglu, and Jeyavijayan Rajendran. Camoperturb: secure ic camouflaging for minterm protection. In *Proceedings of the 35th International Conference on Computer-Aided Design*, page 29. ACM, 2016.

[YMSR17] Muhammad Yasin, Bodhisatwa Mazumdar, Ozgur Sinanoglu, and Jeyavijayan Rajendran. Removal attacks on logic locking and camouflaging techniques. *IEEE Transactions on Emerging Topics in Computing*, 2017.

[YZL+17] Cunxi Yu, Xiangyu Zhang, Duo Liu, Maciej Ciesielski, and Daniel Holcomb. Incremental sat-based reverse engineering of camouflaged logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(10):1647–1659, 2017.

[ZC07] Wei Zhao and Yu Cao. Predictive technology model for nano-cmos design exploration. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 3(1):1, 2007.