Wang XY, Zhou Q, Cai YC *et al.* Spear and shield: Evolution of integrated circuit camouflaging. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 33(1): 42–57 Jan. 2018. DOI 10.1007/s11390-018-1807-6

Spear and Shield: Evolution of Integrated Circuit Camouflaging

Xue-Yan Wang[1], *Student Member, ACM, IEEE*, Qiang Zhou[1,*], *Senior Member, CCF, Member, ACM, IEEE*
Yi-Ci Cai[1], *Senior Member, CCF, Member, IEEE*, and Gang Qu[2], *Senior Member, IEEE*

[1] *Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*

[2] *Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20740, U.S.A.*

E-mail: wangxueyan13@mails.tsinghua.edu.cn; {zhouqiang, caiyc}@mail.tsinghua.edu.cn; gangqu@umd.edu

Received March 22, 2017; revised November 29, 2017.

Abstract Intellectual property (IP) protection is one of the hardcore problems in hardware security. Semiconductor industry still lacks effective and proactive defense to shield IPs from reverse engineering (RE) based attacks. Integrated circuit (IC) camouflaging technique fills this gap by replacing some conventional logic gates in the IPs with specially designed logic cells (called camouflaged gates) without changing the functions of the IPs. The camouflaged gates can perform different logic functions while maintaining an identical look to RE attackers, thus preventing them from obtaining the layout information of the IP directly from RE tools. Since it was first proposed in 2012, circuit camouflaging has become one of the hottest research topics in hardware security focusing on two fundamental problems. How to choose the types of camouflaged gates and decide where to insert them in order to simultaneously minimize the performance overhead and optimize the RE complexity? How can an attacker de-camouflage a camouflaged circuit and complete the RE attack? In this article, we review the evolution of circuit camouflaging through this spear and shield race. First, we introduce the design methods of four different kinds of camouflaged cells based on true/dummy contacts, static random access memory (SRAM), doping, and emerging devices, respectively. Then we elaborate four representative de-camouflaging attacks: brute force attack, IC testing based attack, satisfiability-based (SAT-based) attack, and the circuit partition based attack, and the corresponding countermeasures: clique-based camouflaging, CamoPerturb, AND-tree camouflaging, and equivalent class based camouflaging, respectively. We argue that the current research efforts should be on reducing overhead introduced by circuit camouflaging and defeating de-camouflaging attacks. We point out that exploring features of emerging devices could be a promising direction. Finally, as a complement to circuit camouflaging, we conclude with a brief review of other state-of-the-art IP protection techniques.

Keywords circuit camouflaging, reverse engineering, intellectual property (IP) protection, hardware security

1 Introduction

The continually increasing design complexity and design cost have led to the globalization of integrated circuit (IC) design and fabrication, where rogues may exist in all phases of the supply chain. Design intellectual property (IP) infringement, IC counterfeiting and overbuilding, hardware trojans, side channel attacks, and others have caused serious security and economic concerns in semiconductor industry[1-4]. A major enabler of these malicious behaviors is reverse engineering (RE), which has been developed along with the advances in IC design for good purposes such as fault analysis, chip testing and verification. However, rogues can use commercially available RE tools to clone, pirate, or counterfeit a design, which is defined as dishonest RE[5]. Dishonest RE has helped to discover security vulnerabilities in critical commercial and military systems because it provides the rogues a good understanding of the victim design[①][6], and economically, it has resulted in billions of dollars loss each year[②].

Survey

This work is supported by the National Natural Science Foundation of China under Grant No. 61774091. Gang Qu is supported in part by Air Force Office of Scientific Research Multi-University Research Initiative of USA under Award No. FA9550-14-1-0351.

*Corresponding Author

[①] Reverse engineering for War. https://historylist.wordpress.com/category/war/, Aug. 2016.

[②] Innovation is at risk as semiconductor equipment and materials industry loses up to $4 billion annually due to IP infringement. http://www.semi.org/en/Press/P043775, Aug. 2016.

The traditional digital circuit watermarking and fingerprinting techniques[7-10] are passive IP protection schemes because they do not prevent RE from happening or make it more difficult. Watermarks and fingerprints can be embedded into the IP to make each instance of the IP unique. When necessary, they can be revealed to show the authorship or ownership of the IP and identify the parties that misuse the IP. Although it is hard or impossible to completely remove the watermark and fingerprint, RE attackers can still extract valuable information from the IP and reproduce the IP illegally. The existence of watermarks and fingerprints in the IP can only deter RE, and will not increase the complexity of RE.

Circuit camouflaging is a technique that is applied in combinational logic of application specific integrated circuit (ASIC), which proactively hides the layout information of IPs in aim to make RE exponentially more difficult[11-13]. Specifically, it hides the design information of IC by replacing some conventional logic gates with specially designed camouflaged cells (called camouflaged gates), in which the camouflaged gates have been configured to perform one of the multiple functionalities (the same as that of the replaced conventional logic gate) while maintaining an identical look to RE attackers. Therefore, while the attacker performs top-down reverse engineering, he/she will not know the functionalities of the camouflaged gates and has to pay additional efforts to guess. Once he/she is not able to resolve the real functionalities of all the camouflaged gates, he/she will get an incomplete or deceived netlist, and thus fail to reverse engineer the IC.

As shown in Fig.1, when two conventional logic gates are replaced by camouflaged cells, even with the help of RE tools, an attacker will not be able to identify the real functionalities of the camouflaged gates and have to try other ways[12].

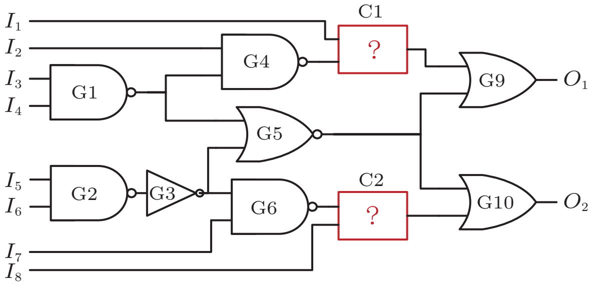


Fig.1. Attackers not knowing the functionalities of camouflaged gates C1 and C2[12]. Gi denotes the traditional logic gates, and I_i and O_i denote input and output signals, respectively.

The design of the camouflaged cells relies on the general belief that RE technology is normally 2∼3 generations behind the latest CMOS (complementary metal oxide semiconductor) design technology. That is, the CMOS design features cannot be completely reverse engineered in several years. Therefore, special logic cells (or camouflaged cells) can be designed to have identical look from the top-view but can be configured to perform one of the multiple functionalities.

In order to defeat circuit camouflaging, various decamouflaging attacks have been proposed: the IC testing based attack applies testing principles to resolve the functionality of each camouflaged gate[12], satisfiability-based (SAT-based) attack③ utilizes SAT solver to gradually prune incorrect functionality combinations of camouflaged gates[14-16], brute force attack[12,17] searches for the correct functionalities for the camouflaged gates by enumerating all possible combinations, and circuit partition based attack[17] partitions camouflaged gates into multiple sub-circuits to attack separately. These powerful attacks spearhead the efforts of circuit de-camouflaging and have become legitimate threats to the effectiveness of circuit camouflaging.

Fortunately, with the invention of each circuit decamouflaging attack, corresponding countermeasures are also introduced: the clique-based method selects interfered gates for camouflaging in order to thwart IC testing based attack[12], multiplexer-based camouflaging increases the brute force complexity[18-19], equivalent class guided camouflaging hampers circuit partition based attack[19], and CamoPerturb[20] and AND-tree camouflaging[21] force the attacker to call the SAT solver exponential amount of times.

This fierce race between sharpening the spears of decamouflaging tools and making the camouflaging shield more robust has quickly elevated the sophistication and maturity level of circuit camouflaging. We believe that such alternative "spear and shield" process is making circuit camouflaging the most effective countermeasure against RE-based attacks. This article reviews the research advance in state-of-the-art circuit camouflaging (with an emphasis on the race between various decamouflaging attacks and the corresponding defending mechanisms), and then analyzes existing challenges and future development directions for circuit camouflaging.

The rest of this article is organized as follows. In Section 2, we introduce the basic concepts of circuit camouflaging and elaborate the existing approaches to creating camouflaged cells. In Section 3, we explain in

③Boolean satisfiability problem. https://en.wikipedia.org/wiki/Boolean_satisfiability_problem/, Nov. 2016.

details the representative de-camouflaging spears. The countermeasures to strengthen the camouflaging shield are presented in Section 4. We point out some applications, research needs and challenges for circuit camouflaging in Section 5. Section 6 gives a brief survey on other methods for IP protection and provides a comparison between these methods and circuit camouflaging. The paper is concluded in Section 7.

2 Foundation: Camouflaged Cells

Camouflaged cells form the foundation of circuit camouflaging technique, and they are used to replace some selected conventional logic gates in the circuit. To keep the function of the camouflaged circuit unchanged, camouflaged cells need to be configured to perform different functionalities, which are the same as those of logic gates being replaced, and they at the same time maintain an identical look to RE attackers. In addition to this basic requirement, they should also perform as many functionalities as possible to increase the confusion level, and simultaneously incur small or no performance overheads (such as area, power, and timing). In this section, we summarize the technologies that have been proposed to build the camouflaged cells.

2.1 True/Dummy Contact Based Camouflaged Cell

One of the most popular ways to build camouflaged cells is with true/dummy contacts[11,22-25]. A true contact spans the dielectric between two adjacent metal layers to represent an electrical connection, and a dummy contact has a gap in the middle to fake the connection between layers. One possible implementation is to modify the shape of polysilicon to create extra overlaps between polysilicon and metal layers and use true/dummy contacts to connect the layers. With different configurations for the contacts being true or dummy, the camouflaged cells can have multiple possible functionalities.

It is infeasible to differentiate between connection (true) and isolation (dummy) of a camouflage connector for real-world RE attackers. On one hand, the contacts appear identical from the top view even under optical or electron microscopy[25-27]; on the other hand, for chemical erosion and imaging-based top-down reverse engineering, the camouflage connectors that are placed in bottom layers are almost eroded when the attacker reaches the layer. The attacker will not know whether a broken/isolated connector is due to chemical erosion or camouflaging[12,24].

Therefore, given that the only difference of the camouflaged cells is the true/dummy configuration for contacts, these camouflaged cells will appear to be identical to RE attackers. {NAND, NOR, XOR}, {INV, BUF}, and multiplexer-based camouflaged cells in the literature are examples of such true/dummy contact based camouflaged cells.

2.1.1 *{NAND, NOR, XOR} Camouflaged Cell*

Fig.2 demonstrates the layout of {NAND, NOR, XOR} camouflaged cell, in which 19 true/dummy contacts are inserted. Vdd and Gnd are short for Voltage Drain Drain and Ground, respectively. Different configurations to implement NAND, NOR, and XOR are listed in Table 1[12].

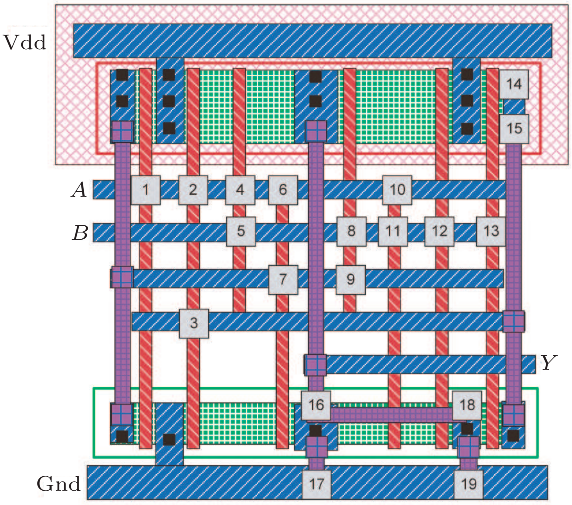


Fig.2. Camouflaged layout of {NAND, NOR, XOR} cell[12]. *A* and *B* denote input signals, and *Y* denotes the output signal.

Table 1. True/Dummy Contact Configurations of the Camouflaged Cell to Implement Different Functionalities[12]

Functionality	Contact	
	True	Dummy
NAND	2, 4, 6, 8, 11, 12, 16, 17	1, 3, 5, 7, 9, 10, 13, 14, 15, 18, 19
NOR	2, 5, 6, 11, 12, 18, 19	1, 3, 4, 7, 8, 9, 10, 13, 14, 15, 16, 17
XOR	1, 3, 4, 7, 9, 10, 12, 13, 14, 15, 18, 19	2, 5, 6, 8, 11, 16, 17

2.1.2 *{INV, BUF} Camouflaged Cell*

{INV, BUF} camouflaged cell is also constructed with true/dummy contacts[20]. As shown in Fig.3, when contact 1 is true and contact 2 is dummy, the cell behaves like an inverter; when contact 1 is dummy and contact 2 is true, it behaves like a buffer. This {INV, BUF} camouflaged cell is widely used in anti-SAT camouflaging approaches[20-21].

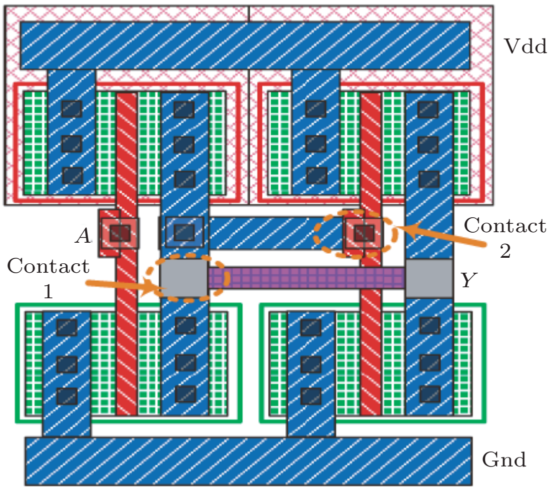


Fig.3. Camouflaged layout of {INV, BUF} cell[20]. A denotes the input signal and Y denotes the output signal.

2.1.3 *Multiplexer-Based Camouflaged Cell*

As shown in Fig.4, in a multiplexer-based camouflaged cell, each input line x_i of the multiplexer is connected to both Vdd and Gnd by two true/dummy contacts, with only one contact being a connection (true) and the other one being an isolation (dummy)[13,19]. Specifically, the Vdd contact to be true and the Gnd contact to be dummy mean that x_i is configured to "1"; the Vdd connector to be dummy and the Gnd connector to be true mean that x_i is configured to "0". With the selection lines being the inputs and the output line being the output, the functionality of the camouflaged cell can be expressed as $Y = (\overline{A} \times \overline{B}) \times x_1 + (\overline{A} \times B) \times x_2 + (A \times \overline{B}) \times x_3 + (A \times B) \times x_4$. Therefore, the camouflaged cell can have 16 possible 2-input 1-output Boolean functions corresponding to 16 possible configurations for x_1, x_2, x_3, and x_4. For example, when x_1, x_2, x_3, and x_4 are configured to be 1110, $Y = \overline{A} \times \overline{B} + \overline{A} \times B + A \times \overline{B} = \overline{A \times B}$, and the camouflaged cell will perform like an NAND gate.

2.2 SRAM-Based Camouflaged Cell

The SRAM-based camouflaged cell conceals its real functionality by storing the configuration information in tamper-proof memories (SRAM). For example, for the camouflaged cell in Fig.4, instead of configuring x_i with true/dummy contacts, it uses memory cells in which the configuration bits are stored. This will achieve similar effect to configure the 4-by-1 multiplexer to perform 16 possible 2-input 1-output Boolean functions[18,28].

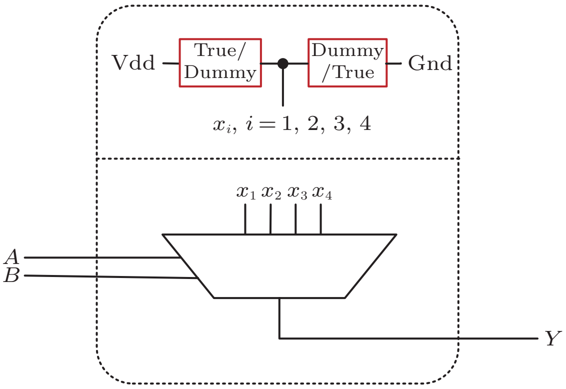


Fig.4. Configuring multiplexer with true/dummy contacts as camouflaged cell.

2.3 Doping-Based Camouflaged Cell

The doping-based camouflaged cell integrates some always-on/off MOS (metal oxide semiconductor) transistors, which are constructed by changing the type and shape of the Lightly-Doped-drain (LDD), specifically, either by changing the polarity of dopant for the source and drain of MOS transistors[29], or by changing the type and length of the LDD implants[30]. Therefore, the doping-based camouflaged cells will have exactly the same metal and polysilicon layers with standard cells in the library[21].

For example, in Fig.5, for a conventional 2-input NAND cell, when the always-on doping scheme is used for NMOS transistor and the always-off doping scheme for PMOS transistor, the camouflaged cell will become an inverter[21].

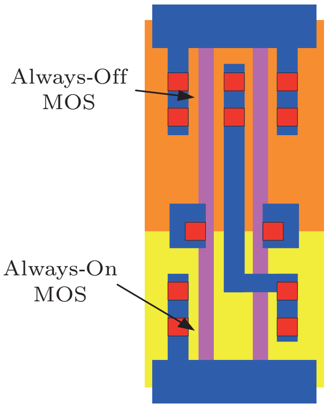


Fig.5. Applying always-on/off transistors in NAND gate to make it perform like an inverter[21].

2.4 Emerging Device Based Camouflaged Cell

Emerging devices have originally been studied as alternatives to CMOS technology in order to meet the scaling challenges. Examples of such devices include FinFETs[31], tunnelFETs (TFETs)[32], carbon nanotube FETs (CNTFETs)[33], graphene-based symmetric tunneling FETs (SymFETs)[34], memristors[35-36] and spin-transfer-torque devices (STT)[37]. Recently however, it has been demonstrated that these devices have unique features which can be naturally utilized in security related applications[38-45].

In the context of circuit camouflaging, emerging technologies have been utilized to construct camouflaged cells. For example, with the unique polarity controllable property of SiNW FETs, one can build camouflaged cells without extra redundant FETs[44,46]. Also, STT-based LUT has been designed as an alternative to SRAM-based LUT[18], with the excellent features of high integration density, high retention time, high endurance near-zero leakage, and thermal robustness[42,47-48].

2.5 Discussions on Various Camouflaged Cells

Using emerging devices to build camouflaged cells is a promising trend and has attracted many attentions in research community. However, despite many advantages over CMOS logic, most emerging devices are still under the simulation phase and there is still a long way to go before we can integrate them in real circuit design. The advantage of SRAM-based camouflaged cell is the re-configurable property, which can be utilized in on-line hardware trojan detection by loading a wrong configuration, once there is trojan being detected. The drawbacks would be that the overhead of memory cells may be rather high and non-volatile secure memory will be needed. Doping-based camouflaged cell poses very high technology requirements, which needs to precisely control the shape or type of LDD. The true/dummy contact based camouflaged cell has been relatively well studied and widely used in academic research. Designers have hoped to increase the number of possible functionalities each camouflaged cell can perform, and reduce the incurred performance overheads. However, there are usually trade-offs between the two metrics. For example, the 16-function multiplexer-based camouflaged cell incurs higher performance overheads than the 3-function {NAND, NOR, XOR} camouflaged cell. Therefore, how to increase the number of possible functionalities for each camouflaged cell while reducing performance overheads still remains a challenge that needs to be addressed urgently.

3 Spears of De-Camouflaging

With existing RE techniques and tools, an attacker can easily get a camouflaged netlist comprising of conventional logic gates and camouflaged gates. To fully reverse engineer the IC, the attacker has to resolve the functionalities of camouflaged gates (which is the hidden information by circuit camouflaging), and such a process is called a de-camouflaging attack. Once de-camouflaging attacks succeed, that is, the functionalities of camouflaged gates are revealed, circuit camouflaging will lose its value and fail to protect the IC and the IPs in the IC from RE-based attacks.

Various de-camouflaging attacks have been proposed in the literature and they have posed serious threats to the effectiveness of circuit camouflaging. In this section, we will analyze four representative attacks on their strength and weakness.

Before we elaborate the reported attacks, it is important to understand the basic attacking models. In this article, we adopt the following assumptions that have been widely used in state-of-the-art circuit camouflaging techniques[12].

1) The attacker is able to extract a camouflaged netlist that consists of conventional logic gates and camouflaged logic gates by state-of-the-art reverse engineering tools and techniques.

2) The attacker can differentiate between regular gates and camouflaged gates. And he/she knows possible functionalities that each camouflaged gate can perform.

3) The attacker can buy an unpackaged functional IC from the market, while he/she only has access to the IC's primary inputs (PI) and primary outputs (PO), namely, he/she can only treat the functional IC as a black box and get corresponding outputs for a given input vector.

3.1 IC Testing Based Attack

Given one camouflaged gate, the most straightforward way to resolve its functionality is to get its input-output behaviors (or truth table of the gate). For a 2-input 1-output camouflaged gate, its functionality has at most 16 possibilities; thus at most four input-output pairs (3-function {NAND, NOR, XOR} camouflaged gate needs two input-output pairs, while 16-function multiplexer-based camouflaged gate needs four pairs) will be sufficient to resolve its functionality.

For example, each camouflaged gate is configured to perform one of the functions of {NAND, NOR, XOR}[12]. Based on the fact that 1) the output of a camouflaged gate under input "00" can differentiate {XOR} from {NAND, NOR} (XOR outputs 0, while both NAND and NOR output 1), and 2) the output under input "01" or "10" can differentiate {NAND} and {NOR} (NAND outputs 1, while NOR outputs 0), and the functionality of a camouflaged gate can be resolved with its outputs under inputs "00" and "01"/"10".

Given that an attacker only has access to the PIs and POs of the functional IC, the input-output pairs of a camouflaged gate can be gotten by justification and sensitization techniques in IC testing[12,49]. Justification is to justify the inputs of a gate to a known value by controlling one or more of the gate's related PIs, and sensitization is to observe the value of a gate's output to a PO by setting all side inputs of gates in between to non-controlling values[12,49].

For example, in Fig.1, the attacker can apply input pattern "010XXXXX" (X represents don't care values, which means it can be either 0 or 1) at the PIs. It justifies the camouflaged gate C1's inputs to "00", and sensitizes C1's output to PO O_1. If O_1 is 0, the functionality of C1 is resolved to be XOR; if O_1 is 1, C1 will be either NAND or NOR. The attacker can then apply input pattern "110XXXXX" at PIs to justify C1's inputs as "10" and sensitize C1's output to O_1. If O_1 is 0, C1 is resolved to be NOR; otherwise, C1 is resolved to be NAND. Similar methods can be applied to resolve the functionality of C2. Such a process can be done very quickly and automatically with existing ATPG tools such as HOPE fault simulation tool[50].

3.2 Brute Force Attack

Unlike IC testing based attack which resolves the functionalities of camouflaged gates individually one by one, brute force attack enumerates all possible functionality combinations of camouflaged gates[12,17]. We call each possible functionality combination an assignment to the camouflaged gates. For example, in Fig.1, both C1 and C2 have three possibilities, namely {NAND, NOR, XOR}. Therefore, for C1 and C2, there will be 3^2 possible functionality combinations (assignments), namely, (NAND, NAND), (NAND, NOR), (NAND, XOR), (NOR, NAND), (NOR, NOR), (NOR, XOR), (XOR, NAND), (XOR, NOR), and (XOR, XOR).

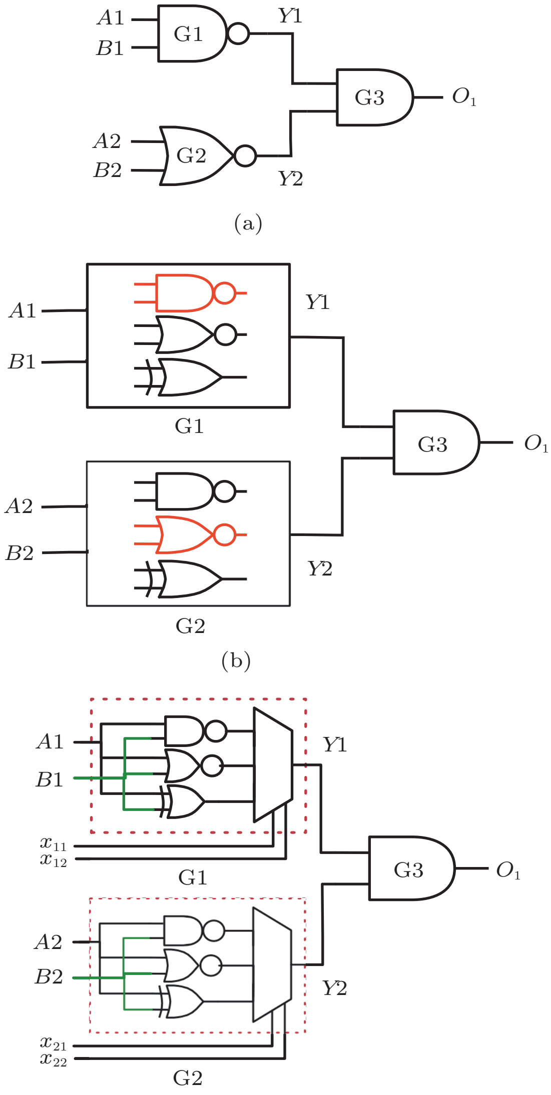
For each possible assignment, the attacker will simulate to apply input patterns at PIs, get the corresponding outputs at POs, and compare these outputs with an unpackaged/functional IC. If they are the same, the attacker has found out the correct assignment and thus camouflaged gates are resolved; otherwise, the attacker will try the next possible assignment and repeat the process. The needed brute force efforts will be M^N, where M is the number of possible functionalities each camouflaged gate can have, and N is the number of camouflaged gates in the camouflaged circuit. Brute force attack is able to find out the correct assignment given enough time. However, the required time increases exponentially as N increases; therefore, it can easily become unacceptable when N is relatively large.

3.3 SAT-Based Attack

SAT-based attack also treats the camouflaged gates as an entity to find the correct assignment, while different from brute force attack, it starts with a set of possible assignments, and prunes the incorrect ones iteratively with discriminating input (DI), until there is only one assignment left or all the left assignments have the same output under any input patterns. A discriminating input is an input pattern which, when applied in a camouflaged netlist, produces incorrect output for at least one incorrect assignment. Due to the fact that on average each discriminating input can prune multiple incorrect assignments, SAT-based attack is able to bypass the exponential complexity in brute force attack, and it has been reported to be able to resolve camouflaged ICs within only minutes[15-16], greatly threatening the security of circuit camouflaging.

Specifically, SAT-based attack starts with a possible assignment set S that contains all possible assignments. When there are N camouflaged cells in the camouflaged circuit, and the functionality of each camouflaged cell has M possibilities, we have $|S| = M^N$. Each time the SAT solver will compute a discriminating input DI i, which means $\exists X_a, X_b \in S$, $C_{X_a}(i) \neq C_{X_b}(i)$, where C_{X_a} and C_{X_b} denote the de-camouflaged circuit under assignments X_a and X_b respectively; therefore, at least one of X_a and X_b is an incorrect assignment. Then this process will be repeated to find more DIs, until with the current found set of DIs, all incorrect assignments can be pruned and the correct one can be extracted. This set of DIs is also called a discriminating set of inputs (DSI). The number of DIs in DSI is an important metric to measure the complexity of SAT-based attack.

Fig.6(a) and Fig.6(b) show circuits before and after G1 and G2 are camouflaged with {NAND, NOR, XOR} camouflaged cells[12]. The attacker only knows that each of them can be one of {NAND, NOR, XOR}, and models them in the way shown in Fig.6(c). For each camouflaged cell G_i, two programming bits x_{i1}, x_{i2} will be added as selection bits which produce the following function $Yi = Ai \oplus Bi \times (\overline{x_{i1}} \times \overline{x_{i2}}) + \overline{Ai \times Bi} \times (\overline{x_{i1}} \times x_{i2}) + \overline{(Ai + Bi)} \times (x_{i1} \times \overline{x_{i2}})$.



(a)

(b)

(c)

Fig.6. (a) Original circuit. (b) Camouflaged circuit with gates G1 and G2 replaced by {NAND, NOR, XOR} camouflaged cells. Gates in red denote their real functionalities[15,20]. (c) Modeling {NAND, NOR, XOR} camouflaged cells.

When $x_{i1}x_{i2}$ are assigned with "00", "01", or "10", Yi will output those of XOR, NAND, or NOR, respectively. Note that $x_{i1}x_{i2}$ is forbidden to be "11" for the functionality of each camouflaged cell only has three possibilities. Therefore, an assignment X = (00, 01) for (G1, G2) means that G1 and G2 are resolved as XOR and NAND, respectively. The goal of SAT-based de-camouflaging will be finding the correct assignment $X = (x_{11}x_{12}, x_{21}x_{22})$ for (G1, G2). As shown in Table 2, by SAT-based attack, with only three DIs, eight incorrect assignments for (G1, G2) can be pruned, and the correct assignment (01, 10) will be left (backgrounded in yellow).

Table 2. Three DIs Pruning All Incorrect Assignments[15]

$(x_{11}x_{12}, x_{21}x_{22})$	(G1, G2)	DI ($A1$ $B1$ $A2$ $B2$)		
		0000	0001	0100
(00,00)	(XOR, XOR)	×	×	√
(00,01)	(XOR, NAND)	×	√	√
(00,10)	(XOR, NOR)	×	√	√
(01,00)	(NAND, XOR)	×	×	×
(01,01)	(NAND, NAND)	√	√	×
(01,10)	(NAND, NOR)	√	√	√
(10,00)	(NOR, XOR)	×	×	×
(10,01)	(NOR, NAND)	√	×	×
(10,10)	(NOR, NOR)	√	×	√

Note: √ represents the correct output, and × represents the incorrect output.

3.4 Circuit Partition Based Attack

Circuit partition based attack applies the "divide and conquer" methodology to partition the camouflaged gates into multiple disjoint sub-circuits based on certain criteria such that these sub-circuits can be attacked sequentially to reduce the complexity of de-camouflaging[13,17]. More precisely, the de-camouflaging complexity will be determined by the largest number of camouflaged gates in a sub-circuit, regardless of how many gates are camouflaged in the circuit.

Fig.7 shows a motivational example. When trying to resolve camouflaged gates individually by IC testing techniques, one may notice that C1's output cannot be sensitized to O_1/O_2 because of the existence of C2 and C3; C3's inputs cannot be justified from I_1, I_2, I_3, and I_4 because of the existence of C1 and C2; both C2's inputs and output cannot be justified or sensitized because of the existence of C1 and C3. Thus IC testing based attack will not work for this camouflaged circuit. If the attacker directly brute force searches all possible assignments for C1, C2, and C3, the complexity would be 3^3. Moreover, the complexity grows exponentially with the number of camouflaged gates.

However, it is not necessary to resolve all the camouflaged gates simultaneously even in brute force attack. In fact, an attacker can resolve the functionalities of camouflaged gates gradually, starting with the gates that are relatively easier to attack. The circuit partition based attack is based on this observation where the attacker first partitions the camouflaged gates into multiple maximum fan-in cones (MFIC) and attacks each MFIC separately. Maximum fan-in cone that is rooted at a primary output (PO) is defined as: $MFIC_{\rm PO} = \{{\rm G}i | \exists\ path,\ {\rm G}i \rightarrow {\rm PO}\}$, where G$i$ is the logic gates in the circuit. The function of an MFIC can be tested directly by feeding the related primary inputs and observing the corresponding primary output. Other gates outside the MFIC, no matter whether they are camouflaged or not, will not be needed to resolve the camouflaged gates in this MFIC. Therefore, the attacker can attack (for example, by brute force searching) each MFIC to reduce the attack complexity of circuit camouflaging.

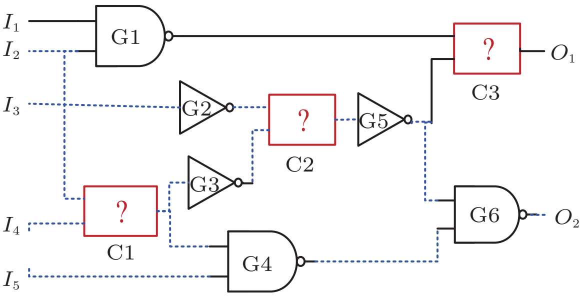


Fig.7. Motivational example of circuit partition based attack. Each of C1, C2 and C3 can be one of {NAND, NOR, XOR}.

For example, in Fig.7, $MFIC_{O_1}$ = {G1, C1, G2, G3, C2, G5, C3}, and $MFIC_{O_2}$ = {C1, G2, G3, G4, C2, G5, G6}. Without circuit partition, the attacker needs to brute force search 3^3 possible assignments for C1, C2, and C3 in the entire circuit. However, by applying circuit partition first, the attacker can brute force search the 3^2 possible assignments for C1 and C2 first in sub-circuit $MFIC_{O_2}$ (marked with dotted lines). After C1 and C2 are resolved, C3 can be resolved by IC testing based attack easily[12].

Fig.8 demonstrates the attack flow when the circuit partition based attack is combined with the brute force attack and IC testing based attack. This approach inherits the advantages and evades the disadvantages of IC testing based attack and brute force attack. A similar strategy can also be applied to combine circuit partition based attack with SAT-based attack.

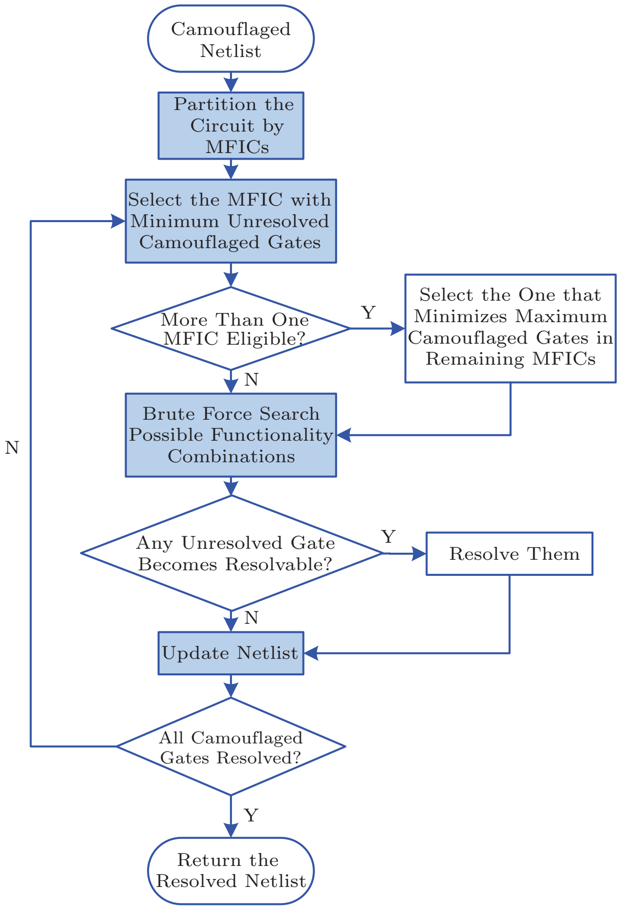

Camouflaged Netlist
Partition the Circuit by MFICs
Select the MFIC with Minimum Unresolved Camouflaged Gates
More Than One MFIC Eligible?
Y
Select the One that Minimizes Maximum Camouflaged Gates in Remaining MFICs
N
Brute Force Search Possible Functionality Combinations
N
Any Unresolved Gate Becomes Resolvable?
Y
Resolve Them
N
Update Netlist
All Camouflaged Gates Resolved?
Y
Return the Resolved Netlist

Fig.8. Attack flow of circuit partition based attack when combined with the brute force attack and the IC testing based attack.

3.5 Discussions on De-Camouflaging Attacks

IC testing based attack is scalable to large circuits because the attack complexity grows only linearly with the number of camouflaged gates. However, it is effective only when the inputs of the camouflaged gate are controllable from primary inputs, and at the same time the corresponding output is observable from primary outputs. Brute force de-camouflaging attack guarantees to resolve all the camouflaged gates, but it suffers from scalability problem, because its complexity grows exponentially with the number of camouflaged gates. For SAT-based de-camouflaging attack, despite being effective for most of the circuits, its effectiveness is limited when it faces hard-SAT designs such as multipliers and cryptographic ciphers, and it cannot guarantee to get the same circuit design with the original one, as there can be more than one assignment that makes the design perform the same function with the original one. Circuit partition based de-camouflaging attack is simple and powerful. It can be applied prior to other de-camouflaging attacks (such as brute force attack and SAT-based attack) in order to reduce their time complexity, or combine with IC testing based attack as a complementation. Moreover, no extra time or space is needed for circuit partition based attack. Each of the de-camouflaging attack has its strength and weakness. In real attack scenarios, the attacker can apply/combine the above attacks flexibly to achieve efficient de-camouflaging attacks. These powerful de-camouflaging attacks have brought a serious concern to the effectiveness of circuit camouflaging as the only proactive method to protect IP from RE-based attacks. In Section 4, we will review the proposed mitigation mechanisms to countermeasure these attacks.

4 Shields of Camouflaging

On one hand, due to the additional area, power and delay overheads brought by camouflaged cells[4], it is not practical to camouflage all conventional logic gates with camouflaged cells. On the other hand, as demonstrated in Section 3, randomly selecting gates to camouflage will result in the camouflaged circuits being vulnerable to various de-camouflaging attacks. To simultaneously thwart the de-camouflaging attacks and meet the performance overhead constraints, multiple camouflaging strategies have been proposed as the shields against de-camouflaging.

4.1 Clique-Based Camouflaging

Recall that IC testing based attack resolves the functionality of one camouflaged gate by controlling primary inputs to justify the inputs of the gate to certain pattern and sensitizing corresponding output to observe from a primary output. Therefore, intuitively, to thwart IC testing based attack, one should ensure that for each camouflaged gate, either its inputs cannot be justified from primary inputs, or its output cannot be sensitized to any primary output.

Based on this observation, Rajendran *et al.* defined that camouflaged gates that do not have any circuit path interfere with other camouflaged gates as isolated camouflaged gates, and two camouflaged gates are interfered if one lies on a path between the other and an output, and/or they converge at some other gate[12]. For example, in Fig.1, C1 and C2 are isolated camouflaged gates, and in Fig.7, the three camouflaged gates C1, C2 and C3 are interfered. If gate G4 is also camouflaged, it will not interfere with C3, but it will interfere with C2 for their outputs meet at G6.

Rajendran *et al.* further proposed enhanced IC camouflaging to judiciously select to-be-camouflaged candidate gates, based on the requirement that only interfered camouflaged gates will be selected[12]. In this way, the camouflaged gates cannot be resolved by IC-testing based attack for each camouflaged gate can only be resolved when the functionalities of other camouflaged gates are known, forming circular dependencies among the camouflaged gates.

For example, in Fig.9, camouflaged gates C1 and C2 are interfered. Neither of C1 and C2 can be resolved by IC testing based attack: C1's output cannot be observed from PO O_1 without resolving C2 first, while C2's output cannot be observed from O_1 without knowing the functionality of C1.

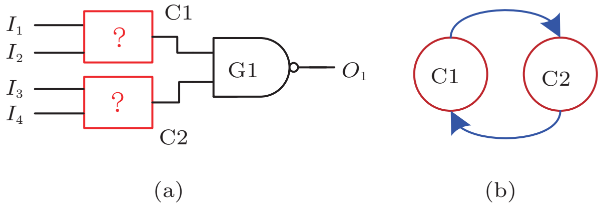


Fig.9. Circuit camouflaging example[12]. (a) Camouflaged gates C1 and C2 being interfered. (b) Interference graph of C1 and C2.

4.2 Equivalent Class Based Camouflaging

Circuit partition based attack partitions camouflaged gates into multiple sub-circuits to target each sub-circuit individually. Therefore, to thwart circuit partition based attack, one should keep the camouflaged gates together to disable the partition. In other words, no matter which eligible sub-circuit the attacker selects to attack in, one should make sure that all the camouflaged gates will belong to that sub-circuit, and then the attack complexity will not be reduced.

Recall the definition of $MFIC_{\rm PO}$ in Section 3, given that an attacker has only access to the PIs and POs of an unpackaged functional IC, $MFIC_{\rm PO}$ will be the minimum unit of sub-circuit whose function can be tested separately. In other words, when camouflaged gates cannot be partitioned by any $MFIC_{\rm PO}$, they will not be able to be partitioned by any sub-circuit that can be tested individually from a functional IC.

[4] A generic {NAND, NOR, XOR} camouflaged cell requires at least 12 transistors, along with a large area of metal connections[12]. The area overheads range from 50% to 200%[43] compared with the 4-T NAND, 4-T NOR and 8-T XOR gates.

According to this observation, Wang *et al.*[13,19] proposed a gate classification method, which classifies gates into the same equivalent class by the set of $MFIC_{\rm PO}$ that they belong to. Specifically, gates that belong to exactly the same set of $MFIC_{\rm PO}$ are classified to the same equivalent class. Then they proposed to select to-be-camouflaged gates from the same equivalent class. The formal definitions for the gate classification method are as follows.

Definition 1. *For a gate G, $MFICS_{\rm G}$ is the set of $MFIC_{\rm PO}$ that G belongs to. Formally, $MFICS_{\rm G} = \{MFIC_{{\rm PO}_i} | G \in MFIC_{{\rm PO}_i}\}$.*

Definition 2. *Gates that belong to the same set of $MFIC_{\rm PO}$ are classified to the same class. Formally, gates $G1$, $G2$, $\cdots$, Gn are partitioned to the same class C if and only if $MFICS_{\rm G1} = MFICS_{\rm G2} = \cdots = MFICS_{{\rm G}n}$.*

According to the definitions, in Fig.10,
$MFIC_{O_1}$={G1, G2, G3, G5},
$MFIC_{O_2}$={G2, G3, G4, G6}, thus
$MFICS_{\rm G1}$=$MFICS_{\rm G5}$={$MFIC_{O_1}$},
$MFICS_{\rm G2}$=$MFICS_{\rm G3}$={$MFIC_{O_1}$, $MFIC_{O_2}$},
$MFICS_{\rm G4}$=$MFICS_{\rm G6}$={$MFIC_{O_2}$}.

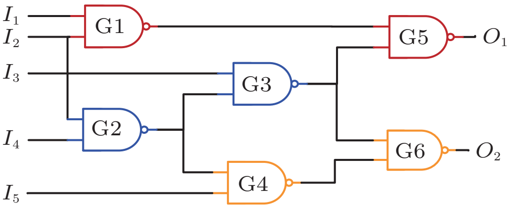


Fig.10. Gates that belong to the same equivalent class are marked in the same color[19].

Gates {G1, G5}, {G2, G3}, and {G4, G6} will be classified into equivalent class C1, C2, and C3, respectively.

Therefore, when selecting to-be-camouflaged gates from the same equivalent class, no matter which $MFIC_{\rm PO}$ an attacker selects to attack in, all camouflaged gates belong to $MFIC_{\rm PO}$, or none of the camouflaged gates belong to $MFIC_{\rm PO}$. The attacker will not be able to partition the camouflaged gates into small sub-circuits to attack individually.

For example, in Fig.10, when G2 and G3 from equivalent class C2 are selected for camouflaging, if the attacker selects to attack in $MFIC_{O_1}/MFIC_{O_2}$, both G2 and G3 belong to $MFIC_{O_1}/MFIC_{O_2}$. And when G1 and G5 from equivalent class C1 are selected for camouflaging, if attacking in $MFIC_{O_1}$, both G1 and G5 are in $MFIC_{O_1}$, while if attacking in $MFIC_{O_2}$, neither of G1 and G5 in $MFIC_{O_2}$.

Note that simply selecting gates to camouflage from one $MFIC_{\rm PO}$ will not contribute to hampering circuit partition based attack, for example, {G1, G2, G3, G5} $\in MFIC_{O_1}$, the attacker can attack {G2, G3} in $MFIC_{O_2}$ first, and then attack {G1, G5} in $MFIC_{O_1}$.

4.3 Thwart SAT Based De-Camouflaging

The key factor to the complexity of SAT-based decamouflaging attack is the power (discriminating ability) of DIs, namely, how many incorrect assignments one DI can prune from the possible assignment set. Therefore, to thwart SAT-based attack, one should try to minimize the power of DIs, and ideally, make one DI able to prune only one incorrect assignment. Given that there are initially exponential amount of possible assignments: M^N where M is the number of possible functionalities for each camouflaged gate and N is the number of camouflaged gates in the circuit, $(M^N - 1)$ DIs will be needed to prune all incorrect assignments. There exists theoretical analysis that supports this observation[21]. And based on this, two camouflaging strategies have been proposed in [20-21].

4.3.1 AND-Tree Camouflaging

The AND-tree camouflaging scheme[21] targets AND-tree structure for camouflaging, which is made up of AND logic gates. As shown in Fig.11 and Table 3, by camouflaging the inputs of AND-tree with {INV, BUF} camouflaged cells, the power (discriminating ability) of one DI will be limited to excluding only one incorrect assignment; thus exponential number of DIs will be needed to prune all incorrect assignments and get the correct assignment (backgrounded in yellow in Table 3). This property is determined by the special AND-tree structure and the {INV, BUF} camouflaged cell, which has been analyzed in detail[21].

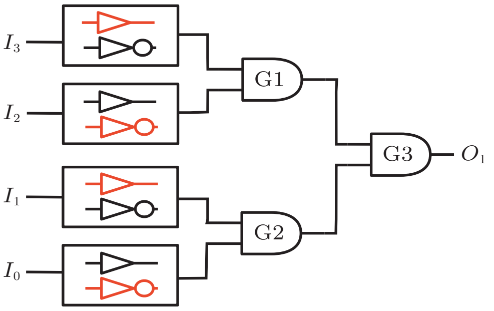


Fig.11. Camouflaging inputs of AND-tree structure camouflaged with {INV, BUF} camouflaged cells. Gates in red denote the actual functionality of camouflaged gates[21].

Table 3. Each Incorrect DI Only Pruning One Incorrect Assignment from the Possible Assignment Set

Possible Assignment	DI															
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
INV INV INV INV	×	√	√	√	√	√	√	√	√	√	×	√	√	√	√	√
INV INV INV BUF	√	×	√	√	√	√	√	√	√	√	×	√	√	√	√	√
INV INV BUF INV	√	√	×	√	√	√	√	√	√	√	×	√	√	√	√	√
INV INV BUF BUF	√	√	√	×	√	√	√	√	√	√	×	√	√	√	√	√
INV BUF INV INV	√	√	√	√	×	√	√	√	√	√	×	√	√	√	√	√
INV BUF INV BUF	√	√	√	√	√	×	√	√	√	√	×	√	√	√	√	√
INV BUF BUF INV	√	√	√	√	√	√	×	√	√	√	×	√	√	√	√	√
INV BUF BUF BUF	√	√	√	√	√	√	√	×	√	√	×	√	√	√	√	√
BUF INV INV INV	√	√	√	√	√	√	√	√	×	√	×	√	√	√	√	√
BUF INV INV BUF	√	√	√	√	√	√	√	√	√	×	×	√	√	√	√	√
BUF INV BUF INV	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
BUF INV BUF BUF	√	√	√	√	√	√	√	√	√	√	×	×	√	√	√	√
BUF BUF INV INV	√	√	√	√	√	√	√	√	√	√	×	√	×	√	√	√
BUF BUF INV BUF	√	√	√	√	√	√	√	√	√	√	×	√	√	×	√	√
BUF BUF BUF INV	√	√	√	√	√	√	√	√	√	√	×	√	√	√	×	√
BUF BUF BUF BUF	√	√	√	√	√	√	√	√	√	√	×	√	√	√	√	×

Note: √ represents the correct output, and × represents the incorrect output.

Note that there exists one critical DI (such as 1010 in Table 3), under which all incorrect assignments will be pruned, and this can make SAT attacker luckily get the correct assignment with this single DI. However, the probability of finding this critical DI is only $\frac{1}{M^N-1}$.

4.3.2 Minterm Perturbation Based Camouflaging

Previous approaches usually camouflage certain conventional logic gates with camouflaged cells, and configure the camouflaged cells to perform the same functionalities with them. Differently, as shown in Fig.12, CamoPerturb modifies/perturbs one minterm of the original design ($\mathrm{C}_{\mathrm{orignal}}$) by replacing one conventional logic gate with another conventional logic gate, which produces the perturbed circuit ($\mathrm{C}_{\mathrm{perturb}}$). To "correct" the modified minterm/function, additional camouflaged module CamoFix is designed. The output of $\mathrm{C}_{\mathrm{orignal}}$ is XORed by CamoFix and $\mathrm{C}_{\mathrm{perturb}}$[20].

Logic gates in CamoFix are camouflaged by {INV, BUF} cells. To reverse engineer the IC, the attacker has to resolve the functionalities of camouflaged cells in CamoFix. Moreover, CamoFix is judiciously designed so that each DI found by SAT solver can only prune one incorrect assignment for the camouflaged gates (which is similar to the result in Table 3). Therefore, the SAT solver needs to be called exponential number of times to find exponential number of DIs to prune all incorrect assignments, exponentially complexing the decamouflaging process.

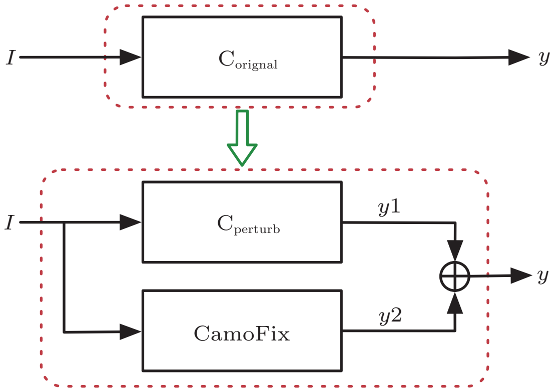

I
$\mathrm{C}_{\mathrm{orignal}}$
y
I
$\mathrm{C}_{\mathrm{perturb}}$
$y1$
y
CamoFix
$y2$

Fig.12. Camouflaging original design $\mathrm{C}_{\mathrm{orignal}}$ with $\mathrm{C}_{\mathrm{perturb}}$ and CamoFix. $\mathrm{C}_{\mathrm{perturb}}$ changes one minterm of $\mathrm{C}_{\mathrm{orignal}}$, CamoFix corrects the minterm, and y denotes the output signal.

4.4 Discussions on Defending Strategies

Most defending strategies (such as clique-based camouflaging, equivalent class based camouflaging, and AND-tree camouflaging) pose constraints to the to-be-camouflaged gate selection process to thwart certain decamouflaging attacks. The drawbacks would be that the posed constraints will reduce the gate selection space for camouflaging, thus reducing the performance optimization space. In addition, AND-tree camouflaging will be limited by the size of the largest AND-tree (the authors of [21] proposed to insert additional AND-tree structure in this case; however, this will bring additional performance overheads). Also, an ideal camouflaging strategy should combine the ideas of multiple strategies to thwart all possible de-camouflaging attacks.

CamoPerturb tries to insert some additional camouflaged cells to the original circuit, instead of replacing some same-function conventional logic gates. It stealthily changes one minterm of a circuit module by replacing one conventional logic gate with another conventional logic gate, and then re-corrects the minterm with one additional specially designed camouflaged module (CamoFix). In this way, the attacker will not be aware of which gates have been replaced in the original circuit module, and how the function of the module has been changed. Thus, the attacker will have to resolve the functionalities of camouflaged gates in the camouflaged module. Removal attack can be a concern for CamoPerturb for a tricky attacker may directly remove the CamoFix module and try to resolve the changed function, or he/she may just redesign the CamoFix module given that its black-box function is known and the scale is small. Also, CamoPerturb suffers from the same "critical DI" problem as the AND-tree camouflaging which can sometimes make the SAT attacker luckily prune all incorrect assignments with only one DI. The need for special logic gates (CamoPerturb needs to replace one logic gate that affects only one minterm) can also pose restrictions on its flexibility.

5 Challenges and Future Opportunities

First, to make this promising circuit camouflaging technique practical in thwarting RE, there still exist many challenges. Perhaps the most significant one is that the overhead in applying CMOS camouflaged cells can be rather high in terms of circuit timing, power consumption, and area, especially when a high level of protection is needed. How to reduce the overhead incurred by circuit camouflaging would continue to be an urgent need. With the development of new technology, developing doping-based and emerging device based camouflaging cells can be promising to solve this problem.

The second challenge is how to design countermeasures against the newly proposed SAT-based de-camouflaging attacks which are very powerful because they leverage the well-developed SAT solvers. Such attacks can effectively exclude incorrect functionality combinations of the camouflaged gates, successfully bypassing the exponential complexity of brute force. Although there is no guarantee that this can be effective in de-camouflaging all circuits and researchers have proposed many countermeasures against it, the type of SAT-based attacks has already become a serious threat to circuit camouflaging.

Third, it will be interesting to study intrinsic reconfigurable properties of emerging devices and how they can be utilized for circuit camouflaging. Many studies have been conducted on this topic. Simulation results have shown promise in using these emerging device based cells for circuit camouflaging. However, there still exist many hurdles in fabricating such emerging device based camouflaged cells and integrating them into a real design.

Since the concept of circuit camouflage was introduced in 2012[12-13], researchers from industry and academia have played both roles of the attacker and defender in the game of spear and shield to advance circuit camouflaging techniques. As we have surveyed, IC testing based attack[12], brute force attack[12], SAT-based attack[14-16], and circuit partition based attack[17] are the best-known de-camouflaging spears to restore the original design from a camouflaged circuit layout. On the other hand, countermeasures such as clique-based selection[12], multiplexer-based camouflaging[18-19], tree-based camouflaging[21], and CamoPerturb[20] were proposed to make the shield of circuit camouflaging stronger in order to defeat RE-based attacks. It is our belief that this race of spear and shield will eventually lead us to a practical, effective, and robust defense system against RE-based attacks.

6 Other Proactive IP Protection Methods

In addition to circuit camouflaging which thwarts reverse engineering based attack, in the literature, many other approaches have been proposed for IP protection. The law enforcement based prevention methods and digital watermarking/fingerprinting based deterrent methods for IP protection can be found in previous work[1]. In this section, to give the readers a more comprehensive understanding of IP protection, we briefly review other proactive techniques which include split manufacturing to hamper foundry overbuilding[51-56], logic locking to thwart supply-chain IP piracy[57-64], and also PUF security primitive which can be integrated to such security schemes[65-71].

Split manufacturing is a strategy to trade off the secure but expensive in-house manufacturing for the more affordable but insecure off-shore fabrication. Specifically, in split manufacturing, the front end of line (FEOL) is outsourced and the back end of line (BEOL) is manufactured by trusted foundry. Thus, the BEOL connection design will not be completely available to untrusted foundry, making it difficult to reconstruct the entire design. Subgraph reconstruction and physical design optimization metrics based attacks have also been proposed to reconstruct the BEOL connection[54]. However, split manufacturing cannot prevent supply-chain attackers from IP piracy or hardware trojan insertions. It cannot prevent end-user reverse engineering either.

Logic locking is a netlist-level technique that locks the function of a circuit. By inserting "key-gates" (usually XOR, XNOR or LUT) to the circuit, only correct keys can enable the correct functionality.

In one implementation, some additional XOR/XNOR key-gates are inserted to the original circuit[57-58]. One input of such a key-gate is an internal signal in the circuit, and the other input comes from the PIs to act as the key. For XOR key-gate, when the key is 1, the XOR gate will perform as an inverter, and when the key is 0, it will work as a buffer. Only with correct keys can the circuit function correctly.

Another possible implementation is to insert reconfigurable logical barriers to separate the combinational network into two parts, with all inputs in one part and all outputs in the other part[59]. Without the correct key, the barriers will mislead the dataflow, resulting in incorrect functions. In sequential circuit, additional states are added to finite state machine (FSM) for the same purpose[60-61]. FSM stays in an obfuscated mode initially, and only a certain input sequence can make FSM enter normal mode states. The input sequence performs as a key in this case.

In logic locking, the keys must be stored in secure and tamper-proof memories, and distribution framework must be established for the IP designer to securely unlock each IC. Hardware security primitives such as physical unclonable function (PUF) can be integrated in logic locking to make the key of each instance of IC unique. This gives better control of the number of ICs being fabricated (against overbuilding) and enables the trace of IC, once IC infringement is detected.

It is worth mentioning that there is a close relationship between logic locking and IC camouflaging. They both hide the function of IC to make the design of IC become difficult to understand. The differences would be that logic locking hides the function by the keys, while circuit camouflaging relies on the same-look different-function camouflaged cells. Previous work has demonstrated that camouflaged circuit can be transformed to its security-equivalent logic locked netlist and vice versa[72]. As such, the concepts of de-camouflaging attacks and the defense countermeasures surveyed in this paper are also applicable to logic locking.

7 Conclusions

Hardware is the root of software, network, and system security. But the security and trust of hardware design faces threats from reverse engineering (RE) based attacks. In this paper, we surveyed the newly proposed IC camouflaging technique which defeats RE-based attacks by judiciously replacing some conventional logic gates with same-look different-function camouflaged cells.

Currently, there are four types of camouflaged cells. The original and well-studied true/dummy contact based camouflaged cell incurs non-trivial performance overheads. Similarly SRAM-based camouflaged cell has even larger overhead than true/dummy contact based camouflaged cell while providing unique reconfigurable property. Doping-based camouflaged cell is relative lightweight but has high technology requirements. Emerging device based camouflaged cell has many advantages over CMOS approaches; however most of these emerging devices are still in the simulation phase. Although the overhead of camouflaged cell was known as one of the major concerns of circuit camouflaging technique from the very beginning, how to reduce the performance overheads incurred by circuit camouflaging still remains as an open problem. With the development of new technology, developing doping-based and emerging device based camouflaging cells can be promising to solve this problem.

We provided a detailed review of the spear and shield race between de-camouflaging attacks and the corresponding countermeasures. IC testing based attack, brute force attack and circuit partition based attack have received many attentions. Effective countermeasures such as clique-based camouflaging and equivalent class based camouflaging have been proposed to successfully defeat these attacks. However, the recently proposed SAT-based attack leverages the power of the off-the-shelf SAT solvers, and poses great threats to the effectiveness of circuit camouflaging. Despite several attempts such as CamoPerturb and AND-tree camouflaging that have limited success against SAT-based attacks, how to effectively thwart SAT-based attacks remains as another open problem that needs to be solved urgently. Utilizing available or inserting hard-SAT circuit structures can be promising to thwart SAT-based attack. For future IC camouflaging techniques, resilience against SAT-based attack should be one important evaluation metric.

Circuit camouflaging is a promising method against RE-based IP piracy. The race of spear and shield between de-camouflaging attacks and their countermeasures will continue to push circuit camouflaging to the next level. The goal of this survey article is to attract more researchers to this interesting area and to have them involved in the development of circuit camouflaging technique that has small overhead and high resilience against various de-camouflaging attacks.

References

[1] Qu G, Potkonjak M. Intellectual Property Protection in VLSI Designs: Theory and Practice. Kluwer Academic Publishers, 2003.

[2] Rostami M, Koushanfar F, Karri R. A primer on hardware security: Models, methods, and metrics. *Proceedings of the IEEE*, 2014, 102(8): 1283-1295.

[3] Jin Y E. Introduction to hardware security. *Electronics*, 2015, 4(4): 763-784.

[4] Lv Y Q, Zhou Q, Cai Y C, Qu G. Trusted integrated circuits: The problem and challenges. *Journal of Computer Science and Technology*, 2014, 29(5): 918-928.

[5] Quadir S E, Chen J L, Forte D, Asadizanjani N, Shahbazmohamadi S, Wang L, Chandy J, Tehranipoor M. A survey on chip to system reverse engineering. *ACM Journal on Emerging Technologies in Computing Systems* (*JETC*), 2016, 13(1): Article No. 6.

[6] Adee S. The hunt for the kill switch. *IEEE Spectrum*, 2008, 45(5): 34-39.

[7] Qu G, Potkonjak M. Fingerprinting intellectual property using constraint-addition. In *Proc. the 37th Annual Design Automation Conf.*, June 2000, pp.587-592.

[8] Dunbar C, Qu G. Satisfiability Don't Care condition based circuit fingerprinting techniques. In *Proc. the 20th Asia and South Pacific Design Automation Conf.*, January 2015, pp.815-820.

[9] Kahng A B, Lach J, Mangione-Smith W H, Mantik S, Markov I L, Potkonjak M, Tucker P, Wang H, Wolfe G. Constraint-based watermarking techniques for design IP protection. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 2001, 20(10): 1236-1252.

[10] Dunbar C, Qu G. A practical circuit fingerprinting method utilizing observability Don't Care conditions. In *Proc. the 52nd ACM/EDAC/IEEE Design Automation Conf.*, June 2015.

[11] Chow L W, Baukus J P, Wang B J, Cocchi R P. Camouflaging a standard cell based integrated circuit: US Patent 8151235, April 3, 2012. http://www.freepatentsonline.com/8151235.html, Dec. 2017.

[12] Rajendran J, Sam M, Sinanoglu O, Karri R. Security analysis of integrated circuit camouflaging. In *Proc. ACM SIGSAC Conf. Computer & Communications Security*, November 2013, pp.709-720.

[13] Wang X Y, Gao M Z, Zhou Q, Cai Y C, Qu G. Gate camouflaging-based obfuscation. In *Hardware Protection through Obfuscation*, Forte D, Bhunia S, Tehranipoor M M (eds.), Springer, 2017, pp.89-102.

[14] Subramanyan P, Ray S, Malik S. Evaluating the security of logic encryption algorithms. In *Proc. IEEE Int. Symp. Hardware Oriented Security and Trust*, May 2015, pp.137-143.

[15] Massad M E, Garg S, Tripunitara M V. Integrated circuit (IC) decamouflaging: Reverse engineering camouflaged ICs within minutes. In *Proc. the 22nd Annual Network and Distributed System Security Symp.*, February 2015.

[16] Liu D, Yu C X, Zhang X Y, Holcomb D. Oracle-guided incremental SAT solving to reverse engineer camouflaged logic circuits. In *Proc. Design Automation & Test in Europe Conf. Exhibition*, March 2016, pp.433-438.

[17] Wang X Y, Zhou Q, Cai Y C, Qu G. Is the secure IC camouflaging really secure? In *Proc. IEEE Int. Symp. Circuits and Systems*, May 2016, pp.1710-1713.

[18] Liu B, Wang B. Embedded reconfigurable logic for ASIC design obfuscation against supply chain attacks. In *Proc. Design Automation & Test in Europe Conf. Exhibition*, March 2014.

[19] Wang X Y, Jia X T, Zhou Q, Cai Y C, Yang J L, Gao M Z, Qu G. Secure and low-overhead circuit obfuscation technique with multiplexers. In *Proc. the 26th Edition on Great Lakes Symp. VLSI*, May 2016, pp.133-136.

[20] Yasin M, Mazumdar B, Sinanoglu O, Rajendran J. Camoperturb: Secure IC camouflaging for minterm protection. In *Proc. the 35th Int. Conf. Computer-Aided Design*, November 2016, Article No. 29.

[21] Li M, Shamsi K, Meade T, Zhao Z, Yu B, Jin Y E, Pan D Z. Provably secure camouflaging strategy for IC protection. In *Proc. the 35th Int. Conf. Computer-Aided Design*, November 2016, Article No. 28.

[22] Chow L W, Baukas J P, Clark Jr W M. Integrated circuits protected against reverse engineering and method for fabricating the same using an apparent metal contact line terminating on field oxide. US Patent 20020096776, Jul. 25, 2002. http://www.freepatentsonline.com/20020096776.pdf, Jan. 2018.

[23] Baukus J P, Chow L W, Clark Jr W M. Method and apparatus using silicide layer for protecting integrated circuits from reverse engineering. US Patent 6117762, Sept. 12, 2000. http://www.freepatentsonline.com/6117762.html, Dec. 2017.

[24] Cocchi R P, Baukus J P, Wang B J, Chow L W, Ouyang P. Building block for a secure CMOS logic cell library. US Patent 8111089, Feb. 7, 2012. http://www.freepatentsonline.com/8111089.html, Dec. 2017.

[25] Chow L W, Clark Jr W M, Harbison G J, Baukus J P. Conductive channel pseudo block process and circuit to inhibit reverse engineering. US Patent 7049667, May 23, 2006. http://www.freepatentsonline.com/8258583.html, Dec. 2017.

[26] Clark Jr W M, Chow L W, Harbison G, Ouyang P. Programmable connection and isolation of active regions in an integrated circuit using ambiguous features to confuse a reverse engineer. US Patent 8564073, Oct. 22, 2013. http://www.freepatentsonline.com/8564073.html, Dec. 2017.

[27] Clark Jr W M, Baukus J, Chow L W. Implanted hidden interconnections in a semiconductor device for preventing reverse engineering. US Patent 7166515, Jan. 23, 2007. http://www.freepatentsonline.com/7166515.html, Dec. 2017.

[28] Liu B, Wang B. Reconfiguration-based VLSI design for security. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2015, 5(1): 98-108.

[29] Becker G T, Regazzoni F, Paar C, Burleson W P. Stealthy dopant-level hardware trojans: Extended version. *Journal of Cryptographic Engineering*, 2014, 4(1): 19-31.

[30] Chow L W, Clark Jr W M, Baukus J P. Covert transformation of transistor properties as a circuit protection method. US Patent 7217977, May 15, 2007. http://www.freepatentsonline.com/7217977.html, Dec. 2017.

[31] Ma K S, Liu H C, Xiao Y, Zheng Y, Li X Q, Gupta S K, Xie Y, Narayanan V. Independently-controlled-gate FinFET 6T SRAM cell design for leakage current reduction and enhanced read access speed. In *Proc. IEEE Computer Society Annual Symp. VLSI*, July 2014, pp.296-301.

[32] Sedighi B, Hu X S, Nahas J J, Niemier M. Nontraditional computation using beyond-CMOS tunneling devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2014, 4(4): 438-449.

[33] Lin S, Kim Y B, Lombardi F. CNTFET-based design of ternary logic gates and arithmetic circuits. *IEEE Trans. Nanotechnology*, 2011, 10(2): 217-225.

[34] Zhao P, Feenstra R M, Gu G, Jena D. SymFET: A proposed symmetric graphene tunneling field-effect transistor. *IEEE Trans. Electron Devices*, 2013, 60(3): 951-957.

[35] Chua L. Memristor—the missing circuit element. *IEEE Trans. Circuit Theory*, 1971, 18(5): 507-519.

[36] Strukov D B, Snider G S, Stewart D R, Williams R S. The missing memristor found. *Nature*, 2008, 453(7191): 80-83.

[37] Roy K, Sharad M, Fan D L, Yogendra K. Computing with spin-transfer-torque devices: Prospects and perspectives. In *Proc. IEEE Computer Society Annual Symp. VLSI*, July 2014, pp.398-402.

[38] Chen A, Hu X S, Jin Y E, Niemier M, Yin X Z. Using emerging technologies for hardware security beyond PUFs. In *Proc. Design Automation & Test in Europe Conf. Exhibition*, March 2016, pp.1544-1549.

[39] Shamsi K, Jin Y. Security of emerging non-volatile memories: Attacks and defenses. In *Proc. the 34th IEEE VLSI Test Symp.*, April 2016.

[40] Arafin M T, Qu G. RRAM based lightweight user authentication. In *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, November 2015, pp.139-145.

[41] Arafin M T, Dunbar C, Qu G, McDonald N, Yan L. A survey on memristor modeling and security applications. In *Proc. the 16th Int. Symp. Quality Electronic Design*, March 2015, pp.440-447.

[42] Winograd T, Salmani H, Mahmoodi H, Gaj K, Homayoun H. Hybrid STT-CMOS designs for reverse-engineering prevention. In *Proc. the 53rd ACM/EDAC/IEEE Design Automation Conf.*, June 2016.

[43] Bi Y, Shamsi K, Yuan J S, Gaillardon P E, De Micheli G, Yin X Z, Hu X S, Niemier M, Jin Y. Emerging technology-based design of primitives for hardware security. *ACM Journal on Emerging Technologies in Computing Systems*, 2016, 13(1): Article No. 3.

[44] Bi Y, Gaillardon P E, Hu X S, Niemier M, Yuan J S, Jin Y. Leveraging emerging technology for hardware security-case study on silicon nanowire fets and graphene symfets. In *Proc. the 23rd IEEE Asian Test Symp.*, November 2014, pp.342-347.

[45] Shamsi K, Wen W J, Jin Y. Hardware security challenges beyond CMOS: Attacks and remedies. In *Proc. IEEE Computer Society Annual Symp. VLSI*, July 2016, pp.200-205.

[46] Bobba S, De Marchi M, Leblebici Y, De Micheli G. Physical synthesis onto a sea-of-tiles with double-gate silicon nanowire transistors. In *Proc. the 49th Annual Design Automation Conf.*, June 2012, pp.42-47.

[47] Suzuki D, Natsui M, Ikeda S, Hasegawa H, Miura K, Hayakawa J, Endoh T, Ohno H, Hanyu T. Fabrication of a nonvolatile lookup-table circuit chip using magneto/semiconductor-hybrid structure for an immediate-power-up field programmable gate array. In *Proc. Symp. VLSI Circuits*, June 2009, pp.80-81.

[48] Mahmoodi H, Lakshmipuram S S, Arora M, Asgarieh Y, Homayoun H, Lin B, Tullsen D M. Resistive computation: A critique. *IEEE Computer Architecture Letters*, 2014, 13(2): 89-92.

[49] Rajendran J, Sinanoglu O, Karri R. VLSI testing based security metric for IC camouflaging. In *Proc. IEEE Int. Test Conf.*, September 2013.

[50] Lee H K, Ha D S. HOPE: An efficient parallel fault simulator for synchronous sequential circuits. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 1996, 15(9): 1048-1058.

[51] Jarvis R W, Mcintyre M G. Split manufacturing method for advanced semiconductor circuits. US Patent 7195931, March 27, 2007. http://www.freepatentsonline.com/7195931.html, Dec. 2017.

[52] Imeson F, Emtenan A, Garg S, Tripunitara M V. Securing computer hardware using 3D integrated circuit (IC) technology and split manufacturing for obfuscation. In *Proc. the 22nd USENIX Conf. Security*, August 2013, pp.495-510.

[53] Valamehr J, Sherwood T, Kastner R, Marangoni-Simonsen D, Huffmire T, Irvine C, Levin T. A 3-D split manufacturing approach to trustworthy system development. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 2013, 32(4): 611-615.

[54] Rajendran J, Sinanoglu O, Karri R. Is split manufacturing secure? In *Proc. Design Automation & Test in Europe Conf. Exhibition*, March 2013, pp.1259-1264.

[55] Vaidyanathan K, Liu R Z, Sumbul E, Zhu Q L, Franchetti F, Pileggi L. Efficient and secure intellectual property (IP) design with split fabrication. In *Proc. IEEE Int. Symp. Hardware-Oriented Security and Trust*, May 2014, pp.13-18.

[56] Jagasivamani M, Gadfort P, Sika M, Bajura M, Fritze M. Split-fabrication obfuscation: Metrics and techniques. In *Proc. IEEE Int. Symp. Hardware-Oriented Security and Trust*, May 2014, pp.7-12.

[57] Roy J A, Koushanfar F, Markov I L. EPIC: Ending piracy of integrated circuits. In *Proc. Design Automation and Test in Europe*, March 2008, pp.1069-1074.

[58] Rajendran J, Pino Y, Sinanoglu O, Karri R. Security analysis of logic obfuscation. In *Proc. the 49th ACM/EDAC/IEEE Design Automation Conf.* June 2012, pp.83-89.

[59] Baumgarten A, Tyagi A, Zambreno J. Preventing IC piracy using reconfigurable logic barriers. *IEEE Design & Test of Computers*, 2010, 27(1): 66-75.

[60] Alkabani Y M, Koushanfar F. Active hardware metering for intellectual property protection and security. In *Proc. the 16th USENIX Security Symp.*, August 2007.

[61] Chakraborty R S, Bhunia S. HARPOON: An obfuscation-based SoC design methodology for hardware protection. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 2009, 28(10): 1493-1502.

[62] Zamanzadeh S, Jahanian A. Automatic netlist scrambling methodology in ASIC design flow to hinder the reverse engineering. In *Proc. the 21st IFIP/IEEE Int. Conf. Very Large Scale Integration*, October 2013, pp.52-53.

[63] Yasin M, Rajendran J J, Sinanoglu O, Karri R. On improving the security of logic locking. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 2016, 35(9): 1411-1424.

[64] Yasin M, Mazumdar B, Rajendran J J V, Sinanoglu O. Sarlock: SAT attack resistant logic locking. In *Proc. IEEE Int. Symp. Hardware Oriented Security and Trust*, May 2016, pp.236-241.

[65] Suh G E, Devadas S. Physical unclonable functions for device authentication and secret key generation. In *Proc. the 44th ACM/IEEE Design Automation Conf.*, June 2007, pp.9-14.

[66] Herder C, Yu M D, Koushanfar F, Devadas S. Physical unclonable functions and applications: A tutorial. *Proceedings of the IEEE*, 2014, 102(8): 1126-1141.

[67] Yamamoto D, Takenaka M, Sakiyama K, Torii N. A technique using PUFs for protecting circuit layout designs against reverse engineering. In *Proc. the 9th International Workshop on Security*, August 2014, pp.158-173.

[68] Wendt J B, Potkonjak M. Hardware obfuscation using PUF-based logic. In *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, November 2014, pp.270-277.

[69] Lee J W, Lim D, Gassend B, Suh G E, Van Dijk M, Devadas S. A technique to build a secret key in integrated circuits for identification and authentication applications. In *Proc. Symp. VLSI Circuits Digest of Technical Papers*, June 2004, pp.176-179.

[70] Zhang J L, Qu G, Lv Y Q, Zhou Q. A survey on silicon PUFs and recent advances in ring oscillator PUFs. *Journal of Computer Science and Technology*, 2014, 29(4): 664-678.

[71] Forte D, Bhunia S, Tehranipoor M M. Hardware Protection through Obfuscation. Springer, 2017.

[72] Yasin M, Sinanoglu O. Transforming between logic locking and IC camouflaging. In *Proc. the 10th Int. Design & Test Symp.*, December 2015.

Xue-Yan Wang received her B.S. degree in computer science and technology from Shandong University, Jinan, in 2013. She is currently pursuing her Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing. She is involved in research with the EDA (Electronic Design Automation) Laboratory. From 2015 to 2016, she was a visiting student in University of Maryland, College Park, MD, USA. Her current research interests include hardware security and efficient algorithms for VLSI physical design.

Qiang Zhou is a professor of the Department of Computer Science and Technology, Tsinghua University, Beijing. He received his B.S. degree in computer science and technology from the University of Science and Technology of China, Hefei, in 1983, M.S. degree in computer science and technology from Tsinghua University, Beijing, in 1986, and Ph.D. degree in control theory and control engineering from Chinese University of Mining and Technology, Beijing, in 2002. His research interests include VLSI layout theory and algorithms.

Yi-Ci Cai is a professor in the Department of Computer Science and Technology, Tsinghua University, Beijing. She received her B.S. degree in electronic engineering from Tsinghua University, Beijing, in 1983, M.S. degree in computer science and technology from Tsinghua University, Beijing, in 1986, and Ph.D. degree in computer science from the University of Science and Technology of China, Hefei, in 2007. Her research interests include design automation for VLSI integrated circuits algorithms and theory, power/ground distribution network analysis and optimization, high performance clock synthesis, and low power physical design.

Gang Qu received his Ph.D. degree in computer science from the University of California, Los Angeles, in 2000. He is currently a professor in the Department of Electrical and Computer Engineering and Institute for Systems Research, University of Maryland at College Park. He is also a member of the Maryland Cybersecurity Center and the Maryland Energy Research Center. Dr. Qu is the director of Maryland Embedded Systems and Hardware Security (MeshSec) Laboratory and the Wireless Sensors Laboratory. His primary research interests are in the area of embedded systems and VLSI CAD with focus on low power system design and hardware related security and trust. He studies optimization and combinatorial problems and applies his theoretical discovery to applications in VLSI CAD, wireless sensor network, bioinformatics, and cybersecurity.