Secure and Low-Overhead Circuit Obfuscation Technique with Multiplexers

Xueyan Wang, Xiaotao Jia, Qiang Zhou, and Yici Cai
CST Department, Tsinghua University
Beijing, China
{wangxueyan13, jxt11}@mails.tsinghua.edu.cn

Jianlei Yang
ECE Department, University of Pittsburgh
Pittsburgh, PA, USA
jiy64@pitt.edu, {zhouqiang, caiyc}@tsinghua.edu.cn

Mingze Gao, Gang Qu
ECE Department, University of Maryland
College Park, MD, USA
{mgao1, gangqu}@umd.edu

ABSTRACT

Circuit obfuscation techniques have been proposed to conceal circuit's functionality in order to thwart reverse engineering (RE) attacks to integrated circuits (IC). We believe that a good obfuscation method should have low design complexity and low performance overhead, yet, causing high RE attack complexity. However, existing obfuscation techniques do not meet all these requirements. In this paper, we propose a polynomial obfuscation scheme which leverages special designed multiplexers (MUXs) to replace judiciously selected logic gates. Candidate to-be-obfuscated logic gates are selected based on a novel gate classification method which utilizes IC topological structure information. We show that this scheme is resilient to all the known attacks, hence it is secure. Experiments are conducted on ISCAS 85/89 and MCNC benchmark suites to evaluate the performance overhead due to obfuscation.

1. INTRODUCTION

With the rapid development of embedded systems and the Internet of Things, application specific integrated circuits (IC) are playing a more and more important role in electronic products market. However, the hardware design intellectual-property (IP)[1, 2] in these circuits is facing severe threats from reverse engineering (RE) attacks, where an attacker analyzes a design and reproduces it with no or much less investment in research and development[3]. These low cost illegitimate products can bring security vulnerabilities to critical commercial and military systems, also, they can be sold at a much lower price, giving them an unfair competitive edge against the authenticated products.

Circuit obfuscation techniques have emerged as an effective countermeasure for RE attacks. These techniques seek to modify the design and implementation of a circuit in order to make it hard to interpret and hence increase the cost and complexity of RE attacks[4–12]. For an obfuscation technique to be effective, we argue that it must satisfy the following two requirements: First, the obfuscated circuit should be resilient to any potential attack and ideally make RE attack complexity exponential. Second, the obfuscation method should require a low design complexity (polynomial to the size of the circuit) and incur little overhead to circuit's performance. Clearly, when the first requirement violated, the obfuscated circuit can be reverse engineered with ease thus fail to achieve its goal of IP protection. When the second requirement violated, the design cost increases and/or the performance of the IP decreases, causing the IP to lose its value.

However, current representative circuit obfuscation methods, K_Based[9, 10] and R_Based[11, 12], do not meet one or both of the requirements. For example, [9–12] are vulnerable to at least one of the known attacks (ITA, CPA, SCA, BFA, which will be elaborated in Section 2.) thus being weak to thwart RE; [10] and [11] require exponential design time to select gates for obfuscation; [12] makes the obfuscated module 2.75x slower. Therefore, it is an urgent need to design a both secure and low-overhead obfuscation approach to thwart RE attacks. Our work in this paper is an attempt towards such an ideal obfuscation scheme with the following innovations:

- Leverage multiplexers (MUXs) and programmable camouflage connectors to replace certain conventional logic gates without changing the function of IC[1].
- Locate candidate to-be-obfuscated gates based on a novel gate classification method, which classifies logic gates to non-intersecting classes utilizing IC topological structure information.

The rest of paper is organized as follows: in Section 2, we survey the known attacks to obfuscated circuits. Section 3 elaborates the proposed multiplexer replacement based circuit obfuscation technique. Experimental results are presented in Section 4 and Section 5 concludes the paper.

[1]We are not the first to obfuscate circuits with MUXs, but we are the first to use programmable camouflage connectors to configure MUXs, and we judiciously guarantee the high-security of obfuscation.

GLSVLSI '16, May 18-20, 2016, Boston, MA, USA

DOI: http://dx.doi.org/10.1145/2902961.2903000

2. PRELIMINARY

2.1 Known Attacks to Obfuscated Circuits

Given an obfuscated circuit, a tricky attacker may apply the following attacks to resolve the original netlist of IC[2].

IC Testing based Attack (ITA). An attacker can leverage justification and sensitization principles in IC testing techniques to get the input-output pairs of an obfuscated gate to resolve its functionality[10, 11]. Justification is the process of justifying the inputs of a gate to a known value by controlling one or more of the IC's primary inputs (PIs), and sensitization is the process of sensitizing the value of a net to a primary output (PO) so that one can observe the value by setting all side inputs of gates in between to non-controlling values.

Circuit Partition Attack (CPA). An attacker can leverage the *'divide and conquer'* methodology to partition obfuscated gates into multiple subcircuits, then target each subcircuit individually. To ensure effectiveness, each subcircuit's function should be ensured not affected by other parts of the circuit, thus can be tested separately from an unpackaged functional IC.

Side Channel Attack (SCA). This indicates a group of attacks that utilize side channel information, such as power and timing, to break security primitives. For gate replacement based obfuscation, an attacker may resolve functionalities of obfuscated gates if such side channels are varied and observable by the attacker when they are configured to perform different functionalities[11].

Brute Force Attack (BFA). An attacker may directly brute force search possible functionality combinations of obfuscated gates[10, 11]. For each possible combination, the attacker will simulate to get the corresponding function of the IC, then compare it with a functional IC bought from the market. If they are the same, the attacker has found out the right combination and thus these obfuscated gates are resolved.

3. CIRCUIT OBFUSCATION WITH MULTIPLEXERS

3.1 Attack Model and Necessary Assumptions

Our discussions in this paper are based on the following assumptions:

1. The attacker seeks to identify the functionality of each obfuscated gate to reverse engineer the obfuscated IC.

2. The attacker is able to extract an obfuscated netlist with regular gates and obfuscated gates by state-of-art reverse engineering tools and techniques.

3. For an unpackaged functional IC bought from the market, the attacker only has access to the IC's PIs and POs.

To resolve the original circuit of an obfuscated one, a tricky attacker may first try to perform SCA and ITA considering their cost is relatively low. Once SCA and ITA infeasible, the attacker has to brute force search possible functionality combinations of obfuscated gates. In this case, he will try to minimize the needed brute force efforts by performing CPA. Thus only when an obfuscation approach resilient to SCA, ITA, and CPA, the BFA/RE complexity can be truly exponential to the number of gates obfuscated.

[2]We elaborate the attacks with the cases for R_Based obfuscation approaches, things are similar for the K_Based ones.

3.2 Configuring Multiplexers to Logic Gates

In our approach, multiplexers and camouflage connectors are combined as 'configurable logic units' to replace certain conventional logic gates. Any 2^m-by-1 multiplexer can be configured with $2 \cdot 2^m$ camouflage connectors to perform 2^{2^m} possible m-input 1-output boolean functions. Specifically, each input line of the MUX is connected to Vdd and Vss by two camouflage connectors, but only one is programmed to be a connection, the other one is programmed to be an isolation. For example in Fig.1, when input lines X1 X2 X3 X4 configured to be 1110, the MUX will perform the functionality of a NAND gate.

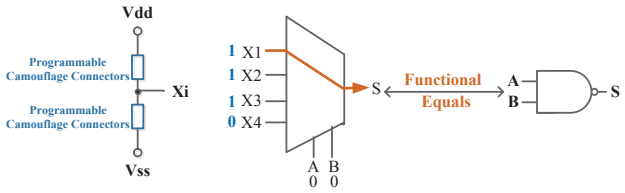

Figure 1: Configure MUX4x1 with programmable camouflage connectors to perform 16 possible 2-input 1-output boolean functions.

We adopt such a structure for two reasons. First, it will be more difficult to attack with more possible functionalities a configurable logic unit could perform. The MUX4x1 can be configured by our scheme to perform 16 possible 2-input 1-output boolean functions, while the configurable CMOS cells in [11] can only perform 3 possible functionalities which are those of {NAND, NOR, XOR}. Second, we only need to configure a MUX to perform certain boolean function one time and the overheads brought by configurable memory cells[12] can be non-trivial, thus we adopt the lightweight 'programmable camouflage connectors'[5, 13–16] instead.

3.3 Locating Candidate Gates based on IC Topological Information

We obfuscate circuits by replacing some conventional logic gates with MUXs that configured with programmable camouflage connectors. Considering the additional performance overhead, it is not realistic to replace all logic gates with MUXs. Thus the first step will be locating candidate gates, following the principle of maximizing attack complexity with minimum logic gates obfuscated.

3.3.1 A novel gate classification method

We first have the following definitions:

Definition 1. A gate is defined to be an *outGate* if its output signal is a PO of the circuit, otherwise, the gate is defined to be an *innerGate*.

Definition 2. The **M**aximum **F**an**I**n-**C**one rooted at a primary output PO_i, $MFIC_{PO_i}$, is the set of gates whose outputs will directly or indirectly feed into PO_i. Formally, $MFIC_{PO_i}$=$\{G|\exists path, G \rightarrow PO_i\}$[3].

Definition 3. For a gate G, $MFICS_G$ is the set of $MFIC_{PO}$ that G belongs to.

Formally, $MFICS_G$=$\{MFIC_{PO_i}|G \in MFIC_{PO_i}\}$.

Definition 4. Gates with exactly the same $MFICS$ are classified to the same class. Formally, gates $G1$, $G2$,

[3]We will also refer $MFIC_{PO}$ as the corresponding subcircuit, which can be easily differentiated according to contexts.

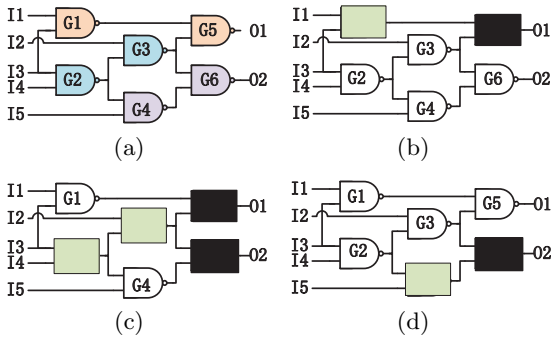

Figure 2: (a) Gates that classified to the same class are marked with the same color. (b) Select gates to obfuscate from class $C1$. The black rectangles indicate securely obfuscated gates which appear as black boxes to the attacker, and the green rectangles indicate candidates to be obfuscated in the selected class. (c) Select gates to obfuscate from class $C2$. (d) Select gates to obfuscate from class $C3$.

$\cdots$, Gn are partitioned to the same class C if and only if $MFICS_{G1} = MFICS_{G2} = \cdots = MFICS_{Gn}$. Class C's $MFICS$ is defined as the $MFICS$ of gates in C.

For example in Fig.2(a), gates $\{G1, G5\}$, $\{G2, G3\}$, and $\{G4, G6\}$ are classified to class $C1$, $C2$, and $C3$, respectively.

3.3.2 *Locating candidate gates*

Our approach is based on the proposed gate classification method to select gates to obfuscate from the same gate class. Specially, once a certain class selected, the *outGates* of the class's $MFICS$ are obfuscated to appear as black boxes to the attacker, blocking sensitizations of outputs of gates in the class to any PO. Then the designer can select the desired number of gates he(she) wants to obfuscate from the class. To minimize performance overhead, some principles can be applied when selecting gates to obfuscate (such as avoiding too many obfuscated gates congested in one single path). Fig.2(b), Fig.2(c) and Fig.2(d) show examples of selecting candidate gates to obfuscate from class $C1$, $C2$ and $C3$, respectively.

3.4 Obfuscation Analysis

Programmable camouflage connectors are leveraged to configure MUXs in our approach. It is infeasible to differentiate between connection and isolation of a camouflage connector for real-world RE attackers. On one hand, the camouflage connectors are carefully designed to be physically identical from the top view under optical or electron microscopy[13–16]. On the other hand, for chemical erosion and imaging based top-down reverse engineering, the camouflage connectors which are placed in bottom layers are almost eroded by the time the attacker reaches the layer. The attacker will not know whether a broken/isolated connector is because of chemical erosion or camouflaging[5, 11]. As an alternative, a tricky attacker may try to resolve the functionalities of obfuscated gates by indirect attacks such as ITA, CPA, SCA, BFA that elaborated in Section 2.1. We now consider the resilience of our obfuscation approach to these possible attacks.

IC Testing Based Attack (ITA). In our gate selection scheme, for each MUX that replaces an *innerGate*, its output cannot be sensitized to any PO of the circuit, because for any $MFIC_{PO_i}$ it belongs to, the *outGate* that directly feeds PO_i is obfuscated. And for each MUX that replaces an *outGate*, its inputs cannot be justified from PIs because the MUX's at least one input blocked.

Circuit partition based attack (CPA). We select gates to obfuscate from the same gate class in which gates belong to exactly the same set of $MFIC_{PO}$. As a result, for any $MFIC_{PO_i}$ of the circuit, either none of the obfuscated gates belong to it, or all the obfuscated gates belong to it. Thus the attacker will not be able to partition the obfuscated gates into multiple subcircuits to perform attacks individually.

Side channel attack (SCA). The differences of power and timing between differently configured MUX4x1 are trivial. Moreover, considering that the attacker do not have access to individual logic gate of unpackaged functional IC, such trivial difference in single logic gate is unobservable when placed in the entire circuit. As a result, the attacker will not be able to get configuration information of MUXs by side channel information.

Brute force attack (BFA). The functionality of each camouflaged MUX appears 16 possibilities to the attacker, and as analyzed above, our approach is resilient to ITA, SCA and CPA, thus the brute force efforts needed by the attacker will be no less than $16^M = 2^{4M}$, where M is the number of *innerGates* obfuscated. Such exponential brute force complexity actually means infeasible for the attacker to perform, especially when M is large.

4. EXPERIMENTAL RESULTS

4.1 Experimental Setup

We investigate the impact of our proposed obfuscation approach to design quality (area and delay) on standard ISCAS 85/89 and MCNC benchmark suites. To accomplish this, a program was written in JAVA to do gate classification and identify candidate gates to replace with MUXs. Synthesis tool ABC program[17] and Oklahoma State University (OSU) standard cell library based on the TSMC 0.35μm PDK are used for performance overhead measurement.

4.2 Performance Overheads of MUX Replacement based Obfuscation

To study the performance overhead of the proposed obfuscation scheme, we consider different number of *innerGates* in the circuit obfuscated. In Table 1, column 2-6 demonstrate the detailed information of each benchmark circuit, and columns 7-8 show the largest gate class information of the circuit. Columns 9-12 indicate an average overhead of 47.7% in delay and 14.8% in area when 5% of the logic gates in the circuit are obfuscated. Although the delay overhead varies from 9.1% to 85.9%, the area overhead is more stable due to the fact that a fixed portion (5% in this case) of the gates are obfuscated. The benefit of such high overhead is the high security against RE attackers. In the smallest 142-gate circuit s713, the attacking complexity is 16^6 or 2^{24}. For the largest circuit s13207, this complexity is 2^{344}.

To achieve a required level of security, we obfuscate 64 and 16 *innerGates*, which delivers attacking complexity of $2^{256}$and 2^{64}, respectively. As we can see from the rest columns in Table 1, both delay and area overheads drop. The most notable observation is that when we obfuscate 16 *innerGates*, there is no delay overhead on four benchmarks (c3540,

Table 1: Performance Overheads when Obfuscating Different Number of Gates.

Bench	Gates	Nets	Area	Delay	Levels	Largest Class		Obfuscate 5% Gates				Obfuscate 64 *innerGates*			Obfuscate 16 *innerGates*		
						$MFIC_{PO}$	*Gates*	total	inner	delay	area	total	delay	area	total	delay	area
s713	142	290	359.2	2.67	16	1	19	7	6	41.2%	14.5%	-	-	-	-	-	-
c432	180	363	448	3.94	23	5	41	9	8	26.1%	16.4%	-	-	-	-	-	-
i2	222	460	598.4	1.82	10	1	222	11	10	52.7%	13.4%	-	-	-	-	-	-
s1196	398	897	1051.2	2.94	17	6	22	17	16	51.0%	14.0%	-	-	-	17	50.7%	14.0%
s1238	445	999	1178.4	3.03	18	1	25	22	21	50.2%	16.2%	-	-	-	17	35.3%	12.3%
too_large	519	1125	1340	4.17	25	1	149	25	24	9.1%	14.7%	-	-	-	17	9.1%	9.8%
c2670	721	1324	1704.8	2.74	18	1	134	36	34	55.5%	15.6%	-	-	-	17	43.4%	8.0%
c3540	861	1967	2360.8	4.94	31	10	82	43	35	59.5%	15.1%	-	-	-	17	0.0%	5.5%
t481	1098	2596	3324	2.46	15	1	1098	54	53	45.5%	13.5%	-	-	-	17	45.5%	4.4%
s5378	1151	2423	2980.8	2.2	14	2	84	57	55	85.9%	18.1%	-	-	-	18	50.9%	5.5%
c5315	1345	2907	3672.8	5.67	34	5	166	67	57	28.6%	14.4%	-	-	-	19	0.0%	3.5%
s9234	1505	3105	4126.4	4.35	23	1	87	75	74	49.9%	17.3%	65	37.2%	15.3%	17	0.0%	2.9%
c7552	1612	3372	4617.6	4.8	28	1	112	80	77	37.9%	12.2%	69	14.6%	10.3%	19	4.6%	2.1%
i10	1904	4145	5135.2	5.94	36	10	121	95	76	64.8%	14.5%	83	48.5%	12.6%	24	16.2%	3.8%
c6288	2267	5220	6140.8	15.18	89	16	142	113	82	55.3%	13.3%	93	23.4%	10.4%	23	12.4%	2.5%
s13207	2480	4741	6831.2	4.23	26	19	137	124	86	50.1%	14.0%	102	39.7%	11.4%	17	0.0%	1.5%
Average										47.7%	14.8%		32.7%	12.0%		20.6%	5.8%

c5315, s9234 and s13207) because none of the obfuscated gates is on any of the critical paths. For the last five large benchmarks with more than 1500 logic gates, the average delay overheads are 32.7% and 6.6%; while the area overheads are 12.0% and 2.6%, respectively. This trend clearly shows that for large real-life circuits, our approach can achieve both low performance overhead and high security level against RE attacks.

5. CONCLUSIONS

Circuit obfuscation is a promising approach to thwart reverse engineering attacks. In this paper, we propose such a scheme based on replacing carefully selected logic gates with special designed MUXs to make RE complexity exponential. We also report the performance overhead of the obfuscated circuits which is expected to be small for real-life circuits.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant No. 61176035 and 61228204. Mingze Gao and Gang Qu are partially supported by AFOSR MURI under award number FA9550-14-1-0351.

References

[1] Gang Qu and Miodrag Potkonjak. Intellectual property protection in VLSI designs: Theory and practice, kluwer academic publishers, ISBN 1-4020-7320-8. January 2003.

[2] Masoud Rostami, Farinaz Koushanfar, and Ramesh Karri. A primer on hardware security: Models, methods, and metrics. *Proceedings of the IEEE*, 102(8):1283–1295, 2014.

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

[4] L.W. Chow, J. Baukus, and W. Clark. Integrated circuits protected against reverse engineering and method for fabricating the same using an apparent metal contact line terminating on field oxide, 2002, US Patent 20020096776.

[5] R.P. Cocchi, J.P. Baukus, B.J. Wang, L.W. Chow, and P. Ouyang. Building block for a secure cmos logic cell library, 2012, US Patent 8,111,089.

[6] Alex Baumgarten, Akhilesh Tyagi, and Joseph Zambreno. Preventing IC piracy using reconfigurable logic barriers. *IEEE Design & Test of Computers*, 27(1):66–75, 2010.

[7] Yousra Alkabani and Farinaz Koushanfar. Active hardware metering for intellectual property protection and security. In *Proceedings of the 16th USENIX Security Symposium*, 2007.

[8] Rajat Subhra Chakraborty and Swarup Bhunia. HARPOON: an obfuscation-based soc design methodology for hardware protection. *IEEE Trans. on CAD of Integrated Circuits and Systems (TCAD)*, 28(10):1493–1502, 2009.

[9] Jarrod A. Roy, Farinaz Koushanfar, and Igor L. Markov. EPIC: ending piracy of integrated circuits. In *Proc. IEEE Design, Automation and Test in Europe (DATE)*, pages 1069–1074, 2008.

[10] Jeyavijayan Rajendran, Youngok Pino, Ozgur Sinanoglu, and Ramesh Karri. Security analysis of logic obfuscation. In *Proc. ACM/IEEE Design Automation Conference (DAC)*, pages 83–89, 2012.

[11] Jeyavijayan Rajendran, Michael Sam, Ozgur Sinanoglu, and Ramesh Karri. Security analysis of integrated circuit camouflaging. In *Proc. ACM Conference on Computer and Communications Security (CCS)*, pages 709–720, 2013.

[12] Bao Liu and Brandon Wang. Embedded reconfigurable logic for ASIC design obfuscation against supply chain attacks. In *Proc. IEEE Design, Automation and Test in Europe (DATE)*, pages 1–6, 2014.

[13] W.M. Clark, L.W. Chow, G. Harbison, and P. Ouyang. Programmable connection and isolation of active regions in an integrated circuit using ambiguous features to confuse a reverse engineer, 2008, US Patent 20080079082.

[14] W.M. Clark, J.P. Baukus, and L.W. Chow. Implanted hidden interconnections in a semiconductor device for preventing reverse engineering, 2007, US Patent 7,166,515.

[15] L.W. Chow, W.M. Clark, G.J. Harbison, and J.P. Baukus. Conductive channel pseudo block process and circuit to inhibit reverse engineering, 2006, US Patent 7,049,667.

[16] L.W. Chow, W.M. Clark, and J.P. Baukus. Integrated circuit with reverse engineering protection, 2005, US Patent 6,897,535.

[17] Robert K. Brayton and Alan Mishchenko. ABC: an academic industrial-strength verification tool. In *Proc. CAV Computer Aided Verification, 22nd International Conference*, pages 24–40, 2010.