Revisit Sequential Logic Obfuscation: Attacks and Defenses

Travis Meade*, Zheng Zhao†, Shaojie Zhang*, David Pan†, and Yier Jin‡
*Department of Computer Science, University of Central Florida
†Department of Electrical and Computer Engineering, University of Texas at Austin
‡Department of Electrical and Computer Engineering, University of Central Florida
travm12@knights.ucf.edu, zhengzhao@utexas.edu, shzhang@cs.ucf.edu, dpan@ece.utexas.edu,
yier.jin@eecs.ucf.edu

***Abstract*—The urgent requests to protection integrated circuits (IC) and hardware intellectual properties (IP) have led to the development of various logic obfuscation methods. While most existing solutions focus on the combinational logic or sequential logic with full scan-chains, in this paper, we will revisit the security of sequential logic obfuscation within circuits where full scan-chains are not available or accessible. We will first introduce attack methods to compromise obfuscated sequential circuits leveraging newly developed netlist analysis tools. We will then propose systematic solutions and provide guidelines in developing resilient sequential logic obfuscation schemes.**

I. Introduction

The globalization of IC supply chain has raised IC/IP privacy concerns. Upon this request, various IC/IP protection methods have been raised among which the leading solutions to prevent reverse engineering attacks (or malicious foundry attacks) are logic obfuscation methods. Most of the existing solutions are targeting combinational circuits assuming that full scan-chains are always available and accessible.

However, orthogonal to existing solutions, this paper reviews the problem of hardware logic encryption and decryption in the realm of sequential logic. An intrinsic desire to learn the fundamental characteristics of sequential logic drives us to build a more effective attack model and design stronger sequential logic encryption accordingly. Although past defensive techniques focus on finite state machine (FSM) encryption, there exist many methods to partially or fully extract FSM logic from gate-level netlists [1]–[3]. While potentially motivated by Trojan detection or even Hardware obfuscation, methods like these can lead to fully reverse engineering netlists.

Unlike the hardware obfuscation method HARPOON [4], the proposed defense method here does not use an entrance-FSM scheme which can be vulnerable to REFSM [1] and fault injection attack. Rather, the focus falls on a given sequential design per se, which is encrypted directly. In the FSM representation of the encrypted design, application of an incorrect key redirect some transitions to incorrect states. The state space itself expands with new states. In this way, the original FSM is entangled with the wrong FSM which makes it much more difficult to be decrypted. In this way, the problem becomes NP-complete even with the fully-reversed FSMs; not to mention the golden model's FSM cannot be reversed without the gate-level netlist. Also worth noting the proposed method can be used in parallel with HARPOON-like techniques.

Our fundamental assumption is the unavailability of a complete scan chain in the design, which either is non-existent (usually for high performance ICs that has a stringent overhead constraints) or can be protected by various scan chain securing methods [5], [6]. To resolve the unknown gate functions and registers, unrolling the sequential design while leveraging the state of the art combinational SAT-based attack [7] creates a simple baseline (unroll-and-SAT attack). SAT-based attack is the latest attack technique for combinationally encrypted designs. We notice the number of unrolling has a significant impact on the time of decryption. On the protector's side, therefore, we try to increase the minimum number of unrolling required to decrypt the design. For general designs that may contain complete scan, we will have to resort to scan chain protection techniques. The more registers to protect, the higher security level becomes and the higher the overhead becomes.

The rest of this paper is organized as follows. Section II presents the state-of-the-art and their limitations. Section III introduces the vulnerabilities of existing sequential logic obfuscation methods as well as possible attacks to these methods. Section IV presents our enhanced sequential logic obfuscation solutions and the design trade-offs between performance overhead and attack complexities of these solutions. The conclusions are drawn in Section V.

II. Related Works

Methods for protecting circuits through obfuscation abound in modern research [8]–[10]. However, many of these methods are completely or partially susceptible to attacks proposed in research [7], [11]. Few methods utilize the temporal naturally occurring in sequential circuits. The more recent sequential models have limited usage of sequential logic [4], [12]. In [12] a set of extra state elements stores a key that is XOR-ed with a state on particular transitions to corrupt the function of the netlist. For an example FSM in Figure 1a. The red transitions for each FSM in Figure 1 denote incorrect transitions that can prevent the netlist from unlocking or are only present in locked FSMs. The key can be extracted by the previously mentioned unroll-and-SAT attack. Alternatively, the protection can be susceptible to FSM extraction, since only one transition is incorrect.

A second popular sequential protection scheme, HARPOON [4], focuses on limiting access to the original FSM by requiring an unlocking sequence of inputs. To do so HARPOON expands the FSM's state space. HARPOON then uses the inserted states to corrupts parts of the circuit. For an example FSM see Figure 1b. However, once the FSM's state is within a correct state, normal circuit execution will not cause circuit corruption. Fault-injection attacks [13]–[15] can

be leveraged to prematurely transition to a correct state. Worse off fault injection is not needed; due to the method's limited degree of inserted logic, FSM recovery tools can extract the key sequence [16].

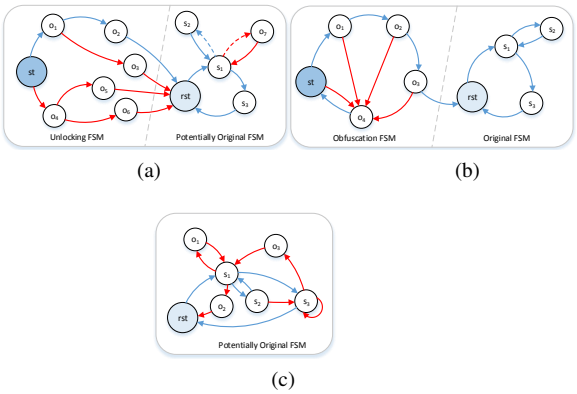

Fig. 1: (a) Interlocking FSM encryption (b) Example HARPOON FSM (c) our proposed sequentially encryption FSM.

With the consideration of the state of the art defense, we present a possible attack method that can recover chips protected with modern sequential obfuscation. We then propose sequential encryption technique to overcome such attacks, by creating a more complex structure, like HARPOON, while potentially augmenting the functional FSM's topology like that in [12]. See Figure 1c for a comparison to other accepted methods. Lastly the method can be evaluated by examining the behavior when attacked by unroll-and-SAT.

III. Attack

A. Introduction to Sequential Logic Obfuscation

Sequential encryption schemes often focus their efforts on the implicit logical FSM. The straightforward approach increases the FSM's state space thereby reducing access of the original FSM. In addition access to the circuit's true logic can require a particular sequence of input vectors. Other methods incorporate special locking states in the updated FSM. These methods select a subset of states that can be accessed by the new reset state but cannot reach the states of the original FSM.

Two main methods for increasing the state space exist. The first method changes the logic of the registers to utilize previously unreachable states. The other method involves inserting additional registers that usually but not necessarily act as flags for the FSM's behavior. Additional registers tend to be very appealing since the state space increase exponentially with the number of inserted registers. The major detriments to a large number of register insertions is the time to unlock, area, and power overhead.

An example of sequential circuit encryption, HARPOON, inserts additional state elements (SE) and combinational logic that adversely affects the behavior of the netlist while the circuit is locked. The inserted SEs control the activation of the inserted combinational modules, that have the potential to corrupt parts of the netlist. Moreover HARPOON's FSM's state space is partitioned into three general sections (modes): obfuscation, authentication, and original. The obfuscation mode, the first part of the obfuscation mode, corrupts parts of the netlist. The authentication mode simply watermarks the netlist. The original mode, as it sounds, does not corrupt the netlist's internal signals and allows for normal execution. The authors assume that an attacker would randomly reverse engineer the netlist which gives the defender a large probability of protection, but a smarter solution exists based on their protection method.

B. Attacks on Sequential Logic Obfuscation

Attacking the HARPOON protection requires first identifying the registers associated with the netlist's mode control. In general finding inserted registers partially reveals the function of the chip's logic. Several techniques can be used to extract these registers.

The first method that can be used was a register classification tool, RELIC [17]. RELIC itself is a tool used to separate parts of the netlist based on implicit features that are induced when including either extra logic or circuitry. RELIC finds repetitive wire patterns by examining the correlation of a wire's structural variables (e.g. fan in size, distance to input/output wires, etc.). The outlying wires tend to fall into the category of logic due to the nature of how netlists are synthesized (i.e. a fixed protocol replicates structure within data words). RELIC might not be capable of finding all the inserted registers in one try. To compensate RELIC is used to find partial register sets, and the sets expand via register dependency.

The second method stems from the register set expansion technique mentioned in the first method. With the process RELIC itself is removed from the equation, and the register dependency becomes the sole method for "classification". This is done by way of Tarjan's Strongly Connected Components (SCC) algorithm [**?**]. The algorithm finds what is commonly referred to as the transitive closure of directed graphs. The algorithm and properties are well detailed in other resources. The graph returned contains a set of vertex sets that represent SCCs of the original graph, which is potentially connected by a set of directed edges that denote how the original graph's components interact. The graph itself is directed and acyclic (see Figure 2).

The Strongly Connected Component graph can also be used to attack more recent protection schemes such as DSD (Dynamic State-Deflection. DSD [18] relies on inserted, persistent logic that is unaffected by the original logic (or original data for that matter). Thus Tarjan's algorithm can detect these inserted state flip-flops. When observing the FSM and the transition probability generated by these inserted FFs the correct state becomes obvious. In general the components that are analyzed are those that contain no incoming edges (i.e. source SCCs). Source SCCs will exist because the graph is acyclic (and presumably non-empty).

Once found, the inserted registers are used by the REFSM tool to construct a partial FSM of the netlist. For protection schemes such as HARPOON the desired FSM section (i.e. original mode) is the authentication sequence's "end". The end is found using Tarjan's SCC algorithm. The FSM is broken down into its components, and the component(s) without outgoing edges (i.e. sink SCCs) are analyzed. If multiple

sinks exist, the one selected is typically the component that has the lowest reachability probability, as the others are probably black-hole states (i.e. states that exist to trap incorrect sequences). These black-hole states are typically included in other protection methods. REFSM then generates the shortest input sequence to enter a state within the supposed normal mode FSM.

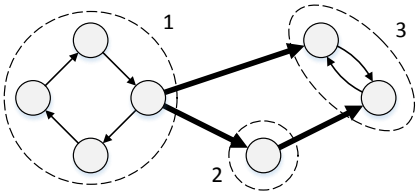

Fig. 2: A graph which is partitioned into its three SCCs. The first being the only source SCC, and the third being the only sink SCC.

The best chance a user has at improving HARPOON without overhauling the method is to increase the complexity of the FSM. With a large enough FSM it becomes infeasible to extract the unlocking sequence. The major concern with this approach is the incurred overhead. Aside from power and area increase due to the increased number of SEs, the major drawback is the time the circuit takes to unlock from power-on.

Alternatively users can incorporate other defense techniques. Although this would also not necessarily ensure protection, it would definitely make reverse engineering even more difficult for adversaries. A typical defense that has been prevalent in current research is the use of gate camouflaging. Even though methods exist that can break standard gate camouflaging, the mixture of methods can slow down or even halt IP piracy.

IV. Defense

To conquer the limitations of HARPOON, we present a new defense that encrypts the state space such that the original state space and the wrong state space are entangled. In this way, SCC-based approaches cannot be easily applied to distinguish the right state or transition from wrong.

In order to defeat SCC, we select the states and its outgoing transitions of the original design to encrypt, so that there is no FSM sink containing exclusively the original state space. By adding the key conditions on the transitions, we can assure a correct transition if a correct key is applied and a wrong transition otherwise. There are several methods that encrypt a given state S^* with assignments of wrong keys.

The first choice is to redirect the state S^* to another state existing in the original state space, where the PO's of this chosen state and S^* are the same under the transitional condition of PI's. The second way also redirects S^* to an existing state but the outputs are different. The third way is to create new states with a certain choice of PO values.

If a wrong transition results in the same PO, then an attacker would have to unroll at least one more round. While possible, creation of fake transitions that mimic the original behavior continues, which increases the required number of unrollings. To break many SAT attack methods insertion of camouflaged AND-trees for encrypted states can be employed [11], [19]. The methods guarantee an exponential number of SAT iterations with respect to the AND-tree size. It should be noted that the major drawbacks of AND-tree insertion is low output corruptibility.

Selection schemes should not only consider the decryption complexity but also the overhead of encryption. Reusing states in the original state space is not necessarily better than creating new states. When adding gates so that a state transition function changes, previously unreachable states have a higher probability of being reached. On the other hand, reusing the original states might require many inserted gates. State reuse is yet another option in consideration when minimizing the overhead.

Still, states and transitions encryption selection challenges researchers. High-level code extraction techniques such as REFSM can recover FSMs from the proposed scheme as it is. By examining the transitions that lead to an encrypted state (presuming the adversary knows it), REFSM can look at the transition (and its conditions) from this encrypted state to figure out the correct key. However, REFSM scales poorly. If the number of states is large, the transitions are complex, or the fake transitions are cleverly chosen that prohibit REFSM from identifying the non-encrypted from the encrypted states easily, as this would exploit REFSM's scalability limitation. As the method proposed in [1] is essentially breadth-first search, we naturally give a higher encryption priority for the deeper states that will be reached later using REFSM. The state reachability probability falls into a second criterion position. The reason being to encrypt the states at the same structural depth entails the same number of unrolling, whence decryption complexity due to unrolling; while operating in these states, the states more likely to be reached during operation can expect a higher degree of output corruption.

In addition to the attack vector of REFSM which can exploit Tarjan's SCC algorithm, the SAT-based attack is also considered. SAT-based attack is another powerful up to date attack which is designed to take advantage of the input-output patterns. The attacker decrypts the key by applying primary inputs PI's and observing primary outputs (PO's) of the unencrypted oracle/golden circuit. However, the protected scan chain removes the attacker's control state register's outputs and limits their observation of register inputs.

In order to assure all the inputs PI's and the register inputs fed into the combinational part are equivalent for the golden and encrypted circuits, the attacker can resort to unrolling. For each unrolling round, the attacker utilizes the same key signals for the new round. Further unrolling will likely decrease the key space further[1]. However, the outcome of unrolling is a circuit that is several times larger than the original. Even if the number of key bits does not increase, Figure 3 shows decryption time increases drastically as the unrolling number increases. It turns out that the required number of unrolling times is equal to the minimal depth of an encrypted state to be reached from the reset state. Hence the strategy to

[1]Some circuit might not decrease the key space with an unrolling.

choose deeper states and transitions to encrypt also works against SAT-based attack. Previous work [20] shows that the depth of small benchmarks (e.g., *s208.1* and *s526*) can exceed hundreds.

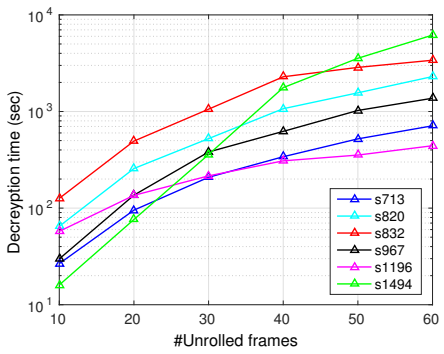

Fig. 3: Logarithmic decryption runtime with respect to the number of unroll times.

The above discussion is based on the condition that complete scan chain is unavailable in the given design. If, however, the scan chain is complete in the design, there is a need to protect some of the state registers to take advantage of sequential encryption. As the protection overhead increases with more registers being covered in scan chain [21], [22], there is a trade-off relation of the number of registers to protect, the overhead of protection, and the security level to achieve.

Figure 4 shows the example of how different choices of scan-chain covered registers can result in different unrolling depth that is required for the final FSM. A good choice of state registers to protect should carefully evaluate the differences.

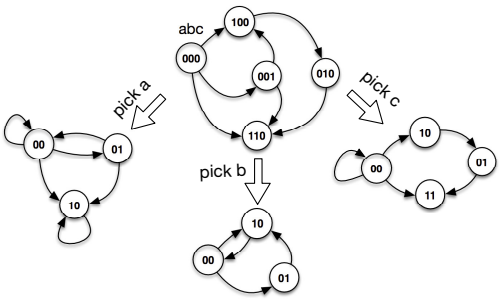

Fig. 4: Off-scan chain registers selection.

V. Conclusions

In this paper, we have revisited existing sequential logic encryption solutions. More specifically, we have introduced the security vulnerabilities of existing solutions and presented several attack methods to break the obfuscation schemes. Resilient sequential logic obfuscation methods were then discussed outlining the preliminary requests to secure sequential circuits without full scan-chain accesses. As our future research tasks, we will materialize the developed solutions to develop efficient sequential logic obfuscation solutions.

References

[1] T. Meade, S. Zhang, and Y. Jin, "Netlist reverse engineering for high-level functionality reconstruction," in *21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2016, pp. 655–660.

[2] Y. Shi, C. W. Ting, B.-H. Gwee, and Y. Ren, "A highly efficient method for extracting fsms from flattened gate-level netlist," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*. IEEE, 2010, pp. 2610–2613.

[3] A. Nahiyan, K. Xiao, K. Yang, Y. Jin, D. Forte, and M. Tehranipoor, "Avfsm: a framework for identifying and mitigating vulnerabilities in fsms," in *Design Automation Conference (DAC), 2016 53nd ACM/EDAC/IEEE*. IEEE, 2016, pp. 1–6.

[4] R. Chakraborty and S. Bhunia, "HARPOON: An obfuscation-based soc design methodology for hardware protection," *IEEE J. Technol. Comput. Aided Design*, vol. 28, no. 10, pp. 1493–1502, 2009.

[5] G. Sengar, D. Mukhopadhyay, and D. Chowdhury, "Secured flipped scan-chain model for crypto-architecture," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 11, pp. 2080–2084, 2007.

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

[7] M. El Massad, S. Garg, and M. V. Tripunitara, "Integrated circuit (ic) decamouflaging: Reverse engineering camouflaged ics within minutes." in *NDSS*, 2015.

[8] M. Yasin, B. Mazumdar, J. J. Rajendran, and O. Sinanoglu, "Sarlock: Sat attack resistant logic locking," in *Hardware Oriented Security and Trust (HOST), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 236–241.

[9] J. A. Roy, F. Koushanfar, and I. L. Markov, "Epic: Ending piracy of integrated circuits," in *Proceedings of the conference on Design, automation and test in Europe*. ACM, 2008, pp. 1069–1074.

[10] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Security analysis of logic obfuscation," in *Proceedings of the 49th Annual Design Automation Conference*. ACM, 2012, pp. 83–89.

[11] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *Hardware Oriented Security and Trust (HOST), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 137–143.

[12] A. R. Desai, M. S. Hsiao, C. Wang, L. Nazhandali, and S. Hall, "Interlocking obfuscation for anti-tamper hardware," in *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*. ACM, 2013, p. 8.

[13] A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache, "Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures," *Proceedings of the IEEE*, vol. 100, no. 11, pp. 3056–3076, 2012.

[14] C. H. Kim and J.-J. Quisquater, "Faults, injection methods, and fault attacks," *IEEE Design & Test of Computers*, vol. 24, no. 6, pp. 544–545, 2007.

[15] K. Rothbart, U. Neffe, C. Steger, R. Weiss, E. Rieger, and A. Mühlberger, "High level fault injection for attack simulation in smart cards," in *Test Symposium, 2004. 13th Asian*. IEEE, 2004, pp. 118–121.

[16] T. Meade, S. Zhang, and Y. Jin, "Netlist reverse engineering for high-level functionality reconstruction," in *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2016, pp. 655–660.

[17] T. Meade, Y. Jin, M. Tehranipoor, and S. Zhang, "Gate-level netlist reverse engineering for trojan detection and hardware security," in *The IEEE International Symposium on Circuits and Systems (ISCAS)*, 2016, pp. 1334–1337.

[18] J. Dofe, Y. Zhang, and Q. Yu, "Dsd: a dynamic state-deflection method for gate-level netlist obfuscation," in *VLSI (ISVLSI), 2016 IEEE Computer Society Annual Symposium on*. IEEE, 2016, pp. 565–570.

[19] M. Li, K. Shamsi, T. Meade, Z. Zhao, B. Yu, Y. Jin, and D. Z. Pan, "Provably secure camouflaging strategy for ic protection," 2016.

[20] M. Mneimneh and K. Sakallah, "Sat-based sequential depth computation," in *Proceedings of the 2003 Asia and South Pacific Design Automation Conference*. ACM, 2003, pp. 87–92.

[21] J. Lee, M. Tehranipoor, C. Patel, and J. Plusquellic, "Securing designs against scan-based side-channel attacks," *IEEE transactions on dependable and secure computing*, vol. 4, no. 4, pp. 325–336, 2007.

[22] D. Hély, F. Bancel, M.-L. Flottes, and B. Rouzeyre, "Securing scan control in crypto chips," *Journal of Electronic Testing*, vol. 23, no. 5, pp. 457–464, 2007.