Enhanced Circuit Security Through Hidden State Transitions

Kyle Juretus
Department of Electrical and Computer Engineering
Drexel University
Philadelphia, Pennslyvania 19104
kjj39@drexel.edu

Ioannis Savidis
Department of Electrical and Computer Engineering
Drexel University
Philadelphia, Pennslyvania 19104
isavidis@coe.drexel.edu

***Abstract*—A technique is developed that uses the state space of an integrated circuit (IC) to increase security. Timing path dependencies and coupling capacitance are utilized to create hidden state transitions that are not observable after netlist extraction. Temporal based transitions are also introduced to shift key dependencies into the time domain. The state space dependencies of the IC serve as a means to counter the SAT attack. Implementing temporal based transitions increases the area of the circuit by 68.42%, the power by 43.17%, and does not impact circuit delay. However, an increased circuit size significantly reduces the overhead of implementing state space encryption. For example, encrypting two registers in the s15850 ISCAS89 benchmark circuit resulted in an area overhead of 0.026%, providing a means to secure sequential logic with minimal overhead.**

***Keywords*—Logic encryption; logic locking; hardware security; sequential obfuscation**

I. Introduction

Commercial foundries are becoming an important manufacturing strategy for fabricating ICs [1], partially due to the required multi-billion dollar investment to construct advanced fabrication facilities [2]. In addition, IC design firms are incorporating third-party intellectual property (IP) to reduce design time and development costs [1]. The rise of commercial foundries and third party IP have shifted a once vertical IC flow to a horizontal design flow with a variety of untrusted entities including third-party IP vendors, fabrication facilities, test facilities, and end-users. Government and commercial ICs, therefore, are vulnerable to IP theft, counterfeiting, overproduction, and the insertion of harmful circuit modifications (hardware Trojans). The challenge for creating secure circuits then becomes protecting against the wide variety of possible threats, including denial of service, theft of information, and/or corrupting functionality [3].

An area of research that addresses the security risks of IP theft, IC counterfeiting, IC overproduction, and hardware Trojan insertion is logic encryption/locking [4], which adds additional circuitry (key gates) to an IC to hide the functionality of the circuit from an adversary. The difficulty to steal IP, counterfeit the IC, produce extra ICs, and even insert hardware Trojans is increased as a malicious foundry, or end user, no longer possesses the entire IC design after reverse engineering. However, the SAT based attack presented in [5] circumvents the security provided by logic encryption and, therefore, requires the development of novel measures to defend against the wide variety of threats facing ICs. The work described in this paper presents hidden state transitions to obfuscate a logical netlist with temporal key dependencies, which also increases the difficulty of executing the SAT attack.

The paper is organized as follows. The threat model for this work is discussed in Section II. An overview of related research on sequential logic encryption is provided in Section III. Hidden state transitions based on frequency and coupling capacitance are described in Section IV as a means to increase security against SAT based attacks. The hidden state transitions are modified to include temporal dependencies within a finite state machine (FSM), allowing for increased security without a partitioned FSM, as described in Section V. Concluding remarks are provided in Section VI.

II. Threat Model

The threat model is the same as that utilized by the SAT attack [5], where the adversary has access to an activated IC capable of producing correct input-output (IO) pairs. In addition, a secure scan-chain is assumed, where the scan-chain output is not available to the adversary, as exposing the IC state to an adversary through the activated IC drastically decreases the security available from logic obfuscation.

A. Vulnerability to the SAT Attack

The gains in security of implementing logic encryption are attributed to the large number of possible IO combinations that prevent an adversary from performing a brute force attack. Fundamentally, the number of IO pairs and key combinations increases exponentially with each additional bit, which limits the feasibility of a brute force attack. However, the SAT attack demonstrated that with limited correct IO pairs from an activated IC, the logic encryption key is extracted for 95% of the ISCAS85 benchmarks within 10 hours [5]. A brute force attack is no longer required as the addition of a single IO constraint to the SAT solver prunes a significant amount of the key space, weakening the security of logic encryption.

B. Preventing the SAT Attack

The SAT attack is premised on the concept that an incorrect output by a given key results in the removal of the said key

and a refinement of the constraints imposed on the solver. Methodologies are developed in this paper to create temporal dependencies in the key, which prevents the discarding of key values once an incorrect output is observed when applying the SAT attack. Instead, the constraints for the SAT attack now require awareness of the current state, which increases the difficultly of eliminating keys. Hidden state transitions are also developed, creating secret FSM transitions that are not logically present in the reverse engineered netlist of the encrypted IC.

III. Related Work

Partitioning the FSM into an obfuscated and functional mode was used to secure logic in HARPOON [6]. For example, a FSM partition that only enters correct operation after the key sequence of P0, P1, and P2 is applied is shown in Fig. 1. The FSM may include *black hole* states, which are never functionally exited, or *gray hole* states, which are very hard to functionally exit, to further deter adversaries from attempting to reverse engineer the IC [7].

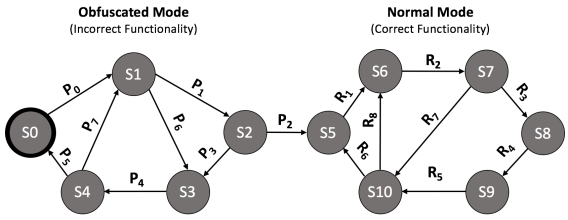

Fig. 1: Modified FSM with an enabling sequence of P0, P1, P2 required to enter normal circuit operation [6].

Partitioning the FSM into obfuscated and functional modes is also described in [8], where a code-word is generated during state transitions when operating in obfuscated mode. An incorrect code word results in incorrect operation of the FSM when in functional mode.

The partitioning of state machines has led to security concerns regarding FSM extraction attacks [9] and fault-injection attacks [10]–[12], as described in [13]. The work in [13] adds incorrect state transitions during normal operation based on the applied key and attempts to increase the number of times the state machine must be unrolled, limiting the SAT attack. The work in [14] aims to eliminate partitioning of the FSM into obfuscated and functional modes by utilizing dynamic state deflection based on an applied key vector.

In this work, hidden state transitions based on frequency and coupling capacitance are implemented as a means to prevent an FSM extraction based attack from determining FSM transitions. In addition, temporally dependent keys are added to the IC to increase the difficulty of reverse engineering an encrypted IC using the SAT attack.

IV. Hidden State Transitions

A means to add hidden transitions to a state machine based on 1) the frequency of an IC, and 2) coupling capacitance are described in this section. Both techniques modify the FSM to create hidden state transitions that are not present within the logic of the FSM. The original state machine shown in Fig. 2 is modified using both techniques to demonstrate feasibility.

A. Frequency Based Hidden State Transitions

The technique utilizes frequency changes to implement hidden state transitions. The FSM shown in Fig. 3 is developed with a frequency based hidden state transition from *S3* to *S4*. The utilization of the frequency domain removes a direct logical representation of the circuit after reverse engineering. Only the IP developer is aware of the transition from *S3* to *S4* at a frequency of 5 GHz created by a strategic timing glitch in the circuit. Therefore, FSM recovery and other techniques that exploit state reachability are not able to discern the state transition. The states *S7* and *S8* are also partitioned from the original circuit, creating the appearance of four state machines when FSM recovery tools are applied.

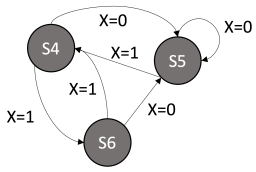

Fig. 2: Original state machine with single input *X*.

Frequency based state traps such as *S7*, which serves as a black-hole state that is not exited, are also possible to prevent an adversary from blindly increasing the frequency. The design transitions to *S7* if the 5 GHz frequency is maintained for three clock cycles. Entering another state machine partition or restarting the sequence to enter functional mode are additional available options when adding hidden states.

SPICE simulation was used to verify the operation of the implemented FSM with frequency dependent hidden states. The results are shown in Fig. 4b and indicate that while in *S3*, when the frequency is increased to 5 GHz, the circuit transitions to *S4* (*S1* to *S3* to *S4*). However, if the frequency is unaltered, as shown in Fig. 4a, the circuit stays in *S3* and transitions to *S2* on the next clock cycle.

The implementation of the hidden states into the FSM as shown in Fig. 3 resulted in an overhead of 97% in area, 44.5% in power, and 32.1% in delay. As the required modifications

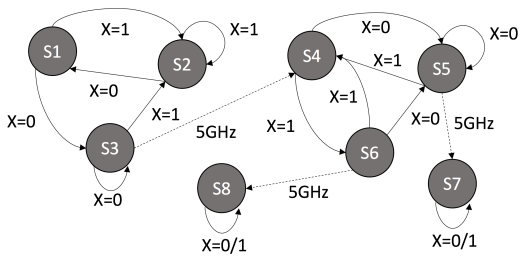

Fig. 3: State machine with hidden transitions from *S3* to *S4*, *S6* to *S8*, and *S5* to *S7* that activate at 5 GHz. A single input *X* is applied to the circuit.

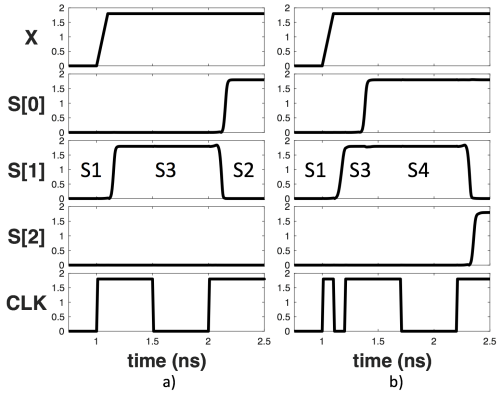

Fig. 4: Simulation of the state machine with a) no frequency based state transition, and b) a temporary increase in clock frequency to enable a hidden state transition to *S4*.

remain relatively independent of circuit size, since an additional flip-flop expands the number of states exponentially, increasing the circuit size decreases the overhead required to add the frequency dependent hidden states. In addition, using *don't care* states for hidden state transitions requires only a small increase in combinational logic.

B. Hidden Transitions Using Coupling Capacitance

In addition to the frequency based transitions presented in Section IV-A, coupling capacitance is used as a means to create hidden state transitions in an FSM. A circuit implementing a coupling based partition is developed to generate a hidden state transition from *S3* to *S4* for the state machine shown in Fig. 3. The topology shown in Fig. 5 utilizes a nearby key gate as the aggressor to ensure the controllability of the switching on the victim line. Minimum spacing between the aggressor and victim line along a coupling length of 100 μm is assumed in a 180 nm technology.

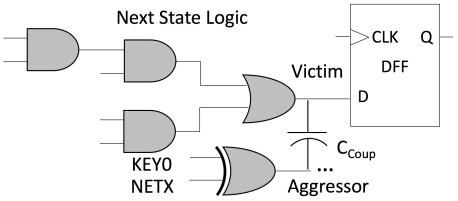

Fig. 5: Coupling capacitive based transition.

To verify the operation of the FSM, a SPICE analysis of the hidden state transitions applying the coupling capacitance based technique was completed. The analysis of the aggressor arrival period that produces a strategic glitch is shown in Fig. 6, indicating that the aggressor switching between -490 ps and 215 ps of the victim switching is sufficient to change states. As a result, for a 1 GHz clock signal, 70% of the clock period is available to switch states using the aggressor. An adversary must, therefore, determine 1) the correct state of the FSM to activate the aggressor, 2) the net acting as the hidden transition, and 3) the signals to control to activate the transition.

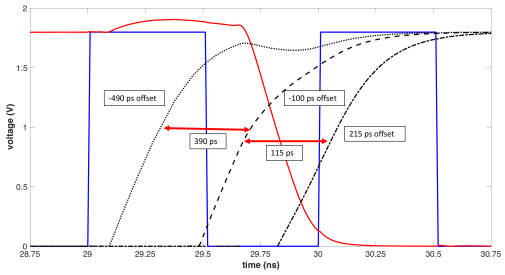

Fig. 6: Analysis of aggressor switching time vs state transition.

Note that utilizing a fault injection attack still results in entering the correct functional state machine from the obfuscated partition (*S1, S2, S3*) shown in Fig. 3, bypassing the added security features. However, a methodology that prevents fault injection attacks from bypassing hidden state transitions is described in Section V.

V. Temporal Based Transitions

While the hidden state transitions provide a means to mask the logical netlist, relying on frequency or coupling capacitance during functional operation is a challenge. Instead, it is possible to add logic to the register clock input that provides the ability to create temporally dependent keys. The simplest circuit addition is to gate the clock and use a key input as a control line. If further control through the addition of pulses to the clock signal is required, the topologies shown in Fig. 7 allow for such flexibility. The first topology, shown in Fig. 7a, gates the clock and applies an XOR to the clock gating control line. As a result, the current state of a given register is held at any given state and time reference of the FSM when *KEY0 = 0*, allowing for hidden transitions to occur. The topology depicted in Fig. 7b allows for more flexibility through the use of an AND/OR gate to implement clock gating [15], [16], which permits the clock to be held low. When holding the clock low and setting *KEY1 = 1*, additional clock pulses are applied to the flip-flops.

To create the temporally dependent circuit, the original FSM is modified by replacing the original transitions with incorrect transitions. The circuit is altered by including the structures shown in Fig. 7a at the clock input of each register of the circuit. For the FSM shown in Fig. 2, the transition *S5* to *S4* is replaced with *S5* to *S6* and the transition *S5* to *S5* is

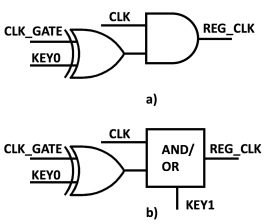

Fig. 7: Circuit topologies to generate hidden transitions: a) Clock signal gated with an AND gate and an XOR control of the clock gating, and b) clock gated with an AND/OR [15], [16] and with an XOR control of the clock gating.

replaced with the transition *S5* to *S4*. The reverse engineered state machine now appears as shown in Fig. 8. When in *S5*, the correct functionality is achieved by setting *K0 = 0* when *X = 1* and *K1 = 0* when *X = 0*, as shown by the thick red transitions in Fig. 8.

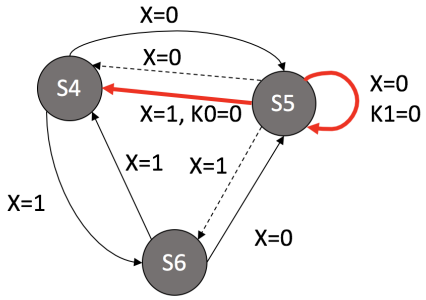

Fig. 8: Altered FSM that requires a temporally dependent key. Dotted lines represent incorrect edges that are corrected by asserting *K0* or *K1* in *S5* (correct transitions are shown as thick red lines).

The adversary no longer observes the original functionality of the FSM after reverse engineering the circuit, preventing FSM recovery tools from determining the original transitions. In addition, fault injection attacks are no longer useful as the state machine is not partitioned into obfuscated and functional regions. Additionally, the temporally dependent key requires a different key sequence for each unrolled instance of the state machine, which limits the efficacy of the SAT attack as a key sequence is no longer pruned if an incorrect result is seen at the output of the circuit for an unrolled design that does not cover the entire FSM. Essentially, the adversary is now forced to validate the temporally dependent key for each state within the FSM.

The implementation of the temporally dependent state machine does not incur any performance overhead as the combinational paths do not require any additional elements. The area and power overhead of, respectively, 68.42% and 43.17% for adding the XOR based clock gating shown in Fig. 7a is large for the implemented FSM. As the logical depth and amount of combinational logic increases in the circuit, the overheads drop significantly. For example, the area overhead of encrypting two registers is 5.08% for the ISCAS89 s298 (75 gates) benchmark, whereas the s15850 (3448 gates) benchmark circuit only incurs a 0.026% penalty in area.

VI. Conclusions

Hidden FSM transitions based on 1) the frequency of an IC and 2) coupling capacitance were introduced in this paper in conjunction with temporal key dependencies as means to increase IC security against intellectual property theft, IC counterfeiting, and IC overproduction. There is an overhead of 97% in area, 44.5% in power, and 32.1% in delay to implement frequency based transitions. The temporally dependent state machine resulted in no performance overhead, but increased area by 68.42% and power by 43.17%. A significant reduction in both area and power overheads is seen as the circuit size increases, with a decrease in area overhead to 0.026% for the 3448 gate ISCAS s15850 benchmark when encrypting two registers utilizing the temporally dependent state space encryption methodology. The use of the developed techniques drastically increases the difficulty of implementing probing based style attacks, such as the SAT attack, and provides a low cost means of securing an IC.

Acknowledgments

This research is supported in part by the Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a, Drexel Ventures Innovation Fund, and the National Science Foundation under Grant CNS-1648878.

References

[1] IARPA, "Trusted Integrated Chips (TIC) Program," IARPA-BAA-11-09, October 2011.

[2] DigiTimes, "Trends in the Global IC Design Service Market," http://www.digitimes.com/news/a20120313RS400.html?chid=2, March 2012.

[3] M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection," *IEEE Design and Test of Computers*, Vol. 27, No. 1, pp. 10–25, February 2010.

[4] J. Roy, F. Koushanfar, and I. Markov, "EPIC: Ending Piracy of Integrated Circuits," *Proceedings of the IEEE/ACM Design, Automation and Test in Europe Conference*, pp. 1069–1074, October 2008.

[5] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the Security of Logic Encryption Algorithms," *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust*, pp. 137–143, May 2015.

[6] S. Chakraborty and S. Bhunia, "HARPOON: An Obfuscation-based SoC Design Methodology for Hardware Protection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 28, No. 10, pp. 1493–1502, October 2009.

[7] Y. Alkabani and K. Farinaz, "Active Hardware Metering for Intellectual Property Protection and Security," *Proceedings of the USENIX Security Symposium*, pp. 291–306, August 2007.

[8] A. R. Desai, M. S. Hsiao, C. Wang, L. Nazhandali, and S. Hall, "Interlocking Obfuscation for Anti-tamper Hardware," *Proceedings of the Cyber Security and Information Intelligence Research Workshop*, pp. 8:1–8:4, January 2013.

[9] T. Meade, S. Zhang, and Y. Jin, "Netlist Reverse Engineering for High-level Functionality Reconstruction," *Proceedings of the Asia and South Pacific Design Automation Conference*, pp. 655–660, January 2016.

[10] 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, November 2012.

[11] C. H. Kim and J. J. Quisquater, "Faults, Injection Methods, and Fault Attacks," *IEEE Design and Test of Computers*, Vol. 24, No. 6, pp. 544–545, November 2007.

[12] K. Rothbart, U. Neffe, C. Steger, R. Weiss, E. Rieger, and A. Muehlberger, "High Level Fault Injection for Attack Simulation in Smart Cards," *Proceedings of the Asian Test Symposium*, pp. 118–121, November 2004.

[13] T. Meade, Z. Zhao, S. Zhang, D. Pan, and Y. Jin, "Revisit Sequential Logic Obfuscation: Attacks and Defenses," *Proceedings of the IEEE International Conference on Circuits and Systems*, pp. 1367–1370, May 2017.

[14] J. Dofe and Q. Yu, "Novel Dynamic State-Deflection Method for Gate-Level Design Obfuscation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. PP, No. 99, pp. 1–13, April 2017.

[15] K. Juretus and I. Savidis, "Reduced Overhead Gate Level Logic Encryption," *Proceedings of the IEEE/ACM Great Lakes Symposium on VLSI*, pp. 15–20, May 2016.

[16] K. Juretus and I. Savidis, "Reducing Logic Encryption Overhead Through Gate Level Key Insertion," *Proceedings of the IEEE International Conference on Circuits and Systems*, pp. 1714–1717, May 2016.