Increasing the SAT Attack Resiliency of In-Cone Logic Locking

Kyle Juretus
Drexel University
Philadelphia, Pennslyvania 19104
Email: kjj39@drexel.edu

Ioannis Savidis
Drexel University
Philadelphia, Pennslyvania 19104
Email: isavidis@coe.drexel.edu

***Abstract*—A method to increase the resiliency of in-cone logic locking against the SAT attack is described in this paper. Current logic locking techniques provide protection through the addition of circuitry outside of the original logic cone. While the additional circuitry provides provable security against the SAT attack, other attacks, such as the removal attack, limit the efficacy of such techniques. Traditional in-cone logic locking is not prone to removal attacks, but is less secure against the SAT attack. The focus of this paper is, therefore, the analysis of in-cone logic locking to increase the security against the SAT attack, which provides a comparison between in-cone techniques and newly developed methodologies. A novel algorithm is developed that utilizes maximum fanout free cones (MFFC). The application of the algorithm limits the fanout of incorrect key information. The MFFC based algorithm resulted in an average increase of 61.8% in the minimum number of iterations required to complete the SAT attack across 1,000 different variable orderings of the circuit netlist while restricted to a 5% overhead in area.**

***Index Terms*—logic locking, SAT attack, key gate selection, hardware security**

I. Introduction

Integrated circuits (ICs) form the root of trust in the computing stack as the software layer relies on the hardware layer to provide a consistent protocol. The trust of the hardware layer is often inherently assumed, creating increased security risks as the manufacturing of ICs transitions towards a horizontal model, where manufacturing, testing, and intellectual property (IP) are procured from third-parties. IP theft, counterfeiting and overproduction of ICs, and the insertion of harmful circuit modifications (hardware Trojans) are all possible threats of untrusted third-parties in the supply chain. Threat models associated with untrusted third-parties have already been observed, as a backdoor [1] and a variety of counterfeit ICs [2], [3] have been discovered within military ICs.

Obfuscation, which aims to limit the amount of topological and logical information recoverable from an IC, has been a major research focus to mitigate the threats of untrusted third-parties. Split manufacturing [4], IC camouflaging [5]–[7], and logic encryption/locking [8] are three of the prominent obfuscation methodologies currently in research. Both split manufacturing and IC camouflaging rely on a trusted foundry, whereas the utilization of an active key in logic locking provides increased protection against a larger threat space.

The security provided by logic locking to defend against IP theft, IC counterfeiting, IC overproduction, and hardware Trojan insertion has been undermined by the satisfiability (SAT) attack [9]. To protect against the SAT attack, a variety of techniques were developed that insert additional logic to artificially control the output corruption provided by logic locking gates, such as the topologies and structures described in [10]–[14]. While the novel structures allow for a provable methodology to increase the number of iterations of the SAT attack, vulnerabilities to removal attacks and modified SAT attacks such as AppSAT [15] and Double DIP [16] threaten the level of security provided by the developed techniques. The work in [12] characterizes the trade-off in the level of provided security between SAT attack resiliency and removal attack resiliency for implemented defensive measures, demonstrating a mutually exclusive relationship amongst the two.

The focus of this paper, therefore, is to enhance the resiliency of traditional in-cone logic locking against the SAT attack, as in-cone locking does not include additional circuitry to control the output corruption of the IC. A novel algorithm utilizing maximum fanout free cones is developed to limit the fanout of incorrect key information, which reduces both the topological and logical information leaked to an adversary executing the SAT attack for each input-output pattern generated.

The paper is organized as follows. An introduction to the SAT attack is provided in Section II. An analysis of the effect that logical reconvergence within a netlist has on the SAT attack is provided in Section III. A novel logic locking gate selection algorithm based on maximum fanout free cones is described in Section IV. Results from execution of the developed selection algorithm are provided in Section V. Some concluding remarks are provided in Section VI.

II. SAT Attack Overview

The SAT attack introduced in [9] decrypted a majority of the ISCAS'85 benchmark circuits within 10 hours, based on the assumption that an attacker has access to an activated IC operating in test mode. The activated IC allows for the determination of correct input-output pairs. The efficiency of the attack is then determined by the number of correct input-output pairs required by the SAT attack to correctly retrieve the key from the circuit.

A reverse engineered netlist is used by the adversary, where all logical information regarding the circuit is known except for the key values applied to implement the logic locking. A

978-1-7281-0397-6/19/$31.00 ©2019 IEEE

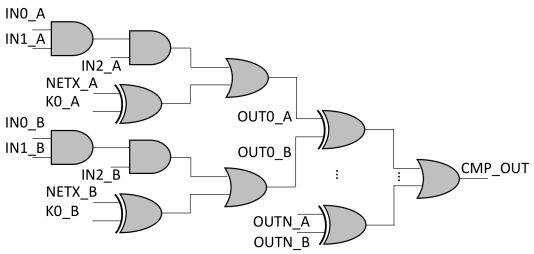

Fig. 1: Miter circuit with replicated *A* and *B* versions of the reverse engineered representation of the secured circuit. Corresponding outputs between the *A* and *B* versions of the circuit are XORed. Each XORed signal is then passed to an OR gate to check for any differentiating output.

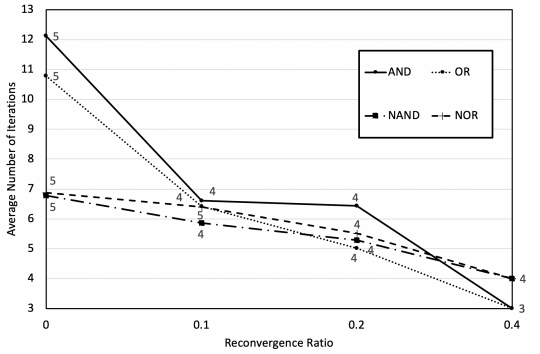

Fig. 2: Analysis of the effect of reconvergence on the number of SAT iterations required to decrypt the key for a homogeneous tree structure consisting of gates of the same type. The largest minimum number of iterations across the 1,000 benchmarks for each gate type and reconvergence ratio is listed above each data point.

miter circuit is generated by duplicating the reverse engineered netlist, creating two copies of the circuit that are referred to as *A* and *B* in Fig. 1. Every output of the duplicated circuits *A* and *B* are XORed together, which results in a logic *1* when a difference between the two outputs is observed. The outputs of all XORs are then applied to an OR gate, generating a signal that evaluates to logic *1* only when a difference between the two circuits is observed.

As the only varying signals between the *A* and *B* copies of the circuit are the key nets, whenever a condition that generates a logic *1* at the output of the miter circuit is observed, at least one key value is pruned from the key-space. The inputs to the miter circuit that result in such a condition are called distinguishing input patterns (DIPs).

Once a DIP is generated, the DIP is applied to an activated IC to determine the correct output. The resulting DIP and correct output are included as additional constraints to the SAT solver. The process is repeated until no further DIPs are found, resulting in the determination of a working key that satisfies the current SAT constraints.

III. Effect of Logical Reconvergence on SAT Attack Resiliency

The reconvergence of paths for each logic level of a circuit is evaluated to characterize the effect on the ability of the SAT attack to efficiently execute. Controlling the level of reconvergence is completed probabilistically by limiting the number of available nodes that connect to the next level of logic. As logic level l requires $2*n$ inputs for n 2-input gates, limiting the number of gates connected to the input of logic level l to less than $2*n$ results in the reconvergence of logical paths in the netlist.

The results of probabilistically controlling the logical reconvergence are shown in Fig. 2, where the reconvergence ratio is swept from 0.0 to 0.4. The reconvergence ratio represents a reduction in the number of inputs feeding logic level l, with the maximum number of inputs being $2*n$. A reconvergence ratio of 0.4 implies a total of $2*n*(1-0.4)$ inputs for logic level l. The generated circuits include a maximum of 16 inputs, with homogeneous gate types throughout the netlist. Each circuit topology is then secured, and 1,000 different random logic locking implementations are generated assuming a maximum overhead in area of 25%. The secured netlist is subjected to 100 different random variable orderings to characterize the level of security against the SAT attack [17]. The average number and the largest minimum number (provided above each data point in Fig. 2) of SAT iterations across the 1,000 analyzed benchmark circuits is shown for each gate type and reconvergence ratio in Fig. 2.

The minimum number of iterations decreases as the reconvergence ratio increases for all gate types. The results indicate an increasing challenge to reduce the amount of key information leaked by each DIP as the ratio increases. The high level of logical fanout associated with the reconvergent paths allows for the incorrect functionality of the circuit due to an erroneous key bit to spread across a larger percentage of the IC. An increased level of incorrect information seen at the output of an IC, therefore, provides an adversary executing the SAT attack with greater detail of the functionality of the IC.

The type of logic gate used in the netlist also effects the level of security provided against the SAT attack. As shown in Fig. 2, a slower degradation in the number of iterations is seen for the NAND and NOR gates. The negation of the NAND and NOR logic prevents a controlling input from propagating to the output of the IC, which implies a masking of the incorrect logical functionality of the circuit as the level of reconvergence increases.

IV. MFFC Selection Overview

Logical reconvergence impacts the effectiveness of the SAT attack by, depending on the reconvergence ratio and type of gate, leaking different levels of incorrect key information. Logical fanouts and reconvergence are, therefore, important circuit parameters when considering where to insert key gates as certain locations provide increased leakage of topological and logical characteristics for the SAT attack to exploit.

Therefore, to increase the difficulty of executing the SAT attack for in-cone logic locking, the amount and location of leaked information that the SAT attack uses to generate DIPs must be controlled. To ensure that the fanout of logical nets does not create unintended leakage channels for the SAT attack to utilize, the concept of maximum fanout free cones (MFFCs) is developed. An MFFC is a cone of logic where any net

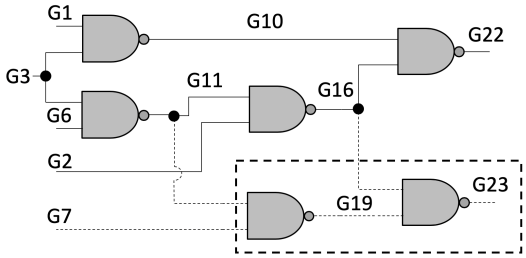

Fig. 3: Example circuit depicting the MFFC of net G23. Gates contained within the MFFC converge at G23. Dashed edges represent possible locations for XOR/XNOR locking that remain within the MFFC of G23.

within the MFFC converges to a single output node [18]. As an example, consider the ISCAS85 benchmark c17 circuit shown in Fig. 3, where the MFFC of net G23 are the gates contained within the dashed box. Since G16 includes a fanout gate (G22) that does not converge to G23, it is not considered a part of the given MFFC.

In addition to the use of an MFFC to quantify leakage channels in combinational logic, edges that converge to a common net in the MFFC (G23 in Fig. 3) are candidates for XOR/XNOR locking as all logical information leaked passes through the common node. The edges from $G16 \rightarrow G23$ and $G11 \rightarrow G19$ shown in Fig. 3 are, therefore, potential locations to insert a locking gate in addition to the fully enclosed edges of $G7 \rightarrow G19$ and $G19 \rightarrow G23$.

To determine the MFFC, a single node is randomly selected and Algorithm 1 is executed for a given list of node inputs. The logic cone of the node under investigation is traversed in topological order, which permits the evaluation of candidate nodes for inclusion into the MFFC by ensuring any output of the candidate node is also contained within the MFFC. The algorithm is applied recursively to input nodes that fit the MFFC criteria, termed $next\_lvl\_gates$ in Algorithm 1. Edges that fanout to a node contained within the MFFC are also added to the MFFC of the given node, but the originating node of the edge is not added to $next\_lvl\_gates$ since all of the node outputs are not contained within the MFFC.

Once the MFFC of each node is calculated, the MFFCs are sorted by a weighted sum of controllability to determine which MFFC is best suited for the insertion of key locked gates. The controllability is heuristically determined by estimating the probability of each node being a logic *1* and a logic *0* and then calculating the absolute difference between the determined probabilities. The controllability values for the nodes in an MFFC are then averaged together and multiplied by the number of nodes in the MFFC, or the number of key gates to insert if the number of locking gates is fully accommodated by the given MFFC.

After the weighted MFFCs are determined, Algorithm 2 is executed to iteratively insert XOR/XNOR gates. The algorithm greedily chooses nets to insert an XOR/XNOR gate based on the order of the weighted MFFCs. The MFFC with the highest score is then iterated through in reverse order, as nets closer to the inputs of the IC are added last to the MFFC. The edges and nodes within the MFFC are examined for marks,

Algorithm 1: Determination of MFFC

```
Input: Logic level gates level_gates
next_lvl_gates = [];
for gate in level_gates do
    fanout_free = True;
    tmp_edges = [];
    foreach gate_output in gate_outputs do
        if gate_output in mffc_nodes then
            /* Add gate → gate_output to
               tmp_edges                    */
        end
        else
            fanout_free = False;
        end
    end
    if fanout_free then
        /* Add node to MFFC                */
        /* Add gate_inputs of gate to
           next_lvl_gates                   */
    end
end
if next_lvl_gates not empty then
    /* Recursively call function with
       next_lvl_nodes                       */
end
```

which represent a previous selection of the edge and/or a penalty applied based on the structure of the netlist. Structural penalties include nets that result in the generation of multiple correct keys, such as back to back key gates or multiple key gates in a buffer chain. The process is repeated until the number of desired key gates are inserted into the netlist.

Algorithm 2: Insertion of Logic Locking Gates

```
Input: Number of Gates to Insert num_gates,
Node MFFCs mffcs
while keys_inserted < num_gates do
    enc_node = get_xor_enc_node(mffcs);
    num_keys_inserted = insert_xor(enc_node);
    keys_inserted += num_keys_inserted;
end
/* Determination of XOR Node           */
/* Weighted MFFC selected and reversed
   so nodes close to inputs are
   selected                             */
for elem in reversed_mffc do
    /* Check elem has not been consumed
       and is not marked                */
    if elem in netlist_gates and not_marked then
        /* Select Node for locking     */
    end
end
```

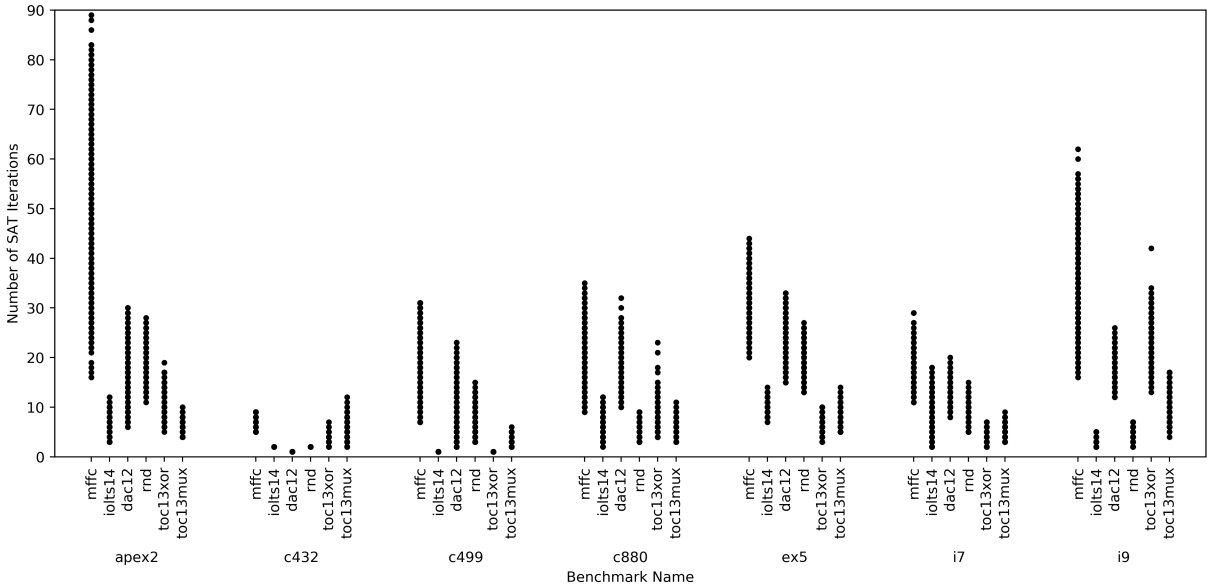

Fig. 4: Number of SAT attack iterations required for in-cone logic locking gate selection methodologies when allowing for a 5% overhead in area. Evaluations are completed for 1,000 random variable orders of each circuit topology.

V. Results of MFFC Based Gate Selection

The gate selection algorithm is applied to the ISCAS'85 benchmark circuits and to the combinational circuits from the Microelectronics Center of North Carolina (MCNC), as described in [9]. Characterization is performed for overheads in area of 5%, 10%, 25%, and 50%. A comparison of MFFC with iolts14 [19] , dac12 [20] , rnd [8], toc13xor [21] , and toc13mux [21] logic locking gate selection methodologies is performed. The number of inserted key gates across the different logic locking strategies remains constant as the area overhead is calculated based on the total number of gates in the original netlist, as was done in [9].

The analysis evaluating in-cone logic locking is performed on 1,000 random variable orderings of the netlist as the runtime of the SAT attack is impacted by the topological order of the netlist [17]. All gate selection algorithms are executed on a Xeon E52687W CPU running at 3 GHz with 95 GB of RAM. The SAT attack execution time is limited to a maximum of 24 hours (86,400 seconds).

The number of SAT iterations for each gate selection methodology when allowing for a 5% overhead in area is provided in Fig. 4. The results indicate that the developed MFFC based selection technique significantly increases the minimum number of iterations required to complete the SAT attack for most of the evaluated benchmark circuits. The one exception was the c880 benchmark, where the minimum number of iterations is essentially equal to the dac12 [20] methodology. The ability to meet or exceed the minimum number of iterations required to complete the SAT attack is an important measure of the security of in-cone logic locking techniques as a single point represented in Fig. 4 is not indicative of the difficulty an adversary faces when performing the SAT attack on an IC. Overall, the MFFC gate selection strategy increases the minimum number of iterations to complete the SAT attack by 61.8% and increases, on average, the mean number of iterations by 80.1% across all benchmark circuits.

VI. Conclusions

An algorithm utilizing maximum fanout free cones is utilized in this paper to increase the resiliency of in-cone logic locking against the SAT attack. Controlling the information leaked through logical fanouts by determining the maximum fanout free cone for each node results in an average increase in the minimum number of iterations to complete the SAT attack by 61.8% and an increase in the average number of iterations by 80.1% when allowing for an overhead in area of 5%. The increase in the level of security provided for by the MFFC selection algorithm for in-cone logic locking methodologies provides a comparison to techniques that add additional circuitry to control the output corruption of the circuit. Novel security measures must be evaluated against the dominant attack vector of which the measure is most susceptible to, which is not limited to the SAT attack. The developed MFFC algorithm increases the resiliency of in-cone logic locking, which is most susceptible to the SAT attack, and provides a means to compare amongst security methodologies.

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 and Grant CNS-1751032.

REFERENCES

[1] S. Skorobogatov and C. Woods, "Breakthrough Silicon Scanning Discovers Backdoor in Military Chip," *Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems*, pp. 23–40, September 2012.

[2] U.S Department of Commerce, "Defense Industrial Base Assessment: Counterfeit Electronics," 2010.

[3] 112th Congress, "Inquiry into Counterfeit Electronic Parts in the Department of Defense Supply Chain," 2012.

[4] R.W. Jarvis and M.G. McIntyre, "Split Manufacturing Method for Advanced Semiconductor Circuits," U.S Patent 7195931, 2004.

[5] J. Baukus, L. Chow, R. Cocchi, P. Ouyang, and B. Wang, "Camouflaging a Standard Cell Based Integrated Circuit," U.S Patent 8151235, 2012.

[6] J. Baukus, L. Chow, R. Cocchi, P. Ouyang, and B. Wang, "Building Block for Secure CMOS Logic Cell Library," U.S Patent 8111089, 2012.

[7] J. Baukus, L. Chow, J. Clark, and G. Harbison, "Conductive Channel Pseudo Block Process and Circuit to Inhibit Reverse Engineering," U.S Patent 8258583, 2012.

[8] J. A. Roy, F. Koushanfar, and I. L. Markov, "Ending Piracy of Integrated Circuits," *IEEE Computer*, Vol. 43, No. 10, pp. 30–38, October 2010.

[9] 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.

[10] Y. Xie and A. Srivastava, "Mitigating SAT Attack on Logic Locking," *Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems*. pp. 127–146, June 2016, Springer.

[11] M. Yasin, B. Mazumdar, J. Rajendran, and O. Sinanoglu, "SARLock: SAT Attack Resistant Logic Locking," *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust*, pp. 236–241, May 2016.

[12] M. Yasin, A. Sengupta, M. T. Nabeel, M. Ashraf, J. Rajendran, and O. Sinanoglu, "Provably-Secure Logic Locking: From Theory To Practice," *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 1601–1618, November 2017.

[13] K. Juretus and I. Savidis, "Time Domain Sequential Locking for Increased Security," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 1–5, May 2018.

[14] K. Juretus and I. Savidis, "Enhanced Circuit Security Through Hidden State Transitions," *Proceedings of the Government Microcircuit Applications and Critical Technology Conference*, pp. 1–4, March 2018.

[15] K. Shamsi, M. Li, T. Meade, Z. Zhao, D. Z. Pan, and Y. Jin, "AppSAT: Approximately Deobfuscating Integrated Circuits," *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust*, pp. 95–100, May 2017.

[16] Y. Shen and H. Zhou, "Double DIP: Re-Evaluating Security of Logic Encryption Algorithms," *Proceedings of the Great Lakes Symposium on VLSI*, pp. 179–184, May 2017.

[17] K. Juretus and I. Savidis, "Importance of Multi-parameter SAT Attack Exploration for Integrated Circuit Security," *Proceedings of the Asia Pacific Conference in Circuits and Systems*, pp. 1–4, October 2018.

[18] J. Cong and Y. Ding, "On Area/Depth Trade-off in LUT-based FPGA Technology Mapping," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 2, No. 2, pp. 137–148, June 1994.

[19] S. Dupuis, P. S. Ba, G. Di Natale, M. L. Flottes, and B. Rouzeyre, "A Novel Hardware Logic Encryption Technique for Thwarting Illegal Overproduction and Hardware Trojans," *Proceeding of the IEEE International On-Line Testing Symposium*, pp. 49–54, July 2014.

[20] M. Yasin, J. J. Rajendran, O. Sinanoglu, and R. Karri, "On Improving the Security of Logic Locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 35, No. 9, pp. 1411–1424, September 2016.

[21] J. Rajendran, H. Zhang, C. Zhang, G. Rose, Y. Pino, O. Sinanoglu, and R. Karri, "Fault Analysis-Based Logic Encryption," *IEEE Transactions on Computers*, Vol. 64, No. 2, pp. 410–424, February 2015.