Toward Efficient Evaluation of Logic Encryption Schemes: Models and Metrics

Yinghua Hu1  Vivek V. Menon2  Andrew Schmidt2  Joshua Monson2
Matthew French2  Pierluigi Nuzzo1

1 Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA, USA, {yinghua, nuzzo}@usc.edu
2 Information Sciences Institute, University of Southern California, Arlington, VA, USA, {vivekv, aschmidt, jmonson, mfrench}@isi.edu

ABSTRACT

Research in logic encryption over the last decade has resulted in various techniques to prevent different security threats such as Trojan insertion, intellectual property leakage, and reverse engineering. However, there is little agreement on a uniform set of metrics and models to efficiently assess the achieved security level and the trade-offs between security and overhead. This paper addresses the above challenges by relying on a general logic encryption architecture that can encompass all the existing techniques, and a uniform set of metrics that can capture multiple, possibly conflicting, security concerns. We apply our modeling framework to four state-of-the-art encryption techniques, showing that it enables fast and accurate evaluation of design trade-offs and provides support for the implementation of compound encryption methods. 1

1 INTRODUCTION

Integrated circuits (ICs) often represent the ultimate root of trust of modern computing systems. However, the decentralization of the IC design and manufacturing process over the years, involving multiple players in the supply chain, has increasingly raised the risk of hardware security threats from untrusted third parties.

Logic encryption aims to counteract some of these threats by appropriately modifying the logic of a circuit, that is, by adding extra components and a set of key inputs such that the functionality of the circuit cannot be revealed until the correct value of the key is applied. Several logic encryption methods have been proposed over the last decade to protect the designs from threats such as intellectual property (IP) piracy, reverse engineering, and hardware Trojan insertion (see, e.g., [6, 7, 9, 13, 17]). However, existing techniques are often tailored to specific attacker models and security concerns, and rely on different metrics to evaluate their effectiveness. It is then difficult to quantify the security of different methods, rigorously evaluate the inherent trade-offs between different security concerns, and systematically contrast their strength with traditional area, delay, and power metrics.

This paper introduces a formal modeling framework for the evaluation of logic encryption schemes and the exploration of the associated design space. We rely on a general functional model for logic encryption that can encompass all the existing methods. Based on this general model, we make the following contributions:

• We define a set of metrics that can formally capture multiple, possibly conflicting, security concerns that are key to the design of logic encryption schemes, such as functional corruption and resilience to different attacks, thus providing a common ground to compare different methods.
• We develop compact models to efficiently quantify the quality and resilience of four methods, including state-of-the-art logic encryption techniques, and enable trade-off evaluation between different security concerns.

Simulation results on a set of ISCAS benchmark circuits show the effectiveness of our modeling framework for fast and accurate evaluation of the design trade-offs. Our models produce conservative estimates of resilience, which are closer to the measurements than previous approaches. Finally, our approach can provide quantitative support to inform system-level decisions across multiple logic encryption strategies as well as the implementation of compound strategies, which can be necessary for providing high levels of protection against different threats with limited overhead.

2 BACKGROUND AND RELATED WORK

Logic encryption techniques have originally focused mostly on a subset of security concerns, and lacked methods to systematically quantify the level of protection against different (and potentially unknown) hardware attacks. A class of methods, such as fault analysis-based logic locking (FLL) [6], mostly focuses on providing high output error rates when applying a wrong key, for example, by appropriately inserting key-controlled XOR and XNOR gates in the circuit netlist. Another class of techniques, based on one-point functions, such as SARLock [17], aims, instead, to provide resilience to SAT-based attacks, a category of attacks using satisfiability (SAT) solving to efficiently prune the space of possible keys [12]. These methods require an exponential number of SAT-attack iterations in the size of the key to unlock the circuit, but tend to expose a close approximation of the correct circuit function. Efforts toward a comprehensive encryption framework have only started to appear.

Stripped functionality logic locking (SFLL) [19] has been recently proposed as a scheme for provably secure encryption with respect to a broad set of quantifiable security concerns, including error rate and resilience to SAT attacks as well as removal attacks, meaning to remove the encryption logic from the circuit. However, while the average number of SAT-attack iterations grow exponentially with the key size, the worst-case SAT-attack duration can become unacceptably low, which calls for mechanisms to explore the combination of concepts from SFLL with other schemes. Zhou [20] provides a theoretical analysis of the contention between error rate and SAT-attack resilience in logic encryption, drawing from concepts in learning theory [14]. Along the same direction, Shamsi et al. [9] develop diversified tree logic (DTL) as a scheme capable of increasing the error rate of SAT- resilient protection schemes in a tunable manner. A recent effort [10] adopts a game-theoretic approach to formalize notions of secrecy and resilience that account for the impact of learnability of the encrypted function and information leakage from
We denote by $Q$ the $Q$-value of the PO is inverted when the circuit function. Consistently with the literature [9], we define the amount of output error induced by logic encryption to protect and resilience of encryption.

Finally, we distinguish between logic encryption, which augments the circuit function via additional components and key bits, and obfuscation [1, 3], which is concerned with hiding the function of a circuit or program (without altering it) to make it unintelligible from its structure. In this paper, we focus on the functional aspects of logic encryption, and leave the modeling of its interactions with obfuscation as a direction for future work.

3 LOGIC ENCRYPTION: MODELS AND METRICS

We denote by $|S|$ the cardinality of a set $S$. We represent a combinational logic circuit with primary input (PI) ports $I$ and primary output (PO) ports $O$ by its Boolean function $f : \mathbb{B}^n \rightarrow \mathbb{B}^m$, where $n = |I|$ and $m = |O|$, and its netlist, modeled as a labelled directed graph $G$. Both $f$ and $G$ may be parameterized by a set of configuration parameters $P$, with values in $P$, related to both the circuit function and implementation. Given a function $f$, logic encryption creates a new function $f' : \mathbb{B}^n \rightarrow \mathbb{B}^m$, where $l = |K|$ and $K$ is the set of key input ports added to the netlist. There exists $k^* \in \mathbb{B}^l$ such that $\forall i \in \mathbb{B}^n, f(i) \equiv f'(i, k^*)$. We call $k^*$ the correct key. We wish to express $f'$ as a function of $f$ and the encryption logic.

3.1 A General Functional Model

We build on the recent literature [18–20] to define a general model, capable of representing the behavior of all the existing logic encryption schemes, as shown in Fig. 1. The function $g(i, k)$ maps an input and key value to a flip signal, which is combined with the output of $f(i)$ via a XOR gate to produce the encrypted PO. The value of the PO is inverted when the flip signal is one. We assume that $g$ is parameterized by a set $Q$ of configuration parameters, with values in $Q$ related to a specific encryption technique.

3.2 Security-Driven Metrics

We can describe how the circuit output is affected by logic encryption via an error table, such as the ones shown in Tab. 1. Based on the general functional model above and the associated error tables, we define a set of security-driven metrics that capture the quality and resilience of encryption.

**Functional Corruptibility.** Functional corruptibility quantifies the amount of output error induced by logic encryption to protect the circuit function. Consistently with the literature [9], we define the functional corruptibility $E_{FC}$ as the ratio between the number of corrupted output values and the total number of primary input and key configurations (the entries in the error table), i.e.,

$$E_{FC} = \frac{1}{2^n l} \sum_{i \in \mathbb{B}^n} \sum_{k \in \mathbb{B}^l} \mathbb{I}(f(i) \neq f'(i, k)),$$

where $\mathbb{I}(A)$ is the indicator function, evaluating to 1 if and only if event $A$ occurs.

**SAT Attack Resilience ($E_{SAT}$).** A SAT attack [12] assumes that the attacker has access to the encrypted netlist and an operational (deobfuscated) circuit, used as an oracle, to query for correct input/output pairs. The goal is to reconstruct the exact circuit function by retrieving a correct key. At each iteration, the attack solves a SAT problem to search for a distinguishing input pattern (DIP), that is, an input pattern $i$ that provides different output values for different keys, i.e., such that $\exists k_1 \neq k_2, f'(i, k_1) \neq f'(i, k_2)$. The attack then queries the oracle to find the correct output $f(i)$ and incorporate this information in the original SAT formula to constrain the search space for the following iteration. Therefore, all the keys leading to an incorrect output value for the current DIP will be pruned out of the search. Once the SAT solver cannot find a new DIP, the SAT attack terminates marking the remaining keys as correct.

Consistently with the literature [17, 19], we quantify the hardness of this attack using the number of SAT queries, hence the number of DIPs, required to obtain the circuit function. Computing this number in closed form is challenging, since it relates to solving a combinatorial search problem, in which the search space generally depends on the circuit properties and the search heuristics on the specific solver or algorithm adopted. Current approaches [19] adopt probabilistic models, where the expected number of DIPs is computed under the assumption that the input patterns are searched according to a uniform distribution. We adopt, instead, a worst-case conservative model and use the minimum number of DIPs to quantify the guarantees of of an encryption technique in terms of SAT-attack resilience. The duration of the attack also depends on the circuit size and structure, since they affect the runtime of each SAT query. In this paper, we regard the runtime of each SAT query as a constant and leave a more accurate modeling of the duration of the attack for future work.

**Approximate SAT-Attack Resilience ($E_{APP}$).** Approximate SAT attacks, such as AppSAT [8] and Double-DIP [11], perform a variant of a SAT attack but terminate earlier, when the error rate at the PO is “low enough”, providing a sufficient approximation of the circuit function. In this paper, we take a worst-case approach by assuming that an approximate SAT attack terminates in negligible time, and define the approximate SAT-attack resilience ($E_{APP}$) as the minimum residual error rate that can be obtained with an
Table 2: Security metrics for four logic encryption techniques \((n = |I|, m = |O|, \text{ and } l = |K|)\).
We evaluated models and metrics on a set of ISCAS benchmark circuits. The blue areas in Fig. 4a-c pictorially represent, as a continuum, the feasible encryption space for different methods and user requirements. For example, Fig. 4a shows that a functional corruptibility ($E_{FC}$) as high as 0.25 can still be achieved with SARLock; however, it can only be implemented for very low, and therefore impractical, key sizes. Fig. 4b highlights the trade-off between SAT attack resilience ($t_{SAT}$) and approximate SAT attack resilience ($E_{APP}$) in DTL. As expected, DTL is able to increase $E_{APP}$ and $E_{FC}$ with the cost of decreasing $t_{SAT}$. The highest possible $E_{FC}$ achieved by DTL is higher than that of SARLock in Fig. 4a. Finally, Fig. 4c exposes a trade-off between $E_{FC}$ and $E_{APP}$ in SFLL. It shows that increasing $E_{FC}$ adversely impacts $E_{APP}$, possibly due to the fact that, as $E_{FC}$ increases, the error distribution is not uniform; while the peak error rate increases, the error can become significantly low for some of the incorrect keys.

We further implemented all the encryption configurations explored in Fig. 4a-c on four ISCAS benchmark circuits, generating 1473 netlists in 15 minutes, to compare the model predictions with the measurements. We used open-source libraries to simulate SAT attacks [12] and report the actual value of $t_{SAT}$. We empirically estimated $E_{FC}$ by averaging the functional corruptibility over 500 logic simulations on the encrypted netlists. By using a similar procedure, an empirical estimate for $E_{APP}$ was obtained by taking the average over 500 logic simulations for each key pattern, and then the minimum corruptibility value over 100 incorrect key patterns. Fig. 6 reports the results for four ISCAS benchmark circuits, showing that the empirical resilience would always exceed the one predicted by our model (blue bar) for both $t_{SAT}$ and $E_{APP}$. For 26% of the design (red bar) the empirical $E_{FC}$ proved to be smaller than the predicted one by a negligible margin ($4 \times 10^{-3}$), which is within the error affecting our simulation-based empirical estimates.

To compare our SAT resilience model for SFLL with the measured number of DIPs, we simulate SAT attacks on the encrypted netlists of four ISCAS circuits using SFLL-HD. For each combination of key size and value of $h$, we generate 10 netlists, by randomly permuting the order of the gates, and compute the average number of DIPs over 10 SAT attacks. As shown in Fig. 5, the average number of DIPs is accurately predicted by our greedy algorithms for all key sizes and $h$ values, while it diverges significantly from the estimates based on the probabilistic model in the literature [19], especially when $h$ is close to zero.

The analysis above suggests that combining multiple techniques can help alleviate the design trade-offs and achieve high security levels across multiple metrics. To test this hypothesis, we encrypted the ISCAS circuit C880 with both SARLock and DTL, by using a logic OR gate to combine their output (flip) signals. We then combined the output of the OR gate with the output of the original circuit via an XOR gate. Fig. 4d shows that the compound strategy significantly alleviates the trade-offs posed by SARLock alone, making it possible to achieve both high functional corruptibility and SAT-attack resilience. For example, the topmost configuration in Fig. 4d achieves $t_{SAT} \geq 2^{14}$ and $E_{FC} \geq 0.48$, which cannot be obtained with SARLock or DTL alone. The compound scheme, where the SARLock block has key size 13 and the DTL block has key size 4, with two AND gates being replaced by one XOR gate in the first layer, is able to provide both high $t_{SAT}$ and $E_{FC}$. Measurement results are, again, in total agreement with our model predictions.
6 CONCLUSIONS

Simulation results show the effectiveness of the proposed models and metrics for fast and accurate evaluation of the design trade-offs as well as the exploration of compound logic encryption strategies, which may be required for protecting against different threats with small overhead. Future extension of this work include the incorporation of overhead models as well as support for structural and learning-based attacks. We plan to also investigate the extension of our framework to sequential logic encryption methods. Finally, we plan to further develop an automated design and verification environment [15] leveraging our models and methods to perform design space exploration and inform system-level design decisions across multiple encryption schemes.

ACKNOWLEDGMENTS

This work was partially sponsored by the Air Force Research Laboratory (AFRL) and the Defense Advanced Research Projects Agency (DARPA) under agreement number FA8560-18-1-7817.

REFERENCES

[1] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, and K. Yang. On the (im) possibility of obfuscating programs. In Annual International Cryptology Conference, pages 1–18. Springer, 2001.
[2] P. Chakraborty, J. Cruz, and S. Bhunia. Sail: Machine learning guided structural analysis attack on hardware obfuscation. In 2018 Asian Hardware Oriented Security and Trust Symposium (AsianHOST), pages 56–61. IEEE, 2018.
[3] S. Goldwasser and G. N. Rothblum. On best-possible obfuscation. In Theory of Cryptography Conference, pages 194–213. Springer, 2007.
[4] Y. Hu, V. Venugopalan, A. Schmidt, J. Monzón, M. French, and P. Nuzzo. Security-driven metrics and models for efficient evaluation of logic encryption schemes. In 2019 17th ACM/IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE). ACM, 2019.
[5] B. Korte, J. Vygen, B. Korte, and J. Vygen. Combinatorial optimization, volume 2. Springer, 2012.
[6] J. Rajendran, H. Zhang, C. Zhang, G. S. Rose, Y. Pino, O. Sinanoglu, and R. Karri. Fault Analysis-Based Logic Encryption. IEEE Transactions on Computers, 64(2):410–424, 2015.
[7] J. A. Roy, F. Koushanfar, and I. L. Markov. Ending Piracy of Integrated Circuits. Computer, 43(10):30–38, 2010.
[8] K. Shamsi, M. Li, T. Meade, Z. Zhao, D. Z. Pan, and Y. Jin. Appsat: Approximately deobfuscating integrated circuits. In Hardware Oriented Security and Trust (HOST), 2017 IEEE International Symposium on, pages 95–100. IEEE, 2017.
[9] K. Shamsi, T. Meade, M. Li, D. Z. Pan, and Y. Jin. On the approximation residulency of logic locking and ic camouflaging schemes. IEEE Transactions on Information Forensics and Security, 14(2):347–359, 2019.
[10] K. Shamsi, D. Z. Pan, and Y. Jin. On the impossibility of approximation-resilient circuit locking. In Hardware Oriented Security and Trust (HOST), 2019 IEEE International Symposium on, pages 161–170. IEEE, 2019.
[11] Y. Shen and H. Zhou. Double dip: Re-evaluating security of logic encryption algorithms. In Proceedings of the on Great Lakes Symposium on VLSI 2017, pages 179–184. ACM, 2017.
[12] P. Subramaniam, S. Ray, and S. Malik. Evaluating the security of logic encryption algorithms. In 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pages 137–143. ACM, 2015.
[13] M. Tehranipoor and F. Koushanfar. A survey of hardware trojan taxonomy and detection. IEEE design & test of computers, 27(1):10–25, 2010.
[14] L. G. Valiant. A theory of the learnable. In Proceedings of the sixteenth annual ACM symposium on Theory of computing, pages 436–445. ACM, 1984.
[15] V. Venugopalan, G. Kolhe, A. Schmidt, J. Monson, M. French, Y. Hu, P. A. Beerel, and P. Nuzzo. System-level framework for logic obfuscation with quantified metrics for evaluation. In 2019 IEEE Secure Development Conference (SecDev). IEEE, 2019.

[16] Y. Xie and A. Srivastava. Anti-SAT: Mitigating SAT Attack on Logic Locking. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, pages 1–1, 2018.

[17] 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, pages 236–241. IEEE, May 2016.

[18] M. Yasin, B. Mazumdar, J. J. Rajendran, and O. Sinanoglu. Ttlock: Tenacious and traceless logic locking. In 2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pages 166–166. IEEE, 2017.

[19] M. Yasin, A. Sengupta, M. T. Nabeel, M. Ashraf, J. J. Rajendran, and O. Sinanoglu. Provably-Secure Logic Locking. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security - CCS ’17, 2017.

[20] H. Zhou. A humble theory and application for logic encryption. IACR Cryptology ePrint Archive, 2017:96, 2017.