PREPRINT - published in VLSI-DAT 2019, doi: 10.1109/VLSI-DAT.2019.8741531

A Critical Evaluation of the Paradigm Shift in the Design of Logic Encryption Algorithms

Dominik Šišejković*, Farhad Merchant*, Rainer Leupers*, Gerd Ascheid* and Volker Kiefer†
*Institute for Communication Technologies and Embedded Systems, RWTH Aachen University, Germany
{sisejkovic, merchantf, leupers, ascheid}@ice.rwth-aachen.de
†Hensoldt Cyber GmbH, Germany
volker.kiefer@hensoldt-cyber.com

***Abstract*—The globalization of the integrated circuit supply chain has given rise to major security concerns ranging from intellectual property piracy to hardware Trojans. Logic encryption is a promising solution to tackle these threats. Recently, a Boolean satisfiability attack capable of unlocking existing logic encryption techniques was introduced. This attack initiated a paradigm shift in the design of logic encryption algorithms. However, recent approaches have been strongly focusing on low-cost countermeasures that unfortunately lead to low functional and structural corruption. In this paper, we show that a simple approach can offer provable security and more than 99% corruption if a higher area overhead is accepted. Our results strongly suggest that future proposals should consider higher overheads or more realistic circuit sizes for the evaluation of modern logic encryption algorithms.**

***Index Terms*—Hardware Security, Logic Encryption, SAT Attack, Metrics, Hardware Trojans**

I. Introduction

Establishing a leading-edge semiconductor foundry costs at least $3 to $5 billion, therefore the production lines must keep running at full capacity to re-compensate the investment [1]. This immense cost has forced many design houses to outsource their Integrated Circuit (IC) design and fabrication services to off-site companies. The active participation of external parties in the IC supply chain introduces serious hardware security threats, such as Intellectual Property (IP) piracy [2], counterfeiting [3] and hardware Trojans [4]. As a reaction to the eminent threats, a variety of countermeasures has been introduced, including logic encryption, IC camouflaging [5] and watermarking [6].

Logic encryption (also known as obfuscation or logic locking) is a premier protection approach against the mentioned threats, based on the insertion of additional key-controlled gates into the design to mask the original functionality [7]. The locked design only performs correctly for all input patterns if the correct secret key is provided to the additional gates. The inserted gates represent the trade-off between security and overhead. More gates can imply an area, power and delay overhead, but also higher security against existing attacks.

However, the security of logic encryption algorithms has recently been challenged with the introduction of a Boolean Satisfiability (SAT) attack that was able to decrypt all previously known approaches. This has led to a paradigm shift in designing logic encryption algorithms; new approaches focus on providing SAT-resilience. Unfortunately, resilience against the SAT attack comes with the price of low corruption rates (e.g., an incorrect key generates a correct output for many input patterns). In addition, a common pattern can be identified; existing proposals focus on fully minimizing overheads while offering SAT resilience, thereby unintentionally enabling new attack vectors and very low corruption rates.

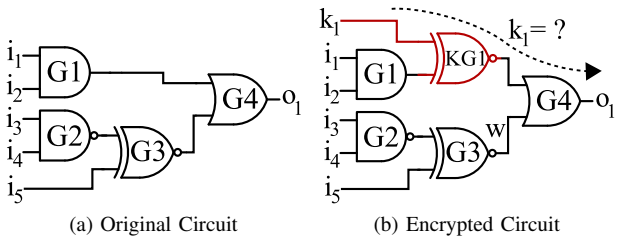
Contributions: In our view, the primary goal of logic encryption design should be to first identify the minimum cost yielding sufficient security and afterwards optimize the overhead impact. In this work, we show that it is possible to achieve high structural and functional corruption as well as attack resilience through the simple random insertion methodology if a higher cost is considered. To the best of our knowledge, this is the first analysis focusing on higher-overhead logic encryption. With the presented results, we indicate the necessity of *re-evaluating existing logic encryption methodologies* with regards to higher overheads, especially since no consensus on acceptable overheads exists.

II. Background

Logic Encryption: Logic encryption relies on obfuscating the functionality and topology of a circuit through the insertion of additional gates. These gates are often referred to as *key gates*, since they are driven by a set of key inputs. If an incorrect key is set to the key inputs, the IC produces incorrect outputs for at least some input patterns. Otherwise, a correct key always implies a correct output. The encryption is usually done on the gate-level netlist, before the design is sent to the foundry. After fabrication, the IC is returned to the designer for activation. The activation key is typically stored in a tamper-proof memory. Logic encryption hinders the process of reverse engineering the design, since the correct key is only known to the original designer [8].

The original proposal known as EPIC [9] is based on the insertion of XOR/XNOR gates at random positions in the netlist. For each bit of the key, one XOR/XNOR gate is inserted at a random location to act as a buffer if the correct key bit is applied. To hinder a simple removal of the gate, the XOR/XNOR gate is combined with an inverter. Since the attacker cannot know if the inverter is part of the original netlist, a removal could disrupt the original functionality.

An example of a logically encrypted netlist is presented in Figure 1. The original circuit (Figure 1 (a)) is encrypted with the insertion of one XNOR gate KG1 (Figure 1 (b)). If the correct key is applied ($k_1 = 1$), KG1 operates as a buffer,



(a) Original Circuit (b) Encrypted Circuit

Fig. 1: Example of Logic Encryption Using One XNOR Gate.

preserving the original functionality of the netlist. Otherwise, an incorrect key ($k_1 = 0$) corrupts the output of gate G1.

Security Metrics: Measuring the security of logic encryption algorithms is still an unsolved challenge. However, we base our analysis on the measure of *structural* and *functional* corruption [10]. Structural corruption describes the corruption of the original circuit structure (topology), whereas functional (output) corruption captures the output corruptibility for incorrect keys. In terms of security, it is beneficial to have high corruption rates in both aspects. High structural corruption is provided by the spatial distribution of key gates that distort the original circuit structure. The more the original topology is altered, the more difficult it is to reverse engineer and understand an encrypted IC out of its structural features. High functional corruption implies that many incorrect keys generate a corrupted output for many input patterns. Functional corruption is sometimes measured through the Hamming distances between the correct and corrupted outputs [11]. This measure is only applicable in terms of attacks that manipulate key bits to minimize the Hamming distance. Apart from the Hill-Climbing attack [12], no such strong attack has been introduced so far. Therefore, we do not consider the Hamming distance for evaluation.

Attack Model: The attack model is defined as follows: (i) the attacker has access to a fabricated and activated IC to use as an oracle (e.g., acquired from open market), (ii) the attacker has access to a netlist of the encrypted IC and (iii) the positions of the key inputs are known. Because of assumptions (i) and (ii), the attacker can compare the input/output relation of the netlist to the activated IC for a selected key.

Attacks on Logic Encryption: One prominent attack on logic encryption is known as the path sensitization attack [13]. This attack exploits the knowledge about the positions of the key gates in the netlist. By calculating a dedicated input pattern, the value of the correct key bit of a single key gate can be sensitized to a primary output. For example, to sensitize the value of k_1 to the output o_1 in Figure 1 (b), a non-controlling value must be forced to the input wire w of gate G4 (0 for OR gate). Additionally, the output of G1 is set to 1. If $w = 0$, gate G4 propagates the output of KG1 to o_1. This can be achieved by assigning 11010 to the inputs i_1 to i_5. This attack can be circumvented by carefully inserting key gates at preselected positions [13]. However, this attack usually just yields a part of the key, as not all key gates are vulnerable to sensitization.

Currently, the strongest attacks on logic encryption are based on the original SAT attack [8]. The SAT attack utilizes SAT solvers to eliminate incorrect keys. Instead of looking for one key, the SAT attack looks for the equivalence class of keys. An equivalence class is defined as a group of keys that cause the same input/output relations in the encrypted circuit for a set of input patterns. In each iteration, the attack calculates special input patterns known as Distinguishing Input Patterns (DIPs). These patterns are used to rule out incorrect keys. The efficiency of the attack lies in its ability to eliminate multiple keys for one DIP. Another version of the SAT attack is known as the partial-break attack [8]. This attack is able to extract partial information about the key values by only analyzing isolated subsets of the circuit.

III. Security Analysis

In recent years, a noticeable shift in the design objectives of logic encryption algorithms has been observed. Before 2015, encryption algorithms were focusing on different security goals, since a strong attack was not present at that time. For example, the random insertion provided by EPIC [9] or the XOR/MUX insertions with the goal to maximize the Hamming distance between the correct and incorrect outputs [14]. In 2015, the SAT attack was introduced [8]. All previous logic encryption algorithms have been shown to be vulnerable to this attack for lower area overheads (approximately up to 50%). Therefore, after 2015, a paradigm shift in the design objectives of encryption algorithms has been observed; the majority of recently published algorithms focus on mitigating the SAT attack [15]. So far, to countermeasure the SAT attack, logic encryption algorithms have relied on achieving very low corruption rates. If a circuit behaves correctly for any incorrect key for all input patterns except one (distinct to each key), then the SAT attack can eliminate only one incorrect key per iteration; thereby running exponentially long to the key size. In addition to low corruption, SAT-resilient approaches often leave structural traces vulnerable to removal attacks [16], [17]. These attacks are able to identify the SAT-countermeasure circuitry and simply remove it or bypass its functionality. One promising approach is provided by Stripped-Functionality Logic Locking (SFLL) [18]. Even though it offers SAT and removal resilience, the SFLL encrypted IC is minimally different in functionality for incorrect keys, e.g., the designer must choose to protect only a set of preselected input patterns.

Through the analysis, we identified two major issues. First, regardless of the latest attack focus, logic encryption should provide two important security aspects: structural and functional corruption. Secondly, most authors have focused on achieving attack resilience with a minimum amount of encryption overhead instead of assuring security aspects regardless of the initial cost [15], [18]. The commonly assumed values are up to 50% area overhead. However, this percentage is not reasonably chosen, as most authors solely focus on small benchmarks (a few hundred to a few thousand gates). In this context, a small area overhead implies only a small amount of additional gates used for encryption (high overhead in small circuits still results in small circuits). Therefore, the applied algorithms cannot exercise high security. In our view, solely

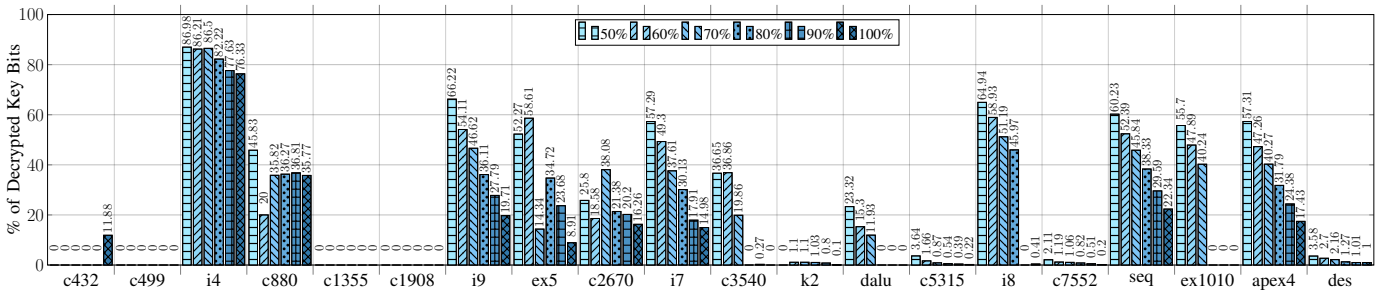


Fig. 2: Percentage of Key Bits Decrypted with the Partial-Break and Fault-Analysis Attack.

TABLE I: Resilience against the SAT Attack per Area Overhead: Marked (Blue) Fields Indicate an Unbroken IC and Each Entry Shows the Used Key Length.

IC	#Gates	50%	60%	70%	80%	90%	100%
c432	161	80	96	112	128	144	160
c499	203	101	121	141	162	182	202
i4	338	169	203	237	270	304	338
c880	384	192	230	268	306	345	383
c1355	547	273	328	382	437	491	546
c1908	881	440	528	616	704	792	880
i9	1035	518	621	725	828	932	1035
ex5	1055	528	633	739	844	950	1055
c2670	1194	597	716	835	954	1074	1193
i7	1315	658	789	920	1052	1184	1315
c3540	1670	835	1001	1168	1335	1502	1669
k2	1815	908	1089	1271	1452	1634	1815
dalu	2298	1149	1379	1609	1838	2068	2298
c5315	2308	1154	1384	1615	1846	2076	2307
i8	2464	1232	1478	1725	1971	2218	2464
c7552	3513	1756	2107	2458	2810	3161	3512
seq	3519	1760	2111	2463	2815	3167	3519
ex1010	5066	2533	3040	3546	4053	4559	5066
apex4	5360	2680	3216	3752	4288	4824	5360
des	6473	3237	3884	4531	5178	5826	6473
	Resilience:	**15%**	**19%**	**35%**	**30%**	**45%**	**55%**

using small benchmarks is not enough to draw conclusions about the security of logic encryption algorithms. This is further evaluated in Section IV. To this day, it is still unclear how much overhead is acceptable, since this is largely driven by industry and the needs of customers. Therefore, in our view, it is of great value to put the focus on assuring security instead of minimizing overheads.

IV. Experimental Evaluation

The objective of the experimental evaluation is to show that high corruption as well as SAT and removal attack resilience can be obtained with the simplest of techniques, if a higher area overhead is allowed.

A. Experimental Setup

For the analysis, we chose the simple logic encryption approach EPIC [9], as it offers convenient features: (i) ease of application (no special framework needed), (ii) simple overhead control (#key gates $\approx$ area overhead), (iii) high structural and functional corruption, (iv) resilience against removal attacks and (v) reasonable resilience against the path sensitization attack. A subset of the ISCAS'85 [19] and MCNC [20] benchmarks was chosen for the evaluation. The benchmarks are listed in Table I.

We evaluated two scenarios: (i) the resilience against the original SAT attack (Table I) and (ii) the percentage of decrypted key bits with the combination of the partial-break and fault-analysis attack (Figure 2). In the first scenario, the attack can only return a full key, while the attacks in the second scenario are able to return a partial key. In both cases, we ran the evaluation on all selected ICs encrypted with 50%, 60%, 70%, 80%, 90% and 100% area overhead. For easier comparison, the % overheads are marked with a *different variant of the color blue* in all presented data.

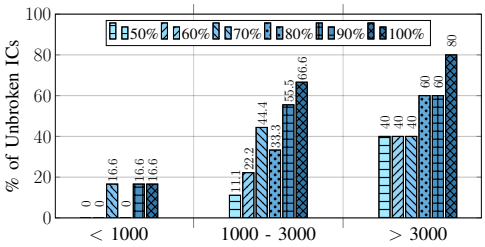


Fig. 3: Resilience against the SAT Attack per Circuit Size Categories (Num. of Gates).

The attack timeout is set to 10 hours, as suggested in the original work [8]. Experiments were run on an Intel i5 CPU@3.2 GHz with 8 GB of RAM. All shown results are sorted according to the IC size (number of gates).

B. Evaluation Results

The results of the first evaluation are presented in Table I. Each entry in the percentage columns denotes the used key length for the particular benchmark and area overhead. Here we define the *resilience rate* as the percentage of unbroken ICs for a given attack. All unbroken (resilient) benchmarks are marked blue. Additionally, the resilience rate per size categories is presented in Figure 3. It can be observed that the number of unbroken ICs grows with the increase of area overhead (resilience rate up to 55%), especially when larger ICs with more than 3000 gates are considered (resilience rate up to 80%). This indicates that larger designs can be secured with even less overhead and still assure high corruption rates. The increase of the resilience rate with increasing area overheads for EPIC is justified by two effects. First, a higher area overhead implies a larger key, i.e. more incorrect keys have to be ruled out by the applied attack to find the correct one. Secondly, a larger key implies a larger amount of added XOR/XNOR gates; these gates result in clauses that are harder to satisfy by the DPLL algorithm used in SAT solvers [8].

The second evaluation concerns the attacks that are able to partially decrypt a key (Figure 2). The percentage of decrypted

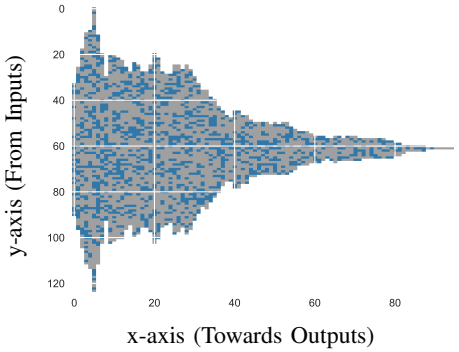


Fig. 4: Visualization of the dalu Circuit for 50% Overhead.

key bits falls with a higher overhead. This is especially important for circuits which are not broken by the general SAT attack. This evaluation indicates the same trend as before; a higher overhead can lead to an acceptable security level.

We measured the functional corruption rate for all ICs with an area overhead of 50% by applying 1000 input patterns for 1000 incorrect keys. All corruption rates were above 99%; only 1% of all measured inputs and incorrect keys generate a correct output. The corruption rate is expected to rise with increasing overheads. Therefore, the measured corruption can be considered as a lower-bound value.

The structural corruption is assured through the random spatial distribution of key gates that are very likely to change the output for incorrect keys and maximally corrupt the circuit topology. To visualize this effect, we present the middle-size circuit *dalu* with 50% area overhead due to encryption in the form of a topologically sorted netlist (Figure 4). Gates are shown as colored fields. All key gates are marker blue and original gates are marked grey. The x and y axis indicate the positions of the gates in the netlist. The visualization shows that the key gates are evenly distributed in the netlist, as expected by EPIC. After logic synthesis, a maximal corruption of the original structure is expected. Since the overhead in EPIC correlates with the amount of added key gates, this example results are valid for any circuit size.

The evaluation results show that both high attack resilience, as well as high structural and functional corruption can be achieved, if *more overhead is allowed*. Since achieving high corruption should have more priority over minimizing overhead, we showed that a secure encryption can be provided with the simplest of techniques. In addition, the results indicate that larger circuits are easier to secure with EPIC, even with lower overheads (Figure 3). This is a valuable conclusion as real life circuits tend to be much larger than a few hundred gates and the storage of larger keys is not a concern. Moreover, higher overhead encryption can be applied only to crucial parts of a design, yielding a lower total area cost. In this work, we are not claiming that, e.g., EPIC is fully SAT-resilient, but we show an important trend based on strong empirical evidence: simple algorithms with high favorable corruption aspects can be considered as *secure in practice* and should, therefore, be re-evaluated.

V. Conclusion

In this paper, we offered a critical analysis of a major concern in hardware security by showing that both resilient encryption and high corruption can easily be achieved if more overhead is allowed. The evaluation results strongly indicate that the existing algorithms are secure if either a higher area overhead is accepted or larger circuit benchmarks are considered. Therefore, future proposals should rethink the design of modern logic encryption algorithms by focusing on a more realistic security evaluation, instead of forcing minimum-overhead solutions. In future work, we plan to further investigate the correlation between the overhead and attack resilience to model the lower-bound on needed overhead to assure security.

References

[1] C. Mims, "The high cost of upholding Moore's law," *Technology Review*, vol. 113, no. 3, pp. 71–72, 2010.

[2] M. Rostami, F. Koushanfar, and R. Karri, "A primer on hardware security: Models, methods, and metrics," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1283–1295, Aug 2014.

[3] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, Aug 2014.

[4] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor, "Hardware trojans: Lessons learned after one decade of research," vol. 22, pp. 1–23, 05 2016.

[5] R. P. Cocchi, J. P. Baukus, L. W. Chow, and B. J. Wang, "Circuit camouflage integration for hardware IP protection," in *2014 51st ACM/EDAC/IEEE DAC*, June 2014, pp. 1–5.

[6] R. D. Newbould, D. L. Irby, J. D. Carothers, J. J. Rodriguez, and W. Holman, "Watermarking ICs for IP protection," *Electronics Letters*, vol. 38, no. 6, pp. 272–274, Mar 2002.

[7] M. Yasin and O. Sinanoglu, "Evolution of logic locking," in *2017 IFIP/IEEE VLSI-SoC*, Oct 2017, pp. 1–6.

[8] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *2015 IEEE HOST*, May 2015, pp. 137–143.

[9] J. A. Roy, F. Koushanfar, and I. L. Markov, "Epic: Ending piracy of integrated circuits," in *2008 DATE*, March 2008, pp. 1069–1074.

[10] H. Zhou, "A humble theory and application for logic encryption," *IACR Cryptology ePrint Archive*, vol. 2017, p. 696, 2017.

[11] S. Patnaik, M. Ashraf, J. Knechtel, and O. Sinanoglu, "Obfuscating the interconnects: Low-cost and resilient full-chip layout camouflaging," in *Proceedings of the 36th ICCAD*, ser. ICCAD '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 41–48.

[12] S. M. Plaza and I. L. Markov, "Solving the third-shift problem in IC piracy with test-aware logic locking," *IEEE TCAD*, vol. 34, no. 6, pp. 961–971, June 2015.

[13] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Security analysis of logic obfuscation," in *DAC 2012*, June 2012, pp. 83–89.

[14] J. Rajendran *et al.*, "Fault analysis-based logic encryption," *IEEE Transactions on Computers*, vol. 64, no. 2, pp. 410–424, Feb 2015.

[15] Y. Xie and A. Srivastava, "Anti-sat: Mitigating sat attack on logic locking," pp. 1–1, 2018.

[16] M. Yasin, B. Mazumdar, O. Sinanoglu, and J. Rajendran, "Removal attacks on logic locking and camouflaging techniques," *IEEE TETC*, pp. 1–1, 2017.

[17] X. Xu, B. Shakya, M. M. Tehranipoor, and D. Forte, "Novel bypass attack and bdd-based tradeoff analysis against all known logic locking attacks," in *CHES*, 2017.

[18] M. Yasin, A. Sengupta, M. T. Nabeel, M. Ashraf, J. J. Rajendran, and O. Sinanoglu, "Provably-secure logic locking: From theory to practice," in *Proceedings of the 2017 ACM CCS*. ACM, 2017, pp. 1601–1618.

[19] M. C. Hansen, H. Yalcin, and J. P. Hayes, "Unveiling the iscas-85 benchmarks: a case study in reverse engineering," *IEEE Design Test of Computers*, vol. 16, no. 3, pp. 72–80, 1999.

[20] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," in *IEEE ISCAS*, May 1989, pp. 1929–1934 vol.3.