Global Attack and Remedy on IC-Specific Logic Encryption

Amin Rezaei*, Ava Hedayatipour*, Hossein Sayadi*, Mehrdad Aliasgari*, and Hai Zhou†
*California State University Long Beach
{amin.rezaei, mehrdad.aliasgari, ava.hedayatipour, hossein.sayadi}@csulb.edu
†Northwestern University
haizhou@northwestern.edu

2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST) | 978-1-6654-8532-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/HOST54066.2022.9840128

***Abstract*—In recent years, semiconductor industry has outsourced the manufacturing to low-cost but not necessarily trusted foundries. This fabless business model encounters new security challenges, including piracy and overproduction. A well-studied solution to prevent unauthorized products from functioning is logic encryption, where a chip is encrypted using a key only known to the designer. However, the majority of the logic encryption solutions are vulnerable due to key uniformity and probing attacks. In this paper, we first present *GSAT*, a Global attack on existing IC-specific logic encryption schemes using the SAT model, that effectively deciphers the hidden global key pluggable to all the encrypted ICs. Next, we propose a highly secure and low-cost remedy called *SPLEnD*: Strong PUF-based Logic Encryption Design. Traditional IC-specific encryption schemes are vulnerable to *GSAT* attack, while *SPLEnD* not only effectively resists *GSAT*, but also balances security and efficiency.**

***Index Terms*—Logic Encryption; Logic Locking; Physical Unclonable Function; SAT-based Attack; Probing Attack**

I. Introduction and Background

Many leading-edge Integrated Circuit (IC) design houses have outsourced their fabrication to offshore foundries for lower labor and manufacturing costs. However, it is difficult to trust these foundries since there is no well-defined universal consummate law enforcement [1]. Also contributing to the distrust are the powerful reverse engineering tools that can extract and duplicate the netlist of the ICs in the market [2].

The main approach to preventing unauthorized access to the ICs is logic encryption (a.k.a. logic locking) that modifies a given netlist with the introduction of key inputs [3], [4]. To make the encrypted circuit functional, the correct key needs to be inserted into a tamper-proof memory by the designer before releasing the IC to the market. Most of the state-of-the-art works [3], [4], [5], [6], [7], [8] employ a uniform key in their designs. In this case, if an attacker deciphers the uniform key on one chip using SAT-based [9], [10] or probing attacks [11], [12], he/she can easily plug-in the key into other chips and make them functional. Therefore, with the existence of a uniform key, if an attacker reads out the correct key after insertion, the security of the entire scheme will be broken [13].

Due to process variations, every chip has slightly different performance. This fact can be exploited to provide every chip with a unique identity. A Physical Unclonable Function (PUF) takes inputs, called challenges, and produces outputs, called responses. Thus, every PUF is a function of Challenge-Response Pairs (CRPs) that are practically unique to each chip. Weak PUFs offer a limited number of CRPs with respect to the number of their building components, while strong PUFs offer an exponential number of CRPs [14].

In this paper, we first present an attack on existing IC-specific logic encryption methods [15], [16] to decipher the hidden global key. Next, to address the challenge of vulnerability to uniformity and probing attacks, we propose a highly secure and low-cost logic encryption design based on non-uniform keys. In this design, we first select a sub-circuit with one critical primary output, and then substitute it with a strong PUF feeding into a MUX. While the input bit length of the sub-circuit is sufficiently large, its observable pattern number is small. Furthermore, the input bits of the MUX and a small number of PUF inputs are programmable.

To the best of our knowledge, this is the first paper that reports the vulnerability of global key leakage on IC-specific logic encryption and proposes a low-cost solution to address it. Table I summarizes the symbols we use in this study.

Possible attacks on logic encryption can be categorized into two models, including SAT-based and probing attacks. For the first model, the assumption is that the attacker has full access to the physical layout of the encrypted circuit and also black-box access to an oracle. The original SAT-based attack [9] can only find out the local key inputs of the non-uniform methods. However, we propose a modified version of this attack in section II-A to decipher the global key as well. For the latter one, the model is even stronger; the attacker not only has the layout in hand, but also white-box access to an activated IC. Thus, he/she can easily probe the wires associated with the key inputs and find out the correct key [12]. In the case of uniform keys, the deciphered key can be plugged into other chips and make them functional. Thus, it is essential to adopt non-uniform and IC-specific key inputs to make the probing attack incompetent.

II. Contributions and Discussions

A. GSAT Attack

Wendt and Potkonjak [15] have proposed replacing an arbitrary sub-circuit with a PUF and a Field-Programmable Gate Array (FPGA). The FPGA is programmed by the designer in order to implement the original replaced circuitry in conjunction with the corresponding PUF behavior. Since the inputs of the FPGA can take any value from the internal signals in the circuit, the designer needs to know the entire CRP space of each PUF. As a result, he/she is limited to the selection of weak PUFs. After the activation process, the

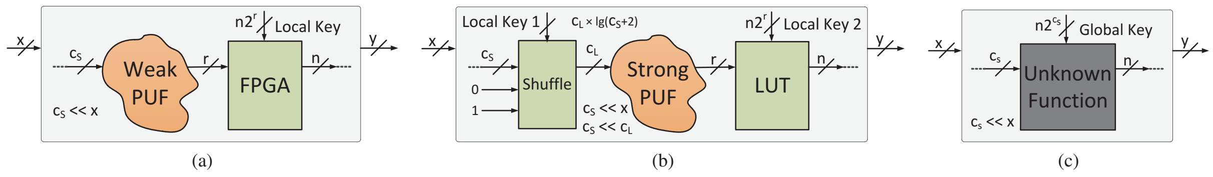


Fig. 1: GSAT attack on traditional IC-specific logic encryption schemes: (a) Weak PUF encryption with local key [15], (b) Strong PUF encryption with local keys [16], and (c) SAT-friendly model with global key

TABLE I: Symbols

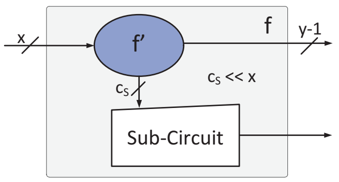
Symbol	Definition
x	Number of primary inputs
y	Number of primary outputs
c_S	Number of sub-circuit inputs
c_L	Number of sub-circuit inputs after augmentation
n	Number of sub-circuit outputs
S	Number of sub-circuit care set patterns
S_0	Number of sub-circuit care set patterns with output “0”
S_1	Number of sub-circuit care set patterns with output “1”
r	Number of PUF outputs
R	Number of PUF output patterns

designer must remove all PUF characterization channels from the ICs, leaving no physical channels to characterize the PUFs for the legally activated ICs on the market. Khaleghi and Rao [16] have adopted a shuffle block with a strong PUF and a multi-output Look Up Table (LUT) pair. In order to make the use of a strong PUF feasible, they limited the designer to just a subset of the input set while fixing most of the PUF inputs. However, as the size of the chosen sub-circuit grows, the complexity of the LUT also grows exponentially. Thus, they are still limited to using small sub-circuits.

Despite the fact that an IC-specific logic encryption scheme is dependent on local key inputs for each IC, it can be converted to a SAT-friendly model with a global key since the boundary of the replaced sub-circuit is easily identifiable in the encrypted circuit layout. Thus, we propose *GSAT*, a Global attack on non-uniform logic encryption schemes using the SAT model. In *GSAT*, we build a netlist with the unknown sub-circuit and then implement all the possible functions for the missing sub-circuit using a global key. Then, with the help of the original SAT-based attack [9] the global key can be efficiently revealed.

Fig. 1 shows the *GSAT* attack on traditional PUF-based encryption methods, [15], [16]. In summary, since the overhead of the FPGA/LUT block is large, these methods require choosing a small sub-circuit to be replaced with PUF and FPGA/LUT pairs. In this case, although the size of the global key is exponential to the size of the sub-circuit inputs, it has a polynomial relation to the size of the primary inputs, and thus GSAT is able to find the correct key in polynomial time.

It is worth noting that the global key size in the missing sub-circuit of the SAT-friendly model has a linear relation with the sub-circuit output size and an exponential relation with the sub-circuit input size. Thus, the drive parameter to increase the



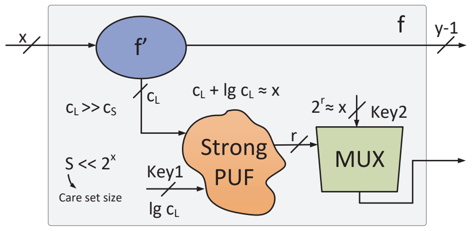


Fig. 2: IC-specific encryption (a) Original circuit (b) SPLEnD

complexity is the sub-circuit input size. As shown in Fig. 1c, **any missing sub-circuit with c_S inputs and n outputs can be modeled in *GSAT* using $n2^{c_S}$ key bits.**

B. SPLEnD Architecture

In order to address the existing challenges in IC-specific logic encryption, we propose *SPLEnD*, a Strong PUF-based Logic Encryption Design. In *SPLEnD*, we replace a small but critical single output sub-circuit with a strong PUF with a primary key input (i.e., $Key1$) and a MUX with a secondary key input (i.e., $Key2$) as shown in Fig. 2.

The input size of the PUF (i.e., $c_L + lgc_L$) is comparable to the size of primary inputs (i.e., x). However, only a small care set $S << 2^x$ is important and the remaining patterns never happen in the regular operation of the circuit. The output size of the PUF is chosen to have a logarithmic relation with the input size (i.e., $r \approx lgx$). Since the behavior of PUF is unique in each IC, **both the primary and secondary key inputs of the *SPLEnD* architecture are different for different ICs.** In other words, assigning a unique combination of $Key1$ and $Key2$ for each IC is essential to rebuilding the functionality of the missing sub-circuit. In this case, we do not even need to hide the key of each IC because it is not pluggable to the other ICs. This makes probing attacks inapplicable while putting the logic encryption research back on the right track.

In order to reduce the overhead of the encryption scheme, we suggest using a single output MUX instead of an FPGA [15] or multiple output LUT [16]. The output of the PUF is the controlling unit, while the input of the MUX is the secondary key input. Basically, the role of the MUX here is to decode the output patterns of the PUF into "0"s and "1"s. Since the size of the PUF output is chosen to have a logarithmic relation with the number of primary inputs, the secondary key input size has a linear relation with the number of primary inputs.

As we discussed in section II-A, the driving parameter to increase the encryption complexity is the sub-circuit input size, not its output size. Thus, we propose selecting a small sub-circuit that includes only one of the primary outputs. If the chosen sub-circuit has other fan-out signals to the rest of the circuit, the corresponding unencrypted circuit tree of those outputs should be duplicated outside of the sub-circuit boundary. Then, we use sequential don't-cares to increase the input size.

SPLEnD requires a large-size input to the PUF. However, when activating the circuit, the designer can only afford to test a small subset of the CRPs. The only way to have a large-sized input but to test only a small subset of them is to have the remaining subset be don't-care. If sequential don't-cares are employed, differentiating the care sets and the don't care sets is not possible for the attacker due to the state explosion problem. If the designer is aware of the existing sequential don't-cares in the circuit, he/she can employ them. Otherwise, we propose to add dummy registers and then utilize retiming and resynthesis [17] to insert the required don't-cares.

Also, in our attack model, we allow the attacker to access all testing capacities available to the designer. Thus, we cannot allow the attacker to know the care subset. Otherwise, the attacker may be able to figure out the function.

C. Security Discussion

In order to prevent the *GSAT* attack, we need to make the chosen sub-circuit have almost the same number of inputs as the primary inputs. In this case, the complexity of the attack on the sub-circuit will be the same as the attack on the whole circuit. Therefore, in *SPLEnD* we proposed choosing a small sub-circuit that includes only one of the primary outputs, and then, we used sequential don't-cares to make the input size large. In this case, **the *GSAT* attack on *SPLEnD* has the same complexity as the attack on the whole circuit.** This leads to an exponentially time-consuming attack with respect to the input size of the circuit.

D. Yield Discussion

The care set patterns in *SPLEnD* divide into only two groups: Group ONE with size S_1 and group ZERO with size S_0 in which they make the single output of the sub-circuit to be "1" and "0" respectively. Suppose the output encoding of size r and $R = 2^r$, the following formula holds for the probability that none of the members of group ONE collides with any member of group ZERO and vise versa.

$$P = \sum_{i=1}^{S_1} \frac{R!}{(R-i)! \times i!} \times \frac{f(i)}{R^{S_1}} \times (\frac{R-i}{R})^{S_0} \tag{1}$$

Where $f(i)$ is as follows:

$$f(i) = \begin{cases} i^{s_1} - \sum_{j=1}^{i-1} \frac{i!}{(i-j)! \times j!} \times f(j), & i > 1 \\ 1, & i = 1 \end{cases} \tag{2}$$

Now given S_1 and S_0, we can compute the minimum number of PUF outputs r based on the following inequality to achieve the desired yield Y:

$$P \geq Y \tag{3}$$

Based on the above probability formulas for the yield, we can see that **there is a linear relationship between the yield and the PUF minimum output size under a fixed care set size.** In addition, **if the care set size grows exponentially, only a polynomial increase is required in the minimum output size to keep the yield untouched.**

To further increase the yield, we can use a small subset of the PUF inputs as the primary key input proposed in Fig. 2. If a chip cannot distinguish the required care set patterns in one primary key value, it may still be able to distinguish them in the other. Since the PUF behavior is unique to each IC, the primary key input that specifies the input polarity should be chosen differently for different ICs.

III. Experimental Results

A. Security Analysis

To show the strength of the *GSAT* attack on traditional IC-specific logic encryption schemes [15], [16], we adopt combinational circuits of ISCAS'85 [18] and MCNC'91 [19]. For each benchmark, a random sub-circuit is chosen with a logarithmic number of inputs corresponding to the number of primary inputs and at most three outputs. The selected sub-circuit is then replaced with an LUT with associated key inputs. The results of the *GSAT* attack on these benchmarks are shown in Table II. As can be seen, the global key of all the benchmarks can be decrypted quickly.

In another experiment, we encrypt the same combinational circuits using our proposed *SPLEnD* architecture. For each benchmark a random sub-circuit that includes one of the primary outputs of the original circuit is chosen with a small number of inputs. The input size of the selected sub-circuit is then increased to be comparable to the primary input size adopting sequential don't-cares via dummy registers. Finally, the chosen sub-circuit is replaced with a strong PUF with primary key input and a MUX with secondary key input. The strong PUF is built based on the proposed XOR arbiter in [20]. Then, we model these encrypted benchmarks using a global key. However, after one day long running of the *GSAT* attack, only the small benchmarks (i.e., apex4, ex5, and ex1010) could be decrypted. The *GSAT* attack can successfully decrypt the small-size benchmarks because even learning the whole circuit is easy in these benchmarks. In other words, we can find out the whole function of the small benchmarks using brute-force checking. For example, ex5 can be completely learned using 256 brute-force queries.

B. Overhead Analysis

Table III reports the percentage of area and critical path increase in the encrypted benchmarks with *SPLEnD* in comparison with the original unencrypted benchmarks. As can be seen, the total overhead of our proposed solution is reasonable compared to the security gain. Moreover, the approach scales well by increasing the circuit size.

TABLE II: GSAT results on the encrypted benchmarks with traditional IC-specific logic encryption methods [15], [16]

Bench	#Pri. In	#Sub. In/Out	#Glo. Key In	CPU Time (s)
apex2	39	5/1	32	0.161
apex4	10	3/1	8	0.023
c432	36	5/3	96	0.318
c499	41	5/2	64	0.306
c880	60	6/3	192	1.724
c1355	41	5/1	32	0.571
c1908	33	5/2	64	0.298
c2670	233	8/1	256	6.428
c3540	50	6/1	64	1.051
c5315	178	8/2	512	5.785
c6288	32	5/1	32	0.369
c7552	207	8/1	256	2.951
dalu	75	6/3	192	1.320
des	256	8/1	256	1.075
ex5	8	3/1	8	0.023
ex1010	10	3/1	8	0.014
i4	192	8/1	256	0.847
i7	199	8/3	768	5.611
i8	133	7/1	128	0.461
i9	88	6/1	64	0.873
k2	46	6/2	128	0.604
seq	41	5/3	96	0.812

TABLE III: Percentage of area and critical path increase in SPLEnD architecture compared to the original benchmarks

Bench	#Pri. In	#Sub. In/Out	#Pri. + #Sec. Key In	Area Inc.	Cri. Path Inc.
apex2	39	3/1	5 + 32	9%	2%
apex4	10	2/1	4 + 16	<1%	<1%
c432	36	4/1	5 + 32	20%	19%
c499	41	6/1	5 + 32	14%	22%
c880	60	5/1	6 + 64	10%	6%
c1355	41	5/1	5 + 32	11%	3%
c1908	33	4/1	5 + 32	6%	<1%
c2670	233	8/1	8 + 256	16%	<1%
c3540	50	6/1	6 + 64	5%	<1%
c5315	178	6/1	7 + 128	12%	<1%
c6288	32	5/1	5 + 32	2%	<1%
c7552	207	8/1	7 + 128	7%	<1%
dalu	75	4/1	6 + 64	4%	5%
des	256	8/1	8 + 256	6%	5%
ex5	8	3/1	3 + 8	<1%	<1%
ex1010	10	3/1	4 + 16	<1%	<1%
i4	192	8/1	7 + 128	21%	12%
i7	199	6/1	7 + 128	15%	2%
i8	133	3/1	7 + 128	8%	<1%
i9	88	6/1	6 + 64	7%	<1%
k2	46	3/1	5 + 32	4%	<1%
seq	41	3/1	5 + 32	2%	<1%

IV. Conclusion

Existing logic encryption solutions have utilized a uniform key in their designs, making them susceptible to SAT-based and probing attacks. In this paper, we took a different approach to logic encryption to first suggest an efficient attack on existing IC-specific logic encryption methods to reveal the hidden global key (i.e., *GSAT*), and then propose a low-cost per chip protection scheme utilizing process variations (i.e., *SPLEnD*). Our experiments showed that traditional IC-specific encryption schemes are vulnerable to textitGSAT attack, while *SPLEnD* architecture not only effectively resists *GSAT*, but also scales well in terms of performance and area.

V. Acknowledgement

This work is supported by NSF under award No. 2131156.

References

[1] S. Trimberger, "Trusted design in FPGAs," In *ACM/IEEE Design Automation Conference (DAC)*, pp. 5-8, 2007.

[2] R. Torrance and D. James, "The state-of-the-art in semiconductor reverse engineering," In *ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 333-338, 2011.

[3] J. A. Roy, F. Koushanfar, and I. L. Markov, "EPIC: Ending piracy of integrated circuits," In *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 1069-1074, 2008.

[4] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Logic encryption: A fault analysis perspective," In *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 953-958, 2012.

[5] K. Juretus and I. Savidis, "Reduced overhead gate level logic encryption", In *ACM Great Lakes Symposium on VLSI (GLSVLSI)*, pp. 15-20, 2016.

[6] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC piracy using reconfigurable logic barriers," In *IEEE Design Test of Computers*, vol. 27, issue 1, pp. 66-75, 2010.

[7] A. Rezaei, Y. Shen and H. Zhou, "Rescuing logic encryption in post-SAT era by locking & obfuscation," In *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 13-18, 2020.

[8] H. Zhou, A. Rezaei and Y. Shen, "Resolving the trilemma in logic encryption," In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1-8, 2019.

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

[10] Y. Shen, Y. Li, S. Kong, A. Rezaei and H. Zhou, "SigAttack: New high-level SAT-based attack on logic encryptions," In *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 940-943, 2019.

[11] H. Wang, D. Forte, M. Tehranipoor, and Q. Shi, "Probing attacks on integrated circuits: challenges and research opportunities," In *IEEE Design Test*, vol. 34, no 5, pp. 63-71, 2017.

[12] S. Engels, M. Hoffmann, and C. Paar, "The end of logic locking? A critical view on the security of logic locking," In *Cryptology ePrint Archive, Report 2019/796*, 2019.

[13] M. T. Rahman, S. Tajik, M. S. Rahman, M. Tehranipoor, and N. Asadizanjani, "The key is left under the mat: On the inappropriate security assumption of logic locking schemes," In *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 262-272, 2020.

[14] U. Ruhrmair and D. E. Holcomb, "PUFs at a glance," In *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, article 347, 2014.

[15] J. B. Wendt and M. Potkonjak, "Hardware obfuscation using PUF-based logic," In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 270,271, 2014.

[16] S. Khaleghi and W. Rao, "Hardware obfuscation using strong PUFs," In *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 321-326, 2018.

[17] S. Malik, E. M. Sentovich, R. K. Brayton, and A. Sangiovanni-Vincentelli, "Retiming and resynthesis: Optimizing sequential networks with combinational techniques," In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, no. 1, pp. 74-84, 1991.

[18] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits and a target translator in Fortran," In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 677-692, 1985.

[19] S. Yang, "Logic synthesis and optimization benchmarks user guide version 3.0," In *Microelectronics Center of North Carolina (MCNC) International Workshop on Logic Synthesis*, 1991.

[20] C. Zhou, K. Parhi, and C. H. Kim, "Secure and reliable XOR arbiter PUF design: An experimental study based on 1 trillion challenge response pair measurements," In *Design Automation Conference (DAC)*, article 10, 2017.