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

† Northwestern University

Abstract

The fabless business model encounters new security challenges, including piracy and overproduction. While there are many logic encryption solutions to prevent unauthorized products from functioning, they 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.

Introduction

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 [1]. 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 employ a uniform key in their designs.

Problem: If an attacker uses SAT-based [2] or probing attacks [3] to decipher the uniform key on one chip, he/she can easily plug the key into other chips and make them functional.

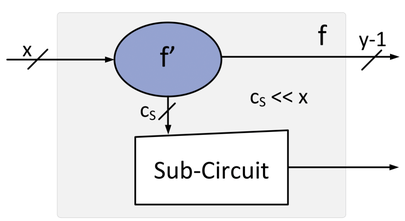
Solution: It is essential to adopt non-uniform and IC-specific key inputs to make SAT-based and probing attacks incompetent.

SPLEnD Architecture

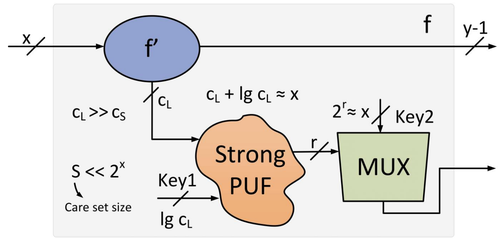
In SPLEnD, we replace a small but critical single output sub-circuit with a strong PUF with a primary key input (i.e., Key1) and a MUX with a secondary key input (i.e., Key2).

The input size of the PUF (i.e., c_L + lg c_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 ≈ lg x). 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. To reduce the overhead of the encryption scheme, we suggest using a single output MUX instead of an FPGA [4] or multiple output LUT [5]. 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.

We select 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.



Original circuit



SPLEnD

Yield Analysis

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.

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

Where *f(i)* is as follows:

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

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: $\mathbf{P \geq Y.}$

Experimental Results

GSAT results on traditional IC-specific logic encryption [4, 5]

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%

Area and critical path increase in SPLEnD

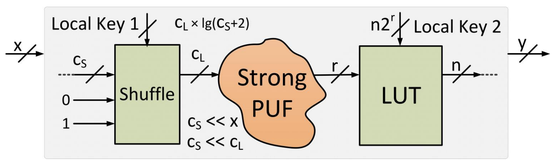
Bench	#Pri. In	#Sub. In/Out	#Glo. Key In	CPU Time (s)
apex2	39	5/1	32	0.161
apex4	10	3/1	8	0.023
c432	36	5/3	96	0.318
c499	41	5/2	64	0.306
c880	60	6/3	192	1.724
c1355	41	5/1	32	0.571
c1908	33	5/2	64	0.298
c2670	233	8/1	256	6.428
c3540	50	6/1	64	1.051
c5315	178	8/2	512	5.785
c6288	32	5/1	32	0.369
c7552	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

Note: GSAT on SPLEnD has the same complexity as the attack on the whole circuit.

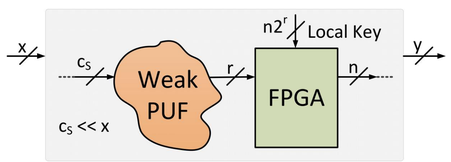
GSAT Attack

Even though IC-specific logic encryption 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. 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 [2] the global key can be efficiently revealed.

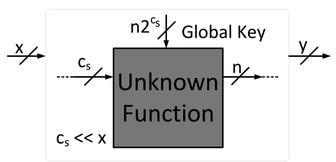
It is worth noting that the global key size in the missing sub-circuit of the SAT-friendly model has a linear relation with the sub-circuit output size and an exponential relation with the sub-circuit input size. Thus, the drive parameter to increase the complexity is the sub-circuit input size.



Weak PUF encryption with local key [4]



Strong PUF encryption with local keys [5]



SAT-friendly model with global key

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

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