SAT-based Bit-flipping Attack on Logic Encryptions

Yuanqi Shen, Amin Rezaei, and Hai Zhou
Northwestern University
yuanqishen2020@u.northwestern.edu, me@aminrezaei.com, haizhou@northwestern.edu

Abstract—**Logic encryption is a hardware security technique that uses extra key inputs to prevent unauthorized use of a circuit. With the discovery of the SAT-based attack, new encryption techniques such as SARLock and Anti-SAT are proposed, and further combined with traditional logic encryption techniques, to guarantee both high error rates and resilience to the SAT-based attack. In this paper, the SAT-based bit-flipping attack is presented. It first separates the two groups of keys via SAT-based bit-flippings, and then attacks the traditional encryption and the SAT-resilient encryption, by conventional SAT-based attack and by-passing attack, respectively. The experimental results show that the bit-flipping attack successfully returns a circuit with the correct functionality and significantly reduces the executing time compared with other advanced attacks.**

I. Introduction

For the sake of lower labor and manufacturing cost, many leading design houses have outsourced their fabrication to offshore foundries. However, it leads to many hardware security issues such as overproduction, piracy and counterfeiting [1], [3], [5], [6], [12]. To overcome these issues, logic encryption is proposed to add extra key gates into an IC design such that the circuit is only functional when key inputs are set correctly. Different logic encryption techniques are proposed, however, almost all of them [2], [4], [7]–[9] can be corrupted by the satisfiability (SAT) attack [11], which utilizes a SAT solver to prune out wrong keys efficiently.

To defeat the SAT attack, SAT proof blocks such as SARLock [15] and Anti-SAT [13] are introduced. However, the error rate is exponentially low even though their key values are wrong. Therefore, bypass attack [14] is proposed to fix a few wrong input-output pairs under a wrong key so that the circuit is still fully functional. Thus, a better encryption strategy is to combine SAT proof blocks with traditional logic encryption methods, so the combined encryption not only has high error rate when key values are wrong, but also can defeat the SAT attack.

However, this improved encryption is still vulnerable. In this paper, we have proposed a SAT-based logic decryption technique called bit-flipping attack. The bit-flipping attack counts how many distinguishing input patterns (DIPs) between two fixed key values to separate the traditional logic encryption key and the SAT proof block key, and returns a fully functional circuit with the help of a SAT solver. A model of bit-flipping attack that specifically targets SARLock and Anti-SAT is also introduced.

II. Related Works

The SAT attack solves a correct key of an encrypted circuit by using a small number of carefully selected input patterns and their corresponding outputs to prune out wrong keys. Assuming $C(X, K, Y)$ represents the conjunctive normal form (CNF) of a locked circuit with input X, key K, and output Y. The SAT attack iteratively finds the assignment of the CNF $C(X, K_1, Y_1) \wedge C(X, K_2, Y_2) \wedge (Y_1 \neq Y_2)$ until unsatisfiable. **When an assignment of X (i.e. X_i) is found, X_i is called a distinguishing input pattern (DIP).** X_i distinguishes the current assignment of K_1 and K_2 since at least one of them generates the wrong output. The correct output Y_i of X_i can be evaluated by an activated IC, and it is used to constrain K_1 and K_2 by adding $C(X_i, K_1, Y_i) \wedge C(X_i, K_2, Y_i)$ to the existing CNF. If there exists no input that can differentiate possible keys, the CNF is no longer satisfiable and the iteration stops. Therefore, a SAT solver can compute a correct key value which satisfies current constraints.

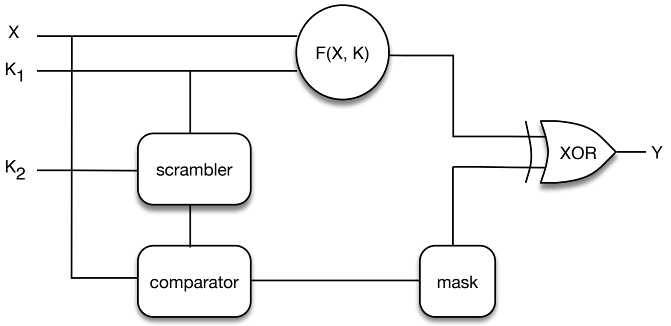

Fig. 1. The general design of SARLock.

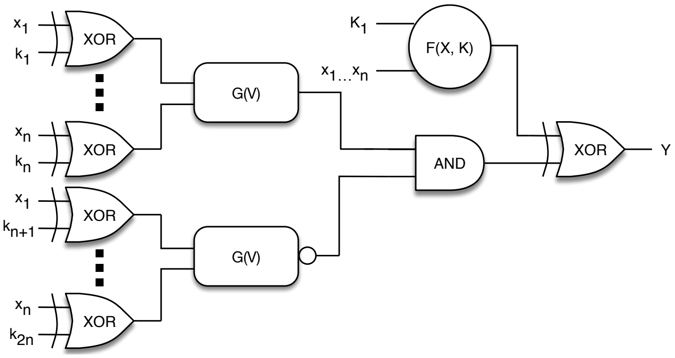

Fig. 2. The general design of Anti-SAT.

One method to defeat the SAT attack is to increase the total number of iterations to prune out all wrong keys. SARLock and Anti-SAT are designed in a way such that the total number of iterations for the SAT attack to reveal the correct key is exponentially large. The design of SARLock as shown in Figure 1 guarantees the output of an encrypted circuit will be flipped when input values are equal to the scramble (e.g. XOR) of the traditional logic encryption key K_1 and the SARLock key K_2, as long as K_1 and K_2 are not assigned correct key values. The mask ensures the output will not be flipped when key values are correct. On the other hand, Figure 2 shows a design that the Anti-SAT block is added on top of the traditional logic encryption technique. There are 2^n correct key combinations for Anti-SAT happening when $k_i = k_{n+i}$ for all $i \in 1...n$, and it can be proved that the number of iterations for the SAT attack to decipher the correct key is lower bounded by 2^n [13].

To defeat SARLock and Anti-SAT, bypass attack [14] is proposed to fix wrong input-output pairs by hardwiring under a wrong key. If an encryption method has very low error rate when the key is wrong, the bypass attack can quickly return a fully functional circuit. However, if the encryption is further combined with a traditional logic encryption technique, the bypass attack is not efficient since it will dramatically increase the overhead of a circuit.

III. Bit-flipping Attack

In this section, we introduce how bit-flipping attack could separate the traditional logic encryption key K_1 and the SAT proof block key K_2, and further decipher both correct K_1 and K_2. We take SARLock and Anti-SAT as examples of the SAT proof block, and illustrate how the bit-flipping attack can successfully decrypt them.

A. Key Separation

One major difficulty to attack the combination of traditional logic encryption and a SAT proof block is the key separation. Since a traditional logic encryption key K_1 is mixed with a SAT proof block key K_2, it is hard to decipher both K_1 and K_2 directly. However, if we could know the exact position of K_1 and K_2, we may divide and conquer.

We realize the difference of these two encryption techniques is the error rate when keys are wrong. Due to the design pattern of traditional logic encryption, a wrong key causes substantial wrong input-output pairs, and different wrong keys lead to different input-output pairs to be wrong. However, error rate of a SAT proof block is usually very small to defeat the SAT attack. For example, existing SAT proof block techniques such as SARLock and Anti-SAT have exponentially low error rate for wrong keys, and encrypted circuits embedded with different wrong keys have very few different input-output pairs. Therefore, to know if a bit belongs to K_1 or K_2, we can have two keys with a difference of only this bit, and count how many DIPs exist.

Based on the analysis, we propose bit-flipping attack shown in Algorithm 1. For circuits $C(X, K_A, Y_A) \wedge C(X, K_B, Y_B)$, we fix K_A as a random key value and flip one bit of K_A to have K_B. A SAT solver is used to find how many different DIPs exist so that Y_A is not equal to Y_B. If a bit in K_2 is flipped, only a few DIPs exist. However, if a bit in K_1 is flipped, more DIPs should exist. To ensure all bits in K_1 can be detected, we repeat this process for several runs, and the flipped bit is in K_1 as long as the number of DIPs is more than the threshold in one of runs.

Once we can carefully separate K_1 and K_2, we can fix K_2 as a random number and perform the SAT attack to decipher the correct K_1. The key containing the correct K_1 is guaranteed with exponentially low error rate. To obtain a fully functional circuit, the bypass attack [14] is adopted to fix a few wrong input-output pairs. We prepare two keys containing the correct K_1, and assign distinct values to K_2. By using a SAT solver to find DIPs between these two keys, we can evaluate correct outputs for these DIPs by an activated IC, and fix these outputs by hardwiring. As a result, we could consider K_2 as correct since all input-output pairs are correct now.

B. Security Analysis of SARLock and Anti-SAT

One potential issue of the bit-flipping attack is the trade off between the execution time and the accuracy of keys. Since we do not know the exact assignment of K_1 which could cause more DIPs after flipping, we have to randomly fixed key values for each iteration. The more iterations we tried, the higher possibility that we could successfully separate K_1 and K_2. However, more iterations lead to more execution time.

Algorithm 1 Bit-flipping Attack

```
Input: Encrypted circuit $C(X, K, Y)$, and activated circuit $eval$.
Output: Fully functional circuit $C_c$.
for $iter <$ fixed iterations do
    $K_A =$ a random key
    for each bit $b \in K_A$ do
        $K_B = K_A$ with bit $b$ flipped
        $i = 0$
        $F_0 = C(X, K_A, Y_A) \wedge C(X, K_B, Y_B)$
        while $sat[F_i \wedge (Y_A \neq Y_B)]$ do
            $X_i = sat\_assignment_X(F_i \wedge (Y_A \neq Y_B))$
            $F_{i+1} = F_i \wedge (X \neq X_i)$
            $i = i + 1$
            if $i >$ threshold then $b$ is in $K_1$, break
        end while
    end for
    $iter = iter + 1$
end for
$K_2 =$ All Key Bits $\setminus K_1$
$K_C, K_D \rightarrow$ Fix $K_2$ as a random number
$K_{k_1} = SAT\_attack(C, eval, \{K_C, K_D\})$
$C_c = bypass\_attack(C, eval, K_{k_1})$
```

Fortunately, if we know the structure of the SAT proof block, the bit-flipping attack can be further developed. In this subsection, we conduct the security analysis of SARLock and Anti-SAT, and propose bit-flipping attack targeting SARLock and Anti-SAT. The new bit-flipping attack is able to guarantee that when performing on SARLock or Anti-SAT, a key containing the correct K_1 can be solved. Therefore, the wrong K_2 can be easily fixed by the bypass attack.

First, we conduct the security analysis of the combination of a traditional logic encryption technique and SARLock. Assume K_A is a random key, and K_B is the key after a bit in K_A is flipped. K_1 denotes the key of the traditional logic encryption technique part, and K_2 denotes the key of SARLock part. We analyze how many DIPs exist between circuits embedded with K_A and K_B.

1) Assume K_A happens to be a correct key.
   a) Assume after flipping, K_B is another correct key. Then there is no DIP.
   b) Assume after flipping, K_B is not a correct key. If the bit we flipped is in K_2, then there is exactly one DIP, which is equal to the scramble of K_1 and K_2 of K_B. However, if the bit we flipped is in K_1, more DIPs are highly possible due to high error rate of traditional logic encryption techniques.
2) Assume K_A is not a correct key.
   a) Assume after flipping, K_B is a correct key. If the bit we flipped is in K_2, there is exactly one DIP, which is equal to the scramble of K_1 and K_2 of K_A. However, if the bit we flipped is in K_1, more DIPs are highly possible.

    b) Assume after flipping, K_B is not a correct key. If the bit we flipped is in K_2, there are exactly two DIPs, which are equal to the scramble of K_1 and K_2 of K_A, and the scramble of K_1 and K_2 of K_B. However, if the bit we flipped is in K_1, more DIPs are highly possible.

Theorem III.1. *There are at most two DIPs if a key bit in the SARLock part is flipped for benchmarks encrypted with the combination of a traditional logic encryption technique and SARLock.*

Similarly, we analyze benchmarks encrypted with the combination of a traditional logic encryption technique and Anti-SAT, and we use an AND gate as the G function. Initially, K_A is a random key, which is composed by a traditional key K_1 and a Anti-SAT key $k_1...k_{2n}$. K_B is the key after a bit in K_A is flipped.

1) Assume in K_A, $k_1...k_n$ are not equal to $k_{n+1}...k_{2n}$, which means the key of Anti-SAT in K_A is incorrect.
    a) If the bit we flipped is in $k_1...k_n$.
        i) Assume after flipping, $k_1...k_n$ are still not equal to $k_{n+1}...k_{2n}$ in K_B. Then there are two DIPs, which are equal to $\overline{k_1}....\overline{k_n}$ in K_A and $\overline{k_1}...\overline{k_n}$ in K_B.
        ii) Assume after flipping, $k_1...k_n$ are equal to $k_{n+1}...k_{2n}$ in K_B. Then there is only one DIP, which is $\overline{k_1}...\overline{k_n}$ in K_A.
    b) If the flipped bit is in $k_{n+1}...k_{2n}$.
        i) Assume after flipping, $k_1...k_n$ are still not equal to $k_{n+1}...k_{2n}$ in K_B. Since K_A and K_B have the same $k_1...k_n$, if a DIP exists it should equal to $\overline{k_1}...\overline{k_n}$. However, the DIP is not equal to $\overline{k_{n+1}}...\overline{k_{2n}}$ for both K_A and K_B based on our assumption. So there is no such DIP.
        ii) Assume after flipping, $k_1...k_n$ are equal to $k_{n+1}...k_{2n}$ in K_B. Then there is one DIP which is equal to $\overline{k_1}...\overline{k_n}$. The Anti-SAT block of K_A generates flipping signal one, and the Anti-SAT block of K_B generates flipping signal zero.
    c) If the bit we flipped is in the traditional key K_1. Then it is easy to claim that the number of DIPs is highly possible to be more than two.
2) Assume in K_A, $k_1...k_n$ are equal to $k_{n+1}...k_{2n}$, which means the key of Anti-SAT is already correct.
    a) If the bit we flipped is in $k_1...k_n$. There is only one DIP, which is $\overline{k_1}....\overline{k_n}$ of K_B.
    b) If the bit we flipped is in $k_{n+1}...k_{2n}$. There is only one DIP, which is $\overline{k_1}....\overline{k_n}$ of K_B.
    c) If the bit we flipped is in K_1. Then it is easy to claim the number of DIPs is highly possible to be more than two.

Theorem III.2. *There are at most two DIPs if a key bit in the Anti-SAT part is flipped for benchmarks encrypted with the combination of a traditional logic encryption technique and Anti-SAT.*

The security analysis of SARLock and Anti-SAT demonstrates the threshold for numbers of DIPs should be set to two when performing the bit-flipping attack. To guarantee that there are more than two DIPs when a bit in the traditional logic encryption key K_1 is flipped, we require a SAT solver to find such K_A and K_B so that three different DIPs exist in one iteration as shown in Algorithm 2. If there is such an assignment, the bit we flipped is guaranteed to be in K_1. Otherwise, we consider the bit we flipped is in K_2.

Algorithm 2 Bit-flipping Attack Targeting SARLock and Anti-SAT

```
Input: Encrypted circuit $C(X, K, Y)$, and activated circuit $eval$.
Output: Fully functional circuit $C_c$.
$F = C(X_1, K_A, Y_A) \wedge C(X_1, K_B, Y_B)$
    $\wedge C(X_2, K_A, Y_C) \wedge C(X_2, K_B, Y_D)$
    $\wedge C(X_3, K_A, Y_E) \wedge C(X_3, K_B, Y_F)$
for $each\ bit\ b \in K_A$ do
    if $sat[F \wedge (Y_A \neq Y_B) \wedge (Y_C \neq Y_D) \wedge (Y_E \neq Y_F) \wedge (X_1 \neq X_2) \wedge (X_2 \neq X_3) \wedge (X_1 \neq X_3) \wedge (K_A^b \neq K_B^b) \wedge (K_A^{bits \setminus b} = K_B^{bits \setminus b})]$ then $b\ is\ in\ K_1$
end for
$K_2 = All\ Key\ Bits \setminus K_1$
$K_A,\ K_B \rightarrow Fix\ K_2\ as\ a\ random\ number$
$K_{k_1} = SAT\_attack(C, eval, \{K_A, K_B\})$
$C_c = bypass\_attack(C, eval, K_{k_1})$
```

IV. Experimental Results

In this section we evaluate the performance of bit-flipping attacks. Original benchmarks are from the ISCAS'85 and the Microelectronics Center of North Carolina. We encrypt original benchmarks with [9] as a traditional logic encryption technique (i.e. RND), which randomly inserts XOR/XNOR gates into an original circuit. Then we further encrypt benchmarks with SARLock or Anti-SAT to evaluate if the bit-flipping attack and the bit-flipping attack targeting SARLock and Anti-SAT can successfully decrypt the correct traditional logic encryption key K_1.

For benchmarks encrypted with Anti-SAT as shown in Figure 2, we prepare two designs of G functions; from n-bit inputs, we randomly select n-1 bits and connect them to an AND gate, then randomly select another n-1 bits, flip each bit of them, and connect them to another AND gate. Then we connect the outputs of these two AND gates to an OR gate. Therefore, the G function has totally four inputs that lead to its output to be 1. We set both the threshold and the number of iterations to 10, and perform the bit-flipping attack to test if this random design can be decrypted. On the other hand, we use the most common design, an AND gate, as the G function to evaluate the bit-flipping attack targeting SARLock and Anti-SAT.

Table I shows the result of performing the bit-flipping attack on benchmarks encrypted with 5 and 10 percentages overload of RND + SARLock or RND + Anti-SAT. K1 in the table means if the correct RND key is decrypted. We can see that out of 68 benchmarks, there are 55 benchmarks that the correct K_1 can be successfully solved (80.9% accuracy). Meanwhile, the bit-flipping attack can be finished within a few minutes for all benchmarks.

We further evaluate the bit-flipping attack targeting SARLock and Anti-SAT. *The experimental result shows that the correct K_1 of all benchmarks can be successfully decrypted within reasonable time.* For comparison, we perform the SAT attack on benchmarks encrypted with RND + Anti-SAT, and Double DIP on benchmarks encrypted with RND + SARLock since Double DIP specifically targets SARLock. Figure 3 and 4 show all encrypted benchmarks can be decrypted quickly by the bit-flipping attack targeting SARLock and Anti-SAT. However, Double DIP and the SAT attack cannot solve the correct K_1 for most of benchmarks within our time limit (five hours), which is indicated by a dashed line. The reason is that multiple correct K_1 values may exist, therefore only one wrong SARLock key can be pruned for Double DIP as shown in [10]; the SAT attack cannot find the correct K_1 without taking exponential iterations to solve the Anti-SAT key K_2.

TABLE I
PERFORMING THE BIT-FLIPPING ATTACK ON RND + SARLOCK (SAR) OR RND + ANTI-SAT (AS). THE CORRECTNESS OF THE RND KEY AND EXECUTION TIME (S) ARE SHOWN IN THE TABLE.

	SAR(5%)		SAR(10%)		AS(5%)		AS(10%)	
ckt	K1	time	K1	time	K1	time	K1	time
apex2	yes	14.584	yes	22.292	yes	28.804	yes	42.332
c1355	yes	65.436	yes	85.584	yes	48.06	yes	74.08
c1908	yes	52.128	yes	53.348	yes	45.54	yes	68.064
c3540	yes	97.488	yes	216.6	yes	150.708	yes	235.632
c432	yes	13.192	yes	12.956	yes	19.324	yes	23.384
c499	yes	40.772	yes	41.88	yes	88.42	yes	101.436
c5315	yes	528.66	yes	699.132	yes	745.72	no	889.432
c880	yes	20.528	yes	27.624	yes	42.132	yes	53.012
dalu	yes	170.388	yes	329.312	yes	264.34	yes	402.836
ex1010	yes	573.276	no	1106.49	no	629.04	no	1257.68
ex5	yes	43.232	yes	65.532	no	40.504	no	76.132
i4	yes	46.76	yes	70.768	yes	149.196	yes	162.464
i7	yes	287.908	yes	289.356	yes	479.572	yes	537.716
i8	yes	343.848	no	495.28	yes	639.384	yes	760.96
i9	yes	76.736	yes	101.44	yes	152.348	yes	180.108
k2	no	118.676	no	200.016	no	183.624	no	283.196
seq	yes	269.552	no	526.496	yes	367.672	no	688.208

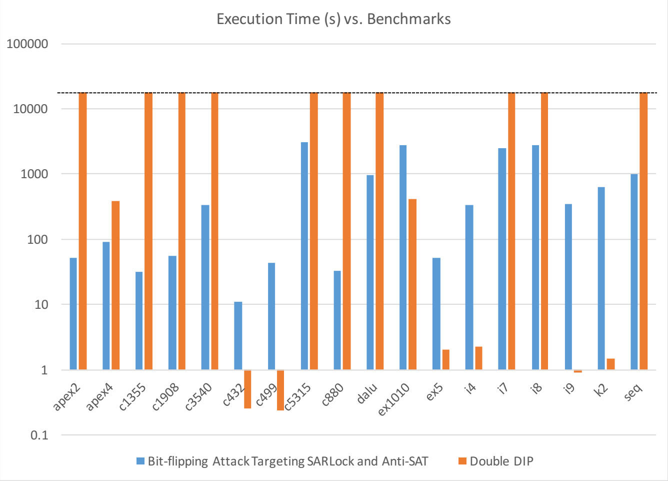

Fig. 3. Execution time of performing Double DIP and bit-flipping attack targeting SARLock and Anti-SAT on benchmarks encrypted with RND + SARLock (5% overhead).

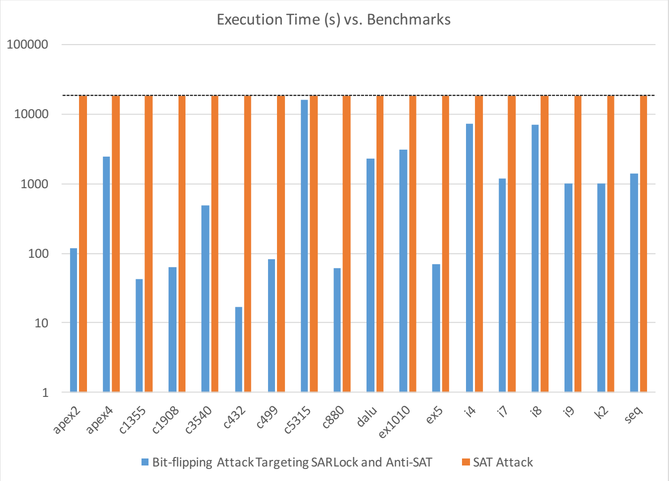

Fig. 4. Execution time of performing the SAT attack and bit-flipping attack targeting SARLock and Anti-SAT on benchmarks encrypted with RND + Anti-SAT (5% overhead).

V. CONCLUSION

In this paper, we propose a new SAT-based logic decryption technique called bit-flipping attack. The bit-flipping attack counts DIPs for two keys with hamming distance equal to one to separate a traditional logic encryption key K_1 and a SAT proof block key K_2, then fix K_2 as a random number, and use a SAT solver to decipher a correct K_1. Once a correct K_1 can be solved, the bypass attack can be applied to obtain a functional circuit. By carefully analyzing SARLock and Anti-SAT, bit-flipping attack targeting SARLock and Anti-SAT is also proposed, and the experiment shows that they efficiently decipher a correct K_1 of all benchmarks encrypted with RND + SARLock or RND + Anti-SAT.

ACKNOWLEDGMENT

This work is partially supported by NSF under CNS-1441695, CCF-1533656, and CNS-1651695.

REFERENCES

[1] M. Abramovici and P. Bradley. Integrated circuit security: new threats and solutions. In *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*, page 55. ACM, 2009.

[2] A. Baumgarten, A. Tyagi, and J. Zambreno. Preventing ic piracy using reconfigurable logic barriers. *IEEE Design and Test*, 27(1), 2010.

[3] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan. Hardware trojan attacks: threat analysis and countermeasures. *Proceedings of the IEEE*, 102(8):1229–1247, 2014.

[4] 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. In *IEEE International On-Line Testing Symposium*, 2014.

[5] United States. Defense Science Board. Task Force on High Performance Microchip Supply. *Defense science board task force on high performance microchip supply*. Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics, 2005.

[6] Michael Pecht and Sanjay Tiku. Bogus: electronic manufacturing and consumers confront a rising tide of counterfeit electronics. *IEEE spectrum*, 43(5):37–46, 2006.

[7] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri. Logic encryption: A fault analysis perspective. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 953–958. EDA Consortium, 2012.

[8] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri. Security analysis of logic obfuscation. In *Proc. of the Design Automation Conf.*, 2012.

[9] Jarrod A Roy, Farinaz Koushanfar, and Igor L Markov. Epic: Ending piracy of integrated circuits. In *Proceedings of the conference on Design, automation and test in Europe*, pages 1069–1074. ACM, 2008.

[10] Y. Shen and H. Zhou. Double dip: Re-evaluating security of logic encryption algorithms. In *Proc. ACM Great Lakes Symposium on VLSI*, 2017.

[11] P. Subramanyan, S. Ray, and S. Malik. Evaluating the security of logic encryption algorithms. In *Proc. IEEE International Symposium on Hardware Oriented Security and Trust*, 2015.

[12] J. Villasenor and M. Tehranipoor. Chop shop electronics. *IEEE Spectrum*, 50(10):41–45, 2013.

[13] Y. Xie and A. Srivastava. Mitigating SAT attack on logic locking. In *Conference on Cryptographic Hardware and Embedded Systems (CHES)*, 2016.

[14] 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 *Conference on Cryptographic Hardware and Embedded Systems*, 2017.

[15] M. Yasin, B. Mazumdar, J. J. V. Rajendra, and O. Sinanoglu. SARLock: SAT attack resistant logic locking. In *Proc. IEEE International Symposium on Hardware Oriented Security and Trust*, 2016.