Mitigating SAT Attack by Integrating Anti-SAT Block into a Locked Circuit

Priyanka(2017VLSI-09), Rahul Gupta(2017VLSI-10): *Students, ABV-IIITM, Gwalior*

1 ABSTRACT

Logic locking is a technique that has been proposed to protect outsourced IC designs from piracy and counterfeiting by untrusted foundries. A locked IC gives the correct functionality only when a correct key is provided. However, it's security is threatened by a new attack called SAT attack, which can decipher the correct key of most logic locking techniques within a few hours even for a large number of keys. This attack iteratively solves SAT formulas which progressively eliminate the incorrect keys till the circuit unlocked. In this project, Anti-SAT block [1] is used to enhance the security of existing logic locking techniques against the SAT attack. The number of SAT attack iterations to reveal the correct key in a circuit comprising an Anti-SAT block is an exponential function of the key-size therefore it makes the SAT attack computationally infeasible.

2 Introduction

Outsourced fabrication of integrated circuit(IC) enables IC design companies to access advanced technology at a low cost. It is cost- effective but the outsourced design faces various security threats since the offshore foundry might not be trustworthy. Logic locking is a technique that has been proposed to protect outsourced IC designs from piracy and counterfeiting by untrusted foundries. In this additional key-controlled logic gates (key-gates), keyinputs and an on-chip memory are inserted into an IC design to hide its original functionality. The key-gate can be implemented using XOR/XNOR gates as shown in Figure 1. A locked IC gives the correct functionality only when a correct key is provided.

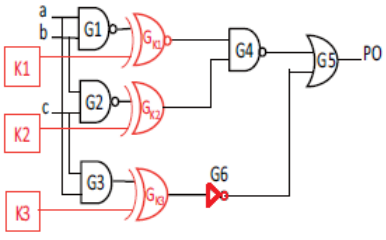

Figure 1. Encrypted circuit consist of XOR/XNOR key gates. The correct key K1,K2,K3 value is 101. This technique is vulnerable to SAT attack

- *Priyanka(2017VLSI-09), Rahul Gupta(2017VLSI-10), ABV-Indian Institute of Information Technology and Management Gwalior, India, 474015.*

3 Existing Problem

The security of logic locking is threatened by a new attack called SAT attack, which can decipher the correct key of most logic locking techniques within a few hours even for a large key-size. This attack iteratively solves SAT formulas to find a set of distinguishing I/O patterns which progressively eliminate the incorrect keys till the circuit is unlocked. SAT attack iteration table for logic encrypted circuit is shown in Figure 2 has Primary inputs a, b and c; Primary output Y and (k0, k1.....k7) are keyinput combinations. In this SAT attack iteratively eliminate wrong key combinations to find out correct key combination. Here only 4 iterations are sufficient to decipher the correct key combination, i.e. k5.

abc	Y	k0	k1	k2	k3	k4	k5	k6	k7	Incorrect keys detected
000	0	1	1	1	1	1	0	1	1	
001	0	1	1	1	1	1	0	1	1	
010	0	1	1	1	1	1	0	1	1	iter 3: k7
011	1	1	1	1	1	1	1	1	1	
100	0	1	1	1	1	1	0	1	0	iter 4: rest incorrect keys
101	1	1	0	1	1	1	1	1	1	
110	1	1	1	1	1	0	1	1	1	iter 1: k4
111	1	1	1	0	1	1	1	1	1	iter 2: k2

Figure 2. SAT attack iteration table for logic encryption

4 Solution to Existing Problem

In this project, Anti-SAT block is used to enhance the security of existing logic locking techniques against the SAT attack. The number of SAT attack iterations to reveal the correct key in a circuit comprising an Anti-SAT block is an exponential function of the key-size therefore it makes the SAT attack computationally infeasible.

4.1 Anti-SAT block design

The efficiency of SAT attack can be estimated by the total execution time: T $=\sum_{i=1}^{\lambda} t_i$, where λ is the total number of SAT attack iterations and t_i is the SAT solving time for i-th iteration. To mitigate SAT attack there is need to keep t_i and/or λ value large. Therefore we propose to insert a comparatively light-weight circuit block (referred to as

Anti-SAT block) which can efficiently increase the number of iterations λ so as to increase the total execution time T.

Figure 3(a) and Figure 3(b) represent two configurations of Anti-SAT block, referred to as type-0 Anti-SAT and type-1 Anti-SAT. They consist of two logic blocks g and $\overline{g}$, which share the same set of inputs $\vec{X} = (X_1 ... X_n)$. The functionalities of g and $\overline{g}$ are complementary to each other. Keygates which are inserted at the inputs of two logic blocks, denoted as $\vec{K_{l1}} = (K_1 ... K_n)$ and $\vec{K_{l2}} = (K_{n+1} ... K_{2n})$. Hence the key-size is 2n. The output of g and $\overline{g}$ are connected to an AND gate (for Fig. 3(a)) or an OR gate (for Fig. 3(b)) to give the final output Y .

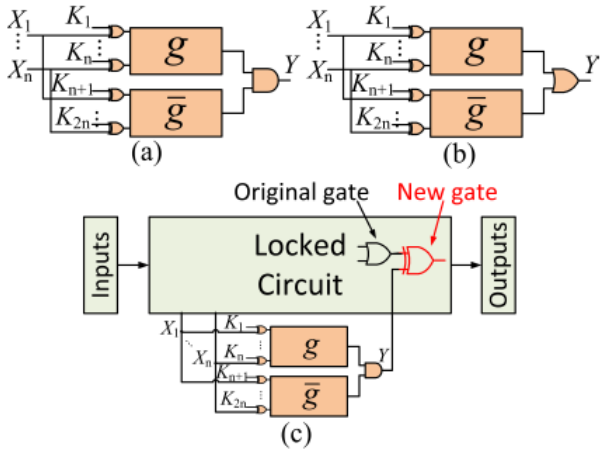

Figure 3. Anti-SAT block configuration: (a) Type-0 Anti-SAT: always outputs 0 if key values are correct; (b) Type-1 Anti-SAT: always outputs 1 if key values are correct. (c) Integrating the Type-0 Anti-SAT block into a circuit.

When key vector of Anti-SAT block is correctly set and output Y is constant, this block follows **Constant-output property**. In this Y always give output 0 for type-0 Anti-SAT (Fig. 3(a)) and output 1 for type-1 Anti-SAT (Fig. 3(b)). For wrong key, output Y can be either 1 or 0 depending on the inputs $\vec{X}$.

Here to validate the constant-output property, the **Correct keys** for the Anti-SAT block would be the ones that make type-0 Anti-SAT block output always 0 and type-1 Anti- SAT block output always 1. It happens when i-th key-bit from K_{l1} and i-th key-bit from K_{l2} have the same value, so the number of correct key combinations c = 2^n for both types of Anti-SAT blocks. Since the Anti-SAT block has 2n keys, the total number of wrong key combinations is $2^{2n} - 2^n$.

4.2 Concept

Terminology: Given a Boolean function g($\vec{L}$) with n inputs, assuming there exists p input vectors that make g equal to one ($1 \leq p \leq 2^n - 1$), we can classify the input vectors $\vec{L}$ into two groups L^T and L^F , where

$L^T = \{\vec{L} \,|\, g(\vec{L})|=1\}$, ($|L^T|$= p)

$L^F = \{\vec{L} \,|\, g(\vec{L})|=0\}$, ($|L^F| = 2^n - p$)

We denote L^T as the on-set of function g, L^F as the off-set of function g, and p as the on-set size.

Definition: (Distinguishing input/output (DIO) pair) For iteration i, the SAT attack find a correct I/O pair to identify a subset of wrong key combinations until none of these are left. An I/O pair at i-th iteration is a DIO, denoted as ($\vec{X^d}_i$, $\vec{Y^d}_i$), if it can identify a unique subset of wrong key combinations that cannot be identified by the previous i-1 DIOs.

Theorem: *If on-set size p of function g is sufficiently close to 1 or sufficiently close to $2^n - 1$, the number of iterations needed by the SAT attack to decipher the correct key is lower bounded by 2^n.*

- Wrong keys WK_i are the keys that are identified by a DIO ($\vec{X^d}_i$, $Y^d{}_i$) at i-th iteration, make a type-0 Anti-SAT output 1. For any given $\vec{X^d}_i$, $\vec{K_{l1}}$ can be selected in p different ways and $\vec{K_{l2}}$ in $2^n - p$ different ways. So the total number of ways to select wrong key is given by $p.(2^n - p)$.
- For any iteration i, and given $\vec{X^d}_i$, the maximum number of incorrect keys is $p.(2^n - p)$. This is the maximum number because there is possibility that some of these keys were identified in previous iterations. Hence the number of unique wrong keys identified in iteration i is upper-bounded by $p.(2^n - p)$.
- The total number of wrong keys are $2^{2n} - 2^n$. Thus, the number of iterations λ required for SAT attack to identify all the wrong keys is lower-bounded by $\lambda \geqslant \frac{2^{2n}-2^n}{p(2^n-p)}$.
- For p $\rightarrow$ 1 or p $\rightarrow 2^n - 1$, we can get maximum value of λ_l as follows:
  $\lambda \geqslant \lambda_l = \frac{2^{2n}-2^n}{p(2^n-p)} \rightarrow \frac{2^{2n}-2^n}{1(2^n-1)} = 2^n$.

5 Example

- We have taken a simple circuit as shown in Figure 4. It consist of 4 gates.
- Now Locked this test circuit with 2 XOR gates having key values K_a and K_b illustrated in Figure 5.

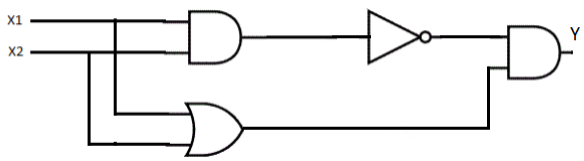

Figure 4. Test(original) circuit to be encrypted

- The number of iteration required to decipher these key values is 1 as given by SAT attack iteration tool.
- After this integrate Anti-SAT block into a locked circuit demonstrated in Figure 6.
- Total Key combinations for the Anti-SAT block = $2^{2n} = 2^4 = 16$, Where n=2. Total correct key combinations = 2^n=4. Therefore Wrong key combination is given by =$2^{2n} - 2^n$= 12.
- Here p=1,therefore the number of iterations are given by $\lambda \geqslant \frac{2^{2n}-2^n}{p(2^n-p)} = \frac{12}{3} = 4$. Also it is verified by SAT attack iteration tool.

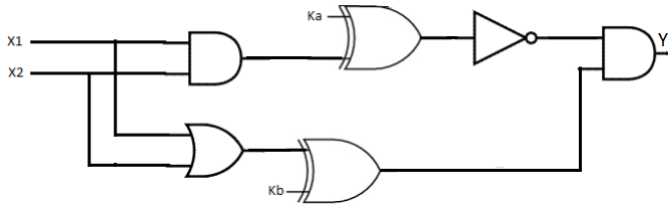

Figure 5. Locked circuit by using XOR Gates having key values K_a and K_b

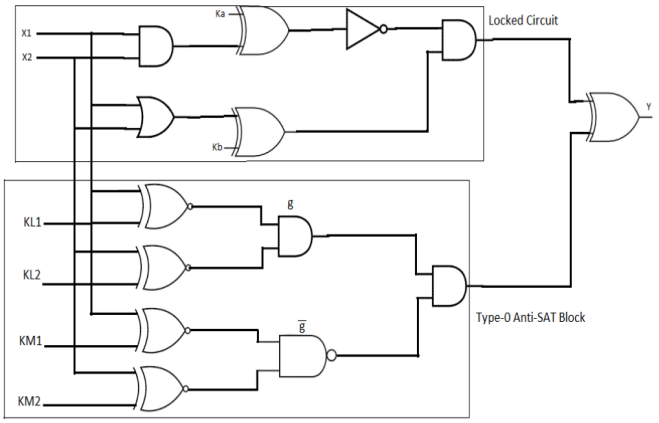

Figure 6. Logic encrypted circuit is integrated with Type-0 Anti-SAT block. K_a and K_b are the keys for locked circuit where as K_{l1}, K_{l2}, K_{M1} and K_{M2} are the keys for Anti-SAT block.

5.1 Distinguishing Input Pair(DIP)

Test circuit consists of 4 DIPs as shown below. It is observed that for each DIP, the wrong key combinations are given by WK_i=$p.(2^n - p)$. If we set the value of P=1, then it maximize the value of WK_i=$2^n - 1$ to increase the time complexity to obtain correct key.

1: If x={X_1=0, X_2=0}, the key values which set the AND gate to 1(here p=1) is KL={KL_1=0, KL_2=0}. The total wrong key combinations for this case are $p.(2^n - p)$=3.

KL_1	KL_2	KM_1	KM_2
0	0	0	1
0	0	1	0
0	0	1	1

2: If x={X_1=0, X_2=1}, the key values which set the AND gate to 1(here p=1) is KL={KL_1=0, KL_2=1}. The total wrong key combinations for this case are $p.(2^n - p)$=3.

KL_1	KL_2	KM_1	KM_2
0	1	0	0
0	1	1	0
0	1	1	1

3: If x={X_1=1, X_2=0}, the key values which set the AND gate to 1(here p=1) is KL={KL_1=1, KL_2=0}. The total wrong key combinations for this case are $p.(2^n - p)$=3.

KL_1	KL_2	KM_1	KM_2
1	0	0	0
1	0	0	1
1	0	1	1

4: If x={X_1=1, X_2=1}, the key values which set the AND gate to 1(here p=1) is KL={KL_1=1, KL_2=1}. The total wrong key combinations for this case are $p.(2^n - p)$=3.

KL_1	KL_2	KM_1	KM_2
1	1	0	0
1	1	0	1
1	1	0	1

6 Functional and Structural Obfuscation

An attacker can find the complementary pair of signal outputs of g and $\overline{g}$ while simulating the circuit. It leads to identification and removal of Anti-SAT block. So there is need to obfuscate the presence of Anti-SAT block in the circuit. It can be done by inserting Key gates at the inputs of Anti-SAT block to break complementary relations between signals, this provides functional obfuscation to the circuit. Structural obfuscation can be embedded by inserting MUX-based logic encryption to increase the inter-connectivity between the logic encrypted circuit and Anti-SAT block. A logic encrypted circuit with functional and structural

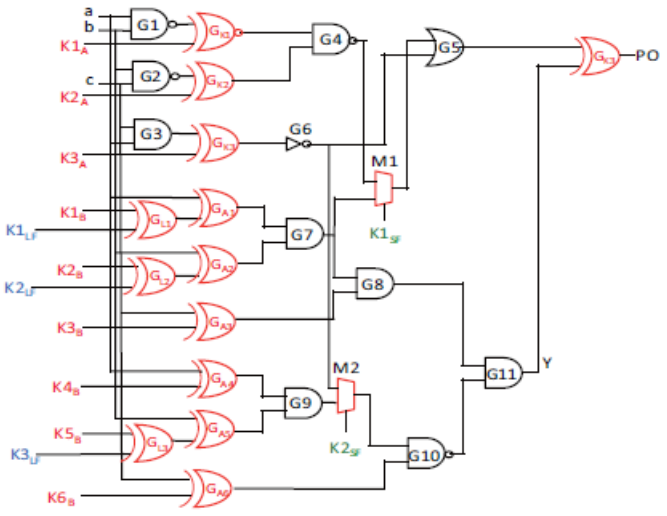

Figure 7. Functional and structural obfuscation between Anti-SAT and logic encrypted circuit of Figure 1(b). $K1_A, \ldots, K3_A$ are the key inputs to the logic encrypted circuit, $K1_B, \ldots, K6_B$ are the key inputs to Anti-SAT circuit, $K1_{LF}, \ldots, K3_{LF}$ (shown in blue color) are the key inputs for functional obfuscation, and $K1_{SF}$, $K2_{SF}$ (shown in green color) are the key inputs for structural obfuscation. M_1 and M_2 are used for MUX-based logic encryption.

Analysis	Logic Locking	Locking with Anti-SAT block	Locking with Structural Obfuscation
Iterations	1	7	9
Time(s)	0.008	0.024	0.048
Correct Key	001	001111111	00110110101

Table 1
Analysis of c17 circuit for different encryption techniques

ASL(Key=3+6)	P=1	P=5	P=$2^n-1=7$
Type'0' ASB(Iterations)	7	6	7
Type'0' ASB(Keys)	001111111	001011011	001101101
Type'0' ASB(Time(s))	0.02	0.016	0.024
Type'1' ASB(Iterations)	7	6	7
Type'1' ASB(Keys)	001101101	001111111	001001001
Type'1' ASB(Time(s))	0.024	0.02	0.024

Table 2
Analysis of c17 circuit for different values of P

obfuscation is illustrated in Figure 7. It hides the presence of Anti-SAT block in logic encrypted circuit.

7 Flow of Work

1. Take any benchmark circuit.
2. Apply encryption by logic locking technique. In this various XOR/XNOR gates inserted at different places in circuit along with key inputs.
3. Design Anti-SAT block and integrate it with encrypted circuit.
4. Take benchmark circuit file and encrypted circuit file as input in SAT attack iteration tool and calculate the number of iterations.
5. Take benchmark circuit file and Anti-SAT block encrypted circuit file as input in SAT attack iteration tool and calculate the number of iterations.
6. Now compare the results obtained from step 4 and 5. The number of iterations in step 5 should be more than step 4 for the effectiveness of Anti-SAT block.
7. The number of iterations in Anti-SAT block encrypted circuit is given by $\lambda \geqslant \frac{2^{2n}-2^n}{p(2^n-p)}$.

8 Analysis

8.1 Analysis of C17 Benchmark Circuit

- Table 1 shows analysis of c17 benchmark circuit with various encryption in terms of number of iterations, time to obtained correct key and correct key combination. It shows time and iterations increases as locked circuit is integrated with Anti-SAT block.
- Table 2 shows the analysis of c17 benchmark circuit for different values of P in Type'0' and Type'1' Anti-SAT block in terms of iterations, keys and time. From this table it can be concluded that number of iterations are maximum, when the value of P is either equal to 1 or equal to $2^n-1=7$.
- Table 3 shows analysis of c17 benchmark circuit with various encryption in terms of number of LUTs consumption and Power dissipation in watt. It is obtained by implementing design on Xilinx Vivado.

Analysis	c17 benchmark circuit	Logic Locking	Locking with Anti-SAT block
LUTs	1	3	8
Power(W)	0.761	0.951	1.293

Table 3
Analysis of c17 circuit in terms of LUTs and Power consumption

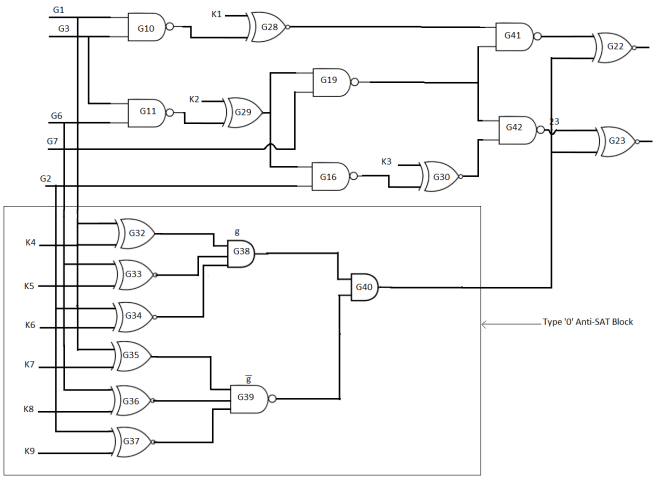

Figure 8. c17 logic encrypted circuit with Anti-SAT block

8.2 Analysis of C432 Benchmark Circuit

- Table 4 shows analysis of c432 benchmark circuit with various encryption in terms of number of iterations, time to obtained correct key and correct key combination. It shows time and iterations increases as locked circuit is integrated with Anti-SAT block. In Anti-SAT block(with key=10+36), SAT attack iteration tool is unable apply SAT attack as it is computationally infeasible for large number of iterations.
- Table 5 shows the analysis of c432 benchmark circuit for different values of P in Type'0' and Type'1' Anti-SAT block in terms of iterations and time. From this table it can be concluded that number of iterations are maximum, when the value of P is either equal to 1 or equal to $2^n-1=511$. For other values of P it gives less number of iterations. That's why Anti-SAT block is designed with P=1 or 2^n-1.
- Table 6 shows analysis of c432 benchmark circuit with various encryption in terms of number of LUTs consumption and Power dissipation in watt. It is obtained by implementing design on Xilinx Vivado.

Analysis	Logic Locking (key=10)	ASL (Key=10+18)	ASL (Key=10+36)
Iterations	5	512	262143
Time	0.076	9.024	Time out
Correct key	1100011000	1100011000 111100110111100110	Not found

Table 4
Analysis of c432 circuit with different encryption techniques

ASL(Key=10+18)	P=1	P=49	P=343	P=$2^n - 1 = 511$
Type'0' ASB(Iterations)	512	191	43	511
Type'0' ASB(Time(s))	9.024	2.322	0.364	9.016
Type'1' ASB(Iterations)	511	185	47	512
Type'1' ASB(Time(s))	8.468	1.86	0.422	9.028

Table 5
Analysis of c432 circuit for different values of P

Analysis	c432 benchmark circuit	Logic Locking	Locking with Anti-SAT block
LUTs	62	62	59
Power(W)	4.236	4.453	4.692

Table 6
Analysis of c432 circuit in terms of LUTs and Power consumption

8.3 Analysis of S27 Benchmark Circuit

- Table 7 shows analysis of S27 benchmark circuit with various encryption in terms of number of iterations, time to obtained correct key and correct key combination. It shows time and iterations increases as locked circuit is integrated with Anti-SAT block.
- Table 8 shows the analysis of S27 benchmark circuit for different values of P in Type'0' and Type'1' Anti-SAT block in terms of iterations, keys and time. From this table it can be concluded that number of iterations are maximum, when the value of P is either equal to 1 or equal to $2^n - 1 = 7$. For P=5, the number of iterations are less than 7.

8.4 Implemented Design

Implemented design of c17 benchmark circuit with various encrypted circuits are shown in Figure 9,10,11 and 12.

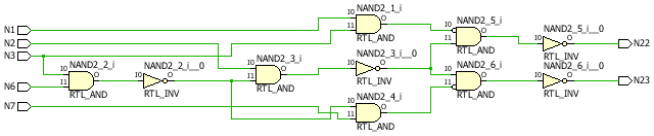

Figure 9. Original c17 benchmark circuit

Analysis	Logic Locking	Locking with Anti-SAT block
Iterations	2	7
Time(s)	0.008	0.024
Correct Key	000	000100100

Table 7
Analysis of s27 circuit for different encryption techniques

ASL(Key=3+6)	P=1	P=5	P=$2^n - 1 = 7$
Type'0' ASB(Iterations)	7	4	7
Type'0' ASB(Keys)	000100100	000111111	000111111
Type'0' ASB(Time(s))	0.024	0.016	0.024
Type'1' ASB(Iterations)	7	5	7
Type'1' ASB(Keys)	000111111	000111111	000001001
Type'1' ASB(Time(s))	0.024	0.012	0.024

Table 8
Analysis of s27 circuit for different values of P

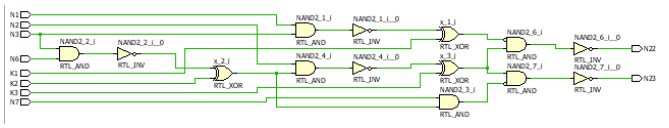

Figure 10. c17 benchmark circuit with logic locking

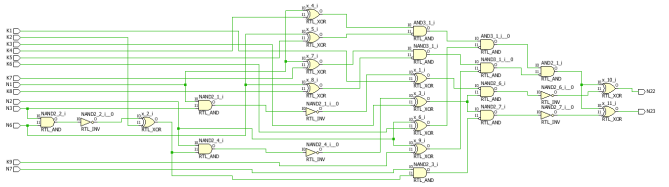

Figure 11. c17 logic encrypted circuit with Anti-SAT block

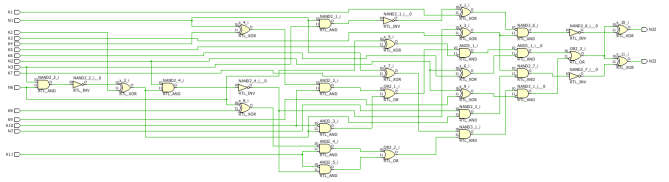

Figure 12. c17 logic encrypted circuit with Anti-SAT block and structural obfuscation

9 Conclusion

In this project, a circuit block called Anti-SAT is used to mitigate the SAT attack on logic locking. It is proved that the iterations required by the SAT attack to reveal the correct key in the Anti-SAT block is an exponential function of the keysize in the Anti-SAT block. In other words, The Anti-SAT block is integrated to a locked circuit to increase its immune to the SAT attack. To protect the Anti-SAT block from removal attacks such as the SPS attack and the partitioning based attack, functional and structural obfuscation technique is also proposed. We have implemented Anti-SAT block in Xilinx Vivado and calculate the number of iterations by SAT solver tool that run on ubuntu. In future we will work on the technique that completely remove SAT attack.

References

[1] Y. Xie and A. Srivastava, "Anti-sat: Mitigating sat attack on logic locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018.

[2] M. Yasin, B. Mazumdar, O. Sinanoglu, and J. Rajendran, "Security analysis of anti-sat," in *Design Automation Conference (ASP-DAC), 2017 22nd Asia and South Pacific*. IEEE, 2017, pp. 342–347.