# On SAT-Based Attacks On Encrypted Sequential Logic Circuits

Yasaswy Kasarabada, Suyuan Chen and Ranga Vemuri
*Digital Design Environments Laboratory*
*University of Cincinnati*, Cincinnati, Ohio USA
Email: kasarayv@mail.uc.edu, chens6@mail.uc.edu and ranga.vemuri@uc.edu

**Abstract**— Logic encryption has emerged as a solution to the hardware intellectual property (IP) protection problem. In recent years, many attack methods have been proposed to counter the protection offered by logic encryption. Most state-of-the-art logic encryption schemes have been shown to be susceptible to one or more of these attack methods. Furthermore, defense methods on sequential circuits assume that a Boolean satisfiability (SAT) based attack can be applied to a sequential circuit only in the presence of a design-for-testability (DFT) architecture. In this paper, we examine the effectiveness of applying the SAT-based attack on logic encrypted sequential circuits lacking a scan architecture. We evaluate the effect of the presence of flip-flop chains in the circuit on the sequential SAT attack time. Furthermore, we analyze the evaluation results and propose an enhancement to the sequential SAT attack.

**Keywords**—Logic encryption, SAT, sequential circuits.

## I. Introduction

Due to lower costs in accessing advanced semiconductor technology, the fabrication process of the integrated circuit (IC) is being outsourced to external foundries [1, 2]. This leads to the introduction of additional vulnerabilities such as design counterfeiting, IC overproduction, and IP piracy [3, 4]. Logic encryption has emerged as a potential solution for preventing IC reverse engineering and counterfeiting [5–7]. Under logic encryption, a circuit is modified to include an additional set of inputs called key inputs. The circuit operates correctly only if the correct value is applied to the key inputs; otherwise the input-output relationship of the circuit is corrupted. The correct key value is stored in a tamper-proof memory placed on the chip, thus making it inaccessible to the attacker. Recently, it was shown that most known encryption schemes were susceptible to Boolean satisfiability (SAT) based attacks [8–11]. Using a set of distinguishing input patterns (DIPs), these attacks iteratively eliminate incorrect keys to obtain the correct key value. To mitigate the effect of these SAT-based attacks, many SAT-resistant countermeasures have also been proposed [12–18]. However, most of these SAT-resistant schemes have been shown to be susceptible to removal attacks [19] and/or bypass attacks [20]. Moreover, logic encryption/decryption schemes for sequential circuits [21, 22] assume that sequential circuits can be attacked using the SAT-based methods only if a DFT architecture is present in the design. We evaluate the validity of this claim using experimental analysis. The contributions of this paper are as follows:

- We present an attack method on sequential circuits without scan architecture, which uses SAT to decipher the required key value in a reasonable amount of time.
- We examine the effect of the presence of flip-flop chains on the attack time using a new light-weight logic encryption scheme. Attack resiliency is analyzed by evaluating the impact of key size and flip-flop chain length on attack time (unroll count) using ISCAS'89 and ITC'99 benchmarks.
- We analyze the limitations of the proposed encryption scheme to suggest an enhancement to the sequential SAT attack which aims to save attack time.

## II. Background

The SAT attack proposed in [8], is a powerful attack method that uses distinguishing input patterns (DIPs) to iteratively eliminate incorrect keys and obtain the *equivalence class of correct keys*. The set of equivalence class of correct keys contains all keys that do not corrupt the input-output relation of the encrypted circuit for all input vectors. Therefore, any assignment using a key belonging to the obtained equivalence class of correct keys will produce the required IC. The attack methodology uses a miter type circuit to generate a Quantified Boolean Formula (QBF) in each iteration that is fed to the SAT solver to obtain the DIP. The DIP is used to generate the correct output vector using an activated IC obtained from the market. The DIP and the correct output vector are added to the QBF as additional constraints to obtain the next DIP. This process is continued till no such DIPs can be obtained. At this point, any key value that satisfies the final QBF can be used as the correct key assignment.

An interesting attack method to reverse engineer camouflaged sequential circuits without scan access using a model checking tool has been proposed [21]. Application of this attack method to decryption of logic encrypted sequential circuits without DFT architecture suffers from a couple of shortcomings. Due to the inability of most model checkers to efficiently solve the state space explosion problem, the attack method does not scale well for large benchmark circuits (with large number of flip-flops). In addition, the paper's most successful termination condition, Unique Completion (UC), checks to see if there is only one remaining completion that agrees with the black-box circuit. This criteria will not be satisfied in the presence of more than one correct key, which makes UC an unsuccessful termination condition while decrypting sequential circuits with equivalent correct keys. Due to these limitations, a novel decryption technique for logic encrypted sequential circuits without DFT architecture is needed.

Various defense methods for sequential circuits against the SAT attack have been proposed [22–24]. An interesting solution [22] proposes to encrypt the outputs of the D-type flip-flops using key-controlled MUXes with $Q$ and $\overline{Q}$ pins acting as the inputs. The authors claim to be able to restrict the controllability and observability of internal scan cells, thus defeating the SAT attack which is claimed to require scan architecture to control/observe internal scan cell values. In the next section, we show that an unroll-and-SAT attack method can be used to defeat this type of encryption scheme even without scan access to internal states.

## III. Sequential SAT Attack

The SAT attack [8] can be applied directly to sequential circuits if the internal scan-cells are accessible through scan architecture. The attack assumes that for each input vector $\vec{X}$ applied to the sequential circuit, any required value can be set to all flip-flops connected to the scan chain. Therefore, using the scan architecture, any sequential circuit can be considered to be equivalent to a combinational circuit $C(\vec{X}, \vec{K}, \vec{Y})$. Sequential defense methods argue that by corrupting the scan chain for incorrect keys, this equivalence can be broken, thus defeating the SAT attack. However, we show that a different method can be used to attack encrypted circuits that either corrupt the scan chains or do not incorporate a scan chain.

### *A.* Attack Methodology

Consider an encrypted sequential system $S_K(V, E, \vec{K})$ with node set $V$, edge set $E$, and key vector $\vec{K} \in \{0, 1\}^k$ such that $|V| = 2^l$, $|E| = 2^{l+m}$, and $V_0$ ($\in V$) is the initial state of the system, where $l$ is the number of flip-flops in the netlist and $m$ is the number of primary inputs. We assume that the $l$ flip-flops are either connected using corrupted scan chains or are unconnected, thus making the internal state uncontrollable and unobservable. An attacker will be able to set the internal state of the system to a required value using only the primary inputs ($\vec{X}$ and $\vec{K}$). In addition, we assume the presence of set-reset logic for each flip-flop which gives the attacker the ability to set the internal state to the initial start state $V_0$. A typical attack model is assumed where the attacker has access to a copy of the encrypted netlist and black-box access to the activated IC obtained from the market.

The attacker begins by obtaining cycle constraint information regarding any possible combinational cycles introduced by the encryption scheme using methods described in [25]. After determining the cycle constraints, the attacker starts by unrolling the encrypted netlist for $unr(\geq 2)$ clock cycles to obtain a combinational copy of the netlist, $S_K^{unr}$. This combinational netlist, $S_K^{unr}$ and the obtained cycle constraints, $CC$ are used by a modified combinational SAT attack tool to obtain a key assignment $\vec{K}_C$. It is guaranteed that the unrolled copy of $S$ with key assignment $\vec{K}_C$ agrees with $eval$ for all input-output combinations. However, it is possible that the sequential circuit $S_K$ with key assignment $\vec{K}_C$, $S_K(V, E, \vec{K}_C)$ may not agree with $eval$ for one or more input-output combination. Therefore, to test if the obtained key assignment is correct for all input-output combinations on $S_K$, the attacker can perform an at-speed testing by implementing $S_K$ on an FPGA device. Using a sufficiently large number of input vectors applied to both $S_K$ and $eval$, the results from both sets of primary outputs can be compared to eliminate incorrect key values. If $\vec{K}_C$ for a certain value of $unr$ results in a disagreeing output observation between $S_K$ and $eval$, $unr$ is incremented and the attack is continued. The attack stops when $S_K$ and $eval$ agree with each other for the obtained key $\vec{K}_C$. Algorithm 1 highlights the described attack method.

**Algorithm 1** Sequential SAT Attack Algorithm

```
Inputs: $S_K(V, E, \vec{K})$ and $eval$.
Outputs: $\vec{K}_C$.
$unr := 2$
$CC := find\_cycle\_constraints(S_K)$
do
    $S_K^{unr} := unroll(S_K, unr)$
    $\vec{K}_C := SAT_{combinational}(S_K^{unr}, CC, eval)$
    $unr := unr + 1$
while $S_K(V, E, \vec{K}_C) \not\equiv eval$
return $\vec{K}_C$
```

### *B.* Sequential SAT Attack Application

For the purpose of experimentation, the attack method is slightly modified to take an optimistic view on the attacker's computational ability. We assume that the attacker can easily verify up to 512 keys for each set of unrolled encrypted circuits and activated ICs. To model this for the experimentation process, we ask the SAT tool to return 512 alternate keys for each unroll iteration. It is important to note that the correct key for $S_K(V, E, \vec{K})$ will also be the correct key for $S_K^{unr}$ for all values of $unr$. Therefore, if less than 512 keys are obtained for a particular value of $unr$, the correct key is guaranteed to be present in the obtained set of keys and hence the attack exits returning the correct key. Whenever this condition occurs during experimentation, we indicate it in the results by putting an asterisk (*) next to the unroll count. If more than 512 keys are returned by the SAT solver, we first check if the correct key is present in the obtained 512 keys. If found, the attack tool exits returning the correct key. If the exact correct key is not found, 32 keys are randomly chosen and tested for equivalence. Equivalence testing is performed on unrolled copies of encrypted and original circuits using the $unr$ value of the current iteration. In addition, for the purpose of equivalence checking, the internal states are made controllable/observable. If all 32 keys fail to be evaluated as equivalent keys, the attack continues with an incremented value of $unr$; else the attack exits returning an equivalent key. In our results, equivalent key outcomes are marked with an exclamation mark (!). The maximum al-

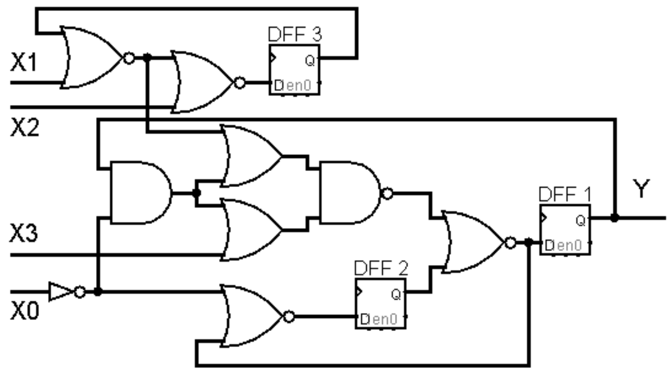


Fig. 1. Modified s27 benchmark

**Algorithm 2** Chain-based Encryption

```
Inputs: Sequential netlist $S(V, E)$
Outputs: Encrypted netlist $S_K(V, E, \vec{K})$
$\tilde{S}(X, Y, L) := simplify\_netlist(S)$
$D_{chains} := find\_all\_dff\_chains(\tilde{S})$
for $L_i \in L_S$ do
    $L_i^j := distance(L_i, Y_j) \; \forall \; Y_j \in Y$
    $L_i^{min} := min(L_i^j)$
end for
$L_{ranking} :=$ order $L_S$, maximize $L_i^{min}$
$L_{key\_size} := select\_nodes(L_{ranking}, key\_size)$
$S_K(V, E, \vec{K}) := encrypt(S, L_{key\_size}, \vec{K})$
```

lowed value of *unr* is set to be 30. If the algorithm reaches this limit without success or takes more than 24 hours to complete the attack on a single unrolled copy, the benchmark is considered to be SAT-resistant. This situation is represented by a hyphen (-) in the results. Section V details the results of running the sequential SAT attack on various benchmarks and encryption schemes.

## IV. Chain-based Encryption

In this section we discuss the limitations of the sequential SAT attack which can assist in the design of an intuitive encryption scheme capable of thwarting the SAT attack and/or making it infeasible.

### *A.* Analysis of Sequential SAT Attack

The sequential SAT attack unrolls a sequential circuit $S_K(V, E, \vec{K})$ to an equivalent combinational circuit in order to decrypt using the combinational SAT attack. This process replaces the $l$ flip-flops in each unrolled copy with buffer wires connected to the corresponding flip-flop inputs from the previous copy, with the exception of the first copy for which the buffer wires are connected to the set-reset logic. So, if a single flip-flop is present in the sequential circuit, then unrolling the circuit for two copies will yield an accurate combinational representation of the circuit. Similarly, if two flip-flops are connected in series such that any fan-out from one flip-flop is contained in the input cone of dependency of the other flip-flop, the sequential circuit will need to be unrolled for at least three clock cycles to obtain an accurate combinational representation.

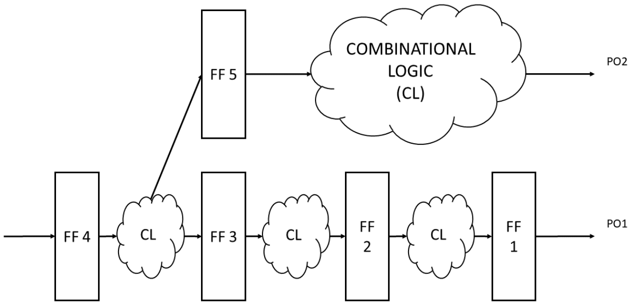


Fig. 2. Effect of sneak paths

For example, consider the sequential circuit shown in Figure 1, a modified version of the s27 benchmark from the ISCAS'89 suite. If this sequential circuit is unrolled only twice, then the effect of input X2 on the output Y would not be observable, thus making the twice unrolled combinational circuit an inaccurate representation of the sequential circuit. This can cause the combinational SAT attack on $S_K^2$ to return an inaccurate key value if a key gate is inserted at the input of DFF3 (similar argument can be made for DFF2). To get an accurate representation, the attack method will need to unroll the sequential circuit for at least three clock cycles so that the effect of X2 can be observable on output Y. Note that three is a lower bound on the number of unrolls required to obtain an accurate combinational representation; the structure of the entailing FSM dictates the upper bound. A more detailed discussion of the number of unrolls required for obtaining the correct key is presented in Section V.

### *B.* Choosing Encryption Points

As seen from the previous example, the sequential circuit needs to be unrolled for at least as many clock cycles as the longest continuous chain of flip-flops in the netlist. In this section, we explore another criterion to be considered for unrolling the netlist. Consider the example as shown in Figure 2. The discussion from the previous section suggests that it might be necessary to unroll the circuit for at least five clock cycles to obtain an accurate representation. This is because the effect of a key gate inserted at the input of FF4 might not be observable at output PO1 unless unrolled for five clock cycles. However, since FF4 has another path (through FF5) to a different primary output PO2, the effect of the key gate might be observable on PO2 after unrolling for only three clock cycles. This shows that it is essential to consider *sneak paths* during the encryption process. A sneak path can be defined as any path from a signal in a flip-flop chain that leads to gates outside the chain's cone of dependency. Sneak paths can potentially cause encryption information to leak to their respective sinks (one or more primary outputs), thus defeating the protection offered by the encryption scheme. Therefore,

TABLE I
Benchmark Details

| Benchmark Circuit | PI Count | PO Count | Gate Count | Flip-flop Count |
|---|---|---|---|---|
| s5378 | 35 | 49 | 2779 | 179 |
| s15850 | 77 | 150 | 9772 | 534 |
| s35932 | 35 | 320 | 16065 | 1728 |
| b14 | 32 | 54 | 9767 | 245 |
| b15 | 36 | 70 | 8367 | 449 |
| b20 | 32 | 22 | 19682 | 490 |
| b21 | 32 | 22 | 20027 | 490 |
| b22 | 32 | 22 | 29162 | 735 |

when encrypting this circuit for SAT resiliency, it is advisable to put a key-gate at the input of FF3 (instead of FF4) to obtain a better solution. This will ensure that the effect of the key gate can be observed only at PO1 after unrolling for at least four clock cycles (instead of at PO2 after unrolling for three clock cycles). It can be seen from this example that choosing the encryption points carefully can force the SAT attack to unroll for more number of clock cycles.

### *C.* Encryption Process

Based on our observations, we propose an encryption strategy described in Algorithm 2. The method starts by reducing the given sequential system $S(V, E)$ to a simplified graph $\tilde{S}(X, Y, L)$ such that $\tilde{S}$ contains only the primary inputs (set $X$), primary outputs (set $Y$) and flip-flops (set $L$). Using this simplified graph, flip-flop chains in $\tilde{S}$ are identified to form a set of all possible chains, $D_{chains}$. The algorithm uses $D_{chains}$, to rank each node $L_i \in L$ on the basis of its distance from a primary output, $L_i^j$. Distance $L_i^j$ is defined as the number of flip-flops in the path between the node $L_i$ and the primary output $Y_j$ $(\in Y)$. For each node $L_i$, the minimum distance value $L_i^{min}$ is considered during the ranking process to ensure that no shorter sneak paths to any other primary outputs exist. Using this ranking information, the top $L_{key\_size}$ nodes are chosen for encryption by adding key-controlled XOR/XNOR gates in $S$. The algorithm exits after all XOR/XNOR gates are added to yield the encrypted circuit $S_K(V, E, \vec{K})$.

## V. Results

This section describes the details of our experimentation of using the sequential SAT attack (Algorithm 1) on logic encrypted benchmarks with no scan architecture. For this purpose, benchmarks from the ISCAS'89 and ITC'99 suites are chosen for evaluation. Table I highlights the details of the chosen benchmarks. The SAT tool [26] provided by authors of the original SAT paper [8] has been used to mount the sequential SAT attack.

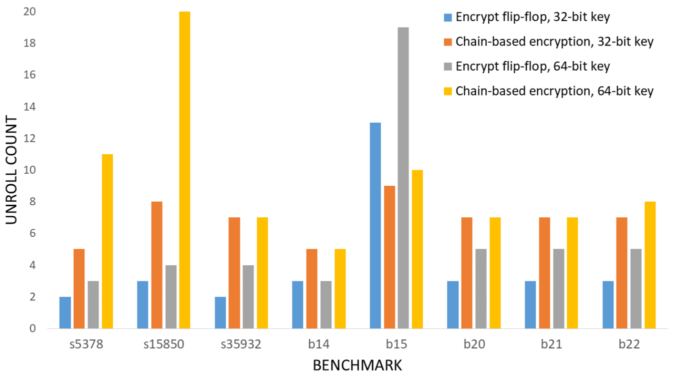


Fig. 3. Comparison of encrypt flip-flop and chain-based encryption against Sequential SAT attack

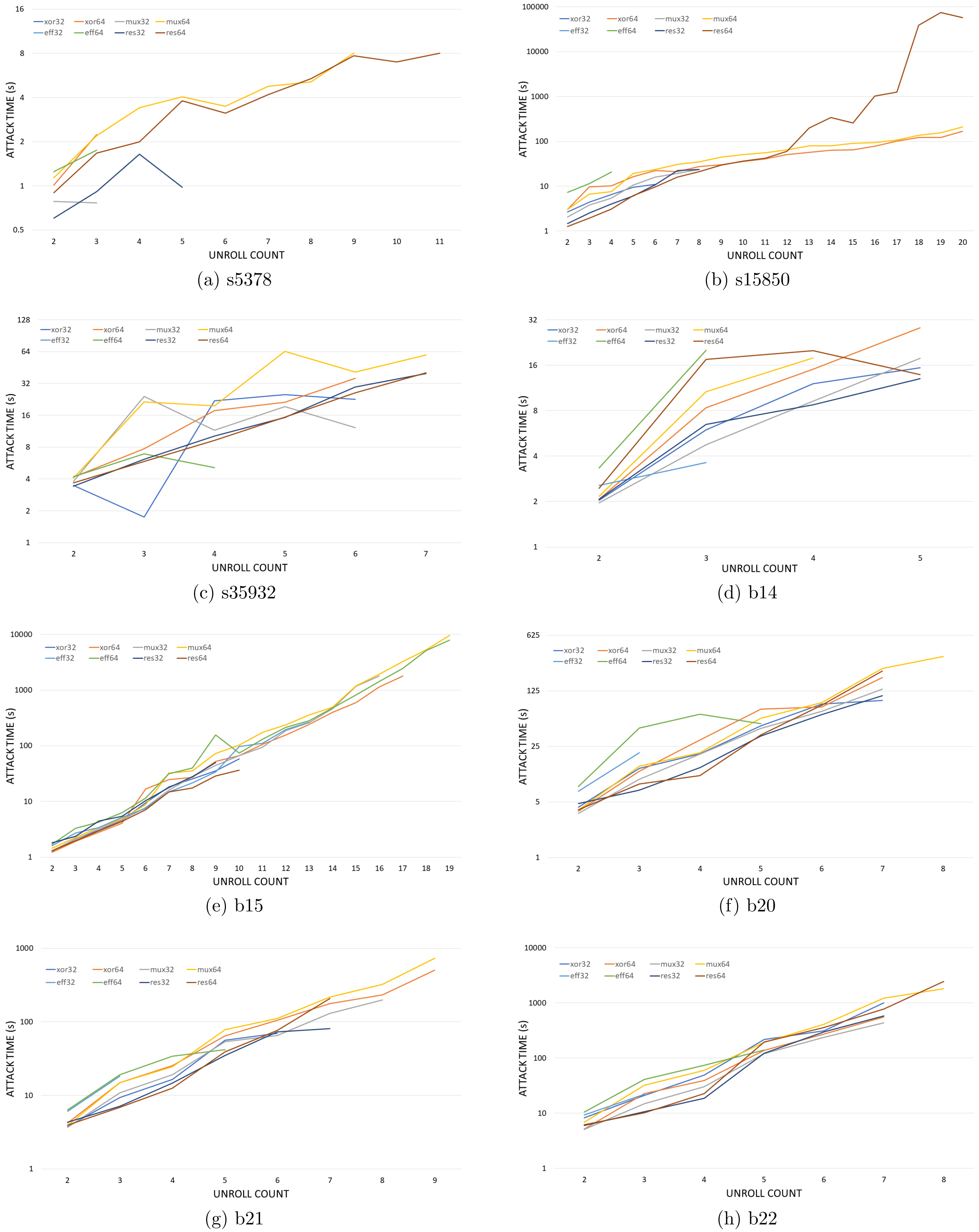
### *A.* Attack Effectiveness

The chosen benchmarks are encrypted using three different encryption strategies using 32-bit and 64-bit keys. X(N)OR scheme randomly selects signals in the netlist to insert a key controlled XOR/XNOR gate. MUX strategy uses a payload selection technique similar to X(N)OR, in addition to further randomizing the selection of signals to be connected as incorrect inputs to the MUXes. EFF encrypts the netlist using the encrypt flip-flop strategy [22]. If the key size is greater than the number of available strong flip-flops[1], the rest of the circuit is encrypted using X(N)OR logic. As can be observed from the results in Tables II and III, most benchmarks were decrypted in under 10 unrolls. Even for large benchmarks, the unroll count does not exceed 10 for most cases. For each benchmark, the sequential SAT attack time is plotted against the unroll count metric for each encryption style and all key sizes in Figure 4. The graph for a particular benchmark, encryption style and key size is plotted up till the unroll count at which the exact or equivalent correct key was found. As can be noted, for most benchmarks, the time taken to decrypt $S^{unr}$ for most values of $unr$ stays under 10 minutes. For larger benchmarks and larger unroll counts, the attack time ranges from 1-3 hours. This shows that the sequential SAT attack can be used as an effective attack technique against the specified encryption techniques, especially encrypt flip-flop.

### *B.* Attack Resiliency

To evaluate the effectiveness of the chain-based encryption (CBE) technique against the sequential SAT attack method, we encrypt the same benchmarks (from Table I) with 32-bit and 64-bit key values. The encrypted benchmarks are attacked using the sequential SAT attack. The results of the experiment are listed in the last three columns of Tables II and III as well as in Figure 4. For all benchmarks, the minimum number of unrolls required to attack the circuit is greater than the largest value, $L_{max}$ in the set $L_{ranking}$ (shown in column 2 of each table). If we

[1] A set of flip-flops is considered strong if all primary outputs in the output cone of dependency of each flip-flop in the set are present in the output cones of dependency of all remaining flip-flops in the set.



(a) s5378

(b) s15850

(c) s35932

(d) b14

(e) b15

(f) b20

(g) b21

(h) b22

Fig. 4. Sequential SAT Attack time vs. Unroll count

TABLE II

Sequential SAT Attack Resiliency (key size = 32)

| Benchmark Circuit | Chain length | X(N)OR | | | MUX | | | EFF | | | CBE | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | unr | itr | tao | unr | itr | tao | unr | itr | tao | unr | itr | tao |
| s5378 | 4 | 2* | 6 | 1.08 | 3* | 5 | 4.32 | 2! | 4 | 4.32 | 5* | 5 | 1.08 |
| s15850 | 6 | 6 | 8 | 0.31 | 8 | 13 | 1.24 | 3* | 4 | 1.27 | 8 | 9 | 0.31 |
| s35932 | 6 | 6* | 5 | 0.18 | 6* | 6 | 0.72 | 2* | 1 | 0.89 | 7* | 1 | 0.18 |
| b14 | 3 | 5* | 14 | 0.32 | 5* | 25 | 1.28 | 3* | 11 | 1.4 | 5* | 4 | 0.32 |
| b15 | 4 | 10 | 8 | 0.36 | 16 | 15 | 1.45 | 13* | 10 | 1.66 | 9 | 4 | 0.36 |
| b20 | 3 | 7* | 20 | 0.16 | 7* | 22 | 0.63 | 3 | 4 | 0.69 | 7* | 20 | 0.16 |
| b21 | 3 | 6 | 15 | 0.15 | 8* | 23 | 0.62 | 3 | 3 | 0.69 | 7* | 20 | 0.15 |
| b22 | 3 | 7* | 19 | 0.1 | 7* | 24 | 0.42 | 3 | 2 | 0.47 | 7* | 14 | 0.1 |

TABLE III

Sequential SAT Attack Resiliency (key size = 64)

| Benchmark Circuit | Chain length | X(N)OR | | | MUX | | | EFF | | | CBE | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | unr | itr | tao | unr | itr | tao | unr | itr | tao | unr | itr | tao |
| s5378 | 4 | 3 | 12 | 2.16 | 9* | 12 | 8.65 | 3! | 8 | 8.65 | 11* | 8 | 2.16 |
| s15850 | 6 | - | - | 0.62 | - | - | 2.48 | 4* | 19 | 2.65 | - | - | 0.62 |
| s35932 | 6 | 6* | 9 | 0.36 | 7* | 12 | 1.44 | 4* | 3 | 1.79 | 7* | 1 | 0.36 |
| b14 | 3 | 5* | 24 | 0.64 | - | - | 2.56 | 3* | 15 | 2.79 | 5* | 8 | 0.64 |
| b15 | 4 | 17* | 21 | 0.73 | 19 | 31 | 2.9 | 19 | 22 | 3.5 | 10* | 9 | 0.73 |
| b20 | 3 | 7 | 31 | 0.32 | 8 | 46 | 1.26 | 5* | 11 | 1.42 | 7 | 32 | 0.32 |
| b21 | 3 | 9 | 41 | 0.31 | 9 | 44 | 1.24 | 5* | 7 | 1.38 | 7 | 32 | 0.31 |
| b22 | 3 | 7* | 36 | 0.21 | 8* | 39 | 0.85 | 5 | 11 | 0.95 | 8* | 24 | 0.21 |

Notes - *unr* column lists the total number of unrolls needed to find a correct or equivalent key.
*itr* column lists iteration counts from running SAT on $S^{unr}$($unr$ = corresponding value from the *unr* column).
*tao* column highlights the total area overheads (%) counted using primitive gates.
* in *unr* column represents the condition when less than 512 keys were found.
! in *unr* column represents the condition when an equivalent key is found.
- in *unr* column represents the condition when the circuit was found to be SAT-resistant.

compare the results of the unroll count for the chain-based encryption with the encrypt flip-flop strategy, we see that the proposed method performs better than encrypt flip-flop for almost all benchmarks and key sizes. To mitigate the effect of randomness introduced by a) choosing 512 keys in each iteration, and b) using X(N)OR strategy to encrypt the key values that are not included in the strong flip-flops (using the encrypt flip-flop strategy), the experiments were performed twice for each benchmark and the most optimal outcome is presented here for analysis. As can be seen from the plot in Figure 3, for 7 of the 8 benchmarks the chain-based encryption performs better than the encrypt flip-flop strategy for both 32-bit and 64-bit keys. Furthermore, for 6 of these benchmarks, the chain-based encryption with a 32-bit key provides better SAT resiliency than the encrypt flip-flop strategy with a 64-bit key. It is important to note that from the results in Tables II and III, it would seem that CBE is only as effective as X(N)OR or MUX. However, column 2 of Tables II and III show that the $L_{max}$ values of all listed benchmarks are less than 7. Since the CBE technique is a deterministic approach, it can be argued that CBE will offer better attack resiliency with circuits which have longer DFF chains (larger $L_{max}$ values).

### *C.* Attack Enhancement

Based on these results, we can assert that a lower bound for the number of unrolls required to determine the correct key value is dependent on the largest unique chain (chain without sneak paths) present in the circuit. This means that $L_{max}$ can be used to derive the lower bound for unrolling the circuit. Note that this does not imply that there are no DFF chains with length greater than $L_{max}$. However, based on our discussion from Section IV-B, we must only consider DFF chains with no sneak paths. Therefore, using this lower bound information, an attacker can avoid unnecessary unrolling of the circuit by starting the attack process (Algorithm 1) with $unr := L_{max} + 1$. This would lead the attacker to save the time required to unroll-and-attack the sequential circuit for $2 \leq unr \leq L_{max}$. Since the time to perform unroll-and-SAT attack on large circuits increases exponentially with unroll count [27], this process could help the attacker conserve a lot of attack time and memory.

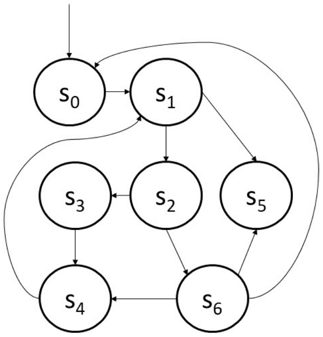


Fig. 5. FSM of b02 benchmark

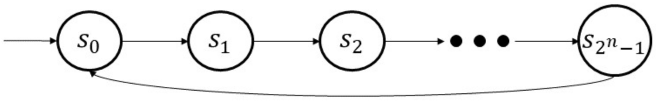


Fig. 6. FSM of n-bit counter

It is important to understand that the lower bound $U_{lower}$ ($= L_{max} + 1$) does not guarantee that the correct key would be found when attacking $S_K^{U_{lower}}$. In theory, the required number of unrolls could lie anywhere between $U_{lower}$ and $U_{upper}$, where $U_{upper}$ is the upper bound for finding the correct key. Typically, $U_{upper}$ would depend on the length of the path from the start state $V_0$ to the deepest state in the FSM. Consider the FSM for benchmark b02 as shown in Figure 5. As can be seen, the state $s_4$ is the deepest state in the FSM with a distance of 5 edges from the first edge, including the edge leading into the initial state ($\rightarrow s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3/s_6 \rightarrow s_4$). Therefore, if the transition from $s_4$ to $s_1$ is encrypted using key-gates, the sequential SAT attack method would need to unroll the system for at least 6 clock cycles to obtain the correct key value. Therefore, $U_{upper} = |P_{E_{deep}}| + 1$, where $P_E$ is the path from the initial state $V_0$ to the edge $E$, and $E_{deep}$ ($\in E$) is the edge leading outward from the deepest state $V_{deep}$ ($\in V$). In the example shown above, $V_{deep} = s_4, E_{deep} = s_4 \rightarrow s_1, |P_{E_{deep}}| = 5$. Now, consider the FSM of an n-bit counter as shown in Figure 6. If the system is encrypted in the transition between the last state ($s_{2^n-1}$) and the initial state ($s_0$), then the attack method would require $2^n$ unrolls to decrypt the correct key value. Therefore, in the worst case, $U_{upper}$ can be $2^l$ where $l$ is the number of flip-flops in the netlist. In practice however, the number of unrolls required to obtain the correct key stays closer to the lower bound, as seen from Tables II and III. It is important to note that the effort needed to determine the lower bound is increased if the circuit is obfuscated in addition to being encrypted.

## VI. **Conclusion and Future Work**

This work presented an attack method based on unrolling the sequential system prior to decryption using a combinational SAT-based attack. Using experimental results, the effectiveness of the attack in decrypting sequential logic encrypted systems is demonstrated. Furthermore, a new light-weight encryption method is presented which is shown to extend the number of unrolls (and hence the attack time) required by the SAT tool to obtain the required key value. On analyzing the results of the sequential SAT attack on this encryption method, an enhancement to the attack methodology is proposed that can assist the attacker in saving crucial attack time.

As seen from our discussion about the SAT attack enhancement, the attack time can be substantially increased if a deep state is leveraged for encryption. If the designer can estimate this deep state, using either formal methods (like model checking) or using simulation over a large amount of time, the transitions leading into and/or outward from this deep state can be encrypted, thus increasing the sequential SAT attack time significantly.

## References

[1] INC IC INSIGHTS, "Trends in the global IC design service market," http://www.icinsights.com/data/articles/documents /1035.pdf. 2018.

[2] David Manners, "Over $100bn revenues for fabless for first time," https://www.electronicsweekly.com/news/business/100 bn-revenues-fabless-first-time-2018-01/. January, 2018.

[3] John Villasenor and Mohammad Tehranipoor, "The hidden dangers of chop-shop electronics: Clever counterfeiters sell old components as new threatening both military and commercial systems," *IEEE Spectrum (cover story)*, October 2013.

[4] Mark Mohammad Tehranipoor, Ujjwal Guin, and Domenic Forte, "Counterfeit integrated circuits," in *Counterfeit Integrated Circuits*, pp. 15–36. Springer, 2015.

[5] 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.* ACM, 2008, pp. 1069–1074.

[6] Rajat Subhra Chakraborty and Swarup Bhunia, "HARPOON: an obfuscation-based SoC design methodology for hardware protection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 10, pp. 1493–1502, 2009.

[7] Sophie Dupuis, Papa-Sidi Ba, Giorgio Di Natale, Marie-Lise Flottes, and Bruno Rouzeyre, "A novel hardware logic encryption technique for thwarting illegal overproduction and hardware trojans," in *On-Line Testing Symposium (IOLTS), 2014 IEEE 20th International.* IEEE, 2014, pp. 49–54.

[8] Pramod Subramanyan, Sayak Ray, and Sharad Malik, "Evaluating the security of logic encryption algorithms," in *Hardware Oriented Security and Trust (HOST), 2015 IEEE International Symposium on.* IEEE, 2015, pp. 137–143.

[9] Kaveh Shamsi, Meng Li, Travis Meade, Zheng Zhao, David Z Pan, and Yier Jin, "AppSAT: Approximately deobfuscating integrated circuits," in *Hardware Oriented Security and Trust (HOST), 2017 IEEE International Symposium on.* IEEE, 2017, pp. 95–100.

[10] Yuanqi Shen and Hai Zhou, "Double DIP: Re-evaluating security of logic encryption algorithms," in *Proceedings of the on Great Lakes Symposium on VLSI 2017.* ACM, 2017, pp. 179–184.

[11] Yung-Chih Chen, "Enhancements to SAT attack: Speedup and breaking cyclic logic encryption," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 23, no. 4, pp. 52, 2018.

[12] Yang Xie and Ankur Srivastava, "Mitigating SAT attack on logic locking," in *International Conference on Cryptographic Hardware and Embedded Systems.* Springer, 2016, pp. 127–146.

[13] Muhammad Yasin, Bodhisatwa Mazumdar, Jeyavijayan JV Rajendran, and Ozgur Sinanoglu, "SARLock: SAT attack resistant logic locking," in *Hardware Oriented Security and Trust (HOST), 2016 IEEE International Symposium on.* IEEE, 2016, pp. 236–241.

[14] Meng Li, Kaveh Shamsi, Travis Meade, Zheng Zhao, Bei Yu, Yier Jin, and David Z Pan, "Provably secure camouflaging strat-

egy for IC protection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017.

[15] Muhammad Yasin, Bodhisatwa Mazumdar, Ozgur Sinanoglu, and Jeyavijayan Rajendran, "CamoPerturb: Secure IC camouflaging for minterm protection," in *Computer-Aided Design (ICCAD), 2016 IEEE/ACM International Conference on*. IEEE, 2016, pp. 1–8.

[16] Muhammad Yasin, Abhrajit Sengupta, Benjamin Carrion Schafer, Yiorgos Makris, Ozgur Sinanoglu, and Jeyavijayan JV Rajendran, "What to lock?: Functional and parametric locking," in *Proceedings of the on Great Lakes Symposium on VLSI 2017*. ACM, 2017, pp. 351–356.

[17] Kaveh Shamsi, Meng Li, Travis Meade, Zheng Zhao, David Z Pan, and Yier Jin, "Cyclic obfuscation for creating SAT-unresolvable circuits," in *Proceedings of the on Great Lakes Symposium on VLSI 2017*. ACM, 2017, pp. 173–178.

[18] Shervin Roshanisefat, Hadi Mardani Kamali, and Avesta Sasan, "SRCLock: SAT-resistant cyclic logic locking for protecting the hardware," in *Proceedings of the 2018 on Great Lakes Symposium on VLSI*. ACM, 2018, pp. 153–158.

[19] Muhammad Yasin, Bodhisatwa Mazumdar, Ozgur Sinanoglu, and Jeyavijayan Rajendran, "Removal attacks on logic locking and camouflaging techniques," *IEEE Transactions on Emerging Topics in Computing*, , no. 1, pp. 1, 2017.

[20] Xiaolin Xu, Bicky Shakya, Mark M Tehranipoor, and Domenic Forte, "Novel bypass attack and BDD-based tradeoff analysis against all known logic locking attacks," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2017, pp. 189–210.

[21] Mohamed El Massad, Siddharth Garg, and Mahesh Tripunitara, "Reverse engineering camouflaged sequential circuits without scan access," in *Computer-Aided Design (ICCAD), 2017 IEEE/ACM International Conference on*. IEEE, 2017, pp. 33–40.

[22] Rajit Karmakar, Santanu Chatopadhyay, and Rohit Kapur, "Encrypt flip-flop: A novel logic encryption technique for sequential circuits," *arXiv preprint arXiv:1801.04961*, 2018.

[23] Kyle Juretus and Ioannis Savidis, "Time domain sequential locking for increased security," in *Circuits and Systems (ISCAS), 2018 IEEE International Symposium on*. IEEE, 2018, pp. 1–5.

[24] Kyle Juretus and Ioannis Savidis, "Enhanced circuit security through hidden state transitions," in *Government Microcircuit Applications & Critical Technology Conference (GOMACTech)*, 2018.

[25] Hai Zhou, Ruifeng Jiang, and Shuyu Kong, "CycSAT: SAT-based attack on cyclic logic encryptions," in *Proceedings of the 36th International Conference on Computer-Aided Design*. IEEE Press, 2017, pp. 49–56.

[26] Pramod Subramanyan, "Decryption tool binaries and benchmark circuits," https://bitbucket.org/spramod/host15-logic-encryption, 2015, 2015.

[27] Travis Meade, Zheng Zhao, Shaojie Zhang, David Pan, and Yier Jin, "Revisit sequential logic obfuscation: Attacks and defenses," in *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 1–4.