Sequential Logic Encryption Against Model Checking Attack

Amin Rezaei
California State University, Long Beach
Long Beach, CA, USA
me@aminrezaei.com

Hai Zhou
Northwestern University
Evanston, IL, USA
haizhou@northwestern.edu

***Abstract*—Due to high IC design costs and emergence of countless untrusted foundries, logic encryption has been taken into consideration more than ever. In state-of-the-art logic encryption works, a lot of performance is sold to guarantee security against both the SAT-based and the removal attacks. However, the SAT-based attack cannot decrypt the sequential circuits if the scan chain is protected or if the unreachable states encryption is adopted. Instead, these security schemes can be defeated by the model checking attack that searches iteratively for different input sequences to put the activated IC to the desired reachable state. In this paper, we propose a practical logic encryption approach to defend against the model checking attack on sequential circuits. The robustness of the proposed approach is demonstrated by experiments on around fifty benchmarks.**

***Index Terms*—Model Checking Attack, Sequential Logic Encryption, Sequential Transformation, Sequential Encryption**

I. Introduction

Logic encryption has attracted much attention due to increasing Integrated Circuit (IC) design costs and growing number of untrusted foundries. To encrypt a circuit with a random n-bit secret key [1], a lightweight traditional approach uses n new gates. First, a combination of n buffers (for key bit "0") and inverters (for key bit "1") are chosen and matched with the bits of the key, and then each selected buffer or inverter is replaced with a key bit controlled XOR gate. In this case, the valid behavior of the circuit only happens when the correct key is applied. Moreover, MUX-based encryption [2] as another traditional approach uses one input of the 2-1 MUX for the correct wire and the other input for the wrong one while the selector of the MUX is the associated key bit. The correct key in traditional approaches will be inserted in a tamper-proof memory in post-fabrication phase or embedded into the circuit using polymorphic logic solutions [3], [4].

Although the removal attack can be easily prevented on traditional schemes, the SAT-based attack [5] can efficiently decipher the secret key. The attack uses two copies of the encrypted circuit with the same input, but different key values under a given constraint to check whether it is still possible to generate different outputs. Such input patterns are called Differentiating Input Patterns (DIPs.) Each DIP is then used to query the activated IC bought from the market to get the correct output. Then, the DIP with the output is used to further constrain the keys. However, one important fact has not been paid enough attention: The SAT-based attack is effective on combinational circuits but it cannot be utilized for decrypting sequential circuits unless the scan chain is accessible to the attacker.

In sequential circuits with the scan chain capability, there are two different modes named as regular and scan. A 2-1 MUX is placed at the input of each Flip-Flop (FF) in order to connect all the FFs in a shift register for one MUX selection while the FFs work in the regular mode for the other MUX selection. The attacker can treat the state inputs the same as the primary ones by using the scan mode if he has access to the scan chain, but that is a big if. The scan chain can be protected by scrambling the testing mode [6], adopting a partial test scheme [7], or encrypting the scan chain [8]. This can be seen as a perceptible weakness of the SAT-based attack when almost all commercial ICs are sequential ones. The scenario becomes more critical when the SAT-based attack can fall into the trap of non-combinational loops and reports a wrong key when unreachable states encryption is implemented [9].

Mimicking the concept of DIPs, a naive attack can be implemented calling an unbounded model checker in each iteration to find Discriminating Input Sequences (DISs) of arbitrary length. However, this scheme results in another impractical attack since multiple calls to an unbounded model checker is exponentially time consuming. Thus, the MC attack is proposed [10] and then improved [11] to find such DISs by adding new input sequences of bounded length in each iteration. The bound will be increased when no more DIS can be found but the correct key is still not deciphered. In this paper, we propose a practical logic encryption approach to defend against the MC attack on sequential circuits.

II. Sequential Logic Encryption

The focus of the state-of-the-art sequential encryption works [12], [13] has been on the behavioral defenses. However, structural solutions need to be designed due to the inherent state explosion problem of the behavioral methods [14].

We define $c(X, S, \delta, s_0, Y, \gamma)$ as a sequential circuit in which X is the set of input vectors, S is the set of all states, δ is the next state function, s_0 is the initial state, Y is the set of output vectors, and γ is the output function. Considering the set of key vectors K, the sequential encryption is as follows:

$$g(X, S, K, \delta, s_0, Y, \gamma) \mid \exists k^* \in K :$$
$$g(X, S, k^*, \delta, s_0, Y, \gamma) \equiv c(X, S, \delta, s_0, Y, \gamma)$$

We can abstract each sequential circuit as a State Transition Graph (STG.) If we consider the shortest paths between the

2021 Design, Automation & Test in Europe Conference & Exhibition (DATE) | 978-3-9819263-5-4/21/$31.00 ©2021110.23919/DATE51398.2021.9474002

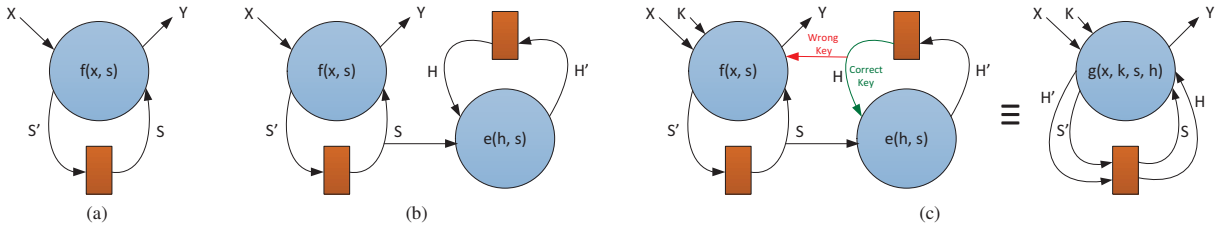


Fig. 1: Logic encryption scheme (a) Original circuit (b) Structural transformation (c) Sequential encryption

initial state and any of the other reachable states, the circuit diameter is defined as the longest path among them.

Theorem 1. *The required number of unrollings for the MC attack is bounded by the length of encrypted circuit diameter.*

Proof. If the diameter is one, s_0 is directly connected to all the other reachable states. Thus, input sequences of unit length can put the STG to any desired state. Now we assume that the diameter is d. If there is a reachable state s_i that cannot be reached from s_0 by input sequences of length less than or equal to d, it means the length of the shortest path between s_0 and s_i is greater than d. This contradicts the assumption that the longest length among the shortest paths between s_0 and any of the other reachable states is d. □

Based on Theorem 1 even if we push the important DISs to the far states from the initial state, still the MC attack complexity is limited by the diameter. Therefore, we need to answer the following questions: *How can we structurally transform the circuit to increase the diameter? How can we encrypt the circuit with a lightweight approach in order to maintain most of the original circuit structure?*

A. Structural Transformation

Four different structural operations can be applied on sequential circuits, named retiming, resynthesis, sweep, and conditional stuttering [15]. Among them sweep has come to our attention since it adds or removes FFs affecting no output. Sweep is usually met as an operation removing redundant FFs to simplify the circuit structure. However, we propose to use sweep as an operation adding a historical register to the circuit to overcome the MC attack.

Algorithm 1: Sequential-Logic-Encryption

Input: Sequential circuit netlist $f(x, s)$, maximum count c_{max}, threshold count c_t, key size n
Output: Encrypted netlist $g(x, k, s, h)$
`/* Random XOR-based encryption with n-bit key */`
$g(x, k, s)$ = Traditional_Encryption($f(x, s)$, n) ;
`/* Add a counter that counts up to` c_{max} `*/`
$g(x, k, s, h)$ = Add_Counter($g(x, k, s)$, c_{max}) ;
for *each* k_i *in* $g(x, k, s, h)$ **do**
  `/* Add MUX with` $c_t + i$ `selector */`
  $g(x, k, s, h)$ = Add_Multiplexer($g(x, k, s, h)$, k_i, c_t) ;
return $g(x, k, s, h)$;

Considering the sequential machine $f(x, s)$ of Fig. 1a, we can introduce an additional machine $e(h, s)$ that changes a historical register based on the current state of the register and the current state of the original machine. Now, if we consider the two machines together as depicted in Fig. 1b, we can push the important DISs to the far states efficiently. Far state here is the state in which the counter reaches a predefined threshold.

Theorem 2. *Sweep is sufficient in order to increase the circuit diameter without constructing the STG.*

Proof. This is a constructional proof. Suppose the circuit diameter is d. To increase the length to $c_{max} > d$ without constructing the STG, the simplest way is to introduce a counter via a sweep operation to be increased by one in each clock cycle. We just need to make sure that the counter size is large enough to count up to c_{max}. □

The current state of the original machine can be also utilized to optimize the counter increase. However, we skip the optimization part in this paper since it does not have any direct influence on the encryption step.

B. Practical Encryption

Suppose a counter that counts up to c_{max} is introduced in the original circuit. We adopt a practical encryption approach by inserting lock gates with secondary key bits to the random locations of the original circuit. We propose the following scheme to activate each secondary key bit m_i:

$$\begin{aligned} m_i = &\vee (\overline{k_i{}^*} \wedge counter = c_t + i \wedge k_i) \\ &\vee (k_i{}^* \wedge counter \neq c_t + i) \\ &\vee (k_i{}^* \wedge k_i) \end{aligned} \tag{1}$$

In which $k_i{}^*$ is a Boolean constant depicts the correct value of the primary key bit k_i. In the above scheme, the secondary key bit m_i is activated when the counter reaches to $c_t + i$. On the other hand, for the counter values below c_t, there is no output inconsistency between the encrypted circuit under the correct key and the wrong ones. Fig. 2 shows one structural realization of Equation 1.

Algorithm 1 depicts the proposed structural encryption scheme. The *Traditional_Encryption* function encrypts the sequential circuit with an n-bit key using XOR-based approach

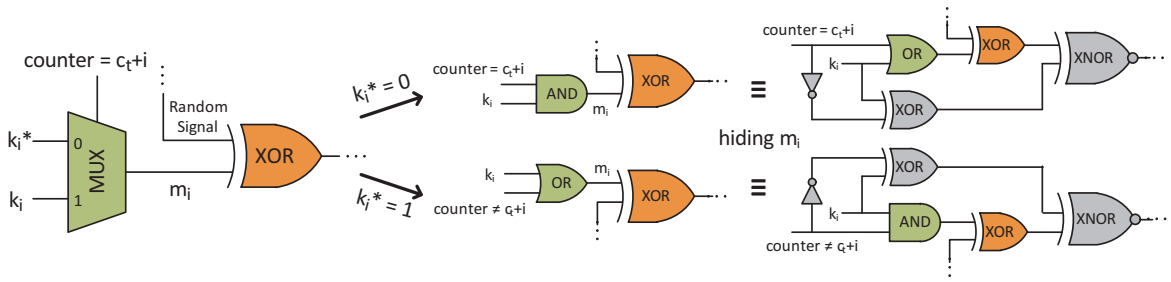


Fig. 2: A structural realization of the proposed practical encryption

[1]. Then, the *Add_Counter* function adds a counter that can count up to c_{max}. Finally, considering the threshold count c_t, the *Add_Multiplexer* function adds a 2-1 MUX for each key bit based on the structure shown in Fig. 2.

Please note that the combination of the threshold value c_t and the key size n should be chosen in a way that the following inequality holds: $c_t + n \leq c_{max}$. In this case, if each time the bound will be increased by one in the MC attack, the below theorem can be proved.

Theorem 3. *If the sequential circuit is encrypted with the proposed practical approach, at least $c_t + n$ unrollings are required for the MC attack to find the correct key.*

Proof. For the counter values below c_t, no DIS can be found. Then, half of the remaining wrong keys can be pruned for each increase of the counter (i.e., for each additional unrolling.) Also, at least n additional unrollings are required to activate the last key bit k_n. Thus, at least $c_t + n$ unrollings are needed to decipher the secret key. □

C. Circuit Obfuscation

The proposed structural transformation not only does not suffer from the state explosion problem, but also it takes advantage of the non-linearity relation between the number of the states and the FFs. Simply speaking, increasing the register size linearly has exponential effect on c_{max}.

However, since the historical register does not affect the output vector of the original circuit, the attacker may try to remove the redundant FFs using a logic synthesis tool. Thus, we need to make sure that the output vector is dependent on the historical register under the wrong keys that is basically the definition of a wrong key in the proposed encryption scheme. In other words, only under the correct key, the historical register does not affect the original output. Thus, as shown in Fig. 1c, the historical register and the regular FFs are not distinguishable unless the correct key is known.

Since the secondary key bits are constants signals, another suggested attack is to identify the secondary key bits (i.e., m_is) and build the circuit considering them as primary key inputs (i.e., k_is.) To prevent such attack, in structural realization of Equation 1, the secondary key bits should be hidden as suggested in the right side of Fig. 2.

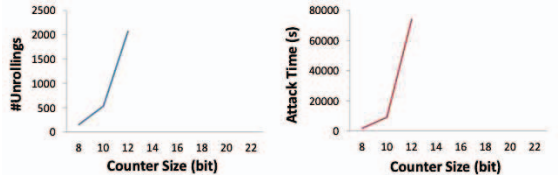
III. Experimental Results

For experiments, we have modified the sequential circuits of ISCAS'89 [16] and ITC'99 [17] to add a counter that can count up to $c_{max} = 256$. Then, the proposed practical encryption approach is adopted with the constant key size of $n = 20$ and the threshold value of $c_t = 0.5c_{max} = 128$. The decryption results under the MC attack [10] is depicted in Table I. For the benchmarks with *NR* results, the attack algorithm did not report any key after one day long running. As anticipated in Theorem 3 at least $c_t + n = 148$ unrollings are required for decrypting each benchmark. As can be seen in the solved benchmarks, the MC attack is powerful enough to find the correct key with minimum possible unrollings. However, if we double the counter size, the DIS checking procedure will be exponentially time consuming. Even for the smallest benchmark (i.e., $b02$) with 16-bit counter size, key size of $n = 20$, and $c_t = 0.5c_{max} = 32768$, the attack program was not able to report any key after one week long running. This is because for such setup at least $c_t + n = 32788$ unrollings are required.

Furthermore, with no loss of generality, we have evaluated the effect of increasing the counter size and the key size on the MC attack using $s208$ benchmark. The results are shown in Fig. 3 and Fig. 4 respectively. Again, we have assumed $c_t = 0.5c_{max}$. For the counter sizes larger than 12 bits, the program did not report any key after one day long running. Evidently, linear increase on the counter size has exponential effect on the attack time. On the other hand, linear increase on the key size still has linear effect on the attack time. Please note that the trend is the same using any other benchmark.

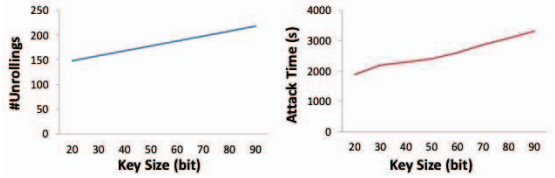
IV. Conclusion

In this paper, we proposed a practical logic encryption scheme to secure a sequential circuit against the MC attack with small circuit modification. In the traditional approaches, the circuit under a wrong key can be modeled as a circuit with static faults in which the effect of wrong key insertion can be detected quickly at the output. However, in our proposed scheme, wrong key insertion leads to intermittent faults when the circuit reaches to specific states. In this case, if the attacker does not test the circuit patiently, he may mistakenly assume a wrong key as the correct one.



(a) Unrollings (b) Attack time

Fig. 3: The MC attack on s208 benchmark with key size n=20 and different counter sizes



(a) Unrollings (b) Attack time

Fig. 4: The MC attack on s208 benchmark with 8-bit counter and different key sizes

TABLE I: The MC attack results on the proposed sequential logic encryption with 8-bit counter and key size n=20

Bench	#In	#Out	#Gates	#FFs	Time (s)	#Unrollings
b01	2	2	46	5	2201	148
b02	1	1	28	4	780	148
b03	4	4	149	30	10189	148
b04	11	8	597	66	NR	-
b05	1	36	935	34	81922	148
b06	2	6	60	9	4532	148
b07	1	8	420	49	7659	148
b08	9	4	167	21	29484	148
b09	1	1	159	28	2882	148
b10	11	6	189	17	6895	148
b11	7	6	481	31	55476	148
b12	5	6	1036	121	NR	-
b13	10	10	339	53	6722	148
b14	32	54	4775	245	NR	-
b15	36	70	8893	449	NR	-
b17	37	97	24194	1415	NR	-
b20	32	22	9419	490	NR	-
b21	32	22	9803	490	NR	-
b22	32	22	15071	735	NR	-
s208	11	2	96	8	1888	148
s298	3	6	119	14	6707	148
s344	9	11	160	15	9014	148
s349	9	11	161	15	9980	148
s382	3	6	158	21	13163	148
s386	7	7	159	6	7044	148
s400	3	6	164	21	32554	148
s420	19	2	196	16	63546	148
s444	3	6	181	21	47801	148
s510	19	7	211	6	11775	148
s526	3	6	193	21	29252	148
s526n	3	6	194	21	31767	148
s641	35	24	379	19	75612	148
s713	35	23	393	19	66504	148
s820	18	19	289	5	24901	148
s832	18	19	287	5	19845	148
s838	35	2	390	32	24094	148
s953	16	23	395	29	37012	148
s1196	14	14	529	18	75132	148
s1238	14	14	508	18	69947	148
s1423	17	5	657	74	NR	-
s1488	8	19	653	6	NR	-
s1494	8	19	647	6	NR	-
s5378	35	49	2779	179	NR	-
s9234	19	22	5597	228	NR	-
s13207	31	121	7951	669	NR	-
s15850	14	87	9772	597	NR	-
s35932	35	320	16065	1728	NR	-
s38417	28	106	22179	1636	NR	-
s38584	12	278	19253	1452	NR	-

REFERENCES

[1] J. A. Roy, F. Koushanfar, and I. L. Markov. Epic: Ending piracy of integrated circuits. In *Design, Automation and Test in Europe (DATE)*, pages 1069–1074, 2008.

[2] 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)*, pages 953–958, 2012.

[3] A. Rezaei, Y. Shen, S. Kong, J. Gu, and H. Zhou. Cyclic locking and memristor-based obfuscation against cycsat and inside foundry attacks. In *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pages 85–90, 2018.

[4] A. Rezaei, J. Gu, and H. Zhou. Hybrid memristor-cmos obfuscation against untrusted foundries. In *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 535–540, 2019.

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

[6] D. Hely, M. L. Flottes, F. Bancel, B. Rouzeyre, N. Berard, and M. Renovell. Scan design and secure chip. In *International On-Line Testing Symposium (OLT)*, pages 219–224, 2004.

[7] M. Inoue, T. Yoneda, M. Hasegawa, and H. Fujiwara. Partial scan approach for secret information protection. In *European Test Symposium (ETS)*, pages 143–148, 2009.

[8] M. Da Silva, M. Flottes, G. Di Natale, and B. Rouzeyre. Preventing scan attacks on secure circuits through scan chain encryption. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, volume 38, pages 538–550, 2019.

[9] A. Rezaei, Y. Li, Y. Shen, S. Kong, and H. Zhou. Cycsat-unresolvable cyclic logic encryption using unreachable states. In *Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 358–363, 2019.

[10] M. E. Massad, S. Garg, and M. Tripunitara. Reverse engineering camouflaged sequential circuits without scan access. In *International Conference on Computer-Aided Design (ICCAD)*, pages 33–40, 2017.

[11] K. Shamsi, M. Li, D. Z. Pan, and Y. Jin. Kc2: Key-condition crunching for fast sequential circuit deobfuscation. In *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pages 534–539, 2019.

[12] Y. M. Alkabani and F. Koushanfar. Active hardware metering for intellectual property protection and security. In *USENIX Security Symposium*, article 20, 2007.

[13] R. S. Chakraborty and S. Bhunia. Harpoon: An obfuscation-based soc design methodology for hardware protection. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, volume 28, pages 1493–1502, 2009.

[14] A. Valmari. The state explosion problem. In *Lectures on Petri Nets I: Basic Models, Lecture Notes in Computer Science*, volume 1491, pages 429–528. Springer, 1998.

[15] L. Li and H. Zhou. Structural transformation for best-possible obfuscation of sequential circuits. In *International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 55–60, 2013.

[16] F. Brglez D. Bryan and K. Kozminski. Combinational profiles of sequential benchmark circuits. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 3, pages 1929–1934, 1989.

[17] F. Corno et al. Rt-level itc'99 benchmarks and first atpg results. In *IEEE Design Test of Computers*, volume 17, pages 44–53, 2000.