A Novel Low Complexity Logic Encryption Technique for Secure Design

Deeksha Sharma[1]and Shweta Agrawal[2]
[1]Research scholar,[2]Assistant Professor,
Department of Electronics and Comm., SRCEM Banmore, Morena, India

***Abstract*—Several applications demand secure devices to prevent loss of data or leak of information. The use of untrusted intellectual property (IP) and foundry may cause IP theft, IC overproduce, counterfeit and or insert malicious functionality called hardware Trojan.The logic encryption technique hides the functionality by inserting additional circuitry such that the design functions correctly with valid key and provides wrong output with invalid key. This work reviews existing insertion based techniques and then presents a new low overhead logic encryption technique. Designs are implemented on Tanner and simulated with 45nm technology file. Simulation results show that proposed techniques technique provides small area, power and delay over the existing logic encryption techniques.**

***Keywords*—Hardware security, Energy Efficiency, Integrated Circuits, VLSI.**

I. Introduction

The integrated circuits (ICs) are used in several applications including transportation, health, banking, energy, and military. However, an application may lose significant information due the use of untrusted IC which utilizes untrusted third-party within the IC design flow. An untrusted foundry has access to the design which may cause intellectual property (IP) theft, IC overproduce, counterfeit and or insert malicious functionality called hardware Trojan (HT). The IP piracy and counterfeiting are expected to cause of $1.7 billion in 2015 [1]. Therefore, the security of the IC has been a prime concern due to the emergence of recent attacks called HT. In the HT, malicious alterations are done during design or fabrications to corrupt the IC. The HT may cause the design malfunction or steal useful information to defame the designer reputation. Further, it is hard to detect the HT, as the adversary inserts Trojans at the rare triggered nets. Therefore, trusted design of an IC requires own foundry and design of all the IP from the scratch which is very difficult and costly in the present rapid changing semiconductor industries.

Design for trust (DFT) techniques [2] are introduced to prevent HT and also makes them easy to detect if inserted. These techniques include design modification such that HT can be detected easily. The techniques are based on hiding the original functionality by inserting additional logic and the IC works correctly when correct key is applied. The IC provides wrong functionality when the invalid key is applied. The key is stored in the temper proof registers. Since the trusted costumer gets the access to the key, it can access the correct functionality. Therefore, logic insertion based DFT techniques have become new area of research for increasing the reliability of the IC.

Several approaches are presented in the literature that insert additional logic to hide the functionality. A XOR/XNORbased insertion technique inserts XOR/XNOR to control the output [2]. It introduces significant area overhead. Apart from large area overhead, the XOR/XNOR based logic exhibits fixed key which is easy to detect reduces the strength of the security. Similar to XOR based insertion, 2x1 mux based, look up table (LUT) based logic encryption techniques are proposed in the literature. The 2x1 mux based design introduces an 2x1 mux to hide the functionality whereas the LUT based technique replaces the current gate with the LUT. The advantage of using LUT (4x1 mux) is that it can provide the functionality of any 2input gate. Therefore, it provides higher security over the other existing. Although, the security is high, it exhibits very large area overhead.

This paper makes an attempt to address the limitations of existing logic encryption techniques and proposed a new encryption technique. The rest of the paper is organized as follows. Section II provides an extensive literature review on various encryption techniques. The proposed encryption technique is detailed in Section III. The simulation results analysis of the proposed encryption over existing are discussed in Section IV. Finally, Section V concludes the paper.

II. Review on Logic Encryption

This section presents review on different techniques to achieve efficient encryption.

2.1 Random XOR/XNOR Insertion

A methodology that stops the piracy of the IC automatically by hiding the functionality. This technique is known as EPIC (Ending Piracy for IC) [2], [3] and its locks the IC by an external key. The advantage of this technique is that the key is unique for each IC and cannot be duplicated and it can be only generated by the authenticated user. Further, the EPIC dose not requires any change in the convention VLSI design flow and can follow similar IC design and verifications as the conventional design.

To protect a design, new key gates are added such that they will be activated/deactivated based on the key value which in turn changes the functionality of the existing design. For example, if a XOR gate is connected with a wire and another input is connected with key, it provides original or inverted value the at the wire depending upon the value of key. Therefore, based on the value of key, the logic value of the wire can be changes. Therefore, random insertion of the XOR gate in the circuit will encrypt the design and will provide the correct functionality when

correct key is applied. An insertion of the NXOR gate in the half adder design is shown in Fig. 1.

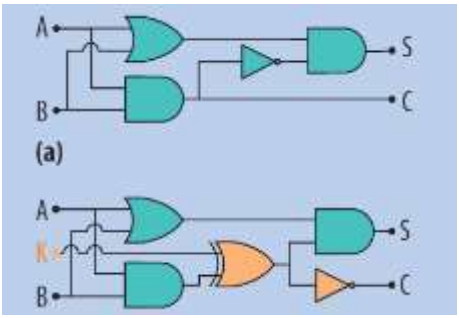
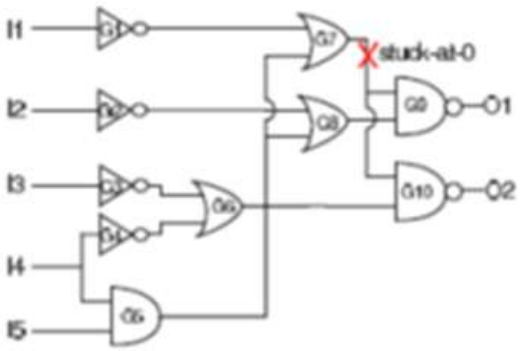


Fig. 1: Circuit diagram a) half adder, b) encrypted using NXOR gate.

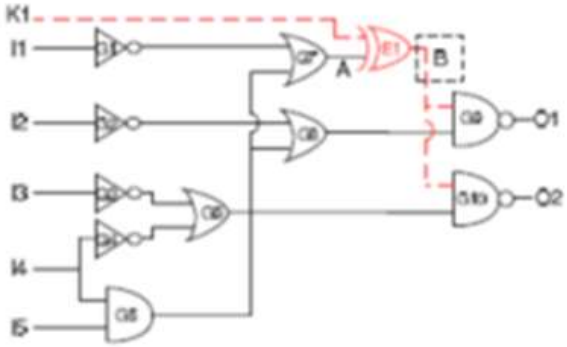
It can be seen from the Figure 2.1(a) that original design of the half adder contains one inverter, one OR and two AND gates whereas the encrypted design using XNOR requires only addition XOR gate. The inverter of the XNOR is propagated in the design such that it becomes difficult to find weather inserted gate is XOR or XNOR. The encrypted half adder works correctly when the valid key is applied which is 1. On the other hand, it produces incorrect output for when the key=0 From the design it can be seen that XOR/XNOR based encryption increase area and latency significantly.

2.2 Fault Analysis Based Insertion

A fault analysis based logic encryption [4] which provides better compression over the random generation based logic encryption. It is shown that logic encryption using random insertion does not guarantee to provide wrong output for the invalid key. The showed that this scenario is similar to the IC testing where the fault is not propagating to the output. In order to ensure the wrong key produces the wrong output, two conditions should be met which are 1) applications of the wrong key should generate a fault and 2) the fault must be propagated to the output. In order to meet the first condition, a trigger logic must be inserted into the design such that activation of this produces a fault, whereas to propagate the fault to the output, fault must be generated at the point which gets sensitize to output.



(a) A faulty circuit



(b) An encrypted circuit

Fig.2: Relation between logic encryption and fault analysis.

The analogy between the fault analysis and the logic encryption can be understood with the example shown in Fig. 2 [5]. Let us consider a stuck-at-0 fault inserted at the output of gate G7. This fault will propagate to the output of both gates when activated. This fault propagate is similar to the circuit shown in Fig.2(b) where applications of wrong key will invert the input and will produce inverted logic at the output. Even though the activation of the fault, for some input, the fault may not propagate to the output. For example, application of 01110 at the input of Fig. 2(a) will activate the fault but will not propagate i.e. it will provide 00 as output which is similar to the functional output. It is nothing but fault masking. It is also possible that the fault may be inserted at multiple places where activation and propagation of one fault may mask the other.

2.3 Smart Logic Encryption

Jeyavijayan et al. [6] demonstrated that an attacker can decipher the key in a linear time, by sensitizing the key to the output. The author demonstrated a methodology that can fix this problem by introducing a small logic encryption technique. In the logic encryption, additional gates are inserted in the design with key input. In order to achieve the correct functionality valid key must be inserted.

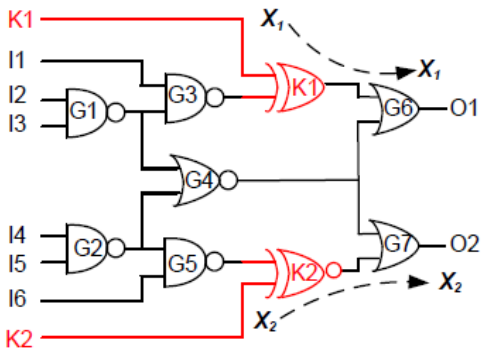


Fig. 3: Logic encrypted circuit using two key gates.

To understand the sensitization of the fault to the output to detect the original key let us consider an example shown in Fig. 3. The application of the 100000 to the input will sensitise the key to the output. Therefore, an attacker can identify the key using this sensitization approach. To

avoid this sensitization based key extraction, logic encryption technique must insert the gate such that sensitization depends on the multiple keys. Therefore, smart logic encryption technique cannot propagate the key to output as shown in Fig. 4. If the effect of key can be muted, then the sensitization attack can determine the key. Therefore, to avoid this sensitization attack attacker should introduce non-mutable key gates.

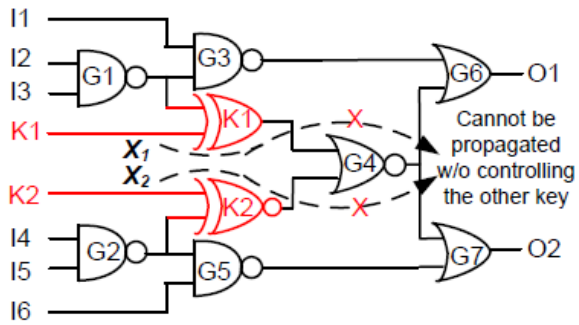


Fig. 4: Smart logic encryption.

If the inserted key gates are non-mutable and the number of key gates are more than 100, then it will take more than several years to extract the key. The author implemented the proposed techniques on the various benchmark inputs and showed that proposed techniques improves the security without much area overhead.

2.4 Logic Cone Analysis Based Encryption

Lee et al. [7] presented a logic cone analysis technique that an attacker can used to reduce the number of brute force attack to decipher the key and then presented a new technique that can be used to further increase the efforts of an attacker. It is shown that that not all the key gates affect the output i.e. only few key affects an output pin. Therefore, brute force is applied to crack these keys. Once the attacker finds these keys, the attacker uses this information to reduce the number of brute force attempts. For example, consider a circuit shown in Fig. 5.

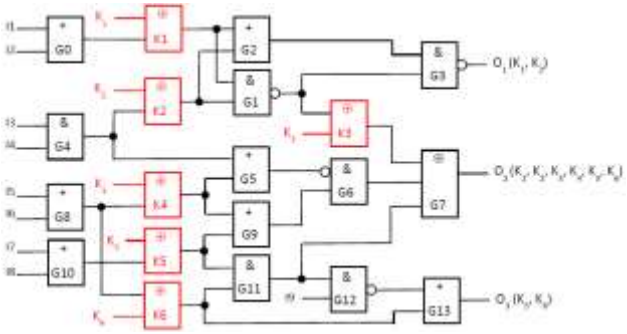


Fig. 5: Circuit withvariant key sensitivity on output.

It can be observed from the Fig. 5 that the circuit exhibits six key gates. Therefore, it requires 2^6 i.e. 64 brute force attempt to crack the key. It can also be seen that not all output pin depends on all key i.e. it depends on small number of keys. For example, the output O_1, O_2 and O_3 depends on two, six and two keys, respectively. Therefore, an attacker can reduce the number of attempts by first applying brute force on the output pin which exhibits less dependence on key and then to the next lower one.

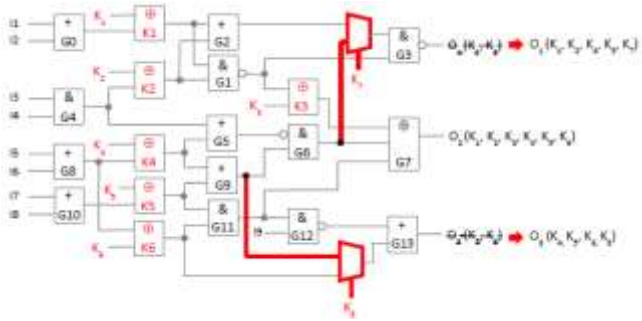


Fig. 6: Mux insertion to increase attacker effort.

For example, for the above circuit, attacker will first decipher the keys K1 and K2 by applying 4 brute force attempts and then extracts keys K5 and K6 by applying 4 more brute force attempts. Now only two key left which can be detected by applying only 4 more attempts which results in overall 12 brute force attempts compared to the 64 attempts required ideally. To improve the security of the design, designer should increase the dependency of each pin to large number of keys. Therefore, to improve the key dependency additional MUXes are inserted. The MUX insertion is done in a non-deterministic manner so that even if an attacker knows the key insertion algorithm, the attacker cannot exploit this knowledge to decipher the logic obfuscation.

2.4 Stack and TG Based Logic Encryption

Juretus et al. [8], [9] presented a replacement based logic encryption where they present new reconfigurable designs. Since the insertion based logic encryption provides large overhead, the author showed that replacement based technique provides better security with very small overhead. In this paper, stacked based topology and transmission gate based topology to design reconfigurable logic. For example, a circuit shown in Fig. 7 below can work as either NAND or NOR gate based on key value. It can be seen from the figure that when key value is logic '0', it provides the functionality of the NAND gate whereas for key equal to logic '1' it works as NOR gate. The advantage of the circuit is its lower implementation complexity which is achieved due to the shared function. Further, the based on the value of key either upper PMOS stack is gated off or the lower NMOS stack is gated off which effectively reduces the capacitance. Thus, this provides improved performance over the existing designs. Although the design exhibits small overhead, it is only applicable to the applications where invalid key does not necessarily produce wrong output. In addition to the above mentioned gate stacked topology, the paper presented a transmission based topology [9] as shown in Fig. 8.

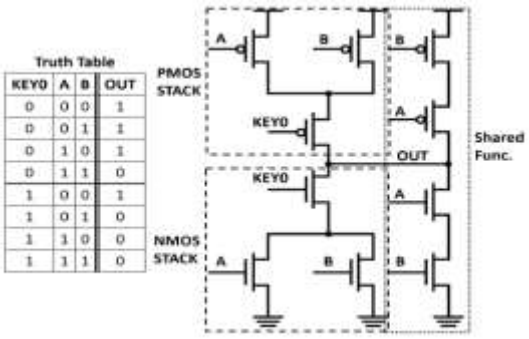


Fig. 7: Reconfigurable circuit for NAND/NOR.

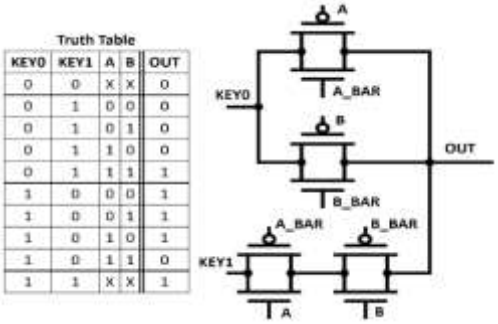


Fig. 8: Transmission gate based encryption.

Based on the value of the key, the circuit can either work as AND or OR gate, for example, the circuit provides the functionality of the AND for key combination of 01 while OR gate for the key combination of 10. The additional advantage of the transmission based topology is that it provides constant logic 0 and logic 1 for the key combination of 00 and 11 respectively. Although, the requires small area, this design fails to provides correct logic due to the transmission logic.

IV. PPROPOSED LOGIC ENCRYPTION

The major limitation of the existing logic encryption techniques is their large overhead due to the additional insertion. For example, in XOR/XNOR based logic encryption techniques, additional XORgate is inserted to encrypt any logic gate. For example, to encrypt an AND gate, it requires large overhead since the number of transistor required to implement an XOR gate is much larger over the AND gate. To reduce this overhead, a new logic encryption technique is proposed as shown in Fig. 9.

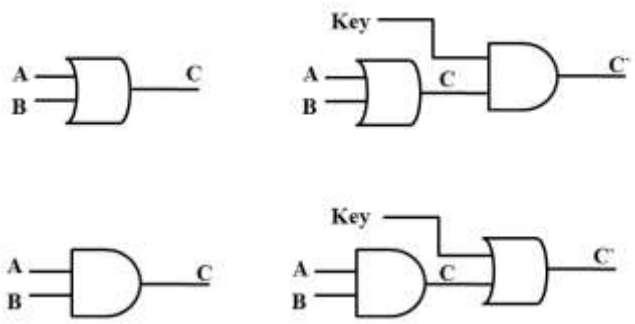


Fig. 9: Proposed logic encryption via OR/AND.

It can be seen from the Fig. 9 that to encrypt any gate an OR gate or an AND gate is inserted after that gate. With valid key only, the design will provide true functionality otherwise it will give wrong output. The valid key with OR gate insertion is logic '0' while for the AND gate is logic '1'. If the invalid key is applied in OR gate which is logic '1', it will constantly provide logic '1' at the output. Similarly, insertion of AND gate with invalid key ('0')provides constant logic '0' at the output. Table 1 shown below represent the correct functionality of the gate with different logic insertion and various value of the key.

Table 1: Truth table of proposed gate with valid/invalid key.

Key	Input		Encrypted gate AND		Encrypted gate OR	
	A	B	C	C'	C	C'
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	0	1	0	0	0
0	1	1	1	0	1	1
1	0	0	0	0	0	1
1	0	1	1	1	0	1
1	1	0	1	1	0	1
1	1	1	1	1	1	1

It can be observed from the Table 1 that encryption via AND gate provide correct functionality for the key=1 and incorrect with logic '0'. The reverse is true for the encryption via OR gate. In order to increase the security of the design, multiple keys can be inserted. Further, different key gate for example both AND and OR gate can be inserted with corresponding different valid key pattern.

IV.EXPERIMENTAL RESULT & ANALYSIS

The existing and proposed encryption techniques are implemented on Tanner to compute design metrics. The net-lists is extracted for all the implemented designs which are then simulated with 45nm predictive technology model fileon spice simulator to achieve metrics. In the simulation area, power and delay are computed for each design.

4.1 Simulation results on Tanner

The proposed and existing encryption techniques are implemented on Tanner to evaluate the design metrics. The schematic of the NANDgate is shown in Fig 10. Similarly, other gates are also implemented and design metrics are evaluated as shown in Table 2.

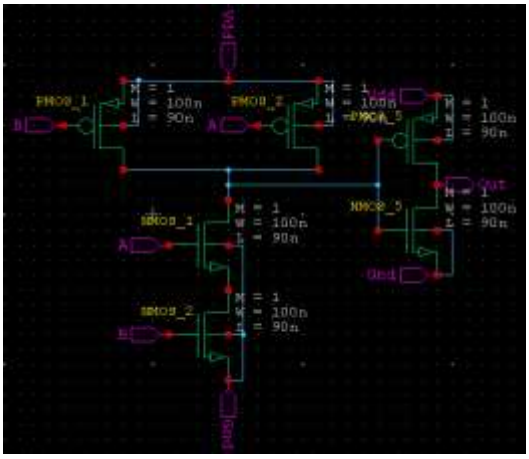


Fig. 10: Schematic of NAND gate on Tanner

Table 2: Design metrics of the basic gates.

Cell	Area (#Tran)	Power (µw)	Delay (ps)
NOT	2	0.072	6.9
AND2	6	0.174	64.37
OR2	6	0.237	57.9
XOR2	12	0.31	31.1
XNOR2	12	0.251	51.1
Mux2_1	12	0.411	67.7
Mux4_1	34	1.11	141.6

Further, an AND gate is encrypted with proposed and existing encrypting techniques and are simulated to achieve the design metrics as shown in Table 3.

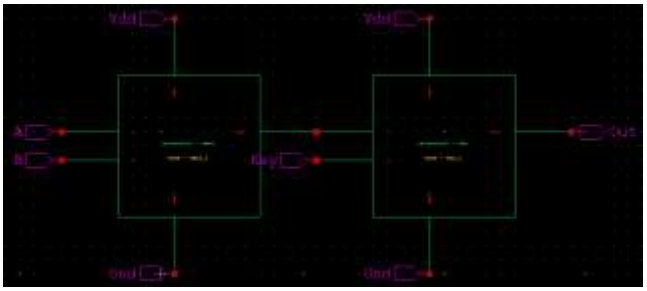


Fig. 11: Schematic of encrypted AND gate.

Table 3: Metrics of variant encrypted AND gate.

Encryption Technique	Area (#Tran)	Power (µw)	Delay (ps)
AND2_XOR2 [3]	18	0.576	223.6
AND2_XNOR2 [3]	18	0.606	130.7
AND2_Mux2_1	18	0.534	141.6
AND2_Mux4_1	34	1.04	327.5
ISCAS_AND_OR [7]	10	0.302	73.7
ISCAS_Tx_Logic [8]	12	0.0183	28.6
AND2_OC	18	0.515	200
AND2_Prop2	12	0.453	125.5

Fig. 12 shows a comparison of different encryption on the basis of implementation complexity. It can be observed from the figure that proposed logic requires small area over the most of the existing techniques.

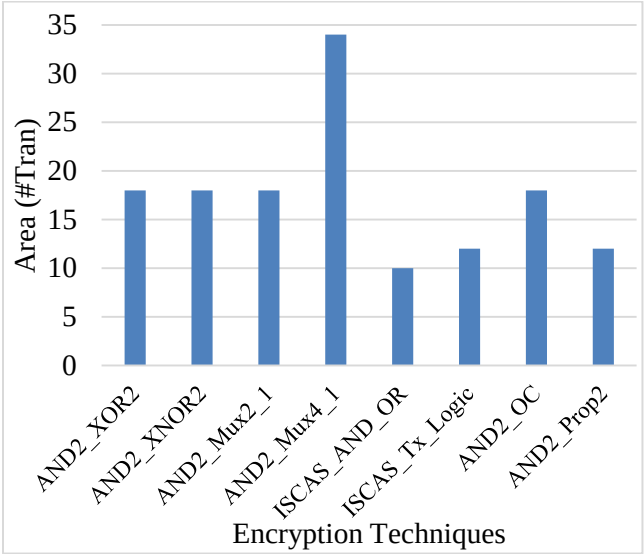


Fig. 12: Area of variant encrypted AND gate.

V. CONCLUSION

Several techniques to encrypt the design have been proposed in the literature to hide the functionality by either inserting the addition logic/circuit or replacing circuit with new reconfigurable logic.This work presents a new insertion base logic encryption technique where either a OR gate or a AND gate is inserted to hide the original functionality. Based on key value design provides correct functionality with valid key while provide incorrect functionality with invalid key. The proposed and existing encryption techniques are implemented on Tanner and simulated to compute the design metrics. The simulation results show that proposed logic encryption technique provides very small overhead over the existing techniques.

REFERENCES

[1]. M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection," IEEE Design and Test of Computers, Vol. 27, No. 1, pp. 10 – 25, Feb. 2010.

[2]. J. A. Roy, F. Koushanfar, and I. L. Markov, "EPIC: Ending piracy of integrated circuits," in *Proc. Design, Auto. Test Europe*, 2008, pp. 1069–1074.

[3]. J. Roy, F. Koushanfar, and I. Markov, ''EPIC: Ending piracy of integrated circuits,'' IEEE Computer, vol. 43, no. 10, pp. 30–38, Oct. 2010.

[4]. J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Logic Encryption: A Fault Analysis Perspective," in IEEE/ACM Design, Auto. and Test in Europe, pp. 953 – 958, Oct. 2012.

[5]. Rajendran, J. and Zhang, H. and Zhang, C. and Rose, G. and Pino, Y. and Sinanoglu, O. and Karri, R., "Fault Analysis-Based Logic Encryption," *IEEE Transaction on Computers*, Vol. 64, No. 2, pp. 410 – 424, Feb. 2015.

[6]. J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Security analysis of logic obfuscation," in *Proc. 49th IEEE Design Automation Conference*, Jun. 2012, pp. 83–89.

[7]. Y. W. Lee and N. A. Touba, "Improving logic obfuscation via logic cone analysis," 2015 16th Latin-American Test Symposium, pp. 1-6.

[8]. K. Juretus and I. Savidis, "Reducing logic encryption overhead through gate level key insertion," 2016 IEEE International Symposium on Circuits and Systems, pp. 1714-1717.
[9]. K. Juretus and I. Savidis, "Reduced overhead gate level logic encryption," International Great Lakes Symposium on VLSI, 2016, pp. 15-20.

