INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2019, VOL. 65, NO. 3, PP. 389-396
Manuscript received April 29, 2019; revised August, 2019. DOI: 10.24425/ijet.2019.129790

Design Protection Using Logic Encryption and Scan-Chain Obfuscation Techniques

V. A. Deepak, M. Priyatharishini, M. Nirmala Devi

***Abstract*—Due to increase in threats posed by offshore foundries, the companies outsourcing IPs are forced to protect their designs from the threats posed by the foundries. Few of the threats are IP piracy, counterfeiting and reverse engineering. To overcome these, logic encryption has been observed to be a leading countermeasure against the threats faced. It introduces extra gates in the design, known as key gates which hide the functionality of the design unless correct keys are fed to them. The scan tests are used by various designs to observe the fault coverage. These scan chains can become vulnerable to side-channel attacks. The potential solution for protection of this vulnerability is obfuscation of the scan output of the scan chain. This involves shuffling the working of the cells in the scan chain when incorrect test key is fed. In this paper, we propose a method to overcome the threats posed to scan design as well as the logic circuit. The efficiency of the secured design is verified on ISCAS'89 circuits and the results prove the security of the proposed method against the threats posed.**

***Keywords*—Hardware Security, Obfuscation, Logic Encryption, Scan-Chain**

I. INTRODUCTION

THE capital costs and maintenance costs for a semiconductor foundry have forced the companies to outsource the fabrication to offshore foundries. This leads to concerns of many untrusted parties within the foundries getting access to the Intellectual Property (IP). These parties can pirate the IP, reverse engineer to obtain the netlist, modify the integrated chip (IC) to insert malicious circuitry in the form of Hardware Trojans (HT). ICs can also be overproduced and sold illegally. The semiconductor industry faces heavy losses as a result of these malpractices that occur in the foundries. As a result, protecting the design of the IP has become a top priority for the companies that outsource the designs to foundries.

One way of protection of the IPs is through logic encryption of the circuit. In these encrypted circuits, only when the correct key vector is given the output of the circuit will operate correctly. Whenever the key vector is incorrect, the output will not match with the circuit's functionality. In logic encryption, the functionality of the circuit is hidden by insertion of key gates. Most of the logic encryption methods use insertion of XOR/XNOR gates at the locations. These key gates can be either inserted randomly [1] or inserted selectively at specific locations of the circuit. The determination of selective insertion key gate locations requires an algorithm [2]. This algorithm identifies the key gate locations. [3] Mentions all the logic encryption techniques that have been suggested and evaluates the advantages of each encryption. It also analyzes the vulnerabilities of each logic encryption technique and how it is overcome by the subsequent techniques. The circuit is encrypted first and then sent to for manufacturing to the foundry. After it has been manufactured, the designer acquires the IC and unlocks the chip by giving the keys correctly. However, these methods are not fully compliant since each key bit can be sensitized to the output of the circuit and the correct test key can be obtained. This vulnerability is overcome through an algorithm wherein all the key bits are made dependent on each other. In that way, sensitization of each bit to the output of the circuit is not possible since by changing one key bit, multiple key bits will get affected.

The other way is through protection by scan design. It was observed during encryption process, the intermediate states can be observed at the output of the scan chain. Hence based on this, the attacker can narrow down on the cipher key. Some countermeasures include clearing all the data in the system whenever a switching of the mode from normal to test is suggested [4]. In the method proposed in [5], where Mirror Key Registers are used for isolating the cipher key whenever there is an insecure operation. The main drawback of the scheme is that it was not feasible to do online testing. Later countermeasures included re-ordering of the scan cells in order to prevent the observation of the correct states by the attacker. However, scan based attacks are independent of the sequence of the scan cells; rather they depend on accessibility to the states of scan chain. Hence, these methods are not resilient against the signature attacks since signature attacks do not depend on the ordering of the scan cells. Hence the main objective is to obfuscate the scan data such that the states are not accessible to the attacker. In the scan data obfuscation proposed in [6], a shift register (SR) sequence is integrated with the scan chain of the CUT. This SR will control the value of the Test Control (TC) bit of the scan flip-flop (SFF). The test data can be scanned through only when the test key is correctly entered. This method obfuscates the scan cells as well as the data which is available at the scan output and overcomes the vulnerabilities of the previously mentioned key and lock methods. This method is however vulnerable to the Test-Mode-Only-Signature-Attack (TMOSA) [6]. This is overcome by dynamically obfuscating the data coming from the scan chain.

This paper discusses the method where scan data is obfuscated dynamically and the key dependent logic encryption of the CUT. A new method is proposed where in the dynamic obfuscation is done to the scan data of the CUT and the key interdependency block is integrated with the CUT. The

Deepak VA, Priyatharishini M and Nirmala Devi M are with Department of Electronics and Communication Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India (e-mail: vadeepak@yahoo.co.in,m_nirmala@cb.amrita.edu,m_priyatharishini@cb.amrita.edu).

proposed methodology protects the CUT from both the side-channel attacks as well as the attacks on the CUT to get the correct test key. The functionality of the CUT matches the expected functionality when the correct test key is entered at the scan chain and at the CUT side. In case of bit mismatch of the test key in either of the security modules, the functionality of the circuit changes and the output does not match with the ideal output. The proposed method obfuscates scan chain of the circuit from the side channel attacks and from the attacks on the circuit. The scan chain of the circuit is secure through the dynamic obfuscation. The functionality of the circuit is secure through the key interdependency block from which the key bits are fed to the key gates.

In this paper, Section II summarizes the evolution of scan based attacks and logic encryption. In Section III the proposed methodology is described for protection of sensitive data in the circuit and Section IV analyzes the proposed approach and section V concludes the work.

II. RELATED WORKS AND MOTIVATION

A. Evolution of scan based attacks

The data that is generated, processed or stored in the chip can be taken from the chip without breaking into the chip. This is done through side-channel attacks. The scan chains are designed for access of the Circuit-Under-Test (CUT) through test access ports. The scan based attacks involve breaking into the chip and getting the information about the data that is going into the CUT and the data that is coming out form the CUT. This data can used to deduce the key of the chip. The scan-based attacks were first proposed as two-phase attack on DES in [4]. An input register Flip-Flop (FF) is chosen in the first phase and its location in the scan chain is obtained. This is done by scanning the responses keeping the scan chain in the test mode and comparing them to the plaintext pairs that are fed in normal mode, having one bit difference loaded. This procedure can also be executed on the AES chip [5]. In [7], the detection of combinational Trojans in the circuit which can be inserted based on different parameters. The parameter considered in this work is side channel parameter of leakage power. Whenever the leakage power is above a threshold value at a certain net, the corresponding net is checked for additional combinational circuit through which Trojan can be inserted. However in these attacks, it is assumed that only cryptographic operations are performed by the registers. Also, it is assumed that the scan design consists of registers alone. There can be a mismatch in practical scenarios. This was proved in [8] where the practical scenarios were chosen. The success of these attack scenarios did not depend on the moment at which the scan chain captured the encryption result. The keys can be analyzed for their correctness by observing the output response through the scan design for each segment of encryption. The correct key is found for all segments through this process and later it is combined to obtain the cipher key. In [9], the countermeasure followed was whenever there was a mode-switching in the CUT, sensitive information in the scan chain would be erased. It was proposed to carry out the attacks only in test mode in order to overcome the countermeasure in [9]. Initially, the scan cells are in test mode and are fed with a known plaintext. When the capture phase occurs, it is encrypted and the result after encryption is obtained through the scan chain. This type of attack does not require switching back of the scan cells to the normal working mode for obtaining the encrypted result.

One of the countermeasures suggested is blocking of cipher key. In this suggested method, whenever there is an insecure operation, sensitive information is prevented from being leaked by isolating the key from the module. The key can be accessed at any given time by switching into regular working mode. However when the design is in the blocked mode, the key is not accessible. From the test mode, only possible mode transfer is to secure mode. This implies that when the design is in secure mode, it cannot go back to the test mode unless the circuit is reset. When circuit is reset, all the intermediate data in the scan chain gets cleared thereby preventing any leak of data that was observed when it was in test mode.

The next countermeasure suggested was obfuscation of scan output. The normal SFFs were replaced by state dependent scan flip-flop (SDSFF) in [10]. The SDSFFs can have output either as its historical state value or as current output response obtained from CUT. Thus, when an incorrect key is entered, few of the SDSFFs will output the current response of CUT while remaining SDSFFs will output a historical value. Later, it was observed that the order of the sub-chains of a scan chain could be reordered. This was done through a test security controller (TSC) and a linear feedback shift register (LFSR). The LFSR will get correctly seeded only when the test key entered and the secure register key match. Only after this, the secure mode testing can be performed with known sub-chain order. If invalid test key is fed, the test data scanned out will contain responses that are deceptive. However, it was later proven that the side-channel attacks as well as signature attacks do not depend on the order of the scan-chain. Hence, scan-based attacks can be successful even when the scan-chain is static obfuscated.

B. Evolution of Logic Encryption

The first of the methods proposed to overcome IP piracy issues were encryption using gates. This type of logic encryption is carried out by randomly inserting key gates into the design. The key gates are a combination of AND/OR or XOR/XNOR [2], [3] or multiplexers (MUX) [11], look Up Tables (LUT). In encryption using gates, one input of the key gate is connected to any net inside the circuit and the other input is the key input. When the correct key input is given then the functionality of the circuit is correct, else the outputs are corrupted.

In encryption using MUX [12], MUXes inserted are of size 2x1. One input of the MUX is correct value; other input is taken from any net in the circuit. The select bit of the MUX is the key bit. MUX based encryption can also be carried out by using an Obfuscation Cell (OC) [13]. An obfuscation cell is a combination of MUX and an inverter. In both the methods, the select line of the MUX acts is the key bit. For correct key input, the circuit gives the correct behavior, whereas if incorrect key bit is given then output obtained mismatches from ideal output. In these methods, each key bit could be sensitized to the output of the circuit and the information regarding the design could be obtained. These encryption techniques were thus proven to be susceptible to path sensitization attacks.

For circuits with sequential elements, encryption using FSMs was proposed. In this method, the state transition graph of the finite state machine is modified such that if an incorrect input sequence is given then the FSM goes into a state from which it will never get out. These states are called black states. Only when a correct input sequence is given to the FSM, correct functionality of the encrypted circuit can be observed. It was later observed that if all the states in FSM are not defined, then attacker could get the details of the chip by transitioning through these states.

This led to encryption of the circuits through evaluation of parameter. In [11], a parameter called fault impact is calculated for each net in the circuit. The parameter is a measure for the insertion of the key gate in the circuit. The key gate is inserted at the net which has highest fault impact. The fault impact is calculated through the following formula:

Fault Impact = $N_0P_0 \times N_0O_0 + N_0P_1 \times N_0O_1$ (1)

Where, N_0P_0 is number of patterns that detect s-a-0 fault at a given net,

$N_0O_0$is number of output bits that are corrupted for a given s-a-0 fault,

N_0P_1 is number of patterns that detect s-a-1 fault at a given net,

N_0O_1 is number of output bits that are corrupted for a given s-a-1 fault.

This technique is observed to be susceptible to various attacks such as logic cone analysis based attack, hill-climbing, path sensitization and SAT-based attacks. The major drawback of the methods is that the key value could be sensitized to the output of the circuit. To overcome this, the method of strong logic locking [12] is proposed where in the key gates are inserted in the circuit such that if any key has to be sensitized to the output, it cannot be done unless all the other key bits are also controlled at some values i.e. all key gates interfere with each other at some location in the circuit. It was observed that this method offered only limited number of key gate positions for a design with a certain number of keys i.e. the number of key gate locations did not match with the number of keys required for encrypting the circuit.

Thus, it could be observed that the key bits cannot be directly placed at the key gate locations of the circuit. In [14], method of passing the key values through a block in which all the key bits are correlated is discussed. The value of each output key bit of this block is dependent on multiple inputs. The Primary Key (PK) bits are input to this block. In this block, an algorithm is used such that all the PKs are interlinked with each other. The Secondary Key (SK) bits are the output of this block. Each SK bit is related to many PK bits. Hence if any one of the PKs is changed, more than one number of SK bits will change.

This method could easily overcome the drawbacks faced by the countermeasures listed previously to this. It can be observed that by obfuscating the scan output of the scan chain and by inserting the key interdependency for the circuit, the security for the circuit cannot be breached.

Table I summarizes the different approaches to counter the scan chain attacks and the logic encryption attacks and their vulnerabilities.

TABLE I
SUMMARY OF VARIOUS SCAN DESIGN AND LOGIC ENCRYPTION METHODS

REFERENCE	SCHEME	PROPOSED METHOD	DISADVANTAGE	CIRCUITS VALIDATED
[2],[3]	Counter measure for IP piracy, path sensitization attacks	Encryption through MUX, XOR/XNOR	Each key bit can be sensitized to the output to obtain the correct key value	ISCAS'85
[4]	Counter measure for side-channel attack	Blocking of cipher key when mode switching occurs	Online testing is disabled, cipher key cannot be accessed at all times	AES
[5]	Counter measure for side-channel attack	Erase the sensitive data whenever mode switching occurs	Test-mode only attacks can be executed successfully	AES
[10]	Counter measure for vulnerability of mode switching method	SDSFF is used in place of SFF; response is mix of current state and previous states.	This method cannot be tested on faulty designs.	RSA
[13]	Counter measure for IP piracy	FSM moves into unknown state when wrong key is entered	Not feasible if all the states are not defined.	ISCAS'89
[15]	Counter measure for side-channel attack	Shuffling of the order of multiple scan sub-chains of a scan chain	Signature attacks can be executed successfully without knowing the order of the scan chain.	Multi-bit LFSRs

III. PROPOSED METHODOLOGY

The proposed methodology consists of 4 steps. First, the scan chain integration is done. This involves connection of SFFs to the inputs of the CUT. The mode of operation of each MUX in the SFF depends on the TC bit. Next, the scan chain is dynamically obfuscated by selecting a few scan cells out of the scan chain. The test control bit of these scan cells is denoted by *tc*. The value of *tc* depends on the test key bit which will be fed to the shift register (SR) of the dynamic obfuscation logic. Next, the test key loading controller is designed. The test key loading controller will determine the time period for which the obfuscated scan cells will remain in test mode. During this time, the test key is fed through the SR. The time for which the cells will remain in test mode will depend on the number of FFs in the SR. Finally, the key interdependency block is integrated with the CUT. The nodes for insertion of key gates are determined through fault analysis based logic encryption algorithm. The primary keys for the CUT are passed through

the key interdependency block and the outputs key bits of the block are one of the inputs to the key gates of the CUT.

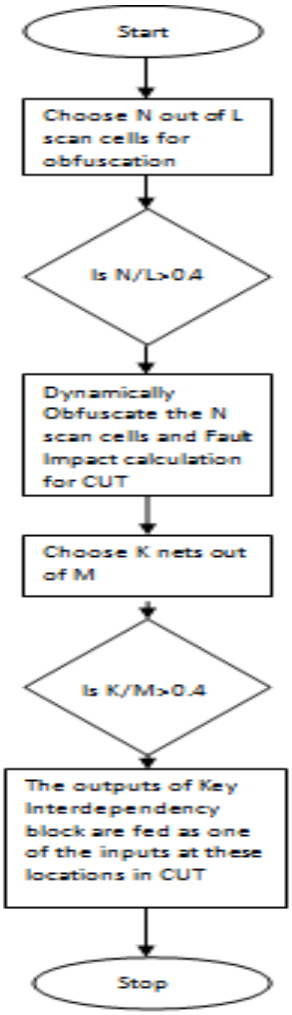

Fig.1. Flowchart for the proposed methodology

A. *Scan Chain Integration*

The scan chain is present in the periphery of the circuit. The scan chain consists of Scan Flip-Flops (SFF) that is connected together. Each SFF consists of a 2x1 MUX connected at the input of a normal FF. The test control (TC) signal is the select line to the 2x1 MUX of each SFF. The TC signal is used to control the switching of the CUT from functional mode to test mode of operation. When TC=0, the CUT is in test mode. When TC is 0, the data is serially fed to the circuit through the scan chain. Then, TC is made high to switch the CUT to functional mode.

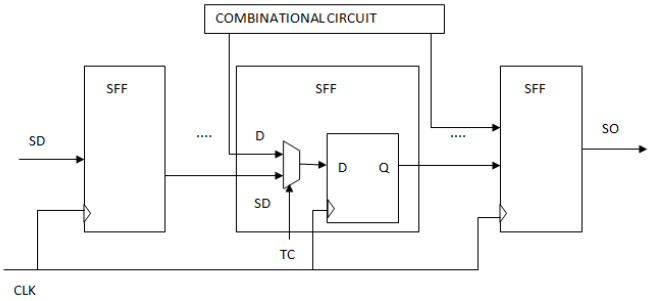

Fig. 2 Scan Chain

In this mode, the output response of the circuit is captured by the SFF present in the scan chain. When, the TC is switched back to 0 after one cycle, the captured output response is shifted through the scan chain to the scan out (SO). Simultaneously, next test input is fed through the scan chain. Figure 2 describes the structure of the scan chain. It can be observed that the data which will be fed to the subsequent FFs depends on the TC value entirely.

B. *Dynamic Obfuscation of the Scan Chain*

The next step is to obfuscate the scan chain of the design. The method followed here is dynamic obfuscation. This method involves obfuscation of the data that will be either fed into or read from the circuit. The Test Control (TC) to the SFF is controlled through the obfuscation technique.

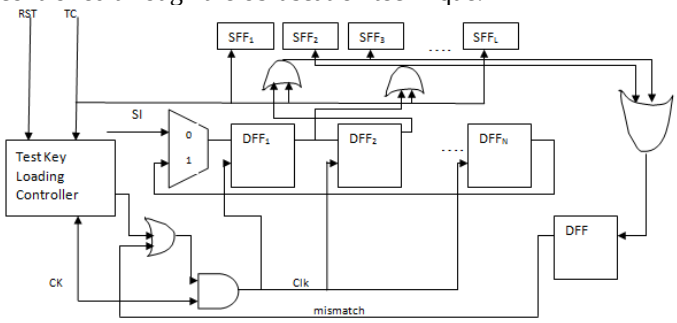

Fig. 3. Dynamic Obfuscation of Scan Cells [6]

Figure 3 describes the dynamic obfuscation scheme followed for obfuscation of data in the scan cells. The test control (*tc)* for selected number of SFFs from the scan chain is controlled by shift registers (SR) and OR gate. Consider M SFFs for the circuit. N SFFs are arbitrarily chosen for dynamic obfuscation. The SR contains N DFFs that are connected serially with 2x1 MUX at the beginning. The output of the last FF is one of the inputs of the MUX, the other input being the test key which has to be loaded.

The *tc* for these N SFFs is given through the output of OR gate. The two inputs of the OR gate are system TC and either of Q or Q` from their paired FF in SR. The choice of Q or Q` depends on the test key that is scanned into the SR. If the correct key is scanned, then during the test mode, the Q or Q` selection matches with the key and *tc* becomes 0. If wrong key is scanned, then *tc* becomes 1 and hence the SFF does not work in test mode and the output gets corrupted. The Q or Q` of the FFs in SR are also connected to an OR gate that is outside the SR. The output of this OR gate is connected to another DFF, whose output is a mismatch signal. This DFF is triggered during negative edge transition of the load SR signal *LD_SR*.

It can be seen that during the test mode, *LD_SR* should be high so that new data is loaded through the SR. When incorrect key is entered, since the DFFs of SR are connected in circular form, the key will be circularly shifted. This changes the *tc* bit of the scan cells that are dynamically obfuscated and the wrong test key will be cyclically shifted in each cycle until a new test key is fed through the SR.

C. *Test key loading controller logic*

The test key loading controller forms an important part of the dynamic obfuscation logic of the scan chain.

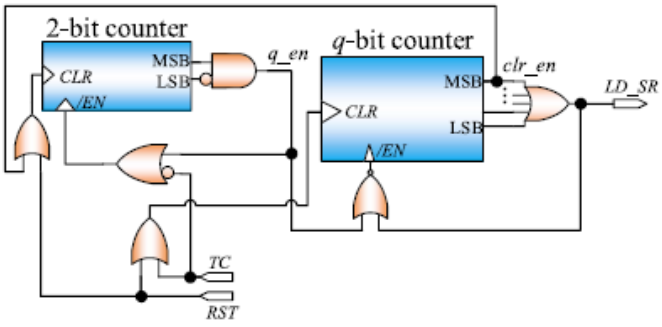

Fig. 4. Test key loading Controller [6]

This controller generates the output *LD_SR* for a time equal to the time required for loading the test key into the circuit. The test key loading controller schematic is shown in figure 4. It consists of q-bit and 2-bit counter. The value of q is determined by the number of SFFs that are selected for dynamic obfuscation. It is given by the expression $q=\log_2 N$. Both the counters are negative level triggered as shown in figure 4. When system RST is high, both counters' output is cleared to 0. The 2-bit counter will be enabled (and ON) only when system TC is high. Similarly, the q-bit counter will be disabled as long as *q_en* is low.

When TC becomes 1, the 2-bit counter starts to count. When the count reaches 10, *q_en* becomes 1 enabling the q-bit counter. Simultaneously, the enable signal for 2-bit counter becomes 1 and hence the 2-bit counter stops counting. Also, the CLR signal for the q-bit counter should become 0. Hence, TC is made 0 for the q-bit counter to get enabled and to ensure that key can be loaded into SR only during the switching of the circuit from functional mode to test mode.

Then, the q-bit counter counts for N clock cycles. The output *LD_SR* is high during this period. Hence, the test key is scanned through the SI port of the MUX into the SR. When the counter reaches N that is when the test key has been loaded, the *clr_en* becomes high, thus enabling the CLR signal of the 2-bit counter. As a result, the *q_en* bit goes low; disabling the q-bit counter and simultaneously *LD_SR* also becomes low. Also, the enable signal for the q-bit counter goes high stopping the count.

D. Obfuscating circuit with key interdependency block

Next, the key interdependency block for the circuit is designed. The key interdependency block is integrated with the circuit in accordance with figure5. An external key block is designed; whose outputs are the key bits to the remaining SFFs of the scan chain. The Primary Key (PK) bits are input to this block.

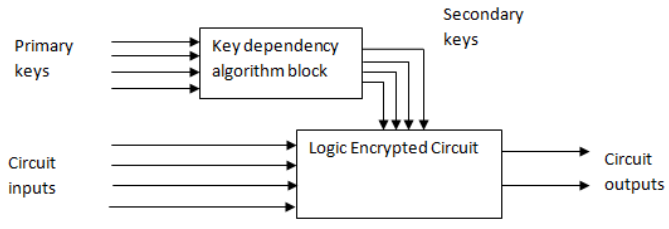

Fig. 5. Key Interdependency Block

In this block, an algorithm is used such that all the PKs are interlinked with each other. The key dependency block in figure 5 shown follows an algorithm in calculating the value of the output of the block. The Secondary Key (SK) bits are the output of this block. Each SK bit is related to many PK bits. The key bits in this case cannot be determined easily through algorithms such as path sensitization. The algorithm1 describes the key interdependency block procedure. The second input of the i^{th} gate in j^{th} stage is the output of the $((i - (2j - 1) + N)\ \%N)^{th}$ gate in the $(j-1)^{th}$ stage.

The output of this block is fed as key inputs to the gates. Whenever there is an incorrect PK bit, the number of key bits that will get affected in SK bits is more than one. To obtain a correct SK bit, multiple PK bits should be correct, since one SK bit depends on multiple PK bits.

As all of these primary key inputs are interlinked, there is more than one key-gate dependent on a single primary input. When one primary key input bit is changed, it affects all these key-gates.

TABLE II
SCAN-IN OF TEST VECTOR AND ITS EFFECT ON THE OTHER SIGNALS

TC	Clk	DFF1	DFF2	DFF3	Tc1	Tc3	Tc5	mismatch
	1	0	1	0	0	0	1	1
	2	0	0	1	0	1	0	1
0	3	1	0	0	1	1	1	1
	4	0	1	0	0	0	1	1
	5	0	0	1	0	1	0	1
1	6	1	0	0	1	1	1	1

Algorithm1. Generate key interdependency block

Input: Number of keys (K), Primary key values (PK).
Output: Secondary key (SK)

1. Read the number of keys and the values in PK.
2. for each stage (in s) do
3. for all values of K do
4. compute the 2^{nd} input to be given to the gate
5. if first stage
6. two inputs are K^{th} value of PK and computed value
7. else
8. the inputs are K^{th} value in $(s-1)^{th}$ stage and computed value in $(s-1)^{th}$ stage
9. end for
10. end for

Let us consider the example a circuit with 5 inputs and 2 outputs. The key gate location in the circuit is determined through fault impact parameter. The fault impact metric is computed for a set of randomized input patterns of the circuit. The expression is given by eq. (1). The key gate locations are identified and XOR gates are placed at these nets. To further strengthen the encryption, some of the XOR gates at the key locations are replaced by XNOR and an inverter. The circuit will give the correct output only for a particular key pattern. For the other patterns, one or more bits of the output will be corrupted. This algorithm will encrypt the circuit. For key size of 4, if the primary key (PK) varies from the correct PK by 1bit, then the number of changes in secondary key (SK) is more than one since through the key dependency block, a primary key will drive multiple secondary keys. As a result,

the input to a key gate in the circuit is dependent on multiple primary keys that are fed to the block. Next the scan chain of the circuit is obfuscated. This is done through dynamic obfuscation.

In the obfuscation of the scan chain, N blocking scan cells are to be paired with N DFFs of the shift register (SR). These N scan cells are selected by non-repetitive sequence P. Each element in the sequence P can vary from 1 to length of the scan chain (L).

The second sequence (T) determines which output of the DFF in the SR should be used for the obfuscation scheme of the scan data. Each element in T can have either 0 or 1. If the element in T has a value 1, then Q` in the DFF of SR is connected to the OR gate, else Q is used for obfuscation of scan data.

In the circuit, consider N=3, P= {1, 3, 5} and T= {0, 1, 1}. This implies that in the scan chain, the first, third and fifth scan flip-flops (SFFs) are chosen for the obfuscation scheme. The mode of operation of these SFFs is controlled by OR operation of TC and output of the DFF in SR. The sequence T implies that for first SFF, the OR gate will have inputs as TC and Q` of DFF1 in SR. Similarly for third and fifth SFFs, the inputs to the OR gate will be TC and Q of DFF2 and DFF3 in SR respectively.

Table II shows a scenario where the wrong test key is scanned and hence as subsequent key bits are fed through SI signal, the mismatch signal goes high indicating that the SI which is fed to the SR is not same as the sequence T. For this set of obfuscation parameters, the scan chain will propagate the correct input/output values only when the scan input (SI) of the circuit is 011, that is, when both SI and T match.

In the test mode, the third SFF will fetch the value from CUT instead of shifting the value from first SFF and hence there will be mismatch in outputs of obfuscated scheme circuit and the circuit without any encryption. When SI is 010, the least significant bit is loaded into the SR in the first cycle. Since mismatch signal goes high, the sequence loaded in SR will shift circularly 001 after the first cycle. In the second cycle, since *tc* port is one for second SFF, it fetches an unknown value x_0 from CUT instead of fetching from the previous SFF. This process continues till MSB of the SI is fed to the SR. Hence, the test vector applied will be a combination of unknown states from the CUT as well as bits from SI whereas the original test vector should have been 010. The values of these unknown states keep changing as internal states of SR keep changing every cycle. From the table II, it can also be seen that since the wrong value of SI is scanned into the scan chain, the mismatch signal remains high always and even though the SFF is in test mode (TC=0), the modes of one or more of the SFFs which are dynamically obfuscated change to 1, causing them to take fetch the data from the CUT instead of fetching from the previous SFF. This will result in corrupted output. The design does not work until both T and SI vectors have the same value.

IV. RESULTS AND DISCUSSION

The proposed method is evaluated on ISCAS'89 circuits. First, the net list has to be obtained for the CUT. For this, randomized set of 1000 patterns are fed to the CUT and the fault list is analyzed. This is done using the HOPE tool simulator. The fault impact of each net is obtained by running a python script on the fault list obtained through the HOPE simulator. The key gate locations are obtained and key gates are inserted. The key dependency block is now attached to the circuit and the inputs of key gates in the circuit are the outputs of the key dependency block. The scan design is then integrated with the circuit to obtain the overall design.

TABLE III
COMPARISON OF PROBABILITIES OF OBTAINING THE CORRECT KEY FOR EACH SCHEME

Key Bits	Dynamic Obfuscation	Key Interdependency	Proposed Scheme
4	0.0625	0.0625	0.0039
8	0.0039	0.0039	$\sim 10^{-5}$
16	1.5258E-5	1.5258E-5	2.328E-10
32	2.328E-10	2.328E-10	5.421E-20
64	5.421E-20	5.421E-20	2.9387E-39
128	2.9387E-39	2.9387E-39	8.6361E-78

For the dynamic obfuscation scheme, the probability of obtaining the correct test key pattern for N-bit test key is $1/2^N$. However, when the obfuscation is decoded, one trivial test key which the attacker will try can be test key sequence T in itself. Once the attacker feeds the scan input key to have same value as the test key sequence T, then the scan design is unlocked. In order to overcome this, the key dependency block which is integrated with the CUT can provide additional encryption to the design. Even when the attacker has decoded the scan design, the key interdependency algorithm will still encrypt the circuit. The attacker will have to enter the correct primary keys in order to obtain the correct secondary keys only after which the circuit will give correct outputs. In the key dependency scheme, when hill climbing attacks are performed, by flipping a key bit in primary side, number of secondary bits which will get affected is at least 20%. Flipping one primary key bit affects multiple secondary bits; hence the convergence towards the correct key is not possible in this attack.

Also, the hamming distance increases as the number of key bits which are incorrect increase. Hence the probability of obtaining the correct key with respect to the key interdependency block is very less since the output key bit does not depend on a single input key bit of the block. It depends on multiple primary keys. Hence, the probability of obtaining the correct primary key bits for the key interdependency block for N bit key pattern is $1/2^N$. The probability of the obtaining correct set of keys for the circuit obfuscated under the proposed method is further less than that of individual schemes.

Table III shows the probabilities of obtaining the test key in each scheme and for the proposed scheme. It can be seen that as the number of key bits increases, it is more difficult to guess the correct key bits for the circuit. The probability of obtaining the key bit for the proposed scheme is equal to square of the probability of obtaining the correct key for the individual schemes. The attacker will require more time to decode the correct bit key sequence since the correct test key has to be given to both the scan chain and the CUT in order to have correct functionality of the circuit. Hence, the proposed scheme thwarts the brute force attacks on the CUT.

Whenever incorrect scan design test key is given in test/functional mode, the corresponding SFF switches to

functional/test mode respectively and hence the output from the scan chain does not match with the ideal output of the circuit. The number of incorrect input bits will be from the SFF which is connected to the incorrect scan design test key bit till the length of the SFF chain. Hence, incorrect inputs are fed to the CUT and the output bits are corrupted.

Also, whenever the primary keys to the key dependency block are incorrect, then there is high corruption observed in the output bits. The hamming distance is observed to be high as the number of incorrect key bits to the dependency block increases. Through the key dependency block, the output corruption is more as compared to the output corruption observed in conventional logic encryption method. The conventional encryption method is same as the key dependency block based encryption method except that the key dependency block is absent in the conventional logic encryption method. As a result, the hamming distance in key dependency block based encryption is more compared to the conventional encryption for same sequence of incorrect key bits. In our proposed scheme, the hamming distance will depend on both the scan test key and the primary test key. Therefore, the percentage of output corruption is more in the proposed scheme than compared to the individual schemes of encryption due to increase in hamming distance.

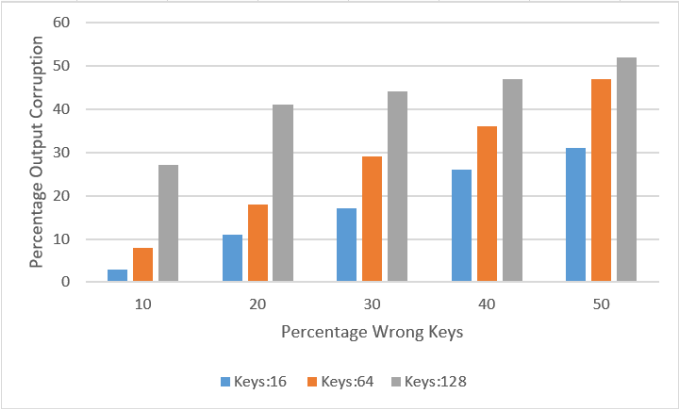

Fig. 6. Percentage output corruption Vs percentage of wrong key bits for different key-bit lengths

Figure 6 shows the variation of observed output from the ideal output as a measure of percentage of wrong key bits fed to the design for s1423 with key interdependency block added. It can be seen that as the percentage of wrong keys increases, the percentage of output bits that are corrupted also increase. This can be observed as the key size increases. The percentage change in output corruption is non-linear due to the high correlation that is present between the key bits of the key interdependency block. Due to this, the correct key cannot be obtained by flipping each key bit in successive attempts. Hence, the proposed method is robust against hill climbing attack wherein the attacker flips the each key bit in order to reduce the hamming distance between the correct key and the input key.

Figure 7 shows comparison of percentage of output corruption that is observed when wrong key bits are fed with and without the key interdependency block to the CUT for s1423 having 128 bit encryption. It can be observed that when the key interdependency block is introduced, then the output corruption is more than 20% but without the key interdependency block, the percentage is less than 20% only (15%). Hence, the output corruption, thereby the hamming distance also increases when key interdependency block is introduced in the design.

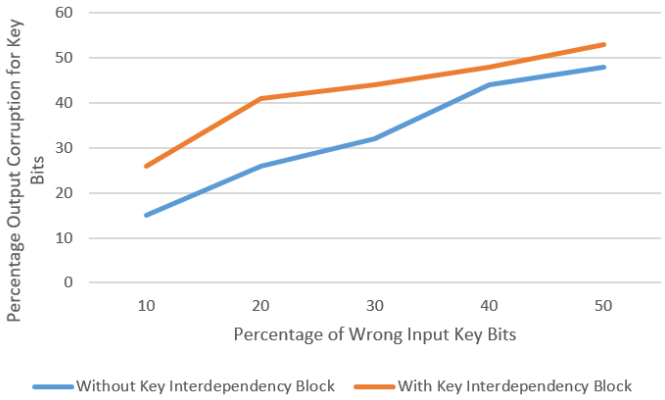

Fig. 7. Percentage output corruption Vs percentage of wrong key bits, with and without key interdependency block

The scan design method proposed by the other secure scan design methodologies has vulnerabilities such as in MKR [5], online testing is inhibited. In countermeasure based on mode-reset scenarios [9], if the attack is executed in test-mode only, then the mode-reset countermeasure fails. Also, the SDSFF used in [10] to overcome the test-mode only attacks fails when a fault occurs in the CUT. The proposed methodology for scan design approach overcomes these drawbacks faced by the other scan designs and is the resilient one in one or more figure of merit without any compromise in testability of the scan design.

V. CONCLUSION

In this work, the obfuscation of scan design and the key interdependency based logic encryption methods is analyzed and a combined approach is proposed to ensure that CUT is secure against side channel attacks as well as attacks on the CUT such as IP piracy, counterfeiting and over production.

In the proposed scheme, the probability that an adversary guesses the correct test key for both scan design and the key interdependency block is measured. This probability of guessing is very small since the correct test key must be entered at both the scan side and the logic encryption side to ensure that the functionality of the CUT is correct. Thus, this secure design prevents the attacker from getting any information of the design under test (CUT).

REFERENCES

[1] Chakraborty, Rajat Subhra, and Swarup Bhunia. "HARPOON: an obfuscation-based SoC design methodology for hardware protection." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 28.10 (2009): 1493-1502.

[2] Rajendran, Jeyavijayan, et al. "Fault analysis-based logic encryption." *IEEE Transactions on computers* 64.2 (2015): 410-424.

[3] Chandini, Bandarupalli, and M. Nirmala Devi. "Analysis of Circuits for Security Using Logic Encryption." *International Symposium on Security in Computing and Communication*. Springer, Singapore, 2018.

[4] Hely, David, et al. "Test control for secure scan designs." *European Test Symposium (ETS'05)*. IEEE, 2005.

[5] Yang, Bo, Kaijie Wu, and Ramesh Karri. "Secure scan: A design-for-test architecture for crypto chips." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 25.10 (2006): 2287-2293.

[6] Cui, Aijiao, Yanhui Luo, and Chip-Hong Chang. "Static and dynamic obfuscations of scan data against scan-based side-channel attacks." *IEEE Transactions on Information Forensics and Security* 12.2 (2017): 363-376.

[7] Karunakaran, Dinesh Kumar, and N. Mohankumar. "Malicious combinational hardware trojan detection by gate level characterization in 90nm technology." *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*. IEEE, 2014.

[8] Ali, Sk Subidh, et al. "Novel test-mode-only scan attack and countermeasure for compression-based scan architectures." *IEEE transactions on computer-aided design of integrated circuits and systems* 34.5 (2015): 808-821.

[9] Atobe, Yuta, et al. "Secure scan design with dynamically configurable connection." *2013 IEEE 19th Pacific Rim International Symposium on Dependable Computing*. IEEE, 2013.

[10] Atobe, Yuta, et al. "State dependent scan flip-flop with key-based configuration against scan-based side channel attack on RSA circuit." *2012 IEEE Asia Pacific Conference on Circuits and Systems*. IEEE, 2012.

[11] Roy, Jarrod A., Farinaz Koushanfar, and Igor L. Markov. "Ending piracy of integrated circuits." *Computer* 43.10 (2010): 30-38.

[12] Yasin, Muhammad, et al. "On improving the security of logic locking." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35.9 (2016): 1411-1424.

[13] Zhang, Jiliang. "A practical logic obfuscation technique for hardware security." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 24.3 (2016): 1193-1197.

[14] Karmakar, Rajit, et al. "A new logic encryption strategy ensuring key interdependency." *2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID)*. IEEE, 2017.

[15] Yang, Bo, Kaijie Wu, and Ramesh Karri. "Scan based side channel attack on dedicated hardware implementations of data encryption standard." *2004 International Conferce on Test*. IEEE, 2004