Activation of Logic Encrypted Chips: Pre-Test or Post-Test?

Muhammad Yasin[†], Samah Mohamed Saeed[§], Jeyavijayan (JV)[ξ] Rajendran and Ozgur Sinanoglu[‡]
yasin@nyu.edu[†], samahs@uw.edu[§], jv.ee@utdallas.edu[ξ], ozgursin@nyu.edu[‡]
† Electrical and Computer Engineering, NYU Tandon School of Engineering, NY, USA
§ Institute of Technology, University of Washington, Tacoma, WA, USA
ξ Erik Jonsson School of Engineering & Computer Science, The University of Texas at Dallas, TX, USA
‡ Electrical and Computer Engineering, New York University Abu Dhabi, Abu Dhabi, U.A.E.

***Abstract*—Logic encryption has been a popular defense against Intellectual Property (IP) piracy, hardware Trojans, reverse engineering, and IC overproduction. It protects a design from these threats by inserting key-gates that break the functionality when controlled by wrong keys. Researchers have taken multiple attempts in breaking logic encryption and leaking its secret key, while they also proposed difficult-to-break logic encryption techniques. Mainly, state-of-the-art logic encryption techniques pursue two different models that differ in when the manufactured chips are activated by loading the secret key on the chip's memory: activation prior to manufacturing test (pre-test) versus subsequent to manufacturing test (post-test).**

In this paper, we shed light on the interaction between manufacturing test and logic encryption. We assess and compare the pre-test and post-test activation models not only in terms of the impact of logic encryption on test parameters such as fault coverage, test pattern count and test power consumption, but also in terms of the impact of manufacturing test on the security of logic encryption. We outline a test data mining attack that can successfully determine the logic encryption key of a pre-test activated chip by utilizing the test data.

I. Introduction

The high complexity and fabrication cost of the Integrated Circuits (ICs) necessitate the globalization of design and manufacturing flow [1]. Intellectual property (IP) cores can be obtained from a third party IP provider to reduce the design efforts of an IC designer significantly. Also, high fabrication cost pushes many companies to outsource their fabrication to third party foundries. A foundry can then outsource test or assembly services to OSAT companies (Outsourced Semiconductor Assembly and Test) [2]. This globalization renders the ICs prone to various security threats. The adversary can be the end-user of the chip, or a rogue agent in the (untrusted) foundry, or in the test facility. He/she can apply different kinds of attacks such as reverse engineering, hardware Trojan insertion [3], IC piracy, and IP piracy [4].

Logic encryption thwarts IP piracy and reverse engineering attacks by locking/encrypting a design with a secret key [4]–[9]. To enable chip-locking features, additional key-gates, e.g. XOR/XNORs, are inserted into the original netlist. One input of each key-gate is driven by a key bit that is supplied from an on-chip secure memory while the other input is the functional net. An encrypted IC will not be functional unless it is activated using the correct key.

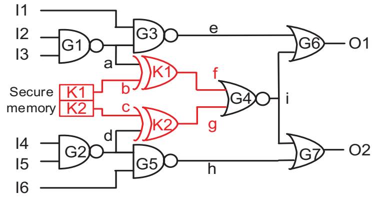

Fig. 1: A netlist encrypted with two key-gates [10].

Each fabricated chip goes through a manufacturing test that screens out the defective chips. Design for Testability (DfT) engineers target generating test patterns that maximize the fault coverage, minimize test pattern count, and reduce test power consumption [11].

State of the art logic encryption frameworks pursue two different activation models, pre-test and post-test activation, that differ in the time of activation of an IC with respect to the manufacturing test. The two models are illustrated in Figure 2 and highlighted in red and green color, respectively.

Pre-test activation. The ICs are activated prior to the manufacturing test, typically conducted in the foundry or outsourced to an OSAT. Since the IP owner does not want to reveal the secret key to the untrusted foundry, on-chip public key cryptographic infrastructure [4] is used to load the secret key securely on the chip. On passing the manufacturing test, the ICs are shipped for assembly/sales directly from the foundry, which is useful in meeting time-to-market constraints.

Post-test activation. The ICs are activated after performing the manufacturing test. Either *remote activation* [12] or *in-house activation* [8] can be employed. In-house activation requires shipping of the locked IC from the foundry to the trusted facility, eliminating the need for on-chip cryptography.

Evolving business and threat models. Fabless and fab-lite are the evolving business models for semiconductor companies [13]. Fabless companies, such as Apple Inc., outsource IC fabrication (to Samsung and TSMC [14]), testing and assembly services. Fab-lite companies such as TI [13] may outsource IC fabrication (to SMIC [15]) and testing, but conduct packaging and assembly in-house. Given the above business models, Apple may activate the ICs remotely using pre-test activation, and TI may activate the ICs in-house using post-test activation. However, the test and security implications of these scenarios have never been studied.

While Samsung fabricates ICs for Apple, it is a direct competitor to Apple in the smartphone market. Such complex interaction and varying levels of trust among the companies necessitate a re-evaluation of the security threats. In this paper, we focus on the security implications of the manufacturing test. The attacker is an agent in the foundry or the test facility who tries to exploit the test data information to break logic encryption. The specific contributions of the paper are:

1) We present a comprehensive analysis of pre-test versus post-test activation in terms of the impact of logic encryption on the test parameters and the impact of manufacturing test on the security of logic encryption.
2) We highlight the security vulnerabilities of pre-test activation. We develop a *test-data mining attack* that can reveal the secret key by analyzing the manufacturing test-data.

978-3-9815370-6-2/DATE16/©2016 EDAA

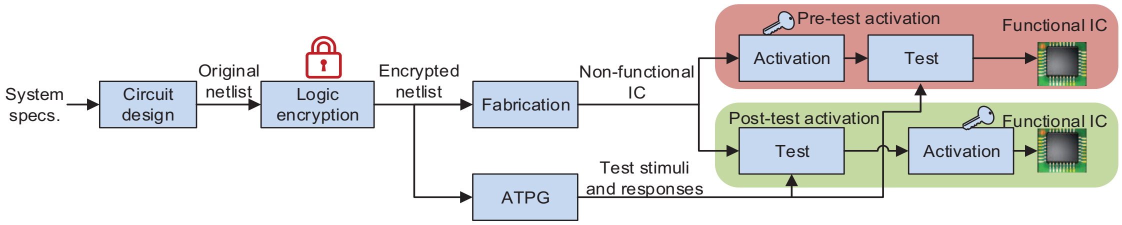

Fig. 2: Logic encryption and IC activation in the IC design flow. Pre-test activation and post-test activation models.

Our proposed attack is equally effective against existing logic encryption techniques.

3) We show that post-test activation provides a secure test environment and improved test quality. The fault coverage for post-test activation is higher, with only a slight increase in test pattern count and power dissipation, compared to pre-test activation.

II. Related Work

A. Logic encryption techniques

Existing logic encryption techniques can be classified as:

- Random logic encryption (RLE) [4] inserts XOR/XNOR key gates at random locations in a netlist.
- Fault analysis based logic encryption (FLE) [5] addresses the limitations of RLE and encrypts an IC such that a random incorrect key corrupts as many outputs as possible.
- Strong logic encryption (SLE) [10] inserts key gates that strongly interfere with each other, and it becomes difficult to sensitize key bits to primary outputs on individual basis.

B. Attacks against logic encryption

EPIC [4], the first logic encryption framework, addressed the issue of overproduction of ICs by introducing a functional locking block that locks the IC with a secret key. The key is loaded onto an IC using public key cryptography infrastructure. One of the limitations of EPIC is that it gives the IP owner little control over the actual number of ICs produced and sold. This concern was addressed by CSST, a framework that incorporates scan locking in addition to functional locking [6], [12]. Both EPIC and CSST require a complex communication infrastructure that occupies a significant area on the chip.

Table I presents a summary of existing attacks against logic encryption and the corresponding threat models. Attacks that exploit the algorithmic weaknesses of logic encryption [7], [10] require access to a reverse engineered netlist and fully functional IC (or a large set of I/O pairs).

The attack in [10] generates key-leaking input patterns by analyzing the encrypted netlist. These patterns are applied to the functional IC to sensitize the key bits to the primary outputs. Another attack [7] uses Boolean satisfiability techniques to prune the incorrect key candidates. The hill climbing search based attack [8] uses test data information to guess the secret key for pre-test activated ICs. The attack tries to achieve zero Hamming distance HD_O between the test response and the encrypted circuit, for multiple random key guesses. The individual bits in the initial key guess are flipped if the flip minimizes the Hamming distance HD_O.

Our proposed test-data mining attack (refer to Section III-C) is both orthogonal and complementary to the attacks in [10] and [7]. These attacks collect a large number of I/O pairs from a functional chip already deployed. The proposed attack does not require access to a functional IC and operates on a small subset of I/O pairs, i.e., the test-data. Compared to the hill climbing search based attack [8], the proposed attack does not need multiple restarts since it is guided by fault coverage, which is a test-relevant metric. Moreover, our attack is equally effective against existing logic encryption techniques, as will be demonstrated in Section V.

III. Pre-test Activation

In pre-test activation, the secret key is loaded onto the IC prior to the manufacturing test. The manufacturing test can be conducted in the foundry or a separate test facility (OSAT [2]). Since an IP owner wants to protect the secret key from being exposed to either the foundry or the OSAT, he can load the secret key securely on the chip using public key cryptography infrastructure. Such infrastructure can incur significant area overhead [4].

As the test is to be conducted with the key in place, the secret key values are applied as constraints on the key inputs during the test generation phase, which can impact the test quality and costs, as well as the security of logic encryption.

A. Threat model

The attacker is a person in the foundry or test facility with access to the following:

1) An encrypted netlist E_K, which can be obtained by reverse engineering [16] or IP piracy [17].
2) Test stimuli T and responses Γ.

B. Impact on test

Key gates added to a design to implement logic encryption introduce new faults on the key lines and the output of the key gates, and can affect testability of the faults in the original design. Pre-test activation can affect the testability of the design as follows:

Faults on the key lines. Certain faults on the key lines cannot be activated, because of the constraints imposed by key value, and remain undetected. For the netlist in Figure 1, the correct value for both key bits K1 and K2 is 0. With these values applied as constraints during ATPG, the stuck-at-0 faults b/0[1] and c/0 cannot be activated.

Faults on the key gate outputs. The observability of the faults on the key gate outputs depends on the observability of the original net where the key gate is inserted. The faults on the key gate outputs may remain untestable if the faults on the original net are difficult-to-observe. In Figure 1, the fault g/1 can be activated, but it cannot be propagated to a primary output since K1=1 is required for its propagation.

Faults in the original design. The controllability of the nets in the logic cone driven by a key gate changes when the key gate is driven by a key value of 1, configuring the key gate into an inverter. This change in controllability can affect the testability of certain faults in the original design.

[1] Faults are represented as net/stuck-at value; e.g., c/0 represents a stuck-at-0 fault on net c.

TABLE I: A comparison of the threat models of the attacks against logic encryption.

Study	**Attacker location**	**Information /assets available to the attacker**	**Attack method**	**Attacker's objective**	**Proposed defense**
CSST [6]	Foundry	Activated ICs that need to be tested	Falsely claim defect-free ICs to be defective and withhold them	IC over-production	Scan locking block
Rajendran et al. [10]	Foundry / end-user	1) Encrypted netlist 2) Activated IC	Sensitization of key bits to outputs	IP Piracy	Strong logic encryption
Subramanyan et al. [7]	Foundry / end-user	1) Encrypted netlist 2) Activated IC	SAT-based algorithm that rules out incorrect keys iteratively	IP Piracy	None
Plaza and Markov [8]	Foundry / test facility	1) Encrypted netlist 2) Test patterns and responses	Start with a random key CK. Flip the bits in CK based on the Hamming distance HD_O	IP Piracy	Test-aware combinational logic locking
Proposed attack	Foundry / test facility	1) Encrypted netlist 2) Test patterns and responses	Find the key that maximizes fault coverage and produces correct output for the test patterns	IP Piracy	Post-test activation

Overall, pre-test activation reduces the fault coverage of a design. For example, in Figure 1, the fault coverage for the pre-test activated netlist is 82.43% as compared to 91.12% for the original (unencrypted) netlist. The undetectable faults in the netlist are a/1, b/0, c/0, d/1, f/1, g/1, and i/0. Eight test patterns are generated by the ATPG tool as listed in Table II. As the key values are constant throughout the entire test, the impact of pre-test activated logic encryption on switching activity, and thus, test power consumption, is negligible.

TABLE II: Test patterns (Pre-test activation) for the netlist in Figure 1. The correct key K_{corr} is used as a constraint during ATPG.

Key(K_{corr})	Stimulus (T)	Response (Γ)
00	011001	10
00	101010	01
00	101111	01
00	011101	10
00	111010	11
00	000111	11
00	110001	00
00	001011	10

C. Impact on security

To highlight the security vulnerabilities of pre-test activation, we develop a *test-data mining attack* that can reveal the secret key used in pre-test activation of logic encryption.

Attack methodology. During the test pattern generation phase, a DfT engineer will apply the correct key K_{corr} as a constraint and obtain a set of test patterns that maximizes fault coverage.

An attacker can therefore apply the test stimuli as input constraints, the test responses as output constraints, and search for the potentially correct key K_P that maximizes the fault coverage under the specified constraints. The attack is an optimization problem: the objective is to maximize the fault coverage FC under the test stimulus T and test response Γ constraints, as follows:

$$\begin{aligned} &\text{maximize} \quad FC \\ &\text{subject to} \qquad \underset{1\leq i\leq N}{\forall} E_K(K_P, T_i) = \Gamma_i \\ &\text{solve for} \quad K_P \end{aligned} \tag{1}$$

The rationale for the attack to return the correct key is that the test patterns have been generated with the objective of maximizing the fault coverage in the presence of the correct key as a constraint. When the same set of test patterns are used as constraints, the key that maximizes the fault coverage will be the one that is used to generate the patterns.

Equation 1 formulates a system of Boolean equations which can be solved using techniques such as Boolean satisfiability (SAT) or Integer Linear Programming (ILP) [18]. Test generation (ATPG) algorithms are capable of solving a system of Boolean equations while maximizing fault coverage at the same time; ATPG is, therefore, a natural candidate for solving the optimization problem in Equation 1. The complexity of the attack is NP-hard [19].

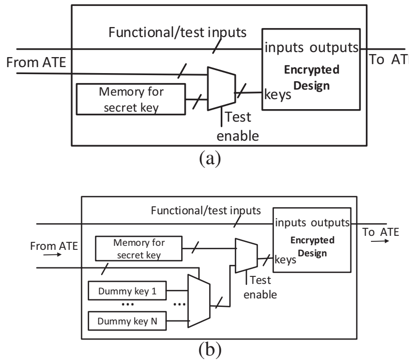

Fig. 3: Methods for selecting key values in post-test activation. (a) $Post_U$: The key values are part of the test patterns. (b) $Post_C$: A few dummy key values are hardcoded in memory.

Let us consider the netlist shown in Figure 1. When the correct key value $K_{corr} = 00$ is used as a constraint, eight test patterns are generated by the ATPG tool as listed in Table II. An attacker will launch the attack described in Equation 1 by applying the test stimuli and responses as constraints, and search for the potential key K_P that maximizes the fault coverage. The only key K_P that maximizes the fault coverage and satisfies these test pattern constraints is 00, and the corresponding fault coverage is 82.43%. None of the other key values satisfies the test pattern constraints.

IV. Post-test Activation

In post-test activation, the manufacturing test is conducted on a locked IC with *the rationale that manufacturing test is a "structural" test, and that the chip need not be functional during the test*. The IC can be activated post-test in one of the following ways:

1) After manufacturing test, defect-free ICs are shipped to a trusted facility, activated by the IP owner, and shipped out for sale.
2) Tested ICs can also be activated remotely, similar to the case of pre-test activation, via public key cryptography infrastructure [6].

A. Impact on test

During the test pattern generation phase, the values for the key inputs can be generated in two different ways:

Test pattern generation without key constraints ($\boldsymbol{Post_U}$). Maximum fault coverage can be obtained if there are no constraints on the key values during ATPG. During the test, the key inputs are treated similarly to the primary inputs and set to the values freely generated by the ATPG. Select logic can be added to connect the key inputs to the Automatic Test Equipment (ATE) during the test and bypass the memory that has to hold the secret key. Figure 3(a) illustrates $Post_U$. In the normal mode, the key lines are driven by the secret key bits (which are loaded post-test), whereas, in the test mode, the key lines are driven by the ATE.

Test pattern generation with key constraints ($\boldsymbol{Post_C}$). The test is conducted using dummy key values. During the test, a few dummy key values can be hardcoded or loaded to the on-chip memory. A key value can be selected using additional multiplexer circuitry, as shown in Figure 3(b). During ATPG, each key value is treated as a constraint, and a set of test patterns is generated with it.

Post-test activation affects the testability of a design:

Faults on key lines and key gate outputs. Activation of the newly introduced faults around the key gates is guaranteed as long as:

- the key-line is either fully controllable, as in case of $Post_U$,
- or the dummy keys cumulatively bring both 0 and 1 to every key line. At least two dummy keys (e.g., all-0 and all-1 keys) are required to achieve this objective.

As an example, the faults b/0 and c/0 that were untestable in pre-test activated netlist become testable in $Post_U$. For $Post_C$, we assume DfT support for two bit-wise complementary dummy key values, 10 and 01. With the first key value 10, the faults b/1 and c/0 on the key lines cannot be detected. The key value 01, however, enables the generation of a test pattern to detect these faults.

Faults in the original design. As the key lines are not constant, the controllability of the nets in the logic cone driven by a key gate is improved. The test generation process can selectively set the key line values to find test patterns for difficult-to-detect faults, and improve the fault coverage. For the netlist in Figure 1, the fault coverage is 100% for $Post_U$. The ATPG tool generates 9 test patterns. Certain faults, such as d/1, that were undetectable for both the original design as well as the pre-test activated design become detectable for the post-test activated design. For $Post_C$, the benefit may be limited since the detection of certain faults may require multiple key bits to have specific values simultaneously. The faults may remain undetected when none of the dummy keys satisfy the required conditions. Using the two dummy key values, we achieve a fault coverage of 100% for the same example. In this case, the ATPG tool generates 12 test patterns.

To summarize, the key gates act as test points [11] that can be beneficially utilized by the test generation tool to improve fault coverage. The overall impact of logic encryption on the testability of the design depends on the design topology, location of the key gates, and the activation method, $Post_U$ or $Post_C$. Moreover, testing certain faults may be redundant from a test perspective as those faults may have no impact on the outputs when the secret key is loaded.

TABLE IV: Execution time of the attacks in seconds. “Hill” refers to the hill climbing attack [8]. “Prop” refers to the proposed attack. The hill climbing attack stops execution when the attack is successful or upon 100,000 initial random keys.

	RLE [4]		FLE [5]		SLE [10]	
Benchmark	Hill	Prop	Hill	Prop	Hill	Prop
s298	488.6	1.7	1.5	1.4	6.9E4	2.6
s400	168.9	3.8	3.2	2.1	2161.1	2.7
s444	155.9	2.5	0.2	1.7	5.2E4	4.2
s713	81.6	4.2	1.5	6.3	9.1E4	6
c432	101.3	10.5	2.7	9.8	1.1E5	10.3
s5378	2.5	546.5	11.0	1335.7	3.3E5	252.8
c5315	9.2	196.6	13.9	195.8	2253.8	153.6
c7552	179.5	261.1	15.4	223	2.4E5	60.6
s9234	33.4	1219.2	10.4	3058.8	4.6E5	915.7
s13207	187.2	2.8E4	15.8	4.7E4	2.2E4	6596.6
s15850	355.6	2.4E4	16.7	8.1E4	3.8E5	6601.5

In $Post_C$, the key values are constant during the scan shift operations and may change during the capture operations. As the scan shift operations dominate the test process, the overall impact of $Post_C$ on test power is expected to be quite small. However, in $Post_U$, the key values are part of the test patterns and increase test power. Section V-C provides an analysis of the impact of activation models on test power.

B. Impact on security

In post-test activation, both test pattern generation and manufacturing test are conducted in the absence of the secret logic encryption key. Regardless of the method, $Post_U$ or $Post_C$, used for test pattern generation, the key bit values are set independent of the secret key. In $Post_U$, each test pattern can contain an arbitrary key value as assigned by the ATPG tool, whereas in $Post_C$ a few key values can be chosen randomly. Since the secret key is not included in test pattern generation process, the test data embeds no information that can be used to infer the secret key.

Any analysis performed by the attacker will only reveal these arbitrary key values and not the secret key value. Therefore, post-test activation has no detrimental impact on the security of logic encryption.

V. Experimental Results

A. Experimental setup

In this section, we present the results of the proposed attack and the impact of activation models on test parameters. We run our experiments on ISCAS85 and ISCAS89 benchmark circuits encrypted with 32 key gates. We apply our analysis to three different logic encryption techniques: random (RLE) [4], fault analysis based (FLE) [5], and strong logic encryption (SLE) [10]. We utilize Tetramax ATPG [20] to generate the test patterns for different activation models and determine the key value that maximizes the fault coverage. We also implement the hill climbing attack [8].

TABLE III: Attack success for different logic encryption techniques. As both the hill climbing attack [8] and the proposed attack are successful against all RLE [4] and FLE [5] circuits, attack results are shown on SLE [10] circuits alone.

Benchmark	s298	s400	s444	s713	c432	s5378	c5315	c7552	s9234	s13207	s15850
The hill climbing attack [8]	No	Yes	Yes	No	No	No	Yes	No	No	Yes	No
Proposed attack	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

B. Security analysis of pre-test activation

First, we demonstrate how pre-test activation renders logic encryption vulnerable to attacks. Table III reports the results of the proposed attack and the hill climbing attack [8]. In the hill climbing attack, $100,000$ random initial keys are tried. The number of inputs applied during the attack is $100,000 \times N_K \times N_P$, where N_K is the number of key bits and N_P is the number of test patterns.

The hill climbing attack [8] breaks all RLE and FLE circuits, but it is not effective against SLE circuits. In comparison, the proposed attack is 100% successful and breaks all the circuits across all the logic encryption techniques. In SLE, the key gates are localized in the logic cones of a few outputs and interfere strongly with each other, rendering Hamming distance a poor metric to guide the local search based attack in [8]. Table IV reports the execution time. The execution time of the proposed attack is dependent upon the logic encryption technique and the number of test patterns. The execution time of FLE is the highest overall as it inserts key gates at the locations that affect the largest number of output bits. In comparison to the proposed attack, the hill climbing attack is the most effective against FLE and can retrieve the secret key within few seconds.

C. Impact on test

In this section, we evaluate the impact of different activation models on test quality and costs. We consider the original design and the following three scenarios for the encrypted designs: $Post_U$, $Post_C$, and Pre, where Pre denotes pre-test activation. We report the fault coverage, test pattern count, and switching activity for different encryption techniques.

Table V shows that $Post_U$, as a result of full control over the key inputs, attains the highest fault coverage for most of the benchmark circuits across all logic encryption techniques. In some cases, $Post_U$ can achieve a fault coverage higher than that of the original circuit, by detecting certain faults that were untestable in the original circuit. $Post_C$, which uses only two dummy keys to illustrate the case with the least demanding DfT support, achieves a fault coverage level comparable to $Post_U$. Pre achieves the lowest fault coverage, as the constraints imposed by the key values render certain faults untestable.

The test pattern count for both $Post_U$ and $Post_C$ is slightly larger than that for Pre in most of the circuits, as shown in Table VI. This is expected, as the fault coverage for Pre is also the lowest. For $Post_U$ activation model, the test pattern count is, on average, the smallest for FLE. FLE inserts key-gates that influence the largest number of outputs, which increases the number of faults detected per pattern.

Table VII shows the percentage switching activity for the encrypted designs, modeling power consumption during the test. In general, the activation models differ only slightly in terms of the switching activity. The switching activity for $Post_U$ is the highest, on average, as the key values are not constant during scan shift operations. Pre imposes the maximal constraints on the ATPG tool, leading to a lower number of transitions in the test patterns. The switching activity is, on average, the lowest in SLE as the key gates are localized in certain parts of the circuit and do not impact most of the circuit outputs.

VI. DISCUSSION

Attack using less information. In our attack, we assumed that the test patterns were generated with the objective of achieving 100% fault coverage. Our attack shows that this benefits the attacker. Hence, one would like to provide less information to the attacker. One way is to reduce the target fault coverage, which will eventually reduce the set of test patterns and responses given to the attacker. To gain insights into how much information the attack needs to be successful, we executed the attack with only a partial set of test vectors. Table VIII shows the attack results for two SLE circuits: c7552 and s9234. In the s9234 circuit, the key can be recovered

TABLE V: Fault coverage for different logic encryption techniques and three activation models.

		RLE [4]			FLE [5]			SLE [10]		
Benchmark	Original circuit	$Post_U$	$Post_C$	Pre	$Post_U$	$Post_C$	Pre	$Post_U$	$Post_C$	Pre
s298	99.47	99.78	98.85	94.68	99.57	99.57	94.96	99.57	97.99	94.96
s400	100	100	99.88	96.09	100	100	96.20	100	99.41	96.20
s444	100	100	100	96.20	100	100	96.20	100	99.41	96.20
s713	100	100	99.90	96.70	100	99.90	96.70	100	99.70	96.80
c432	99.57	100	99.62	96.61	100	99.81	96.42	99.62	99.62	96.61
s5378	99.95	99.95	99.95	99.41	99.95	99.93	99.40	99.95	99.83	99.41
c5315	99.90	99.90	99.90	99.39	99.90	99.90	99.42	99.90	99.90	99.42
c7552	99.39	99.58	99.58	98.90	99.40	99.40	98.92	99.71	99.71	98.92
s9234	99.95	99.95	99.95	99.56	99.96	99.93	99.55	99.95	99.59	99.56
s13207	97.97	98.04	98.04	97.86	98.28	98.28	97.86	98.22	97.74	97.68
s15850	93.69	93.84	93.98	93.81	94.18	94.23	93.57	93.41	93.38	93.32

TABLE VI: Test pattern count for different logic encryption techniques and three activation models.

		% increase for RLE [4]			% increase for FLE [5]			% increase for SLE [10]		
Benchmark	Original circuit	$Post_U$	$Post_C$	Pre	$Post_U$	$Post_C$	Pre	$Post_U$	$Post_C$	Pre
s298	30	43.3	13.3	-13.3	6.7	10.0	6.7	26.7	36.7	16.7
s400	36	19.4	19.4	-2.8	13.9	-11.1	-13.9	11.1	13.9	0.0
s444	37	8.1	29.7	5.4	2.7	2.7	0.0	16.2	16.2	0.0
s713	58	13.8	13.8	0.0	3.4	0.0	-1.7	12.1	10.3	-5.2
c432	62	33.9	30.6	8.1	25.8	8.1	4.8	35.5	41.9	25.8
s5378	249	-2.0	5.2	0.0	-6.8	1.6	0.8	-3.2	2.0	-1.6
c5315	114	14.0	21.1	12.3	11.4	6.1	3.5	10.5	7.9	0.9
c7552	168	-1.2	1.2	-7.1	-4.8	0.0	-1.2	10.7	10.1	-1.2
s9234	329	4.3	3.3	-1.2	-4.0	3.0	2.1	0.3	0.9	0.3
s13207	506	-1.2	2.6	0.6	-2.8	4.3	1.0	7.1	2.2	1.0
s15850	524	-1.5	1.7	-1.0	-2.7	3.4	0.2	0.2	1.7	0.6

TABLE VII: Percentage switching activity during test for different logic encryption techniques and three activation models.

		% increase for RLE [4]			% increase for FLE [5]			% increase for SLE [10]		
Benchmark	Original circuit	$Post_U$	$Post_C$	Pre	$Post_U$	$Post_C$	Pre	$Post_U$	$Post_C$	Pre
s298	3.8	8.9	4.4	1.8	10.5	-7.4	-6.3	-7.8	-1.7	-0.9
s400	5.4	-1.5	7.6	6.0	7.2	-0.4	0.0	-2.9	-2.1	-2.6
s444	5.4	-4.1	0.0	-1.6	12.3	-3.4	-3.0	-5.3	-4.6	-6.9
s713	11.1	1.9	1.3	0.5	3.7	0.6	0.7	0.9	0.0	0.8
c432	5.1	-4.6	-1.8	-0.5	1.3	0.4	-0.2	1.9	0.0	0.0
s5378	49.6	-0.4	-1.1	-1.1	2.1	-0.4	-0.4	-0.5	-0.7	-0.6
c5315	33.0	1.6	0.2	-0.1	6.8	-1.7	-2.0	2.0	-0.3	-0.9
c7552	36.1	0.3	-0.2	-0.3	2.8	-0.3	-0.4	-0.9	0.2	-0.4
s9234	53.3	0.0	-0.7	-0.6	-0.3	-0.3	-0.3	-0.2	-0.3	-0.2
s13207	44.7	1.2	-2.5	-0.6	2.8	-4.2	-1.0	-6.6	-2.1	-1.0
s15850	49.6	1.6	-1.7	1.0	2.7	-3.3	-0.2	-0.2	-1.7	-0.6

TABLE VIII: Attack success on SLE for partial set of test vectors.

	Percentage of test vectors		
Benchmark	10%	20%	30%
c7552	No	No	Yes
s9234	Yes	Yes	Yes

by using as few as 10% of the test vectors. In the c7552 circuit, however, 30% or more of the test vectors are required to recover the key. An attack using a smaller percentage of test patterns also leads to a reduction in the execution time.

Impact of don't cares. Redundancy in test data in the form of don't care values (X's) can be exploited to protect against the proposed attack. Specifying the X's judiciously to de-sensitize key bits on the outputs can create test data that leads to less effective attacks. The resulting test vector and response pairs would fail to prune the key search space, providing resilience against the proposed attack.

Defense against the proposed attack.Using the post-test activation model with the proposed DfT support, it becomes possible to test an encrypted IC securely with minimum overhead. The test patterns used in post-test activation do not reveal secret key information. Although we show that pre-test activation model is vulnerable to the proposed test data mining attack, our future work will investigate how to secure pre-test activation model against the proposed attack.

VII. CONCLUSION

In this paper, we analyze and compare the impact of pre-test and post-test activation of logic encryption on test quality and security. We investigate how logic encryption affects the testability of a design and illustrate how the constraints imposed by the secret key deteriorate the fault coverage in logic encrypted chips with pre-test activation. Post-test activation, however, offers freedom in controlling the key inputs judiciously to improve the fault coverage and enhance test quality with minimal DfT support. On the security analysis front, we propose a test data mining attack that exploits the information in test data and circumvents the logic encryption in pre-test activation with 100% success rate. The attack uses common DfT tools and requires only a fraction of test data, unlike the previous attacks that require a functional chip. This work should alert the companies about the security liabilities of using pre-test activation as part of their business model.

VIII. ACKNOWLEDGEMENT

This work was supported in part by the Semiconductor Research Corporation and Advanced Technology Investment Company (SRC/ATIC) under Grant 2013-HJ-2440-S4; the New York University/New York University Abu Dhabi (NYU/NYUAD) Center for Research in Information Security Studies and Privacy (CRISSP); and the National Science Foundation, Computing and Communication Foundations (NSF/CCF) under Grant 1319841.

REFERENCES

[1] Defense Science Board, “Defense Science Board (DSB) study on High Performance Microchip Supply,” 2005. [Online]. Available: http://www.acq.osd.mil/dsb/reports/ADA435563.pdf

[2] B. Wire, “Research and Markets: Outsourced Semiconductor Assembly and Test Market (OSAT) Trends,” 2014, [Aug 22, 2015]. [Online]. Available: www.businesswire.com/news/home/20140324005628/en/Research-Markets-Outsourced-Semiconductor-Assembly-Test-Market

[3] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, “Trustworthy Hardware: Identifying and Classifying Hardware Trojans,” *Computer*, vol. 43, no. 10, pp. 39–46.

[4] J. Roy, F. Koushanfar, and I. Markov, “EPIC: Ending Piracy of Integrated Circuits,” in *Proc. Design, Automation and Test in Europe*, 2008, pp. 1069–1074.

[5] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, “Logic Encryption: A Fault Analysis Perspective,” in *Proc. Design, Automation Test in Europe*, 2012, pp. 953–958.

[6] M. T. Rahman, D. Forte, Q. Shi, G. K. Contreras, and M. M. Tehranipoor, “”CSST: preventing distribution of unlicensed and rejected ics by untrusted foundry and assembly”,” in *Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, 2014, pp. 46–51.

[7] P. Subramanyan, S. Ray, and S. Malik, “Evaluating the Security of Logic Encryption Algorithms,” in *IEEE International Symposium on Hardware Oriented Security and Trust*, 2015, pp. 137–143.

[8] S. M. Plaza and I. L. Markov, “Protecting Integrated Circuits from Piracy with Test-aware Logic Locking,” in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2014, pp. 262–269.

[9] M. Yasin, J. Rajendran, O. Sinanoglu, and R. Karri, *IEEE Trans. on CAD of Integrated Circuits and Systems*, 2016, to be published.

[10] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, “Security Analysis of Logic Obfuscation,” in *Proc. IEEE/ACM Design Automation Conference*, 2012, pp. 83–89.

[11] M. Bushnell and V. D. Agrawal, *Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits*. Springer Science & Business Media, 2000, vol. 17.

[12] G. Contreras, M. Rahman, and M. Tehranipoor, “Secure Split-Test for Preventing IC Piracy by Untrusted Foundry and Assembly,” in *Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, 2013, pp. 196–203.

[13] P. McLellan, “A Brief History of the Foundry Industry, part 2,” 2013, [Sep 1, 2015]. [Online]. Available: https://www.semiwiki.com/forum/content/2109-brief-history-foundry-industry-part-2-a.html

[14] AppleInsider, “Samsung reportedly nabs 75% of Apple's next-gen 'A9' SoC orders,” 2015, [Aug 10, 2015]. [Online]. Available: http://appleinsider.com/articles/15/01/26/samsung-to-reportedly-take-75-of-apples-next-gen-a9-soc-orders

[15] S. P. Releases, “SMICs Beijing Fab Wins TI Quality Excellence Award,” 2014, [Aug 10, 2015]. [Online]. Available: http://www.smics.com/eng/press/press_releases_details.php?id=107870

[16] R. Torrance and D. James, “The State-of-the-Art in Semiconductor Reverse Engineering,” in *Proc. IEEE/ACM Design Automation Conference*, 2011, pp. 333–338.

[17] M. Rostami, F. Koushanfar, and R. Karri, “A Primer on Hardware Security: Models, Methods, and Metrics,” *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1283–1295, 2014.

[18] E. Clarke, A. Gupta, J. Kukula, and O. Strichman, “SAT Based Abstraction-Refinement Using ILP and Machine Learning Techniques,” in *Computer Aided Verification*. Springer, 2002, pp. 265–279.

[19] B. Krishnamurthy and S. B. Akers, “On the Complexity of Estimating the Size of a Test Set,” *IEEE Trans. Comput.*, vol. 33, no. 8, 1984.

[20] S. U. Manual, “TetraMAX ATPG User Guide,” *Version X-2005.09*, pp. 249–264, 2005.