A Low-Cost Solution for Protecting IPs Against Scan-Based Side-Channel Attacks

Jeremy Lee, Mohammad Tehranipoor, and Jim Plusquellic
Department of CSEE
University of Maryland Baltimore County
{jlee36,tehrani,plusquel}@umbc.edu

***Abstract*— Scan designs used for testing also provide an easily accessible port for hacking. In this paper, we present a new low-cost secure scan design that is effective against scan-based side-channel attacks. By integrating a test key into test vectors that are scanned into the chip, testing and accessing scan chains are guaranteed to be allowed only by an authorized user. Any attempt to use the scan chain without a verified test vector will result in a randomized output preventing potential side-channel attacks. The proposed technique has a negligible area overhead, has no negative impact on chip performance, and places several levels of security over the scan chain protecting it from potential attacks.**

I. INTRODUCTION

Design for testability (DFT) has greatly eased manufacturing testing with an efficient method of yielding a high fault coverage. Scan test has contributed greatly to the success of DFT, but recently it has also been used to assist in non-invasive attacks to steal important information such as intellectual property (IP) or secret keys [1][2]. The scan-based side-channel attacks have added to an already growing customer concern of hardware security [3][4][5]. As more information security measures are implemented on chip and more companies would like to keep their designs or IP protected, additional security measures must be implemented to defend from the multitude of intrusive and side-channel attacks that exist.

Scan test provides extensive controllability and observability over a chip. While beneficial for testing purposes, these properties also make reverse engineering the chip much easier. Testability and security inherently contradict each other. The higher degree of controllability and observability allowed, the easier it is to test the circuit under test (CUT). Security of a chip attempts to limit controllability and observability in order to protect and obscure any important information from any unauthorized persons. Most security measures implemented in hardware have been used to protect information, but if these measures are leaking information through side-channels, which has become the case for scan-based testing, additional hardware measures must be taken to prevent such events from occurring.

If the designer does not consider testability when designing for security, he may lose the efficient and reliable test mechanism. However, if security is overlooked in favor of testability, too much information may be too easily accessible. In order to find a solution, the designer must also consider the hacker and the expected skill set of the hacker [6]. A proper balance of all these factors is necessary when considering an appropriate solution.

A. Prior Work

Hardware security has been receiving a lot of attention with the increased use of cryptochips in various applications like smart cards and other embedded systems [7][8]. Much of the discussion has been focused on tamper resistant designs in order to prevent intrusive and side-channel attacks [3][4][9]. But with all this attention towards security, scan chains have only recently been shown to be just as dangerous [2].

A traditional method of scan security has been to place polysilicon fuses near the pin connections and blow them after manufacturing testing or completely cut off the test interface with a wafer saw [7]. This however eliminates any possibility for in-field testing. Many have gotten around the concern by using BIST to test the entire design [10] or a hybrid method that uses BIST on sensitive portions and scan on the remainder [11]. Although a viable solution, the fault coverage still does not reach the levels of scan and ATPG.

There has recently been an increased focus to secure scan designs without completely destroying access to the test interface. An encoding/decoding security scheme has been presented as a potential solution but have not been fully developed yet [12][13].

To prevent finding secret key information, a simple solution proposed is to use a second register, called the mirror key register (MKR), to prevent any critical data from entering the scan chain when in test mode [1]. Although effective, application of this solution is fairly limited since this approach can only protect information and not the actual IP the chip may contain. Scan scrambling was presented in [14], which divides the scan chain into subchains and uses logic and a random number generator to randomly reconnect the subchains together again and internally scramble data. Using this strategy with a large number of subchains yield a high complexity, but also begins to create significant area and timing overhead. Another technique checks a "golden signature" after a well defined consistent system reset to ensure secure switching between system and test modes [15]. A lock & key technique was developed in [6], which also divides the scan chain into subchains, but rather than connecting all of the subchains back together, all the subchains are independent of each other. By using a randomly seeded LFSR, one subchain at a time would be randomly enabled to scan-in and scan-out while all other subchains are left unaffected. This technique also has the disadvantage of scaling poorly for a very large number of subchains.

B. Contribution and Paper Organization

We propose a low-cost secure scan solution that can be used to prevent scan-based side-channel attacks while maintaining the testability features of scan. By allowing for varying levels of security, our solution is flexible and can easily be integrated into the scan insertion flow. This low-cost solution requires that a test key be included into all test vectors that are to be

Proceedings of the 24th IEEE VLSI Test Symposium (VTS'06)
0-7695-2514-8/06 $20.00 © 2006 IEEE

used for scan test by inserting dummy flip-flops into the scan chains. By doing so, it verifies that all vectors scanned-in come from an authorized user and the correct response can be safely scanned-out after functional mode operation. If the correct key is not integrated into the vector, an unpredictable response will be scanned-out making analysis very difficult for an attack. By using an unpredictable response, a hacker would not be able to immediately realize that their intrusion has been detected as could be done if the CUT were to immediately reset [15]. Our approach to scan security requires little area overhead and does not negatively impact important aspects of the original design such as performance.

We have organized the paper as followed. In Section II we present how an attack can be made using the scan chain. Section III presents our low-cost secure scan solution, which can be used to prevent scan-based attacks. Section IV provides an overview of how our solution can be integrated into the current scan insertion flow. Section V analyzes the impact on design and shows the implementation results. Finally, we conclude our discussion in Section VI.

II. Scan-Based Attack Categorization

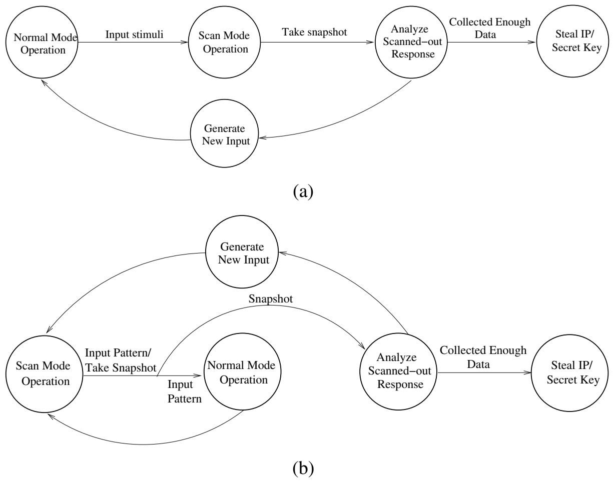
Developing a secure scan design is dependent on targeting both the type of attacker [6] and how they can potentially make the attack. We categorize the scan-based attacks into two types: *scan-based observability* and *scan-based controllability/observability* attacks. Each requires that a hacker has access to the test control (TC) pin. The type of attack depends on how a hacker decides to apply stimuli. The low-cost secure scan design we propose removes the hacker's ability to correlate test response data by creating a random response when an unauthorized user attempts access.

A. Scan-Based Observability Attack

A *scan-based observability* attack relies on a hackers ability to use the scan chain to take snapshots of the system at any time, which is a result of the observability from scan-based testing. Figure 1(a) diagrams the necessary steps to perform a scan-based observability attack. The hacker begins this attack by observing the position of critical registers in the scan chain. First, a known vector is placed on the primary input (PI) of the chip and the chip is allowed to run in functional mode until the targeted register is supposed to have data in it. At this point, the chip is placed into test mode using TC and the response in the scan chain is scanned-out. The chip is reset and a new vector that will cause a new response only in the targeted register is placed on PI. The chip again is run in functional mode for the specific number of cycles and then set into test mode. The new response is scanned-out and analyzed with the previous response. This process continues until there are enough responses to analyze where in the scan chain the targeted register is positioned.

Once the targeted register is determined, a similar process can be used to either determine a secret key in the case of cryptochips or determine design secrets for a particularly innovative chip.

B. Scan-Based Controllability/Observability Attack



(a)

(b)

Figure 1. Summary of the steps necessary to perform a successful scan-based attacks: (a) Scan-based observability attack and (b) Scan-based controllability/observability attack.

Scan-based controllability/observability attacks take a different approach to applying stimuli to the CUT, which is shown in Figure 1(b). Scan-based controllability/observability attacks begin by applying the stimuli directly into the scan chain as opposed to the PI. In order to mount an effective attack, the hacker must first determine the position of any critical registers as was done for the scan-based observability attack. Once located, the hacker can load the registers with any desired data during test mode. Next, the chip can be switched to functional mode using the vector the hacker scanned-in, potentially bypassing any information security measures. Finally, the chip can be switched back to test mode to allow the hacker a level of observability the system primary output (PO) would not provide otherwise.

As opposed to using a known vector to scan in to the chain, hackers also have the opportunity to choose a random vector to induce a fault in the system. Based off of the fault-injection side-channel attack [16][17], by inducing a fault, the chip may malfunction potentially revealing critical data. The scan chain becomes an easily accessible entry point for inducing a fault and makes the attack easily repeatable. In order to protect from such side-channel attacks, additional hardware security measures must be included in the design.

III. Low-Cost Secure Scan (LCSS)

We propose a low-cost secure scan (LCSS) solution that provides flexibility in design and integrates smoothly with current scan insertion flow. After performing scan insertion, the testing process is minimally affected while not affecting functional mode operation.

The state of the scan chain is dependent on a test key that is integrated into all test vectors. We define two possible states for the chain: *secure* and *insecure*. By integrating the key, all vectors scanned-in can be verified to be from a trustworthy source (secure). Without a correct key integrated into the test vector, when scanning in a new vector and out the response, the response will be randomly altered to prevent reverse engineering of critical data that is being stored in registers (insecure). By altering the response scanned out of the chain, both the scan-based observability and scan-based controllabil-

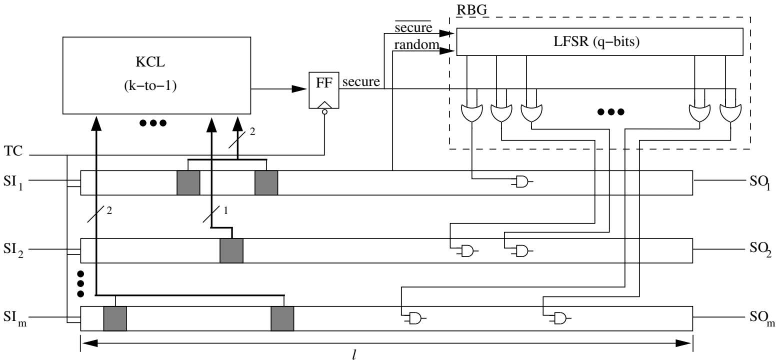


Figure 2. General architecture for low-cost secure scan design.

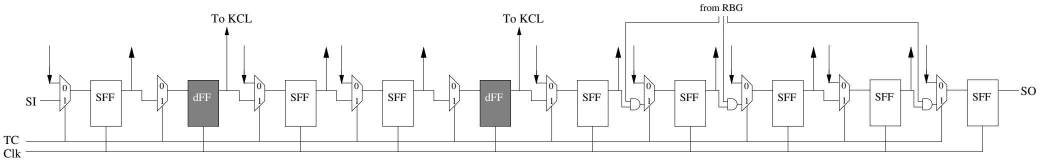


Figure 3. Example low-cost secure scan chain with integrated dummy flip-flops and random response network.

ity/observability attacks are prevented since any attempt to correlate the responses from various input will prove unsuccessful due to the random altering of the data.

The low-cost secure scan architecture is shown in Figure 2 with a more detailed look at a secure scan chain in Figure 3. In order to use the same key for every test vector, dummy flip-flops (dFFs) are inserted and used as test key registers. Each dFF is designed similarly to a scan cell except that there is no connection to a combinational block. The number of dFFs included in the scan chain depends on the level of security the designer would like to include since the number of dFFs determine the size of the test key. When implementing low-cost secure scan for multiple-scan design, the test key is inserted into the scan chain before it is broken into multiple scan chains. This ensures that the key can be randomly distributed throughout the many scan chains without needing to have a constant number of key registers in each chain.

All dFFs are concurrently checked by the Key Checking Logic (KCL), which is made of a block of combinational logic. The k-input block, where k is the total number of dFFs in the scan design (length of the test key), fans into a single FF that is negative edge sensitive to TC. As the CUT switches from test mode to functional mode (TC falls), the FF clocks in the output of the key checking logic. The KCL-FF is then used to inform the remainder of the secure design of the current secure or insecure state of the vector in the scan chain.

The third component of the low-cost secure scan solution ensures the random response in the scan chain when the test key fails to be verified by the KCL. The output of the KCL-FF fans out to an array of q 2-input OR gates. The second input of each OR gate comes from a q-bit LFSR that has been randomly seeded using one of a variety of options including, but not limited to, the value already present at reset, a $random$ signal from a FF in the scan chain as shown in Figure 2, or a $random$ signal from the output of a separate random number generator [18]. The former option provides the least amount of overhead, but potentially the least secure, while the latter has the most security, but also the most overhead. By also using an $\overline{secure}$ signal to the LFSR, also shown in Figure 2, the LFSR seed can continually be changed by an additional random source. Together, the LFSR and OR gate array make up the Random Bit Generator (RBG). The RBG output is used as input to the Random Response Network (RRN) that have also been inserted into the scan chain. The RRN can be made of both AND and OR gates to equalize the random transitions and prevent the random response from being all zeros or all ones. The optimal choice for randomness would be to use XOR gates, but since XORs add more delay, our design choice was to use AND and OR gates. Since the dFFs are used to check the test key, dFFs must be before any gates of the RRN in the scan chain as shown in Figure 3. If this property is not held, any key information that is trying to pass a gate of the RRN in the scan chain may potentially get altered either preventing the test key from ever being verified or even randomly changing a value to the correct key.

Normal mode operation of the CUT is unaffected by the addition of the low-cost secure scan design since the dFFs are only used for testing purposes and are not connected to the original design.

A. Test Flow

Our low-cost secure scan design deviates very little from current scan test flow. Since security of the scan chain is ensured by integrating a test key into the test vectors themselves, no additional pins are necessary to use low-cost secure scan.

After a system reset and TC has been enabled for the first time, the secure scan design begins in an insecure state causing any data in the scan chain to be modified as it passes through each RRN gate in the chain. In order to begin the testing process, the secure scan chain must be initialized with the test key in order to set the output of the KCL-FF to 1. Only the test key is required to be in this initialization vector since any other data beyond the first RRN gate most likely will be modified. During this time the KCL will constantly be check the dFFs for a correct key. After the initialization vector has been scanned-in, the CUT must be switched to functional mode for one clock in order to allow the KCL-FF to capture the result from the KCL. If the KCL verifies the key stored in the dFFs, the KCL-FF is set to 1 and propagates the signal to the RRN, which becomes transparent for the next round of testing allowing the new vector to be scanned-in without alteration.

Testing can continue as normal once the initialization process has been finished. However, the chain can return to insecure mode at any time during scan testing if the correct test key is not present in all subsequent test vectors. Should that occur, the RRN will again affect the response in the scan chain and the initialization process must again be performed in order to resume a predictable testing process.

IV. Low-Cost Secure Scan Insertion

Including low-cost secure scan minimally changes the current scan insertion flow. We were able to implement the LCSS insertion flow using Synopsys Design Compiler [19] and small add-ons functions we developed in C. The add-ons performed tasks like automating the KCL creation depending on the desired number of dFFs, creating a desired LFSR size for the RBG, performing RRN randomization and inserting RRN gates. In order to allow smooth RRN insertion, we instantiated temporary placeholder FFs in the scan chain that would later be replaced after the scan chain had been stitched together. The entire process became automated by a single script once the add-on functions were created. We have summarized the low-cost secure scan insertion flow in the following steps.

Low-Cost Secure Scan Insertion Flow

1) Define # of dFFs (size of key and KCL).
2) Define KCL key (random or user defined).
3) Define # of LFSR bits.
4) Define # of RRN gates (must be $\leq$ # of LFSR bits).
5) Load and Compile KCL and RBG.
6) Load and Compile Targeted Design (TD).
7) Set Current Design to TD.
8) Initialize KCL and RBG in TD.
9) Initialize dFFs in TD.
   - Connect CLK of TD to dFFs.
   - Connect Q of all dFFs to respective KCL port.
10) Initialize RRN placeholder FFs.
    - Connect CLK of TD to all placeholder FFs.
    - Connect D of all placeholder FFs to respective RBG port.
11) Reorder scan chain placing dFFs before all RRN placeholder FFs.
12) Perform scan insertion.
13) Replace RRN placeholder FFs with actual RRN gates.
14) Load and Compile new netlist with Low-Cost Secure Scan included in TD

V. Analysis and Results

Overall, our low-cost solution has little affect on important overhead aspects of chip design but significantly increases the security of the scan chain and still allows the testability of scan design.

A. Overhead

When implementing our secure scan solution, we wanted to affect area, test time, and performance as little as possible to maintain a low-cost strategy.

•**Area**: The KCL is the only component in our LCSS design that could potentially create a substantial area overhead. The size of the KCL for our designs remains fairly small, but is entirely dependent upon the size of the key and the value of key. The logic to perform the key verification at most will always be $(k-1)$ gates plus the number of inverters. The number of inverters in the KCL are dependent on the value of the key and will never be greater than k. Since k inverters would translate to a key of all zeroes or all ones depending on the KCL implementation, an actual key should have approximately the same number of ones and zeros ($\frac{k}{2}$ inverters). So on average, the size of the entire KCL will be $\frac{3k}{2}-1$ gates.

We implemented our low-cost secure scan and conventional scan on several ISCAS'89 benchmarks [20]. When including LCSS, we used a test key size of 10-bits, a 4-bit LFSR, and 10 RRN gates. The second column of Table I shows the total size of benchmarks before scan insertion, which includes the area of both combinational logic and FFs. Columns three and four list the sizes of the benchmarks after conventional scan and LCSS insertion, respectively. In columns five and six, we have shown the ratio of the overhead created by conventional scan and LCSS.

The area overhead of the low-cost secure scan design is fairly significant over the size of the benchmark, but this is also true for conventional scan. If one considers the overhead of our technique over conventional scan, which is shown in column seven, the area overhead is quite minimal. So, as long as scan is going to be used, our technique has very minimal impact on area. Also, as the benchmarks become larger, the overhead becomes less significant since our secure scan size is fixed. We expect that this would hold true even more so for modern designs due to their immense size.

The LCSS sizes reported (column four) include dFFs that have been added to the ISCAS'89 benchmark scan chains. However, as designs grow larger, usually dummy flip-flops are inserted into scan chains for various purposes [21]. It may be possible to use these dFFs as key registers. By using the already present dFFs, the impact of low-cost secure scan can be reduced further. The eighth column in Table I shows the overhead if dFFs did not have to be added. By considering this, we reduce the overhead of secure scan over conventional scan by more than half as shown by comparing columns seven and nine.

•**Test Time**: For modern designs with a very large number of scan cells, including additional test key registers will not affect scan test time by a significant amount. The total LCSS test time (T_{LCSS}) and test time increase (T_{inc}) over conventional scan can be summarized by

$$T_{LCSS} = (n_{comb}+3)\cdot\frac{n_{ff}+k}{m} + n_{comb} + 4 \qquad (1)$$

$$\text{and } T_{inc} = (n_{comb}+3)\cdot\frac{k}{m} + \frac{n_{ff}}{m}, \qquad (2)$$

TABLE I

AREA OVERHEAD OF SCAN AND LOW-COST SECURE SCAN ON ISCAS'89 BENCHMARKS.

Benchmark Name	Size of Bench	Bench w/ Scan	Bench w/ LCSS	Scan / Bench Overhd (%)	LCSS / Bench Overhd (%)	LCSS - Scan Overhd(%)	LCSS w/o dFFs / Bench Overhd (%)	LCSS w/o dFFs - Scan Overhd (%)
s13207	6298	7550	7711	19.9	22.4	2.5	21.0	1.1
s15850	6124	7156	7317	16.9	19.2	2.3	18.0	1.1
s35932	19986	23442	23603	17.3	18.1	0.8	17.6	0.3
s38417	18169	21297	21458	17.2	18.1	0.9	17.6	0.4
s38584	17433	19985	20146	14.6	15.6	1.0	15.0	0.4

TABLE II

TEST TIME INCREASE (T_{inc}) OF LCSS WITH 10, 40, AND 80-BIT KEYS ON ISCAS'89 BENCHMARKS.

	Test Time Increase (%)	
	s35932	s38584
m	18	15
LCSS $k=$ 10-bit	1.5	1.7
LCSS $k=$ 40-bit	3.3	3.8
LCSS $k=$ 80-bit	5.6	6.8

where n_{ff} is the total number of flip-flops in the design without including dFFs, k is the number of test key registers, m is total number of scan chains, and n_{comb} is the number of combinational vectors. Equation 1 accounts for the initialization process, chain test, and all test sequences. Equation 2 is the result of subtracting the total test time of conventional scan from T_{LCSS}. T_{LCSS} could be reduced further if the initialization process and chain test occurred concurrently, but this is dependent upon the location of the dFFs. As modern designs continue to have more scan cells and k remains fairly small in comparison, the test time increase will become less significant.

We have included potential test time increase (T_{inc}) percentages in Table II for conventional scan and secure scan using a key size of 10, 40, and 80-bits on two ISCAS'89 benchmarks. We assume the designs use multiple scan chains (m scan chains) and the number of chains are set to a value that will keep each chain length at approximately 100 cells when conventional scan is used. We have also set $n_{comb} = 100$. The results show that T_{inc} can be negligible for large designs when k is much smaller than n_{ff}.

•**Performance**: Performance of a design is not affected by low-cost secure scan any more than it would by conventional scan test. The speed of the design remains only affected by the inclusion of the scan chain itself and not further hindered by the inclusion of dummy flip-flops or RRN gates between the scan cells.

Power consumption during normal mode operation is affected very little. However, consumption during test mode only becomes a concern should the designer use a very large key with many RRN gates since this could considerably increase switching activity during scan operation. But as long as the key length and number of RRN gates remain small in comparison to the total number of scan cells, additional power consumption will be minimal.

B. Affect on Security and Testability

While maintaining a low-cost overhead, our secure scan solution is able to greatly increase the security of the scan chain while still providing the high controllability and observability conventional scan usually provides.

•**Security**: If a hacker attempts to perform an attack, any vector scanned-in or any critical data scanned-out will be randomly altered by the RRN throughout the chain. With each additional RRN gate used in a scan chain, the probability of determining the original value drops.

If the attacker would like to perform a scan-based side-channel attack without being hindered by the security measures in place, the hacker must bypass seven security elements:

1) The test key,
2) The location of each test key register in the scan chain (or test vector),
3) The number of RRN gates in each scan chain,
4) The type of RRN gate being used,
5) The random LFSR seed,
6) The LFSR polynomial, and
7) In the case of multiple-scan designs, the hacker must also determine the configuration of the decoder/encoder on the input/output of the scan chain.

The last five security elements are very difficult to determine without disassembling the chip and using expensive equipment.

Each level of security adds to the complexity of determining the contents of the scan chain. The k-bit key alone provides 2^k potential combinations. Our implementation placed all dFFs in the first $\frac{2}{3}$ of each scan chain. Choosing k-bits out of the first $\frac{2n}{3}$ scan cells in the chain creates an additional level of complexity to the security strategy used in low-cost secure scan, where n is the total scan length including dFFs. There are $\binom{\frac{2n}{3}}{k}$ possible key location combinations. We placed the RRN gates in the later $\frac{1}{3}$ of each scan chain creating a complexity of $\binom{\frac{n}{3}}{r}$ where r is the total number of RRN gates used in the design. Determining the type of RRN gate doubles the combinations. Also, the LFSR seed and LFSR polynomial each provide 2^q potential combinations. Combining all of these components together result in

$$2^{k+2q+1} \cdot \binom{\frac{2n}{3}}{k} \cdot \binom{\frac{n}{3}}{r} \qquad (3)$$

total combinations. When considering multiple-scan chain decoding/encoding, Equation 3 shows the minimum number of possible combinations. We have summarized the complexity of low-cost secure scan in Table III on two ISCAS'89 benchmarks using various key sizes. For simplicity, we keep the LFSR size fixed at 4-bits ($q = 4$). Table III shows that as we increase the key and as the number of scan cells increase, the complexity dramatically increases.

• Scan-Based Observability Attack Prevention: With the low-cost secure scan design in place, if the hacker attempts to run a system in functional mode and immediately takes a snapshot with access to the test control pin, any response on SO will result in randomized data due the lack of a correct test key and the RRN integrated into the chain. A hacker is given a false hope of a successful attack since a response is still scanned-out, but attempting to correlate the responses would fail.

TABLE III

COMPLEXITY OF LCSS WITH 10, 40, AND 80-BIT KEYS ON ISCAS'89 BENCHMARKS.

	Complexity	
	s35932	s38584
m	18	15
LCSS k = 10-bit	$2^{19}\binom{1159}{10}\binom{580}{r}$	$2^{19}\binom{958}{10}\binom{479}{r}$
LCSS k = 40-bit	$2^{49}\binom{1179}{40}\binom{590}{r}$	$2^{49}\binom{978}{40}\binom{489}{r}$
LCSS k = 80-bit	$2^{89}\binom{1206}{80}\binom{603}{r}$	$2^{89}\binom{1004}{80}\binom{502}{r}$

- Scan-Based Controllability/Observability Attack Prevention: If the hacker attempts to scan-in a vector, unless the hacker knows the test key, it will be randomly altered by the RRN essentially creating a new vector unknown to the hacker. When the hacker cycles TC, similar to the scan-based observability attack, the hacker will not be able to easily correlate a particular portion of the response to any critical registers needed to perform an effective attack.

This security design is also effective against a fault-injection attack. The hacker may choose a vector at random to potentially force the CUT into a fault condition. However, due to the random stimuli alterations of the integrated RRN, the hacker will have difficulty reproducing the same fault with the same vector.

●**DFT**: In a multiple-scan design the test key can be distributed throughout all of the scan chains given that all of the chains are equal in length and the dFFs are placed before the RRN gates. Given that there may be hundreds of chains in the design, some chains may have one or two key registers while other chains have none at all. Still, only one KCL is necessary to check the key no matter how many chains are in the design. Placement of the RRN gates only occurs after the scan chain has been divided into multiple chains to prevent the potential of some chains being left unsecured by the RRN. Each of the scan chains can use all q-bits of the LFSR to randomize the data or use a different subset combination. Due to the flexibility of low-cost secure scan, there are a wide variety of customizable arrangements for the dFFs and RRN in multiple-scan designs while maintaining a conventional multiple-scan design interface.

Enhanced-scan techniques like test compression can still be used without any alterations since no changes are made to the interface of the scan chains. Even launch-off shift (LOS) and launch-off capture (LOC) can be implemented as normal. It may be necessary to add an additional dead cycle between capture and the next initialization cycles for LOS and an additional dead cycle between initialization and launch cycles for LOC, but the end result still successfully captured.

Including chips with low-cost secure scan in SoC designs or chips that need in-field testing do not need any special cases since the scan interface has not been changed beyond that of conventional scan design. By this level of abstraction, no additional pins are required by our low-cost secure scan design and map to a JTAG interface in the same fashion as conventional scan.

VI. CONCLUSION

We have presented a low-overhead secure scan solution that can be used to prevent scan-based side-channel attacks. By integrating the proposed technique into the scan insertion flow, the complexity of determining secret information significantly increases since a hacker must bypass up to seven levels of security until being able to access the scan chain. The hacker also may waste valuable time performing an attack without realizing a security strategy is being used since a random response is still output by the scan chain as opposed to resetting the chip or setting all values to zero/one. This strategy is flexible and can be extended to a variety of security levels depending on the needs and preference of the designer. The goal of adding security as opposed to removing testability allows in-field testing that could later prove invaluable depending on the application.

VII. ACKNOWLEDGMENT

The authors would like to thank Nisar Ahmed for his many helpful discussions.

REFERENCES

[1] B. Yang, K. Wu, and R. Karri, "Secure Scan: A Design-for-Test Architecture for Crypto Chips," in *Proc. of 42nd Annual Conference on Design Automation*, June 2005, pp. 135–140.

[2] ——, "Scan Based Side Channel Attack on Dedicated Hardware Implementations of Data Encryption Standard," in *Proc. of the IEEE Int. Test Conf. (ITC)*, 2004, pp. 339–344.

[3] S. Ravi, A. Raghunathan, and S. Chakradhar, "Tamper Resistance Mechanisms for Secure Embedded Systems," in *Proc. of the 17th Intl. Conf. on VLSI Design*, 2004, pp. 605–611.

[4] P. Kocher, R. Lee, G. McGraw, A. Raghunathan, and S. Ravi, "Security as a New Dimension in Embedded System Design," in *Proc. of the 41st Annual Conference on Design Automation*, June 2004, pp. 753–760.

[5] K. Tiri and I. Verbauwhede, "A VLSI Design Flow for Secure Side-Channel Attack Resistant ICs," in *Proc. of Design, Automation and Test in Europe*, Mar. 2005, pp. 58–63.

[6] J. Lee, M. Tehranipoor, C. Patel, and J. Plusquellic, "Securing Scan Design Using Lock & Key Technique," in *Intl. Symposium on Defect and Fault Tolerance in VLSI Systems*, 2005.

[7] O. Kömmerling and M. G. Kuhn, "Design Principles for Tamper-Resistant Smartcard Processors," in *USENIX Workshop on Smartcard Technology*, 1999, pp. 9–20.

[8] M. Renaudin, F. Bouesse, P. Proust, J. Tual, L. Sourgen, and F. Germain, "High Security Smartcards," in *Proc of the Design, Automation and Test in Europe Conf.*, 2004.

[9] R. Anderson and M. Kuhn, "Tamper Resistance – a Cautionary Note," in *Proc. of the Second USENIX Workshop on Electronic Commerce*, 1996, pp. 1–11.

[10] K. Hafner, H. C. Ritter, T. M. Schwair, S. Wallstab, M. Deppermann, J. Gessner, S. Koesters, W.-D. Moeller, and G. Sandweg, "Design and Test of an Integrated Cryptochip," *IEEE Design and Test of Computers*, pp. 6–17, Dec. 1991.

[11] R. Zimmermann, A. Curiger, H. Bonnenberg, H. Kaeslin, N. Felber, and W. Fichtner, "A 177 Mbit/s VLSI Implementation of the International Data Encryption Algorithm," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 3, Mar. 1994.

[12] R. Goering, "Scan Design Called Portal for Hackers," Oct. 2004. [Online]. Available: http://www.eetimes.com/news/design/showArticle.jhtml?articleID=51200154

[13] S. Scheiber, "The Best-Laid Boards," Apr. 2005. [Online]. Available: http://www.reed-electronics.com/tmworld/article/CA513261.html

[14] D. Hély, M.-L. Flottes, F. Bancel, B. Rouzeyre, N. Bérard, and M. Renovell, "Scan Design and Secure Chip," in *Proc. of the 10th IEEE Intl. On-Line Testing Symposium*, 2004.

[15] D. Hély, F. Bancel, M.-L. Flottes, and B. Rouzeyre, "Test Control for Secure Scan Designs," in *Proc. of European Test Symposium*, 2005.

[16] D. Boneh, R. A. Demillo, and R. J. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults," *Lecture Notes in Computer Science*, vol. 1233, pp. 37–51, 1997.

[17] E. Biham and A. Shamir, "Differential Fault Analysis of Secret Key Cryptosystems," *Lecture Notes in Computer Science*, vol. 1294, pp. 513–527, 1997.

[18] B. Jun and P. Kocher, "The Intel Random Number Generator," Cryptography Research, Inc., Tech. Rep., 1999, white Paper Prepared for Intel Corporation.

[19] Synopsys DFT Compiler, "User Manual for Synopsys Toolset Version 2004.06," Synopsys Inc., 2004.

[20] "ISCAS'89 Benchmarks," 1989. [Online]. Available: http://www.fm.vslib.cz/ kes/asic/iscas

[21] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing*. Kluwer Academic Publishers, 2000.