Securing IEEE 1687-2014 Standard Instrumentation Access by LFSR Key

Hejia Liu* and Vishwani D. Agrawal
Auburn University, ECE Dept., Auburn, AL 36849, USA
hzl0037@tigermail.auburn.edu; agrawvd@auburn.edu

***Abstract*—IEEE 1687-2014 Standard provides an effective method for accessing on-chip instruments for testing, debugging and board configuration. The standard, however, causes a safety problem because anyone can access the chip instruments, set inputs and obtain safety critical information. In recent work, a lock in the segment insertion bit (SIB) and a corresponding unlocking key application procedure have been proposed for securing the 1687. This paper provides a linear feedback shift register (LFSR) based key generation mechanism that enhances the security of 1687 very significantly. By reconfiguring m (a small number) scan flip-flops into an LFSR that generates the key to unlock the SIB, we show a $(2^{m-1}-1)/2$ times increase in the expected break-in time.**

***Keywords:* Hardware security, IEEE 1687-2014 Standard, JTAG Standard, PCB Testing.**

I. Introduction

The IEEE Standard 1149.1, also known as the JTAG standard, shown in Figure 1, was originally intended for board test. The test access port (TAP) interface is also used as a gateway to internal scan and in many other test, debug and programming features [1]. As IEEE 1149.1 became better established, the capabilities of TAP were explored. It can provide a higher level of access into the chip without the need for intrusive methods. Thus, TAP is an effective way to customize the board test. It helps to realize communication between instruments inside the chip by implementing 1687 [4], [6], [8], [9]. The development of IJTAG (Internal-JTAG) has been triggered by the growing level of complexity and functionality within the silicon chip or die. An IP (intellectual property) core with an IEEE 1687 compliant interface is called an instrument. IEEE 1687 network is a functional part that uses TAP as entrance to communicate with embedded instruments in chips for the purpose of testing, debugging and configuration as Figure 2 shows [5], [7].

In the context of IJTAG, the IP cores are predefined blocks of analog, digital or mixed signal circuitry performing particular functions, such as a clock a generator, an interface to an external measurement probe, a radio tuner, an analog signal converter, a digital signal processor, etc. These so-called embedded instruments may be internally designed circuit blocks reused across a product line, or third party IP cores purchased from an external source. Typically, they have a control (digital) interface, such as a command register, associated commands and synchronization signals, and a data path [7]. Specifically, a standard scan cell, called *segment insertion bit* (SIB), could be inserted in the scan chain of 1687. When certain bit is updated into SIB, it allows the SIB to provide access to instruments or other scan path. By including/excluding network segment, SIB effectively changes the length of the scan-path. This flexibility of 1687 saves engineering time by automating design tasks, and potentially reduces the aggregated test sequence length for IP blocks of an SoC, thus reducing test time and tester memory size.

* Present address: Broadcom, 16340 W. Bernardo Dr., San Diego, CA 92127, hejialiu@broadcom.com.

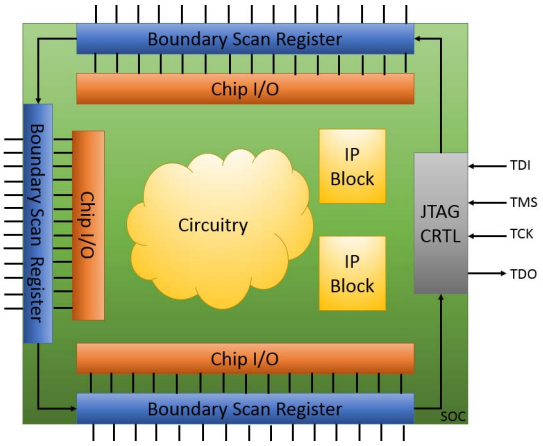

Fig. 1. IEEE 1149.1 JTAG boundary scan interface [1].

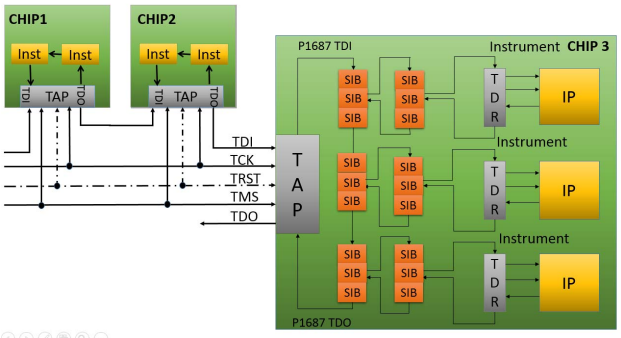

Fig. 2. An example of IEEE P1687 network for three chips [7].

In a plain 1687 network, anyone who can shift certain data into the SIB and set the 1149.1 TAP controller state to UpdateDR can open any SIB in the chain and access the instrument behind that SIB. An intruder, assuming that the IEEE 1687 is implemented in the chip, can access the instrument by shifting in instruction code from TDI to make the TAP controller work on 1687 set. By shifting in pattern to the 1687 scan chain, and loading 1 in some segments, the intruder can access the new path and new instrument at a *time* cost of only a few clock cycles.

Responding to concerns for security, several structures have been proposed. In this paper, we present a security scheme for modifying SIB by adding a security LFSR (SLFSR) as a cover. We consider the strategies the intruder may use and analyze situations faced. We compute the expected time of unlocking the SIB and compare with previously proposed security measures. We examine the assumptions made in our design and analysis.

A. Related Work

Figure 3 shows the standard SIB of IEEE 1687 [7] for which designs for security have been recently proposed [3], [11].

978-1-4673-9739-1/15 $31.00 © 2015 IEEE
DOI 10.1109/ATS.2015.23

IEEE computer society

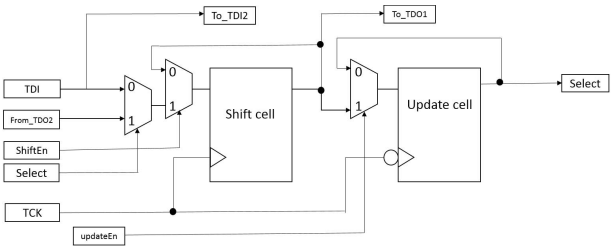

Fig. 3. Example of a standard SIB [7].

The standard SIB provides two scan chain reconfiguration capabilities: 1) bypass the segment of scan chain behind SIB (closed status), and 2) reconfigure the segment of scan chain behind SIB into the scan path (open status). The open status allows access to certain private instruments. To make any undesirable access of instruments difficult, several types of secured SIBs have been proposed. These require increased effort to break-in. A locking-SIB (LSIB) uses an n-bit key signal to gate the UpdateEn [3]. UpdateEn cannot proceed to the update cell unless this n-bit key is correct. For incorrect key bits, the SIB will remain locked irrespective of UpdateEN so that the select signal will keep the SIB in the closed status. A trap-bit SIB [11] uses the select signal as a feedback 'key to UpdateEn. In this case, UpdateEn will not be delivered if existing value in the update cell is wrong.

To make the intrusion into the circuit more difficult, several types of locking-SIB were proposed to increase the effort of possible intrusion. In a locking-SIB (LSIB) using an n-bit key signal to gate the UpdateEn, UpdateEn cannot be passed on to the update cell unless the n-bit key is correct. If the key is not correct, the SIB will remain locked holding the select signal to 0 and keeping the SIB in the close status. A trap-bit SIB uses the select signal as a feedback key to UpdateEn. Thus, UpdateEn will not be applied to update cell if existing value in the update cell is wrong. One may combine these two designs of locking and trapped SIB [11].

The work mentioned above [11] provides several impressive obfuscation strategies. The length of the scan chain is always seen as a feedback to successive break-in attempts. Upon a failed attempt to unlock an LSIB the intruder may increase the assumed chain length. Sometimes, unlocking a new LSIB or open a new path will lock another LSIB (Honeytrap LSIBs). Inserting LSIBs that are initially open on the circuit and locked when the correct keys are set provides negative feedback to the intruder, who must overcome the propensity of increasing the chain length to decrease the length of scan chain and unlock the target LSIB. By hiding an increase or decrease in chain length will provide no feedback to the intruder. All of these strategies increase the expected cost of unlocking a LSIB and thus increase the locking efficiency. Besides these strategies, a sequence filter [2] can also restrict access and enhance the security of the scan chain. A sequence filter includes a finite state machine (FSM) and additional logic inserted between the TAP and the scan network. However, it will be perfect if we could reuse the existing circuit to realize the function of the sequence filter.

B. SIB, Locking SIB and Trap

To access the embedded instruments with BIST interface TDR (Test Data Register) of Figure 2, one or more SIBs need

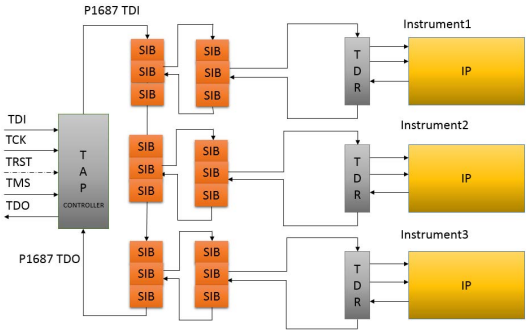

Fig. 4. A flat 1687 accessing network.

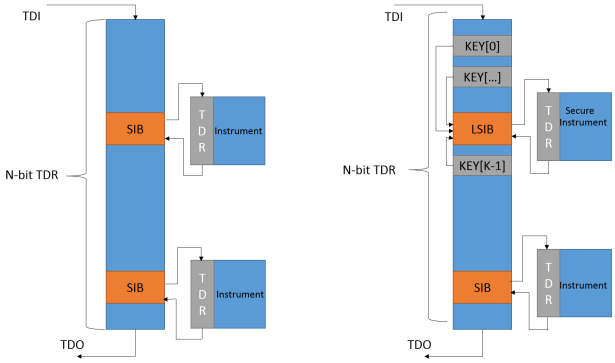

Fig. 5. Unsecured and secure 1687 instrument access.

to be open. If select is '0 (both selects are connected) and the SIB is closed, TDI will be the input of SIB and scan out from To_TDO in Figure 3. If both of SIBs are closed, the scan path is composed of SIBs. If the shift cell of SIB is 1 and after UpdateDR the value is clocked into update cell, the select will be 1 and From_TDO2 will be selected as the scan-in input, and TDRs from Instruments are configured in the scan chain.

When we open all SIBs, the scan path includes all TDRs in Figure 4. Once a value is updated into update cell, it will remain there until the Global reset is pulled down or updateDR is executed by the state machine. IEEE 1687 is very condensed compared to the traditional IEEE 1149.1 when the instruments are in large number amounting to hundreds of BIST circuits. We can hide 20-bit TDR behind each SIB to organize in five groups. The original scan path is only 5 bits long. Based on the test objective, we could exclude the BIST we do not need and make the scan length as small as possible. If there is important information, such as Chip ID hidden behind the SIB, then to prevent an intruder from opening SIB in this simple structure will be too hard. The intruder can find the scan length by shifting in a d bit signature and observe the number of clock cycles to receive the signature at scan-out. Then based on this length, one could randomly guess a possible position of SIB, shift in a sequence to update it, such as $1111\cdots$, and then shift in the signature again to measure the scan chain length. An increased length means there is at least one SIB opened. The intruder may now shift the information from behind the SIB out. To prevent an intrusion we increase the time required for such a break-in.

Figure 5 is a flat structure of LSIB with a k-bit key and total length n. The locking-SIB (LSIB) using an n-bit key

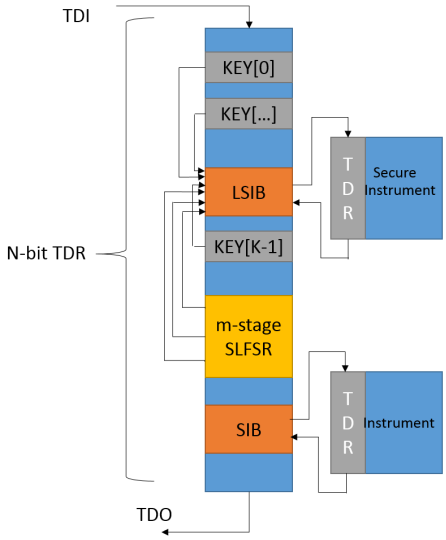

Fig. 6. An m-stage SLFSR as the cover of a hidden secure chain.

combined with UpdateEn as inputs to an AND gate blocks the UpdateEn [3]. UpdateEn cannot be delivered to the update cell unless the n-bit key is correct.

II. Adding Hidden Secure Chain and SLFSR

We use a dual mode m-stage SLFSR (Secure Linear Feedback Shift Register) with ShiftEn in the secured design as a cover of TDO. Parallel outputs from SLFSR are used as a feedback key to an LSIB as shown in Figure 6. To open the LSIB, we must shift in the correct key, which is produced as a sequence from the SLFSR. The select signal from the update cell of LSIB will decide the status of SLFSR. When LSIB is closed, the select will set the LFSR in the feedback-shift mode. When LSIB is open, the select will disconnect all feedback signals from D inputs in the LFSR and set it in the normal shift mode. When ShiftEn is 0, the state of SLFSR will be frozen. When ShiftEn is 1, the SLFSR simply shifts data from TDI. This scheme can be realized by adding multiplexers to a traditional LFSR. When the LSIB is locked, the TDO will be a repeating sequence. Only when LSIB is unlocked, the SLFSR will not be an obstacle in the scan path and we can get correct information from shifting out the pattern in the scan path. It is better to set the secure chain and SLFSR behind the SIB, because if we put this part in a necessary scan path, every time we use this path, we need to unlock it. Sometimes we do not need to access in the TDR behind the LSIB with SLFSR and setting a SIB to configure this path gives the hidden secure path more flexibility.

When the resetting state of LSIB is closed, the SLFSR is in the feedback-shift mode. The TDO is the output of SLFSR, which keeps repeating in a cycle no longer than $2^m - 1$. Figure 7 shows a 3-stage $(m = 3)$ dual mode SLFSR with ShiftEn. When "set" is 0, the SLFSR will clear to initial output state "111 when select is 1, or to "011 when select is 0. When ShiftEn is 1 and select is 0, successive output patterns will be 001, 100, 010, 101, 110, 111, and then repeat from 011. We could choose any output pattern as the unlocking pattern.

Unless a user is authorized to know the structure of design, it cannot be determined how many bits are set in the secure

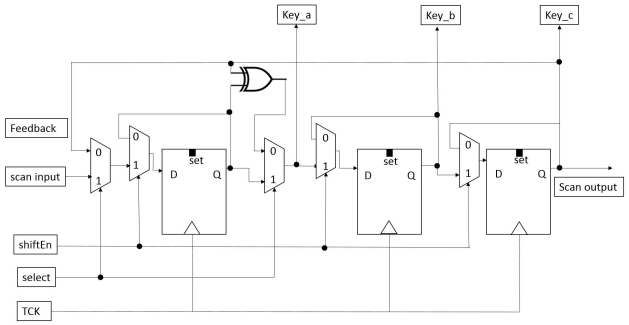

Fig. 7. A 3-stage dual mode SLFSR with ShiftEn.

chain by observing the output at TDO. An intruder must guess the length of the secure chain and no information is available to enhance the probability of getting the correct length.

In the basic LSIB with trap, an intruder can figure out the scan chain length by shifting in a d-bit signature and count cycles until the signature is received at TDO. Once the length is known, basic enumeration can unlock the LSIB. In using SLFSR as a cover, the intruder has no idea of the length of hidden secure chain, which can only be guessed.

We set the last bit of hidden secure chain as either LSIB or key. The length of the hidden secure chain extends from the beginning of the chain to the end of care bits (LSIB or key), which is n. Suppose, the intruder guesses it as n^*. There are three conditions we will discuss.

Condition 1: $n^* < n$. Suppose the intruder knows that an SLFSR is implemented. Just by analyzing the output, one can determine the repeat cycle length (R) for the SLFSR (in this example, $R = 7$). Initially, the intruder applies a reset to clear all possible traps in the hidden secure chain. There are 5 clock cycles from the reset to shiftDR status. After shifting an n^*-bit pattern, one has to test whether or not the pattern is working. Analyzing the output is a reliable method to distinguish the status of SLFSR and LSIB. The intruder needs at least $2R$ clock cycles in the shiftDR state to observe the repeating of TDO sequence, which also means LSIB is in the locking status. If it is not repeating, it means LSIB is unlocked. If it is failed, the intruder has to shift in the same pattern and update it with other output patterns of SLFSR. We assume that to combine the pattern with the next sequence of SLFSR, the intruder must wait for 1 clock cycle in shiftDR before shifting in an n^* bit pattern to combine with the r sequence, requiring extra r clock cycles. If the intruder shifts in random values in the waiting cycles, after running out of n^* bits possibility, the result fails to provide any information on whether n^* is long enough.

The reset, shift and update procedures are executed by the state machine of TAP controller. Figure 8 is a complete procedure to test one pattern. Consider a real situation where the intruder only shifts in n^* bits while n bits of the hidden secure chain are care bits. The intruder has no data on $n - n^*$ bits of the secure chain. As long as there is a single care bit that is wrong, the intruder will not be able to unlock the LSIB. In this situation, one has to exhaust all possible combinations of n^* and with no good result. This strategy can be realized by designing a special reset state. The cost (time) to test an n^*-bit pattern is shown in the Figure 8.

$$\text{Cost of 1 attempt} = n^* + r + 2R + 10 \quad (1)$$

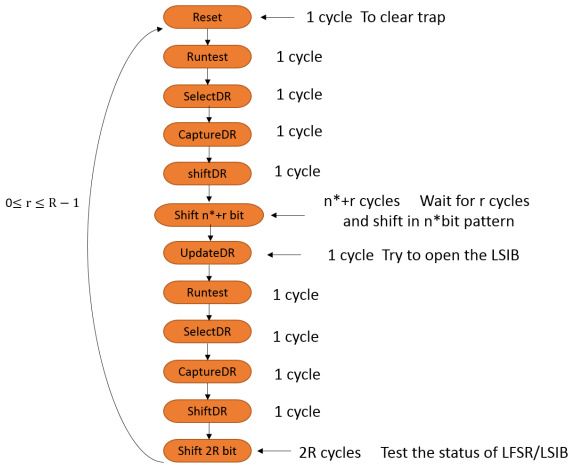

Fig. 8. Complete cycles needed to test one pattern.

We calculate the expectation E of the number of extra waiting cycles the intruder needs on average:

$$E = \sum_{r=0}^{R-1} \frac{r}{7} = 3, \quad when\ R = 7 \tag{2}$$

$$\text{Cost of testing } n^* \text{ bit pattern} = n^* + E + 2R + 10 \tag{3}$$

For n^* bits, there are total of 2^{n^*} patterns. Therefore, the cost of attempting all of them is,

$$(n^* + E + 2R + 10)2^{n^*} \tag{4}$$

After running out of all possibilities, the intruder will increase the shift length for the next attempt. The step of increasing this length depends on intruder's strategy. As long as the n^* is less than n, there are always care bits not touched. The intruder needs to exclude every possible key combination of length n^*. If the LFSR output pattern is known, then

$$\text{Probability} = \frac{1}{R} \tag{5}$$

Total necessary cost when $n^* < n$ is given by,

$$\sum_{\forall n^*} (n^* + E + 2R + 10)2^{n^*} R \tag{6}$$

where n^* is the guess about length each time and depends on the intruder strategy. This is the worst condition for the intruder who needs to enumerate all key combinations as a feedback then increase the length. If one increases the length and the length n^* is still less than n, then that will require another enumeration. The total cost of enumerations will drastically increase the time for a break-in. We consider an example of attempt length $n-1$, i.e., $(n^* = n-1)$. Assume that SLFSR has 3-bit with a repeating cycle of 7 $(R = 7)$. Assume that when intruder realizes that n^* is shorter than the correct length, he just increases it by 1 bit and eventually is lucky to obtain the correct length. Average number of enumerations is,

$$2^{n^*} \sum_{r=0}^{R-1} (n^* + r + 2R + 10) \tag{7}$$

Condition 2: $n^* = n$. In this condition, a lucky intruder just guesses the correct length of hidden scan chain and has certain probability to open the LSIB. It is not necessary to combine the shift in pattern with R sequences. For each attempt, intruder has a probability $1/R$ of combining with the right output pattern of SLFSR and unlocking LSIB. We need to calculate the expectation of how many extra waiting cycles we need to spend on average and the probability of shifting in the right keys.

TABLE I
EXPECTED TIME TO UNLOCK LSIB IN n-BIT SCAN CHAIN WITH 3-STAGE SLFSR USING CONDITION 1 AND 10MHZ CLOCK.

LFSR bits, k	n	Expected time		Clock cycles	Log of ratio over no SLFSR
		Days	Years		
8	32	1.96E-04	5.36E-07	1.69E+08	3.52
16	64	1.42E+06	3.90E+3	1.23E+18	10.76
32	128	4.78E+25	1.31E+23	4.13E+37	25.22
40	160	2.52E+35	6.89E+32	2.17E+47	32.44
48	192	1.28E+45	3.50E+42	1.10E+57	39.67
56	224	6.35E+54	1.74E+52	5.48E+66	46.90
64	256	3.09E+64	8.47E+61	2.67E+76	54.12

Cost to test one n-bit pattern

$$\begin{aligned} &= \frac{\text{cost of 1 attempt}}{\text{prob(present with right LFSR output pattern)}} \\ &= (n + 2R + 10 + E)R \end{aligned} \tag{8}$$

$$\text{Expected attempts} = 2^{k+1} \tag{9}$$

$$\text{Expected cost} = (n + 2R + 10 + E)R2^{k+1} \tag{10}$$

Condition 3: $n^* > n$. Intruder uses a very aggressive strategy and guesses a large n^*. In this situation, the expected cost equation should be the same as that for Condition 2. As long as the intruder shifts in n-bit patterns, the cost will only depend on the cost for each attempt, probability of guessing the right SLFSR, cost of shifting extra cycles and the expected attempts. Thus,

$$\text{Expected cost} = (n^* + 2R + 10 + E)R2^{k+1} \tag{11}$$

The difference between Conditions 2 and 3 is n^*. In Condition 3, n^* is larger than n. It means intruder needs to incur more cost for each attempt, which is $n^* - n$.

$$\text{Increased cost of Condition 3 over 2} = (n^* - n)R2^{k+1} \tag{12}$$

III. EXPERIMENTAL RESULTS

A. Experimental Results for Condition 1

Condition 1 is the worst case for the intruder who assumes a scan chain length n^*, enumerates all key combinations, and then increases the length. If n^* is still less than the real length n, another round of enumeration begins. The total cost of enumerations will drastically increase the break-in time. We consider an example. The attempt length is $n - 10$, i.e., $n^* = n - 10$ and SLFSR has 3-bits with a repeat cycle of 7, i.e., $R = 7$. Assume that the intruder, realizing that n^* is smaller than the correct length, increases to 10 bits to obtain the correct length. From experimental results in Table I, a 64-bit scan chain can have an expected time of 3,898 years if the intruder guessed in 54-bit and failed to unlock LSIB. The

TABLE II
EXPECTED TIME TO UNLOCK LSIB IN n-BIT SCAN CHAIN WITH 3-STAGE SLFSR USING CONDITION 2 AND 10MHZ CLOCK.

SLFSR bits, k	n	Expected time		Clock cycles	%Increase over no SLFSR
		Days	Years		
8	32	2.32E-07	6.36E-10	2.01e+05	395.9596
16	64	9.34E-05	2.56E-07	8.07e+07	377.9141
32	128	1.06E+01	2.90E-02	9.14E+12	365.6357
40	160	3.28e+03	8.98E+00	2.83E+15	362.8169
48	192	9.85E+05	2.70E+03	8.51E+17	360.8592
56	224	2.90E+08	7.93e+05	2.50E+20	359.4203
64	256	8.37E+10	2.29e+08	7.23E+22	358.3181
80	320	6.74E+15	1.85E+13	5.82E+27	356.7407
96	384	5.24E+20	1.44E+18	4.53E+32	355.6663

order of increase compared to the flat structure without SLFSR is linear in n, as can be proven.

For $n^* = n-10$, necessary cost for attempt on n^* bit pattern

$$= 2^{k+1}[(n+2R+10)R+E] + 2^{n^*}\sum_{r=0}^{R-1}(n^*+r+2R+10) \quad (13)$$

Expected cost for attempt on n bit pattern

$$= 2^{k+1}[(n+2R+10)R+E] \quad (14)$$

Cumulative cost

$$= 2^{n^*}\sum_{r=0}^{R-1}(n+E+2R+10).\ n^* = n-10 \quad (15)$$

The cumulative cost in cycles compared to Condition 2 is on the order of $n^* log(2n^*R)$.

From this unlucky Condition 1 result, it is really a bad idea to guess the length smaller than the correct length. Only 10 care bit trials from the intruder will enumerate all possibilities for the length n^*. That will lead to the expected time increase exponentially. If the intruder makes a wrong guess at 64 bit, then in Condition 1, it will cause the expected time to be on the order of thousand years. In conclusion, If it is a very small hidden secure chain, the intruder may have a chance to open it if the first time guessed length is less than the correct length, such as 32 bits. If the hidden secure chain is longer, the intruder has to pay for a wrong decision in more ways than one, being unable to unlock the LSIB and not having enough time to realize that the guess was wrong.

B. Experimental Results for Condition 2

Condition 2 is the best condition favoring the intruder. Table II compares the break-in time for n-bit scan chain with LSIB and trap without SLFSR. The chain length n includes both the hidden secure chain and SLFSR. The 3-bit SLFSR has a repeat cycle length of 7. Clock is 10MHz. From the result, to unlock the hidden secure chain with SLFSR, intruder needs at least 3.5 times as many cycles in comparison to scan chain without SLFSR to unlock LSIB after guessing the length correctly. When the key is 40-bits, n is 160-bits, the expected time is 8.98 years. That could be considered high security.

C. Experimental Results for Condition 3

In Condition 3, an intruder plans to shift in n^* bits, which is only 5 bits more than the correct length, i.e., $n^* = n+5$. From Table III, the the effort for break-in is a little larger than that in Table II. In a very small hidden chain, e.g., a chain with $n = 32$ bits, the increase is 35% larger than that for Condition 2. With increasing size of the hidden chain, the increase is approaching the Condition 2. However, this is a very special example in which n^* is only 5 bits larger than the real hidden scan chain. If the intruder is in Condition 3, the cost will linearly increase with n^*.

TABLE III
EXPECTED TIME TO UNLOCK LSIB IN n-BIT SCAN CHAIN WITH 3-STAGE SLFSR USING CONDITION 3 AND 10MHZ CLOCK.

LFSR bits, k	n	Expected time		Clock cycles	%Increase over no SLFSR
		Days	Years		
8	32	2.53E-07	6.93E-10	2.19E+05	431.3131
16	64	9.88E-05	2.71E-07	8.53E+07	399.3865
32	128	1.09E+01	2.99E-02	9.44E+12	377.6632
40	160	3.37E+03	9.23E+00	2.90E+15	372.6761
48	192	1.00E+06	2.76E+03	8.71E+17	369.2124
56	224	2.95E+08	8.09E+05	2.55E+20	366.4447
64	256	8.52E+10	2.33E+08	7.36E+22	364.7166
80	320	6.84E+15	1.87E+13	5.91E+27	361.9259
96	384	5.30E+20	1.45E+18	4.58E+32	360.0249

D. Analyzing Intruder's Strategies

We can combine the results from Condition 2 and Condition 3. It is not hard to conclude if the intruder uses aggressive strategies to guess n^* larger or at least equal to n, the expect cost result will in the same order $klog(2n^*R)$ as long as n^* is not over n too much. If the intruder uses conservative strategies and engages in Condition 1, he will definitely engage in two conditions. Because when intruder figures out he is in Condition 1, he has to increase his length and approach to Condition 2 or Condition 3. So either Condition 2 or Condition 3 will apply. The cumulative cost compared to aggressive strategies will be on the order of $n^* log(2n^*R)$ and n^* is in the different order of k. That is the reason why Condition 1 will cost exponentially increase compared to Conditions 2 and 3. If k is very close to n, in the worst case these three conditions will be almost same. The worst case means intruder in Condition 1 does not need to accumulate the cost too many times. If the intruder starts guessing the length from 1 bit and the incremental step for length is also 1 bit, then to solve an n bit hidden secure chain, we have an extra cumulative cost, given by

$$\sum_{n^*=1}^{n-1}\sum_{r=0}^{R-1}(n^*+r+2R+10)2^{n^*} \quad (16)$$

That could be very large. If n is 64 bits, it could take 36 million years when the clock frequency is 10MHz. That is only the cumulative cost from 1 bit to 63 bits. No matter what strategies intruder uses, the hidden secure chain will provide at least 3 times unlocking time compared to the flat structure without SLFSR. To protect from intrusion successfully, we need to consider the worst case, which is Condition 2. By analyzing the order for the magnitude of expected time, we could build the structure we need. If we need a secure design with expected unlocking (break-in) time of at least 1 year, we have an approximate equation:

$$log_{10}\frac{(2^k R\ cycles)/(frequency\ in\ Hz)}{365\ days \times 24 \times 3600} \geq 0 \quad (17)$$

Based on the requirement of length, we could decides how many keys we need. A potential disadvantage in this proposal

is that we add more feedback keys to the LSIB cell. If we use 3-stage SLFSR and 3 key bits, then there will 6 bits of feedback keys totally providing to the LSIB cell. If these 6 bits are coming from a 6-bit key, then the expected unlocking time will be larger than that for Condition 2.

For a secure chain without SLFSR, expected unlocking time is given by

$$(10 + 2n + d)2^{k+1} \tag{18}$$

For a secure chain with SLFSR, in the worst case (Condition 2), total expected unlocking time is,

$$(10 + n + 2R)2^{k+1-m}(2^m - 1) \tag{19}$$

Comparing Eqs. 18 and 19, we find an efficiency ratio, as

$$\frac{\text{Expected time without SLFSR}}{\text{Expected time with SLFSR}} \approx 2, \text{ when } n \to \infty \tag{20}$$

This shows that if we replace the k bits with a k-bit SLFSR, it might help an intruder under a certain condition. If the intruder can predict the correct (or almost correct) length, the unlocking time will be less.

IV. Conclusion

We have introduced an SLFSR key to cooperate with the original LSIB with traps and built a hidden secure chain that prevents an intruder from finding out the chain length using shifting procedures. The worst case for us is when the intruder guesses the length correctly. Even in this worst condition, the hidden secure chain still provides a 3.5 times improvement compared to the flat structure with three-stage SLFSR.

Several additional requirements based on the analysis are needed. No parallel output of secure network is allowed in the Capture state of TAP. All bits in scan chain should reset to 0 at Reset state. Otherwise, during the shift process controlled by the intruder, the untouched bits, the bits intruder does not intend to change, will have random values. As long as the intruder continues shifting, there is certain small probability that the random values will match the key. Even if 1 bit of key is not matched, the attacker has to keep attempting. If we set the chain in a specified value, which is the locking value of last care bits, it is impossible in the shifting process to form the correct keys when we randomly set k bits as 1 or 0. As long as we could keep 1 bit wrong in Condition 1, we could make intruder to emulate all the probability in the attempt length. Besides, we also assume if an intruder intends to test n^*-bits, he would not shift in random values in the waiting cycles (to combine the pattern with different SLFSR output pattern). When an intruder shifts in 3 bits randomly in the waiting cycles, that means the intention was not to test n^* bit patterns but test just n^* bits. We are considering intruders who follow certain strategies and are not pure gamblers. Besides, shifting in random values in the waiting cycle will not make the results to verify the assumption on n^* bits. Meanwhile, care bits must be present in the last bit of hidden secure chain. Otherwise, the effective secure chain is shorter than total length so that the intruder's condition will be changed. As a simple case, if the hidden chain length is 5 bits, the effective length is only 2 bits which spread in the first and second bit of the scan chain. When an intruder guesses the length as 2, in our analysis, he is guessing in the Condition 1. However, if the true condition is Condition 3, then not only this kind of structure increases the probability of intruder succeeding, but it also will cause less time for breaking in than our expectation.

The hidden secure chain with SLFSR provide two defense mechanisms. One is the remaining length of secure chain under cover of SLFSR. The other is a two-dimensional locking. We use k bits in the scan chain as one dimension of locking, and feedback from SLFSR as the other dimension. Only when both dimensions are in the correct position, the LSIB will be unlocked. No matter how conservative or aggressive strategy the intruder chooses, he cannot avoid spending more time to open the LSIB compared to Condition 2. Even in the worst case, Condition 2, will show much improvement. Although some intruder may get lucky, there is no reason to believe that every intruder will have the instinct to guess correctly. It is just like a gamble with rules designed by us. This structure can also be used in other security enhancement strategies we mentioned earlier. This structure can be applied to chips or boards. The board connections are easier to probe than inter-die connections of 3D stacked integrated circuits [10]. The lack of control and observation in 3D stacked integrated circuits may be useful in preventing an intruder from probing the 1687 network, however, there is need to extend security to 3D stacked integrated circuits.

References

[1] "IEEE Standard for Test Access Port and Boundary-Scan Architecture, IEEE Std 1149.1-2013 Revis. IEEE Std 1149.1-2001," May 2013.

[2] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Access Port Protection for Reconfigurable Scan Networks," *J. Electronic Testing: Theory and Applications*, vol. 30, no. 6, pp. 711–723, Dec. 2014.

[3] J. Dworak, A. Crouch, J. Potter, A. Zygmontowicz, and M. Thornton, "Dont Forget to Lock your SIB: Hiding Instruments using P1687," in *Proc. International Test Conference*, 2013.

[4] A. Jutman and K. Shibin, "Effective Scalable IEEE 1687 Instrumentation Network for Fault Management," *IEEE Design & Test of Computers*, pp. 26–35, Sept. 2013.

[5] E. Larsson and F. G. Zadegan, "Accessing Embedded DfT Instruments with IEEE P1687," in *Proc. 21st Asian Test Symp.*, 2012, pp. 71–76.

[6] K. Posse, A. Crouch, J. Rearick, B. Eklow, M. Laisne, B. Bennetts, J. Doege, M. Ricchetti, and J.-F. Cote, "IEEE P1687: Toward Standardized Access of Embedded Instrumentation," in *Proc. International Test Conf.*, 2006, pp. 1–8.

[7] K. Posse *et al.*, "IEEE Std 1687-2014: IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device." IEEE Computer Society, Approval date: 3 Nov 2014.

[8] J. Rearick, B. Eklow, K. Posse, A. Crouch, and B. Bennetts, "IJTAG (Internal JTAG): A Step Toward a DFT Standard," in *Proc. International Test Conf.*, 2005.

[9] J. Rearick and A. Volz, "A Case Study of Using IEEE P1687 (IJTAG) for High-Speed Serial I/O Characterization and Testing," in *Proc. International Test Conf.*, 2006, pp. 1–6.

[10] Y. Sun and J. Dworak, "Using an FPGA in a 3D Stacked IC to Prevent LSIB Bitstream Snooping," in *Proc. IEEE North Atlantic Test Workshop*, May 2015.

[11] A. Zygmontowicz, J. Dworak, A. Crouch, and J. Potter, "Making it Harder to Unlock an LSIB: Honeytraps and Misdirection in a P1687 Network," in *Proc. Design, Automation and Test in Europe Conf.*, Mar. 2014.