Design-for-Security vs. Design-for-Testability: A Case Study on DFT Chain in Cryptographic Circuits

Yier Jin
Department of Electrical Engineering and Computer Science
University of Central Florida
{yier.jin@eecs.ucf.edu}

***Abstract*—Relying on a recently developed gate-level information assurance scheme, we formally analyze the security of design-for-test (DFT) scan chains, the industrial standard testing methods for fabricated chips and, for the first time, formally prove that a circuit with scan chain inserted can violate security properties. The same security assessment method is then applied to a built-in-self-test (BIST) structure where it is shown that even BIST structures can cause security vulnerabilities. To balance trustworthiness and testability, a new design-for-security (DFS) methodology is proposed which, through the modification of scan chain structure, can achieve high security without compromising the testability of the inserted scan structure. To support the task of secure scan chain insertion, a method of scan chain reshuffling is introduced. Using an AES encryption core as the testing platform, we elaborated the security assessment procedure as well as the DFS technique in balancing security and testability of cryptographic circuits.**

I. Introduction

Malicious modifications to integrated circuits (ICs) have reshaped the structure of the IC supply chain recently. The threats from hardware Trojans, which can be inserted into target circuits at various stages of the design flow, force hardware designers/testers to reformat previous designing/testing techniques and take security into consideration. Upon this request, a new set of designing/testing techniques, design-for-security (DFS), are being proposed to balance the testability and the security of targeted circuits. The ultimate requirements of DFS techniques are two-fold: First, DFS methods should preserve the functionality of previously proposed testing methods and also ensure that the additional testing structures will not affect the circuit's trustworthiness. Second, new techniques should be proposed and integrated into the standard IC design flow so that the fabricated devices are less likely to be attacked by hardware Trojans.

This paper focus on the first level of DFS techniques as we try to evaluate the trustworthiness of current testing structures and propose solutions to enhance their security if these techniques are proven to cause security vulnerabilities to the target designs. Among all testing techniques, design-for-test (DFT) is the industrial standard to improve the controllability and observability inside circuits, especially for large-scale digital circuits and system-on-chip (SoC) designs where the inputs/outputs provide limited information about the operations of the internal logic. The DFT scan chain combined with automatic test pattern generation (ATPG) has been widely used because it can achieve high fault coverage, low design cost, easy implementation, and fast testing speed [1]. Commercial EDA tools are also developed to automate the scan chain insertion process, e.g., DFT Compiler, Encounter DFT Architect, etc. However, the inserted scan chains have also been exploited by attackers mainly because the scan chain provides an easy way to extract internal sensitive information. Various attacking methods have been developed targeting the scan chain to leak internal sensitive information. In [2], the authors used pairs of known plaintext to learn internal scan structure and recovered the DES encryption key. In [3], the scan chain attacks were expanded from secret-key algorithms to public-key algorithms and were able to decipher secret keys from RSA and ECC designs. Countermeasures have been proposed to maintain the testability of the inserted scan chain, and also prevent known scan chain based attacks [4]–[8]. Though proven to be effective in preventing simple scan chain attacks, these methods were revisited recently after more powerful scan chain based attacks emerged [9], [10]. In parallel with enhanced DFT methods, researchers have also worked on the implementation of more secure built-in-self-test (BIST) techniques on functional modules. A self-test architecture has been applied in crypto-devices with low performance and area overhead [11], [12].

However, all previously proposed DFT techniques, scan chain based attacks, and countermeasures are ad hoc solutions to balance testability/security and design cost. There lacks a fundamental solution to protect the DFT structures through formal evaluations of these structures. Orthogonal to all previous efforts, we propose to formally evaluate the trustworthiness of DFT structures based on a newly developed gate-level information assurance scheme within the scope of proof-carrying hardware. Information flow of all signals in the entire circuit will be tracked so that all leakage paths can be detected. New design-for-security solutions will also be proposed to re-construct the scan chain with the purpose of high security and high testability. Relying on the new DFS method, we can evaluate and prove security of DFT structures at the early stage of design flow to reduce the testing cost.

II. Proof-Carrying Hardware

A. *Proof-Carrying Hardware IP*

Proof-carrying code (PCC) was developed in the software domain to provide a way of determining whether code from potentially untrusted sources are safe to execute [13]. The verification method of PCC is accomplished by establishing a formal, automatically verifiable proof showing that questionable code obeys a set of formalized properties. A similar security evaluation method was proposed in the hardware domain recently. In [14], the authors introduced the application of

978-1-4799-3765-3/14 $31.00 © 2014 IEEE
DOI 10.1109/ISVLSI.2014.54

IEEE computer society

Proof-Carrying Hardware (PCH) in FPGAs and reconfigurable devices. A proof is generated to demonstrate that an agreed-upon specification function is combinatorially equivalent to the FPGA implementation (aka FPGA bitstream file). The authors in [15] then presented a new Proof-Carrying Hardware Intellectual Property (PCHIP) framework that helps guarantee the specified security properties are fulfilled by HDL code. An IP acquisition and delivery protocol is also proposed on the Coq proof assistant platform [16] to ensure the trustworthiness of purchased IP cores from untrusted IP vendors.

B. Dynamic Information Assurance Scheme [17]

The PCHIP framework introduced in [15] outlined the basic working procedure for proof-carrying based RT-level IP protection methods, but it did not specify details of security properties for individual designs. Considering the fact that different hardware IP cores often share similar security properties, in order to lower the design cost of the PCHIP framework and reuse previously developed property-proof pairs, authors in [17] chose data secrecy properties as prevailing properties and developed a dynamic information assurance scheme to protect data secrecy of general hardware IP cores. The proposed information flow tracking schemes assume that the data secrecy properties are agreed upon by IP vendors and IP consumers beforehand, so that the procedure of security property definition, a critical step when applying PCHIP framework, becomes trivial.

III. Gate-Level Information Assurance

The dynamic information assurance scheme provides a high-confidence IP protection option, but it falls short of trust evaluation on the gate-level netlist. To prevent unauthorized duplications and to protect the ownership of IP cores, design houses prefer selling a synthesized netlist, or even hard cores, to providing RT-level designs. Upon this request, a gate-level information assurance scheme which is similar to the IP transaction and proof preparation procedure as outlined by the dynamic information assurance scheme, but works for a synthesized netlist has been developed [18]. The properties formalization and proof generation of the gate-level scheme are both performed on the Coq proof assistant platform, similar to other hardware proof-carrying code schemes.

Figure 1 illustrates the preparation process of the trusted bundle defined by the gate-level information assurance scheme. According to the functional specification, HDL code will first be prepared by IP vendors and then be synthesized to netlist based on a specific technology library. Relying on the gate-level Coq formal logic and the signal sensitivity transition model, the IP vendor will convert the netlist into a structural Coq netlist. The applied data secrecy properties, if presented in English, meaning "no internal sensitive information will be leaked through primary outputs of the target design", will be formalized from English description into three kinds of theorems relating to the stable sensitivity list: 1) Existence; 2) Accessibility; 3) Trustworthiness.

Figure 2 shows the data secrecy property verification procedure performed by the IP consumer in the proposed scheme. Upon receiving the trust bundle (the trust bundle includes a synthesized netlist, security theorems and proof, and initial/stable sensitivity lists), an IP consumer will first check the contents of the initial signal sensitivity list and the stable signal sensitivity list, which represent the circuits initial secrecy status and its stable status, respectively. The validity of the initial list is checked to ensure that sensitivity levels are appropriately assigned to all input/output/internal signals. The circuit's stable sensitivity status contains complete information of the distribution of sensitive information across the whole circuit, so the stable list will then be carefully evaluated to detect any Trojan channels that may leak sensitive information to primary outputs. Only if both sensitivity lists pass the initial checking, can IP consumers proceed to the next step of properties verification by an automatic property checker. A "PASS" signal provides evidence that the netlist does not contain any malicious leaking channels prohibited by the data secrecy properties. However, a "FAIL" result is a warning signal that some of the data secrecy properties are breached in the delivered IP netlist.

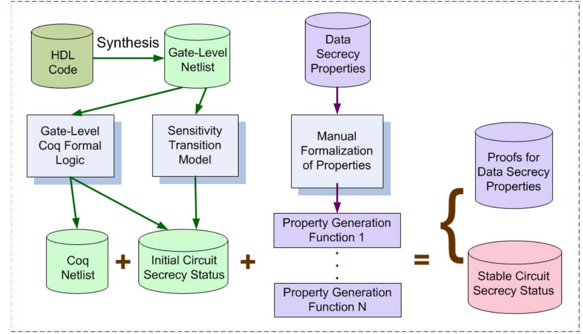

Fig. 1. Trusted bundle preparation in the gate-level information assurance scheme [18]

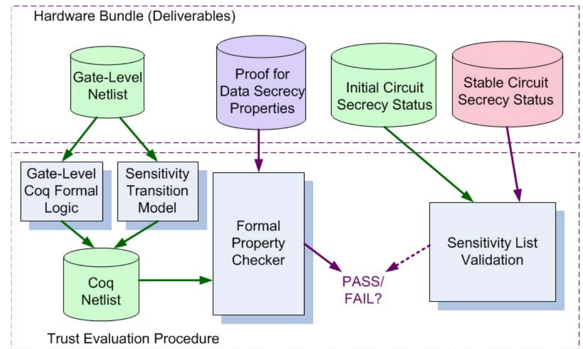

Fig. 2. Data secrecy property verification in the gate-level information assurance scheme [18]

The gate-level information assurance framework includes two main components: gate-level Coq semantic model and data secrecy property definitions. Details of the definition of both components can be found in [18].

IV. Security Analysis on DFT Techniques

The Design-for-Test (DFT) technique was developed along with the development of integrated circuits and is even more

important for modern integrated systems where billions of transistors fit on one chip. The ratio of testing cost to the overall cost of a fabricated IC has increased significantly over the last few decades forcing the industry to invest heavily in developing low-cost, high-efficiency testing methods. Among all of the DFT techniques, scan chain based testing/debuging/validation methods are the industrial standard with more than 80% of ICs equipped with some kind of DFT scan chains.

As controllability and observability increases because of the insertion of DFT scan chains, however, so do security vulnerabilities to the target design. Attackers may extract the scan chain structure and construct covert leakage channels to steal internal information without physically intruding into the target circuit [2], [3]. To counter these attacks, various solutions have been proposed to prevent data leakage through scan chains [4]–[8]. However, all of these countermeasures are solutions that counter known attack types without further consideration of future attack types. That is, none of these countermeasures has ever tried to formally evaluate the security vulnerability of the DFT scan chain and tried to enhance the DFT technique from the design-for-security (DFS) aspect. Using an AES encryption core as a test platform, we will demonstrate that even though pre-DFT netlists can be proven to comply with the selected security properties relying on the gate-level information assurance scheme, it will fail the same set of security properties after the insertion of a scan chain. We then evaluate the security of BIST enhanced circuit designs and show that the insertion of a BIST structure also fails the security properties validation.

A. *Pre-DFT Security Analysis*

Using Synopsis Design Compiler, the sample AES RT-level code was first synthesized to generate a pre-DFT netlist where no scan chains are inserted. The synthesized netlist is of much larger size than the RT-level description for the obvious reasons that high-level description is mapped to gate-level implementations. For example, there are 1964 signals defined in the netlist description compared to the 95 signals defined in the HDL code. Some of the converted signal definitions of the Coq netlist are shown below, where the values $0, 1, \ldots, 1963$ point to each signal's position in the centralized sensitivity list. From the definition of all the signals, we can tell that the length of the centralized sensitivity list of the AES Coq netlist is 1964.

```
(* Signal Definitions *)
Definition key_0 : signal := 0.
Definition key_1 : signal := 1.
Definition key_2 : signal := 2.
Definition key_3 : signal := 3.
Definition key_4 : signal := 4.
......
Definition sa33_sr_4 : signal := 1960.
Definition sa33_sr_5 : signal := 1961.
Definition sa33_sr_6 : signal := 1962.
Definition sa33_sr_7 : signal := 1963.
```

Some of the gate-level operations in the synthesized netlist and their counterparts in the Coq netlist are shown below. The similarity between the gate instantiation in the synthesized netlist and expressions in the Coq netlist makes it easier to develop automation tools for code conversion. Using Perl scripting language, we developed a set of code auto-conversion and theorem auto-generation tools so that all the Coq representative and security theorems presented in this paper are generated automatically, a significant step toward testing cost reduction.

```
(* Synthesized Netlist *)
assign N505 = ld;
EDFFX1 \text_in_r_reg[127]  ( .D(text_in[127]),
    .E(n790), .CK(clk), .Q(text_in_r[127]) );
EDFFX1 \text_in_r_reg[126]  ( .D(text_in[126]),
    .E(n790), .CK(clk), .Q(text_in_r[126]) );
EDFFX1 \text_in_r_reg[125]  ( .D(text_in[125]),
    .E(n790), .CK(clk), .Q(text_in_r[125]) );
EDFFX1 \text_in_r_reg[124]  ( .D(text_in[124]),
    .E(n790), .CK(clk), .Q(text_in_r[124]) );
......
CLKINVX1 U2100 ( .A(w0[26]), .Y(n17) );
CLKINVX1 U2101 ( .A(w0[10]), .Y(n10) );
CLKINVX1 U2102 ( .A(w0[1]), .Y(n6) );
CLKINVX1 U2103 ( .A(w0[9]), .Y(n9) );
CLKINVX1 U2104 ( .A(w0[0]), .Y(n5) );

(* Coq Netlist *)
Definition aes : code :=
assign_b N505 (ld);
nonblock_assign_ex text_in_r_127
    (ECOND_B n790 text_in_127 text_in_r_127);
nonblock_assign_ex text_in_r_126
    (ECOND_B n790 text_in_126 text_in_r_126);
nonblock_assign_ex text_in_r_125
    (ECOND_B n790 text_in_125 text_in_r_125);
nonblock_assign_ex text_in_r_124
    (ECOND_B n790 text_in_124 text_in_r_124);
......
assign_ex n11 (ECLKINV w0_11);
assign_ex n17 (ECLKINV w0_26);
assign_ex n10 (ECLKINV w0_10);
assign_ex n6 (ECLKINV w0_1);
assign_ex n9 (ECLKINV w0_9);
assign_ex n5 (ECLKINV w0_0);
```

Finally, the formal theorems to prove data secrecy properties are of the same format as those in the dynamic scheme and have been successfully proven on the pre-DFT Coq netlist. Due to page limitation, only security theorems are listed whereas the proof contents to these theorems are omitted in this paper.

```
(* Existence/Stability *)
Lemma aes_sen_stable : update_sensitivity
        aes aes_stable_list = aes_stable_list.

(* Accessibility *)
Theorem stable_list_accessability : forall t : nat,
        t > 5 ->  (check_sensitivity t aes
            aes_initial_list) = aes_stable_list.

(* Trustworthiness *)
Theorem no_leaking_1 :
        nth done aes_stable_list 0 = 0.
Theorem no_leaking_2 :
        nth text_out_0 aes_stable_list 0 = 0.
......
Theorem no_leaking_N :
        nth text_out_127 aes_stable_list 0 = 0.
```

B. *Post-DFT Security Analysis*

We have demonstrated that the pre-DFT netlist complies with the specified security properties. We then need to verify whether further modifications to the trusted netlist will affect

its security or not. Note that there are quite a few operations which are needed to modify the synthesized netlist such as buffer insertion and clock tree rerouting. Among these modifications, the DFT scan chain insertion is the most widely used technique to increase the controllability and observability of the target designs. Using Synopsys DFT Compiler, we insert one scan chain into the synthesized AES netlist to achieve high controllability and observability at the cost of slightly larger chip area and some extra input/output pins dedicated for scan chain operations.

According to the gate-level information assurance scheme and the code auto-conversion tool, the scan chain enhanced AES netlist is then converted into Coq netlist automatically where 2050 signals are defined (more than the 1964 signals defined in pre-DFT netlist). Extra signals include the scan chain enable `test_se` and the scan chain input `test_si`.

```
Definition key_0 : signal := 0.
Definition key_1 : signal := 1.
...
Definition  test_si : signal := 387.
Definition  test_se : signal := 388.
...
Definition sa33_sr_5 : signal := 2047.
Definition sa33_sr_6 : signal := 2048.
Definition sa33_sr_7 : signal := 2049.
```

Another source of extra signals is the inverted output of DFFs. In the original netlist, many of the DFF inverted output `QN` are left float while in the scan chain enhanced netlist, some of these `QN` outputs are used as part of the scan chain to balance the circuit performance and area consumption. Those scan DFFs with both `Q` and `QN` connected are instantiated such that each scan DFF is converted into two Coq expressions to represent the different behaviors of `Q` and `QN`. For example, the scan register `\text_in_r_reg[126]` connects wires `n2631` and `n2437` to its `Q` and `QN` outputs, respectively. In the converted Coq netlist, two sequential assignments are generated to present the behavior of `n2631` and `n2437`.

```
(* Scan Chain Enhanced Netlist *)
SEDFFX1 \text_in_r_reg[126]  ( .D(text_in[126]),
    .SI(n2632), .E(n2547), .SE(test_se),
    .CK(clk), .Q(n2631), .QN(n2437) );
SEDFFX1 \text_in_r_reg[125]  ( .D(text_in[125]),
     .SI(n2633), .E(n2545), .SE(test_se),
     .CK(clk), .Q(n2632), .QN(n2439) );

(* Coq Expressions for Scan DFFs *)
nonblock_assign_ex n2631 (ECOND_EX n2547
 (ECOND_B test_se n2632 text_in_126) (ESIG n2631));
nonblock_assign_ex n2437 (ECOND_EX n2547
 (EINV_EX (ECOND_B test_se n2632 text_in_126))
 (ESIG n2437));
nonblock_assign_ex n2632 (ECOND_EX n2545
 (ECOND_B test_se n2633 text_in_125) (ESIG n2632));
nonblock_assign_ex n2439 (ECOND_EX n2545
 (EINV_EX (ECOND_B test_se n2633 text_in_125))
 (ESIG n2439));
```

We then tried to prove the data secrecy properties on the post-DFT Coq netlist. The same set of security theorems are generated for existence, accessibility and trustworthiness of the circuit stable sensitivity list. However, we could not even prove the first lemma of **existence** on the scan chain enhanced netlist. When executing from the initial security status, the Coq netlist will leak sensitive information to primary outputs and scan-out pins before it can achieve a stable status[1]. For example, we have all signals in the Coq circuit evolve and their sensitivity level be updated for 10 cycles based on the sensitivity transition model and get the circuit secrecy status `aes_sen_10cycle`. Even though the `aes_sen_10cycle` is not the stable list, the output sensitivity checking reveals that the primary outputs and the scan output are contaminated by sensitive information, an indication that leakage paths exist in the post-DFT netlist. The following code shows that the `text_out` contains sensitive information after 10 cycles of evolution (note that the scan output shares the same pin of `text_out_127`).

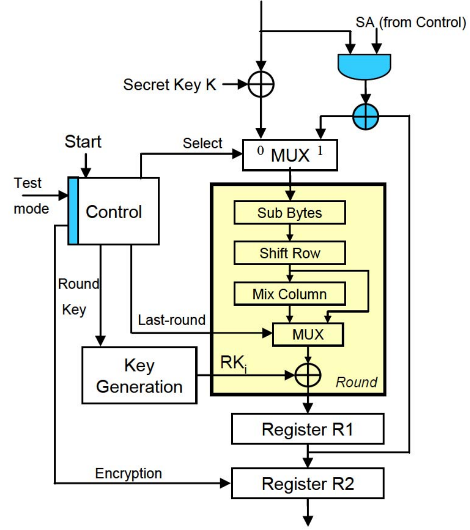

Fig. 3. Self-Test Techniques for AES Devices [11]

```
Definition aes_sen_10cycle :=
  check_sensitivity 10 aes_DFT aes_initial_list.
Compute (nth text_out_0 aes_sen_10cycle 0).
     = 2
     : sense
Compute (nth text_out_127 aes_sen_10cycle 0).
     = 1
     : sense
```

C. BIST Security Analysis

Based on the gate-level information assurance scheme, we have proved that scan chain insertion would deteriorate the circuit security status and impose leakage paths to the once trusted designs. The main reason behind the existence of leakage paths is that the inserted scan chain interferes with the internal data flow and creates additional paths connecting registers with sensitive data to registers with non-sensitive data so that data under protection may be propagated to primary outputs, bypassing the special sensitivity downgrading operations [17].

[1]Note that in our experimentation, we are unable to find a stable status because after many evolution steps, the memory consumption increases exponentially and exceeds the maximum memory size. But it does not affect our conclusion that the data secrecy properties cannot be proved in the scan chain enhanced netlist.

Meanwhile, built-in-self-test (BIST) provides an alternative way for IC testing and enables on-line fault detection at the post-deployment stage. However, due to the area cost and performance overhead, BIST is mostly used in memory chips and is rarely used in functional modules. Nevertheless, researchers find ways to reuse on-chip resources to apply BIST structures on SoCs with trivial overhead. One example is the self-test techniques proposed in [11] where a modified block cipher logic can perform test pattern generator (TPG), signature analyser (SA) and self-test functionality. The proposed BIST does not insert any scan chain to the original design so it is considered to be more secure than normal scan chain based testing methods. We replicated the BIST construction method in [11] by modifying the RT-level AES code. Figure 3 shows the BIST enhanced AES encryption core where the newly added logic for BIST configuration is shown in gray. In the original design, S-box input blocks `sa33,...,sa00` will be assigned plaintext inputs or the output of previous round according to the value of the load enable signal `ld_r`. After the enhancement of the BIST, the round output is XORed with the masked input before it goes to the next encryption round. Part of the RTL code modification is shown below where the code modifications are mostly on the initial adding round key step.

```
// Original initialization
always @(posedge clk)  sa33 <= #1 ld_r ?
  text_in_r[007:000] ^ w3[07:00] : sa33_next;
always @(posedge clk)  sa23 <= #1 ld_r ?
  text_in_r[015:008] ^ w3[15:08] : sa23_next;
always @(posedge clk)  sa13 <= #1 ld_r ?
  text_in_r[023:016] ^ w3[23:16] : sa13_next;
always @(posedge clk)  sa03 <= #1 ld_r ?
  text_in_r[031:024] ^ w3[31:24] : sa03_next;
......

// BIST enhanced initialization
assign sa_input[007:000] =
  (sa[007:000] & text_in[007:000]) ^ sa33_next;
assign sa_input[015:008] =
  (sa[015:008] & text_in[015:008]) ^ sa23_next;
assign sa_input[023:016] =
  (sa[023:016] & text_in[023:016]) ^ sa13_next;
assign sa_input[031:024] =
  (sa[031:024] & text_in[031:024]) ^ sa03_next;
...
always @(posedge clk) sa33 <= #1 ld_r ?
 text_in_r[007:000] ^ w3[07:00] : sa_input[007:000];
always @(posedge clk) sa23 <= #1 ld_r ?
 text_in_r[015:008] ^ w3[15:08] : sa_input[015:008];
always @(posedge clk) sa13 <= #1 ld_r ?
 text_in_r[023:016] ^ w3[23:16] : sa_input[023:016];
always @(posedge clk) sa03 <= #1 ld_r ?
 text_in_r[031:024] ^ w3[31:24] : sa_input[031:024];
```

We then evaluate whether the BIST enhanced AES design can still be trusted under data secrecy properties. With similar synthesis constraints of the original AES circuit, we synthesized the BIST equipped AES core to generate the post-BIST netlist. The same gate-level information assurance scheme is applied to the synthesized netlist for data secrecy evaluation. To our surprise, even though only minor modifications have been made on the BIST enhanced AES core, we cannot prove the data secrecy properties on the new netlist, a clear indication that there exists leakage paths due to the insertion of BIST logic. Similar to the DFT scan chain case, the security status after 10 cycles already reveals leakage paths to leak sensitive information at the `text_out` primary output.

```
Definition aes_sen_10cycle :=
  check_sensitivity 10 aes_BIST aes_initial_list.
Compute (nth text_out_0 aes_sen_10cycle 0).
     = 1
     : sense
Compute (nth text_out_127 aes_sen_10cycle 0).
     = 1
     : sense
```

The security evaluation results on scan chain and BIST inserted designs demonstrate that most of the currently used DFT techniques, while providing high controllability and observability, also impose information leakage paths to the original designs. Further, different from other scan chain attacking methods that try to leak internal information through the scan output pin, the proof-carrying based analysis demonstrates that leakage paths also exist in primary outputs.

V. Security Enhancement of DFT Techniques

As we demonstrated in Section IV, most DFT techniques are developed under the constraints of increased controllability and observability, while security has long been omitted in the DFT domain. In order to increase circuit security, yet still preserve testability, we developed a new design-for-security (DFS) technique with the goal of improving the security of DFT techniques so that DFT enhanced circuits can still pass security assessment. Being our first DFS demonstration, we developed a scan chain reshuffling method through which the inserted scan chain will not disturb the internal data flow, leaving the internal secrecy status intact. To support our work in scan chain reshuffling, a secure scan chain generation/insertion tool is developed which takes the synthesized netlist and the stabilized circuit secrecy status as inputs and generates a secure scan chain enhanced netlist. Additional parameters may also be defined to decide the amount of inserted scan chains. The key idea behind the proposed scan chain reshuffling method is to preserve the data sensitivity ordering within the target circuit. Rules are defined to ensure data can only flow from registers of low sensitivity levels to registers of high sensitivity levels in any constructed scan chains. Therefore, the inserted scan chain will increase circuit testability without imposing leakage paths to the primary outputs. That is, when we insert DFT scan chain into the synthesized netlist, we add more internal paths through which internal signals, including those carrying sensitive information, can spread across the whole circuit. We then need to reshuffle the order of the scan chain connections so that high level sensitive data will not contaminate low level sensitive data through the scan chains.

To better illustrate the proposed DFS framework based on scan chain reshuffling, we list part of the Coq netlist converted from the netlist equipped with shuffled scan chain. From the sample code below we can find that scan chain is connected in the sequence of `sa00_0` → `text_in_127` → `text_in_126` → ... → `text_in_123` because we know that the internal signal `sa00` is of lower sensitivity level than the plaintext input `text_in`. All security theorems, which are also listed below, are proved given the shuffled scan chain enhanced netlist for all primary outputs.

```
(* Coq Representative of Post-SecScan Netlist *)
Definition aes : code :=
assign_b N505 (ld);
nonblock_assign_ex text_in_r_127 (ECOND_EX n790
  (ECOND_B test_se sa00_0 text_in_127)
    (ESIG text_in_r_127));
nonblock_assign_ex text_in_r_126 (ECOND_EX n790
  (ECOND_B test_se text_in_r_127 text_in_126)
    (ESIG text_in_r_126));
nonblock_assign_ex text_in_r_125 (ECOND_EX n790
  (ECOND_B test_se text_in_r_126 text_in_125)
    (ESIG text_in_r_125));
nonblock_assign_ex text_in_r_124 (ECOND_EX n790
  (ECOND_B test_se text_in_r_125 text_in_124)
    (ESIG text_in_r_124));
nonblock_assign_ex text_in_r_123 (ECOND_EX n792
  (ECOND_B test_se text_in_r_124 text_in_123)
    (ESIG text_in_r_123));
......

(* Data secrecy property theorems *)
(* Stability *)
Definition aes_stable_list :=
    check_sensitivity 8 aes aes_initial_list.
Lemma aes_sen_stable : update_sensitivity
        aes aes_stable_list = aes_stable_list.

(* Accessibility *)
Theorem stable_list_accessability : forall t : nat,
        t > 8 ->  (check_sensitivity t aes
            aes_initial_list) = aes_stable_list.

(* Trustworthiness *)
Theorem no_leaking_1 :
        nth done aes_stable_list 0 = 0.
Theorem no_leaking_2 :
        nth text_out_0 aes_stable_list 0 = 0.
......
Theorem no_leaking_N :
        nth text_out_127 aes_stable_list 0 = 0.
```

Even though the reshuffled scan chain prevents the construction of leakage paths to primary outputs, the proposed method cannot prohibit data leaking through the scan out pin. In fact, we found that the sensitivity level of scan out `test_so` is 2, meaning that the scan out pin carries sensitive data. In order to ensure that the underlying circuit achieves the same security level as the original design, further protection methods should be applied to prohibit illegal access to the scan out pin, such as scan chain disabling after implementation[2].

```
Definition aes_stable_list :=
    check_sensitivity 8 aes_DFS aes_initial_list.
Compute (nth test_so aes_stable_list 0).
     = 2
     : sense
```

VI. Conclusions

The widely accepted DFT techniques in IC testing area also introduce security concerns. In order to evaluate the security of inserted DFT structures at the early stage of design flow, a formal DFT techniques trustworthiness assessment method has been proposed. Using an AES encryption core as the testing platform, we formally evaluated the trustworthiness of DFT techniques, including DFT scan chains and BIST structures, with the results showing that the insertion of scan chain and some BIST structures will insert security vulnerabilities to the target design. To preserve the testability of the DFT techniques and ensure no leakage paths to primary outputs would be imposed, a design-for-security (DFS) method based on scan chain reshuffling was developed to balance the testability and security of integrated circuits.

[2]We want to emphasize that the major difference between the original DFT structure and the enhanced DFT structure is that no information leakage paths are available through primary outputs in the enhanced structure.

Acknowledgements

This work was supported by the National Science Foundation (NSF-1319105).

References

[1] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing For Digital, Memory, And Mixed-Signal VLSI Circuits*. Norwell, MA: Kluwer, 2000.
[2] B. Yang, K. Wu, and R. Karri, "Scan based side channel attack on dedicated hardware implementations of data encryption standard," in *Test Conference, 2004. Proceedings. ITC 2004. International*, 2004, pp. 339–344.
[3] R. Nara, N. Togawa, M. Yanagisawa, and T. Ohtsuki, "Scan-based attack against elliptic curve cryptosystems," in *Proceedings of the 2010 Asia and South Pacific Design Automation Conference*, 2010, pp. 407–412.
[4] B. Yang, K. Wu, and R. Karri, "Secure scan: A design-for-test architecture for crypto chips," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 25, no. 10, pp. 2287–2293, 2006.
[5] D. Hély, F. Bancel, M.-L. Flottes, and B. Rouzeyre, "A secure scan design methodology," in *Proceedings of the conference on Design, automation and test in Europe: Proceedings*, 2006, pp. 1177–1178.
[6] G. Sengar, D. Mukhopadhyay, and D. Chowdhury, "Secured flipped scan-chain model for crypto-architecture," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 11, pp. 2080–2084, 2007.
[7] J. Lee, M. Tehranipoor, C. Patel, and J. Plusquellic, "Securing designs against scan-based side-channel attacks," *Dependable and Secure Computing, IEEE Transactions on*, vol. 4, no. 4, pp. 325–336, 2007.
[8] S. Paul, R. Chakraborty, and S. Bhunia, "Vim-scan: A low overhead scan design approach for protection of secret key in scan-based secure chips," in *VLSI Test Symposium, 2007. 25th IEEE*, 2007, pp. 455–460.
[9] J. Da Rolt, G. Di Natale, M. L. Flottes, and B. Rouzeyre, "Are advanced dft structures sufficient for preventing scan-attacks?" in *VLSI Test Symposium (VTS), 2012 IEEE 30th*, 2012, pp. 246–251.
[10] J. Rolt, A. Das, G. Natale, M.-L. Flottes, B. Rouzeyre, and I. Verbauwhede, "A new scan attack on rsa in presence of industrial countermeasures," in *Constructive Side-Channel Analysis and Secure Design*, ser. Lecture Notes in Computer Science, W. Schindler and S. Huss, Eds. Springer Berlin Heidelberg, 2012, vol. 7275, pp. 89–104.
[11] M. Doulcier, M. L. Flottes, and B. Rouzeyre, "AES-based BIST: Self-test, test pattern generation and signature analysis," in *Electronic Design, Test and Applications, DELTA 2008. 4th IEEE International Symposium on*, 2008, pp. 314–321.
[12] G. Di Natale, M. Doulcier, M. L. Flottes, and B. Rouzeyre, "Self-test techniques for crypto-devices," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 18, no. 2, pp. 329–333, 2010.
[13] G. C. Necula, "Proof-carrying code," in *POPL '97: Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 1997, pp. 106–119.
[14] S. Drzevitzky, U. Kastens, and M. Platzner, "Proof-carrying hardware: Towards runtime verification of reconfigurable modules," in *International Conference on Reconfigurable Computing and FPGAs*, 2009, pp. 189–194.
[15] E. Love, Y. Jin, and Y. Makris, "Proof-carrying hardware intellectual property: A pathway to trusted module acquisition," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 25–40, 2012.
[16] Y. Bertot and P. Castean, *Interactive theorem proving and program development: Coq'Art: the calculus of inductive constructions*. Springer, 2004.
[17] Y. Jin, B. Yang, and Y. Makris, "Cycle-accurate information assurance by proof-carrying based signal sensitivity tracing," in *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2013, pp. 99–106.
[18] Y. Jin, "EDA tools trust evaluation through security property proofs," in *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*, 2014, pp. 1–4.