HARDWARE TROJAN IDENTIFICATION AND DETECTION

Samer Moein[1], Fayez Gebali[1], T. Aaron Gulliver[1], And Abdulrahman Alkandari[2]

[1]Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada

[2]Department of Computer Science, Public Authority for Applied Education and Training, Kuwait City, Kuwait

ABSTRACT

The majority of techniques developed to detect hardware trojans are based on specific attributes. Further, the ad hoc approaches employed to design methods for trojan detection are largely ineffective. Hardware trojans have a number of attributes which can be used to systematically develop detection techniques. Based on this concept, a detailed examination of current trojan detection techniques and the characteristics of existing hardware trojans is presented. This is used to develop a new approach to hardware trojan identification and classification. This identification can be used to compare trojan risk or severity and trojan detection effectiveness. Identification vectors are generated for each hardware trojan and trojan detection technique based on the corresponding attributes. Vectors are also defined which represent trojan risk or severity and trojan detection effectiveness.

KEYWORDS

Hardware trojan detection, hardware trojan identification, trojan severity, trojan risk, trojan detection effectiveness

1. INTRODUCTION

With the increasing globalization of Integrated Circuit (IC) design and production, hardware trojans have become a serious threat to manufacturers as well as consumers. The use of ICs in critical applications makes the effects of these trojans a very dangerous problem. Unfortunately, the use of untrusted foundries and design tools cannot be eliminated since the complexity of ICs and the sophistication of their manufacture has grown significantly. Establishing a trusted foundry for fabrication is beyond the capabilities of most IC producers. Therefore, it is essential that effective hardware trojan detection techniques be developed.

A *hardware trojan* is defined as a malicious component embedded in an IC which causes abnormal behavior [1]. Hardware trojans can be implemented in microprocessors, microcontrollers, network and digital signal processors, Field Programmable Gate Arrays (FPGAs), Application Specific Integrated Circuits (ASICs), and other ICs. Figure 1 presents the classification of hardware trojan detection techniques proposed in [2]. They can be classified as destructive or non-destructive. Destructive techniques (i.e. reverse engineering), are primarily used to obtain a trojan free chip, referred to as a Golden Chip (GC), and can be extremely expensive and time consuming [3]. Therefore, it is often not practical to test chips using destructive techniques. Further, Process Variations (PVs) can result in false positives for trojan free chips when they are compared to a GC, and testing only a portion of the chips may be ineffective as an adversary can insert a trojan in only a small percentage of the chips.

DOI:10.5121/ijcis.2017.7201

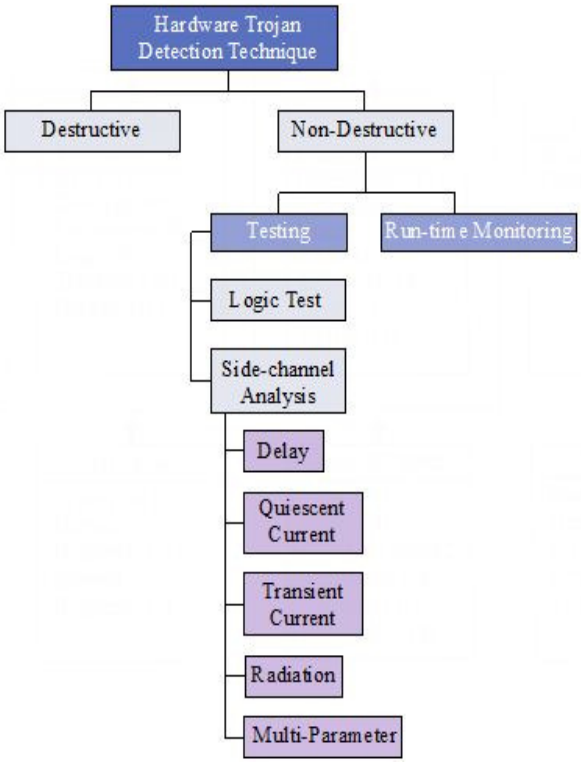

Figure 1: Existing Hardware Trojan Detection Classification

Non-destructive techniques can be classified as testing or run-time monitoring methods. Testing can be supported by design for security circuits, e.g. scan chains and self-test circuitry. This improves trojan detection effectiveness but requires that the test circuitry not be compromised. A trojan may not be triggered during testing or it may be designed to avoid activation until after testing is completed. Therefore, a trojan that does not change the chip layout or design can be very difficult to detect during testing.

Testing approaches can be classified as logic testing or side-channel analysis. Logic testing methods employ random test vectors in an attempt to activate trojan circuits and observe their effect at the chip outputs. The difficulty with this approach is the complexity of testing all internal nodes and logic values as chips can have very high gate densities which makes comprehensive testing intractable. Side-channel analysis is based on the fact that any modification to a chip should be reflected in parameters such as the dynamic power [4–8], leakage current [9,10], path-delay characteristics [11–13], Electromagnetic (EM) radiation [14], or a combination of these parameters [15, 16]. However, side-channel techniques suffer from sensitivity to errors due to PVs and noise. This creates false positives and allows infected chips to go undetected. A good detection technique should have a high probability of detecting an infected chip and a low false positive probability. The advantage of side-channel techniques over logic-testing approaches is not having to activate a trojan to detect it. For example, parametric or inactive trojans require an internal or external trigger to become active.

Side-channel techniques are commonly employed and are very effective when ICs have low complexity and are not dense. However, detecting small or distributed trojans in complex or dense chips can be a significant challenge. For this reason, trojan circuits are typically very small

compared to the IC design. They are often inserted in blank areas in the chip layout during the fabrication phase or implemented by rewiring existing circuitry.

Run-time monitoring is used to continuously monitor chip operation to detect the effects of malicious circuitry and initiate mitigation techniques. This can be achieved by exploiting pre-existing circuit redundancy such as a reconfigurable core [17] in a multicore system [18] to avoid infected parts of the circuit [2]. However, this can increase the chip area and delay leading to reduced performance. Run-time monitoring approaches greatly improve chip reliability when trojans pass the test phase [19]. Another approach is to use self-destructive packaging to disable chips or discard the output when a trojan is detected.

The high complexity of chips and the effects of PVs make many detection techniques proposed in the literature ineffective. Therefore, new techniques must be developed or approaches combined to improve performance. Most detection methods have been developed for a trojan designed specifically to test the effectiveness of the technique. This is problematic as even a small change in the trojan circuitry can result in detection failure. A better approach is to systematically examine the properties of existing trojans and design detection techniques based on the results of this investigation. This is important, as designing a detection technique that can protect against multiple trojans is a challenging task.

The contributions of this paper are as follows:

1. The relationships between trojan attributes are examined and values assigned to each attribute to indicate the associated risk and effectiveness of detection techniques.

2. Identification vectors are determined for hardware trojans and trojan detection techniques which represent the corresponding trojan attributes.

3. Vectors are also given to represent trojan severity or risk and detection effectiveness, respectively.

The remainder of this paper is organized as follows. Section 2 reviews hardware trojan attributes and existing hardware trojan detection techniques to illustrate the proposed approach to trojan identification. The trojan attributes are studied and risk and detection effectiveness values are assigned in Section 3. Section 4 presents examples of hardware trojan vectors, and Section 5 gives examples of hardware trojan detection vectors. Finally, Section 6 provides some concluding remarks.

2. HARDWARE TROJANS AND THEIR DETECTION

Any attempt to address hardware security concerns should begin with a classification of the threats based on the processes involved in the IC production life cycle. A comprehensive model for trojan classification was presented in [20] which is based on eight categories: insertion, abstraction, effect, logic type, functionality, activation, physical layout, and location. A classification of attributes based on this model is illustrated in Figure 2. The relationships between these attributes were presented in [21] and used to identify the attributes that can be detected using a given technique. For example, a technique that can detect a sequential trojan circuit can also detect a combinational trojan circuit, or a technique that can detect a small trojan can also detect a large trojan. Methods used to detect a trojan in one category may also be useful in detecting trojans in other categories. As an example, a trojan introduced during the specification phase may affect attributes in other categories, so to be effective a technique should detect if a specification attribute has been compromised. In this paper, hardware trojans and hardware trojan detection techniques are examined based on the corresponding trojan attributes and their relationships.

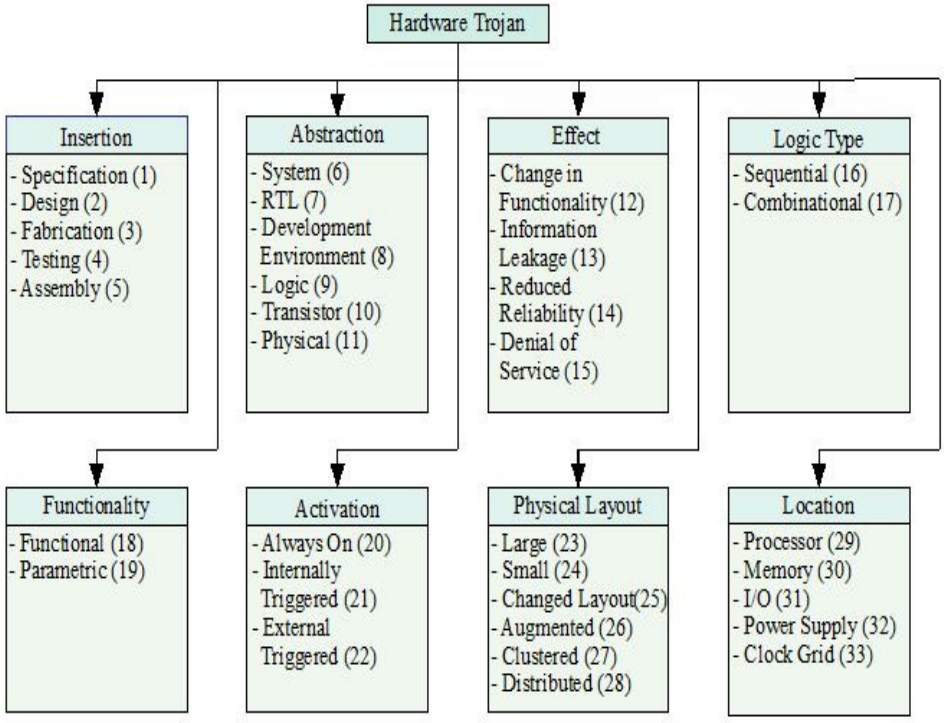

Figure 2: Classification of Hardware Trojan Attributes

Most hardware trojan detection techniques are based on side-channel analysis. However, noise and PVs can affect the accuracy of side-channel information. Thus, multiple parameters are typically measured to improve the detection performance [15]. For example, in [8] power consumption and delay were measured and combined with gate level characteristics. The use of multiple techniques is shown as combined and hybrid blocks in Figure 3. This approach to trojan detection was presented in [30]. Table 1 [30] provides a summary of the attributes for the detection techniques considered in this paper. Each technique can detect hardware trojans with certain attributes. The letter C indicates that a technique can protect against the attribute, M means the technique may provide protection, while an empty entry means it cannot protect against trojans with this attribute. In addition, a technique may require results from a golden chip to compare with measurements from a Chip Under Test (CUT). This is indicated by R in the GC column. If a technique considers PVs, this is indicated by C in the PV column.

3. HARDWARE TROJAN ATTRIBUTES

A comprehensive investigation of hardware trojan attributes was presented in [20]. These attributes provide a complete characterization of trojans. In [21], the relationships between the attributes were studied and used to describe the trojan life cycle from the insertion phase to the location phase. Assigning weights to these attributes was also considered. The goal was to identify related combinations of attributes and exploit these connections for detection purposes.

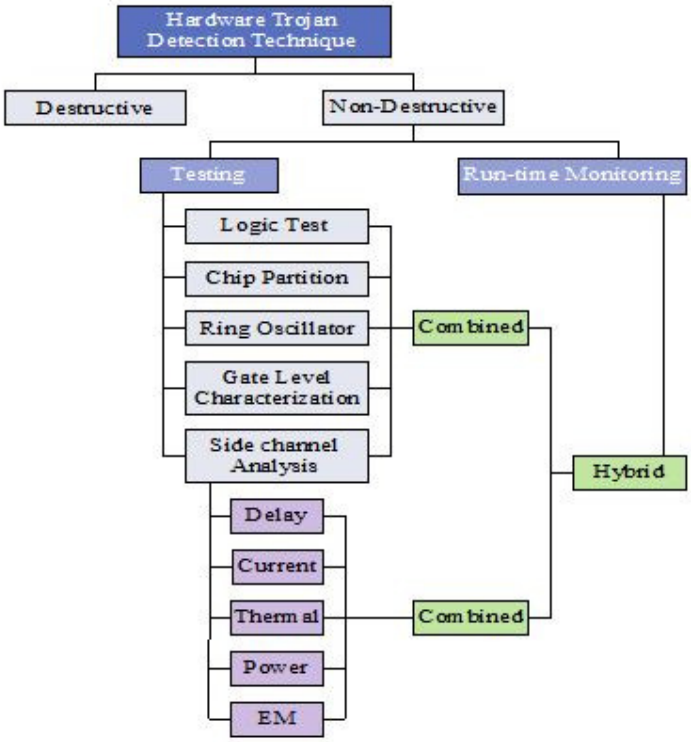

Figure 3: Proposed Hardware Trojan Detection Classification

Table 1: Classification of Hardware Trojan Detection Techniques

	Attributes																																		
	Insertion					Abstraction						Effect				Type	Functionality			Activation			Physical Layout						Location						Chip Attributes	
Techniques	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	GC	FV
[4]		C			C						C	C	C		C	C	C	C	C	C			C	C		C	C		C	C	C	M	M		C
[5]				C		C							C				C		C	C				C		C	C		C	M	C	M	M		C
[6]	M		C			M					C		C				C		C	C				C		C	C		C	C	C	M	M	R	C
[7]			C						C			C					C	C		C	C		C	C		C	C		C	M	M	M	M	R	
[8]		C	C					C	C	C	C	C	C	C		C	C	C	C	C	C		C	C		C	C		C	M	M	M	M		
[9]		C							C			C				C		C			C		C	C	C	C	C	C	M	C	M	M	R	C
[10]		C	C				C	C	C			C				C	C	C	C	C	C		C	C		C	C	C	C	M	M	M	M	R	C
[11]			C					C				C	C			C	C	C	C	C	C		C	C	C		C		C	M	C	M	M	R	C
[12]		C	C					C	C					C	C		C		C	C			C	C	C	C	C	M	C	M	M	M	M		C
[13]		C	C					C	C		C	C	C				C	C	C	C			C	C	C	C	C	C	C	M	M	M	M	R	
[14]		C						C	C						C	C		C				C	C	C	C	C	C	C	M	M	C	M	M	R	
[15]			C				C		C			C	C			C	C	C	C		C		C	C	C	C	C	C	C	M	M	M	M	R	C
[16]			C						C			C				C	C	C		C	C		C	C		C	C		C	M	M	M	M	R	C
[19]			C					C				C		C		C	C	C		C	C	C		C		C		C	C	C	C			R	C
[22]		C	C				C	C	C			C	C			C		C			C		C	C	C	C	C	C	C	M	M	M	M		C
[23]		C							C		C	M	M			C	C	C	M	C	C		C	C		C	C	C	C	M	M	M	M	R	C
[25]	C	C					C								C	C	C	C			C		C		C	C	C		C	M	M	M	M	R	C
[26]		C	C				C					C				C	C			C			C	C	C	C	C	C	C	M	M	M	M	R	C
[27]		C	C						C	C		C			C		C	C		C	C		C	C	C	C	C	C	C	M	M	M	M	R	C
[28]		C	C				C	C	C				C		C	C	C	C		C	C	C	C		C		M	M	C	M	C	M	M	R	
[29]		C	C					C	C			C				C	C	C			C		C	C		C	C		C	M	M	M	M		

Each hardware trojan has a number of attributes, and detection techniques have been developed to detect trojans with some or all of these attributes. However, the combinations of attributes that can lead to detection have not been considered. In this paper, the trojan attributes are examined in detail, and they are ranked within each category.

The attributes in each category are used to identify trojans and evaluate their risk or severity. They can also be used to identify trojan detection techniques and evaluate their effectiveness. Each trojan is assigned two vectors I_T and C_T. The first identifies the corresponding attributes, while the second presents the attributes in terms of their risk or severity. Each trojan detection technique also has two vectors I_D and C_D. I_D identifies the attributes that can be detected, while C_D presents the attributes in terms of the effectiveness of the technique. Thus there are four vectors corresponding to trojan identification I_T, trojan detection identification I_D, trojan risk or severity C_T, and trojan detection effectiveness C_D.

A hardware trojan is identified by the following vector based on the attributes involved in the trojan

$$I_T = I_R I_A I_E I_L I_F I_C I_P I_O, \tag{1}$$

where $\{I_R, I_A, I_E, I_L, I_F, I_C, I_P, I_O\}$ represents the {Insertion, Abstraction, Effect, Logic type, Functionality, Activation, Physical layout, Location} category values, respectively. Each value specifies the attributes involved in the trojan within the category.

The attributes that identify a trojan detection technique is given by the vector

$$I_D = I_T I_G, \tag{2}$$

where $\{I_T, I_G\}$ represents the {Trojan parameters, Chip attribute} category values, respectively. I_G is used to specify if the technique requires a golden chip and/or considers process variations. Each value specifies the attributes that the detection technique can be used against.

The risk level of a trojan is given by the vector

$$C_T = C_R C_A C_E C_L C_F C_C C_P C_O, \tag{3}$$

where $\{C_R, C_A, C_E, C_L, C_F, C_C, C_P, C_O\}$ represents the {Insertion, Abstraction, Effect, Logic type, Functionality, Activation, Physical layout, Location} category risk values, respectively.

The effectiveness of a trojan detection technique is given by the vector

$$C_D = C_T C_G, \tag{4}$$

where $\{C_T, C_G\}$ represents the {Trojan parameters, Chip attribute} category effectiveness values, respectively. C_G determines if a trojan detection technique requires a golden chip and/or considers process variations.

The values for each category are determined as described in the following sections.

3.1. Insertion (R) Category

The insertion category consists of five attributes: specification (1), design (2), fabrication (3), testing (4), and assembly (5).

3.1.1. Insertion Identification (IR)

The identification parameters for the attributes in this category are based on the fact that starting from specification, the existence of a trojan in an attribute may affect subsequent attributes in the category. Thus, the number of attributes that may be affected is used for identification. Column I_R in Table 2 shows that the value 5 is assigned to the specification attribute, which means that a trojan inserted during the specification phase may affect all other attributes. Further, a trojan inserted in the design phase can affect the three subsequent attributes.

3.1.2. Insertion Risk/Effectiveness (CR)

The risk and effectiveness values for the attributes in this category are based on the effect on subsequent attributes. For example, a trojan inserted in the specification phase can affect the design and propagate through the remaining sequence of attributes so that fabrication, testing, and assembly may also be affected. Table 2 gives the insertion attributes with the corresponding identification I_R and risk/effectiveness C_R values.

From a trojan perspective, I_R = 3 means that the trojan is inserted during fabrication, and C_R = 3 means that the fabrication attribute is infected, and both the testing and assembly attributes may also be infected. From a trojan detection perspective, I_R = 3 means that the detection technique can detect a trojan inserted during the fabrication phase, and C_R = 3, means that detecting a trojan that has infected the fabrication attribute also protects the chip from trojans in the remaining attributes within the category.

Therefore, the I_R and C_R ranges for the attributes within the insertion category are

$$1 \leq I_R \leq 5, \quad 1 \leq C_R \leq 5 \tag{5}$$

Table 2: Insertion Attribute Values

K	Attribute	I_R	C_R
1	Specification	5	5
2	Design	4	4
3	Fabrication	3	3
4	Testing	2	2
5	Assembly	1	1

3.2. Abstraction (A) Category

The abstraction category consists of six attributes: system (6), RTL (7), development environment (8), logic (9), transistor (10), and physical (11).

3.2.1. Abstraction Identification (IA)

The identification parameters for the attributes in this category are based on the fact that starting from system, the existence of a trojan in an attribute may affect subsequent attributes within the category. Thus, the number of attributes that may be affected is used for identification. Column I_A in Table 3 shows that the value 6 is assigned to the system attribute, which means that a trojan inserted at the system level may affect all other attributes. Further, a trojan inserted in the RTL can affect the 5 subsequent attributes.

3.2.2. Abstraction Risk/Effectiveness (CA)

The risk and effectiveness values for the attributes in this category are also based on the effect on subsequent attributes. For example, a trojan inserted at the system level can affect the RTL and thus also the development environment, logic, transistor, and physical attributes.

Table 3 shows the abstraction attributes with their assigned values. From a trojan perspective, I_A = 5 means that the trojan is inserted at the RTL level, and C_A = 5 means that if the RTL attribute is infected, all subsequent levels (development environment, logic, transistor, and physical), may also be infected. From the trojan detection perspective, I_A = 5 means that the detection technique can detect a trojan inserted at the RTL level, and C_A = 5 means that a trojan inserted at the RTL level can also be detected in the remaining attributes within the abstraction category.

The I_A and C_A ranges for the attributes within the abstraction category are

$$1 \leq I_A \leq 6, \quad 1 \leq C_A \leq 6 \tag{6}$$

Table 3: Abstraction Attribute Values

K	Attribute	I_A	C_A
6	System	6	6
7	RTL	5	5
8	Development Environment	4	4
9	Logic	3	3
10	Transistor	2	2
11	Physical	1	1

3.3. Effect (E) Category

The effect category consists of four attributes: changes in functionality (12), information leakage (13), reduced reliability (14), and denial of service (15).

3.3.1. Effect Identification (IE)

For n trojan effect attributes, there are 2^n-1 different combinations. From [21], n=4 different effects are considered, so there are 15 combinations. These combinations are assigned the values 1 to F to uniquely identify them. Column I_E in Table 4 shows that the value 5 identifies a trojan that can change the system functionality and leak information from the system.

3.3.2. Effect Risk/Effectiveness (CE)

The effect category attributes are assigned severity values based on the affected system, and thus can differ between systems. Thus, the values assigned here are for illustration purposes only. If the system is located in a government agency, information leakage is the most critical attribute in this category. This is followed by change in functionality and denial of service, and then reduced reliability. Based on these assumptions, $C_E = \{2, 4, 1, 2\}$ are assigned to {Change in functionality, Information leakage, Reduced reliability, Denial of service}, respectively, and these are shown in Table 4.

A trojan with a combination of attributes within this category has a severity which is the sum of the corresponding severity values. For example, column C_E in Table 4 contains the value 5 twice. The first is when a trojan has the effects information leakage and reduced reliability so that C_E (13) + C_E (14) = 4 + 1 = 5, while the second is when a trojan has the effects change in functionality, reduced reliability, and denial of service so that $C_E(12) + C_E(14) + C_E(15) = 2 + 1 + 2 = 5$. In terms of identification, the first of these combinations is assigned I_E = 8 while the second is assigned $I_E = D$.

Table 4 shows the effect attributes with their assigned values. For example, $I_E = C$ and C_E = 8 means that from a trojan perspective, it has the effects change in functionality, information leakage, and denial of service. From a trojan detection perspective, $I_E = C$ means that the detection technique can detect a trojan that has the attributes change in functionality, information leakage, and denial of service, and C_E = 8 means that the technique can detect a trojan with the effects change functionality, information leakage, and denial of service.

The I_E and C_E ranges for the attributes within the effect category are

$$1 \le I_E \le F,$$
$$1 \le C_E \le 9 \quad (7)$$

Table 4: Effect Attribute Values

K	Attribute	I_E	C_E
12	Change in Functionality	1	2
13	Information Leakage	2	4
14	Reduced Reliability	3	1
15	Denial of Service	4	2
12 & 13		5	6
12 & 14		6	3
12 & 15		7	4
13 & 14		8	5
13 & 15		9	6
14 & 15		A	3
12 &13 & 14		B	7
12 &13 & 15		C	8
12 &14 & 15		D	5
13 &14 & 15		E	7
12 &13 & 14 & 15		F	9

3.4. Logic Type (L) Category

The logic type category contains two attributes: sequential (16), and combinational (17).

3.4.1. Logic Type Identification (IL)

A trojan can have one of these attributes or both, so three values are needed for this category. As shown in Table 5, I_L = 1 denotes a combinational logic trojan, I_L = 2 denotes a sequential logic trojan, and I_L = 3 denotes a trojan with both combinational and sequential logic.

3.4.2. Logic Type Risk/Effectiveness (CL)

Sequential logic consists of combinational logic and memory, so combinational logic is contained in sequential logic. Thus, a sequential logic trojan is more dangerous than a combinational logic trojan because there are more factors that can be used for activation. These factors are unknown and unexpected, while a combinational logic trojan is always on which makes detection easier. Therefore, here the sequential logic attribute is assigned a severity value twice that of the combinational logic attribute, as shown in Table 5. These values can be modified based on the particular circumstances, but the value for sequential logic should be higher. In Table 5, C_L = 1 for a combinational logic trojan, C_L = 2 for a sequential logic trojan, and C_L = 3 for a trojan designed with both logic types.

The I_L and C_L ranges for the attributes within the logic type category are

$$1 \le I_L \le 3,$$
$$1 \le C_L \le 3 \quad (8)$$

Table 5: Logic Type Attribute Values

K	Attribute	I_L	C_L
16	Sequential	2	2
17	Combinational	1	1
16 & 17	Both	3	3

3.5. Functionality (F) Category

The functionality category consists of two attributes: functional (18), and parametric (19).

3.5.1. Functionality Identification (IF)

This category consists of two attributes, and a trojan can be designed to have one or both types. Therefore, to identify the attributes within this category, three different values are needed. As shown in Table 6, I_F = 1 is used to identify a functional trojan, I_F = 2 is used to identify a parametric trojan, and I_F = 3 is used to identify a trojan which is both parametric and functional.

3.5.2. Functionality Risk/Effectiveness (CF)

Both attributes have different capabilities, but parametric trojans can affect system operations, which means that they include functional trojan effects. Further, some parametric trojans are designed to leak sensitive information without affecting the system functionality, which makes them more dangerous than functional trojans. In fact, a trojan designed with both features will be more dangerous. Therefore the severity value for the parametric attribute should be higher than the value for the functional attribute, and a trojan with both attributes should have the highest severity, and this is reflected in Table 6.

hese values are only for illustration purposes and others can be assigned based on the system risks associated with the attributes. From a trojan perspective, I_F = 2 means that a parametric trojan has been inserted, while C_F = 2 means that if a parametric trojan is inserted, it has a severity of 2 out of a maximum of 3. From a trojan detection perspective, I_F = 2 means that the trojan detection technique can detect a parametric trojan, while C_F = 2 means that if a detection technique is designed to detect only parametric trojans, the effectiveness is 2 out of a maximum of 3.

The I_F and C_F ranges for the attributes within the functionality category are

$$1 \leq I_F \leq 3,$$
$$1 \leq C_F \leq 3 \quad (9)$$

Table 6: Functionality Attribute Values

K	Attribute	I_F	C_F
18	Functional	1	1
19	Parametric	2	2
18 &19	Both	3	3

3.6. Activation (C) Category

The activation category consists of three attributes: always on (20), internally triggered (21), and externally triggered (22).

3.6.1. Activation Identification (IC)

There are three different activation mechanisms, giving 7 possible combinations. A different value is assigned to each combination to uniquely identify them, as shown in Table 7. This shows that a value of 6 in column I_C identifies a trojan that can be internally or externally activated.

3.6.2. Activation Risk/Effectiveness (CC)

The risk (severity) of the activation category attributes depends on the system affected, and so can differ between systems. Thus, values are assigned here only for illustration purposes. It is assumed that a trojan which is externally triggered is the hardest to detect because it is unlikely to be activated during testing. An internally triggered trojan is more likely to be detected during testing as this can occur accidentally. Further, an always on trojan that is not triggered is the easiest to detect. Based on these assumptions, C_C = {4, 2, 1} is used to represent {Externally triggered, Internally triggered, Always on}, respectively, as shown in Table 7.

A trojan with a combination of these attributes has a severity equivalent to the sum of the values for the attributes. For example, the value 5 in column C_C in Table 7 indicates that a trojan is always on but activated externally, as $C_C(20) + C_C(22) = 1 + 4 = 5$. From a trojan perspective, $I_C = 4$ means that the trojan is always on and activated internally, while $C_C = 3$ means that if the trojan is always on and activated internally, it has a severity of 3 out of a maximum of 7. From a trojan detection perspective, $I_C = 4$ means that the technique can detect a trojan if it is always on and activated internally. $C_C = 3$ means that if a detection technique is designed to detect always on trojans that may or may not be internally triggered, the effectiveness is 3 out of a maximum of 7.

The I_C and C_C ranges for the attributes within the activation category are

$$1 \le I_C \le 7,$$
$$1 \le C_C \le 7 \qquad (10)$$

Table 7: Activation Attribute Values

K	Attribute	I_C	C_C
20	Always On	1	1
21	Internally Triggered	2	2
22	Externally Triggered	3	4
20 & 21		4	3
20 & 22		5	5
21 & 22		6	6
20 & 21 & 22		7	7

3.7. Physical Layout (P) Category

The physical layout category consists of six attributes: large (23), small (24), changed layout (25), augmented (26), clustered (27), and distributed (28).

3.7.1. Physical Layout Identification (IP)

It is clear that there are related attributes within this category. For example, if a detection technique is able to detect a small trojan then it is also able to detect a large one. Further, a trojan can be classified as either small or large. This argument also applies to changed layout and augmented, and clustered and distributed. A trojan has one attribute from each of these groups, so the individual attributes are not assigned identification values, as shown in Table 8. Eight different combinations of attributes need to be identified, so the values 1 to 8 are used in the table. For example, the value 4 in column I_P in Table 8 indicates a large trojan inserted without changing the chip layout which is distributed throughout the chip.

3.7.2. Physical Layout Risk/Effectiveness (CP)

A trojan has three attributes related to the three groups within this category. It is clear that a small trojan is harder to detect than a large one. If the chip layout is changed, it is a strong indication that the chip is infected. On the other hand, if the layout is not changed it may still be infected, so additional effort is required to decide if a chip is trojan free or infected. Further, a distributed trojan is more dangerous than a clustered one.

For example, a trojan may be large but distributed so it is a number of small circuits, each of which can have different effects and activation mechanisms. Therefore, the small, augmented, and distributed attributes have greater risk than the large, changed layout, and clustered attributes, so $C_P = \{1, 2, 1, 2, 1, 2\}$ are assigned to {Large, Small, Changed layout, Augmented, Clustered, Distributed} respectively, as shown in Table 8. Every trojan has a combination of three attributes (one from each group), so the severity for a trojan is the sum of the values for the associated

attributes. For example, the value of 6 in column C_P in Table 8 indicates that a small trojan has been inserted without changing the layout and is distributed throughout the chip $C_P(24) + C_P(26) + C_P(28) = 2 + 2 + 2 = 6$. It is clear that a small, augmented, and distributed trojan is the worst case for this category.

From a trojan perspective, $I_P = 3$ means a large, clustered trojan has been inserted without changing the chip layout, while $C_P = 3$ means a large, clustered trojan has been inserted, and it has changed the chip layout. From a trojan detection perspective, $I_P = 3$ means that the trojan detection technique can detect a large, clustered trojan even if it has not changed the chip layout, while $C_P = 3$ means that if a detection technique is designed to detect a large, clustered trojan that changed the chip layout, it has an effectiveness of 3 out of a maximum of 6.

The I_P and C_P ranges for the attributes within the physical layout category are

$$1 \leq I_P \leq 8, \quad 3 \leq C_P \leq 6 \tag{11}$$

Table 8: Physical Layout Attribute Values

K	Attribute	I_P	C_P
23	Large	-	1
24	Small	-	2
25	Changed Layout	-	1
26	Augmented	-	2
27	Clustered	-	1
28	Distributed	-	2
23 & 25 & 27		1	3
23 & 25 & 28		2	4
23 & 26 & 27		3	4
23 & 26 & 28		4	5
24 & 25 & 27		5	4
24 & 25 & 28		6	5
24 & 26 & 27		7	5
24 & 26 & 28		8	6

3.8. Location (O) Category

The location category consists of five attributes: processor (29), memory (30), I/O (31), power supply (32), and clock grid (33).

3.8.1. Location Identification (IO)

The five possible trojan locations result in 31 possible combinations, and each is assigned a unique identifier from 1 to V. A trojan is inserted in one of these locations, while a trojan detection technique may have the ability to detect a trojan in more than one location. The first five values 1 to 5 are used to identify single locations. The remainder are used only with detection techniques to indicate in which locations a trojan can be detected. The value 2 in column I_O in Table 9 means that the trojan is located in the memory and the detection technique can detect a trojan in the memory. $I_O = F$ indicates that the detection technique can detect a trojan located in the power supply or clock grid.

3.8.2. Location Risk/Effectiveness (CO)

In general, each particular location attribute has similar risk, so they are assigned $C_O = 1$. The value of C_O is defined as the number of locations affected by the trojan. For example, a trojan inserted in the I/O may receive commands (external trigger) to activate some processor actions or leak data stored in memory, so it is assigned $C_O = 2$.

For detection techniques, the effectiveness is defined as the number of locations in which a trojan can be detected, e.g. the value 2 in column C_O in Table 9 indicates that a technique can detect a trojan inserted in two different locations. The specific locations are given by I_O. For example, if $I_O = 8$, the technique can detect a trojan in the processor or power supply. From a trojan perspective, $I_O = 3$ means that it is inserted in the I/O, while $C_O = 5$ means that the trojan affects all five locations, namely processor, memory, I/O, power supply, and clock grid.

From a trojan detection perspective, $I_O = 3$ means that the technique can detect a trojan inserted in the I/O only, while $C_O = 5$ means that it can detect a trojan in the processor, memory, I/O, power supply, or clock grid, which is the maximum effectiveness.

The I_O and C_O ranges for the attributes within the physical layout category are

$$1 \leq I_O \leq V,$$

$$1 \leq C_O \leq 5 \tag{12}$$

3.9. Chip Attribute (G) Category

The chip attribute category consists of two attributes: golden chip (GC) and process variation (PV). This category only pertains to detection techniques.

3.9.1. Chip Attribute Identification (IG)

The two attributes result in 4 possible combinations, so the values 0 to 3 are used to uniquely identify them. The value of 3 in column I_G in Table 10 indicates that the detection technique requires a golden chip (GC) and considers process variations (PVs).

3.9.2. Chip Attribute Effectiveness (CG)

Obtaining reference measurements using a golden chip is an expensive process that requires reverse engineering to ensure that the chip is trojan free. Process variations are important to consider as they can affect detection reliability, resulting in false positives and false negatives. Therefore, a detection technique that can detect a trojan without the need for a golden chip but considers the process variations is the best, and so is assigned $C_G = 4$. Conversely, a detection technique that requires a golden chip but does not consider process variations is the most expensive and least effective, and so is assigned $C_G = 1$..

A detection technique that neither requires a golden chip nor considers process variations is assigned $C_G = 2$, while a technique that requires a golden chip but considers process variations is assigned $C_G = 3$. The I_G and C_G ranges for the attributes within the activation category are

$$0 \leq I_G \leq 3,$$

$$1 \leq C_G \leq 4 \tag{13}$$

Table 9: Location Attribute Values

K	Attribute	I_o	C_o
29	Processor	1	1
30	Memory	2	1
31	I/O	3	1
32	Power Supply	4	1
33	Clock Grid	5	1
29 & 30		6	2
29 & 31		7	2
29 & 32		8	2
29 & 33		9	2
30 & 31		*A*	2
30 & 32		*B*	2
30 & 33		*C*	2
31 & 32		*D*	2
31 & 33		*E*	2
32 & 33		*F*	2
29 & 30 & 31		*G*	3
29 & 30 & 32		*H*	3
29 & 30 & 33		*I*	3
29 & 31 & 32		*J*	3
29 & 31 & 33		*K*	3
29 & 32 & 33		*L*	3
30 & 31 & 32		*M*	3
30 & 31 & 33		*N*	3
30 & 32 & 33		*O*	3
31 & 32 & 33		*P*	3
29 & 30 & 31 & 32		*Q*	4
29 & 30 & 31 & 33		*R*	4
29 & 30 & 32 & 33		*S*	4
29 & 31 & 32 & 33		*T*	4
30 & 31 & 32 & 33		*U*	4
29 & 30 & 31 & 32 & 33		*V*	5

Table 10: Chip Attribute Values

Attribute	I_G	C_G
none	0	2
GC	1	1
PV	2	4
GC & PV	3	3

4. HARDWARE TROJAN EXAMPLES

A hardware trojan is assigned two vectors $\{I_T, C_T\}$ each consisting of eight elements. Each element represents the corresponding category value for the trojan. I_T identifies the attributes associated with the trojan while C_T indicates the severity of the trojan based on the attributes. These vectors provide a complete characterization of the trojan, and can be used to compare trojans. The vectors for two hardware trojan are shown in Table 11. Trojan A has identification I_T = {2 6 2 1 2 1 7 7} and severity C_T = {2 6 4 1 2 1 5 2}, while trojan B has identification I_T = {3 3 1 2 1 2 8 1} and severity C_T = {3 3 2 2 1 3 6 1}.

These trojans can be compared based on the severity for each category in C_T. C_R = 2 for trojan A and C_R = 3 for trojan B means that trojan A is inserted during the testing phase (attribute 4), and it may affect the assembly phase (attribute 5), whereas trojan B is inserted in the fabrication phase (attribute 3), and it may affect the testing and assembly phases (attributes 4 and 5). The insertion of trojan B may affect more phases than trojan A, so it has higher severity.

C_A = 6 for trojan A and C_A = 3 for trojan B means that trojan A is inserted at the system abstraction level (attribute 6), and the other levels RTL, development environment, logic, transistor, and physical (attributes 7, 8, 9, 10, and 11) may be affected, whereas trojan B is inserted at the logic abstraction level (attribute 9), so it may also affect the transistor and physical abstraction levels (attributes 10 and 11). The insertion of trojan A may affect more abstraction levels than trojan B, so it has higher severity.

C_E = 4 for trojan A and C_E = 2 for trojan B means that the effect of trojan A is to leak information (attribute 13) from a chip, whereas the effect of trojan B is to change the system functionality or cause denial of service (attributes 12 or 15). Although trojan B can change the chip functionality or prevent it from working as expected, trojan A is considered more serious.

Table 11: Hardware Trojan Examples

	Identification (I_T)								Severity (C_T)							
Trojan	I_R	I_A	I_E	I_L	I_F	I_C	I_P	I_O	C_R	C_A	C_E	C_L	C_F	C_C	C_P	C_O
Trojan A [21]	2	6	2	1	2	1	7	7	2	6	4	1	2	1	5	2
Trojan B [21]	3	3	1	2	1	2	8	1	3	3	2	2	1	3	6	1

C_L = 1 for trojan A and C_L = 2 for trojan B means that the logic type of trojan A is combinational (attribute 17), while that of trojan B is sequential (attribute 16). The severity of trojan B is higher than trojan A since a sequential trojan is harder to detect than a combinational one.

C_F = 2 for trojan A means that it is a parametric trojan (attribute 19), which means it could change the chip functionality, leak information, and/or reduced reliability (attributes 12, 13, and/or 14), C_F = 1 for trojan B indicates that it is a functional trojan (attribute 18), which typically changes the chip functionality (attribute 12). Trojan B has a lower severity because the effects of trojan A are hidden and the victim is less likely to be aware of its existence.

C_C = 1 for trojan A and C_C = 3 for trojan B means that trojan A is always on (attribute 20), while trojan B is always on and is internally triggered (attributes 20 and 21). The severity for trojan B is higher since it is internally triggered.

C_P = 5 for trojan A means that it has one of the following physical layout attributes: large, augmented, and distributed (attributes 23, 26, and 28), small, changed layout, and distributed (attributes 24, 25, and 28), or small, augmented, and clustered (attributes 24, 26, and 27). The trojan physical layout identifier I_P = 7 specifies that it is the last of these combinations. On the

other hand, $C_P = 6$ for trojan B indicates that the physical layout attributes are small, augmented, and distributed (attributes 24, 26, and 28). The severity of trojan B is higher than trojan A because it has a worse combination of attributes.

$C_O = 2$ means that trojan A can be inserted in two locations. The trojan location identifier $I_O = 7$ indicates that these locations are the processor or memory (attributes 29 or 30). On the other hand, $C_O = 1$ for trojan B means it can be inserted in only one location. From I_O, this location is the processor (attribute 29). Thus the severity of trojan A is higher since it can be inserted in more locations.

Typically, some vector elements will be higher for trojan A than trojan B, while others will be lower for trojan A than trojan B. Attackers and defenders must therefore decide which are the more important. This decision is based on the attacker capabilities and the system under consideration. For example, if the testing phase is done in-house and so is considered secure, trojan A cannot be inserted in the system.

However, if the attacker is part of the testing group, trojan A can be inserted. Further, if the fabrication stage is not secure, as required for trojan B, then any modification at this stage could affect the testing phase even if the attacker is not in the testing group. Therefore, the comparison should also be based on assumptions regarding the attacker and defender. For example, if the system deals with sensitive information, trojan A will be more dangerous than trojan B because it was designed to leak information. However, if the system is located in an aircraft, trojan B will be more dangerous as a denial of service attack or change in functionality could cause a crash.

5. TROJAN DETECTION TECHNIQUE EXAMPLES

A trojan detection technique is assigned two vectors $\{I_D, C_D\}$, each consisting of nine elements. The first identifies the trojan attributes that the technique is effective against. The second specifies the effectiveness against trojans which have these attributes. These vectors provide a complete description of the trojan detection technique. Table 12 presents the identification and effectiveness vectors for a number of hardware trojan detection techniques. The effectiveness of the detection techniques can easily be compared using this information to determine which is best for a given system. We consider a comparison of the two techniques given in Table 13.

The first technique is from [8] and has identification $I_D = \{3\ 3\ B\ 1\ 2\ 4\ 7\ V\ 1\}$ and effectiveness $C_D = \{3\ 3\ 7\ 1\ 2\ 3\ 5\ 5\ 2\}$, while the second technique is from [16] and has identification $I_D = \{3\ 3\ 1\ 3\ 1\ 4\ 7\ V\ 4\}$ and effectiveness $C_D = \{3\ 3\ 2\ 3\ 1\ 3\ 5\ 5\ 3\}$.

From Table 13, both techniques have the same values of C_R, C_A, C_C, C_P and C_O, but differ in in the other four effectiveness parameters. $C_E = 7$ for the first technique means that it can detect trojans that affect three attributes, and $I_E = B$ indicates that these attributes are change in functionality, information leakage, and reduce reliability (attributes 12, 13, and 14).

On the other hand, $C_E = 2$ for the second technique means that it is designed to detect a trojan inserted to change functionality or create a denial of service (attributes 12 or 15). $I_E = 1$ indicates that it is change in functionality (attribute 12). A defender should choose the first technique if the target system deals with sensitive information so that a trojan designed to leak information is the main threat.

Table 12: Hardware Trojan Detection Techniques

	Identification									Effectiveness								
Technique	I_R	I_A	I_E	I_L	I_F	I_C	I_P	I_O	I_G	C_R	C_A	C_E	C_L	C_F	C_C	C_P	C_O	C_G
[4]	4	1	C	3	3	4	7	V	3	4	1	8	3	3	3	5	5	4
[5]	2	6	2	1	2	1	7	V	3	2	6	4	1	2	1	5	5	4
[6]	5	6	2	1	2	1	7	V	4	5	6	4	1	2	1	5	5	3
[7]	3	3	1	1	1	4	7	V	2	3	3	2	1	1	3	5	5	1
[8]	3	3	B	1	2	4	7	V	1	3	3	7	1	2	3	5	5	2
[9]	4	3	1	2	1	2	8	V	4	4	3	2	2	1	2	6	5	3
[10]	4	5	1	3	3	4	8	V	4	4	5	2	3	3	3	6	5	3
[11]	3	4	5	3	3	4	5	V	4	3	4	6	3	3	3	4	5	3
[12]	4	4	A	1	2	1	8	V	3	4	4	3	1	2	1	6	5	4
[13]	4	4	5	1	3	1	8	V	2	4	4	6	1	3	1	6	5	1
[14]	4	4	4	2	1	3	8	V	2	4	4	2	2	1	4	6	5	1
[15]	3	5	5	3	3	2	8	V	4	3	5	6	3	3	2	6	5	3
[16]	3	3	1	3	1	4	7	V	4	3	3	2	3	1	3	5	5	3
[19]	3	4	6	3	1	7	8	G	4	3	4	3	3	1	7	6	3	3
[22]	4	5	5	2	1	2	8	V	3	4	5	6	2	1	2	6	5	4
[23]	4	3	5	3	3	4	8	V	4	4	3	6	3	3	3	6	5	3
[25]	5	5	4	3	1	2	3	V	4	5	5	2	3	1	2	4	5	3
[26]	4	5	1	1	1	1	8	V	4	4	5	2	1	1	1	6	5	3
[27]	4	3	1	1	1	4	8	V	4	4	3	2	1	1	3	6	5	3
[28]	4	5	9	3	1	7	2	V	2	4	5	6	3	1	7	4	5	1
[29]	4	4	1	3	1	2	7	V	1	4	4	2	3	1	2	5	5	2

C_L = 1 for the first technique means that it is designed to detect only combinational trojans (attribute 17), while C_L = 3 for the second technique means that it can detect both combinational and sequential trojans (attributes 16 and 17). A defender should choose the latter technique as it can protect against both types of trojans.

C_F = 2 for the first technique means that it can only detect parametric trojans (attribute 19), while C_F = 1 for the second technique indicates that it can only detect functional trojans (attribute 18). If information leakage is the main concern, a defender should choose the latter technique.

C_G = 2 for the first technique means that it does not consider process variations and does not need reference measurements for detection. On the other hand, C_G = 3 for the second technique means that it considers process variations but also requires reference measurements for detection. A defender should choose the latter technique if false alarms due to process variations are important to avoid.

Table 13: Hardware Trojan Detection Technique Examples

	Identification (I_D)									Effectiveness (C_D)								
Technique	I_R	I_A	I_E	I_L	I_F	I_C	I_P	I_O	I_G	C_R	C_A	C_E	C_L	C_F	C_C	C_P	C_O	C_G
[8]	3	3	B	1	2	4	7	V	1	3	3	7	1	2	3	5	5	2
[16]	3	3	1	3	1	4	7	V	4	3	3	2	3	1	3	5	5	3

Trojan detection techniques should be compared based on the effectiveness values. Typically, some values are higher for one technique while others are higher for another technique. A defender must decide which attribute categories are more important based on the target system and which attributes are secure for this system.

6. CONCLUSION

The sophistication of hardware trojan detection techniques has been increasing in an attempt to improve detection rates. However, this makes it difficult to compare different methods and their effectiveness. A comprehensive evaluation of the attributes of hardware trojans is used here to classify detection techniques. This evaluation is also used to measure and compare the severity of these trojans. The attributes are ranked and weighted according to their importance in the detection process. The proposed approach provides a means of evaluating existing as well as new trojan detection techniques. In addition, it can be used to compare detection techniques and determine their effectiveness.

REFERENCES

[1] M. Banga and M. Hsiao, "A region based approach for the identification of hardware trojans," in Proc.IEEE Int. Workshop on Hardware-Oriented Security and Trust, pp. 40–47, 2008.

[2] S. Narasimhan and S. Bhunia, "Hardware trojan detection," in Introduction to Hardware Security and Trust, M. Tehranipoor and C. Wang (Eds.), Springer, New York, NY, pp. 339–364, 2012.

[3] C. Bao, D. Forte, and A. Srivastava, "On reverse engineering-based hardware trojan detection," IEEE Trans. Comput.-Aided Design Integr. Circuits and Sys., vol. 35, no. 1, pp. 49–57, 2015.

[4] C. Marchand and J. Francq, "Low-level implementation and side-channel detection of stealthy hardware trojans on field programmable gate arrays," IET Computers Digital Tech., vol. 8, no. 6, pp. 246–255, 2014.

[5] Y. Liu, K. Huang, and Y. Makris, "Hardware trojan detection through golden chip-free statistical side-channel fingerprinting," in Proc. ACM/EDAC/IEEE Design Automation Conf., pp. 1–6, 2014.

[6] Y. Liu, Y. Jin, and Y. Makris, "Hardware trojans in wireless cryptographic ICs: Silicon demonstration & detection method evaluation," in Proc. IEEE/ACM Int. Conf. on Computer-Aided Design, pp. 399–404, 2013.

[7] D. Karunakaran and N. Mohankumar, "Malicious combinational hardware trojan detection by gate level characterization in 90nm technology," in Proc. Int. Conf. on Computing, Commun. and Networking Tech., pp. 1–7, 2014.

[8] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey, "Hardware trojan horse detection using gate-level characterization," in Proc. IEEE/ACM Design Automation Conf., pp. 688–693, 2009.

[9] Y. Cao, C. Chang, and S. Chen, "A cluster-based distributed active current sensing circuit for hardware trojan detection," IEEE Trans. Inform. Forensics Security, vol. 9, no. 12, pp. 2220–2231, 2014.

[10] X. Mingfu, H. Aiqun, and L. Guyue, "Detecting hardware trojan through heuristic partition and activity driven test pattern generation," in Proc. Commun. Security Conf., pp. 1–6, 2014.

[11] Y. Jin and Y. Makris, "Hardware trojan detection using path delay fingerprint," in Proc. IEEE Int. Conf. on Hardware-Oriented Security and Trust, pp. 51–57, 2008.

[12] M. Li, A. Davoodi, and M. Tehranipoor, "A sensor-assisted self-authentication framework for hardware trojan detection," in Proc. Design, Automation and Test in Europe Conf. and Exhibition, pp. 1331–1336, 2012.

[13] P. Kumar and R. Srinivasan, "Detection of hardware trojan in SEA using path delay," in Proc. IEEE Students' Conf. on Elect., Electronics and Computer Sci., pp. 1–6, 2014.

[14] O. Soll, T. Korak, M. Muehlberghuber, and M. Hutter, "EM-based detection of hardware trojans on FPGAs," in Proc. IEEE Int. Symp. on Hardware-Oriented Security and Trust, pp. 84–87, 2014.

[15] A. Nowroz, K. Hu, F. Koushanfar, and S. Reda, "Novel techniques for high-sensitivity hardware trojan detection using thermal and power maps," IEEE Trans. Comput.-Aided Design Integr. Circuits and Sys., vol. 33, no. 12, pp. 1792–1805, 2014.

[16] S. Narasimhan et al., "Hardware trojan detection by multiple-parameter side-channel analysis," IEEE Trans. Comput., vol. 62, no. 11, pp. 2183–2195, 2013.

[17] M. Abramovici and P. Bradley, "Integrated circuit security: New threats and solutions," in Proc. Workshop on Cyber Security and Inform. Intelligence Res., article no. 55, 2009.

[18] D. McIntyre, F. Wolff, C. Papachristou, and S. Bhunia, "Dynamic evaluation of hardware trust," in Proc. IEEE Int. Workshop on Hardware-Oriented Security and Trust, pp. 108–111, 2009.

[19] T. F. Wu et al., "TPAD: Hardware trojan prevention and detection for trusted integrated circuits," IEEE Trans. Comput.-Aided Design Integr. Circuits and Sys., vol. 35, no. 4, pp. 521–534, 2016.

[20] S. Moein, S. Khan, T. A. Gulliver, and F. Gebali, and M. W. El-Kharashi, "An attribute based classification of hardware trojans," in Proc. Int. Conf. on Computer Engineering and Sys., pp. 351–356, 2015.

[21] S. Moein, T. A. Gulliver, F. Gebali, and A. Alkandari, "A new characterization of hardware trojans," IEEE Access, vol. 4, pp. 2721-2731, 2016.

[22] S. Narasimhan et al., "TeSR: A robust temporal self-referencing approach for hardware trojan detection," in Proc. IEEE Int. Symp. on Hardware-Oriented Security and Trust, pp. 71–74, 2011.

[23] X. Wang, H. Salmani, M. Tehranipoor, and J. Plusquellic, "Hardware trojan detection and isolation using current integration and localized current analysis," in Proc. IEEE Int. Symp. on Defect and Fault Tolerance of VLSI Systems, pp. 87–95, 2008.

[24] C. Lavin et al., "RapidSmith: Do-it-yourself CAD tools for Xilinx FPGAs," in Proc. Int. Conf. on Field Programmable Logic and Applic., pp. 349–355, 2011.

[25] P. Kitsos and A. G. Voyiatzis, "FPGA trojan detection using length-optimized ring oscillators," in Proc. Euromicro Conf. on Digital System Design, pp. 675–678, 2014.

[26] X. Zhang and M. Tehranipoor, "RON: An on-chip ring oscillator network for hardware trojan detection," in Proc. Design, Automation and Test in Europe Conf. and Exhibition, pp. 1–6, 2011.

[27] A. Ferraiuolo, X. Zhang, and M. Tehranipoor, "Experimental analysis of a ring oscillator network for hardware trojan detection in a 90nm ASIC," in Proc. IEEE/ACM Int. Conf. on Computer-Aided Design, pp. 37–42, 2012.

[28] D. Forte, C. Bao, and A. Srivastava, "Temperature tracking: An innovative run-time approach for hardware trojan detection," in Proc. IEEE/ACM Int. Conf. on Computer-Aided Design, pp. 532–539, 2013.

[29] X. Cui, K. Ma, L. Shi, and K. Wu, "High-level synthesis for run-time hardware trojan detection and recovery," in Proc. ACM/EDAC/IEEE Design Automation Conf., pp. 1–6, 2014.

[30] S. Moein, J. Subramnian, T. A. Gulliver, and F. Gebali, and M. W. El-Kharashi, "Classification of hardware trojan detection techniques," in Proc. Int. Conf. on Computer Engineering and Sys., pp. 357–362, 2015.

Authors

Samer Moein received the B.Sc. and M.Sc. degrees from Kuwait University, Kuwait, in 2004 and 2011, respectively, and the Ph.D. degree from the University of Victoria, Victoria, BC , Canada, in 2015, all in computer engineering. He is currently a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Victoria. His research interests include computer security, cryptography, and cryptoprocessors.

Fayez Gebali received the B.Sc. (Hons.) degree in electrical engineering from Cairo University, the B.Sc. (Hons.) degree in mathematics from Ain Shams University, and the Ph.D. degree in electrical engineering from the University of British Columbia. He is a Professor with the Department of Electrical and Computer Engineering, University of Victoria, where he is currently Department Chair. His research interests include parallel algorithms, networks-on-chip, 3-D integrated circuits, digital communications, and computer arithmetic. He held an NSERC Postgraduate Scholarship at the University of British Columbia.

T. Aaron Gulliver received the Ph.D. degree in electrical engineering from the University of Victoria, Victoria, BC, Canada, in 1989. From 1989 to 1991, he was a Defence Scientist with the Defence Research Establishment Ottawa, Ottawa, ON, Canada. He has held academic appointments with Carleton University, Ottawa, and the University of Canterbury, Christchurch, New Zealand. He joined the University of Victoria in 1999, where he is a Professor with the Department of Electrical and Computer Engineering. In 2002, he became a fellow of the Engineering Institute of Canada. In 2012, he was elected as a fellow of the Canadian Academy of Engineering. From 2000 to 2003, he was Secretary and a member of the Board of Governors of the IEEE Information Theory Society. He is currently an Area Editor of the IEEE Transactions on Wireless Communications. His research interests include information theory and communication theory, algebraic coding theory, multicarrier systems, smart grid, and security.

Abdulrahman Alkandari received the B.Sc. and M.Sc. degrees in computer engineering from Kuwait University, Kuwait, in 2004 and 2011, and the Ph.D. degree in computer science from the International Islamic University Malaysia in 2014. He is an Assistant Professor with the Department of Computer Science, Public Authority for Applied Education and Training (Basic Education College). His research interests include intelligent systems, traffic engineering, algorithms, smart phone applications, IoT, smart cities, and wireless sensor.