Classification of Hardware Trojan Detection Techniques

Samer Moein, Jayaram Subramnian, T. Aaron Gulliver, Fayez Gebali
Department of Electrical and Computer Engineering
University of Victoria, Victoria V8W 2Y2, Canada
Email: {samerm, jayaram, agullive, fayez}@uvic.ca

M. Watheq El-Kharashi
Computer and Systems Engineering Department
Ain Shams University, Cairo 11517, Egypt
Email: watheq.elkharashi@eng.asu.edu.eg

***Abstract*—The majority of techniques that have been developed to detect hardware trojans are based on only specific attributes. Further, the ad hoc approaches that have been employed to design methods for trojan detection are largely ineffective. These trojans have a number of attributes which can be used in a systematic way to 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.**

I. 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 critical that effective hardware trojan detection techniques be developed.

A *hardware trojan* is defined as a malicious component embedded in an IC which causes abnormal behaviour [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. Fig. 1 shows 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)), as they are extremely expensive and time consuming. Therefore it is often not practical to test chips using destructive techniques. Further, Process Variations (PVs) can have a significant influence on distinguishing false positives and trojan free chips based on a GC. Note that 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.

Non-destructive techniques can be classified into testing and run-time monitoring methods. Testing can be supported by design for security circuits (i.e. scan chains and self-test circuitry). These circuits improve trojan detection effectiveness and coverage, but require that the test circuitry not be compromised. A trojan may not be triggered, or it may be designed to avoid activation during testing to prevent signal emissions until testing is completed. Therefore, a trojan that does not change the chip layout or design can be difficult to detect using testing.

Testing approaches can be classified into logic-testing or side-channel analysis. Logic-testing approaches generate random test vectors and employ applications in an attempt to activate trojan circuits and observe their effects at the outputs. The difficulty with this approach is the complexity of testing all internal nodes and possible logic values as chips now 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 [3]–[7], leakage current [8], [9], path-delay characteristics [10]–[12], Electromagnetic (EM) radiation [13], or a combination of these parameters [14], [15]. 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 the 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 the 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 the circuit.

Run-time monitoring is used to continuously monitor chip operation to detect the effects of malicious circuitry and initiate techniques for mitigation. This can be achieved by exploiting pre-existing redundancy in the circuit such as a reconfigurable core [16] in a multicore system [17] to avoid an infected part 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. In addition, chips can be equipped with self destructive packaging to disable them or discard the output when a trojan is detected.

The high complexity of chips and PVs makes many detection techniques inadequate. Therefore, new techniques must be developed or approaches combined to improve performance.

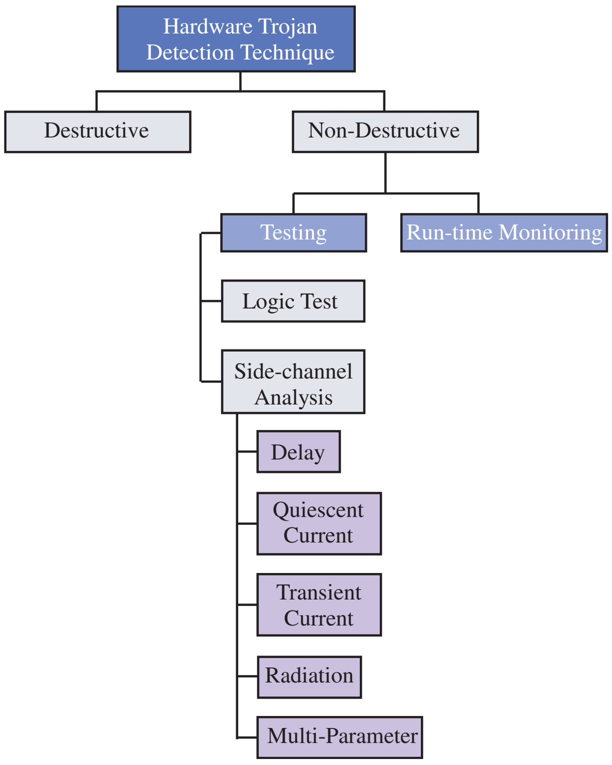

Fig. 1. Existing Hardware Trojan Detection Classification

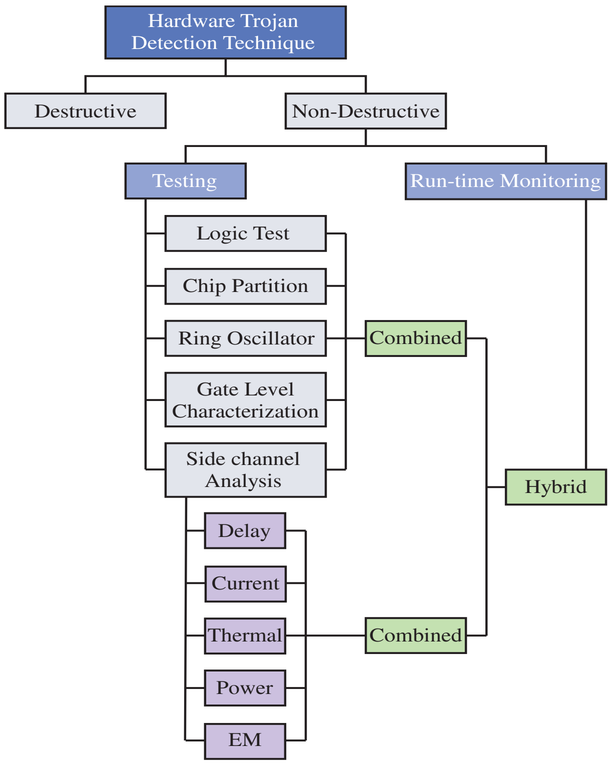

Fig. 3. Proposed Hardware Trojan Detection Classification

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 render the detection method ineffective. 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 trojan properties is a challenging task.

II. Hardware Trojans

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 [18] which is based on eight key categories: insertion, abstraction, effect, logic type, functionality, activation, physical layout, and location. A classification of attributes based on this model is illustrated in Fig. 2. The relationships between these attributes presented in [19] identifies the attributes that can be detected using a given technique, i.e. 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. For example, a trojan inserted in the insertion phase may affect attributes in other categories, i.e. 99.99% of all trojans are inserted in the specification phase, so to be effective a technique should detect if a specification attribute has been compromised. In this paper, a new classification of hardware trojan detection techniques is presented which is based on trojan attributes and their relationships.

III. Hardware Trojan Detection Techniques

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 [14]. For example, in [7] 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 Fig. 3. This is a new approach to trojan detection. Compared to Fig. 1, the proposed approach combines multiple side channel analysis techniques and/or other techniques to improve the effectiveness of trojan detection. Because of the possibility of a trojan evading detection during testing, run-time monitoring is very important. For example, non-destructive technique can be combined with destructive packaging or discarding the output when a trojan is detected.

A. *Side-channel Techniques*

1) Delay:

a) Path Delay Sensors: A sensor to detect a hardware trojan in an IC was developed in [11]. This sensor predicts the delay characteristics of specific sequences of operations in the circuit and compares these with reference values. A significant difference indicates that a trojan may be present. Numerous experiments were conducted with different trojan

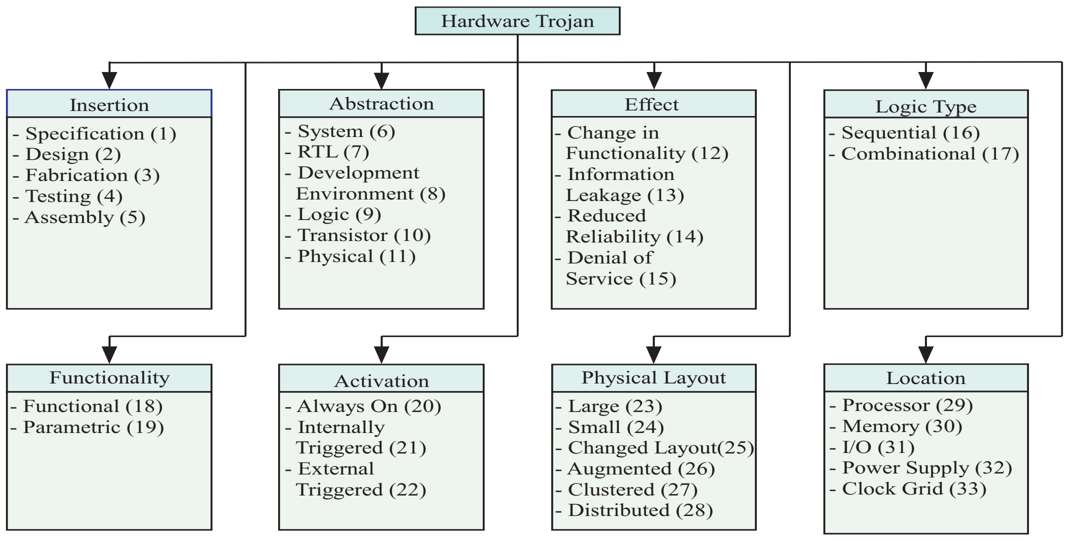

Fig. 2. Classification of Hardware Trojan Attributes

delays and circuit paths. This approach was able to identify trojans with reasonable accuracy compared to non-sensor based techniques. However, the sensors increase the chip area unless they are inserted in empty spaces. This also requires that delay measurements be made within the chip which may be difficult to obtain.

The trojan presented in [11] is inserted during the design stage in the logic abstraction level and causes changes in the circuit parameters to reduce reliability. It is composed of only gates and is always on.

b) Path Delay Fingerprint: A trojan detection technique based on the path delay introduced by trojans was developed in [10], [12]. This method differentiates between the delay for a GC and the IC being tested. The effects of PVs are considered to limit the false positive rate.

The trojan presented in [10] causes changes in circuit function or information leakage and is inserted during the fabrication stage in the logic abstraction level. This trojan can be implemented using combinational or sequential logic and is internally triggered by a specific sequence or a counter. The trojan in [12] also causes changes in circuit function or information leakage and is inserted during the manufacturing or design stages in the layout or logic abstraction levels. It is implemented using combinational logic and so is always on.

2) Current:

a) Self-referencing: A self-referencing approach was employed in [20] to detect hardware trojans. A test vector is used to create a leakage current signature for a GC. The leakage current for an IC under test is compared with the GC signature to determine if a trojan may exist. Different size trojans were inserted in the IC to determine the detection sensitivity.

The trojan presented in [20] causes changes in the circuit function or leaks information, and is inserted during the design stage in the logic abstraction level. This trojan is implemented using sequential logic and is internally triggered by a counter.

b) Current Integration: In [21], current integration method was used to detect hardware trojans. Typically, the leakage current and current consumed by a chip is measured and compared with that of a GC. Instead of using the current directly, it is integrated to convert current into charge which magnifies any differences in the measured current. Therefore, this approach can be useful in identifying small trojans where the effect on the current is masked by PVs and noise.

The trojan presented in [21] causes changes in circuit function and is inserted during the design phase in the layout abstraction level. This trojan can be implemented using combinational or sequential logic and can be internally triggered by a counter or a specific sequence.

3) Power Consumption: Power measurements for random test patterns can be used to generate a power signature using a GC. This signature can be based on the power consumption as well as the noise generated from surrounded circuits and PV effects. By applying the same random test patterns to all chips, they can be divided into groups according to their power signatures. Reverse engineering can then be applied to a chip from each set to determine which groups are trojan free. The results obtained can be used to choose chips to be used to generate the GC signature, as well as determine suspicious ships which may contain a trojan.

a) Power Consumption Trace: In [3], trojans implemented at the layout level in FPGAs were considered with standard evaluation boards. A side-channel technique using power consumption was developed for trojan detection.

The trojan presented in [3] can cause Denial of Service (DoS), changes in circuit functions and information leakage. It is inserted during the design stage in the layout abstraction level. This trojan can be implemented using either sequential or combinational logic and is internally triggered by a counter or a specific condition.

b) Process Control Monitoring: The detection of hardware trojans which leak information was considered in [4] using a statistical analysis of the output power consumption to classify ICs as infected or trojan free. The advantage of this approach is that a golden chip is not required. Instead, Monte-Carlo simulation is used to determine the classification threshold. A hardware trojan was designed within the PVs of a chip to determine the effectiveness of this approach. It was shown that good trojan detection accuracy can be achieved, but a detailed statistical model is required to achieve suitable accuracy, so this approach can require significant time and resources.

The trojan presented in [4] leaks information and is inserted during the fabrication stage in the system abstraction level, so it is parametric and the trojan is always on.

c) Power Consumption: In [5] a side-channel technique based on power consumption was considered to detect hardware trojans that create information leakage. A hardware trojan was designed within PV limits, and the difference in power consumption between an infected and a trojan free IC was evaluated. It was found that comparing the power consumption does not aid in trojan detection, but a statistical analysis provided better performance. This shows the usefulness of statistical analysis in hardware trojan detection.

The trojan presented in [5] leaks information and was inserted during the fabrication stage in the system abstraction level. Since it leaks information, this trojan is parametric and is always on.

4) Electromagnetic (EM) Radiation: In [13], hardware trojan detection was considered based on Electromagnetic (EM) radiation. Trojans were inserted next to I/O pins an their effect on the EM emissions was measured [22]. It was shown that significant differences can exist in the emissions at different chip locations due to the presence of hardware trojans. In particular, it is easier to detect a trojan at the corners of chips because of the proximity to the power lines.

The trojan presented in [13] causes DoS and is inserted during the design stage in the layout or by development environment abstraction level. The trojan is sequential and is externally triggered.

B. Multiple Parameters

Most side-channel techniques monitor just the leakage current to detect hardware trojans. This requires a GC to determine current anomalies which indicate that a hardware trojan may be present. However, this method has several disadvantages such as PV effects and noisy measurements. Therefore, techniques which consider multiple parameters have been developed based on different chip emissions to increase the detection rate and decrease the number of false positives.

1) Thermal and Power: The approach in [14] uses thermal and power maps to detect hardware trojans. To improve detection accuracy, statistical analysis as well as thresholding and clustering are used. Further, the effects of both PVs and noise are included in the model.

The trojan presented in [14] causes changes in circuit function and is inserted during the manufacturing stage in the RTL abstraction level. This trojan can be implemented using either sequential or combinational logic and is internally triggered by a counter or a specific condition.

C. Hybrid Techniques

1) Logic Testing:

a) Current and Operating Frequency: In [15], hardware trojan detection was considered using random test vectors to generate different circuit emissions. The leakage current, transient supply current and maximum operating frequency were used in the trojan detection process. Techniques to improve the detection sensitivity were employed, and power gating was used to prevent undesirable circuit switching. A disadvantage of this approach is that it is not as effective as logic testing for small trojans.

The trojan presented in [15] causes changes in circuit function and is inserted during the fabrication stage in the logic abstraction level. This trojan can be implemented using either combinational or sequential logic and is internally triggered by a specific sequence or a counter.

2) Gate Level Characterization (GLC):

a) Power Consumption: The use of gate level characterization (GLC) in the detection of hardware trojans was examined in [6]. GLC considers the timing delay, switching power and leakage power of each gate in the circuit. Equations are developed for each gate and a trojan variable is used to determine the presence of a trojan. Experiments were performed using several benchmarks to obtain the GLC of simple gates such as AND, OR, NOT, NOR and NAND. The disadvantage of GLC is that the characterizing all gates in a circuit is a very time consuming process and the complexity can make it intractable to include all gates in the circuit.

The trojan presented in [6] causes changes in the function of the circuit and is inserted during the fabrication stage in the logic abstraction level. This trojan can be implemented using a small number of gates and is internally triggered.

b) Power Consumption and Delay: In [7], hardware trojans were also detected using GLC. The delay and switching power characteristics of each gate was determined using a number of measurements.

The trojan presented in [7] causes changes in the function of the circuit or parametric changes to reduce reliability, or leaks information. It was inserted during the fabrication stage in the logic/layout abstraction levels. This trojan can be implemented using combinational or sequential and is internally triggered by a specific sequence or a counter.

D. Ring Oscillator (RO)

1) Length Optimized Ring Oscillators: In [23], a Ring Oscillator (RO) was employed to detect hardware trojans. This is based on differences in the RO frequency due to the presence of a trojan. Both sequential and combinational logic trojans were considered and different RO lengths examined to determine the effect of this length on trojan detection. Trojans were introduced without changing the layout of the circuit. It was shown that a RO of length 7 provides adequate detection performance. The disadvantage of this scheme is that it is difficult to extend the results to other conditions and circuit configurations.

The trojan presented in [23] causes DoS and is inserted during the specification or design stages in the RTL abstraction level. This trojan can be implemented using either sequential or combinational logic and is internally triggered by a counter or a specific condition.

2) Ring Oscillator Network: In [24], the detection of hardware trojans was examined using a RO network. The oscillator frequency can change if a hardware trojan is present in the circuit, and ROs are distributed throughout the circuit can thus be used for monitoring purposes. To limit the effects of PVs and noise, statistical and outlier analysis are used.

The trojan presented in [24] causes changes in the function of the circuit and is inserted during the fabrication stage of the logic abstraction level. This trojan can be implemented using gates or combinational trojans and is either always on or internally triggered by a specific condition.

In [25], another RO technique was introduced to detect hardware trojans. A number of trojans and different RO locations were considered to determine the best location for the ROs. Rather than considering the oscillator frequency, principal component analysis was used to extract relevant information from the measurements. The average of a number of measurements was used to reduce the effects on noise.

The trojan presented in [25] causes changes in the function of the circuit and was inserted during the logic abstraction level. This trojan can be implemented using combinational logic and is internally triggered by a given sequence or specific condition.

E. Chip Partition Technique (CPT)

1) Current: In [8], time and power signatures were used to compare a Chip Under Test (CUT) with a GC. Current sensors are used to obtain a signature. This sensor consists of a current mirror, a current comparator and a scan register. The circuit is partitioned into regions and each region is tested separately using a corresponding sensor. If the current signature differs significantly from that of the GC, a trojan may be present.

The trojan presented in [8] causes changes in the function of the circuit and is inserted during the design stage in the logic/layout abstraction levels. This trojan can be implemented using sequential logic and is internally triggered based on a counter.

2) Power: In [9], hardware trojans were detected by partitioning the chip into regions and using the power ports in each region to detect abnormal activity. Test patterns were used in each region to examine the behavior and the effects of noise were reduced by appropriate placement of the ports.

The trojan presented in [9] causes changes in the function of the circuit and was inserted during the manufacturing/design stage in the developmental abstraction level. This trojan can be implemented using either combinational or sequential logic and is internally triggered by a specific condition or a counter.

F. Run-Time Monitoring

1) Temperature Tracking: In [26], hardware trojans were detected during run-time using thermal maps. A framework for temperature based detection was proposed which consists of design, test and run-time detection. A statistical characterization of the ICs is determined using thermal sensors placed in the design. During testing, the noise levels are determined, and these results are used at run-time along with thermal measurements to detect trojans.

The trojan presented in [26] causes changes in the function of the circuit or DoS, and is inserted during the design stage in the logic abstraction level. This trojan can be implemented using either combinational or sequential logic and is internally or externally triggered.

2) Redundancy: In [27], design rules were developed to aid in trojan detection during run-time and provide fast recovery from attacks by deactivating the trojan. Eight design rules were proposed for hardware trojan detection based on the test results for IP codes from several vendors. These rules were optimized based on constraints such as latency, area, number of operations, and cost. This approach was used mainly to determine the trustworthiness of IP cores from third party vendors.

The trojan presented in [27] causes changes in the function of the circuit and was inserted during the design stage in the development abstraction level. This trojan can be implemented using combinational or sequential logic and is internally triggered by a specific sequence or a counter.

G. Summary

Table I provides a summary of the attributes for the detection techniques considered in this section. 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 cell means that the technique cannot protect against the attribute. In addition, a technique may require results from a golden chip to compare with measurements from a CUT. This is indicated by an R in the GC column. Moreover, if a technique considers PVs, this is indicated by a C in the PV column. Each technique can detect a trojan with a specific set of attributes. This is an ad hoc approach to detection as minor changes to the corresponding trojan may render the technique ineffective. Further, it is difficult to compare different detection techniques using the results in the literature. Table I provides a means of identifying and classifying trojans to aid in developing an effective detection technique.

IV. Conclusion

The complexity of hardware trojan detection techniques has been increasing in an attempt to improve the detection rate. However, this makes it difficult to compare different approaches and their effectiveness. A comprehensive evaluation

TABLE I
CLASSIFICATION OF HARDWARE TROJAN DETECTION TECHNIQUES

	Attributes																																	Chip Attribute	
	Insertion					Abstraction						Effect				Type		Functionality		Activation			Physical Layout						Location						
Technique	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	PV
[3]		C									C	C	C		C	C	C	C	C	C	C		C	C		C	C		C	C	C	M	M		C
[4]				C		C							C				C		C	C				C		C	C		C	M	C	M	M		C
[5]	M		C			M					C		C				C		C	C				C		C	C		C	C	C	M	M	R	C
[6]			C						C			C					C	C		C	C		C	C		C	C		C	M	M	M	M	R	
[7]			C						C	C	C	C	C	C			C		C	C	C		C	C		C	C		C	M	M	M	M		
[23]	C	C					C								C	C	C	C			C		C		C	C	C		C	M	M	M	M	R	C
[24]		C	C				C					C					C	C		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	C	C	C	C	M	M	M	M	R	C
[13]		C						C	C						C	C		C				C	C	C	C	C	C	C	M	M	C	M	M	R	
[8]		C							C			C				C		C			C		C	C	C	C	C	C	C	M	C	M	M	R	C
[9]		C	C				C	C	C			C				C	C	C	C	C	C		C	C		C	C	C	C	M	M	M	M	R	C
[14]			C				C		C			C				C	C	C			C		C	C	C	C	C	C	C	M	M	M	M	R	C
[20]		C	C				C	C	C			C	C			C		C			C		C	C	C	C	C	C	C	M	M	M	M		C
[21]		C							C		C	M	M			C	C	C	M	C	C		C	C		C	C	C	C	M	M	M	M	R	C
[15]			C						C			C				C	C	C		C	C		C	C		C	C		C	M	M	M	M	R	C
[10]			C					C				C	C			C	C	C	C	C	C		C	C	C		C		C	M	C	M	M	R	C
[11]		C	C					C	C					C	C		C		C	C			C	C	C	C	C	M	C	M	M	M	M		C
[12]		C	C					C	C		C	C	C				C	C	C	C			C	C	C	C	C	C	C	M	M	M	M	R	
[26]		C	C				C	C	C				C		C	C	C	C		C	C	C	C		C		M	M	C	M	C	M	M	R	
[27]		C	C					C	C			C				C	C	C			C		C	C		C	C		C	M	M	M	M		

of the attributes of hardware trojans is used here to classify detection techniques. The attributes are assigned weights according to their importance in the detection process. The proposed approach provides a means of evaluating existing and new trojan detection techniques.

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. 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

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

[17] 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.

[18] 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.*, 2015.

[19] S. Moein, T. A. Gulliver, and F. Gebali, "A new characterization of hardware trojan attributes," *IEEE Trans. Inform. Forensics Security*, submitted.

[20] S. Narasimhan, X. Wang, D. Du, R. S. Chakraborty, and S. Bhunia, "TeSR: A robust temporal self-referencing approach for hardware trojan detection," in *IEEE Int. Symp. on Hardware-Oriented Security and Trust*, pp. 71–74, 2011.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.