An Automated Tool for Evaluating Hardware Trojans and Detection Methods

Nicholas Houghton, Samer Moein, Fayez Gebali, and T. Aaron Gulliver
Department of Electrical and Computer Engineering
University of Victoria
P.O. Box 1700 STN CSC
Victoria, B.C. V8W 2Y2
Email: {nhoughto, samerm, fayez, agullive}@uvic.ca

Abstract—The growing concern for hardware security has spawned numerous trojan detection methods. Due to the complexity of integrated circuits (ICs), detection methods developed thus far have only been successful for specific trojans. This makes it difficult to compare hardware trojans and the performance of detection methods. In this paper, a systematic and automated approach to analyzing the characteristics of trojans and detection methods is presented. Universal adoption of the techniques in this system will aid in collaboration and standardization in the field. It will also provide a centralized database of existing hardware trojans and detection methods. A discussion of the automated tool design is given including a case study to demonstrate its usefulness.

I. INTRODUCTION

The field of hardware security is relatively new and as a consequence lacks accepted techniques for analysis and comparison. The variety of Integrated Circuit (IC) design and construction techniques has resulted in a diversity of structures, behavior and locations of hardware trojans. As a consequence, the majority of detection methods have been designed for specific trojans. As an example, side-channel methods such as power [1]–[5], current [6], [7], delay [8]–[10], and Electromagnetic (EM) radiation [11], or a combination of these techniques [12], all rely on certain trojan attributes to achieve success. Each of these methods can fail with only a minor change to the trojan. In [13], the nonuniform nature of currently available methods was discussed, as well as the lack of a means of analyzing and comparing them.

Hardware trojans are comprised of attributes which can be used to characterize them, such as the entry point in the circuit, effect, and location. A taxonomy for hardware trojans which incorporates the IC life-cycle and characteristics was presented in [14]. This taxonomy can be employed to detect hardware trojans as well as analyze and compare detection methods. With an effective means of characterizing trojans, designers and attackers can select the best approach to detection given the available resources. However, analysis can be difficult and prone to error. Thus, a tool called the Hardware Trojan System (HTS) has been developed to automate this process. It allows users to quickly and easily select trojan characteristics based on their observations. These characteristics are input to the system and attribute tables are used for analysis. The HTS automates the necessary computations, provides centralized documentation, and compiles a database of trojans and detection methods. This database can be used by developers and attackers to search existing trojans and detection methods for design purposes.

The contributions of this paper are as follows.

1) A new technique is proposed for evaluating trojans and detection methods.
2) A web-based tool which automates the evaluation and comparison processes is presented.
3) A database to store the analysis results is developed.

The remainder of this paper is organized as follows. Section II discusses the new comprehensive taxonomy proposed in [14]. The attributes are studied and trojan risk (severity) and detection effectiveness (coverage) values are assigned. Sections III and IV provide a description of the on-line tool while Section V provides a case study to demonstrate its use. Finally, some concluding remarks are given in Section VI along with suggestions for future work.

II. HARDWARE TROJAN TAXONOMY

Several hardware trojan taxonomies have been proposed in the literature [15]–[18]. In [15], trojans were organized based solely on their activation mechanisms. A scheme based on the location, activation and action of a trojan was presented in [16], [17]. However, these approaches do not consider the manufacturing process. A taxonomy was proposed in [18] which employs five categories: insertion, abstraction, activation, effect and location. While this is more extensive than previous methods, it fails to account for the physical characteristics of the trojan. The taxonomy proposed in [14] is the most comprehensive as it considers all known attributes a trojan may possess and also their relationships. This makes it possible to not only analyze the merits of a detection method compared to others but also allows for the investigation of the coverage of a method in detecting trojans. This taxonomy is comprised of thirty-three attributes organized into eight categories as shown in Fig. 1. These categories can be arranged into four levels as shown in Fig. 2.

1) The insertion (chip life-cycle) level comprises the attributes pertaining to the IC production stages.
2) The abstraction level corresponds to where in the IC abstraction the trojan is introduced.
3) The properties level comprises the behavior and physical characteristics of the trojan. It contains the taxonomy.
categories effect, logic type, functionality, activation and layout.

4) The location level corresponds to the location of the trojan in the IC.

The properties level has the following categories.

- The effect category describes the disruption or effect a trojan has on the system.
- The logic type category describes the circuit logic that triggers the trojan, either combinational logic or sequential.
- The functionality category differentiates between trojans which are functional or parametric.
- The activation category differentiates between trojans which are always on or are triggered.
- The layout category is based on the physical characteristics of the trojan.

A hardware trojan or detection method possesses a number of the thirty-three attributes in Fig. 1. Each trojan or detection method is assigned two values for each category, and these are given in the vectors \( I_P \) and \( C_P \). \( I_P \) is the identification vector which is used to differentiate between the possible attribute combinations. For a detection method, \( C_P \) is the effectiveness or coverage vector while for a trojan it is the severity or risk vector. Fig. 3 provides an example of the attribute combinations and the corresponding values for the effect category (denoted by \( I_E \) and \( C_E \)).

Table I provides a comparison of two hardware trojans. Trojan A has a lower severity than Trojan B in the insertion category, denoted by \( C_R \). This indicates that Trojan B can be inserted in more stages of the manufacturing process than Trojan A. Table II gives a similar comparison between two detection methods. The method in [19] has a higher coverage in the effect category than the method in [20], indicating that it can detect more trojans. Table II and Fig. 3 indicate that the method in [19] is capable of detecting attributes 12, 13 and 14 whereas the method in [20] is only capable of detecting attribute 12. Any trojans or detection methods can be compared in a similar manner. Note that when a comparison is performed the techniques may have equal values in some categories. For example, from Tables I and II, the detection methods in [19]
TABLE I: Severity Vectors for Two Hardware Trojans

<table>
<thead>
<tr>
<th>Technique</th>
<th>Parameters (I_P)</th>
<th>Severity (C_P)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>I_R I_A I_E I_L I_F I_C I_P I_O</td>
<td>C_R C_A C_E C_L C_F C_C C_P C_O</td>
</tr>
<tr>
<td>Trojan A [14]</td>
<td>2 6 2 1 1 2 1 7 7</td>
<td>2 6 4 1 2 1 5 2</td>
</tr>
<tr>
<td>Trojan B [14]</td>
<td>3 3 1 2 1 2 8 1</td>
<td>3 3 2 2 1 3 6 1</td>
</tr>
</tbody>
</table>

TABLE II: Coverage Vectors for Two Hardware Trojan Detection Methods

<table>
<thead>
<tr>
<th>Technique</th>
<th>Parameters (I_P)</th>
<th>Coverage (C_P)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>I_R I_A I_E I_L I_F I_C I_P I_O</td>
<td>C_R C_A C_E C_L C_F C_C C_P C_O</td>
</tr>
<tr>
<td>[19]</td>
<td>3 3 B 1 2 4 7 V 1</td>
<td>3 3 7 1 2 3 5 5 2</td>
</tr>
<tr>
<td>[20]</td>
<td>3 3 1 2 1 4 7 V 4</td>
<td>3 3 2 3 1 3 5 5 3</td>
</tr>
</tbody>
</table>

and [20] as well as Trojan B all have the same value in the $C_R$ category, which indicates that they have the same coverage and severity, respectively. Thus, the methods in [19] and [20] are capable of detecting Trojan B in the $C_R$ category.

III. THE DETECTION TOOL

When a new method of detecting hardware trojans is developed, it should be evaluated using current trojans and compared with other detection methods. The HTS detection tool allows developers to investigate trojan detection methods systematically and perform a quantitative analysis. A user can select values for each of the eight categories as well as provide the method name and description. The corresponding vector is then stored in the database. The HTS provides a number of automated tools such as for the trojan classification in [21] which employs the categorization scheme in [14]. This tool provides a means of determining the severity of trojans, and this can be saved in the database for use with the detection tool. The detection tool allows users to search the database for detection method coverage (effectiveness) and trojan severity (risk) vectors. Once vectors have been chosen for both a detection method and a trojan, a user can use the compare button to perform a comparison between the two vectors. For example, the results of a comparison are shown in the bottom row of Fig. 4. A 1 is displayed when the detection method has a value greater than or equal to the corresponding trojan value, and a 0 otherwise. The zeros in Fig. 4 indicate that the detection method will fail to detect the trojan in the specification phase of the IC life cycle ($I_R$) and based on the logic type ($I_F$).

While the detection tool can be employed for individual comparisons, its greatest potential is with the centralized database. Currently the tool only provides comparisons of trojans and detection methods entered by the user. Universal adoption of the HTS will provide a centralized database of all known methods and trojans. This database will allow the detec-

![Comparison](image)

**Fig. 4:** Detection, coverage and severity vectors.
tion tool to perform comprehensive searches for comparison results. Attackers can use this information to design trojans and defenders can use it to develop solutions for security weaknesses in their designs.

IV. THE WEB ENVIRONMENT

The hardware trojan detection tool was designed as a web utility for portability and easy distribution. The system is comprised of an application server which performs the computations and generates page markup to minimize the burden on client-side browsers. The server communicates directly with a remote database used to store user login and account information, application data (attributes, categories and hardware trojan matrices). Both the application server and the database are hosted on the Microsoft Azure Cloud platform. This platform provides reliability, portability and flexibility with on-demand resources that are automatically managed for scalability and the ability for maintenance to take place anywhere. Fig. 5 shows a block diagram of the automated tool.

The technologies employed are as follows.

- **ASP.NET Web Form**: A user interface focused, event-driven model of the .NET framework. It allows powerful data-binding, separation of server-client side activities, a native security structure, and increased client performance [22].
- **Entity Framework**: An object-relational database mapper designed for the .NET framework. It provides a library of high speed SQL statements wrapped in C# commands to simplify development and ensure performance [23].
- **D3.js**: A Java-script library for visualizing data with HTML, SVG and CSS [24].
- **Azure**: The Microsoft cloud system [25].

Fig. 5: Block diagram of the hardware trojan automated tool.

Fig. 6 provides an overview of the structure of the trojan system website. The home, contact, about, and application information pages are accessible to all traffic. The application information page contains three sub-pages providing information on each of the primary applications. Users are required to create an account and be logged in to access the remainder of the site. Email confirmation is used to verify user accounts.

V. CASE STUDY

Consider a circuit designed to compute a function \( F(x) \) for a system to authenticate user-password pairs \( x \) and \( F(x) \). The system performs the arithmetic operation \( F(x) = x^2 \) to validate users. The use case requires ten users \( I_0 \) to \( I_9 \) which requires four input bits. The largest function output is 81 meaning seven bits are required for output, \( Z_1 \) to \( Z_7 \), as illustrated in Fig. 7. A trojan can be inserted into this circuit as shown in Fig. 8 (called a backdoor trojan). The outputs of the original and infected circuits are compared in Table III.

Fig. 6: An overview of the website architecture.

Fig. 7: The trojan free circuit (Circuit A).

Fig. 8: The backdoor trojan circuit (Circuit B).
TABLE III: Outputs of the Circuits in Figs. 7 and 8 [14]

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Circuit A</th>
<th>Circuit B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z_1</td>
<td>Z_2</td>
<td>Z_3</td>
</tr>
<tr>
<td>Z_5</td>
<td>Z_6</td>
<td>Z_7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Circuit A</th>
<th>Circuit B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z_1</td>
<td>Z_2</td>
<td>Z_3</td>
</tr>
<tr>
<td>Z_5</td>
<td>Z_6</td>
<td>Z_7</td>
</tr>
</tbody>
</table>

The four input bits of the circuit \( x_1 \) to \( x_4 \) are used to identify the users \( I_0 \) to \( I_9 \). A simple test will show that the circuit outputs the desired \( F(x) = x^2 \) for each of the users. However, upon closer inspection it is noted that the inputs corresponding to \( x = 10 \) to \( x = 15 \) are not used. This is a typical vulnerability which can be exploited by an attacker. Inputs \( x = 10 \) and \( x = 11 \) to Circuit A produce results which are not correct according to \( F(x) = x^2 \). This is an intended result used as a security feature for these don’t care conditions. If an attacker is able to make the modification in Fig. 8, inputs 10 and 11 will provide correct results according to \( F(x) = x^2 \), which can be used to allow access to the system by a malicious user. From the previous description of the trojan and referring to Fig. 1, this trojan is comprised of combinational logic (17), its functionality is functional (18), and it is externally triggered (22). Assuming that this trojan was used to create a denial of service (DoS) attack (15), the attributes of this trojan are 15, 17, 18 and 22. The web tool provides an easy to use on-line store style selection when entering input. When the system performs the analysis, a directed graph representation of the result of the classification method in [21] is displayed as shown in Fig. 9. The HTS provides a vector representing the severity of the trojan in which is shown in Fig. 10. There is an option to save the results with a title and description. The user is then
able to move to the detection comparison page shown in Fig. 4. On this page, the saved severity vector can be selected and loaded into the detection tool along with any saved detection method to perform a comparison.

On the detection tool page, a coverage vector for a new detection method was created, as shown in Fig. 11, named method5, and saved in the database. The coverage vector for this method was then selected along with the severity for the trojan. Choosing the comparison button provides the results in the bottom row of Fig. 12. The red zeros show the user in which of the eight categories the detection method fails to detect the trojan.

VI. Conclusion

The relatively new field of hardware security has yet to provide a structure that allows developers and security professionals an efficient means of collaboration. The Hardware Trojan System (HTS) is an automated tool that can be employed to simplify the investigation and development of hardware trojans and detection techniques. The universal adoption of this system will provide cohesion to the field of hardware security, and centralizing the techniques in the HTS will expedite adoption. The generation of vectors corresponding to the category attributes for both trojans and detection methods will aid in trojan generation as well as countermeasures. These vectors allow quick and accurate assessment based on the attributes. Defenders concerned with a particular vulnerability can browse the database for an appropriate detection technique. An attacker who has found a weakness in a particular system can search for existing trojans that can be used. The automated tool allows developers to generate and store values which can be used to compare trojans. The severity and coverage vectors are stored in the database along with descriptive information provided by the user. A case study was provided which demonstrates the ease of use of this tool and its usefulness.

The implementation of data mining techniques and an interface that allow users to browse the archive of saved work is left for future work. The addition of this functionality will provide a centralized database of known hardware trojans and detection techniques.

REFERENCES


