Tsinghua Science and Technology

Volume 26 | Issue 1 Article 9

2021

HTDet: A Clustering Method Using Information Entropy for Hardware Trojan Detection

Renjie Lu
the University of Chinese Academy of Sciences, Beijing 101408, China.

Haihua Shen
the University of Chinese Academy of Sciences, Beijing 101408, China.

Zhihua Feng
Beijing Institute of Computer Technology and Application, Beijing 100854, China.

Huawei Li
the State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China.

Wei Zhao
the University of Chinese Academy of Sciences, Beijing 101408, China.

See next page for additional authors

HTDet: A Clustering Method Using Information Entropy for Hardware Trojan Detection

Authors

Renjie Lu, Haihua Shen, Zhihua Feng, Huawei Li, Wei Zhao, and Xiaowei Li

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 05/10 pp48–61
DOI: 10.26599/TST.2019.9010047
Volume 26, Number 1, February 2021

HTDet: A Clustering Method Using Information Entropy for Hardware Trojan Detection

Renjie Lu, Haihua Shen*, Zhihua Feng, Huawei Li, Wei Zhao, and Xiaowei Li

Abstract: Hardware Trojans (HTs) have drawn increasing attention in both academia and industry because of their significant potential threat. In this paper, we propose HTDet, a novel HT detection method using information entropy-based clustering. To maintain high concealment, HTs are usually inserted in the regions with low controllability and low observability, which will result in that Trojan logics have extremely low transitions during the simulation. This implies that the regions with the low transitions will provide much more abundant and more important information for HT detection. The HTDet applies information theory technology and a density-based clustering algorithm called Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to detect all suspicious Trojan logics in the circuit under detection. The DBSCAN is an unsupervised learning algorithm, that can improve the applicability of HTDet. In addition, we develop a heuristic test pattern generation method using mutual information to increase the transitions of suspicious Trojan logics. Experiments on circuit benchmarks demonstrate the effectiveness of HTDet.

Key words: Hardware Trojan (HT) detection; information entropy; Density-Based Spatial Clustering of Applications with Noise (DBSCAN); unsupervised learning; clustering; mutual information; test patterns generation

1 Introduction

With the development of society, security issues have become the focus of attention, such as secure DHCPv6 mechanism, secure web, and secure authentication protocol for mobile payment, and so on[1–5]. The globalization of the modern Integrated Circuit (IC) industry has also raised increasing hardware security issues. For example, Intellectual Property (IP) cores provided by third parties are widely used in IC design to reduce development cost and shorten the marketing cycle[6]. As the third-party IP cores are designed by outsourced vendors, an adversary can easily implement some malicious logics, referred to as Hardware Trojans (HTs), into IP cores.

HTs are lightweight structures in large-scale IC designs, which commonly contain two components: Trojan trigger and Trojan payload[7]. The Trojan trigger is responsible for monitoring signals to determine whether the trigger signal has arrived. If the Trojan trigger is not activated, HTs stay dormant and do not affect the original circuit. If the Trojan trigger is activated, the Trojan payload will perform specific malicious operations, such as changing functionality, degrading performance, and revealing secret information[8]. Since most of HTs usually have extremely rare trigger conditions, it is very challenging to detect suspicious Trojan logics in the Circuit Under Detection (CUD).

The existing HT detection techniques can be roughly classified into six major groups: reverse

• Renjie Lu, Haihua Shen, and Wei Zhao are with the University of Chinese Academy of Sciences, Beijing 101408, China. E-mail: lurenjie17@mails.ucas.ac.cn; shenhh@ucas.ac.cn; zhaowei163@mails.ucas.ac.cn.
• Zhihua Feng is with Beijing Institute of Computer Technology and Application, Beijing 100854, China. E-mail: zhihua_feng@126.com.
• Huawei Li and Xiaowei Li are with the State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China. E-mail: lihuawei@ict.ac.cn; lxw@ict.ac.cn.
* To whom correspondence should be addressed.
Manuscript received: 2019-04-01; revised: 2019-08-23; accepted: 2019-08-29

engineering[9–11], side-channel analysis[12–18], Gate-Level Information Flow Tracking (GLIFT)[19–21], static structure analysis[22–27], statistical feature analysis[28–32], and functional testing[33–36]. In reverse engineering, a fabricated chip is completely dissected layer-by-layer in order to reconstruct the IC design to detect malicious modifications. Reverse engineering approaches consume prohibitively high cost, and it is impossible to carry out reverse engineering for each chip under test. In side-channel analysis, the impacts of HTs on circuit delay, transient current, and leakage power, and so on, can be used to detect whether there are the HTs in CUD. Side-channel analysis approaches can detect HTs inserted in the post-fabrication stage. However, side-channel analysis usually requires a "golden circuit" for impact comparison and also it is susceptible to process variations or environmental noise, which can result in many false positives. GLIFT-based Trojan detection techniques rely on gate-level information flow tracking to detect Trojans. To account for hardware specific information flow, the GLIFT technique tracks information flow through Boolean gates. At the gate level, all information flow appears at the most basic level of abstraction which allows detecting information flow that is inherently not visible at the software level. Like software virus detection technique, static structure analysis methods detect HTs by analyzing the circuit structure characteristics. Although the static structure analysis is an effective HT detection approach, it can only detect known types of HTs. Intrinsic differences exist between Trojan logics and normal circuit; therefore, statistical feature analysis approaches can be used to detect potential HTs in CUD. Functional testing approaches try to generate test vectors to activate potential HTs and propagate HTs' effects on the primary outputs. Although functional testing is independent of process variations and environmental noise, it usually consumes a significant amount of time due to the high concealment of HTs.

The key insight of our approach is that HTs are usually inserted in the regions with low controllability and low observability in order to maintain high concealment, which will result in Trojan logics featuring extremely low transitions during the simulation. In the field of information theory, if an event is improbable, much more information will be provided when the event happens; that is, the logical regions with the very low transitions will provide us with much more abundant and more important information for Trojan detection. In this paper, we propose a novel HT detection method using information entropy-based clustering, calld HTDet. First, digital stimuli are generated for the CUD. Then the information entropy of the signal sequence of each wire is calculated, and a typical density-based clustering algorithm called Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is applied to detect all suspicious Trojan logics. Furthermore, a heuristic test pattern generation method using mutual information is developed to increase the transitions of these suspicious Trojan logics. In summary, this paper has the following contributions.

- To the best of our knowledge, this is the first attempt to use information entropy technology to detect HTs in the hardware design, and the proposed HTDet can achieve good experimental results.
- An unsupervised learning algorithm, DBSCAN, is used for Trojan detection, which means that HTDet does not require "golden circuit". Furthermore, HTDet does not need to trigger Trojan logics. As long as the information entropy of the circuit logic is extremely low, HTDet can detect them based on the density-reachable relationship.
- We develop a heuristic test pattern generation method using mutual information technology to increase the transitions of suspicious Trojan logics.
- We conducted lots of evaluations on TrustHub benchmarks[37], which showed that HTDet can effectively detect suspicious Trojan logics with negligible false positives.

The rest of this paper is organized as follows. Sections 2 and 3 introduce the theoretical basis and the threat model, respectively. We present the methodology of the HTDet in detail in Section 4. Section 5 presents the test pattern generation method for suspicious Trojan logics. Experimental analysis is presented in Section 6. Section 7 briefly summarizes the related works. Finally, we conclude this paper in Section 8.

2 Theoretical Basis

In this paper, we perform the HT detection using information theory technology[38]. In this section, we provide the theoretical basis of the proposed approach.

2.1 Information entropy

Information entropy is also known as the self-information, which is the average rate at which information is produced by a data source. Entropy is a measure of uncertainty associated with a random

variable.

Let X be a discrete random variable, and its probability distribution be consistent with $p(x) = P(X = x)$, where $x \in X$; hence, the entropy $H(X)$ of X can be explicitly written as

$$H(X) = -\sum_{x \in X} p(x) \log_b p(x) \tag{1}$$

where b is the base of the logarithm used. In this paper, b is equal to the mathematical constant e. In the case of $p(x) = 0$, the value of $0 \log_b 0$ is taken to be 0, which is consistent with the limit,

$$\lim_{p(x) \to 0^+} p(x) \log_b p(x) = 0 \tag{2}$$

2.2 Joint entropy

In information theory, joint entropy is a measure of the uncertainty associated with a set of variables. In this paper, we focus on the joint entropy of two random variables.

Similarly, let X and Y be two discrete random variables, and their probability distribution be $p(x, y)$, where $x \in X$ and $y \in Y$; hence, the joint entropy $H(X, Y)$ of X and Y can be presented as

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log_b p(x, y) \tag{3}$$

2.3 Conditional entropy

In information theory, the conditional entropy quantifies the amount of information needed to describe the outcome of a random variable Y when the value of another random variable X is known.

The entropy $H(Y|X)$ of Y conditioned on X can be defined as follows,

$$\begin{aligned} H(Y|X) &= \sum_{x \in X} p(x) H(Y|X = x) = \\ &\sum_{x \in X} p(x) \left[-\sum_{y \in Y} p(y|x) \log_b p(y|x) \right] = \\ &-\sum_{x \in X} \sum_{y \in Y} p(x, y) \log_b p(y|x) \end{aligned} \tag{4}$$

It is worth noting that $H(X)$, $H(X, Y)$, and $H(Y|X)$ can conform to the chain rule, that is

$$\begin{aligned} &H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log_b p(x, y) = \\ &-\sum_{x \in X} \sum_{y \in Y} p(x, y) \log_b [p(x) p(y|x)] = \\ &-\sum_{x \in X} \sum_{y \in Y} p(x, y) [\log_b p(x) + \log_b p(y|x)] = \\ &-\sum_{x \in X} p(x) \log_b p(x) - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log_b p(y|x) = \\ &H(X) + H(Y|X) \end{aligned} \tag{5}$$

2.4 Mutual information

The mutual information of two variables is a measure of the mutual dependence between the variables. More specifically, the mutual information quantifies the amount of information obtained about one random variable by observing the other random variable.

Let X and Y be two discrete random variables, and their joint probability distribution be $p(x, y)$; hence, the mutual information $I(X; Y)$ between X and Y can be defined as

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log_b \frac{p(x, y)}{p(x) p(y)} \tag{6}$$

According to the relationships among probability distributions and the chain rule, $I(X; Y)$ can also be expressed as

$$\begin{aligned} I(X;Y) &= \sum_{x \in X} \sum_{y \in Y} p(x, y) \log_b \frac{p(x|y)}{p(x)} = \\ &\sum_{x \in X} \sum_{y \in Y} p(x, y) [\log_b p(x|y) - \log_b p(x)] = \\ &H(X) - H(X|Y) = \\ &H(X) + H(Y) - H(Y, X) \end{aligned} \tag{7}$$

3 Threat Model

The threat model of the proposed method is based on several assumptions.

- With the globalization of chip design, the adversaries can have more opportunities to insert HTs into a digital circuit design than before. It can be the gate-level netlist or register transfer language.
- Our threat model assumes that the given hardware design is in the form of a digital circuit design.
- The goal of the attack is to change functionality, destroy the IC, and/or leak secret information through logical attack, rather than through side-channels such as current, power, or electromagnetism.

4 HTDet Methodology

In this section, first, we provide the feasibility analysis of HTDet. Then the technical details of HTDet are presented. The core problem is whether the information entropy technology and clustering algorithm can be used to effectively detect suspicious Trojan logics in the CUD.

4.1 Feasibility analysis

The key insight of HTDet is that the significant differences exist between the Trojan logics and the rest of the circuit. More specifically, the HT is usually inserted in the regions with low controllability and low observability, which causes the Trojan logic to have a very low transition probability. Moreover, in the field of information theory[38], if an event is very probable, little information is provided when it happens. Conversely, if an event is improbable, much more information will be provided when it happens.

That is, the regions with low transitions will provide us with more abundant and more important information for HT detection. However, directly applying the transition probability for Trojan detection will result in high false positives. For example, we consider that the signal wires (from W_1 to W_{14}) have the transition probabilities listed in Table 1.

Due to the density-reachable relationship between low transition probabilities and high transition probabilities, signal wires from W_1 to W_{10} can be reported as suspicious Trojan logics as shown in Fig. 1 (blue line), while the use of information entropy can significantly reduce false positives. As shown in Fig. 1 (orange line), signal wires from W_1 to W_7 can be reported as suspicious Trojan logics.

This is because the information entropy can cause a gap in the connectivity between low transition probabilities and high transition probabilities, and it is more sensitive to low transition probabilities as shown in Fig. 2. It can be seen that the density-reachable relationship between signal wires (from W_1 to W_7) is much closer than the density-reachable relationship between low transition probabilities and high transition probabilities.

Table 1 Signal wires and corresponding transition probabilities.

Wire	Transition probability	Wire	Transition probability
W_1	$\frac{1}{1000}$	W_8	$\frac{1}{20}$
W_2	$\frac{1}{800}$	W_9	$\frac{1}{10}$
W_3	$\frac{1}{500}$	W_{10}	$\frac{1}{8}$
W_4	$\frac{1}{200}$	W_{11}	$\frac{1}{5}$
W_5	$\frac{1}{100}$	W_{12}	$\frac{3}{10}$
W_6	$\frac{1}{80}$	W_{13}	$\frac{1}{2}$
W_7	$\frac{1}{50}$	W_{14}	$\frac{6}{10}$

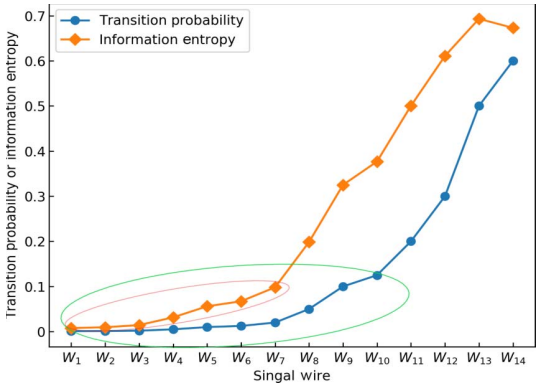


Fig. 1 HT detection comparison between transition probability and information entropy.

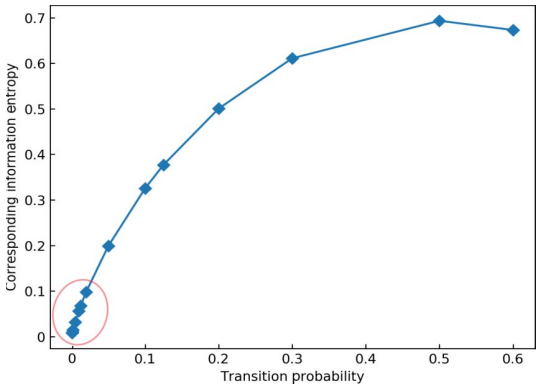


Fig. 2 Distribution of information entropy for probabilities listed in Table 1.

probabilities.

It has been proven that the information entropy takes the maximum value when $p(\text{transition})$ is equal to $p(\text{non-transition})$. In other words, when $p(\text{transition}) = p(\text{non-transition}) = 0.5$, the corresponding information entropy can take the maximum value. According to Eq. (1), the transition probability-information entropy curve is shown in Fig. 3. Because the information entropy has symmetry, the minimum value can be taken when

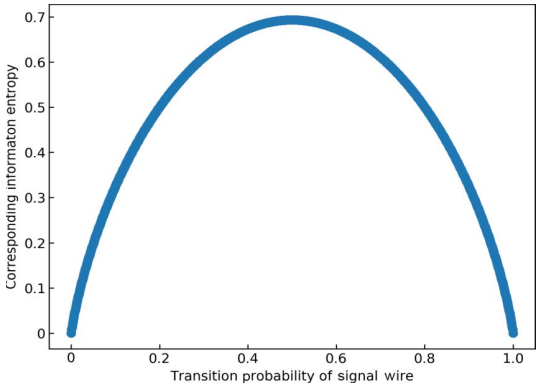


Fig. 3 Transition probability-information entropy curve.

$p(\text{transition}) = 0$ or $p(\text{transition}) = 1$. Based on this conclusion, we should exclude the noise data that have very low information entropy due to very high transition probability.

In addition, the mutual information technology can measure the correlations between primary inputs and internal signal wires, which is beneficial to test patterns generation. Therefore, we first propose applying the information theory technology in the field of HT detection.

4.2 Application of information entropy

To apply the information entropy technology for HT detection, we first use functional testing to generate digital stimuli for the CUD. We believe that the set of test patterns developed during design verification can satisfy this step. The goal of this step is to perform functional testing for the CUD with high coverage as much as possible. After the functional testing, we can obtain the original waveform of each signal wire in the CUD, which contains only binary values (0 or 1). Our goal is to use the information entropy to evaluate the controllability and observability of each circuit logic such that we can effectively distinguish Trojan logics from the rest of the circuit.

However, we cannot directly use the original waveform for HT detection. For example, the signal transition occurs only once in OW_1, while OW_2 has five signal transitions, as shown in Fig. 4a. Because the HTs usually are inserted in the regions with low controllability and low observability, which cause the Trojan logic to have a very low transition probability. Hence, the logical region of OW_1, rather than that of OW_2, is more likely to be a Trojan logic. However, because the probabilities of 0 and 1 in OW_1 are the same as in OW_2, the information entropies of both OW_1 and OW_2 are 0.6931 according to Eq. (1).

We should focus on the distribution of signal transitions rather than the distribution of 0 and 1 such that we can use the information entropy to evaluate the controllability and observability of each circuit logic. To this end, we encode the original waveform according to the following rules. We assume that the original waveform $\text{OW} = \langle s_1, s_2, \ldots, s_n, s_{n+1}\rangle$. For each signal pair $\langle s_i, s_{i+1}\rangle$, $i = 1, 2, \ldots, n$, if $\langle s_i, s_{i+1}\rangle = \langle 0, 0\rangle$, we encode s_i as 0; if $\langle s_i, s_{i+1}\rangle = \langle 0, 1\rangle$, we encode s_i as 1; if $\langle s_i, s_{i+1}\rangle = \langle 1, 0\rangle$, we encode s_i as 1; if $\langle s_i, s_{i+1}\rangle = \langle 1, 1\rangle$, we encode s_i as 0. The encoded waveforms of the original waveforms (OW_1 and OW_2) are shown in Fig. 4b. Then, we use Eq. (1) to calculate the information entropy of each encoded waveform. The information entropy of Encode_1 (corresponding to OW_1) is approximately equal to 0.3488, and the information entropy of Encode_2 (corresponding to OW_2) is approximately equal to 0.6870, which is more in line with the expected results.

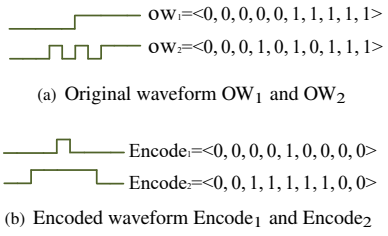


Fig. 4 Comparison between original waveform and encoded waveform.

We apply the information entropy to distinguish the differences between Trojan logics and the normal circuit. As shown in Fig. 5, we can obtain the information entropy of each wire in the given circuit after functional testing (10^6 cycles). It can be seen that the information entropy at the output of the AND gate is 0.138 20, that at the input (top) of the AND gate is 0.229 66, and that at the input (bottom) of the AND gate is 0.662 71 due to different circuit structures. Lots of experiments demonstrated that the information entropy of each wire was almost consistent with the controllability measure[39] of this signal wire.

4.3 HT detection-based clustering

It is worth noting that our circuit analysis focuses on the states of internal wires in CUD rather than circuit structures. For convenience of discussion, we define $\text{CUD} = \langle \text{PI}, W, \text{POUT}\rangle$, where PI is the set of primary inputs, W is the set of internal signal wires, and POUT is the set of primary outputs. More formally, $\text{PI} = \{\text{pi}_1, \text{pi}_2, \ldots, \text{pi}_l\}$, $W = \{w_1, w_2, \ldots, w_m\}$, and $\text{POUT} = \{\text{pout}_1, \text{pout}_2, \ldots, \text{pout}_n\}$. After functional testing, we encode each original waveform of CUD and calculate the information entropy of each encoded

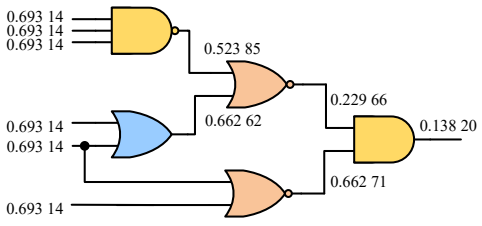


Fig. 5 Information entropy of each wire in the given circuit fragment.

waveform. Once the above step is complete, we apply a typical density-based clustering algorithm called DBSCAN[40] to perform HT detection in the information entropy space composed by W and POUT.

In the given data space, the density is defined as the number of data points within a specified radius (r), and the core point has more than the specified number of data points (MinPts) within its r-neighborhood, and the border point has less than MinPts within its r-neighborhood, but it is in the r-neighborhood of a core point, and any point that is not a core point or border point is called a noise point. Moreover, date point q is directly density-reachable from another point p, if p is a core point and q is within the r-neighborhood of p. Data point q is density-reachable from another point p, if there is a path of points $p_1(p) \rightarrow p_2 \rightarrow \cdots \rightarrow p_{n-1} \rightarrow p_n(q)$ such that point p_{i+1} is directly density-reachable from point p_i. Data points p and q are density-connected, if there is a data point o, such that both p and q are density-reachable from o.

The basic idea of the DBSCAN is to find the maximal set of density-connected points. In other words, all points within the same cluster are mutually density-connected. Algorithm 1 shows the clustering process in the information entropy space.

Algorithm 1 HT detection-based clustering

Input: Information Entropy Space (IES), r, MinPts
Output: Suspicious Trojan logics

function FINDCOREPOINT(IES, r, MinPts)
  $C = 0$
  for $\forall$ unvisited point $P \in$ IES **do**
    mark P as visited
    NeighborPts $\leftarrow$ all points within P's r-neighborhood
    if size of NeighborPts $<$ MinPts **then**
      mark P as noise point
    else
      $C =$ **next** cluster
      Clustering(P, NeighborPts, C, r, MinPts)
function CLUSTERING(P, NeighborPts, C, r, MinPts)
  add P to cluster C
  for $\forall$ point $Q \in$ NeighborPts **do**
    if Q is not visited **then**
      mark Q as visited
      Q_NPts $\leftarrow$ all points within Q's r-neighborhood
      if size of Q_NPts $\geqslant$ MinPts **then**
        NeighborPts $\leftarrow$ NeighborPts $\cup$ Q_NPts
    if Q is not yet member of any cluster **then**
      add Q to cluster C
function REPORTTROJANS(all clusters)
  Report the cluster with lowest average information entropy as suspicious Trojan logics.

5 Test Patterns Generation for Suspicious Trojan Logics Using Mutual Information

As described in Section 4, the proposed HT detection method can find suspicious Trojan logics. This section introduces a heuristic test pattern generation method using mutual information, which can further increase the transitions of suspicious Trojan logics. As is depicted in Fig. 6, the correlation between each suspicious Trojan logic and each primary input is measured by the mutual information. If the mutual information is greater than the threshold, the corresponding primary input is strongly correlated to this suspicious Trojan logic and is referred to as Strongly Correlated Primary Input (SCPI). Therefore, each suspicious Trojan logic will maintain a Set of SCPI (SSCPI). Then, a heuristic method is developed to select minimum SCPIs while covering all suspicious Trojan logics.

5.1 Feasibility analysis

In the field of information theory, the mutual information between X and Y can measure the mutual dependence between the two variables; that is, the mutual information can measure the correlation between two variables[41]. If X and Y are independent, their mutual information is zero. If X is a deterministic function of Y(Y also is a deterministic function of X), knowing the value of X can determine the value of Y and vice versa. In this case, the mutual information between X and Y is the same as the $H(X)$ and as the $H(Y)$.

Naturally, each circuit logic can be expressed as a Boolean function of different primary inputs, which conforms to the statement of the correlation. For

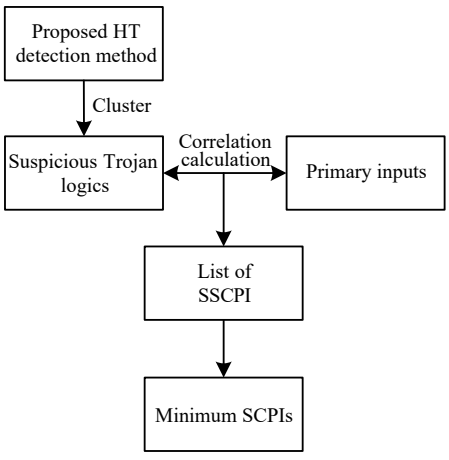


Fig. 6 Overview of test pattern generation method.

example, we can obtain three Boolean functions $d = ab$, $e = \overline{c}$, and $f = ab + \overline{c}$ for the circuit structure shown in Fig. 7. Hence, we can know that d and c, e and a, and e and b are independent, such that their mutual information must be zero, and e is a deterministic function of c, such that their mutual information is the same as $H(c)$ and $H(e)$, and the mutual information $I(d;a)$ should be equal to the mutual information $I(d;b)$, because they are of the same circuit logic. It is worth noting that the mutual information $I(f;a)$ is different from the mutual information $I(f;c)$, because they are of different circuit logics (AND gate and inverter). In short, the higher the mutual information of two variables, the stronger the variables correlation.

5.2 Correlation calculation using mutual information

We consider that the set of primary inputs PI $= \{\mathrm{pi}_1, \mathrm{pi}_2, \ldots, \mathrm{pi}_l\}$, and consider that the set of suspicious Trojan logics SW $= \{\mathrm{sw}_1, \mathrm{sw}_2, \ldots, \mathrm{sw}_t\}$, where $t \leqslant m + n$. First, we calculate mutual information $I(\mathrm{sw}_i; \mathrm{pi}_j)$ between each suspicious Trojan logic sw_i and each primary input pi_j, where $i = 1, 2, \ldots, t$ and $j = 1, 2, \ldots, l$. According to Eq. (7), $I(\mathrm{sw}_i; \mathrm{pi}_j) = H(\mathrm{sw}_i) + H(\mathrm{pi}_j) - H(\mathrm{pi}_j, \mathrm{sw}_i)$. Because each encoded waveform only contains 0 (non-transition) and 1 (transition), $H(\mathrm{pi}_j, \mathrm{sw}_i) = -\sum_{\mathrm{pi}_j \in \{0,1\}} \sum_{\mathrm{sw}_i \in \{0,1\}} p(\mathrm{pi}_j, \mathrm{sw}_i) \log_b p(\mathrm{pi}_j, \mathrm{sw}_i)$, according to Eq. (3). If $I(\mathrm{sw}_i; \mathrm{pi}_j)$ is greater than the threshold, we refer to the primary input pi_j as the SCPI of suspicious Trojan logic sw_i. For each sw_i, the threshold is equal to $\sum_{\mathrm{pi}_j \in \mathrm{PI}} \dfrac{I(\mathrm{sw}_i; \mathrm{pi}_j)}{l}$, where l is the number of primary inputs. Finally, each suspicious Trojan logic will have an SSCPI. The strong correlations between primary inputs and suspicious Trojan logics can constitute a strong correlation list, as shown in Table 2.

5.3 Test patterns generation

Our goal is to select the minimum number of SCPIs while covering all suspicious Trojan logics. We define $\{\mathrm{pi}_j\}$ to be a set of suspicious Trojan logics whose

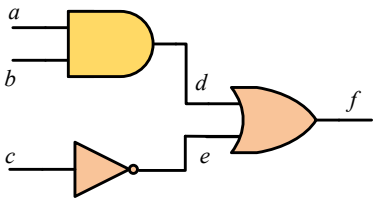


Fig. 7 Mutual information analysis for given circuit structure.

Table 2 Strong correlation list: 1 indicates pi_j is an SCPI of sw_i and 0 indicates not.

SW	PI				
	pi_1	pi_2	pi_3	$\cdots$	pi_l
sw_1	1	0	1	$\cdots$	1
sw_2	0	1	1	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
sw_t	1	1	0	$\cdots$	1

SSCPI includes pi_j, and define the "+" operation between sets is equivalent to the "union" operation between sets, while the "-" operation between sets is equivalent to the "difference" operation between sets. For example, $\{\mathrm{pi}_1\} = \{\mathrm{sw}_1, \mathrm{sw}_t\}$, $\{\mathrm{pi}_l\} = \{\mathrm{sw}_1, \mathrm{sw}_2, \mathrm{sw}_t\}$, $\{\mathrm{pi}_1\} + \{\mathrm{pi}_l\} = \{\mathrm{sw}_1, \mathrm{sw}_2, \mathrm{sw}_t\}$, and $\{\mathrm{pi}_l\} - \{\mathrm{pi}_1\} = \{\mathrm{sw}_2\}$. Therefore, the problem can be abstracted as the following formula, where $x_j \in \{0, 1\}$. If pi_j is selected, $x_j = 1$; otherwise $x_j = 0$.

$$\begin{aligned} &\min \sum_{x_j \in \{0,1\}} x_j \\ &\text{s.t.} \sum_{\mathrm{pi}_j \in \mathrm{PI}} x_j \times \{\mathrm{pi}_j\} = \mathrm{SW} \end{aligned} \tag{8}$$

$$f(k, y) = \begin{cases} \min\{f(k-1, y), f(k-1, y-\{\mathrm{pi}_k\})+1\}, & \\ & \text{if } \{\mathrm{pi}_k\} \subseteq y; \\ f(k-1, y), & \text{otherwise} \end{cases} \tag{9}$$

We develop a heuristic method to solve this problem. Here, $f(k, y)$ indicates the optimal solution when PI $= \{\mathrm{pi}_1, \ldots, \mathrm{pi}_k\}$ and SW $= y$. As shown in Eq. (9), $f(l, \mathrm{SW})$ is the optimal solution of Eq. (8). Then we perform constrained-random simulation, setting all the primary input at logic 0 or logic 1, which is not in SCPIs. For the rest of the primary inputs in SCPIs, we still generate full-random stimuli to perform simulation.

6 Experiments and Evaluations

The HTDet was evaluated on different digital circuit designs from the TrustHub benchmark[37]. All circuits were synthesized by Synopsys Design Compiler (DC) with Semiconductor Manufacturing International Corporation cell library for 90-nm silicon-on-insulator process. All circuits were simulated by Verilog compiler simulator with a high coverage. We conducted data processing experiments and data analysis experiments on a computer with 2.8 GHz Intel Core i7 CPU and 8 GB memory[42]. Brief information about the benchmarks used in our experiments is provided in Table 3.

Table 3 Brief information of the circuits under detection.

Circuit	Number of units	Features of HT
RS232_T1000	215	Trojan trigger is a combinational comparator; change functionality
RS232_T1100	217	Trojan trigger is a sequential comparator; change functionality
RS232_T1200	216	Trojan trigger is a sequential comparator; change functionality
RS232_T1300	213	Trojan trigger is a combinational comparator; change functionality
RS232_T1400	215	Trojan trigger is a sequential comparator; change functionality
RS232_T1500	216	Trojan trigger is a sequential comparator; change functionality
RS232_T1600	214	Trojan trigger is a sequential comparator; change functionality
s15850_T100	2182	Trojan trigger consists of two comparators and two flip-flops; leak an internal signal
s35932_T200	5438	Trojan trigger is a comparator; denial of service
s38417_T100	5341	Trojan trigger is a comparator; change functionality, denial of service

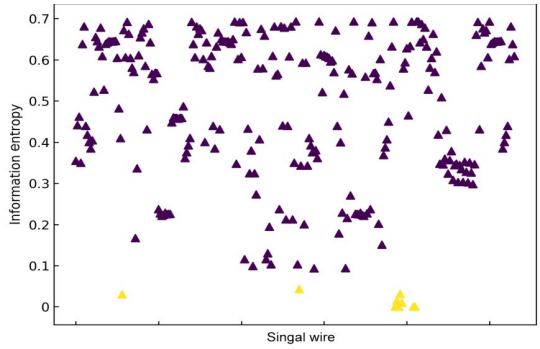
6.1 Clustering comparison between information entropy space and transition probability space

In our experiments, HTDet could detect all suspicious Trojan logics in the CUD. Taking RS232_T1000 and RS232_T1100 as examples, we present the differences of clustering between the information entropy space and transition probability space. Figures 8a and 8b show the results of clustering using information entropy for RS232_T1000 and RS232_T1100 benchmarks, respectively.

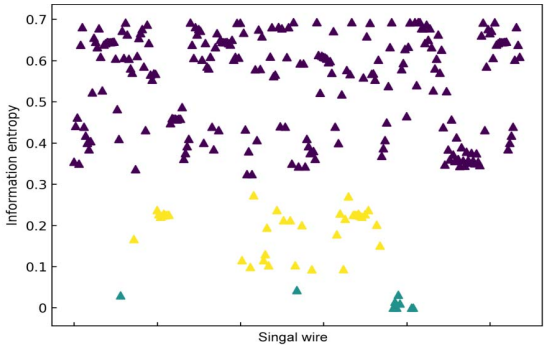
According to Algorithm 1, the cluster with the lowest average information entropy is reported as suspicious Trojan logics. As shown in Fig. 8, although HTDet could differentiate the information entropy space into several clusters (the points of the same color represent the same cluster), the circuit logics with extremely low information entropy were always divided into one cluster according to the density-reachable relationship.

Similarly, we also used transition probability for Trojan detection. Figures 9a and 9b show the results of clustering for RS232_T1000 and RS232_T1100, respectively.

It can be seen that the use of transitions will result in high false positives. However, the information entropy can effectively distinguish the Trojan logics from the normal circuit. To have more insight on the difference

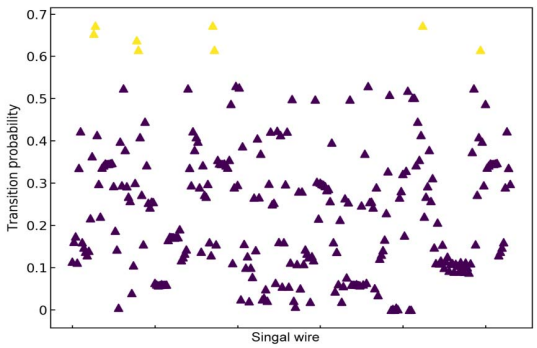


(a) Clustering for RS232_T1000 benchmark

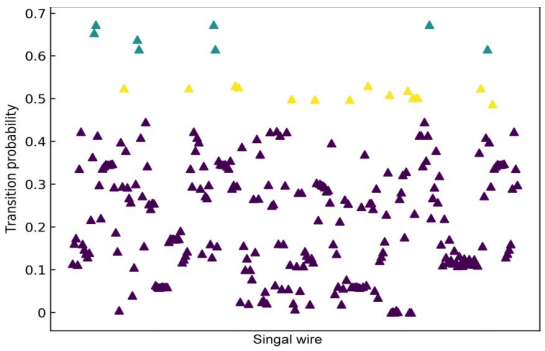


(b) Clustering for RS232_T1100 benchmark

Fig. 8 Clustering in information entropy space.



(a) Clustering for RS232_T1000 benchmark



(b) Clustering for RS232_T1100 benchmark

Fig. 9 Clustering in transition probability space.

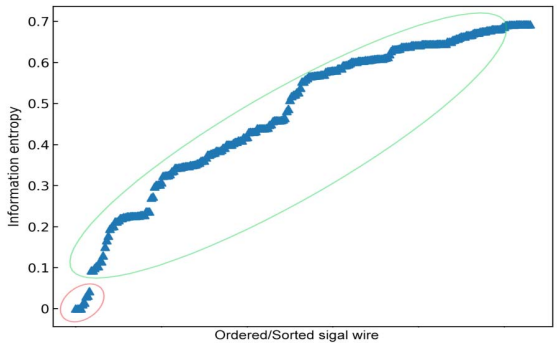
between information entropy and transition probability, we sorted the information entropy space and transition probability space of the RS232_T1000 benchmark from lowest to highest. The distributions of information entropy and transition probability are illustrated in Fig. 10.

As shown in Fig. 10a, the area with low information entropy (red) and other areas (green) have a clear density-unreachable relationship. However, the area with low transition probability and other area are still density-reachable (red), as shown in Fig. 10b, which will lead to poor Trojan detection. Because the information entropy can amplify the difference between low transition probability and high transition probability, it can effectively detect suspicious Trojan logics.

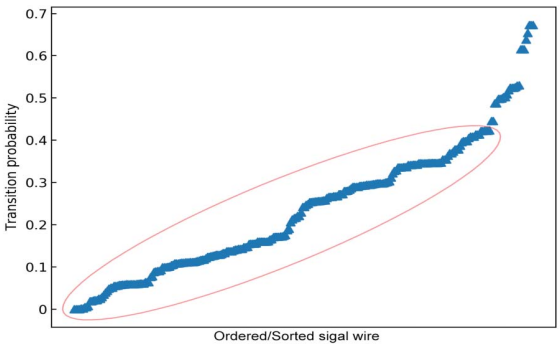
6.2 HT detection performance and parameter analysis

To further evaluate the effectiveness of the HTDet, we manually checked the suspicious Trojan logics reported by Algorithm 1. The results are presented in Table 4. MinPts and r are the parameters used in the clustering process.

The sensitivity of the results is measured by the True



(a) Distribution of information entropy



(b) Distribution of transition probability

Fig. 10 Difference between information entropy space and transition probability space for RS232_T1000 benchmark.

Table 4 Results of manual check.

Circuit	MinPts	r	TPR (%)	TNR (%)
RS232_T1000	2	0.05	62	99
RS232_T1100	5	0.04	67	99
RS232_T1200	5	0.04	89	99
RS232_T1300	2	0.05	89	99
RS232_T1400	5	0.04	61	99
RS232_T1500	5	0.04	73	99
RS232_T1600	5	0.04	62	99
s15850_T100	4	0.05	96	99
s35932_T200	5	0.05	93	99
s38417_T100	4	0.05	100	99

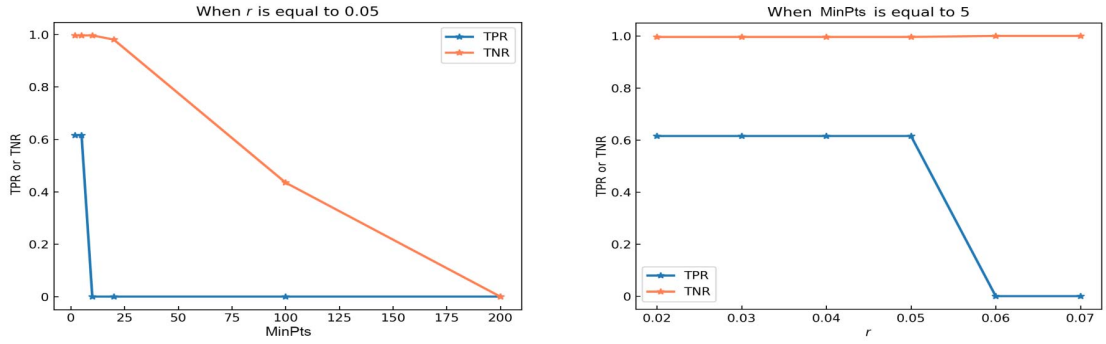
Positive Rate (TPR), i.e., the number of Trojan logics correctly detected as a percentage of the total number of Trojan logics. We also provide the True Negative Rate (TNR) results, which tell us the ratio of the true negatives over the number of non-Trojan logics. The False Positive Rate (FPR) is the fraction of logics that are falsely flagged as being suspicious Trojan logics, which is equal to 1-TNR. It can be seen that the HTDet can effectively detect Trojan logics of CUD with the extremely low false positives.

We also analyzed the effect of parameters MinPts and r on the HT detection performance. The experimental results indicate that the appropriate values of parameters are also necessary for Trojan detection. Taking RS232_T1000 as an example, when r was fixed to 0.05, both TPR and TNR declined as MinPts increased, as shown in Fig. 11a. This is because the number of noise point gradually increased with MinPts. Similarly, when MinPts was fixed to 5 and r increased, the TPR gradually declined but the TNR was almost constant, as shown in Fig. 11b. This is because all data points were clustered as normal logics when r was equal to 0.06 or 0.07.

6.3 Comparison with existing methods

We compared the experimental results with existing supervised learning methods[23–25], which include the support vector machine, multi-layer neural network and random forest, to detect HT. Table 5 summarizes the results. Compared with the methods proposed in Refs. [23] and [24], HTDet could greatly improve the TNR, although the obtained TPR was slightly reduced. Compared with the method in Ref. [25], HTDet could improve the average TPR by 4.7% and had more stable experimental results. It can be seen that the HTDet could obtain 79% average TPR and 99% average TNR, which is a better trade-off between TPR and TNR.

We also compared the experimental results with



(a) Effect of MinPts on TPR and TNR

(b) Effect of r on TPR and TNR

Fig. 11 Parameter analysis on RS232_T1000 benchmark.

Table 5 Comparison with the supervised learning methods.

(%)

Circuit	TPR				TNR			
	Method in [23]	Method in [24]	Method in [25]	HTDet	Method in [23]	Method in [24]	Method in [25]	HTDet
RS232_T1000	53	100	100	62	31	24	99	99
RS232_T1100	58	78	50	67	27	25	99	99
RS232_T1200	80	91	88	89	26	55	100	99
RS232_T1300	89	86	100	89	26	65	100	99
RS232_T1400	83	100	98	61	22	15	100	99
RS232_T1500	83	82	95	73	24	47	99	99
RS232_T1600	89	97	93	62	26	28	99	99
s15850_T100	93	81	78	96	66	96	100	99
s35932_T200	100	67	**8.3**	93	59	88	100	99
s38417_T100	100	83	33	100	76	98	100	99
Average	83	86.5	74.3	79	39	54	99.6	99

other simulation-based methods[29,30]. An HT detection method based on functional analysis is proposed in Ref. [29], and an HT detection method based on signal correlation is proposed in Ref. [30]. Table 6 presents the results. The HTDet does not need the "golden circuit" since it is an unsupervised learning method, but it can obtain better HT detection performance in order to achieve a good trade-off between TPR and TNR.

In this study, we did not attempt to find all Trojan logics, but tried to find the set of most suspicious logics, which can effectively reduce the authentication time. In addition, a manual check after the automatic HT detection is always necessary.

Table 6 Comparison with the other methods: "-" indicates that the result is not clear.

(%)

Circuit group	TPR			TNP		
	Method in [29]	Method in [30]	HTDet	Method in [29]	Method in [30]	HTDet
RS232	-	-	72	92	-	99
s15850	-	61	96	99	99	99
s35932	-	27	93	99	99	99
s38417	-	100	100	99	99	99

6.4 Effectiveness analysis of test pattern generation method

We selected three typical circuit benchmarks (RS232_T1000, RS232_T1100, and s15850_T100) to evaluate the effectiveness of the proposed test pattern generation method. In these three benchmarks, the Trojan trigger was the combinational structure, the sequential structure, and the hybrid structure, respectively. Let the transition of each suspicious logic sw_i be tr_i during the simulation, where $\mathrm{sw}_i \in \mathrm{SW}$, and $i = 1, 2, \ldots, t$. Let $\mathrm{tr}_{\max}$ be the maximum of tr_i. Let $\mathrm{tr}_{\mathrm{ave}}$ be equal to $\sum_{i=1}^{t} \left(\mathrm{tr}_i / t\right)$. Then, the maximum transition and average transition are used to measure the effectiveness of test patterns. After obtaining SCPIs, we set all the primary inputs, which are not in the SCPIs, at logic 0 or logic 1. For the primary inputs in the SCPIs, we still generated full-random stimuli to perform simulation. After 10^6 cycles of simulation, the obtained transitions of suspicious Trojan logics are summarized in Table 7.

It can be seen that the proposed test pattern generation method could effectively increase the maximum

Table 7 Transitions comparison: "Before_∗" indicates full-random test stimuli and "After_∗" indicates constrained-random test stimuli using our approach.

Circuit	tr_{max}	tr_{ave}
Before_RS232_T1000	722	224.67
After_RS232_T1000	768	230.89
Before_RS232_T1100	719	224.39
After_RS232_T1100	746	231.56
Before_s15850_T100	716	64.19
After_s15850_T100	954	96.48

transition and average transition of these suspicious Trojan logics, which means that it can reduce the activation time.

7 Related Work

Hardware Trojan detection is a challenging problem. Lots of studies on HT detection have been conducted in the past decades, and they can be roughly classified into reverse engineering, side-channel analysis, GLIFT-based technique, static structure analysis, statistical feature analysis, and functional testing.

Bao et al.[10, 11] proposed that using reverse engineering to dissect the chip under detection can guarantee the detection of any malicious modifications in the chip. However, the cost of this method is too much, as it takes several weeks to analyze the chip under detection; hence, the reverse engineering method can only be applied to the ICs with small scale and simple structure.

In side-channel analysis[12–18], the impacts of HTs (e.g., circuit delay, transient current, leakage power, and heat analysis) are used to detect whether HTs are present in the CUD. However, the circuit is more susceptible to process variations and environmental noise due to the present nanoscale technologies.

GLIFT-based Trojan detection methods rely on gate-level information flow tracking to perform HT detection[19–21]. In GLIFT, each data bit is associated with a taint bit, and the data propagation is monitored by tracking the taint bits as they flow through Boolean gates. To track the propagation of taint bits, each standard cell gate is augmented with its corresponding tracking logic gate (referred to as GLIFT logic). However, the GLIFT logic generation is a difficult problem due to its inherent complexity. Moreover, GLIFT logic can produce false positive results[20].

A score-based classification method has been proposed for identifying HTs in CUD[22]. This technique comprehensively analyzes the characteristics of Trojan logics introduced at TrustHub[37], and then uses a strategy of conditional judgment for HT detection. Hasegawa et al. proposed learning structure features for Trojan detection[23–25]. For this purpose, support vector machine, multi-layer neural network, and random forest were applied to learn circuit structure features, seperately. In Ref. [26], the triggering characteristics of Trojan circuits are summarized, and a feature analysis technique based on flip-flop level information flow graph is proposed. Moreover, a multilevel HT detection framework, that combines flip-flop level and combinational logic level structure feature analysis, has been proposed[27]. Reference [28] analyzes the time to generate a transition in functional Trojans. The transition is modeled by geometric distribution, and the number of clock cycles required to generate a transition is estimated. FANCI[29] considers that between Trojan logic and normal logic, a significant difference exists in the input-to-output dependency; thus, it flags logics that have weak input-to-output dependency as suspicious Trojan logics by Boolean function analysis. In Ref. [30], an HT detection method using signal correlation is proposed. It basically estimates the statistical correlation between signals in a circuit for Trojan detection with the use of ordering points to identify the clustering structure algorithm. Furthermore, Ref. [31] proposed a reference-free HT detection scheme based on controllability and observability. This paper indicates that the characteristics of controllability and observability between Trojan gates and genuine gates have significant difference. In Ref. [32], an HT detection approach using natural language processing technology is proposed. It considers that design teams of commercial chips will have a specific design style due to the existence of established design specifications; therefore, the statistical method can be used to detect abnormal circuit logics.

Functional testing-based HT detection approaches[33–36] try to generate random test patterns to activate the HTs in CUD. If the logical values of primary outputs do not match the correct results, a Trojan is detected. The primary challenge of the functional testing-based method is that the Trojan circuit is much smaller than the original circuit, and HTs usually have a dormant nature. Hence, detecting potential HTs in CUD by traditional functional testing is difficult.

Different from the traditional functional verification

approaches, HTDet is a novel HT detection technique based on information entropy. We consider that the Trojan is usually inserted in the regions with low controllability and low observability in order to maintain high concealment, which will result in the Trojan logics featuring extremely low transitions during the simulation. Our approach does not require pushing the Trojan logic to the triggering state. As long as the information entropies of circuit logics are extremely low, the HTDet can flag them as suspicious Trojan logics based on the density-reachable relationship. Although the information theory has been applied in many fields, to the best of our knowledge, this is the first attempt to use the information theory technology to detect HTs in a hardware design.

8 Conclusion

In this paper, we propose a novel HT detection method called HTDet, which can effectively distinguish Trojan logics and normal logics using the information entropy technique. The HTDet is an unsupervised learning method and can quickly find suspicious Trojan logics without the "golden circuit". The HTDet does not need to trigger the Trojan logics during the simulation, and it flags circuit logics with extremely low information entropy as suspicious Trojan logics. In addition, we develop a heuristic method to increase the transitions of suspicious Trojan logics using mutual information. The experimental results demonstrate that the HTDet can obtain 79% average TPR and 99% average TNR, which is a better trade-off between TPR and TNR. In the future, we aim to study the automatic selection of parameters (r and MinPts) to achieve the optimal experimental performance. Moreover, we will study the method that can effectively detect newer HTs designs.

References

[1] L. Li, G. Ren, Y. Liu, and J. Wu, Secure DHCPv6 mechanism for DHCPv6 security and privacy protection, *Tsinghua Science and Technology*, vol. 23, no. 1, pp. 13–21, 2018.

[2] S. Liang, Y. Zhang, B. Li, X. Guo, C. Jia, and Z. Liu, Secureweb: Protecting sensitive information through the web browser extension with a security token, *Tsinghua Science and Technology*, vol. 23, no. 5, pp. 526–538, 2018.

[3] K. Fan, H. Li, W. Jiang, C. Xiao, and Y. Yang, Secure authentication protocol for mobile payment, *Tsinghua Science and Technology*, vol. 23, no. 5, pp. 610–620, 2018.

[4] X. Jin, Q. Wang, X. Li, X. Chen, and W. Wang, Cloud virtual machine lifecycle security framework based on trusted computing, *Tsinghua Science and Technology*, vol. 24, no. 5, pp. 520–534, 2019.

[5] Y. Wu, Y. Lyu, and Y. Shi, Cloud storage security assessment through equilibrium analysis, *Tsinghua Science and Technology*, vol. 24, no. 6, pp. 738–749, 2019.

[6] M. Tehranipoor, H Salmani, X. H. Zhang, R. Karri, J. Rajendran, and K. Rosenfeld, Trustworthy hardware: Trojan detection and design-for-trust challenges, *Computer*, vol. 44, no. 7, pp. 66–74, 2011.

[7] Q. Q. Wang, R. Geiger, and D. G. Chen, Hardware Trojans embedded in the dynamic operation of analog and mixed-signal circuits, presented at National Aerospace and Electronics Conference (NAECON), Dayton, OH, USA, 2015.

[8] B. Shakya, T. He, H. Salmani, D. Forte, S. Bhunia, and M. Tehranipoor, Benchmarking of hardware trojans and maliciously affected circuits, *Journal of Hardware and Systems Security*, vol. 1, no. 1, pp. 85–102, 2017.

[9] W. C. Li, Z. Wasson, and S. Seshia, Reverse engineering circuits using behavioral pattern mining, presented at the International Symposium on Hardware-Oriented Security and Trust (HOST), San Francisco, CA, USA, 2012.

[10] C. X. Bao, D. Forte, and A. Srivastava, On application of one-class SVM to reverse engineering-based hardware trojan detection, presented at the 15th International Symposium on Quality Electronic Design, Santa Clara, CA, USA, 2014.

[11] C. X. Bao, D. Forte, and A. Srivastava, On reverse engineering-based hardware trojan detection, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 1, pp. 49–57, 2016.

[12] R. Rad, X. X. Wang, M. Tehranipoor, and J. Plusquellic, Power supply signal calibration techniques for improving detection resolution to hardware trojans, in *Proc. of International Conference on Computer-Aided Design*, San Jose, CA, USA, 2008, pp. 632–639.

[13] S. Wei and M. Potkonjak, Scalable hardware trojan diagnosis, *IEEE Transactions on Very Large Scale Integration*, vol. 20, no. 6, pp. 1049–1057, 2012.

[14] J. Li and J. Lach, At-speed delay characterization for IC authentication and trojan horse detection, presented at the International Workshop on Hardware-Oriented Security and Trust, Anaheim, CA, USA, 2008.

[15] P. L. Song, F. Stellari, D. Pfeiffer, J. Culp, A. Bonnoit, B. Wisnieff, and M. Taubenblatt, MARVEL-malicious alteration recognition and verification by emission of light, presented at the International Symposium on Hardware-Oriented Security and Trust, San Diego, CA, USA, 2011.

[16] K. Q. Hu, A. Nowroz, S. Reda, and F. Koushanfar, High-sensitivity hardware trojan detection using multimodal characterization, in *Proceedings of the Conference on Design, Automation and Test in Europe*, Grenoble, France, 2013, pp. 1271–1276.

[17] A. Nowroz, K. Q. Hu, F. Koushanfar, and S. Reda, Novel techniques for high-sensitivity hardware trojan detection using thermal and power maps, *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 12, pp. 1792–1805, 2014.

[18] S. Narasimhan, D. Du, R. Chakraborty, S. Paul, F. Wolff,

C. Papachristou, K. Roy, and S. Bhunia, Hardware trojan detection by multiple-parameter side-channel analysis, *Transactions on Computers*, vol. 62, no. 11, pp. 2183–2195, 2013.

[19] W. Hu, B. Mao, J. Oberg, and R. Kastner, Detecting hardware trojans with gate-level information-flow tracking, *Computer*, vol. 49, no. 8, pp. 44–52, 2016.

[20] W. Hu, D. Mu, J. Oberg, B. Mao, M. Tiwari, T. Sherwood, and R. Kastner, Gate-level information flow tracking for security lattices, *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 20, no. 1, p. 2, 2014.

[21] W. Hu, J. Oberg, A. Irturk, M. Tiwari, T. Sherwood, D. Mu, and R. Kastner, Theoretical fundamentals of gate level information flow tracking, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 8, pp. 1128–1140, 2011.

[22] M. Oya, Y. H. Shi, M. Yanagisawa, and N. Togawa, A score-based classification method for identifying hardware-trojans at gate-level netlists, in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, Grenoble, France, 2015, pp. 465–470.

[23] K. Hasegawa, M. Oya, M. Yanagisawa, and N. Togawa, Hardware trojans classification for gate-level netlists based on machine learning, presented at the 22th International Symposium on On-Line Testing and Robust System Design (IOLTS), Sant Feliu de Guixols, Spain, 2016.

[24] K. Hasegawa, M. Yanagisawa, and N. Togawa, Hardware trojans classification for gate-level netlists using multi-layer neural networks, presented at the 23th International Symposium on On-Line Testing and Robust System Design (IOLTS), Thessaloniki, Greece, 2017.

[25] K. Hasegawa, M. Yanagisawa, and N. Togawa, Trojan-feature extraction at gate-level netlists and its application to hardware-trojan detection using random forest classifier, presented at the International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 2017.

[26] S. Yao, X. M. Chen, J. Zhang, Q. Y. Liu, J. Wang, Q. Xu, Y. Wang, and H. Z. Yang, Fastrust: Feature analysis for third-party IP trust verification, presented at the International Test Conference (ITC), Anaheim, CA, USA, 2015.

[27] X. M. Chen, Q. Y. Liu, S. Yao, J. Wang, Q. Xu, Y. Wang, Y. P. Liu, and H. Z. Yang, Hardware trojan detection in third-party digital intellectual property cores by multilevel feature analysis, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 7, pp. 1370–1383, 2018.

[28] H. Salmani, M. Tehranipoor, and J. Plusquellic, A novel technique for improving hardware trojan detection and reducing trojan activation time, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 1, pp. 112–125, 2012.

[29] A. Waksman, M. Suozzo, and S. Sethumadhavan, FANCI: Identification of stealthy malicious logic using boolean functional analysis, in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, Berlin, Germany, 2013, pp. 697–708.

[30] B. Cakir and S. Malik, Hardware trojan detection for gate-level ICs using signal correlation-based clustering, in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, Grenoble, France, 2015, pp. 471–476.

[31] H. Salmani, COTD: Reference-free hardware trojan detection and recovery based on controllability and observability in gate-level netlist, *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 2, pp. 338–350, 2017.

[32] H. H. Shen, H. Z. Tan, H. W. Li, F. Zhang, and X. W. Li, LMDet: A "Naturalness" statistical method for hardware trojan detection, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 4, pp. 720–732, 2018.

[33] R. Chakraborty, S. Paul, and S. Bhunia, On-demand transparency for improving hardware trojan detectability, presented at the IEEE International Workshop on Hardware-Oriented Security and Trust, Anaheim, CA, USA, 2008.

[34] F. Wolff, C. Papachristou, S. Bhunia, and R. Chakraborty, Towards trojan-free trusted ICs: Problem analysis and detection scheme, in *Proceedings of the Conference on Design, Automation and Test in Europe*, Anaheim, CA, USA, 2008, pp. 1362–1365.

[35] W. T. Cheng, M. Sharma, T. Rinderknecht, L. Y. Lai, and C. Hill, Signature-based diagnosis for logic BIST, presented at the IEEE International Test Conference, Santa Clara, CA, USA, 2006.

[36] G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassab, A. Hassan, and J. Rajski, Logic BIST for large industrial designs: Real issues and case studies, in *Proceedings of International Test Conference*, Atlantic, NJ, USA, 1999, pp. 358–367.

[37] H. Salmani, M. Tehranipoor, and R. Karri, On design vulnerability analysis and trust benchmarks development, presented at the 31st International Conference on Computer Design (ICCD), Asheville, NC, USA, 2013.

[38] H. Akaike, Information theory and an extension of the maximum likelihood principle, *Springer*, pp. 199–213, 1998.

[39] L. H. Goldstein and E. L. Thigpen, SCOAP: Sandia controllability/observability analysis program, presented at the 17th Design Automation Conference, Minneapolis, MN, USA, 1980.

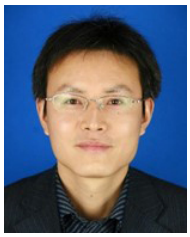
[40] W. T. Wang, Y. L. Wu, C. Y. Tang, and M. Hor, Adaptive Density-Based Spatial Clustering of Applications with Noise (DBSCAN) according to data, presented at the International Conference on Machine Learning and Cybernetics (ICMLC), Guangdong, China, 2015.

[41] H. C. Peng, F. H. Long, and C. Ding, Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[42] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, and et al. API design for machine learning software: Experiences from the scikit-learn project, presented at the ECML PKDD Workshop: Languages for Data Mining and Machine Learning, Prague, Czech Republic, 2013.

Renjie Lu is currently working toward the MS degree at the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China. His current research interests include hardware security, natural language processing, and machine learning.

Haihua Shen received the PhD degree from Tsinghua University, Beijing, China, in 2002. She is a professor at the University of Chinese Academy of Sciences, Beijing, China. She had been a researcher with the Institute of Computing Technology, Chinese Academy of Sciences from 2002 to 2013. She also served as the deputy chief engineer of Loongson Inc. from 2008 to 2013, fully responsible for chip verification. Her research interests include computer architecture, design verification, hardware security, design for reliability, etc. She has published over 50 technical papers and holds over 20 patents in the above areas.



Zhihua Feng received the MS and PhD degrees from Northwestern Polytechnical University, China, in 2007 and 2015, respecitively. Since 2007, he has been a professor with Beijing Institute of Computer Technology and Application, China. His research interests include reconfigurable computing technology, hardware IP security and trust, machine learning, and SSD controller.

Huawei Li received the BS degree from Xiangtan University, Xiangtan, China, in 1996, and the MS and PhD degrees from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 1999 and 2001, respectively. She has been a professor at ICT, CAS, since 2008. She was a visiting professor at the University of California, Santa Barbara, CA, USA, from 2009 to 2010. Her current research interests include testing of Very Large-Scale Integration (VLSI)/SoC circuits, design verification, design for reliability, and error tolerant computing. She has authored over 120 technical papers and holds 20 Chinese patents in the above areas.

Wei Zhao is currently working toward the MS degree at the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China. Her current research interests include hardware security, software engineering, and machine learning.

Xiaowei Li received the BEng and MEng degrees from the Hefei University of Technology, Hefei, China, in 1985 and 1988, respectively, and the PhD degree from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 1991. He was an associate professor at the Department of Computer Science and Technology, Peking University, Beijing, China, from 1991 to 2000. In 2000, he joined ICT, CAS, as a professor, where he is currently the deputy director of the State Key Laboratory of Computer Architecture. He has authored or coauthored over 300 papers in journals and international conferences, and holds 50 patents and 30 software copyrights. His current research interests include VLSI testing, design for testability, design verification, dependable computing, and wireless sensor networks.