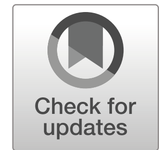
Journal of Hardware and Systems Security (2019) 3:426–439
https://doi.org/10.1007/s41635-019-00081-x



Implementation-Based Design Fingerprinting for Robust IC Fraud Detection

James Shey[1,2] · Naghmeh Karimi[1] · Ryan Robucci[1] · Chintan Patel[1]

Received: 20 February 2019 / Accepted: 2 October 2019 / Published online: 13 November 2019
© Springer Nature Switzerland AG 2019

Abstract
With the global spanning of integrated circuit (IC) and electronic device supply chains, the ability of an untrusted foundry to alter a design for intellectual property (IP)/IC piracy increases. To tackle this threat, this paper proposes a design-based fingerprinting methodology based on machine learning schemes. The proposed method considers the effect of process variations, measurement noise, and device aging. The proposed fingerprinting scheme can identify if the circuit is original or has been altered by an adversary to hide piracy. Changing a gate to an equivalent counterpart does not change the functionality of the circuit and we assess these as altered circuits. Altering the circuit can arise if an adversary gains access to the register-transfer level (RTL) or netlist of a design in an untrusted supply chain or uses the datasheet to implement a functionally equivalent design. Experimentally we determine that our method can detect a pirated chip with near 100% accuracy when classifying new ICs, and above 96% accuracy when classifying at any age up to 7 years in the presence of noise if at least 2.5% of the gates in the IC have been altered by an adversary.

Keywords IC aging · IP piracy · Fingerprinting · Side-channel power analysis · Machine learning

1 Introduction

Both intellectual properties (IPs) and the related integrated circuits (ICs) have been threatened by fraud in recent years [1]. To this end, there is an urgent need to prevent such piracies and ensure that an IC design has not been altered through the supply chain. Common piracy avoidance methods such as active metering schemes require additional hardware and/or impose delay as well as area overhead [2–4]. To address this problem, this paper proposes a technique using side-channel analysis to monitor the transient power consumption of an IC in order to detect if the IC has been altered during the manufacturing process. Figure 1a depicts the high-level view of the proposed scheme. Additionally, this method can be used by a field maintenance technician to verify that the ICs contained within their device are authentic.

As shown in Fig. 1a, this paper proposes an authenticity certification for a manufacturer to ensure that they are only using the low-cost IC designs from a trusted designer in non-vertical supply chains with untrusted foundries and 3rd party suppliers. Note that this paper only deals with the trustworthiness of foundries and 3rd party suppliers, and we consider the designer is trustful. This is accomplished through validation of ICs at a trusted test facility using our method to distinguish trusted ICs from pirated versions. Additionally, this method can be used to validate the ICs in the field by a field maintenance technician even after they have been used for a while (aged ICs) and to ensure that recovered devices have not been altered. The test facility will use the provided fingerprint to distinguish original designs from pirated versions through a precise tolerance on the fingerprint (see Fig. 1b). This method is aimed at

✉ James Shey
jshey1@umbc.edu; shey@usna.edu

Naghmeh Karimi
nkarimi@umbc.edu

Ryan Robucci
robucci@umbc.edu

Chintan Patel
cpatel2@umbc.edu

1 Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore, MD, USA

2 Electrical and Computer Engineering Department, U.S. Naval Academy, Annapolis, MD, USA

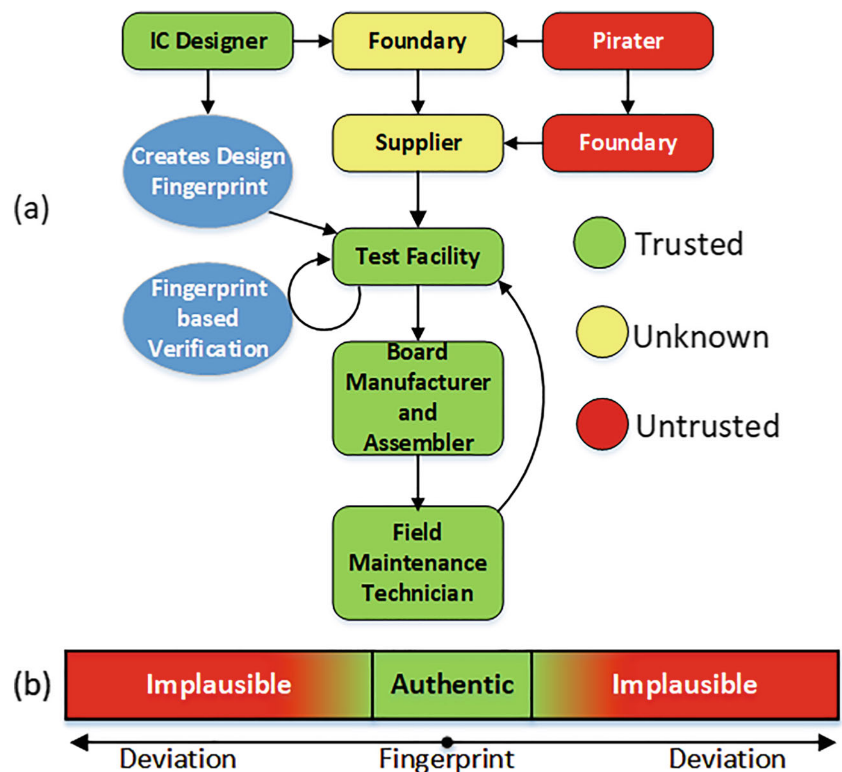


Fig. 1 **a** Supply chain allowing a board manufacturer/assembler to verify components received from a foundry via a trusted/untrusted supplier. Our method provides the design-based fingerprints that the trusted test facility will utilize to detect an altered IC from an original IC. Piracy detection can also be initiated by a field maintenance technician who can send an IC recovered from a device in the field to a test facility for identification. **b** Continuum showing likeliness measures of an IC compared with a given fingerprint used for authenticity certification. Green shows a match (original IC) and red depicts a mismatch (altered IC). (modified from [5])

low-cost ICs which have a cost constraint. Implementation of modern protection mechanisms may be cost prohibitive.

Our threat model covers two possible attack vectors: reverse engineering and resynthesis of register-transfer level (RTL). In the first case, the adversary has access to the product datasheets and uses reverse engineering to create a functional implementation of the original design. Creating the same implementation from a datasheet is unrealistic. In the second attack, the adversary gains access to the RTL design and then uses logic synthesis to create a new netlist. An adversary may gain access to the RTL when the design is passed from a design group to a synthesis and verification group either within a company or between companies [6]. Without knowing the parameters used for the original synthesis, it is difficult to recreate the netlist of the original circuit. In both cases, the adversary can use the original or another foundry to implement the new implementations and introduce them into the supply chain.

The objective of the implementation-based design fingerprint method is to increase the cost for the attacker while not adding design cost or sacrificing performance. The aim is to significantly raise the complexity for an attacker who intercepts a functional design, has access to a full functional description, or performs reverse engineering, and then wishes to reengineer an alternate physical implementation (with the same functionality). In fact, the attacker is assumed to have full access to the functional description. Using an implementation-based fingerprint also addresses the case where protecting against design theft is not practical. For example, a cycle-accurate description of a general purpose processor may be freely available in documentation. It would not require unreasonable effort for an attacker to redesign a functional equivalent version, though perhaps with suboptimal reliability or quality assurance. The outlined scheme provides a method to specifically identify a particular physical design, but does not protect against replicating its functionality. The other methods described in the “Related Work” section (Section 2) can be complementary to this method.

Our proposed scheme aims to thwart such IP piracy. The main contributions of this paper include the following:

1. A design-based fingerprinting scheme that can detect IP piracy;
2. Demonstrating the aging, noise, and process variation resiliency of the proposed fingerprinting scheme;
3. Extensive simulation-based results demonstrating the accurate differentiation of original and altered circuits.

In practice, the power-based signature extracted from a device after manufacturing can be different from the design-based signature due to process variations, noise, aging effects, or due to piracy-imposed netlist changes. Being able to distinguish a change in the power signature due to a netlist variation from the other three is desired. As the former demonstrates IP piracy, the other cases are benign. Classifying the difference between an aged IC and an altered IC

is a difficult problem because the current consumption is a function of both the underlying architecture and the age of the IC. There is a need for a classifier that ignores the aging, process variation, and noise-induced variations yet takes the circuit gate-level netlist changes into account.

The experimental results show with 96% accuracy that we can detect IP piracy if more than 2.5% of the gates have been changed from the original netlist by an adversary. Such detection is regardless of the device age, process variations, and the noise. Figure 1b shows the spectrum of an IC regrading its fingerprint spanning between original IC (authentic) and altered IC (implausible).

The remainder of this paper is organized as follows: Section 2 presents related work in the field, Section 3 covers background material relevant to the paper, Section 4 method used to create the design-based fingerprint, Section 5 covers the experimental setup used to validate the fingerprint, Section 6 discusses the results of the experiments, Section 7 discusses circuit partitioning and its applicability to this work, and Section 8 presents the conclusion and avenues for future work.

2 Related Work

To protect an IP against piracy, a designer can add some elements to the circuitry that normally is not needed. Such circuitry can be used to uniquely identify the design [3]. Common IP and IC protection schemes are as follows: watermarking, obfuscation, device fingerprinting, and physical unclonable function (PUF) based. The majority of prior work in the fields of IP and IC protection require additional circuity [7, 8]. IP protection can be divided into Soft and Hard-IP cores. Soft-IP is often given as synthesizable register-transfer level (RTL) or netlists. The underlying design of Soft-IP can be altered, whereas Hard-IP cores are given as layout files such as GDSII files and are difficult to alter [9]. This work also has some ties to Trojans and recycled ICs.

2.1 Watermarking

Watermarking is the insertion of markings known only to the designer and can be inserted at multiple times during the design process. Such markings can be used to uniquely identify the IP contained within an IC and as such watermarking is a detection method for IP theft [3]. The markings can cover a wide range of possibilities, from the way a Verilog file is implemented to creating a particular transistor layout pattern and can be divided into Soft-IP and Hard-IP protection [10]. Implementation of Soft-IP watermarking can be done by encoding data into the unused transitions in a finite-state machine (FSM) [10], merging FSM states [11], encoding the watermark in Look-Up Tables (LUTs) [12], and verification of FSM watermark through power analysis [4]. Hard-IP protection techniques include encoding data into the gate lengths and widths [9] and inserting the watermark into the scan chain [13].

The full extent of the performance impact of watermarking using FSMs is difficult to predict [13]. Additionally, IP watermarking requires design time and therefore cost to implement on any level, whereas our approach can provide a level of protection without the design overhead for watermarking.

2.2 Obfuscation

Obfuscation refers to the insertion of additional elements to the circuit such that by using the correct key, the device will operate correctly, but without the key, the exact functionality cannot be determined [14]. This can be accomplished with combinational circuits that do not produce the correct output and with sequential circuits that do not transition to the correct state in an FSM [15]. Obfuscation can be broken into static and dynamic [16]. Static obfuscation requires one key to operate correctly whereas dynamic, the key changes with time. Of the two cases, static is easier to overcome [16].

In obfuscation, additional circuitry is needed as well as design time to implement obfuscation. An additional problem with obfuscation is once the key is known or is released, all of the designs that used that key are no longer protected. This key can be learned from an uncontrolled source or the key can be broken by brute force like encryption [16]. Obfuscation is a deterrent and can potentially be used in conjunction with our method.

2.3 Device Fingerprinting

Device fingerprinting, a form of hardware metering, refers to the act of ensuring that each IC has a unique identifier and each IC can be tracked through the supply chain by its identifier [2]. Recent work in this area deals with leakage and switching power of the gates to create a device fingerprint [17]. Each device has a unique signature and therefore needs to be handled individually after manufacturing. They must be tracked through the production cycle [18], whereas our method creates one signature for a design based on simulations. It is therefore less costly to implement and managing one signature for the design than a signature for each individual IC.

2.4 PUF-Based IP Protection

The introduction of PUFs into designs can be used to protect the underlying design. The addition of PUFs, their inherent uniqueness, adds the ability to individually identify ICs.

These PUFs can be combined with other techniques to create a more robust protection scheme.

HARPOON combines PUFs and obfuscation to create an IC which will only operate in the normal mode if given the correct key. The key is determined by the PUF [19]. This improves the traditional obfuscation because each key must be derived by the developer based on the output of the PUF and is therefore unique to the IC.

Combining reconfigurable LUTs, PUFs, and FSMs, a designer can make a pay-per-use IP protection scheme in a field-programmable gate array (FPGA) [20]. This uses a PUF as an input to an FSM. The FSM also requires an input developed by the designer based on the PUF output. The developer's input is stored in a LUT for reuse. This method allows for an IP owner to control the use of their IP on a particular FPGA.

With the introduction of PUFs into the designs, and their inherent differences between each implementation, PUF-based IP protection is device specific so each device needs to be tracked individually. This differs from our approach where the individual IC does not need to be tracked, but the design can be verified on the IC. Additionally, our technique requires less interaction from the designer and is therefore less costly.

2.5 Trojans

The majority of prior work in the fields of IP and IC protection require additional circuitry [7, 8]. Another set of detection methods exist for detecting hardware Trojans; Trojans are additional elements added to a circuit that includes a trigger and payload to produce a desired effect [1, 21, 22].

Altering the final circuit during the manufacturing process is very similar to Trojan insertion. Hardware Trojans are elements that are maliciously inserted in circuits and include trigger and payload parts. The trigger can be a specific input vector, or a sequence of input values that activate the Trojan circuit. When activated, the payload can result in circuit malfunction or data leakage [1, 23]. Trojan detection methods mainly rely on characterizing path delays [23, 24], and/or electromagnetic emissions [22]. However, in our threat model, the malicious change does not alter the functionally of a victim circuit, nor does it result in data leakage.

Using path-delay-based Trojan detection schemes to detect our victim circuits is costly because of the scalability issue of these methods with the exponentiation growth of the number of paths needed to be monitored. In practice, for our threat model, the adversary does not need to target the non-critical or near-critical paths for gate changes. When Trojans are placed on non-critical paths, path-delay-based techniques encounter scalability issues as they need to monitor several paths. In addition, although our gate change can be labeled as a Trojan, Trojan-detection schemes that rely on functionality change, fail to detect our pirated ICs as functionality is not changed, based on our threat model. On the other hand, we believe that our technique can be adapted for detecting Trojans as well. However, further investigation is needed to confirm this ability and therefore in this paper, we focus on distinguishing the ICs which follow our threat model and leave the Trojan detection problem as a future research.

2.6 Recycled ICs

Some researchers focus on detecting recycled ICs, i.e., the used ICs that were sold as new ones [25–27]. In particular, [25] is similar to our method in investigating aging effects, but the authors' main focus is to detect aged ICs, whereas ours is to create a classifier that can identify an IC, new or aged, from an altered IC. Additionally, this paper is limited by the use of a one-class classifier with only good examples. The accuracy is affected by process variation, whereas our approach can overcome process variation and includes training cases of known altered circuits to reduce the likelihood of overfitting the data set.

3 Background

3.1 Aging

To fully understand how the age of an IC will affect its performance, one must understand the major aging phenomenons and how these affect the operation and characteristics of the transistors. IC designers consider the aging-induced performance shift of transistors embedded in an IC during design such that the device will perform properly throughout its normal lifetime. The dominant phenomenons that degrade transistor performance over its life are bias temperature instability (BTI) and hot carrier injection (HCI), but the full aging process of an IC is a complex process with multiple variables [28].

3.1.1 Bias Temperature Instability

BTI plays a major role in IC aging. It is due to a buildup of interface traps and oxide trapped charge [29]. BTI can be further broken down into negative bias temperature instability (NBTI) and positive bias temperature instability (PBTI).

NBTI and PBTI occur in p-channel metal-oxide-semiconductor field-effect (PMOS) transistors and n-channel metal-oxide-semiconductor field-effect (NMOS) transistors, respectively. In practice, the impact of NBTI is

more dominant than PBTI beyond 45 nm technology nodes. NBTI occurs in a PMOS transistor when a negative voltage is applied to its gate. In this mechanism, positive interface traps are generated at the Si/SiO_2 interface. As a result, the threshold voltage increases and the PMOS transistor becomes slower and fails to meet timing constraints. In contrast to NBTI, PBTI occurs when a positive voltage is applied to the gate of an NMOS transistor. This eventually increases the transistor threshold-voltage [30–32]. Both NBTI and PBTI are recovered partially when the transistor is off because the dielectric traps can be emptied. This relies on a removal of the stressing voltage and a corresponding delay time. Note that the magnitude of stress and recovery both depend on the temperature, voltage source, and the circuit workload. Thereby, similar ICs aged with different input vectors will exhibit different aging rates.

3.1.2 Hot Carrier Injection

HCI is an aging mechanism where an electron from the channel has enough energy to tunnel through the gate oxide or substrate and create an increased gate or substrate leakage current, respectively [28, 33]. This process can result in collisions between the high energy electrons and crystalline structure of the device, which causes permanent damage if the collisions occur in the gate oxide [28, 33].

For a transistor subjected to HCI, there is a chance that the electron becomes trapped in the gate's dielectric which acts to raise the threshold voltage. This change degrades the current carrying capacity for a given gate voltage and reduces the switching frequency [28]. HCI is more dominant in NMOS transistors and occurs when a transition (rise or fall) occurs in the gate input of the transistor. Effects caused by HCI are, unlike BTI, irreversible and compounds over the life of the device.

3.2 Noise

Noise is ever present in a circuit and can affect the performance of a classifier by adding randomness to a circuit. The dominant sources are thermal and flicker noise and are described below in more detail.

3.2.1 Thermal Noise

Thermal noise is due to the random thermal motion of charge carriers [34]. It is present in all electrical circuits and is often modeled with a Gaussian distribution.

3.2.2 Flicker Noise

Flicker noise, also called 1/f noise and pink noise, is present at all frequencies, but is dominant at low frequencies. Two theories of the causes of Flicker Noise are the carrier number fluctuation theory and the mobility fluctuation theory [35]. The carrier number fluctuation theory is caused by the random trapping and escaping of charges in the oxide traps near the Si/SiO_2 interface. Charges are moved through tunneling from the channel to the trap location. While mobility fluctuation theory is based on fluctuations in the bulk mobility induced by fluctuations in phonon population through phonon scattering [35].

4 Proposed IC Identification Methodology

The method presented here creates a design-based fingerprint through simulation of a given circuit. To extract the aimed fingerprints, we first simulate the circuit with a known set of input vectors. Then, the fast Fourier transformation (FFT) is taken of these current traces and a support vector machine (SVM) classifier uses this data to create a classifier to distinguish original and altered ICs. This classifier or fingerprint is then saved to be recalled when a circuit is needed to be identified. This method is shown in Fig. 2 and is based on our prior work [5], which presents a design-based fingerprinting scheme using a side-channel analysis scheme. This technique is used to determine if an IC has been altered from an original design. In our previous work [5], we showed that with almost 100% accuracy, we can identify the altered ICs from the original one in the presence of process variations. In this paper, we extend our previous work and consider the effect of aging and measurement noise as well and show how those phenomena would affect the classification of an IC as original or altered.

5 Experimental Setup

5.1 Simulation Setup

The work presented here is simulation based, and we used four ISCAS benchmark circuits [36] shown in Table 1. An in-house tool was deployed to generate the transistor-level model of the considered benchmarks, using a 45-nm technology extracted from the open-source NANGATE library [37]. The aging simulations were performed with MOS Reliability Analysis (MOSRA) tool from Synopsys' HSPICE in two steps [38]: the first step, the drift of transistors' electrical parameters under both BTI and HCI aging was evaluated. The second step uses the degradation of the transistor-level device characteristics computed in step one, to generate the current traces used for our proposed fingerprints. MOSRA simulates the effects of both BTI and HCI and therefore the first simulation is input dependent because of the recovery process present with BTI aging.

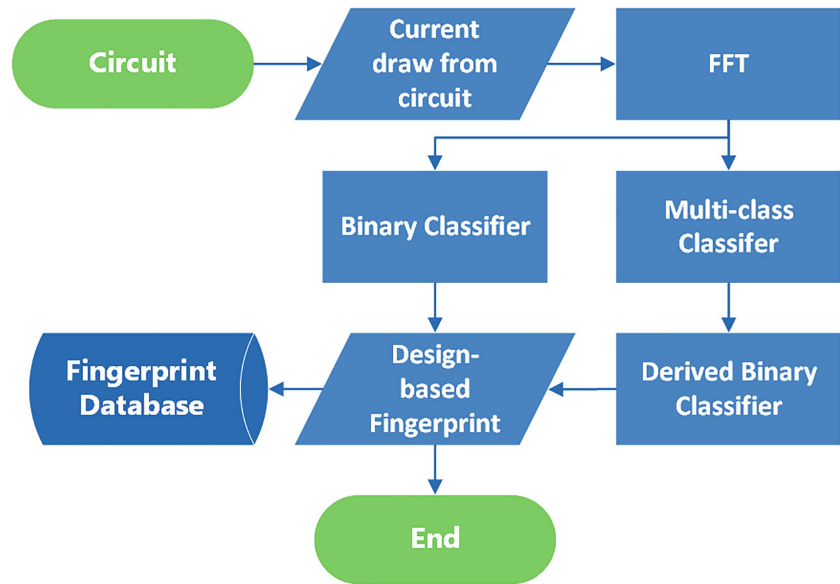


Fig. 2 Flowchart demonstrating the extraction of design-based fingerprints from current traces using binary and multi-class classifiers. A FFT is performed on current traces and the results passed to multiple classifiers, which combine to create the design-based fingerprint

For this paper, the circuits were aged at 1-year increments to an age of 7 years. Note that our goal is identifying a circuit even if it has been aged to an unknown age with an unknown input. Therefore, for the first step of MOSRA simulations, we use randomly generated inputs while for the second step of MOSRA simulations, i.e., the step in which current measurements are performed, we use known inputs. The noise simulations were performed using Synopsys' transient noise analysis in HSPICE and we used the default HSPICE parameters for noise sources.

Simulations were carried out using the following process variation parameters with a Gaussian distribution: transistor gate length L, $3\sigma = 10\%$; threshold voltage V_{TH}, $3\sigma = 30\%$, and gate-oxide thickness t_{OX}, $3\sigma = 3\%$. This process variation reflects a 45-nm process in commercial use today [39]. These process variations are on the high end of typical range, for instance, L varies from 5 to 10% [31, 39–41]. The simulations were conducted assuming 45 °C operating temperatures and 10-ps temporal resolution.

For each simulation, 100 input vectors are fed to each circuit in succession and the current draw of the circuit is monitored. Prior to being fed into the circuit, the input vectors for each circuit are first run through four inverters to model a driver circuit that has limited slew rate. The inverters are not aged, do not have process variation, and have a different power source than the tested benchmark. This ensures that the current drawn by the benchmark circuit will not be corrupted by the current drawn by the driving circuit and that the simulated input to the IC would be closer to a real-world input that has a limited slew rate. The FFT of current consumption is taken and is the input to the classifier. Using the FFT yields the frequency representation of current and is used because it yielded higher accuracies than using the time-domain [5].

To create the negative inputs for the classifier, altered circuits were created. An altered circuit has the same functionality of the original and is created by selecting a random gate(s), and replacing it(them) with an inverted gate followed by an inverter (e.g., an AND gate becomes a NAND gate followed by a NOT gate, a NOR gate becomes an OR gate followed by a NOT gate). For each altered circuits, small numbers of gates were changed that ranged from one to thirty gates depending on the size of the circuit. For c432, c499, and c1355, 1, 2, 5, and 10 gates were altered and for c5315, 1, 2, 5, 10, 15, and 30 gates were altered. The overall number of changed gates overall represents a small percentage of the overall circuit as shown in Table 2. For

Table 1 ISCAS'85 circuit descriptions

Benchmark	Number of gates	Function
c432	215	27-Channel
c499	245	32-Bit single-error-correcting circuit
c1355	589	32-Bit single-error-correcting circuit
c5315	2972	9-Bit ALU

Table 2 Percent change for ISCAS'85 circuit for given number of gate changes (no. of gate changes/total no. of gates)

	Number of gate changes				
Benchmark	1	2	5	10	15
c432	0.47%	0.93%	2.33%	4.65%	6.98%
c499	0.41%	0.82%	2.04%	4.08%	6.15%
c1355	0.17%	0.34%	0.85%	1.70%	2.55%
c5315	0.03%	0.07%	0.17%	0.34%	0.51%

this paper, only changes below 5% were used to create the classifiers. Using these, small changes were done because larger more pronounced alterations affect the current more than a smaller number of changes and, for this paper, the more difficult task of detecting a small number of changes was chosen [5] and the larger percentages of gate changes were not used.

5.2 Classification Setup

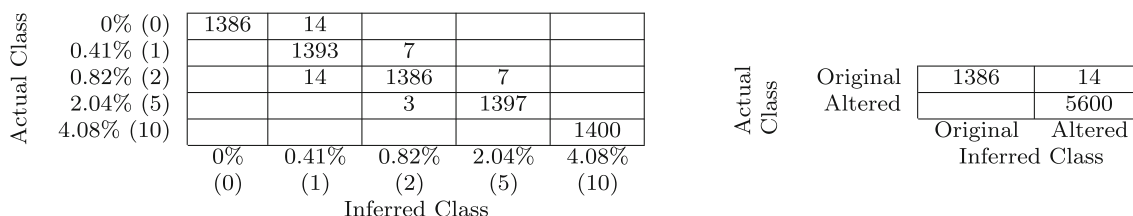
An SVM is chosen for classification as it provided the highest accuracy when compared with other machine learning techniques. A similar scheme was used in prior work [5, 25, 42]. This paper uses a SVM with a kernel with polynomial order of two. Additionally, 5-fold cross-validation is used to limit overfitting. A binary SVM with a kernel maps the data onto a higher dimensional feature space and then attempts to find a boundary-defining expression that groups the classification together and minimizes errors [43]. A multi-class classifier is similar to the binary classifier but defines multiple boundaries to separate the various classes and maximize their separation [44]. For this work, the multi-class classifier is derived from the number of changes that are made in the circuit and uses a 1-vs-1 approach. A derived binary classifier takes a multi-class classifier and reduces it down to original circuit and altered circuit labels, where an altered circuit refers to any circuit containing an altered layout from the original benchmark. This is shown in Fig. 3, with the multi-class classifier on the left and the reduced binary classifier on the right.

In the multi-class confusion matrix shown in Fig. 3, the columns show the number of gates changed in our experiments. Zero refers to no-change (original circuit) and 1, 2, 5, and 10 refer to the altered circuit which is different from the original circuit in 1, 2, 5, or 10 gates. The numbers in the parenthesis depict the percentage of gates changed over the total number of gates in each benchmark circuit. In a confusion matrix, the rows are the ground truth, while the columns represent the outcome of the classifier. In Fig. 3, the first row ground truth is 0 meaning an original circuit and out of 1400, the classifier predicted that 1386 were original while 14 were classified as altered. To create the reduced binary confusion matrix, the multi-class confusion matrix is broken down into four regions. The first is the original circuit that is classified as original, which is 1386 in this example. The second is the original circuit classified as altered (classified as either 1, 2, 5, or 10 changes), which in this case is 14. The third region is altered circuits that were identified as original. This would be the sum of the entries that are in the first column, but not in the first row. The final region is altered circuits that are classified as altered. In this case, we do not care a 1 gate change is classified as a 1 gate change, we only care that it is not classified as a 0 gate change (original circuit). In our case that is 1393+7+14+1386+7+3+1397+1400=5600. The resultant binary confusion matrix is shown on the right side of Fig. 3.

5.3 Classifier Creation and Testing of the Classifier

The FFT of the current traces of 100 Monte Carlo simulations of the original circuit including process variation, are aged to 7 years in 1-year steps, are used to create the true cases of the training data set. The altered cases of the training data set are composed of circuits with altered gates as discussed above. For each of these altered circuits, we performed 10 Monte Carlo simulations to take process variations into account and each circuit was aged for 7 years with aging steps of 1 year to create the false cases for the training data set. The current traces from the true and false cases were fed into the SVM classifier with a multi-class outcome and then reduced to a binary outcome. For the cases where a threshold was set for the detection of changes, the classifier disregarded the false cases that were below the threshold level.

This work looks specifically at process variation, noise, and aging and their effects on being able to detect a fraudulent IC. To test the created classifier, four test cases were created for each IC. Each test case is composed of 100 Monte Carlo simulations of the original unaltered circuit and five of each altered case (different number of changed gates) each with ten Monte Carlo simulations. The first classifier test case is created with process variation without the effects of aging and noise and will be referred as P. The second classifier test case is created with aging and process



Actual Class \ Inferred Class	0% (0)	0.41% (1)	0.82% (2)	2.04% (5)	4.08% (10)
0% (0)	1386	14			
0.41% (1)		1393	7		
0.82% (2)		14	1386	7	
2.04% (5)			3	1397	
4.08% (10)					1400

Actual Class \ Inferred Class	Original	Altered
Original	1386	14
Altered		5600

Fig. 3 Multi-class confusion matrix (left) and resultant reduced binary confusion matrix (right). Confusion matrices are for the c499 benchmark detecting all changes in a test data set

variation and is referred to as *PA*. The third test set uses process variation and noise to examine the effects of noise on the classifier referred to as *PN*. The last classifier test evaluates the performance of the classifier with respect to process variation, aging, and noise. For this case, the circuits are aged to 7 years and are subject both to process variation and noise referred to as *PAN*. For the cases with aging (*PA* and *PAN*), two data sets were created, each data set contained circuits that were aged with a unique input vectors for the first step of the MOSRA simulation. Otherwise, the data sets are the same. This is done to simulate devices that are recovered from the field and the inputs they received while they aged are unknown. This ensures that the classifier is not overfitting that data from just one aging process and to show the input vectors while a circuit age is an independent variable.

Accuracy, sensitivity, precision, and F1 score are used for performance evaluation of a classifier and are defined by Eqs. 1, 2, 3, and 4 respectively. For this work, we use accuracy, sensitivity, and F1 score to compare performance.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\#\,\text{Positives} + \#\,\text{Negatives}} \quad (1)$$

$$\text{Sensitivity} = \frac{\text{True Positives}}{\#\,\text{Positives}} \quad (2)$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (3)$$

$$F1 = \frac{2 * \text{Sensitivity} * \text{Precision}}{\text{Sensitivity} + \text{Precision}} \quad (4)$$

These allow direct comparison of the results of the method using different circuits and different minimum change thresholds.

6 Experimental Results and Analysis

6.1 Verifying Resynthesized Circuits Change

Our work relies on the fact that a circuit that is synthesized with different settings will have different implementations. Reverse engineering a design based on functionality does not get the metadata of the design such as constraints on the synthesis. To demonstrate this, we resynthesized the ISCAS'85 c499 circuit that includes 245 gates. With 245 gates the basis of our comparison, the circuit was resynthesized with different parameters. Resysnthesizing with a minimum size constrained resulted in a circuit with 164 gates (33% change). This is shown in Fig. 4 with a minimum number of gates constraint and a maximum fan-out of 2 constraint. This highlights that without knowing all the parameters for synthesis, the circuit can be drastically different and therefore detectable.

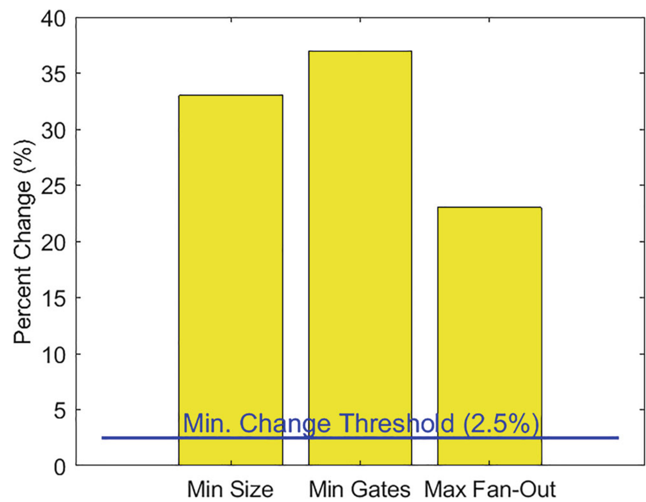


Fig. 4 Percentage change for c499 circuit with the following constraints: Minimum size constraint, Minimum number of gates constraint, and Maximum fan-out of 2 constraint. These are shown in contrast to the minimum change threshold for detection that our technique uses (Section 6.3). Below this threshold, the accuracy of the classifier is as low as 75% and above this threshold, accuracy is above 95% as shown in Fig. 6

6.2 Binary vs Reduced Multi-class Classifier

Our end goal is to tell if a circuit is original or altered which requires a binary classifier. Combining all of the non-original outputs of the multi-class classifiers to altered output results in an equivalent a binary classifier. To see this illustrated, the confusion matrices of the multi-class classifier are compressed to create a reduced binary confusion matrix. A confusion matrix is a way to visualize the effectiveness of a classifier. The rows represent the ground truth while the columns represent the inferred output from the classifier. An ideal classifier would result in a diagonal matrix where a classifier would correctly identify all cases. The matrices are reduced by combining all the results of altered circuits together (1, 2, 5, etc. changes) as shown in Fig. 3. The reduced binary classifier is used to determine our accuracy, sensitivity, and F1 score.

The comparison of the accuracy of a binary and reduced multi-class classifiers using the training data set is shown in Table 3. In all cases, the multi-class classifier performed as well or better than the binary classifier. Using a multi-class classifier may capture the subtle differences that small number of changes cause to the current draw from the circuit and will be used though the remainder of the paper.

6.3 Minimum Change Threshold

The confusion matrix for the c1355 circuit multi-class classifier has been evaluated to detect all changes with *PA* test set. The results are shown in Fig. 5a. The binary accuracy for this classifier is 74.9% and below our goal

Table 3 Comparison of the performance of a binary classifier with a reduced multi-class classifier

	Binary classifier	Reduced
c432	99.5	99.8
c499	99.8	99.8
c1355	98.0	99.3
c5315	99.6	99.6

Note that the reduced multi-class classifier performed as well or better than the binary classifier

of 90%. This is performance is representative of the other benchmarks analyzed in this paper (Table 1). To approach the goal, a new classifier was created that detected more than 0.17% change in the circuit by disregarding circuits that had only 1 gate altered. This resulted in confusion matrix shown in Fig. 5b and the accuracy improved to 89.8%. Raising the minimum change threshold, the change needed for our circuit to detect a circuit as being altered, to more than 0.85% change (disregarding the circuit simulations with 1 or 2 gate changes) yielded a new classifier and resulted in the confusion matrix shown in Fig. 5c and the accuracy increased to 96.8%. Continuing to raise the minimum change threshold to 1.70% is shown in Fig. 5d with an accuracy of 98.6%. The trend of increasing accuracy and sensitivity led to the formation of the idea to set a minimum change threshold.

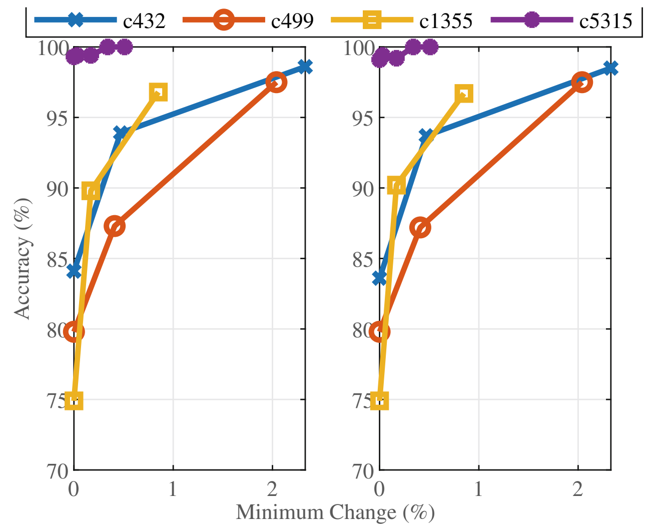
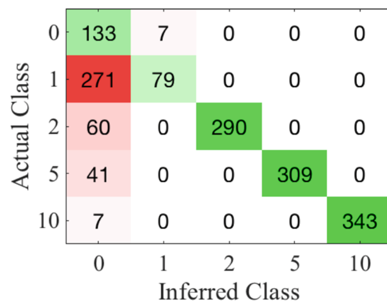


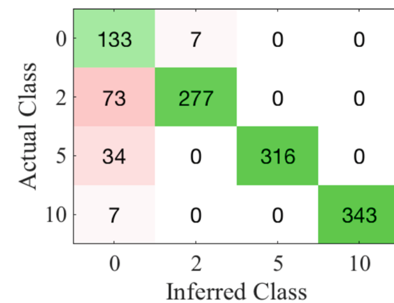
Fig. 6 Accuracy of classifiers (for *PA* case) to detect the pirated circuit from original one considering different minimum change threshold. Results are shown for a circuit aged with two different input vectors (Input Vector 1 (Left) and 2 (Right))

The trend of increase accuracy and sensitivity with the increase of the number of changed gates has been observed for all benchmark circuits we used in this paper. The accuracy and sensitivity for the classifiers for two different aged test data sets (Input Vector 1 and Input Vector 2) are shown in Figs. 6 and 7 respectively. In these figures, the test data sets are created by simulating the circuits with

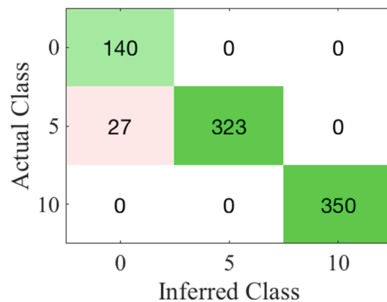
Fig. 5 **a–d** Confusion matrices showing the performance of multi-class classifiers on detecting various amounts of change on a test data set for the c1355 benchmark circuit with *PA* and various minimum change thresholds. The class refers to the number of alterations in a circuit (i.e., class 0 means there were no alterations, class 1 means there is one alteration in the circuit). Confusion matrices of other circuits (e.g., c432, c499, c5315) show similar performance



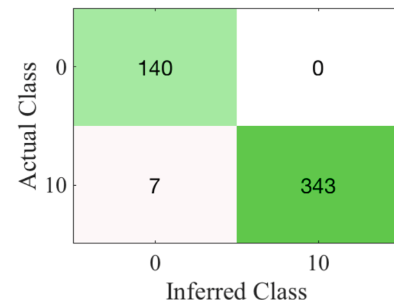
(a) Confusion matrix showing the performance of the multi-class classifier on detecting all changes.



(b) Confusion matrix showing the performance of the multi-class classifier on detecting more than 0.17% change (more than 1 gate change).



(c) Confusion matrix showing the performance of the multi-class classifier on detecting more than 0.85% change (more than 5 gate changes).



(d) Confusion matrix showing the performance of the multi-class classifier on detecting more than 1.70% change (more than 10 gate changes).

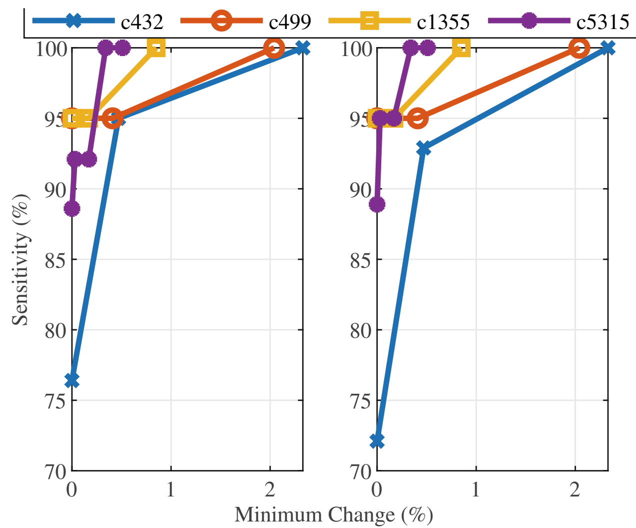


Fig. 7 Sensitivity of classifiers (for *PA* case) to detect the pirated circuit from original one considering different minimum change threshold. Results are shown for a circuit aged with two different input vectors (Input Vector 1 (Left) and 2 (Right))

different randomly generated input vectors (Input Vector 1 and Input Vector 2) during the aging process. Then, for the measurement phase, same input vector was used to create the training data set and the results were given to the classifier. In all cases, the accuracy (Fig. 6) was above 96% for a threshold value lower than 2.5%. The take away point from this observation is that a circuit is altered by more than 2.5% (i.e., 2.5% of the gates has been changed by the adversary while maintaining the circuit functionality intact), the classifier will predict it as altered with greater than 96% accuracy. The sensitivity (Fig. 7) is similar and with a minimum change threshold of 2.5%, the sensitivity is greater than 90%.

6.4 Classifier Testing

To fully test the proposed method, four sets of test data sets (*P*, *PA*, *PN*, and *PAN*) outlined in Section 5.3 were analyzed. Examining each of these test data sets ensures that the classifier is not overly fitting the data and shows the performance of the classifier with different parameters and ensures that the classifier is robust.

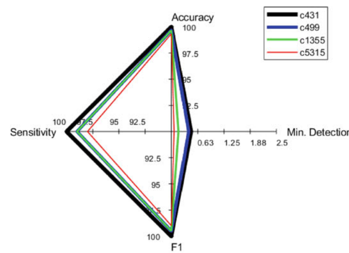
6.4.1 Classifier Testing in Presence of Process Variation (*P*)

This set of results demonstrate the accuracy, sensitivity, and F1 score for case where only process variation was considered in the simulation. The results are depicted in Fig. 8a. Similar to [5], in this experiment, the classier was able to identify the original circuit from the altered one with over 98% accuracy, sensitivity, and F1 score. Note that the blue bars in this figure show the minimum percentage of the altered gates to achieve this the reported accuracy, sensitivity, and F1 score.

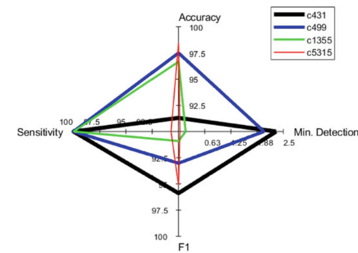
6.4.2 Classifier Testing in Presence of Process Variation and Aging (*PA*)

The next set of results is shown in Fig. 8b and represents the performance factors (accuracy, sensitivity, and F1 score) for the case where both process variations and aging were considered in the simulation (*PA*). As discussed in Section 6.3, for each benchmark circuit, we consider the

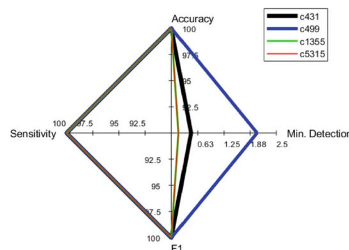
Fig. 8 **a**–**d** Minimum change threshold and associated performance (accuracy, sensitivity, and F1 score) of classifiers for ISCAS'85 c432, c499, c1355, and c5315 circuits. Note that for all test cases, the minimum detection is below the threshold level of 2.5%



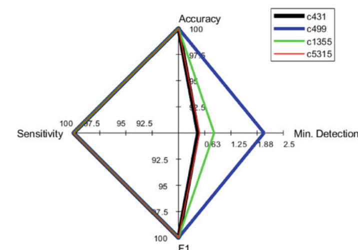
(a) Minimum change threshold and the associated performance for a test data set simulated with *P*.



(b) Minimum change threshold and the associated performance for a test data set simulated with *PA*.



(c) Minimum change threshold and the associated performance for a test data set simulated with *PN*.



(d) Minimum change threshold and the associated performance for a test data set simulated with *PAN*.

Table 4 Comparison of the performance of the design-based fingerprint to detect the original RISC-V processor from pirated versions with different minimum change thresholds

Detect more than:	Accuracy	Sensitivity	F1 score
1% Change	79.5%	0.84	73.7%
2% Change	77.2%	0.84	76.4%
3% Change	92.0%	100%	92.6%

Performance increase as minimum floor increases

threshold value as the minimum percentage of gates that when altered, all performance metrics of our proposed classifier are higher than 90%.

6.4.3 Classifier Testing in Presence of Process Variation and Noise (*PN*)

Examining the *PN* test data set resulted in Fig. 8c. As expected, it performed slightly worse than just process variation because noise may push the data point over the boundary of the SVM and therefore the data was miss classified. To compensate, the minimum change threshold was raised to maintain the performance of the classifier.

6.4.4 Classifier Testing in Presence of Process Variation, Aging, and Noise (*PAN*)

The final test data set investigated the robustness of the classifier with respect to the *PAN* test set with the results shown in Fig. 8d. Comparing the results with *PN* results (Fig. 8c), the minimum change threshold level had to be raised to maintain our performance metrics (accuracy, sensitivity, and F1 score) above 90%. This is expected when looking at the difference between *P* (Fig. 8a) and *PA* (Fig. 8b) data sets, in which the minimum change threshold of *PA* case was raised to keep the performance metrics above 90%. Note that the minimum change level for the c432 circuit was not as high as in the *PA* test set. One difference is that the testing using *PAN* aged the circuit to 7 years and did not use the intermediate years as was done in the testing using *PA*.

6.5 RISC-V Fingerprinting

The design-based fingerprint technique has been applied to the ISCAS'85 benchmark to evaluate the techniques performance on larger circuits; simulations were done on a RISC-V processor [45] (Available at [46]). This processor was compiled into a gate netlist that contains 22384 elements, where an element is a logical gate or flip-flop. Using process variation, 25 original circuits were simulated along with four circuits with 1%, 2%, and 3% alterations, each of with five process variations. The results are shown in Table 4. This work is initial and needs to be fully evaluated, but points to the design-based fingerprinting method working on larger circuits.

7 Discussion—Circuit Partitioning

To extend out work to larger circuits, the circuit can be partitioned into smaller sections by utilizing various procedures like the one outlined in [47, 48]. A circuit can be broken down into the power grid and the underlying CMOS logic, which can be further subdivided [49]. The

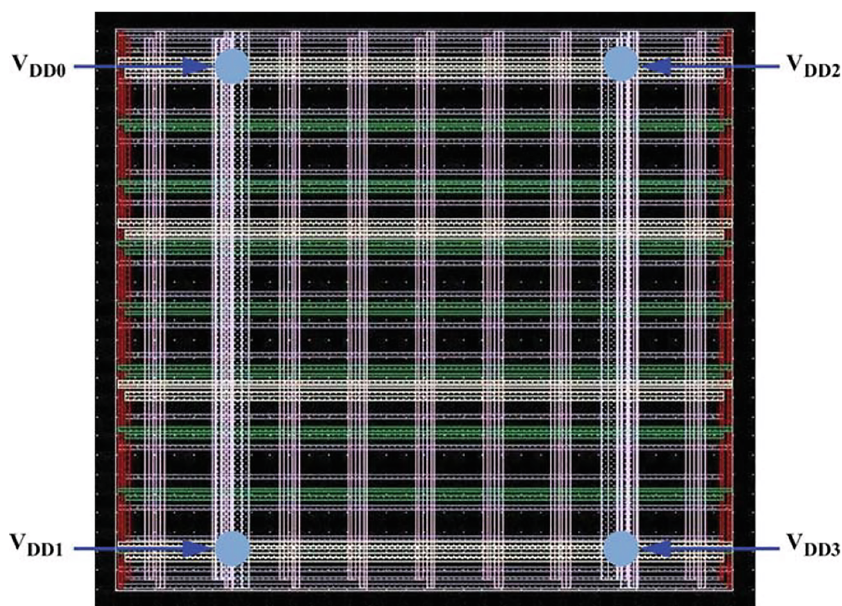


Fig. 9 Power grid layout for c499 circuit showing multiple power points to monitor (Fig. 15 from [49])

power to the IC is delivered via multiple pins where each are monitored. The power grid layout for c499 is shown in Fig. 9. Our procedure can then be applied to each partition by first validating the power grid. Then the various parts of the underlying CMOS logic can be tested and validated by selecting proper input vectors to activate portions of the logic. Using this method may require additional control circuitry to power off or put to sleep parts of the IC if the circuit cannot be adequately subdivided into small areas. It is possible to validate not only the whole circuit, but each of the partitions which may contain IP from different vendors with the outlined method.

Embedded systems and system on a chip (SoC) are both excellent candidates for the procedure proposed here. These systems are designed in blocks and the procedure outlined can be used to validate each block. The validation procedure, Universal Verification Methodology (UVM), divides the design into smaller pieces the with two verification levels: IP verification and sub-system verification [50]. Our method can be incorporated into these verification steps for each of the smaller pieces to have a complementary validation method to backup methods already in use.

8 Conclusions and Future Work

This paper targets a case of IP piracy in which an adversary who has access to the gate-level or RTL netlist of the circuit will resynthesize the design and fabricate the circuit under their name. In practice, resynthesizing is performed to hide the original source of the chip. To combat this, we propose a method to identify an IC as original or altered throughout its lifetime. In our experiments, we consider the effect of device aging, process variations, and noise as well.

Our method outlines a process to create a design-based fingerprint that can identify an IC or IP regardless of the age of the IC. The process uses current consumption of a circuit to include process variation and noise with a known set of input vectors that creates an SVM classifier that can be used to identify an IC that has had greater that 2.5% gates changed. This technique uses no additional circuitry and can detect a pirated IC with 96% accuracy. The pirated IC can be a new one or can be an IC used previously (up to 7 years). It is noteworthy that 96% accuracy was obtained in the presence of measurement noise and process variations.

Base on our work, there are multiple exciting paths for follow on work. Future work will include further examination of noise to include varying magnitudes of thermal and flicker noise, the effects on performance of the classifier, as well as their layout-level routing and power-grid effects. Evaluating the design-based fingerprint beyond 45-nm processes is left to future work. Varying the process includes varying the process variation parameters and correspondingly the minimum detection floor will have to be reevaluated with the new parameters. The aging process is dependent on temperature. Thereby, another new direction would be considering the effect of temperature during the aging process in the accuracy of the proposed method. Temperature accelerates aging and because the classifier is independent of age, we expect temperature will have little effect on the classifier. Implementing alternate machine learning techniques and including various kernels to lower the minimum detection level is reserved for future work. Initial findings for applying the technique to the RISC-V processor are presented, but additional work needs to be performed to fully validate this application. Finally, this technique may be applied to detecting hardware Trojans, but additional work needs to be done to validate this application.

Acknowledgments We would like to thank Brien Croteau for his help developing ideas.

References

1. Tehranipoor M, Koushanfar F (2010) A survey of hardware trojan taxonomy and detection. IEEE Des Test of Comput 27(1):10–25. https://doi.org/10.1109/MDT.2010.7
2. Liu B, Jin Y, Qu G (2015) Hardware design and verification techniques for supply chain risk mitigation. In: International Conference Computer-Aided Design and Computer Graphics (CAD/Graphics), pp 238–239. https://doi.org/10.1109/CADGRAPHICS.2015.53
3. Ni M, Gao Z (2004) Watermarking system for IC design IP protection. International Conference on Communications, circuits and system vol 2, pp 1186–1190. https://doi.org/10.1109/ICCCAS.2004.1346387
4. Marchand C, Bossuet L, Jung E (2014) IP watermark verification based on power consumption analysis. In: IEEE International System-on-Chip Conf. (SOCC), pp 330–335. https://doi.org/10.1109/SOCC.2014.6948949
5. Shey J, Karimi N, Robucci R, Patel C (2018) Design-based fingerprinting using side-channel power analysis for protection against IC piracy. In: Proceedings of IEEE International Symposium on VLSI ISVLSI
6. Rostami M, Koushanfar F, Karri R (2014) A primer on hardware security: models, methods, and metrics. Proc IEEE 102(8):1283–1295. https://doi.org/10.1109/JPROC.2014.2335155
7. Liu M, Kim CH (2017) A powerless and non-volatile counterfeit IC detection sensor in a standard logic process based on an exposed floating-gate array. In: Symposium on VLSI Technology, pp T102–T103. https://doi.org/10.23919/VLSIT.2017.7998211
8. Newbould RD, Carothers JD, Rodriguez JJ, Holman WT (2002) A hierarchy of physical design watermarking schemes for intellectual property protection of IC designs. In: IEEE International Symposium Circuits and System, pp IV–862–IV–865 https://doi.org/10.1109/ISCAS.2002.1010594
9. Bai F, Gao Z, Xu Y, Cai X (2007) A watermarking technique for hard IP protection in full-custom IC design. in: Int. Conf. Comms, Circuits and Systems, pp 1177–1180. https://doi.org/10.1109/ICCCAS.2007.4348256

10. Xu W, Zhu Y (2011) A digital copyright protection scheme for soft-IP core based on FSMs. in: Int. Conf. Consumer Electronics, Communications and Networks (CECNet), pp 3823-3826. https://doi.org/10.1109/CECNET.2011.5768225
11. Echavarria J, Morales-Reyes A, Cumplido R, Salido MA (2014) FSM merging and reduction for IP cores watermarking using genetic algorithms. In: International Conference on ReConFigurable Computing and FPGAs (ReConFig14), pp 1–7. https://doi.org/10.1109/ReConFig.2014.7032525
12. Lin M, Tsai G, Wu C, Lin C (2007) Watermarking technique for HDL-based IP module protection. In: International Conf. on intelligent information hiding and multimedia signal processing (IIH-MSP 2007), vol 2, pp 393–396. https://doi.org/10.1109/IIH-MSP.2007.326
13. Huang X, Cui A, Chang C (2017) A new watermarking scheme on scan chain ordering for hard IP protection. In: IEEE International Symposium on Circuits and Systems (ISCAS), pp 1–4. https://doi.org/10.1109/ISCAS.2017.8050823
14. Qu G (2002) Publicly detectable watermarking for intellectual property authentication in VLSI design. IEEE Trans Comput-Aided Des Integr Circ Syst 21(11):1363–1368. https://doi.org/10.1109/TCAD.2002.804205
15. Yu Q, Dofe J, Zhang Z (2017) Exploiting hardware obfuscation methods to prevent and detect hardware trojans. In: IEEE International Midwest Symposium on Circuits and System (MWSCAS), pp 819–822. https://doi.org/10.1109/MWSCAS.2017.8053049
16. Koteshwara S, Kim CH, Parhi KK (2018) Key-based dynamic functional obfuscation of integrated circuits using sequentially triggered mode-based design. IEEE Trans Inf Forens Secur 13(1):79–93. https://doi.org/10.1109/TIFS.2017.2738600
17. Wei S, Nahapetian A, Potkonjak M (2011) Robust passive hardware metering. In: IEEE/ACM International Conference Computer-Aided Design (ICCAD), pp 802–809. https://doi.org/10.1109/ICCAD.2011.6105421
18. Koushanfar F (2012) Hardware metering: a survey, pp 103–122. Springer, New York. https://doi.org/10.1007/978-1-4419-8080-9_5
19. Chakraborty RS, Bhunia S (2009) Harpoon: an obfuscation-based soc design methodology for hardware protection. IEEE Trans Comput-Aided Des Integr Circ Syst 28(10):1493–1502. https://doi.org/10.1109/TCAD.2009.2028166
20. Roy DB, Bhasin S, Nikolić I, Mukhopadhyay D (2019) Combining puf with rluts: a two-party pay-per-device ip licensing scheme on fpgas. ACM Trans Embed Comput Syst 18(2):12:1–12:22. https://doi.org/10.1145/3301307
21. Agrawal D, Baktir S, Karakoyunlu D, Rohatgi P, Sunar B (2007) Trojan detection using IC fingerprinting. In: IEEE Symposium Security Privacy, pp 296–310. https://doi.org/10.1109/SP.2007.36
22. He J, Zhao Y, Guo X, Jin Y (2017) Hardware trojan detection through chip-free electromagnetic side-channel statistical analysis. IEEE Trans Very Large Scale Integr (VLSI) Syst 25(10):2939–2948. https://doi.org/10.1109/TVLSI.2017.2727985
23. Jin Y, Makris Y (2008) Hardware trojan detection using path delay fingerprint. In: IEEE International Workshop on Hardware-Oriented Security and Trust, pp 51–57. https://doi.org/10.1109/HST.2008.4559049
24. Vaikuntapu R, Bhargava L, Sahula V (2016) Golden IC free methodology for hardware trojan detection using symmetric path delays. In: International Symposium on VLSI Design and Test (VDAT), pp 1–2. https://doi.org/10.1109/ISVDAT.2016.8064895
25. Huang K, Liu Y, Korolija N, Carulli JM, Makris Y (2015) Recycled IC detection based on statistical methods. IEEE Trans Comput-Aided Des Integr Circ Syst 34(6):947–960. https://doi.org/10.1109/TCAD.2015.2409267
26. Guo Z, Xu X, Rahman MT, Tehranipoor MM, Forte D (2018) SCARe: an SRAM-based countermeasure against IC recycling. IEEE Trans Very Large Scale Integr (VLSI) Syst 26(4):744–755. https://doi.org/10.1109/TVLSI.2017.2777262
27. Guin U, Forte D, Tehranipoor M (2016) Design of accurate low-cost on-chip structures for protecting integrated circuits against recycling. IEEE Trans Very Large Scale Integr (VLSI) Syst 24(4):1233–1246. https://doi.org/10.1109/TVLSI.2015.2466551
28. Guerin C, Huard V, Bravaix A (2007) The energy-driven hot-carrier degradation modes of nMOSFETs. IEEE Trans Device Mater Rel 7(2):225–235. https://doi.org/10.1109/TDMR.2007.901180
29. Tsetseris L, Zhou XJ, Fleetwood DM, Schrimpf RD, Pantelides ST (2005) Physical mechanisms of negative-bias temperature instability. Appl Physics Lett. https://doi.org/86:142103-142105
30. Rosa GL, Guarin F, Rauch S, Acovic A, Lukaitis J, Crabbe E (1997) NBTI-channel hot carrier effects in PMOSFETs in advanced CMOS technologies. In: Proceedings of IEEE International Reliability Physics Symposium, pp 282–286. https://doi.org/10.1109/RELPHY.1997.584274
31. Karimi N, Huang K (2016) Prognosis of NBTI aging using a machine learning scheme. In: IEEE International Symposium Defect and Fault Tolerance VLSI and Nanotechnology Systems (DFT), pp 7–10. https://doi.org/10.1109/DFT.2016.7684060
32. Karimi N, Danger JL, Guilley S (2018) Impact of aging on the reliability of delay pufs. J Electron Test (JETTA) 34(5):571–586. https://doi.org/10.1007/s10836-018-5745-6
33. Zhou C, Jenkins KA, Chuang PI, Vezyrtzis C (2018) Effect of HCI degradation on the variability of MOSFETS. In: IEEE International Relations in Physics Symposium (IRPS), pp P–RT.1–1–P–RT.1–4. https://doi.org/10.1109/IRPS.2018.8353684
34. Ziel AVD (1962) Thermal noise in field-effect transistors. Proc IRE 50(8):1808–1812. https://doi.org/10.1109/JRPROC.1962.288221
35. Hung KK, Ko PK, Hu C, Cheng YC (1990) A unified model for the flicker noise in metal-oxide-semiconductor field-effect transistors. IEEE Trans Electron Dev 37(3):654–665. https://doi.org/10.1109/16.47770
36. Brglez F, Fujiwara H (1985) A neutral netlist of 10 combinational benchmark circuits and a targeted translator in FORTRAN. In: IEEE International Symposium Circuits and System (ISCAS)
37. NanGate, Inc (2018) NanGate FreePDK45 open cell library. http://www.nangate.com/?page_id=2325
38. Synopsys, Inc (2018) MOS device aging analysis with HSPICE and CustomSim. https://www.synopsys.com/content/dam/synopsys/verification/white-papers/mosra-wp.pdf
39. Karimi N, Chakrabarty K (2013) Detection, diagnosis, and recovery from clock-domain crossing failures in multiclock SoCs. IEEE Trans Comput-Aided Des Integr Circ Syst 32(9):1395–1408
40. Hwang EJ, Kim W, Kim YH (2009) Impact of process variation on timing characteristics of mtcmos flip-flops for low-power mobile multimedia applications. In: Proceedings of International Symposium Integrated Circuits, pp 332–335
41. Mahor V, Chouhan A, Pattanaik M (2012) A novel process variation tolerant wide fan-in dynamic or gate with reduced contention. In: International Conference on Computers and Devices for Communication (CODEC), pp 1–4. https://doi.org/10.1109/CODEC.2012.6509271
42. Chang D, Ozev S, Sinanoglu O, Karri R (2014) Approximating the age of RF/analog circuits through re-characterization and statistical estimation. In: Design, Automatic and Test in Europe Conference on (DATE), pp 1–4. https://doi.org/10.7873/DATE.2014.048
43. Huang GB, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern Syst 42(2):513–529. https://doi.org/10.1109/TSMCB.2011.2168604

44. Hsu CW, Lin CJ (2002) A comparison of methods for multiclass support vector machines. IEEE Trans Neural Netw 13(2):415–425. https://doi.org/10.1109/72.991427
45. Celio C, Chiu P, Asanović K, Nikolić B, Patterson D (2019) Broom: an open-source out-of-order processor with resilient low-voltage operation in 28-nm cmos. IEEE Micro 39(2):52–60. https://doi.org/10.1109/MM.2019.2897782
46. RISC-V Foundation (2019) RISC-V cores. https://riscv.org/risc-v-cores/
47. Cong J, Lim SK (2004) Edge separability-based circuit clustering with application to multilevel circuit partitioning. IEEE Trans Comput-Aided Des Integr Syst 23(3):346–357. https://doi.org/10.1109/TCAD.2004.823353
48. Kadiyala Rao S, Robucci R, Patel C (2014) Simulation based framework for accurately estimating dynamic power-supply noise and path delay. J Electron Test 30(1):125–147. https://doi.org/10.1007/s10836-013-5425-5
49. Singh A, Plusquellic J, Phatak D, Patel C (2006) Defect simulation methodology for iDDT testing. J Electron Test 22(3):255–272. https://doi.org/10.1007/s10836-006-9318-8
50. Yun Y, Kim J, Kim N, Min B (2011) Beyond UVM for practical SoC verification. In: 2011 International SoC Design Conference, pp 158–162. https://doi.org/10.1109/ISOCC.2011.6138671

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.