Towards the Avoidance of Counterfeit Memory: Identifying the DRAM Origin

B. M. S. Bahar Talukder*, Vineetha Menon†, Biswajit Ray§, Tempestt Neal‡, Md Tauhidur Rahman¶
*§¶*Department of ECE, University of Alabama in Huntsville,* Huntsville, AL, USA
†*Department of CS, University of Alabama in Huntsville,* Huntsville, AL, USA
‡*Department of CSE, University of South Florida,* Tampa, FL, USA
{*bms.btalukder, †vineetha.menon, §biswajit.ray, ¶tauhidur.rahman}@uah.edu, ‡tjneal@usf.edu

***Abstract*—Due to the globalization in the semiconductor supply chain, counterfeit dynamic random-access memory (DRAM) chips/modules have been spreading worldwide at an alarming rate. Deploying counterfeit DRAM modules into an electronic system can have severe consequences on security and reliability domains because of their sub-standard quality, poor performance, and shorter life span. Besides, studies suggest that a counterfeit DRAM can be more vulnerable to sophisticated attacks. However, detecting counterfeit DRAMs is very challenging because of their nature and ability to pass the initial testing. In this paper, we propose a technique to identify the DRAM origin (i.e., the origin of the manufacturer and the specification of individual DRAM) to detect and prevent counterfeit DRAM modules. A silicon evaluation shows that the proposed method reliably identifies off-the-shelf DRAM modules from three major manufacturers.**

***Index Terms*—Manufacturer identification, IC forgery, DRAM forgery, DRAM counterfeiting, Anti-counterfeiting, Counterfeit memory.**

I. Introduction

With the globalization of the semiconductor supply chain and the growth of the semiconductor market value, counterfeit integrated circuits (ICs) have become an established threat to the semiconductor community [1]–[4], [58]. Counterfeit electronic parts and the risks associated with them have been increasing rapidly, which is reflected in the recent news [1]–[4]. Many commercially available memory chips are fabricated worldwide in untrusted facilities and, therefore, a counterfeit memory chip/module can easily enter into the supply chain in different formats: recycled, re-marked, tampered, out-of-spec, forged-documented, defective, cloned, overproduced etc. [1], [2], [5]–[8]. Recent studies show that the global market share of counterfeit IC is worth $\$169$ billion, and $\sim 17\%$ of which is contributed by memory chips [2], [5].

The inclusion of counterfeit components in an electronic system can endanger personal and national privacy, sabotage critical infrastructure, and damage the viability of entire business sectors because of their sub-standard quality, poor performance, and a shorter life-span [1], [5]–[8]. A counterfeit chip can fail any time after being deployed in the system, or they can be exploited to leak sensitive information or to allow remote access and endanger the integrity, confidentiality, and safety of a system by performing invasive or non-invasive fault-injection attacks [9]–[11]. Furthermore, recent experimental studies suggest that some DRAM modules are more vulnerable to rowhammer attack, a method of changing the restricted memory contents by repeated access to their adjacent rows [9], [11], because of their poor resiliency against noise, interference, etc.

A single solution to detect or prevent counterfeiting is unrealistic because of the diversity of counterfeit types, sources, and refinement techniques [1], [6]–[8]. Several discrete countermeasures have been proposed by the industry and academic researchers, which can be categorized into two major types: (i) electrical-based testing and (ii) physical-inspection based testing [1], [2], [5]–[8]. Some solutions are applicable to chips that are already in the market, and some solutions are integrated with the original chips for future tracking or metering [1], [2], [5]–[8]. Most of the solutions are ineffective for memory chips because they might require expensive equipment, expensive testing set-up, maintenance of an expensive database, and exhaustive enrollment process [2], [5], [12]. Besides, most physical-inspection based solutions are invasive and therefore, not applicable for mass-volume detection [2]. For mass-volume verification and low-cost testing, electrical-based testing is required that is non-invasive [5]. However, a non-invasive solution for counterfeit DRAM identification is extremely difficult because they may remain functional at the time of purchase and pass standard product qualification tests. Also, most existing solutions focus on a single counterfeit type (e.g., detecting recycled chips) [1], [2], [12]. Furthermore, current regulations are expensive and require a series of expensive testing methodologies [2], [5]. Therefore, a proper solution is required to identify counterfeit memory chips before deploying them in mission-, safety-, and security-critical systems.

In this paper, we propose a machine-learning-based technique to identify the origin of a DRAM manufacturer along with DRAM specification (i.e., density, grade, etc., see Sec. III for details) by exploiting DRAM latency (the required time to move charge from one location to another location in DRAM [13]–[15]) variations to detect and prevent major counterfeit types. The major contributions of this paper include:

- We propose a framework to identify the origin of the DRAM manufacturer by exploiting the facts that the architectural, layout, and manufacturing process variations are reflected in latency variations. The framework is also capable of verifying specification of individual DRAM module.
- We extract the most appropriate features from the latency-based erroneous patterns in DRAM modules to amplify the variations among manufacturers and specifications.
- We propose a machine learning approach to determine

the origin of the DRAM manufacturer based on the extracted features. The same method also separates DRAM modules of different specifications that are from the same manufacturer.
- We validate our proposed framework with off-the-shelf memory modules (commercial grade) from three major manufacturers- Micron, Samsung, and SK Hynix [16].
- We validate the robustness of our proposed technique against temperature and voltage variations.

The rest of the paper is organized as follows. In Section II, we present the background of DRAM architecture, read/write operation, low-latency induced errors. We also present a set of motivations for our proposed framework in Section II. We highlight our major objectives and necessary assumptions in Section III. We propose the manufacturer identification framework in Section IV. The experimental results are presented and discussed in Section V. We highlight the major limitations and the scopes of our proposed work in Section VI. We conclude our article in Section VII.

II. Background and Motivations

A. DRAM Architecture and Latency Variations

A DRAM system can have one module or several modules depending on the memory requirement. A DRAM module is divided into one or multiple ranks [13], [17]. Each rank consists of several DRAM chips; and together, they provide a wide data-bus (usually 64 bits). The DRAM memory structure is analogous to a 2-D memory cell array. For simplicity, we can consider that each bit of the 64-bit word comes from 64 individual cell arrays [13], [17]. The rows of the DRAM are known as wordline (or page). The columns are known as bitline, and the chip density determines the total number of rows. The bitlines are connected to the row-buffer, a series of sense-amplifiers. Several DRAM cells are connected to a wordline. A DRAM cell consists of two components- an access transistor and a capacitor to hold the charge. The access transistor connects the capacitor with a bitline and is controlled by the wordline. The state of charge in the capacitor determines the memory content (i.e., '1' or '0'). A fully charged capacitor represents logic '1' and an empty capacitor represents logic '0'. However, the stored charge in the capacitor leaks away, which leads to an incorrect reading after a certain amount of time. So, to ensure the integrity, the content of a DRAM cell needs to be refreshed periodically before the memory contents flip. This time interval is known as the retention time, which is 32 or 64 milliseconds (ms) [17]. A failure to refresh before the retention time can alter the memory content. To

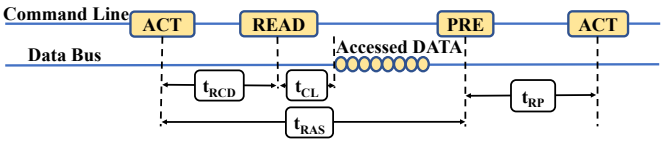


Fig. 1: DRAM timing at reading cycle [13].

operation. A read operation starts with an ACTIVATE (ACT) command executed by the memory controller. t_{RCD} is called the *Activation* latency which is required to activate (turn 'ON') the access transistor properly. The activated access transistor creates a conducting path between the storage capacitor and the bitline. The charge stored in the capacitor perturbs the bitline voltage. The sense-amplifier senses the perturbed voltage and amplifies it to an appropriate binary value. At this moment, the memory controller applies the READ command to read the data and fetch it to the data bus. The minimum time latency between the READ command and the first data bit to appear in data-bus is called *Column Access Strobe* latency or *CAS* latency (t_{CL}). DRAM's read operation is a destructive process. Therefore, the charge on the DRAM storage capacitor needs to be restored after each successful reading. The time required to activate an access transistor and to restore the charge on the corresponding storage capacitor is known as the *Row Active* latency or *Restoration* latency (t_{RAS}). At the end of the restoration process, the memory controller again applies the *PRE* command to re-initiate all the bitlines for the next read/write operation. The *PRE* command precharges all bitlines to V_{ref}. The time required to precharge all bitlines properly is called *Precharge* Time (t_{RP}). The *PRE* command also deactivates all previously activated access transistor. The $t_{RAS} + t_{RP}$ is the total time required to read a DRAM row properly; this total time is called *Row Cycle* Time (t_{RC}).

Usually, the DRAM manufacturer specifies a set of timing parameters for reliable read and write operation [17]. At the reduced timing latency below the standard value, we experience unreliable read and write operations, which is different from one module to another [13], [14], [17]. In our proposed method, we capture the architectural, layout and manufacturing process variations by exploiting the errors originated at the reduced *Activation* latency (t_{RCD}).

B. Counterfeit, Existing Work, and Motivations

The modern horizontal semiconductor supply chains involve several parties to reduce the fabrication cost and time to market [1], [2], [5], [18]. In this model, a chip is designed in one place while fabricated in a different place. Because of traveling IPs in different formats and involvement of untrusted parties, the modern semiconductor supply chain suffers from counterfeiting (such as hardware trojan or malicious change in third-party IP or chip layout, cloning IPs/ICs, remarking, etc.) [1], [58], [59]. Fig. 2 shows the IC/DRAM design flow for authentic-chip and pirated-chip production cycle. The untrusted party (the third-party IP developer, the foundry, the assembly, the distributor, etc.) can perform counterfeiting at different phases of the manufacturing process. An untrusted party can send out overproduced, and out-of-spec/defective ICs/DRAM chips to the market. The untrusted party also can clone by stealing IPs or by reverse engineering (from a post-fabricated product) the original chip to avoid research and development (R&D) costs. Recycled ICs also can be added to the supply chain at different stages (e.g., in the foundry or the assembly). Below, we summarize (but not limited to) the motivations of our proposed work.

i) Counterfeit DRAM chips are spreading at an alarming rate: Like other electronic chips, counterfeit DRAM

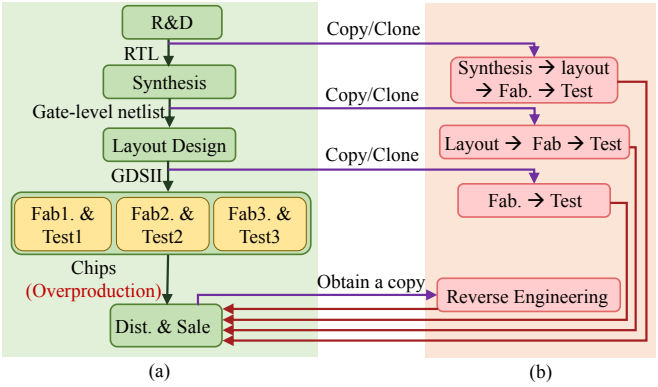


Fig. 2: (a) Authentic-chip production cycle vs. (b) counterfeit-chip production cycle [19].

chips have been spreading worldwide at an alarming rate [3], [4], which is only worsening the current semiconductor industry crisis [2], [5], [58].

ii) Existing countermeasures and their limitations: In 2015, the Department of Defense imposed several rules to stop entering counterfeit components in the defense supply chain [5]. These rules create more accountability and require proper testing standards along with maintaining a database if the electronic components are acquired from the untrusted party. Researchers, industries, and several organizations have developed several test plans or standards that require a series of testing methods. Visual inspection, X-ray imaging, scanning electron microscopy (SEM), energy disruptive spectroscopy, terahertz spectroscopy, etc., are the most common physical inspection-based techniques that are used to detect counterfeit chips [1], [6]–[8]. Some imaging techniques capture the internal features of a sample and might be very effective to identify certain counterfeit types. However, these physical inspection-based techniques usually require expensive tool, long test time for sample collection and imaging, subject matter experts [1]. Curve trace testing (CTT), parameter testing, burn-in testing, aging-based analysis, etc. are the most common electrical test methods that can be used to identify counterfeit components [20]–[22]. Existing burn-in and aging analysis-based techniques partially destructive because months to years of the device are consumed from the accelerated aging. The parameter testing is useful but requires expensive test setup and complex test programs. On the other hand, the CTT (electrical testing to capture damaged packages, broken or damaged wires, cracked die, etc.) is only useful for detecting recycled ICs [2], [5].

Researchers also have proposed pre-fabrication techniques to prevent counterfeiting such as hardware metering ([1], [23]), secure split test (SST) ([58]), placing on-chip sensors ([2], [20], [21], [24]), electronic chip ID ([1]), DNA marking ([25]), RFID-based tracking ([26]), blockchain-based traceability ([27]), PUF-based techniques ([10], [14], [28]–[31]), etc. Hardware metering and SST are used to provide post-fabrication control but require hardware changes and complex supply chain management. These techniques also need to change the fabrication flow. Moreover, these methods require active and exhaustive communication between the foundry and the design house. Besides, each of the chips needs to go through the unlocking process, which is tedious and adds extra steps in the supply chain. On the other hand, on-chip sensors are used to monitor the device degradation but need some additional sensors and monitoring units. ECID tags each chip with a unique ID by adding some non-programmable memory (such as OTP or ROM). The ECID-based solution is susceptible to tampering. In DNA marking technique, a mixed plant DNA is used to create a new DNA sequence [25]. Later, this sequence is applied with the ink to mark the chip package. However, this technique suffers from a complex authentication scheme. For the PUF-based technique, the unique ID generated from each chip can be used to identify authentic chip. However, the PUF-based techniques require an extensive database for registration purposes. The registration and authentication procedures are exhaustive as well. The DRAM memory itself can be used as a PUF, but the uniqueness of PUF can make the solution very challenging. Moreover, most of the existing solutions address a single counterfeit type.

Our proposed machine-learning based technique has some advantages compared to other possible manufacturer identification techniques:

- In the proposed machine-learning based technique, we store minimal information (a few statistical parameters) to attest and detect a large group of memory modules/chips. On the other hand, for example, in PUF-based technique, we need to create a golden data-set by accumulating all challenge-response pairs for an individual memory module/chip.
- Our proposed technique does not need any exhaustive registration processes. Analyzing a small number of samples from a large group of modules is sufficient enough to reveal the detail statistical information of that group.
- Furthermore, the machine-learning based technique can serve some particular purpose; for example, the user might want to test the authenticity of the purchased device without knowing any details of chip design. In such a case, the user does not need to have exclusive access to any sensitive information (e.g., the challenge-response database of PUF-based scheme).
- This technique does not require any hardware modification unlike many existing techniques such as: hardware metering, SST, on-chip sensor-based authentication, ECID, etc. [1].
- The proposed solution is invasive and less expensive compared to some other techniques.

iii) Importance of attesting the DRAM origin: Attesting the origin of the foundry or the manufacturer is a critical need to (a) enforce the license agreement, (b) ensure and track the quality of the chip, (c) rank the manufacturer based on the quality and durability of their products, (d) protect the intellectual property, (e) ensure accountability, and (f) stop the spread of counterfeit memory chips worldwide. Counterfeiters usually hide the origin of the foundry, fake the quality of the original memory modules/chips, and infringe on the rights of the original manufacturer [3], [4], [6]–[8], [58]. On the other hand, verification of individual DRAM module's specification can detect certain counterfeit types such as upgrading a DRAM

module through remarking or forged documentation.

The existing techniques on attesting of the foundry rely on simulation or test data from fabrication and packaging facilities [6], [7]. Unfortunately, in most cases, the testing data are not made publicly available and, therefore, a party that does not have access to those test data cannot make the classifier and (or) cannot verify the ratified foundry. In contrast, in our proposed technique, the DRAM chips can be authenticated based on trained classifier provided by the manufacturer or a trusted third party. In the proposed technique, the verifier does not require any prior knowledge of the manufacturing process. The classifier input, a set of features, can be easily evaluated in any low-cost embedded or FPGA-based system.

iv) Vulnerability of counterfeit memory chips: A counterfeit memory module can be more vulnerable to attacks because of their less resiliency against noise. For example, recent studies demonstrate that some DRAM modules possess inferior quality than others, which are more susceptible to rowhammer attack [11]. Our experimental results also suggest that a recycled DRAM chip is $\sim 8\%$ more vulnerable to rowhammer attack.

III. Objectives and Assumptions

The objective of this work is to identify the DRAM origin (i.e., the origin of the manufacturer and the specification of individual DRAM) reliably by capturing all variabilities. The major variations to attest and identify the origin of the manufacturer include:

- **Architectural variations:** Manufacturers usually optimize the DRAM architecture in various ways to support the specifications and product cost [34]. For example, manufacturers shrink the die size to reduce the cost per cell, which can cause the DRAM more susceptible to noise, more vulnerable, and less robust. The minimum required latency parameters vary from one architecture to another. Therefore, for a given reduced timing latency, different DRAMs from different manufacturers create a different amount of disturbance errors [13], [14].
- **Layout variations:** Chip layout variation from one manufacturer to another may originate from several sources such as chip area, floorplanning, placement, and routing, etc. [35]. Because of the layout variation, the RC paths (i.e., the delay) and some other electrical parameters vary, which are reflected in the DRAM latency parameters [13], [14].
- **Process variations:** The intrinsic process variation can be either random or systematic [36], [37]. The random process variation can be considered as noise and varied among the chips fabricated on a single silicon wafer. On the other hand, the systematic process variation depends on the quality of the fabrication plant and also related to the microstructural locality and pattern. The process variation can lead to different error patterns at a given reduced latency parameter and can reveal the information about the fabrication plant and the microstructure.

Our proposed technique of identifying the memory manufacturer is based on the following assumptions.

- **Assumption on memory class**: A manufacturer ships memory module with a part number on it, which contains the manufacturer information and chip specification, such as density, speed grade, package, temperature range, bus width, die generation, etc. [38]. In addition to part-number, a manufacturer also provides additional module specification on the module label [39], such as JEDEC PCB layout version [40], SPD version [32], manufacturing country, manufacturing lot number, etc. Timing parameters of the DRAM module are specified into SPD data [32]. In this work, two DRAM modules are considered as two different classes, if one of the following information is mismatched: i) manufacturer, ii) part number, and iii) PCB layout version/SPD data. A change in one specification can lead to different GDSIIs (related to the die generation, and specification), packaging, PCB layouts, or SPD data. Note that a manufacturer may send a single GDSII file to different fabrication plants. We assume that fabrication plants with the same GDSII follow similar design rules to minimize the effect of systematic process variations.

  In this article, 'sample', 'positive sample', and 'negative sample' mean memory module under test, memory module that originally belong to the target class, and memory modules that originally do not belong to the classifier target class (i.e., belong to the outlier region), respectively.
- **Assumption on data training, and verification**: We extract different features to capture the architectural, layout, and process variation. These features are trained to learn a statistical model. In our proposed scheme, the manufacturer or a trusted party is responsible for training the statistical model and releasing it for public use. We also assume that, while training the statistical model for a particular memory class, manufacturers do not have any statistical information from other memory classes (i.e., from negative class). However, in practice, the manufacturer may collects a few random samples from other classes. Training statistical model with some negative examples might be beneficial, but it is almost impossible to collect all samples from all negative classes because of the diversity of memory chips/modules and several manufacturers. We also assume that the chips/modules that are used for the enrollment in the proposed technique are authentic. The regular consumers should able to verify their purchased DRAM class by only using the statistical model. We also assume that consumers do not have any knowledge on memory architecture and manufacturing process.

IV. Proposed Method

To extract all possible variabilities, we reduce the DRAM timing latency and obtain the signatures (i.e., the error pattern or fail bit count) that reflect the architectural, layout, and process variations. Below, we present a framework to identify the DRAM class that involves several steps.

Step 1: Data acquisition. The experimental results show that the latency-induced error pattern depends on the data written into the memory, the amount of latency reduction, and the DRAM module. To capture the maximum variations among the memory clasess, we characterize the erroneous read pattern

after writing four different sets of data: (i) Data set 1: Written with solid data pattern (all 1's), (ii) Data set 2: Written with inverse solid data pattern (all 0's), (iii) Data set 3: Written with column stripe data pattern ($101010\cdots$), (iv) Data set 4: Written with inverse column stripe data pattern ($010101\cdots$). We read above data patterns at the reduced *Activation* time (t_{RCD}) to get module dependent erroneous outputs.

Step 2: Feature selection. It is crucial to select the optimum number of features since the performance of classifiers is sensitive to the choice of the features and features' attributes such as correlation, noise, and other factors. In this step, we will select the key features that can effectively capture architectural, layout, and process variations observed in DRAM since they directly impact the accuracy, computation time, and storage (of golden data) of our proposed technology. The classification models are created based on a total of 26 features collected from the four sets of data. Features are extracted from the whole data that is read out from one page. A single bank from 1GB memory module contains 16k+ pages per bank (eight banks per module), and for the case of a 2GB memory module, each bank includes 32k+ pages. Each memory page contains 1,024 words and each word contains 64-bits of data. The data collected (at reduced t_{RCD}) from each memory pages are then rearranged into a 1,024×64 (denoted as d_R of size $w \times b$, where, $w =$ number of words in a page, $b =$ number of bits in each word) binary array. Moreover, for each page, we create another array, d_F (same size of d_R) which tracks the location of flipped bits. Note that, the $d_F(i,j) = 1$, if $d_R(i,j)$ is flipped with respect to the actual data that is written to the DRAM otherwise, it will be 0. The following features have been chosen from each page to identify the DRAM origin (i.e., the origin of the manufacturer and the specification of individual DRAM).

Feature 1 (Ψ_1): The total number of flips, also known as failed bit count (FBC), is used to capture the data dependency, process variation, and layout variation of the DRAM chips. The silicon results show that the FBC counts change from one DRAM module to another module.

Feature 2 (Ψ_2): The subset of FBC bits that are flipped to logic 1.

Feature 3 (Ψ_3): The compression ratio (r) depends on the distribution of ones and zeros in a string (i.e., randomness). The compression ratio is defined as Eq. 1.

$$r = \frac{S_u}{S_c} \tag{1}$$

Where S_u and S_c are the sizes of the uncompressed and compressed data respectively.

Our preliminary experimental results show that the compression ratio of d_R varies from one manufacturer to another. We compress data using standard ZLIB library [41] and then compared the data size with the original data. The ZLIB library is optimized for the minimal computational overhead while compressing the data.

Feature 4 (Ψ_4): The whole block of d_F is divided into a set of smaller blocks (each block is 64×1 of size and denoted by B_w). The standard deviation on the FBCs in these B_ws are considered as a feature. This feature captures the spatial locality of FBC along the dimension w. The higher value of standard deviation represents a greater spatial locality.

Feature 5 (Ψ_5): The block d_F is divided into a smaller block, B_b (of size 1×8) and then FBC is counted on each of the smaller blocks. Then we choose the standard deviation of those FBCs as the feature Ψ_5. The spatial locality along the dimension b is captured with this feature Ψ_5.

Feature 6 (Ψ_6): Like Ψ_4, we calculate the standard deviation of FBCs on 64 blocks (of size 1024×1). This feature captures the fact that some 2-D memory arrays are more error-prone than others.

Feature 7 (Ψ_7): Like 5, we calculate the standard deviation of FBCs on 1024 blocks (of size 1×64). This feature explores the fact that some bitlines are more error-prone than others.

All features except the Ψ_2 is extracted from all four data sets. The feature Ψ_2 is only extracted from the dataset 3 and dataset 4 (see Step 1).

Step 3: Machine-learning algorithms for detecting the DRAM origin. After extracting the most suitable feature, we develop a machine-learning based technique to identify counterfeit DRAM modules. In our proposed technique, we use one-class classifier. Although one-class classifier is a more complex statistical problem, recent works demonstrated that it is more advantageous compared to other machine learning based techniques while detecting counterfeit ICs [7], [22], [42]–[44]. On the other hand, in the traditional binary-class classifier, if the statistical diversity is enormous in the negative samples, the classifier might provide poor decision boundary due to the small negative train data [45], [46]. In such a scenario, it will be very expensive or even impossible to collect data from the negative class covering the wide statistical diversity. This situation is particularly true for counterfeit IC detection as counterfeit ICs can be introduced from a wide variety of sources (see Sec. II-B). On the other hand, the one-class classifier [45]–[49] is trained by only positive class samples. In our proposed method, we use Support Vector Data Description (SVDD) [45], [46] to detect the outliers of a specific class. SVDD creates a spherical decision boundary in feature-space around the train data-set of a given class. For a given training data $x_i \in \mathcal{R}^n, i = 1, 2, 3, ..., l$, Tax et al. [46] solved the following optimization problem given by Eq. 2.

$$\begin{aligned} &\min_{R,\theta,\xi} R^2 + \mathcal{C}\sum_{i=1}^{l}\xi_i \\ &\text{subject to, } \|\varphi(\boldsymbol{x}) - \boldsymbol{a}\|^2 \leq R^2 + \xi_i, i = 1, 2, 3, ..., l \\ &\xi_i \geq 0, i = 1, 2, 3, ..., l \end{aligned} \tag{2}$$

Here, ξ_i is a slack variable and $\varphi(\boldsymbol{x})$ is the mapping function from the lower dimension to a higher dimension. R and $\boldsymbol{a}$ are the radius and center of the encircling boundary, and $\mathcal{C}$ is the regularization parameter. A smaller value of $\mathcal{C}$ causes more training samples to be treated as an outlier. A sample will be considered as an outlier if $\|\varphi(\boldsymbol{x}) - \boldsymbol{a}\|^2 > R^2$. However, Eq. 2 can be efficiently solved by the Eq. 3.

$$\begin{aligned} &\max_{\alpha} \sum_i \alpha_i \mathcal{K}_{i,i} - \sum_{i,j} \alpha_i \alpha_j \mathcal{K}_{i,j} \\ &\text{where, } \sum_{i=1}^{l} \alpha_i = 1 \end{aligned} \tag{3}$$

Here, $\mathcal{K}$ is the kernel function (i.e., $\mathcal{K}_{i,j} = \langle \varphi(x_i)^T \cdot \varphi(x_j) \rangle$). In our case, we have chosen a radial basis function (Eq. 4) as a kernel function [50]. The radial basis function is useful when the data are not linearly separable.

$$\mathcal{K}_{i,j} = exp(-\gamma \left\| x_i - x_j \right\|^2), \gamma > 0 \tag{4}$$

In Eq. 4 γ is a free parameter. A larger value of γ enables the classifier to capture more complex attributes of the training data. On the other hand, the classifier model might suffer from overfitting problem if the value of γ is too large. However, the $\mathcal{C}$ and γ can be optimized more efficiently by introducing artificial outliers and applying k-fold cross-validation [51]. Moreover, the classifier accuracy can be increased by introducing some real negative examples during training [46].

Step 4: Constructing a framework to detect DRAM manufacturer. Fig. 3 presents the proposed framework to identify the DRAM origin (i.e., the origin of the manufacturer and the specification of individual DRAM). In the proposed framework, we assume that the manufacturer or a trusted party provides the classifier model to the consumer and also defines a threshold for *Positive Page Rate* (PPR). *The positive page rate* (PPR) is defined as follows-

$$PPR = \frac{\textit{No. of pages that are classified as 'positive'}}{\textit{No. of test pages from the memory module}} \tag{5}$$

In Fig. 3, the steps, shown in the blue region, are performed by the OEM (Original Equipment Manufacturer), and the steps shown in the green region are performed at the consumer end. All other steps (covered by the orange region) can be processed in either consumers' system or manufacturers' system. Initially, the OEM or a trusted party trains a classifier based on all page data that are captured from one or multiple DRAM samples of the target class. The OEM or trusted party should also specify the number of memory pages that need to be tested from a memory module to prove its authenticity. Then, based on the sample statistics, the OEM should choose a threshold (λ_{PPR}) to decide whether a DRAM is manufactured by them or not. If the PPR from the test module is higher than the threshold, then the memory module should be considered as authentic. The selection of the threshold value λ_{PPR} and the number of test pages (n) depend on the quality of the classifier and the manufacturing process. Higher process variations might cause a large statistical diversity on the manufactured memory modules and may increase the chance of miss-classification. Besides, a larger process variation may lead to a higher statistical variation among the memory pages from the same DRAM module. In such a case, we might need more randomly sampled memory pages (i.e., a larger value of n) to capture all architectural and manufacturing process variations of a DRAM module. The choice of λ_{PPR} mostly depends on the quality of classifier. Fig. 4a shows that the distribution of PPR from positive samples and negative samples have an overlapping region. In such case, it is not possible to select a λ_{PPR} that creates a clear boundary between the positive samples and the negative samples. On the other hand, if the distribution of the PPR is mutually exclusive (Fig. 4b), selecting a λ_{PPR} within the interval [$PPR_{neg,max}$, $PPR_{pos,min}$] will separate positive and negative samples. Therefore, the ideal goal should be, maximizing the separation between $PPR_{neg,max}$ and $PPR_{pos,min}$ for a suitable value of λ_{PPR} during classification. As it is difficult/impossible to collect the negative class data that covers the whole distribution (discussed in Step 3), the λ_{PPR} should be defined with the highest possible value (i.e., $\lambda_{PPR,op} = PPR_{pos,min}$). In our proposed scheme, the OEM should train a classifier C_m, corresponding to a specific memory class and make the classifier parameter public. Then, the user should choose random n test pages from the memory module that is under test. The general information given with the classifier C_m should enable the user to extract features form those selected pages. Then, for each of those n test pages, the OEM/user should test the extracted features using the classifier C_m. If the PPR (calculated from Eq. 5) is higher than the threshold λ_{PPR}, the memory module should be marked as authentic. Otherwise, it should be identified as a counterfeit one.

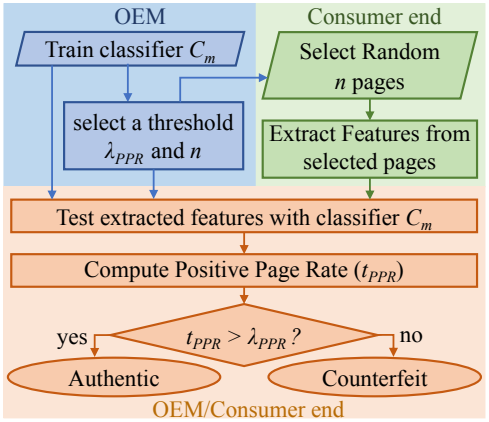
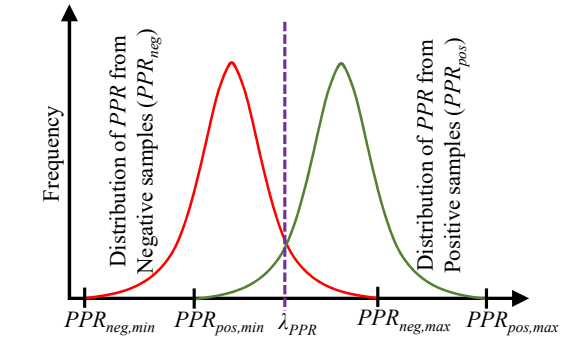


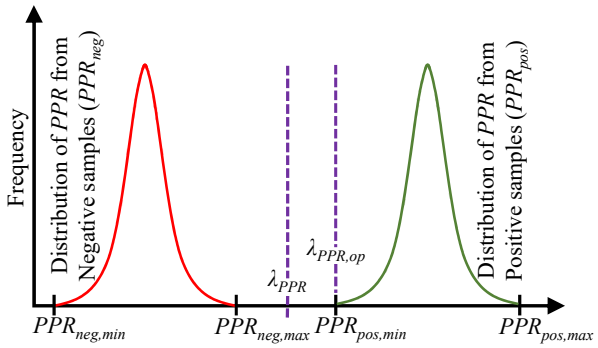
Fig. 3: The proposed framework that is used to prove the authenticity of the origin of the DRAM manufacturer along with specification.

V. Results and Analysis

In our experiment, we have collected data from 25 commercial off-the-shelf single rank DDR3 SODIMMs (small outline dual in-line memory module) from 3 major DRAM manufacturers (see Table 1) [16]. We have tagged the memory class based on the part number, the Garber version (reference PCB layout version [40]), and the SPD [32] data (see Table 1). Among the first six classes, we found at least one mismatch in their SPD data (detailed of SPD data is not presented in this article). On the other hand, the last two classes only differed by their PCB layout version and SPD version. From each memory module, we have collected data from all memory banks (each DRAM module contains eight banks). The testing platform has been implemented using a Xilinx Virtex ML605 evaluation board with SoftMC [52]. Data have been written and fetched from the DRAM memory module with two 32-byte data bursts. For all memory modules, we have observed error patterns below $7.5ns$ of *Activation* time (the recommended *Activation* time is in between $10ns$ to $15ns$ [17]). In this work, we have conducted our experiment with a $5ns$ of *Activation* time which should be achievable by most of the system. In order to quantify



(a)



(b)

Fig. 4: Selecting λ for case- (a) when the distribution of PPR_{neg} and PPR_{pos} are overlapped, (b) when the distribution of PPR_{neg} and PPR_{pos} are mutually exclusive.

the robustness of our proposed technique, we have evaluated our proposed method in four different operating condition- i) nominal voltage (1.5v) and room temperature (25°C) (NVRT), ii) high voltage and room temperature (HVRT), iii) low voltage and room temperature (LVRT), and iv) nominal voltage and high temperature (NVHT). For HVRT and LVRT, we have changed the input voltage (V_{DD}) by $\pm 1\%$ as most of the memory controllers limit the voltage ripple within $\pm 1\%$ [53]. For NVHT, we have changed the operating temperature by $+15$°C from the room temperature.

Manufacturer	Part Name †	Country Origin	Quantity	SPD-Garber Version	Class tag
Micron	M1	China	2	10-C1	1
	M2	Singapore	3	10-B1	2
		China	4		
Samsung	S1	China	1	10-B1	3
	S2	China	1	11-B2	4
	S3	China	4	11-B2	5
		Philippines	3		
SK Hynix	H1	Korea	5	10-B1	6
		China	1	11-B2	7
		Korea	1		

TABLE I: Memory modules used in the data set.

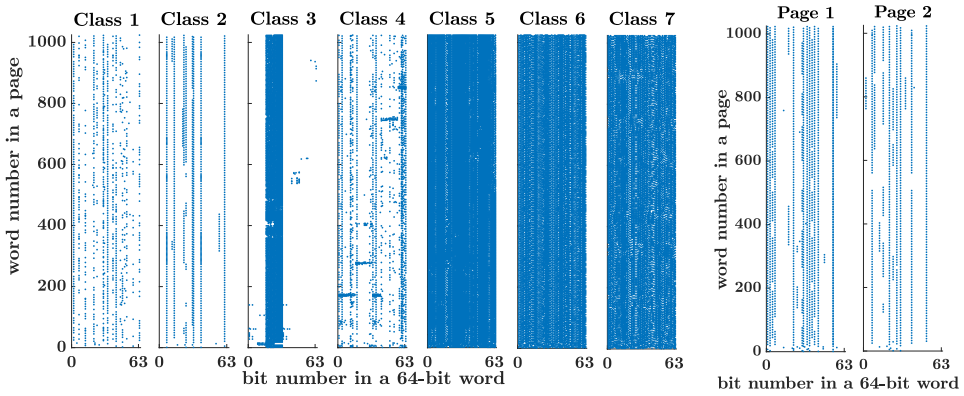
†***M1:*** *MT4JSF12864HZ-1G4D1,* ***M2:*** *MT8JSF12864HZ-1G4F1,* ***S1:*** *M471B2873EH1-CF8,* ***S2:*** *M471B2873GB0-CH9,* ***S3:*** *M471B5773DH0-CH9,* ***H1:*** *HMT325S6BFR8C-H9;*
M1, M2, S2: *1GB 1333MT/s,* ***S1:*** *1GB 1066MT/s,* ***S3, H1:*** *2GB 1333MT/s*

Fig. 5 presents the spatial locality of failed bits in a randomly chosen page form each memory class at NVRT operating condition. From the scatter plot, we observe that the error pattern is different for different classes. Note that, the PCB layout version only differs class 6 and class 7 and the subtle difference is difficult to understand from the figure (Fig. 5a). In Fig. 5b, we have presented the spatial locality of failed bits on two random pages from the same memory module of the same class. Although there is some similarity in their texture, the pattern is not consistent. The features extracted from these samples (as discussed in Sec. IV) are still capable of separating these classes.

From each memory page, we have extracted a total of 26 features as described in Sec. IV. We have applied these 26 features directly to train and test the classifier. However, we have used the Linear Discriminant Analysis (LDA), a linear transformation [54], to provide the best visualization of the class separability. The LDA projects the data into a lower dimension feature-space by keeping the maximum separability among the classes. In the LDA, the lower dimension and the higher dimension features are linearly dependent. The data distribution in lower dimension feature-space is presented in Fig. 6. In this figure, we have only considered the most significant 5 dimensions ($\Phi_1, \Phi_2, \Phi_3, \Phi_4$, and Φ_5) in the new feature-space that provides the maximum separability (explained variance). From the figure, we observe that each of the class forms a cluster in the feature space, which enables the separation of manufacturers.

To demonstrate our proposed method, we trained one-class SVDD classifier (as discussed in Sec. IV) for each class where we assume that the manufacturer does not have any prior knowledge of the memory modules from other class (See Sec. III). The classifier was only trained at NVRT operating condition, and the same classifier was used to test the data from other operating conditions. Training one class classifier is a more complex statistical problem compared to the multi-class problem. We used LIBSVM library to implement the one class classifier [55]. For each class, we selected only one module to train the classifier (using 26 features from all pages collected at NVRT condition) and then tested all the pages from the rest of the 24 modules with the trained classifier with all operating conditions.

To validate our algorithm, we have chosen all possible combinations of training and testing data sets. In Table II, we have presented the result from the one-class classifier. The third, fourth, and fifth columns of the table represent the mean, the standard deviation, and the minimum PPR from the positive samples for each classifier (PPR is calculated from each test module). The sixth, seventh, and eighth columns of the table represent the mean, the standard deviation, and the minimum PPR from the negative samples. For the ideal case, the PPR_{pos} and PPR_{neg} should be 100% and 0% respectively. The standard deviation for both cases should be 0%. A larger gap between the PPR_{pos} and PPR_{neg} provides us the flexibility of choosing appropriate values of λ_{PPR} and n for identifying the origin of the manufacturers along with specification with high confidence (Sec. IV). Our silicon results provide a satisfactory difference between the PPR_{pos} and



(a) (b)

Fig. 5: (a) Spatial locality of failed bits from a randomly chosen page from each class, (b) Spatial locality of randomly chosen two pages from same memory modules.

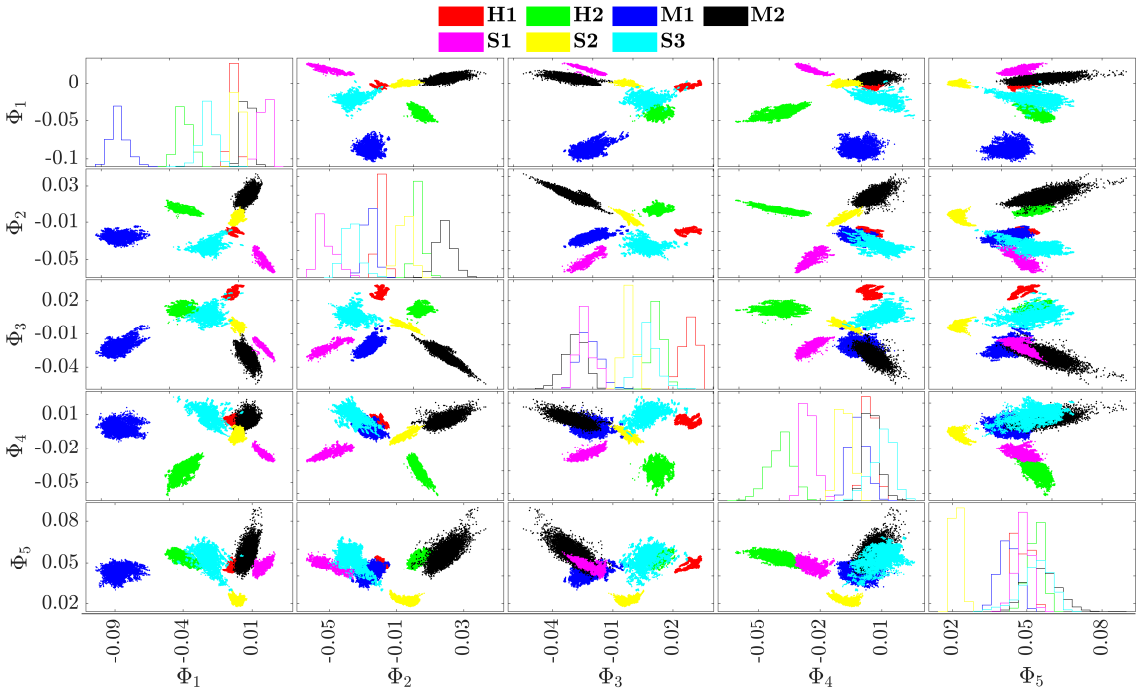
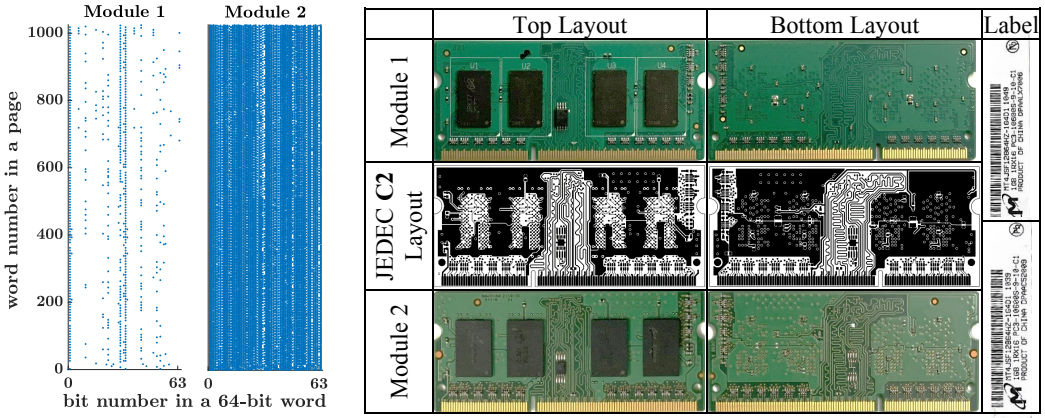


Fig. 6: Visualizing data in feature-space.

PPR_{neg}, which can be further improved by learning statistical model with more positive samples and/or introducing negative samples. Table II also presents that a small change in voltage and temperature has a very insignificant effect on classifier performance. This is expected because a small change in voltage or temperature has a negligible effect on *Activation* time. [13], [14], [33], [56].

A suspicious DRAM module: From the results shown in Table 3, for class 1, we observe that the PPR_{pos} is 0% (the ideal PPR_{pos} is 100%). Note that we have two samples available for this class: one is used for training, and another one is for testing. Therefore, we suspect that one of them is counterfeit. Fig. 7a presents the spatial locality of failed bits from 2 random pages from those two samples. The results show that they have distinct FBC properties. Furthermore, the dissimilarities found in visual inspection (Fig. 7b) suggests that one of them might be counterfeit (i.e., from a fake manufacturer). The layout difference between these two modules suggests that the reference layout version should be different for these two modules. However, the reference layout version is described as 'C1' on both modules' label. From the SPD data, we have found that the reference raw card (i.e., layout) version is specified as 'C' (which represents- 'C0', 'C1', 'C2' etc.) for both modules. For further investigation, we have checked the layout provided by the JEDEC [40] and found that the second module layout version is 'C2' instead of 'C1' (as shown in Fig. 7b). Therefore, we conclude that the second module is either from the fake manufacturer or mislabeled (with layout version 'C1').



(a) (b)

Fig. 7: (a) Spatial locality of erroneous bits of 2 random pages of each sample from Class 1, (b) Visual appearance of each sample from Class 1 (Module 2 is suspicious).

Class Tag	Operating Condition	$PPR_{pos,\mu}$	$PPR_{pos,\sigma}$	$PPR_{pos,min}$	$PPR_{neg,\mu}$	$PPR_{neg,\sigma}$	$PPR_{neg,max}$
1	NVRT	0.00	0.00	0.00	0.04	0.26	1.77
	HVRT	0.00	0.00	0.00	0.03	0.20	1.37
	LVRT	0.00	0.00	0.00	0.03	0.22	1.51
	NVHT	0.00	0.00	0.00	0.05	0.31	2.08
2	NVRT	99.07	1.16	95.40	0.00	0.02	0.17
	HVRT	98.04	3.54	87.60	0.00	0.01	0.10
	LVRT	99.09	0.98	96.49	0.00	0.02	0.19
	NVHT	99.33	0.58	97.53	0.00	0.01	0.07
3*	NVRT	–	–	–	0.00	0.01	0.06
	HVRT	–	–	–	0.00	0.01	0.06
	LVRT	–	–	–	0.00	0.01	0.05
	NVHT	–	–	–	0.00	0.01	0.07
4*	NVRT	–	–	–	0.00	0.00	0.00
	HVRT	–	–	–	0.00	0.00	0.00
	LVRT	–	–	–	0.00	0.00	0.00
	NVHT	–	–	–	0.00	0.00	0.00
5	NVRT	84.19	6.92	59.79	0.38	1.02	6.8
	HVRT	84.74	7.16	59.31	0.41	1.10	7.19
	LVRT	84.19	6.06	60.92	0.40	1.02	6.7
	NVHT	82.56	7.63	57.00	0.38	1.03	6.96
6	NVRT	90.29	9.31	69.58	1.58	4.16	25.91
	HVRT	90.35	9.39	68.98	1.54	4.14	25.78
	LVRT	81.98	9.49	69.74	1.69	4.52	28.65
	NVHT	90.32	9.31	69.17	1.52	3.80	23.64
7	NVRT	73.81	10.46	66.41	1.56	4.73	21.72
	HVRT	74.22	9.36	68.11	1.61	4.91	22.98
	LVRT	71.55	14.67	61.18	1.56	4.75	21.19
	NVHT	76.01	6.53	71.39	1.53	4.72	22.15

*For class 3 and class 4, we have only one sample which is used to train the statistical model. There is no positive test sample left for these two classes.

TABLE II: Results from the one-class classifier.

VI. Limitation and Future Work

Our proposed method can be used to detect a wide variety of counterfeit DRAM chips such as remarked, forged documented, cloned (and fabricated in a different plant), defective, out-of-spec, tampered, etc. However, the proposed work cannot identify overproduction [58].

In the future, we aim to explore other non-standard memory operations [28], [30], [31], [33], [57] as well to extract more robust features and entropy for better accuracy of our technique. Also, we will extend our technique for other volatile and non-volatile memory chips (e.g., flash memory, SRAM, etc. [60], [61]).

VII. Conclusion

In this paper, we proposed a simple non-invasive and low-cost scheme for identifying the origin of a DRAM manufacturer and verifying individual DRAM's specification. The proposed method exploits the DRAM latency variations to capture the architectural, layout, and process variations. At first, we chose the most appropriate features from the DRAM signature, and then we used a one-class classifier to verify the memory class without knowing the information from other classes (i.e., other manufacturers).

Since our current work only exploits the learning ability of a one-class classifier, we will explore additional machine learning techniques, such as emsemble learning and classifier fusion, for improved generalization across a broader set of manufacturers in future work. We will also explore additional features and filtering, wrapper, and embedded feature analysis techniques to better understand the impact of each feature on identifying a DRAM manufacturer.

Acknowledgment

This work was supported by the National Science Foundation under Grant Number CNS-1850241. We would like to thank UAH for filing patent, UAH reference: UAH-P-18038.

References

[1] U. Guin et al., "Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain," in Proceedings of the IEEE, vol. 102, no. 8, pp. 1207-1228, 2014.

[2] D. J. Forte and R. S. Chakraborty, "Counterfeit Integrated Circuits: Threats, Detection, and Avoidance," CHES 2018.

[3] "Counterfeit DRAM chip ring found," Available: https://www.cnet.com /news/counterfeit-dram-chip-ring-found [Accessed: 08-Aug-2019]

[4] "Counterfeit Product Alert," Available: https://thecounterfeitreport.com/product/483/Micron-20Elpida-DRAM-Memory-Chips.html [Accessed: 08-Aug-2019]

[5] U. Guin, N. Asadizanjani, and M. Tehranipoor, "Standards for Hardware Security," GetMobile: Mobile Comp. and Comm. 23, 1 (2019), 5-9.

[6] J. B. Wendt, F. Koushanfar, and M. Potkonjak, "Techniques for Foundry Identification," In Proceedings of the 51st Annual Design Automation Conference (DAC '14), ACM, Article 208, 6 pages.

[7] A. Ahmadi et al., "A machine learning approach to fab-of-origin attestation," In Proceedings of the 35th International Conference on Computer-Aided Design (p. 92).

[8] R. L. Helinski et al., "Electronic forensic techniques for manufacturer attribution," In Hardware Oriented Security and Trust (HOST), 2016 IEEE International Symposium on (pp. 139-144).

[9] Z. B. Aweke et al., "ANVIL: Software-based protection against next-generation rowhammer attacks," ACM SIGPLAN Notices, vol. 51, no. 4, pp. 743−755, 2016.

[10] A. Schaller et al., "Intrinsic Rowhammer PUFs: Leveraging the Rowhammer effect for improved security," 2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), 2017.

[11] M. Lanteigne, "How Rowhammer Could Be Used to Exploit Weaknesses in Computer Hardware," Third I/O Inc., Mar. 2016, Available: www.thirdio.com/rowhammer.pdf [Accessed: 08-Aug-2019]

[12] Z. Guo et al., "SCARe: An SRAM-Based Countermeasure Against IC Recycling," IEEE Transactions on Very Large Scale Integration (VLSI) Systems 26, no. 4 (2018): 744-755.

[13] K. K. Chang et al., "Understanding latency variation in modern DRAM chips: Experimental characterization, analysis, and optimization," ACM SIGMETRICS Performance Evaluation Review (Vol. 44, No. 1, pp. 323−336), June, 2016

[14] J. S. Kim et al., "The DRAM Latency PUF: Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Trade-off in Modern DRAM Devices",24th International Symposium on High-Performance Computer Architecture, 2018.

[15] Talukder, BMS Bahar, Biswajit Ray, Domenic Forte, and Md Tauhidur Rahman. "PreLatPUF: Exploiting DRAM Latency Variations for Generating Robust Device Signatures." IEEE Access 7 (2019): 81106-81120.

[16] "DRAM chip market share by manufacturer worldwide from 2011 to 2019." May, 2019. Available: https://www.statista.com/statistics/271726/global-market-share-held-by-dram-chip-vendors-since-2010.

[17] JEDEEC, "DDR3 SDRAM Standard," July 2012

[18] B. Liu and G. Qu, "VLSI supply chain security risks and mitigation techniques: A survey, Integration, the VLSI Journal, pp. 1−10, 2016.

[19] A. Basak and S. Bhunia, "P-Val: Antifuse-Based Package-Level Defense Against Counterfeit ICs," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 35, no. 7, pp. 1067−1078, 2016.

[20] X. Zhang and M. Tehranipoor, "Design of On-Chip Lightweight Sensors for Effective Detection of Recycled ICs," IEEE Transactions on Very Large Scale Integration Syst., vol. 22, no. 5, pp. 1016−1029, 2014.

[21] K. He, X. Huang, and S. X.-D. Tan, "EM-based on-chip aging sensor for detection and prevention of counterfeit and recycled ICs," IEEE/ACM International Conference on Computer-Aided Design, 2015.

[22] K. Huang, J. M. Carulli, and Y. Makris, "Counterfeit electronics: A rising threat in the semiconductor manufacturing industry," IEEE Int. Test Conference, 2013.

[23] Y. M. Alkabani and F. Koushanfar, "Active Hardware Metering for Intellectual Property Protection and Security", 16th {USENIX} Security Symposium, Boston, MA, 2007.

[24] X. Zhang, N. Tuzzio, and M. Tehranipoor, "Identification of recovered ICs using fingerprints from a light-weight on-chip sensor," Proceedings of the 49th Annual Design Automation Conference on - DAC 12, 2012.

[25] J. A. Hayward and J. Meraglia, "DNA Marking and Authentication: A unique, secure anti-counterfeiting program for the electronics industry," International Symposium on Microelectronics, vol. 2011, no. 1, pp. 000107−000112, 2011.

[26] K. Elkhiyaoui, E.-O. Blass, and R. Molva, "CHECKER: On-site checking in RFID-based supply chains." In Proceedings of the fifth ACM conference on security and privacy in wireless and mobile networks, pp. 173−184. ACM, 2012.

[27] M. N. Islam, V. C. Patii, and S. Kundu, "On IC traceability via blockchain," 2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), 2018.

[28] Sutar, S. et al., "D-PUF: An Intrinsically Reconfigurable DRAM PUF for Device Authentication and Random Number Generation," ACM Trans. Embed. Comput. Syst. 17, 1, Article 17, December 2017.

[29] F. Tehranipoor et al., "DRAM based Intrinsic Physical Unclonable Functions for System Level Security," Proceedings of the 25th edition on Great Lakes Symposium on VLSI - GLSVLSI 15, 2015.

[30] W. Xiong et al. , "Run-Time Accessible DRAM PUFs in Commodity Devices," Lecture Notes in Computer Science Cryptographic Hardware and Embedded Systems − CHES 2016, pp. 432453, 2016.

[31] M. S. Hashemian et al., "A Robust Authentication Methodology using Physically Unclonable Functions in DRAM Arrays," Design, Automation & Test in Europe Conference & Exhibition, 2015.

[32] JEDEC, "Serial Presence Detect (SPD), General Standard," Available: https://www.jedec.org/sites/default/files/docs/4_01_02R19.pdf.

[33] K. K. Chang et al., "Understanding reduced-voltage operation in modern dram devices: Experimental characterization, analysis, and mechanisms," Proceedings of the ACM on Measurement and Analysis of Computing Systems, 1(1), 10.

[34] B. Jacob, S. W. Ng, and D. T. Wang, "Memory systems: cache, DRAM, disk," Morgan Kaufmann, 2010.

[35] D. Clein, "CMOS IC layout: concepts, methodologies and tools," Boston: Newnes, 2000.

[36] Y. Cao et al., "Design sensitivities to variability: extrapolations and assessments in nanometer VLSI," 15th Annual IEEE International ASIC/SOC Conference, 2002.

[37] K. J. Kuhn et al., "Process Technology Variation," IEEE Transactions on Electron Devices, vol. 58, no. 8, pp. 21972208, 2011.

[38] SK hynix, "DDR SDRAM MODULE PART NUMBERING," Jun-2014. [Online]. Available: https://www.skhynix.com/static/filedata/fileDownload.do?seq=190, [Accessed: 08-Aug-2019].

[39] SK hynix, "Technical Support," [Online]. Available: https://www.skhynix.com/eng/support/technicalSupport.jsp, [Accessed: 08-Aug-2019].

[40] JEDEC, "Design Files for ddr3," Available: https://www.jedec.org/standards-documents/focus/memory-module-designs-dimms/ddr3/all.

[41] Deutsch & Gailly. ZLIB Compressed Data Format Specification. Available: http://www.ietf.org/rfc/rfc1950.txt [Accessed: 08-Aug-2019].

[42] K. Huang et al., "Recycled IC Detection Based on Statistical Methods, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 6, pp. 947960, 2015.

[43] O. Sinanoglu et al., "Reconciling the IC test and security dichotomy, 2013 18Th Ieee European Test Symposium (Ets), 2013.

[44] K. Huang, J. M. Carulli, and Y. Makris, "Parametric counterfeit IC detection via Support Vector Machines, IEEE Int. Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, 2012.

[45] W.-C. Chang, C.-P. Lee, and C.-J. Lin, "A Revisit to Support Vector Data Description (SVDD)," Technical report 2013.

[46] D. M.J. Tax, and R. P.W. Duin, "Support Vector Data Description, Machine Learning, vol. 54, no. 1, pp. 45−66, 2004.

[47] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett,"New support vector algorithms," Neural Computation, 12, 2000, 1207-1245

[48] E. Castillo, D. P.-Barral, B. G. Berdiñas and Oscar F.-Romero, "Distributed One-Class Support Vector Machine, Int. Journal of Neural Sys., vol. 25, no. 07, p. 1550029, 2015.

[49] S. S. Khan and M. G. Madden, "One-class classification: taxonomy of study and review of techniques, The Knowledge Engineering Review, vol. 29, no. 03, pp. 345−374, 2014.

[50] B. Schölkopf et al., "Comparing support vector machines with Gaussian kernels to radial basis function classifiers, IEEE Transactions on Signal Processing, V. 45, 1997

[51] D. M.J. Tax, and R. P.W. Duin, "Uniform Object Generation for Optimizing One-class Classifiers," Journal of Machine Learning Research 2, pp. 155−173, 2001.

[52] H. Hassan et al., "SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies," IEEE Int. Symp. on High Performance Computer Architecture, 2017, pp. 241-252.

[53] Texas Instruments, "Complete DDR, DDR2 and DDR3 Memory Power Solution Synchronous Buck Controller, 3-A LDO, Buffered Reference for Embedded Computing Systems," Nov. 2010, Available: http://www.ti.com/lit/ds/symlink/tps59116.pdf, [Accessed: 08-Aug-2019]

[54] J. P. Cunningham, Z Ghahramani, "Linear Dimensionality Reduction: Survey, Insights, and Generalizations", Journal of Machine Learning Research 16 (2015)

[55] C.-C. Chang and C.-J. Lin, "LIBSVM : a library for support vector machines," ACM Transactions on Intelligent Sys. and Tech., 2:27:1–27:27, 2011.

[56] K. Chandrasekar et al.,"Exploiting Expendable Process-Margins in DRAMs for Run-Time Performance Optimization," Design, Automation & Test in Europe Conference & Exhibition, 2014.

[57] Halderman, J. A. et al., "Lest we remember: cold-boot attacks on encryption keys," Communications of the ACM 52, no. 5 (2009): 91−98.

[58] M. T. Rahman et al., "CSST: Preventing distribution of unlicensed and rejected ICs by untrusted foundry and assembly." In 2014 IEEE International symposium on defect and fault tolerance in VLSI and nanotechnology systems (DFT), pp. 46-51. IEEE, 2014.
[59] Karimian, Nima et al., "Genetic algorithm for hardware Trojan detection with ring oscillator network (RON)." In 2015 IEEE International Symposium on Technologies for Homeland Security, pp. 1-6, 2015.
[60] Rahman, M. Tauhidur et al., "Systematic correlation and cell neighborhood analysis of sram puf for robust and unique key generation." Journal of Hardware and Systems Security 1, no. 2 (2017): 137-155.
[61] Kumari, Preeti et al., "Independent detection of recycled flash memory: Challenges and solutions." In 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 89-95. IEEE, 2018.