An Innovative Data Mining Technique for Automatic Anomaly Detection in Physical Unclonable Functions

Mohammad Reza Heidari Iman, Sergio Vinagrero Gutierrez, Ioana Vatajelu, Giorgio Di Natale

▶ **To cite this version:**

Mohammad Reza Heidari Iman, Sergio Vinagrero Gutierrez, Ioana Vatajelu, Giorgio Di Natale. An Innovative Data Mining Technique for Automatic Anomaly Detection in Physical Unclonable Functions. IEEE 28th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS 2025), May 2025, Lyon, France. ⟨10.1109/DDECS63720.2025.11006802⟩. ⟨hal-05085384⟩

HAL Id: hal-05085384

https://hal.science/hal-05085384v1

Submitted on 31 Jul 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons CC BY-NC 4.0 - Attribution - Non-commercial use - International License

An Innovative Data Mining Technique for Automatic Anomaly Detection in Physical Unclonable Functions

Mohammad Reza Heidari Iman[1], Sergio Vinagrero Gutierrez[2], Elena-Ioana Vatajelu[1], and Giorgio Di Natale[1]
[1]Univ. Grenoble Alpes, CNRS, Grenoble INP*, TIMA, 38000 Grenoble, France
[2]Institut des Nanotechnologies de Lyon (INL)
{mohammadreza.heidari-iman, ioana.vatajelu, giorgio.di-natale}@univ-grenoble-alpes.fr
sergio.vinagrero@ec-lyon.fr

***Abstract*—Physical Unclonable Functions (PUFs) offer a promising alternative to conventional cryptographic techniques to secure sensitive information in modern circuits. PUFs leverage inherent process variability to dynamically generate unique secrets, eliminating the need for data storage. However, a significant challenge in PUF-based security is differentiating between valid PUFs and those that may have been tampered with or are invalid (*i.e.*, not belonging to the original design). This paper presents an innovative data mining-based technique for detecting anomalies and identifying tampered or invalid PUFs. The proposed method extracts a set of rules that describe the expected behavior of the PUF, where deviations from these rules indicate potential security issues and vulnerabilities. Experimental results demonstrate that the method effectively detects invalid or tampered PUFs, highlighting its potential to strengthen PUF-based security systems.**

***Index Terms*—physical unclonable functions, PUF security, pattern detection, anomaly detection, data mining in PUF**

I. Introduction

The rise of the Internet of Things (IoT) is driving the deployment of hundreds of interconnected devices that require secure identification, typically managed through centralized servers. Physical unclonable functions present a significant alternative to conventional cryptographic systems based on non-volatile memory (NVM). By harnessing intrinsic manufacturing variations, PUFs produce unique, unclonable responses that eliminate the need for storing sensitive information in memory, thereby reducing the risk of data leakage.

PUFs have a wide range of applications, from securing cryptographic keys and hardware authentication to anti-counterfeiting and tamper-proofing mechanisms. Unlike traditional approaches that rely on storing sensitive data in NVM, PUFs inherently generate data on demand, minimizing the attack surface and enhancing security. This makes PUFs particularly valuable in resource-constrained IoT environments, where lightweight and energy-efficient security solutions are critical.

In mathematical terms, a PUF is a function characterized by random properties, which links an input (referred to as a *Challenge*) to an output (*i.e.*, a *Response*). When various devices are given an identical challenge, they generate responses in a random and unique way. This property makes them ideal to generate unique keys that can be used on communication protocols. PUF-based protocol implementations often rely on centralized databases storing Challenge-Response Pairs (CRPs), enabling secure validation during authentication for device identification, resulting in massive storage requirements [1]. To address these storage challenges, some approaches propose compact PUF models [1, 2] as a substitute for CRP storage by leveraging patterns across responses, analogous to compression algorithms.

This paper introduces a novel method to leverage data mining techniques for PUF traceability and lifecycle management, providing an in-depth explanation and extended discussion of the method, which was briefly introduced in [3]. By examining patterns across multiple PUF designs, architectures, and foundries, we propose methods to verify device legitimacy and detect tampering by comparing real-time responses with expected patterns. These approaches work because in reality there is no fully ideal PUF. Some CRPs are slightly biased or may present inter-correlation, and these approaches are able to identify the unique properties generated by the bias. We argue that these patterns are unique per PUF architecture, design and foundry, enabling the unique identification of various PUF designs in a system. Furthermore, periodic monitoring can be employed to flag devices nearing the end of their operational life or showing anomalous behavior, enhancing both security and lifecycle management in IoT networks relying on PUFs.

In summary, the contributions of this paper are as follows:

- An innovative data mining-based algorithm, specifically designed for the context of PUF security, is proposed. To the best of our knowledge, this is the first data mining-based approach for effectively detecting tampered and invalid PUFs.
- The proposed method is highly scalable, efficient, and lightweight, as the dataset used to mine the rules can be a small fraction of the real system.

The paper is organized as follows: The state-of-the-art is presented in Section II, and Section III presents the preliminaries. Section IV outlines the technical details of the proposed method. Section V presents the experimental results, and Section VI concludes the paper.

II. State of the art

Mathematical models describing PUF behavior are crucial for validating their security claims, serving a role as foundational as that of mathematical abstractions in other

* Institut National Polytechnique Grenoble Alpes

cryptographic functions and security primitives. These abstractions not only provide a foundation for rigorous security assessments but also enable meaningful comparisons between different PUF designs. Since their introduction, numerous evaluation frameworks have been proposed and refined to characterize the behavior and performance of PUFs. Among these, the metrics introduced by Maiti et al. [4] have remained the canonical ones for assessing PUF performance. These metrics, which include Uniformity, Bit-aliasing, Inter-Hamming Distance (commonly referred to as Uniqueness), and Intra-Hamming Distance (Reliability), are particularly significant in evaluating a PUF's randomness and temporal behavior.

We bring attention specially to Uniformity, Bit-aliasing and Uniqueness. These metrics are crucial as they measure the randomness of PUFs and they are the ones that can reveal any bias, which can impact their security. A brief description of these metrics and their computation is provided. In this context, $r_{i,l}$ is the l-th binary bit of an n-bit response from a chip i and k represents the number of devices.

Uniformity (equation 1) measures the ratio of 1s and 0s across all responses for each PUF instance, where deviations from an even distribution can compromise security by enabling attackers to predict responses, given they have access to a subset of the responses.

$$\text{Uniformity}_i = \frac{1}{n}\sum_{l=1}^{n} r_{i,l} \times 100\% \tag{1}$$

Bit-aliasing (equation 2) measures the ratio of 1s and 0s across responses for each challenge. Moreover, an uneven distribution decreases the Entropy of that challenge, which reduces the ability of the PUF of uniquely identifying devices.

$$\text{Bit-Aliasing}_l = \frac{1}{k}\sum_{i=1}^{k} r_{i,l} \times 100\% \tag{2}$$

Lastly, Uniqueness (equation 3), assesses the PUF's capability to distinguish between different devices by calculating the average Hamming distance between responses from distinct instances.

$$\text{Uniqueness} = \frac{2}{k(k-1)}\sum_{i=1}^{k-1}\sum_{j=i+1}^{k}\frac{HD(R_i,R_j)}{n} \times 100\% \tag{3}$$

While numerous proposals can be found in the literature, they provide only a partial view of PUF behavior and do not address inter-dependencies or correlations within responses that could signal potential vulnerabilities. Understanding spatial and temporal correlations among PUF responses is a critical aspect of security analysis, as such correlations can lower the system's Entropy and can reveal vulnerabilities that attackers could exploit.

Willsch et al. [5] introduced statistical tests to detect spatial correlations, emphasizing how dependencies within PUF responses could undermine Uniqueness and randomness. Similarly, the work presented in [6] explored the relationship between Bit-aliasing and Uniqueness, proposing correlation-based metrics that effectively capture subtle dependencies often overlooked by the canonical metrics. Building on this, Wilde et al. [7] conducted a comprehensive analysis of spatial correlations in various architectures, providing frameworks to assess and mitigate these vulnerabilities. While the authors provided a good basis for understanding these types of phenomena, they did not extensively address the causes or consequences of these correlations on the security of the system. Arul et al. [8] focused on spatial auto-correlation in memory-based PUFs, highlighting the impact of intrinsic dependencies on security properties and underscoring the importance of careful design to minimize these effects. One of the recent works shown in [9] examined heavily biased instances in Arbiter PUFs and their impact on Uniqueness, advocating for correlation-based metrics that are more sensitive to these issues. The authors in [10] synthesized findings from multiple studies on spatial correlations in weak PUFs, proposing methods to quantify and address these dependencies comprehensively. Their work underscored the critical role of spatial correlation in evaluating PUF security. Furthermore, the authors in [11] provided a novel framework to identify PUF security weaknesses by leveraging correlation spectra in Boolean functions.

The ability to detect patterns and correlations among PUF responses enables researchers not only to analyze PUF performance comprehensively but also to gain a deeper understanding of their vulnerabilities. This work takes these efforts a step further by leveraging response correlations to identify PUF instances originating from different architectures, designs, and foundries. We propose that such patterns are inherently unique to specific PUF designs, and their analysis could enable the creation of systems capable of identifying different PUF designs in the field across their lifecycle.

In addition to randomness analysis, the canonical metrics play a role in estimating the learnability of PUFs, providing insights into the difficulty of modeling specific designs. However, these metrics alone are insufficient to evaluate the resilience of PUFs against the wide array of state-of-the-art modeling techniques, as some techniques can exploit weaknesses in certain architectures. Research efforts have largely focused on developing advanced machine learning (ML)-based models and countermeasures against ML-based attacks, but cloning attacks targeting weak PUFs have also emerged as an effective threat. These attacks aim to replicate the unique characteristics of PUFs to produce duplicates that generate identical responses.

A variety of promising cloning techniques, ranging from non-invasive to fully invasive, have been documented in the literature. These techniques exploit physical phenomena to effectively clone PUFs, with notable success observed in memory- and RO-based PUF designs. In the context of Ring Oscillator-based PUFs, [12] demonstrated that targeted ageing can be used to clone the behaviour and responses of RO-PUFs. Similarly, [13] utilized BTI-ageing techniques to target SRAM cells, successfully duplicating PUF responses. These attacks are particularly concerning because the circuitry used to evaluate PUFs is often left unsecured. The authors in [14, 15] employed high-resolution imaging to capture the unique power-up states of SRAM cells, replicating their physical structure to duplicate response characteristics. Several countermeasures to hinder cloning and side-channel attacks have been proposed in the literature, with notable examples such as the works in [16, 17].

Countermeasures to physical cloning attacks have been proposed in the literature, such as tamper-sensitive optical

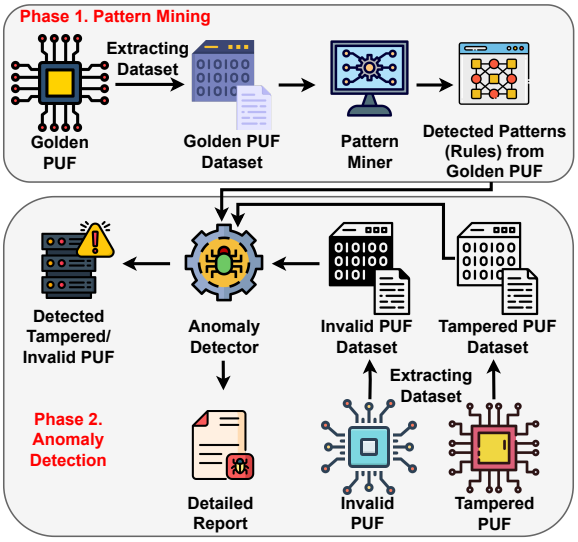

Figure 1: General Flow of the Proposed Method

PUFs [18], which leverage structural changes to signal tampering attempts, and resistive preformed resistive random access memory (ReRAM)-based PUFs [19], which leverage fabrication variability and classical physical tampering detection mechanisms to enhance security. However, PUFs are not immune to sophisticated invasive techniques, as briefly demonstrated by Helfmeier et al. [20]. These findings underscore the need for robust systems capable of detecting and mitigating such attacks. Indeed, side-channel and cloning attacks remain persistent challenges in the PUF domain and present a substantial threat to the integrity of PUFs, underscoring the need for continued innovation in securing PUF-based systems.

The challenge of detecting tampered or modeled PUF responses remains largely unexplored. This paper tries to address this gap by proposing a novel method to identify tampered PUFs through statistical discrepancies and correlation among responses. By addressing this critical gap, this approach contributes to a more comprehensive defense strategy for PUF-based systems and paves the way for future research to develop techniques that differentiate between model-generated and real-device responses.

III. Preliminaries

In this section, we briefly explain the data mining concepts and definitions that have been adapted for use in PUF security in this paper.

Definition 1: ***Frequent itemsets*** refer to a set of response bits in a CRP dataset that occur with a frequency, either across devices or across challenges, indicating significant relations and associations between the response bits.

Definition 2: An ***Association Rule (AR)*** is defined as an implication of the form $\mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{X}, \mathcal{Y} \subseteq \mathcal{I}$, with $\mathcal{X} \cap \mathcal{Y} = \emptyset$, and $\mathcal{I}$ is a set of items [21–23]. $\mathcal{X}$ and $\mathcal{Y}$ are referred to as *frequent itemsets*. In this paper, an *association rule* represents the correlations between the responses in the CRP dataset, either across devices or across challenges.

Definition 3: ***Support*** is a metric in data mining that indicates how frequently an itemset appears in the CRP dataset [24, 25]. This metric takes a value between 0 and 1. For the rule $\mathcal{X} \rightarrow \mathcal{Y}$, the value of support is calculated using the following formula [24]:

$$Supp(\mathcal{X} \rightarrow \mathcal{Y}) = P(\mathcal{X} \cup \mathcal{Y}) \tag{4}$$

In (4), $P(\mathcal{X} \cup \mathcal{Y})$ represents the probability associated with $\mathcal{X} \cup \mathcal{Y}$, where $\mathcal{X} \cup \mathcal{Y}$ indicates that a record contains both $\mathcal{X}$ and $\mathcal{Y}$, that is, the union of itemsets $\mathcal{X}$ and $\mathcal{Y}$.

Definition 4: The **min_supp** value is the threshold and a minimum value for *support* to decide whether an itemset is frequent (*i.e.,* occurs frequently in the CRP dataset) or not [24]. If the frequency of the itemset is more than this threshold, the itemset is considered a frequent itemset [21]. A higher value of min_supp leads to generating commonly occurring association rules, while a lower value of it leads to generating rarely occurring and correlated ARs [25, 26].

Definition 5: ***Confidence*** is a metric that indicates how often a rule is found to be true [22]. For the rule $\mathcal{X} \rightarrow \mathcal{Y}$, this value is calculated using the following formula [22, 24]:

$$Conf(\mathcal{X} \rightarrow \mathcal{Y}) = P(\mathcal{Y}|\mathcal{X}) \tag{5}$$

It evaluates the degree of certainty for the detected association rule. This is defined as the conditional probability $P(\mathcal{Y}|\mathcal{X})$, which represents the probability that a record containing $\mathcal{X}$ also contains $\mathcal{Y}$. The value of confidence ranges between 0 and 1.

Definition 6: The **min_conf** is the minimum threshold for *confidence* [24]. A higher value of min_conf results in fewer but more correlated association rules [24, 25].

IV. Proposed Methodology

Figure 1 illustrates the general flow of the proposed method. It consists of two main phases: *Pattern Mining* and *Anomaly Detection*. Initially, in the *Pattern Mining* phase, the behavior of the PUF is studied through the golden CRP dataset extracted during enrollment. Ultimately, in the second phase, the mined association rules will be used to flag tampered or invalid instances. In the *Pattern Mining* phase, the proposed pattern miner extracts the set of rules (patterns) from the CRP dataset. In the *Anomaly Detection* phase, new instances are tested against the mined rules from the previous phase, and differences in the number of matching rules are flagged as potentially tampered or invalid instances, accompanied by a detailed report on the detected PUFs. In the following subsections, each phase of the method is discussed in more detail.

A. Pattern Mining

During this phase our proposed algorithm leverages techniques from data mining and association rule mining to extract patterns from the CRP dataset at enrollment. Our algorithm operates in three main steps: Initialization and Preprocessing, Rule Mining, and Time Labeling. After preprocessing the dataset, during the rule mining step, a set of association rules in the general form of $antecedent \rightarrow consequent$ is mined. In the last step of the algorithm (Time Labeling), our proposed algorithm is employed to mine a set of association rules in the forms of $next[\mathcal{N}] = antecedent \rightarrow next[\mathcal{N}]consequent$, $before[\mathcal{N}] = antecedent \rightarrow before[\mathcal{N}]consequent$ and $always = antecedent \rightarrow (always)consequent$. The combined use of $next[\mathcal{N}]$ and $before[\mathcal{N}]$ rules reveals correlations between responses, while $always$ rules identify con-

Algorithm 1: Proposed Pattern Miner

```
1  Input: 𝒩, 𝒟𝒮, min_supp
2  Output: next[𝒩] = antecedent → next[𝒩]consequent,
     before[𝒩] = antecedent → before[𝒩]consequent
     /* Initialization and Preprocessing                    */
3  ℛ = antecedent → consequent
4  L_1 = {frequent 1−itemsets ∈ 𝒟𝒮′}
5  K = 2
6  forall f ∈ 𝒟𝒮 do
7      𝒟𝒮′ = MoveUp(f(𝒩))
     /* Rule Mining                                         */
8  while L_{k−1} != ∅ do
9      C_k = generate_candidate_itemsets(L_{k−1})
10     L_k = prune_infrequent_itemsets(C_k, min_supp)
11     k = k+1
12 foreach frequent itemset L_i ∈ L do
13     foreach subset S of L_i do
14         if (S != ∅) && (S != L_i) then
15             confidence = support(L_i) / support(S)
16             if confidence >= min_conf then
17                 ℛ ← association_rule(S => L_i)
18 return ℛ
     /* Time Labeling                                       */
19 if (𝒩 == 0) and (ℛ.antecedent == (t ∈ 𝒟𝒮′)) and
     (ℛ.consequent == (f ∈ 𝒟𝒮′)) then
20     always ← label(ℛ)
21 else if (𝒩 > 0) and (ℛ.antecedent == (t ∈ 𝒟𝒮′)) and
     (ℛ.consequent == (f ∈ 𝒟𝒮′)) then
22     next[𝒩] ← label(ℛ)
23 else if (𝒩 > 0) and (ℛ.antecedent == (f ∈ 𝒟𝒮′)) and
     (ℛ.consequent == (t ∈ 𝒟𝒮′)) then
24     before[𝒩] ← label(ℛ)
```

sistent behaviors analogous to autocorrelation. Together, these rules analyze response reliability and inter-correlation patterns, enabling the detection of tampered or invalid PUFs by flagging deviations from the expected behavior as potential security issues. Once extracted, the mined rules are stored in the server and the original dataset can be discarded, reducing the overall storage needs, at the trade-off of reducing the certainty of the validity of the devices.

Algorithm 1 outlines the steps of the pattern miner for extracting rules from the CRP dataset. Here $\mathcal{DS}$ represents the input dataset, $\mathcal{DS}'$ denotes the preprocessed dataset, a dataset row represents the CRPs of an instance and a step interval refers to the distance between rows. Furthermore, min_supp and min_conf represent the minimum support (Definition 4) and minimum confidence (Definition 6) values for the support (Definition 3) and confidence (Definition 5) metrics, respectively. The algorithm generates three types of rules in the forms $always$, $next[\mathcal{N}]$ and $before[\mathcal{N}]$, where $\mathcal{N}$ specifies the step interval used to determine correlations between a given row in the dataset and the row located $\mathcal{N}$ steps after or before it. The dataset columns are divided into two groups, target values, represented by t, and feature values, represented by f, and the pattern miner extracts correlations between these t and f values. In line 7 of the algorithm, the dataset is prepared for mining the $next[\mathcal{N}]$ and $before[\mathcal{N}]$ patterns by moving all feature values $\mathcal{N}$ records up from their original positions (MoveUp($f(\mathcal{N})$)), while the position of target values remains unchanged.

The resultant dataset after preprocessing is then fed into lines 8 to 18 of Algorithm 1 to extract association rules. Executing these lines of the algorithm on the preprocessed dataset yields a set of association rules in the form of $antecedent \rightarrow consequent$.

In lines 8 to 11 of Algorithm 1, frequent itemsets (Definition 1) of different sizes (1-itemsets, 2-itemsets, etc.) are extracted iteratively until the list of frequent itemsets becomes empty. Specifically, the algorithm mines frequent itemsets with support values (Definition 3) that exceed the min_supp threshold (Definition 4), while pruning those that do not. In line 9 of the algorithm, C_k represents the candidate itemsets of size k, generated by combining frequent (k-1)-itemsets. In line 10, L_k denotes the set of frequent k-itemsets. In this algorithm, 1-itemsets consist of individual bits of the CRP dataset, 2-itemsets are pairs of bits, etc.

After extracting the frequent itemsets and adding them to the L_k list, the association rules are generated from the list of frequent itemsets in lines 12 to 18 of the algorithm. Due to space constraints, we invite interested readers to explore [24] for an in-depth discussion on mining frequent itemsets and association rules. The association rules mined in the previous steps of Algorithm 1 are then forwarded to the Time Labeling step, where they produce temporal association rules—$always$, $next[\mathcal{N}]$, and $before[\mathcal{N}]$—which are crucial for the *Anomaly Detection* phase. Details of the *Anomaly Detection* phase are provided in the next subsection.

B. Anomaly Detection

During this phase, the *Anomaly Detector* employs the set of rules mined from the golden CRP dataset and a new set of CRPs, which may be invalid or tampered. The new CRPs are compared with either the $antecedent$ or $consequent$ of the rules that have been stored on the server. Any irregularity between the expected pattern of bit-flips (*i.e.*, reliability) defined by the number of matched rules given the new CRPs, is flagged as a potential security issue.

V. Experimental Results

We validated our proposed method using a dataset of CRPs obtained from 84 STM32L152RE microcontrollers manufactured by ST Microelectronics [27]. Each microcontroller has 80 kB of memory, corresponding to 655,360 bits (*i.e.*, PUF responses). Some of the memory regions have been omitted for the analysis since the stack and program memory are mostly filled with 0s in all devices, which can lead to reduced entropy, as showcased in the bottom part of the heatmap in Figure 4.

An initial assessment of the canonical PUF metrics Uniformity, Bit-aliasing and Uniqueness has been performed and their histograms are shown in Figure 5. As mentioned before, ideally, these metrics should be centered around 0.5, indicative of unbiased behavior. The results show that Uniformity values are consistently near 0.5, indicating well-distributed responses. However, Bit-aliasing exhibits a bimodal distribution, strongly suggesting biased responses across challenges. This bias appears to influence the Uniqueness metric, which has a mean value skewed

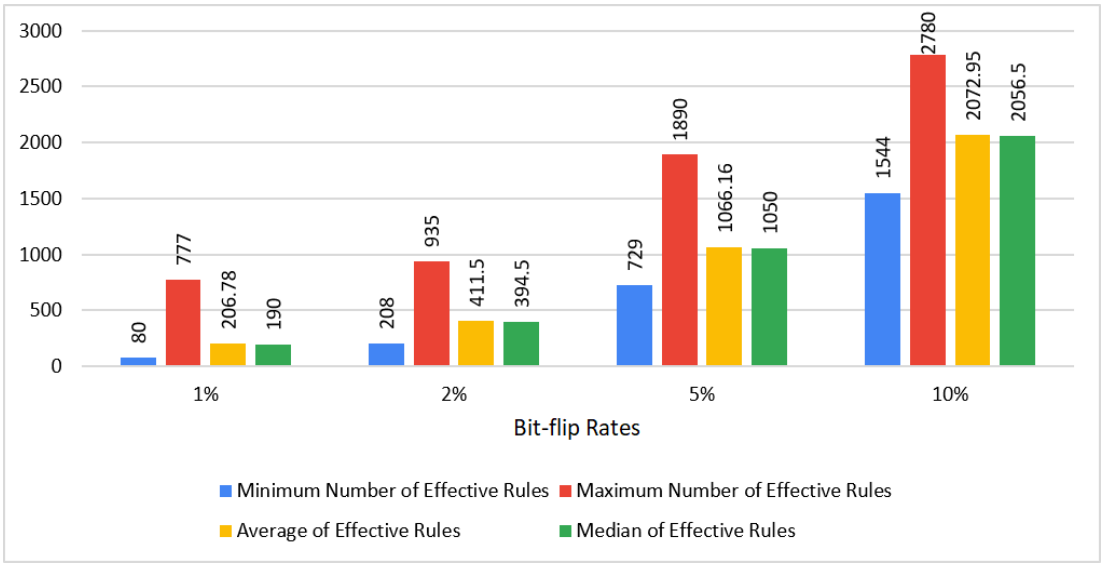

Figure 2: Fault Detection Results for Different Bit-flip Rates

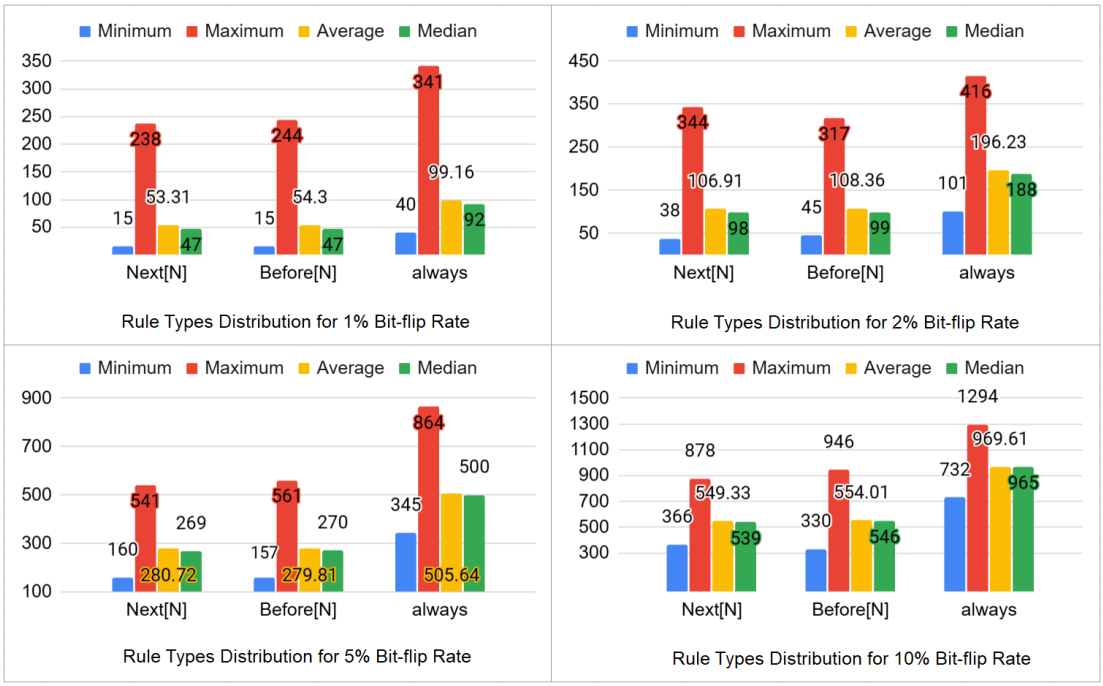

Figure 3: Rule Types Distribution for each Bit-flip Rate

toward 0.4 rather than the ideal 0.5. The mined rules from our method aim to capture and extract the underlying dependencies among these biased responses.

For this study, a total of 16000 rules were mined in approximately 15 minutes using a low-end computer, demonstrating that the proposed method is efficient and not resource-intensive. To simulate the behavior of tampered PUFs, new instances were generated by uniformly flipping responses in the CRP dataset at rates of 1%, 2%, 5%, and 10% over 1000 iterations.

Figure 2, which summarizes the experimental results, presents the minimum and maximum number of effective rules capable of detecting faults for each flipping rate, along with the calculated average and median values of these rules over 1000 iterations of bit-flipping. The calculated averages and medians for each bit-flip rate are closely aligned, which indicates that the proposed detection mechanism is consistent, robust, and scalable across all tested bit-flip rates, without being significantly impacted by outliers.

Figure 3 presents the distribution of different types of mined rules for each bit-flip rate, alongside their minimum, maximum, median, and average number of mined rules. The average and median values for the $next[\mathcal{N}]$ and $before[\mathcal{N}]$ rules are very close across all bit-flip rates, as the combination of these rules accurately measures the interdependence between responses, which serves as the

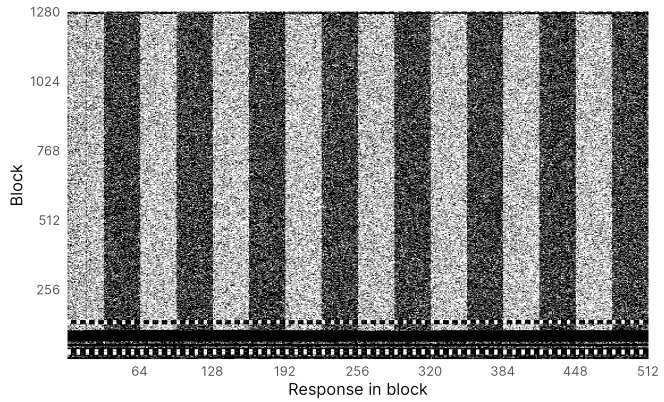

Figure 4: Heatmap of the SRAM bits across the 84 devices

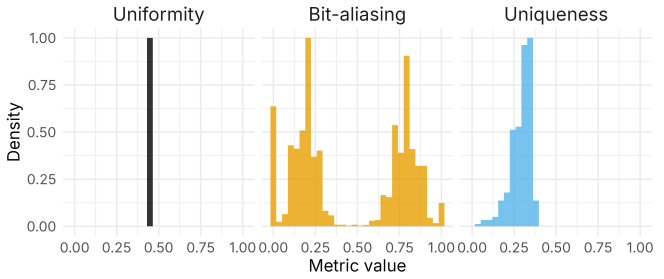

Figure 5: Histograms of Uniformity, Bit-aliasing and Uniqueness for the SRAM-PUF

reference to discriminate new instances.

VI. Conclusion

In this paper, we proposed a data mining-based method to extract rules that describe the behavior of a PUF regarding the dependence and correlation between responses, which are valuable for detecting tampered or invalid instances. To detect tampered and invalid PUFs, the proposed method automatically mined association rules in the forms of $always$, $next[\mathcal{N}]$, and $before[\mathcal{N}]$. Experimental results from 84 SRAM-based PUFs showed that the method effectively leverages the correlation among responses to correctly identify tampered and invalid instances with high accuracy using the mined rules. Nevertheless, the proposed method should be examined on different PUF datasets to comprehensively assess its accuracy and performance across various datasets and scenarios. Furthermore, the work could be extended to assess the effectiveness of detecting CRPs generated with state-of-the-art models or statistical techniques.

Acknowledgement

This work is partially funded by the “Resilient Trust” and “Neuropuls” projects of the EU’s Horizon Europe research and innovation programme under grant agreements No. 101112282 and No. 101070238.

References

[1] A. Ali-Pour, D. Hely, V. Beroulle *et al.*, “Elaborating on sub-space modeling as an enrollment solution for strong puf,” in *2022 18th Intl. Conf. on DCOSS*. IEEE, 2022, pp. 394–399.

[2] ——, “Strong puf enrollment with machine learning: A methodical approach,” *Electronics*, vol. 11, no. 4, p. 653, 2022.

[3] M. R. Heidari Iman, S. Vinagrero Gutierrez, E.-I. Vatajelu *et al.*, “Late breaking results: Automatic anomaly detection method in physical unclonable functions using data mining techniques,” in *2025 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2025, pp. 1–2.

[4] A. Maiti, J. Casarona, L. McHale *et al.*, “A large scale characterization of ro-puf,” in *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*. IEEE, pp. 94–99.

[5] B. Willsch, J. Hauser, S. Dreiner *et al.*, “Statistical tests to determine spatial correlations in the response behavior of puf,” in *2016 12th Conference on Ph. D. Research in Microelectronics and Electronics (PRIME)*. IEEE, pp. 1–4.

[6] L. Feiten, M. Sauer, and B. Becker, “On metrics to quantify the inter-device uniqueness of pufs.”

[7] F. Wilde, B. M. Gammel, and M. Pehl, “Spatial correlation analysis on physical unclonable functions,” vol. 13, no. 6, pp. 1468–1480.

[8] T. Arul, N. A. Anagnostopoulos, S. Reißig *et al.*, “A study of the spatial auto-correlation of memory-based physical unclonable functions,” in *2020 European Conference on Circuit Theory and Design (ECCTD)*. IEEE, pp. 1–4.

[9] Y. Wei, W. Rao, and N. Devroye, “Apuf production line faults: Uniqueness and testing,” in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, pp. 1–6.

[10] N. Mexis, T. Arul, N. A. Anagnostopoulos *et al.*, “Spatial correlation in weak physical unclonable functions: A comprehensive overview,” in *2023 26th Euromicro Conference on Digital System Design (DSD)*. IEEE, pp. 70–78.

[11] D. Chatterjee, A. Hazra, and D. Mukhopadhyay, “Formal analysis of puf instances leveraging correlation-spectra in boolean functions,” in *International Conference on Security, Privacy, and Applied Cryptography Engineering*. Springer, pp. 142–158.

[12] H. Cook, J. Thompson, Z. Tripp *et al.*, “Cloning the unclonable: Physically cloning an fpga ring-oscillator puf,” in *2022 International Conference on Field-Programmable Technology (ICFPT)*, pp. 1–10.

[13] S. Duan and G. Sai, “Bti aging-based physical cloning attack on sram puf and the countermeasure,” pp. 1–11.

[14] C. Helfmeier, C. Boit, D. Nedospasov *et al.*, “Cloning physically unclonable functions,” in *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*. IEEE, pp. 1–6.

[15] D. Nedospasov, J.-P. Seifert, C. Helfmeier *et al.*, “Invasive puf analysis,” in *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*. IEEE, pp. 30–38.

[16] T. Kroeger, W. Cheng, J.-L. Danger *et al.*, “Cross-puf attacks: Targeting fpga implementation of arbiter-pufs,” vol. 38, no. 3, pp. 261–277.

[17] T. Kroeger, W. Cheng, S. Guilley *et al.*, “Effect of aging on puf modeling attacks based on power side-channel observations,” in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, pp. 454–459.

[18] B. R. Anderson, R. Gunawidjaja, and H. Eilers, “Initial tamper tests of novel tamper-indicating optical physical unclonable functions,” *Applied Optics*, vol. 56, no. 10, pp. 2863–2872, 2017.

[19] T. Wilson and B. Cambou, “Tamper-sensitive pre-formed reram-based pufs: Methods and experimental validation,” *Frontiers in Nanotechnology*, vol. 4, p. 1055545, 2022.

[20] C. Helfmeier, C. Boit, D. Nedospasov *et al.*, “Physical vulnerabilities of physically unclonable functions,” in *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2014, pp. 1–4.

[21] M. R. Heidari Iman, G. Jervan, and T. Ghasempouri, “ARTmine: Automatic association rule mining with temporal behavior for hardware verification,” in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2024, pp. 1–6.

[22] M. R. H. Iman, P. Chikul, G. Jervan *et al.*, “Anomalous file system activity detection through temporal association rule mining,” in *9th ICISSP*, 2023, pp. 733–740.

[23] M. R. Heidari Iman, J. Raik, M. Jenihhin *et al.*, “An automated method for mining high-quality assertion sets,” *MICPRO*, vol. 97, p. 104773, 2023.

[24] J. Han, M. Kamber, and J. Pei, “6 - mining frequent patterns, associations, and correlations: Basic concepts and methods,” 2012, pp. 243–278.

[25] A. Roberts, M. R. Heidari Iman, M. Bellone *et al.*, “ADAssure: Debugging methodology for autonomous driving control algorithms,” in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2024, pp. 1–6.

[26] M. R. Heidari Iman, *Enhancing Assertion-Based Verification in Hardware Designs through Data Mining Algorithms*. TalTech Press, 2024, serial number: 37/2024. [Online]. Available: https://doi.org/10.23658/taltech.37/2024

[27] S. Vinagrero, H. Martin, A. de Bignicourt *et al.*, “SRAM-Based PUF Readouts,” vol. 10, no. 1, p. 333.