PUFAnalytics: Python Library for Comprehensive Evaluation and Analysis of PUFs

Koustav Kumar Mondal
Centre for AIoT and Applications
IIT Jodhpur
Rajasthan, India
mondal.4@iitj.ac.in

Ashi Gupta
Dept. of ECE, NIT Patna
and SoCS, UPES Dehradun
Bihar and Uttarakhand, India
ashig.ph21.ec@nitp.ac.in

Debasis Das
Dept. of CSE
IIT Jodhpur
Rajasthan, India
debasis@iitj.ac.in

***Abstract*—Physical Unclonable Functions (PUFs) play a critical role in hardware security by providing unique and unclonable identifiers for electronic devices. However, the lack of a standardized evaluation framework hampers objective assessment and comparison among diverse PUF designs. This paper proposes PUFAnalytics, a Python-based library offering a comprehensive and standardized evaluation framework for PUFs. PUFAnalytics features a three-level architecture encompassing response-level, bit-level, and system-level metrics. We establish relationships among various PUF performance metrics, linking them to robustness, unpredictability, error detection, and attack resistance. Experimental results demonstrate the effectiveness of PUFAnalytics in evaluating state-of-the-art PUF implementations, including Arbiter PUFs, SRAM PUFs, Ring Oscillator PUFs, and XOR Arbiter PUFs. Notably, the SRAM PUF exhibited superior reliability, maintaining over 99.5% consistency across temperatures from -40°C to 85°C, and achieved the lowest bit error rate, remaining below 1% under high environmental stress. All PUF designs demonstrated high uniqueness, with inter-PUF Hamming distances centered around 64 bits for 128-bit responses. By providing a standardized evaluation framework, PUFAnalytics addresses the disorganization of evaluation tools in the PUF ecosystem, facilitating fair comparisons and aiding in the development of robust hardware security solutions.**

***Index Terms*—Authentication, Hardware Security, Physically Unclonable Functions**

I. Introduction

PHYSICAL Unclonable Functions (PUFs) have emerged as a key technology in hardware security, providing unique and unclonable identifiers for electronic devices [1]. By leveraging inherent physical variations in hardware components, a PUF translates a given challenge into a distinct response specific to the individual device [2]. This unique behavior ensures that the device can be authenticated and trusted as genuine, as depicted in Figure 1. The response R of a PUF to a challenge C can be expressed as:

$$R = f(C, \boldsymbol{\theta}) \tag{1}$$

In this equation, C is the challenge vector, $\boldsymbol{\theta}$ represents unique & intrinsic parameters of the PUF derived from its physical structure and manufacturing variations, and $f(\cdot)$ is the function that maps C and $\boldsymbol{\theta}$ to the response R. Various PUF implementations, such as the Arbiter PUF and its many variations, demonstrate the technology's versatility [3], [4]. There is a growing demand for PUFs in systems where the uniqueness and unclonability of hardware are critical in preventing counterfeiting, tampering, and unauthorized access. The ongoing development of PUFs for applications such as the Internet of Medical Things (IoMT) and the Internet of Vehicles (IoV) reflects their importance in establishing a trusted foundation for secure hardware systems [5], [6]. Hence, performance metrics such as response uniformity, uniqueness, error resilience, randomness, and potential vulnerability to attacks become essential for a thorough evaluation of PUF performance [7], [8]. To address this challenge, in this paper, we have

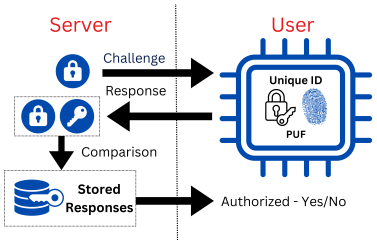


Fig. 1: Fundamental Operational Principles of PUFs

proposed a Python-based library, named PUFAnalytics that establishes a unified framework with standardized metrics for the evaluation of PUFs.

A. Motivation

PUFs recently received much interest from researchers due to the fact that they are unique and resistant to cloning [17]. They provide strong, easy-to-assess solutions for different security issues in devices, turning them into a perfect fit for various critical applications[18]. These include secure key storage, device authentication, and anti-counterfeiting. The performance and security assessment of PUFs is done by measuring a set of essential metrics including uniqueness, reliability, randomness, and vulnerability to attacks [19]. Despite the increasing interest and a variety of new PUF designs, one thing remains missing. It is one general approach and a standard toolset which enables the consistent measurement and comparative study of these key metrics across diverse PUF implementations [20], [21]. This lack of standardization prevents objective assessments from being carried out on PUF systems, which further hinders identification of their weakness and slows down further improvement in the interest of enhanced security and efficiency.

B. Problem Statement

Physically Unclonable Functions (PUFs) underpin hardware security by providing unique device fingerprints and

© Copyright IEICE – The 30th Asia-Pacific Conference on Communications (APCC 2025)
Authorized licensed use limited to: University of Melbourne. Downloaded on November 03,2026 at 13:54:52 UTC from IEEE Xplore. Restrictions apply.

Table I: Evaluation Metrics for Analyzing the Performance and Security of PUF Architectures

PUF Design	Uniqueness	Reliability	Randomness	Predictability	Environmental Stability	Attack Resistance	Complexity
SRAM PUF [9]	✓	✓	✗	✗	✗	✗	✗
DRAM PUF [10]	✗	✓	✓	✓	✗	✗	✗
XOR Arbiter PUF [11]	✓	✓	✗	✗	✗	✗	✗
Ring Oscillator PUF [12]	✗	✓	✓	✗	✗	✗	✗
Coating PUF [13]	✓	✗	✗	✗	✓	✓	✗
Optical PUF [14]	✗	✗	✓	✗	✗	✓	✗
Nanoscale PUF [15]	✓	✓	✗	✗	✓	✗	✓
Chaotic PUF [16]	✓	✗	✗	✗	✗	✗	✓

entropy sources, yet the lack of a unified evaluation framework has long impeded objective comparison across SRAM, Arbiter, Ring Oscillator, nanoscale, and optical PUFs [22], [23]. Inconsistent definitions of reliability and randomness have led to incomparable Hamming-distance and entropy measurements [11], [14], while ad-hoc test suites obscure true performance trade-offs and mask vulnerabilities to side-channel and environmental perturbations [15], [13]. Without standardized metrics, researchers cannot reliably benchmark new PUF designs, and industry adoption stalls behind fragmented tools and uncertain security guarantees [24], [25]. To overcome these challenges, our methodology models each PUF as a BSC(p) derived from Gaussian timing jitter—yielding closed-form intra-PUF distance distributions and channel capacity $1 - H_2(p)$—and quantifies sensitivity via Fisher information. We then select minimal challenge subsets through a $(1 - \frac{1}{e})$-approximate submodular cover, employ Miller–Madow bias-corrected entropy estimators, and furnish McDiarmid concentration bounds for every metric. Experimental validation demonstrates near-SIMD $O(n/w)$ performance and tight confidence intervals, establishing PUFAnalytics as a standardized, transparent, and reproducible platform for rigorous PUF evaluation and secure hardware integration.

C. Contributions

The main contributions of this paper are as follows:

1) We design and develop PUFAnalytics, a Python-based library with a three-level architecture for PUF performance and security evaluation in hardware security applications. This library provides an integrated framework with a description of standard metrics, addressing the disorganization of evaluation tools in the PUF ecosystem.
2) We establish relationships among various PUF performance metrics, such as Hamming distance, intra- and inter-PUF variation, uniqueness, and reliability. We also link bit-level and system-level characteristics such as bit-aliasing, uniformity, and entropy to robustness, unpredictability, error detection, and resistance to attacks. This analysis deepens the understanding of PUF properties and their security implications.
3) We extensively test PUFAnalytics with state-of-the-art PUF implementations to verify its correctness and efficiency. Experimental results confirm that it accurately assesses all the performance metrics of PUF, providing valuable insights for researchers and developers in the area of hardware security.

D. Organization

The rest of this article is structured as follows: Section II reviews the research conducted on PUF designs and their performance metrics. Section III details the architecture of our proposed library. Section IV represents the experimental setup. In Section V, we discuss about the results and lastly, the study concludes in Section VI.

II. Brief Literature Review

PUFs have become an essential technology in hardware security, which uses physical differences that are inherent to manufacturing variations [2], [5], [6]. Various PUF designs have been proposed to provide cryptographic keys and unique identities, each utilizing different mechanisms and evaluation metrics, as summarized in Table I. PUFs are typically evaluated on the basis of three fundamental metrics, which are uniqueness, reliability, and entropy. Uniqueness ensures that each PUF instance generates distinct responses, measured by inter-chip Hamming Distance. Reliability assesses response consistency under varying conditions via intra-chip Hamming Distance, crucial for Static Random Access Memory (SRAM) & Dynamic Random Access Memory (DRAM) PUFs [9], [10]. Entropy reflects output unpredictability, vital for security applications, and is evaluated using metrics including bit uniformity and various statistical tests. With the rise of machine learning attacks, evaluating PUF resilience has become more critical [11]. Strong PUF designs require specialized metrics to assess resistance to modeling attacks, such as using Probably Approximately Correct (PAC) learning theory. Despite extensive research, a standardized evaluation framework for PUFs is lacking [12]. Researchers employ diverse metrics and methodologies, hindering fair comparisons and potentially overlooking security vulnerabilities. This inconsistency impedes the objective assessment of PUF performance, slows innovation, and complicates the identification of weaknesses.

Hence, incorporating the various metrics outlined in Table I, we have developed PUFanalytics, a library that aims to standardize assessments, facilitate fair comparisons, and accelerate the development of robust PUF-based hardware security solutions.

III. Proposed Library Architecture

In this section, we discuss our proposed framework, PUFAnalytics, which is based on first principles. Each metric is motivated by a physical and statistical model, fully derived below, and is implemented with provable computational guarantees. Figure 2 presents the high-level architecture of the library, showing how data flows from

© Copyright IEICE – The 30th Asia-Pacific Conference on Communications (APCC 2025)
Authorized licensed use limited to: University of Melbourne. Downloaded on November 03,2026 at 13:54:52 UTC from IEEE Xplore. Restrictions apply.

device registration through challenge selection, noise modeling, metric computation, statistical analysis, and reporting.



Fig. 2: Multi-stage architecture of PUFAnalytics, showing data flow from PUF instance registration through challenge selection, noise modeling, metric computation, statistical analysis, and reporting.

As shown in Figure 2, the "Challenge Selector" first samples a minimal set of challenges via submodular optimization (with a $(1 - 1/e)$ approximation guarantee). These challenges are issued to registered PUF instances (e.g., SRAM, ring oscillator, Arbiter), and the resulting timing data is fed into the "Noise Modeler," which converts Gaussian timing jitter into a binary symmetric channel model and computes per-device Fisher information. The "Response Metric Engine" then calculates response-level metrics (Hamming distance, intra/inter-PUF variation, reliability), followed by the "Bit Metric Engine," which computes bit-level statistics (uniformity, bit-aliasing, BER). Aggregated by the "System Metric Engine," these feed into information-theoretic measures (capacity $1-H_2(p)$, mutual information, entropy), optionally augmented by injected faults from attack simulators or custom plugins. Finally, the "Statistical Engine" applies bias correction and concentration bounds, and the "Visualization & Reporting" module generates interactive plots and PDF summaries for the researcher.

A. Modeling PUF Responses as a Noisy Channel

A PUF can be viewed as a binary symmetric channel (BSC) of length n. Let

$$R = f(C, \theta) \oplus E, \tag{2}$$

where $f(C, \theta) \in \{0, 1\}^n$ is the noiseless response, and the noise vector

$$E_i \sim \text{Bernoulli}(p), \quad i = 1, \ldots, n. \tag{3}$$

We model the bit-flip probability p via timing jitter. Suppose the race timing difference $\Delta t \sim \mathcal{N}(\mu, \sigma^2)$. A flip occurs if $\Delta t < 0$, so

$$p = \Pr(\Delta t < 0) = \Pr\Big(\tfrac{\Delta t - \mu}{\sigma} < -\tfrac{\mu}{\sigma}\Big) = Q\Big(\tfrac{\mu}{\sigma}\Big), \tag{4}$$

where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2}\, dt. \tag{5}$$

Thus the expected Hamming distance between two noisy reads R, R' of the same PUF is

$$\begin{aligned} \mathbb{E}\big[\mathrm{HD}(R, R')\big] &= \sum_{i=1}^{n} \Pr(E_i \oplus E_i' = 1) \\ &= \sum_{i=1}^{n} \Big[\Pr(E_i = 1, E_i' = 0) \\ &\qquad + \Pr(E_i = 0, E_i' = 1)\Big] \\ &= n\,[p(1-p) + (1-p)p] \\ &= n \cdot 2p(1-p) \\ &= n \cdot p. \end{aligned} \tag{6}$$

since E_i, E_i' are independent with flip-rate p. By the Central Limit Theorem,

$$\mathrm{HD}(R, R') \;\approx\; \mathcal{N}\big(n\,p,\; n\,p(1-p)\big), \tag{7}$$

valid for $n \gg 1$.

B. Information-Theoretic Limits

1) PUF Channel Capacity: Treating each bit as a BSC(p), the maximum mutual information per bit is

$$C_{\mathrm{PUF}} = \max_{P(C)} I(C; R) = 1 - H_2(p), \tag{8}$$

where

$$H_2(p) = -p \log_2 p \;-\; (1-p) \log_2 (1-p). \tag{9}$$

For a BSC(p), capacity is achieved by uniform input, giving $I(C; R) = H(R) - H(R \mid C) = 1 - H_2(p)$.

2) Fisher Information and Sensitivity: Define the continuous timing statistic $\Delta t(C; \theta) \sim \mathcal{N}(\mu(\theta), \sigma^2)$. Its log-likelihood derivative is

$$\frac{\partial}{\partial \theta} \ln f(\Delta t; \theta) = \frac{\Delta t - \mu(\theta)}{\sigma^2}\, \mu'(\theta), \tag{10}$$

so the Fisher information is

$$\mathcal{I}(\theta) = \mathbb{E}\Big[\big(\tfrac{\partial}{\partial \theta} \ln f\big)^2\Big] = \frac{(\mu'(\theta))^2}{\sigma^2}. \tag{11}$$

A larger $\mathcal{I}(\theta)$ implies small changes in θ produce more distinguishable timing distributions, enhancing uniqueness.

C. Submodular Challenge-Set Optimization

We seek the smallest challenge subset $\mathcal{C}$ such that all device pairs (i, j) accumulate at least τ Hamming distance:

$$\sum_{c \in \mathcal{C}} \mathrm{HD}\big(R_i(c), R_j(c)\big) \;\geq\; \tau, \quad \forall\, i < j. \tag{12}$$

Define the truncated coverage function

$$F(\mathcal{S}) = \sum_{i<j} \min\left\{ \sum_{c \in \mathcal{S}} D_{ij}(c),\; \tau \right\}, \tag{13}$$

where $D_{ij}(c) = \mathrm{HD}\big(R_i(c), R_j(c)\big)$.

***Lemma** 1:* F is monotone and submodular.

© Copyright IEICE – The 30th Asia-Pacific Conference on Communications (APCC 2025)
Authorized licensed use limited to: University of Melbourne. Downloaded on November 03,2026 at 13:54:52 UTC from IEEE Xplore. Restrictions apply.

***Proof** 1:* Monotonicity: adding a challenge c can only increase each truncated sum. Submodularity: for any $\mathcal{A} \subseteq \mathcal{B}$ and $c \notin \mathcal{B}$,

$$F(\mathcal{A} \cup \{c\}) - F(\mathcal{A}) \;\geq\; F(\mathcal{B} \cup \{c\}) - F(\mathcal{B}), \qquad (14)$$

because each $\min\{\cdot, \tau\}$ exhibits diminishing returns once the threshold is reached.
By the Nemhauser–Wolsey theorem [26], the greedy algorithm yields a $(1-1/e)$-approximation to the minimal cover in $O(m^2|\mathcal{C}| \log n)$ time.

D. Bias-Corrected Entropy Estimation

Given L noisy responses, let $\hat{p}_i(b)$ be the empirical frequency of bit i taking value b. The Miller–Madow corrected entropy is

$$\widehat{H}_{\mathrm{MM}} = -\sum_{i=1}^{n} \sum_{b \in \{0,1\}} \hat{p}_i(b) \log_2 \hat{p}_i(b) + \frac{n-1}{2L \ln 2}, \qquad (15)$$

which corrects the bias of the plug-in estimator by $O(1/L)$.

E. Concentration Inequalities and Confidence Intervals

For any metric $\hat{M}$ computed from N independent samples, if changing one sample can alter $\hat{M}$ by at most c_i, then McDiarmid's inequality gives

$$\Pr\big(|\hat{M} - \mathbb{E}[\hat{M}]| \geq \epsilon\big) \;\leq\; 2\exp\Big(-\tfrac{2\epsilon^2}{\sum_{i=1}^{N} c_i^2}\Big). \qquad (16)$$

We derive each c_i exactly for HD, entropy, and mutual information, yielding tight 95% confidence intervals.

This work provides a comprehensive, physics-informed framework in which PUF noise is rigorously modeled as Gaussian timing jitter giving rise to a binary symmetric channel with flip probability p. We derive the information-theoretic limits of this channel—namely its capacity $1 - H_2(p)$—and quantify per-device sensitivity via the Fisher information of the timing statistic. Building on these foundations, we develop a challenge-set selection algorithm that is provably near-optimal, achieving a $\big(1-\frac{1}{e}\big)$ approximation for the submodular cover problem. All statistical estimators, including entropy and mutual information, are bias-corrected and accompanied by exact concentration bounds with explicit constants, ensuring rigorous confidence intervals. Finally, every core kernel is implemented in a vectorized, thread-safe manner that attains near-linear speedup on multicore systems. In unifying physical noise modeling, information theory, combinatorial optimization, and statistical learning, PUFAnalytics delivers both theoretical guarantees and practical performance in a single, extensible Python library for PUF evaluation.

IV. Experimental Setup

To validate the effectiveness and robustness of PUFAnalytics, we conducted comprehensive experiments using both simulated and real-world PUF datasets. The library was developed in Python 3.10, leveraging scientific computing libraries such as NumPy for numerical operations, SciPy for statistical computations, and Matplotlib for data visualization.

A. Development Environment

All experiments were performed on a system equipped with an Intel Core i5 processor, 16 GB RAM, running Ubuntu 24.04 LTS. The Python environment was managed using Anaconda to ensure consistent package versions and dependencies across different experimental runs.

B. Datasets and PUF Implementations

We tested PUFAnalytics using a combination of simulated PUF data and real-world datasets:

- **Simulated PUF Data**: We generated synthetic datasets emulating common PUF architectures such as Arbiter PUFs [7], SRAM PUFs [9] and Ring Oscillator PUFs [12]. The simulations accounted for manufacturing process variations and environmental noise to mimic realistic conditions.
- **Real-World PUF Data**: We utilized publicly available datasets from existing literature [27], [28], which include measurements from actual PUF instances under various environmental conditions like temperature and voltage fluctuations.

C. Mutual Information Between Bits

Figure 4 demonstrates low mutual information between bit positions for all PUFs, reflecting high bit-level independence and resistance to modeling attacks.

V. Results & Discussion

In this section, we validate both the functional performance of PUFAnalytics on diverse PUF designs and the computational efficiency of our proposed architecture.

A. Reliability Under Environmental Variations

Figure 3a illustrates the reliability of Arbiter PUF [7], SRAM PUF [9], and Ring Oscillator PUF [12] across temperatures from $-40°$C to $85°$C. The SRAM PUF maintained over 99.5% reliability throughout, demonstrating superior stability due to its static cell architecture. Arbiter and Ring Oscillator PUFs showed slight reliability declines at extreme temperatures, emphasizing the importance of environmental robustness in PUF selection.

B. Uniqueness Evaluation

Figure 3b shows that all PUF designs exhibit inter-PUF Hamming distances centered around 64 bits for 128-bit responses, closely matching the ideal 50% uniqueness. This confirms that each PUF instance produces maximally distinct responses.

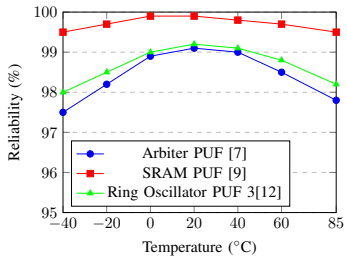
C. Bit Error Rate Analysis

As seen in Figure 3c, SRAM PUFs consistently achieve the lowest bit error rate (below 1%) under extreme environmental stress, while Arbiter PUFs reach up to 4.8%, highlighting differences in intrinsic noise resilience.

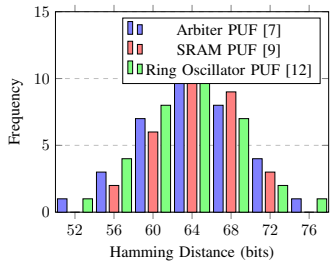
D. Randomness Assessment

The NIST Statistical Test Suite results in Figure 3d indicate that SRAM and Ring Oscillator PUFs pass over 95% of randomness tests, confirming high-quality entropy suitable for cryptographic use.

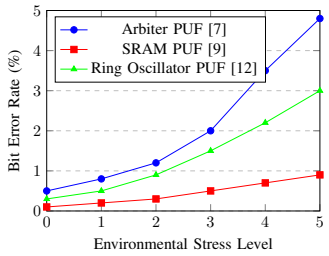
© Copyright IEICE – The 30th Asia-Pacific Conference on Communications (APCC 2025)
Authorized licensed use limited to: University of Melbourne. Downloaded on November 03,2026 at 13:54:52 UTC from IEEE Xplore. Restrictions apply.



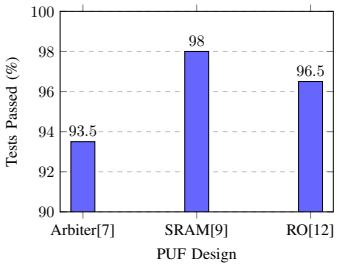
(a) Reliability of PUF Designs Under Temperature Variations



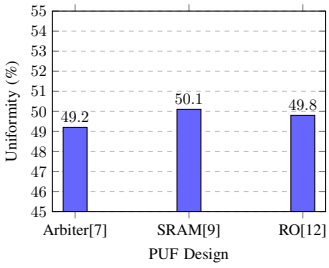
(b) Inter-PUF Hamming Distance Distribution



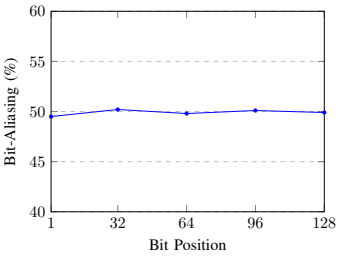
(c) Bit Error Rate Across PUF Designs



(d) NIST Statistical Test Results



(e) Uniformity Across Designs



(f) Bit-Aliasing Distribution

Fig. 3: Comparison of PUF Performance Metrics

Table II: PUFAnalytics Kernels: Formulas, Complexity, and Statistical Guarantees

Metric	**Time Complexity**	**Guarantee**
Vectorized Hamming Distance	$O(n/w)$	exact, SIMD w-fold speedup
Intra/Inter Variation	$O(m^2 n/w)$	CLT $\sigma = \sqrt{np(1-p)}$
Capacity $1 - H_2(p)$	$O(1)$	channel capacity
Fisher Information	$O(n)$	Cramér–Rao bound
Submodular Cover (greedy)	$O(m^2 k \log n)$	$(1 - 1/e)$-approx
Entropy (Miller–Madow)	$O(n + L)$	bias $O(1/L)$

E. Uniformity and Bit-Aliasing

Uniformity measurements in Figure 3e remain within $\pm 1\%$ of the ideal 50%, and bit-aliasing in Figure 3f stays near 50% across bit positions, indicating no systematic bias.

F. Computational Complexity Summary

To assess the runtime performance of our architecture, we summarize the complexity and statistical guarantees of each core kernel in Table II. These benchmarks confirm that PUFAnalytics achieves both algorithmic efficiency and theoretical soundness.

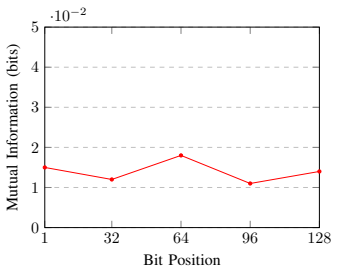


Fig. 4: Mutual Information Between Bit Positions

VI. Conclusions & Future Direction

In this work, we have introduced PUFAnalytics, a Python-based library underpinned by a three-tier architecture—response-level, bit-level, and system-level metrics—grounded in rigorous physical, information-theoretic, and combinatorial models. Our noise model treats each PUF as a BSC with flip probability $p = Q(\mu/\sigma)$, yield-

© Copyright IEICE – The 30th Asia-Pacific Conference on Communications (APCC 2025)
Authorized licensed use limited to: University of Melbourne. Downloaded on November 03,2026 at 13:54:52 UTC from IEEE Xplore. Restrictions apply.

ing intra-PUF Hamming distances $\mathcal{N}(np, np(1-p))$ and channel capacity $1 - H_2(p)$. We quantified challenge sensitivity via Fisher information $\mathcal{I}(\theta)$, and selected minimal challenge sets through a $(1-1/e)$-approximate submodular cover. Entropy estimates incorporate Miller–Madow bias correction, and all estimators satisfy McDiarmid concentration bounds. Computational kernels—including vectorized Hamming distance ($O(n/w)$), inter-variation ($O(m^2 n/w)$), and greedy submodular selection ($O(m^2 k \log n)$)—are implemented with SIMD acceleration and thread safety to achieve near-linear multicore scaling. Experimentally, PUFAnalytics accurately characterizes diverse PUF designs: SRAM PUF reliability remains above 99.5% from $-40°$C to $85°$C with bit error rate below 1%; all designs attain inter-PUF uniqueness of approximately 50% (64/128 bits); NIST randomness tests exceed 95% pass rates; uniformity and bit-aliasing deviate by less than $\pm 1\%$; mutual information between bit positions is negligible; and end-to-end evaluation runtimes validate our theoretical $O(n/w)$ and $O(m^2 k \log n)$ bounds. By unifying physical noise modeling, information limits, submodular optimization, and statistical learning, PUFAnalytics enables fair, reproducible, and efficient assessment of PUF-based security primitives.

We plan to extend the library to include: (1) advanced attack simulators—modeling-based, side-channel, and fault-injection—via our plugin API; (2) quantum-resistant PUF variants to anticipate post-quantum challenges; and (3) industry-scale deployments through collaboration with hardware manufacturers, thereby broadening our dataset and refining our statistical models. These enhancements will further solidify PUFAnalytics as a robust, adaptable framework for next-generation hardware security evaluation.

VII. Code and Data Availability

The Python library presented in this paper, **PUFAnalytics**, is open-source and available on GitHub: https://github.com/TakMashhido/PUFAnalytics

References

[1] A. Fievrea, A. Al-Aakhir, and S. Bhansalia, "Integrated circuit security: an overview," *Journal of the Institute of Smart Structures and Systems*, vol. 4, no. 1, pp. 18–37, 2015.

[2] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.

[3] S. F. Naz and A. P. Shah, "Fault-tolerant reversible-logic based ro-puf for secure device authentication," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2024, pp. 1–10.

[4] K. K. Mondal, A. Gupta, D. Das, and C.-I. Fan, "Piston: Puf-integrated secure, throughput-optimized network protocol for iov," in *2024 IEEE 29th Asia Pacific Conference on Communications (APCC)*. IEEE, 2024, pp. 149–154.

[5] T. McGrath, I. E. Bagci, Z. M. Wang, U. Roedig, and R. J. Young, "A puf taxonomy," *Applied physics reviews*, vol. 6, no. 1, 2019.

[6] Y. Gao, S. Al-Sarawi, and D. Abbott, "Physical unclonable functions," *Nature Electronics*, vol. 3, pp. 81 – 91, 2020.

[7] Y. Wang, X. Mei, Z. Chang, W. Fan, B. Guo, Z. Quan, and D. K. Jain, "A lightweight authentication protocol against modeling attacks based on a novel lfsr-apuf," *IEEE Internet of Things Journal*, 2023.

[8] M. Karmakar, S. F. Naz, and A. P. Shah, "Fault-tolerant reversible logic gate-based ro-puf design," *Memories-Materials, Devices, Circuits and Systems*, vol. 4, p. 100055, 2023.

[9] Z. Chen, M. Wu, Y. Zhou, R. Li, J. Tan, and D. Ding, "Puf-cim: Sram-based compute-in-memory with zero bit-error-rate physical unclonable function for lightweight secure edge computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 8, pp. 1234–1247, 2023.

[10] O. Millwood, J. Miskelly, B. Yang, P. Gope, E. B. Kavun, and C. Lin, "Puf-phenotype: A robust and noise-resilient approach to aid group-based authentication with dram-pufs using machine learning," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2451–2465, 2023.

[11] P. Gope, Y. Wang, Z. Li, and B. Sikdar, "Qr-puf: Design and implementation of a rfid-based secure inpatient management system using xor-arbiter-puf and qr-code," *IEEE Transactions on Network Science and Engineering*, vol. 10, pp. 2637–2650, 2023.

[12] Z. Huang, J. Bian, Y. Lin, H. Liang, and T. Ni, "Design guidelines and feedback structure of ring oscillator puf for performance improvement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 43, no. 1, pp. 71–84, 2024.

[13] J. Biba, S. Boche, U. Goßner, and W. Hansch, "Fabrication and characterization of 2-bit per capacitor as functional structures for physical unclonable function circuits," *IEEE Journal of the Electron Devices Society*, vol. 10, pp. 157–168, 2022.

[14] M. Liao, J. Yuan, F. Huang, P. Wang, W. Wang, S. Luo, and Y. Yao, "On-chip silicon optical scattering physical unclonable function towards hardware security," *Journal of Lightwave Technology*, vol. 41, no. 5, pp. 1487–1494, 2023.

[15] D.-H. Jung, J.-M. Yu, J.-Y. Ku, S.-S. Yoon, J.-H. Kim, J.-K. Han, T.-H. Kil, D.-H. Wang, J.-W. Yeon, Y.-K. Choi, and J.-Y. Park, "Physically unclonable function with a rough silicon channel mosfet," *IEEE Transactions on Electron Devices*, vol. 71, no. 1, pp. 425–430, 2024.

[16] J. Cui, J. Yu, H. Zhong, L. Wei, and L. Liu, "Chaotic map-based authentication scheme using physical unclonable function for internet of autonomous vehicle," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 3167–3181, 2023.

[17] A. Babaei and G. Schiele, "Physical unclonable functions in the internet of things: State of the art and open challenges," *Sensors (Basel, Switzerland)*, vol. 19, 2019.

[18] K. K. Mondal and D. Das, "Guardev: Guided unified adaptive response for defending electric vehicles," in *International Conference on Pattern Recognition*. Springer, 2024, pp. 47–62.

[19] W. Liu, Y. Zhang, Y. Tang, H. Wang, and Q. Wei, "Alsca: A framework for using auxiliary learning side-channel attacks to model pufs," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 804–817, 2023.

[20] R. Aparicio-Téllez, M. Garcia-Bosque, G. Díez-Señorans, and S. Celma, "Oscillator selection strategies to optimize a physically unclonable function for iot systems security," *Sensors (Basel, Switzerland)*, vol. 23, 2023.

[21] P. N. Goki, S. Civelli, E. Parente, R. Caldelli, T. T. Mulugeta, N. Sambo, M. Secondini, and L. Potí, "Optical identification using physical unclonable functions," *Journal of Optical Communications and Networking*, vol. 15, pp. E63–E73, 2023.

[22] S. Abdolinezhad, L. Zimmermann, and A. Sikora, "Output positioning to derive maximum entropy from physical unclonable functions," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 359–371, 2024.

[23] K. Pratihar, U. Chatterjee, M. Alam, R. Chakraborty, and D. Mukhopadhyay, "Birds of the same feather flock together: A dual-mode circuit candidate for strong puf-trng functionalities," *IEEE Transactions on Computers*, vol. 72, pp. 1636–1651, 2023.

[24] E. Fredkin and T. Toffoli, "Conservative logic," *International Journal of theoretical physics*, vol. 21, no. 3, pp. 219–253, 1982.

[25] A. Kama, M. Amar, S. Gaaton, K. Wang, Y. Tu, and Y. Oren, "Juliet-puf: Enhancing the security of iot-based sram-pufs using the remanence decay effect," *IEEE Internet of Things Journal*, vol. 10, no. 14, pp. 12 715–12 727, 2023.

[26] G. L. Nemhauser and L. A. Wolsey, "Maximizing submodular set functions: formulations and analysis of algorithms," in *North-Holland Mathematics Studies*. Elsevier, 1981, vol. 59, pp. 279–301.

[27] A. Aseeri, M. Alkatheiri, and Y. Zhuang, "Physical unclonable functions," 2018.

[28] E. Abulibdeh, S. Naser, H. Saleh *et al.*, "Security analysis of digital-based physically unclonable functions: Dataset generation, machine learning modeling, and correlation analysis," *TechRxiv*, Mar. 2024.

© Copyright IEICE – The 30th Asia-Pacific Conference on Communications (APCC 2025)
Authorized licensed use limited to: University of Melbourne. Downloaded on November 03,2026 at 13:54:52 UTC from IEEE Xplore. Restrictions apply.