Processing In-Memory PUF Watermark Embedding With Cellular Memristor Network

Alex James, *Senior Member, IEEE*, Chithra Reghuvaran, *Senior Member, IEEE*, and Leon Chua, *Fellow, IEEE*

***Abstract*—The cellular neural network (CNN or CeNN) is known to be useful because of its suitability in real-time processing, parallel processing, robustness, flexibility, and energy efficiency. CeNNs have a large number of interconnected processing elements, which can be programmed to produce a wide range of patterns, including regular and irregular patterns, random patterns, and more. When implemented in memristive hardware, the pattern generator ability and inherent variability of memristive devices can be explored to create Physical Unclonable Functions (PUFs). This work reports a method of using memristive CeNNs to perform image processing tasks along with PUF image generation. The CeNN-PUF has dual mode capability combining data processing and encryption using PUF image watermarking. The proposed method provides unique device-specific image watermarks, following a two-stage process of (1) device-specific secret mask generation and (2) watermark embedding. The system is evaluated using multiple CeNN cloning templates and the robustness of the method is validated against ML attacks. A detailed analysis is presented to evaluate the uniqueness, randomness and reliability against different environmental changes. The experimental validation of the proposed model is done on FPGA Xilinx Zynq-7010 processor and benchmarked the system against quantization noise.**

***Index Terms*—Physical unclonable function (PUFs), cellular neural network (CeNN), memristive crossbar array, image watermarking, resistive RAM (RRAM), structural dissimilarity index (SDIM), reliability.**

I. Introduction

THE physical unclonable functions (PUFs) can be used to create patterns as unique identifiers or fingerprints that can be used as a watermark for tracking digital content's origin and authenticity. The PUF watermarking embeds a unique and unpredictable pattern generated by PUF known as PUF response, into digital content. This response can be extracted from the digital content and compared with the original PUF response to validate the authenticity of the content. These PUF responses are complicated to clone, making it a secure means to track the origin of digital content. Further, making the embedding difficult can also make removing or modifying the watermark difficult. Due to this, they find use in several security applications using unique identifiers, secret keys, public keys and authentications [1], [2].

PUF-based logic is well-situated to be implemented in low-cost and energy-efficient hardware. While existing microcontrollers and microprocessors can be used, they tend to be less energy efficient than application-specific integrated circuits. Among the emerging hardware, memory-based/memristive PUF has evolved as a powerful alternative to Complementary Metal–Oxide–Semiconductor (CMOS) based designs because of its simple structure, low power, high randomness, and availability in computing [12].

Watermarking in images helps to track the origin and ownership of digital images and helps prevent unauthorized copying. Spatial and transform domain watermarking are commonly used techniques. The spatial domain watermarking embeds the watermark directly into image pixels while in a transform domain approach watermark is embedded in a transformed representation of the image.

Many times images need to be processed with image filters, such as when it has noise or needs to be transformed. Processing images with image filters followed by PUF watermarking requires sequential processing. This requires the implementation of the weight summation operation followed by PUF circuits. Rather than using dedicated spatial filters, an alternative approach is to make use of trainable Cellular neural networks (CeNN)[1] to create various filters. Further, CeNN processes information locally within their neighborhood and can be used to generate patterns. The behavior of the CeNN cells can also be used as a unique identifier. When implemented in hardware, cell behavior is affected by process variations, temperature, and other factors, resulting in a unique, uncloneable characteristic. The behavior of the CeNN cells is then used as the basis for authentication, by comparing it to a reference pattern stored in the device.

Memristor-based systems have been shown to be efficient in the implementation of both PUF and CeNN hardware. The first demonstration of the memristor-based PUF was presented

Received 12 February 2024; revised 12 November 2024; accepted 8 January 2025. Date of publication 23 January 2025; date of current version 8 December 2025. This work was supported by the Ministry of Electronics and IT, Government of India. *(Corresponding Author: Alex James.)*

Alex James is with the School of Electronic Systems and Automation, Digital University Kerala, Trivandrum 695317, India (e-mail: apj@ieee.org).

Chithra Reghuvaran is with University College Dublin, Dublin, Ireland.

Leon Chua is with the University of California, Berkeley, CA 94720 USA.

Digital Object Identifier 10.1109/TETC.2025.3528336

[1]It may be noted that historically, CNN is the abbreviation used for Cellular Neural Networks. To avoid confusion with convolutional neural networks, we explicitly note CeNN as the preferred notation for Cellular Neural Networks in this work.

TABLE I
MEMORY-BASED PUF WORKS IN THE LITERATURE: A COMPARISON, SSIM: STRUCTURAL SIMILARITY INDEX MEASURE, *DUAL MODE OPERATION INCLUDES MEMORY OPERATION AND PUF APPLICATION

PUF Method	PUF Parameter	Application	PUF Properties				
			Uniqueness	Randomness	Reliability	ML attack	Dual mode operation*
Polyomino based M-PUF [3]	Write time	Cryptography	✓	✓	-	✓	-
WTMPUF [4]	Write time	IC privacy, Counterfeiting, Secret Key Storage	✓	-	-	-	-
mrPUF [5]	Readout Time	Cryptography	✓	-	-	✓	-
mrSPUF [1]	Resistance Variation	Revocation or update of key information	✓	✓	✓	✓	-
Selected Bit-line Current (SBC) PUFs [6]	Set Voltage	Encryption	✓	✓	✓	✓	-
Hybrid Memristor-CMOS PUF [7]	Delay	Cryptography	✓	-	✓	✓	-
Polyomino PUF [8]	Delay	Cryptography	✓	-	-	✓	-
Modified XbarPUF [2]	Switching Time	Secret Key Generation	✓	-	-	✓	-
Halide Pervoskite memristor PUF [9]	Switching Time	Cryptography and Device Authentication	✓	✓	✓	✓	-
XbarPUF [10]	Resistance	Cryptography	✓	-	✓	-	-
Memristive Chaotic Oscillator [11]	Resistance	Image Watermarking	✓	-	-	✓	-
Buffer Free memory-based PUF (BF-MPUF) [12]	Threshold voltage	Cryptography and Data storage	✓	-	✓	-	✓
CeNN-PUF (proposed)	**Resistance**	**Image Watermarking**	✓	✓	✓	✓	✓

in [8]. A large number of other memory-crossbar based PUFs have been proposed in the literature, for example, metal-oxide memristor based or Resistive RAM (RRAM) [1], [2], [4], [5], [6], [7], Spin-transfer torque magnetic RAM (STT-MRAM) [12] etc. Table I shows a comparison of the memory-based PUF works in the literature. The table lists the different PUF methods with reference to the considered PUF parameter, target applications and the PUF properties considered for analysis. Table I shows that the different PUF methods use variations in device parameters like resistance state, switching time and threshold voltages. These unpredictable probabilistic characteristics of memristor crossbars form the basis for PUF applications.

The existing memristor-based PUF works have been successful in addressing applications with secure key generation. However, they introduced an undesired on-chip area and power overhead due to the additional buffer for storing data during the time of key generation. Hence such methods are not an optimal solution for PUF watermarking.

A PUF method with dual mode capability, i.e., simultaneous memory operation and PUF application, is required for watermarking, which can help optimize the hardware requirements. The proposed CeNN-based PUF can generate PUF feature masks without additional hardware circuits and thus reduces hardware overheads in ensuring security. Alternatively, the use of in-memory processing and memristor arrays removes the von Neumann bottleneck and makes the CeNN-based PUF implementations more efficient.

The memristive CeNN implementation makes use of crossbar arrays, that often have device and cycle-to-cycle variability. These variations generate randomness in the resistive states of the memristors and can be utilized for the generation of PUF feature masks. In the area of CeNN, numerous works have reported the use of CeNN for image processing but very few works have reported the generation of PUF from CeNN. The analog cellular neural networks with application in PUF design were proposed in [13], [14]. The PUF designs in [13], [14] are also used in secret key generation for cryptography. The image processing capabilities of CeNN have not been utilized yet in PUF designs.

In the paper, we report the use of memristive CeNN that can simultaneously perform image processing as well as be used for PUF-based image watermarking. The cloning templates of the CeNN and its connections are modeled as memristor-crossbars in the proposed CeNN architecture. The massive aggregation of cells in CeNN creates high randomness. The proposed system is evaluated with different cloning templates for smoothing, edge detection, and half-toning using the Stanford-PKU RRAM model [15], [16]. The SPICE model in [15], [16] captured the essential device variation characteristics including resistance distributions and hence considered in this paper for modeling PUF. The experimental validation of the proposed model is done on the FPGA processor, Xilinx Zynq-7010, and benchmarked the system against quantization noise. A detailed analysis is presented to evaluate the uniqueness, security, and robustness of CeNN. The security of the CeNN-PUF against ML attacks is analyzed. The on-chip area and power requirements of the proposed CeNN architecture are also presented.

This paper is organized into the following sections: The proposed CeNN-PUF image watermarking technique is introduced in Section II. Section III discusses the details of various possible attacks on CeNN-PUF. Section IV reports the results and related discussions and Section V provides the concluding remarks.

II. PROPOSED CENN-PUF IMAGE WATERMARKING

The PUF is a method of using intrinsic manufacturing variability during the fabrication process to generate an unclonable signature for every single device [17]. In conventional methods, the intrinsic variability is extracted by adding a specific PUF circuit which may be an ASIC or part of a system on chip [17]. Conventional PUF circuits are implemented in the digital domain which receives a sequence of challenge bits,

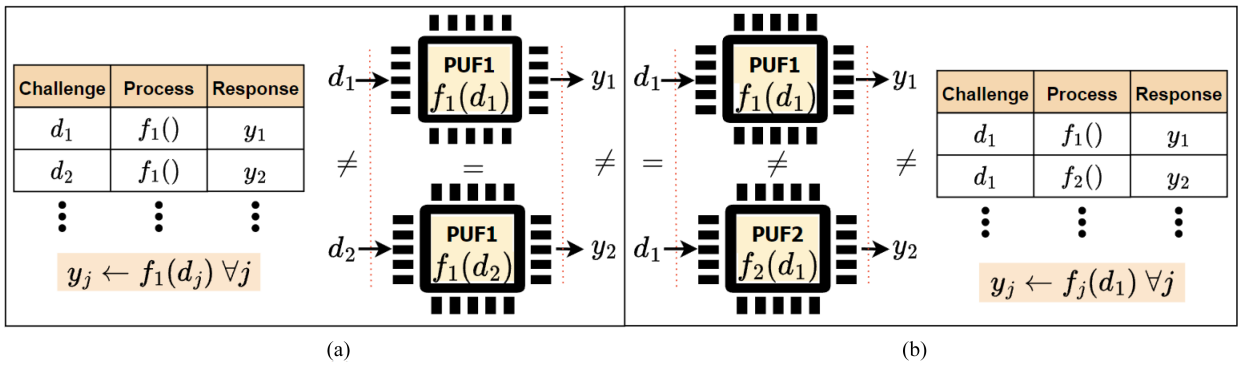


Figure 1. PUF concept (a) PUF generating different responses for different challenges, $y_j \leftarrow f_1(d_j)\ \forall j$ and (b) PUF generating different responses because of process variations, $y_j \leftarrow f_j(d_1)\ \forall j$.

d_j, as the input and generates a sequence of response bits, y_j as the output, Figure 1(a). Mathematically, such PUF can be expressed as a probabilistic procedure that maps randomized input challenges to output responses, $y_j \leftarrow f_1(d_j)\ \forall j$, where j represents the number of Challenge-Response-Pairs (CRPs). The function $f_1(d_j)$ translates input challenges d_j to output responses y_j.

The images are the most frequently shared data in various applications such as smart healthcare, military communication and broadcast monitoring [18]. Watermarking is a commonly used copyright protection tool for image transmission, tracking and processing. The conventional PUF introduces undesired on-chip area and power overhead due to the additional PUF circuitry for key generation. Such methods are not an optimal solution for PUF image watermarking. Hence we propose a memory-based system that can generate PUF-based feature images for watermarking as shown in Figure 1(b). Here the PUF system is made of memristive crossbar arrays and the memory device parameter variations in resistance state, switching time and threshold voltage generate PUF images. In Figure 1(b), the function $f_j()$ captures the device variations and generates PUF. Whereas in conventional methods, the intrinsic variability of the memory device is given as input to a specific PUF circuit [17], i.e, the input d_j varies with device variations and the function $f_1()$ translates to responses y_j. In the proposed system, the function $f_j()$ captures the device variations and generates PUF since the system combines an image processing task and the generation of PUF response. The proposed PUF process can be expressed as:

$$y_j \leftarrow f_j(d_1)\ \forall j \tag{1}$$

In (1), the mapping function captures the intrinsic manufacturing variabilities and translates to device-specific responses in the analog domain. The proposed work uses PUF images extracted from memristive CeNN architecture for watermarking. The entropy source considered here is the randomness in resistance variation of the CeNN memristor crossbar array.

A. PUF Image Watermarking

For implementing the proposed analog PUF, the RRAM variations are introduced during the ASIC fabrication. These variations ensure a unique identity for the device for PUF watermarking. The block diagram representation of the proposed PUF image watermarking system is represented in Figure 3(a). The PUF feature images are generated using memristive CeNN architecture. In Figure 3(a), the memristive CeNN-PUF system combines an image processing task and the generation of PUF response. The CeNN architecture generates a feature image for watermarking, with the source of PUF entropy mapped as the randomness in resistance variation of the CeNN memristor crossbar array. The feature image output from the CeNN circuit is watermarked with the original image to generate the watermarked PUF image as shown in Figure 3(a). The proposed CeNN-PUF method can generates device-specific feature images for compressed and cropped input images also. The CeNN-PUF architecture consists of two main tasks, (1) Secret Mask Generation and (2) Watermark embedding.

Generating PUF response: The feature response for watermarking is generated using the memristive CeNN. The original image, d, is fed as input to the CeNN crossbar network. For an input image size of $M \times N$, the required neural network architecture size is $M \times N$. In CeNN architecture, the input current to each cell is the weighted summation of feedforward and feedback inputs [19], [20]. The process variations result in the randomness of the memristor resistive states. The variations cause fluctuations in the input current and introduce randomness in the output y_j. This forms the principle for generating PUF feature images for watermarking. The memristive CeNN translates the input to device-specific output PUF responses. The possibility of varying the cloning template makes the CeNN architecture flexible for various target applications.

Watermark Embedding: The generated PUF image, y_j, for watermarking is embedded on to the original image, d. The additive rule for watermark embedding states that the CeNN output is added to the original image according to a perceptual masking factor, Λ, i.e., $z_j = d + \Lambda \bullet y_j$ [21]. The masking factor can be calculated using the equation, $\Lambda = (1-\varphi)\alpha + \varphi \bullet \beta$ where φ, α, β are watermark embedding factors [21]. $\varphi \to 0$ implies watermark embedding is strong and $\varphi \to 1$ implies weak watermark embedding. The watermarked image from each device will be, therefore, unique. At the receiver end, the original

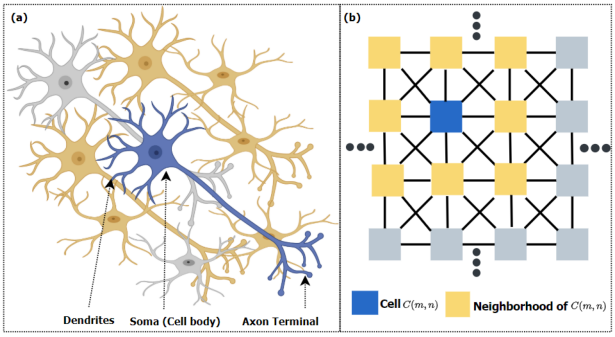


Figure 2. (a) Biological Neural Network (the blue colored neuron is connected directly to six neighboring neurons which are shown in yellow color) and (b) Structure of CeNN showing 3×3 neighborhood of $C(m,n)$.

image is extracted using the information of the feature image, y_j, at the receiver.

B. Memristive Cellular Neural Network

The CeNN, developed by L.O Chua and L. Yang [22], mimics the features of neural networks and cellular automata and finds applications in the area of image processing, Figure 2 [22], [23], [24], [25], [26]. The neurons in the biological neural network consist of the soma (cell body), dendrites and axon, Figure 2(a). The neurons are connected directly to their neighbors through axons and dendrites. The structure of CeNN features the multi-dimensional array arrangement of neural networks and the neighborhood interconnections between biological neurons, Figure 2(b) [19], [20]. A $M \times N$ rectangular grid array of CeNN network consists of $M \times N$ basic units called cells and each cell is connected only to its neighboring cells defined within a radius, r. The neighbours of cell $C(m,n)$ are $C(k,l)$ with $\{|k-m| \leq r\}$ and $\{|l-n| \leq r\}$, where $m \in \{1,\ldots,M\}$, $n \in \{1,\ldots,N\}$ [20].

For an input image of size $M \times N$, the CeNN network also consists of $M \times N$ in generating the feature mask. The input to cell $C(m,n)$ is the normalized pixel value $d_{m,n}$ of the input image. The connections from each cell $C(m,n)$ to its neighbors $C(k,l)$ are defined by cloning templates, $A(m,n;k,l)$ and $B(m,n;k,l)$ for feedback and feedforward connections respectively [23]. The input of $C(m,n)$ is connected to $C(k,l)$ through the feedforward weights $B(m,n;k,l)$. The output of the cell denoted by $y_{m,n}$ is fed to $C(k,l)$ through the feedback weights $A(m,n;k,l)$. The output current, $i_{m,n}$, is the weighted summation of feedforward and feedback inputs and can be mathematically expressed as [19], [20]:

$$i_{m,n}(t) = \sum_{k,l} A(m,n;k,l) y_{k,l} + \sum_{k,l} B(m,n;k,l) d_{k,l} + I_b \tag{2}$$

where I_b is the constant bias current. The state voltage of each cell, $x_{m,n}$(t), is determined based on the equivalent cell resistance, R_C, and capacitance, C, values respectively. R_C can take any values between 1 KΩ and 1 MΩ. Time constant $\tau = R_C C$ is usually between 10^{-8} to 10^{-5} s [22]. The function $y_{m,n} = \frac{1}{2}(|x_{m,n}(t)+1| - |x_{m,n}(t)-1|)$ translate the state voltage to feature image pixel values [22]. Thus $M \times N$ CeNN network generates a feature image of size $M \times N$.

Several implementations of the memristor CeNN memristor cell structures are presented in [24], [25], [28]. The CeNN structure in [24], [25], [28] uses a single memristor in the cell $C(m,n)$ instead of the resistive circuit element. The proposed memristor-based circuit implementation of the CeNN structure is shown in Figure 3(b). The figure shows the cell structure of a single cell $C(m,n)$. The proposed circuit design realizes the feedback and feedforward connections to the cell in the form of a memristor crossbar array as shown in Figure 3(b). As in Figure 3(b), the weighted summations $A(m,n;k,l)y_{k,l}$ and $B(m,n;k,l)d_{k,l}$ can be realized using two memristive crossbars, one for feedback path and the other for feedforward path. The $A(m,n;k,l)$ and $B(m,n;k,l)$ matrices form the conductance values of the crossbar and $y_{k,l}$, $d_{k,l}$ forms the horizontal inputs. The matrix $A(m,n;k,l)$ and $B(m,n;k,l)$ need to be unrolled from $(2r+1) \times (2r+1)$ matrix to $1 \times (2r+1)^2$ 1D array as shown in Figure 3(b). The cloning templates and bias currents can take both positive and negative values. Hence, we need to define two columns in memristor crossbars, one for positive and other for negative weights, i.e, $A(m,n;k,l) \propto \{A^+(m,n;k,l) - A^-(m,n;k,l)\}$ and $B(m,n;k,l) \propto \{B^+(m,n;k,l) - B^-(m,n;k,l)\}$ and $I_b \propto \{I_b^+ - I_b^-\}$. Hence the output current can be rewritten as:

$$\begin{aligned} i_{m,n}(t) = & \sum_{k,l} \left(A^+(m,n;k,l) - A^-(m,n;k,l)\right) y_{k,l}(t) \\ & + \sum_{k,l} \left(B^+(m,n;k,l) - B^-(m,n;k,l)\right) d_{k,l} \\ & + \left(I_b^+ - I_b^-\right) \end{aligned} \tag{3}$$

The equivalent resistance and capacitance determine the state voltage of the cell with respect to $i_{m,n}$. The non-linearity between the input and output is introduced by the activation function, i.e., $y_{m,n}(t) = f(x_{m,n}(t))$. Here, we use *tanh* activation function for representing the non-linearity.

Dynamic Range: For the CeNN, the dynamic range of cell states [22], $x_{m,n}$, at any time, $t > 0$ is defined as :

$$\max |x_{m,n}| \leq V_{\max}, \forall \{1 < m < M\}, \{1 < n < N\} \tag{4}$$

where $V_{\max}$ can be computed as $1 + R_C|I| + R_C[\sum |A(m,k;k,l)| + |B(m,k;k,l)|]$ [22]. In the circuit design, the parameters should be chosen such that $R_C|I| \ll 1$, $R_C|A(m,k;k,l)| \ll 1$ and $R_C|B(m,k;k,l)| \ll 1$ such that the $v_{\max}$ value should be within the power supply range of specifications of the circuit components.

Stability: The transient curve of the cellular neural network must always converge to a stable steady state when driven by an input image. The magnitude of stable equilibrium points should be greater than 1. After the transient decays and settle to at $t = t_1$, then for any time $t > t_1$, $x_{m,n} > 1$.

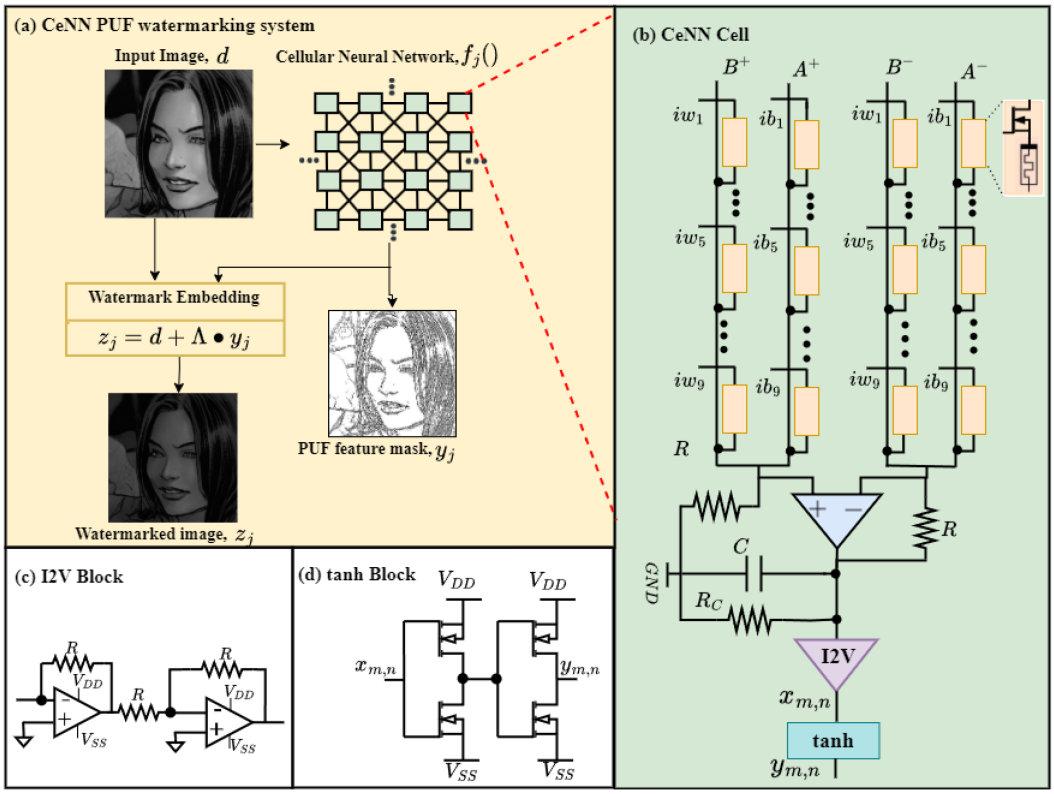


Figure 3. CeNN-PUF concept (a) Block Diagram Representation of the proposed CeNN PUF watermarking system, (b) Circuit level implementation of C(m,n) where $w_1 = d(m,n;m-1,n-1)$, $w_5 = d(m,n;m,n)$, $w_9 = d(m,n;m+1,n+1)$, $b_1 = y(m,n;m-1,n-1)$, $b_5 = y(m,n;m,n)$ and $b_9 = y(m,n;m+1,n+1)$ (c) I2V: Current to voltage converter and (d) Circuit implementation of *tanh* activation function [27].

III. Attack Formulation

The concept of 'secret key' is eliminated in PUF which eliminates the attacks on cryptosystems [29], [30]. The PUF is mostly tamper-evident but any physical manipulation like environmental conditions may change the CRP mapping based on the stability of underlying memory devices [31]. The literature shows that strong PUF structures can be mathematically approximated and cracked by Machine Learning (ML) algorithms [30]. The section analyses these attacks for the proposed CeNN-PUF image watermarking technique.

A. Machine Learning Attacks

The most common type of attack is the Machine Learning (ML) attack where the attacker learns the CRPs of the registered device to correctly predict the future responses [29], [31]. The objective of the attacker is to learn the PUF behavior from the known CRPs [31]. A Covariance Matrix Adaptation Evolution Strategies (CMA-ES) machine learning attack model is presented for XOR PUF in [30]. The main idea of CMA-ES is to generate random PUF instances and check the model accuracies to find the nearest model for XOR PUF [30]. We formulate the ML attack model similar to the CMA-ES approach in [30]. In the proposed CeNN-PUF, we assume that the PUF circuitry is known to the attacker and the attacker can generate the set of response pairs to train an ML model.

For the CeNN-PUF, the input is an image from the camera sensor. The source of entropy for CeNN-PUF is the random resistance variations of memristive crossbar arrays. The attacker needs to generate the number of responses equal to the number of resistance variation possibilities of memristors in a $M \times N$ CeNN array. The generated dataset is then used to generate ML models to find the nearest model for CeNN-PUF. Here the model accuracies are considered as the fitness parameter as in [30].

The massive aggregation of cells in CeNN creates high randomness creating strong PUF for watermarking. Consider a $M \times N$ rectangular grid array CeNN. Each cell with $2 \times (2r+1)^2$ memristors for representing positive weights and $2 \times (2r+1)^2$ memristors for representing negative weights of feed-forward and feedback cloning templates. The memristors will be programmed according to the values of forward and feedback cloning templates. The dimension of feature vectors for CeNN-PUF is a polynomial relation provided by:

$$N_{a-CRP} \sim O\left(4MN\left(2r+1\right)^2\right)^l \tag{5}$$

In (5), N_{a-CRP} scales exponentially with the number l. Each programmed memristor can have resistance variations, denoted by l, caused by various factors including material stack & switching physics, range of operating currents from devices, endurance & retention, temperature dependence, programming non-linearity, wire resistance, etc. [32]. For ML attack modeling, the attacker needs to generate the number of responses equal to

the number of resistance variation possibilities of memristors. The dataset is then used to train a neural network model to predict the device identity. But, the N_{a-CRP} values scale exponentially with an increase in image size and l. Thus the time required to train a neural network model exponentially increases with an increase in image size and l. The CMA-ES based attack model will work only with the assumption that the fundamental behavior of the system does not change over a period of time. The CeNN-PUFs are analog PUFs with a large number of variability parameters. The randomness in variability parameters makes the proposed PUF more resilient to ML attacks.

IV. Results and Discussion

In this paper, we use the Stanford-PKU SPICE RRAM model to capture the essential device variation characteristics including resistance distribution modeling for PUF [15]. The high and low resistive states are set to $R_{ON} = 500\ K\Omega$ and $R_{OFF} = 2\ M\Omega$ in the Stanford-PKU RRAM Model [15]. The RRAM technology exhibits switching variation which can be temporal (cycle-to-cycle) or spatial (device-to-device). The exhibited resistance variability at the device level forms the basis for our proposed PUF watermarking system. Maximum parameter variation of R_{OFF} and R_{ON} values for HfO_x/TiO_x RRAM devices are experimentally observed to be $\pm 20\%$ [1], [2]. Hence the maximum variation of R_{OFF} and R_{ON} is limited to $\pm 20\%$ for all the analyses considered in this paper.

The illustration of the proposed CeNN is shown with an example using smoothing operation and binary output cloning templates as in [22]. The network size is 4×4. The circuit parameters of the cell $C(m, n)$ include the cloning templates with values: $A = [0, 1, 0; 1, 2, 1; 0, 1, 0]$, $B = [0, 0, 0; 0, 0, 0; 0, 0, 0]$, $I = 0$, $C = 10^{-9} F$ and $R_c = 10\ \text{k}\Omega$. An initial capacitor voltage is set to 0 V. The state variable $x_{m,n}$ of 4×4 cellular neural network is presented in Figure 4(a) at $t = 5\ \mu$S. From Figure 4(a), the maximum absolute value of the state variable is 1.5, which is almost equal to the theoretical value $v_{\max} = 1.8$ calculated using the $v_{\max}$ equation. The transient behavior of one cell C(2,2) is shown in Figure 4(b). The initial value of $x_{2,2}$ is 1 with the maximum value of 1.5 at around $t = 1\ \mu$S. Since the cell state is always above 1 during the entire transient regime, the cell output, $y_{2,2}$, remains at 1 as shown in Figure 4(b). The magnitude of all stable equilibrium points is greater than 1; thereby, the condition of stability of CeNN network is also satisfied. Figure 4(c) and (d) show the state values of $x_{m,n}$ at $t = 5\ \mu$S and transient waveform of cell C(2,2) with 20% variation in R_{ON} and R_{OFF}. The result shows that the maximum absolute value of the state variable is 1.96 (Figure 4(c)) and the magnitude of the stable equilibrium point is 1.2, which is greater than 1 (Figure 4(d)).

Table II provides the area and power requirements of the proposed CeNN architecture. The 22 nm PTM High-K PTM models are used as transistor switches [35]. The on-chip area and power requirements of the proposed CeNN architecture are compared with an equivalent circuit implementation of CeNN in [33], [34] and [22]. The proposed CeNN cell is implemented using memristive crossbar arrays as in Figure 3. In [33] and [34], the equivalent CeNN cell resistance is represented using memristors whereas [22] presents the opamp-based implementation of the CeNN cell. The results show that the proposed CeNN architecture shows better area and power reduction by using memristive low-power devices. The power requirement increases with the size of the image.

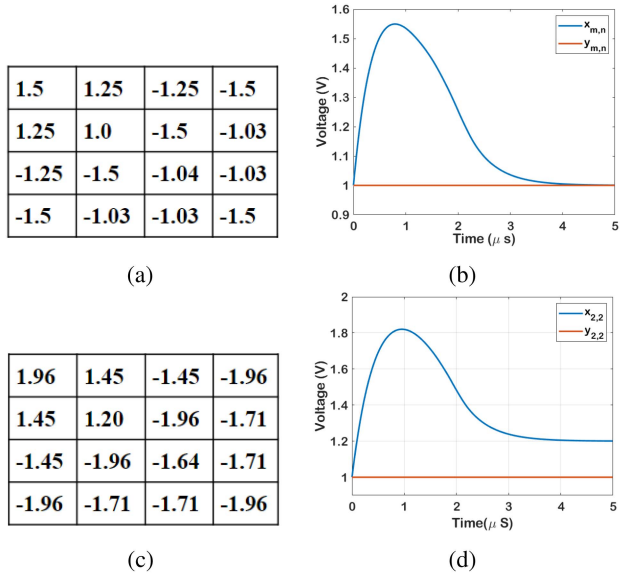


Figure 4. 4×4 Cellular Neural Network (a) Final state, $x_{m,n}$ at $t = 5\ \mu$S, ideal case (b) Transient waveform of cell C(2,2) (c) Final state, $x_{m,n}$ at $t = 5\ \mu$S, with 20% variation and (d) Transient waveform of cell C(2,2), with 20% variation.

TABLE II
On-Chip Area and Power Requirements of CeNN Networks: Comparison

Parameters	Memristive CeNN [33]	M-CeNN [34]	Ref [22]	Proposed memristive CeNN
Technology Node	22 nm CMOS	22nm CMOS	22nm CMOS	22nm CMOS & Stanford-PKU SPICE RRAM model [16]
Power (mW)				
3×3	39.51	16.47	10.15	3.30
28×28	3441	1430	884.35	235
64×64	17E3	7495	4620.28	1499
Area (μm^2)				
3×3	60.91	25.29	15.12	5.04
28×28	5306	2203	1317.12	439
64×64	0.027E6	0.011E6	6881.28	2293

A. Evaluation of CeNN-PUF Watermarking

The evaluation of the proposed CeNN-PUF watermarking system is done in the MATLAB environment. The equivalent high-level model of the proposed CeNN is designed and simulated in MATLAB. The Kaggle face2comics dataset is used for this study. It contains 10K paired face to comics data. In the face2comics dataset, the size of each image is 512×512 pixels. These images form the input of the CeNN arrays. Figure 5 shows the CeNN output of a single input image for conductance variation and template variation. The cloning templates used for

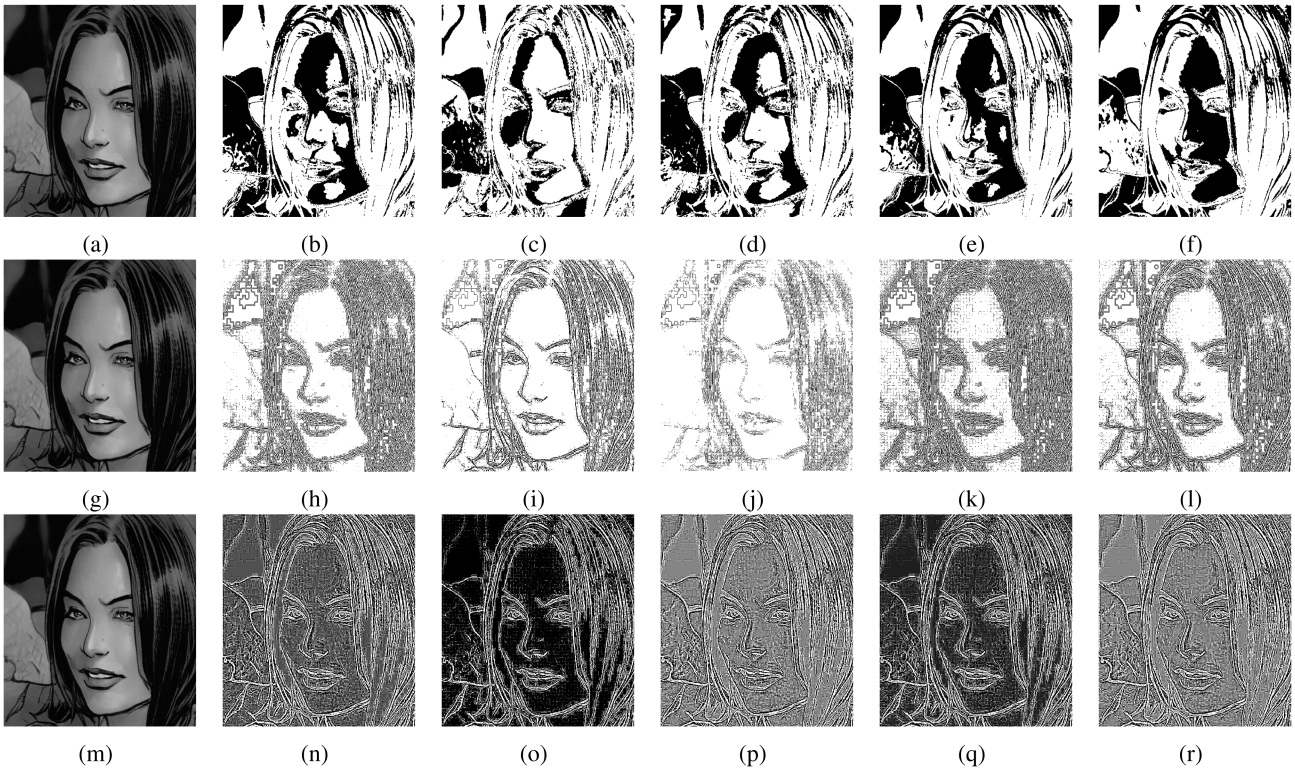


Figure 5. CeNN PUF output for the cloning template "Smoothing with binary output" (a) original image (b) CeNN output without variations (c) CeNN output with $R_{ON}-0.2R_{ON}, R_{OFF}-0.2R_{OFF}$ (d) CeNN output with $R_{ON}+0.2R_{ON}, R_{OFF}-0.2R_{OFF}$ (e) CeNN output with $R_{ON}-0.2R_{ON}, R_{OFF}+0.2R_{OFF}$ (f) CeNN output with $R_{ON}+0.2R_{ON}, R_{OFF}+0.2R_{OFF}$. CeNN PUF output for the cloning template "Edge Detection" (g) original image (h) CeNN output without variations (i) CeNN output with $R_{ON}-0.2R_{ON}, R_{OFF}-0.2R_{OFF}$ (j) CeNN output with $R_{ON}+0.2R_{ON}, R_{OFF}-0.2R_{OFF}$ (k) CeNN output with $R_{ON}-0.2R_{ON}, R_{OFF}+0.2R_{OFF}$ (l) CeNN output with $R_{ON}+0.2R_{ON}, R_{OFF}+0.2R_{OFF}$. CeNN PUF output for the cloning template "Half Toning" (m) original image (n) CeNN output without variations (o) CeNN output with $R_{ON}-0.2R_{ON}, R_{OFF}-0.2R_{OFF}$ (p) CeNN output with $R_{ON}+0.2R_{ON}, R_{OFF}-0.2R_{OFF}$ (q) CeNN output with $R_{ON}-0.2R_{ON}, R_{OFF}+0.2R_{OFF}$ (r) CeNN output with $R_{ON}+0.2R_{ON}, R_{OFF}+0.2R_{OFF}$.

TABLE III
SDIM OF PUF WATERMARKED IMAGES WITH DIFFERENT CLONING TEMPLATES

Variation	T1			T2			T3		
	ξ=0.2	ξ=0.1	ξ=0.02	ξ=0.2	ξ=0.1	ξ=0.02	ξ=0.2	ξ=0.1	ξ=0.02
$(R_{ON}+\xi R_{ON}, R_{OFF}+\xi R_{OFF})$	0.535	0.255	0.063	0.264	0.204	0.120	0.716	0.628	0.590
$(R_{ON}+\xi R_{ON}, R_{OFF}-\xi R_{OFF})$	0.191	0.102	0.033	0.210	0.107	0.098	0.715	0.543	0.431
$(R_{ON}-\xi R_{ON}, R_{OFF}+\xi R_{OFF})$	0.167	0.093	0.034	0.125	0.077	0.057	0.717	0.578	0.434
$(R_{ON}-\xi R_{ON}, R_{OFF}-\xi R_{OFF})$	0.412	0.239	0.062	0.263	0.220	0.148	0.713	0.543	0.321

T1: Smoothing with binary output template, T2: Edge detection template, T3: Half Toning Template.

processing are smoothing with binary output, edge detection, and half-toning. The cloning templates for edge detection are: $A=[0,0,0;0,1,0;0,0,0]$, $B=[-1,-1,-1;-1,8,-1;-1,-1,-1]$, $I=-1$ [36]. The cloning templates for half toning are: $A=[-0.07,-0.1,-0.07;-0.1,1,-0.1;-0.07,-0.1,-0.07]$, $B=[0.07,0.1,0.07;0.1,0.32,0.1;0.07,0.1,0.07]$, $I=-0$ [36]. The figure compares the CeNN output resistance variation cases like $(R_{ON}+\xi R_{ON}, R_{OFF}+\xi R_{OFF})$, $(R_{ON}+\xi R_{ON}, R_{OFF}-\xi R_{OFF})$, $(R_{ON}-\xi R_{ON}, R_{OFF}+\xi R_{OFF})$ and $(R_{ON}-\xi R_{ON}, R_{OFF}-\xi R_{OFF})$ with 20% variation. The results show that the resistance variations can generate unique PUF feature images that can be used for watermarking. The randomness in the resistance variations generates unique feature images using CeNN architecture for different cloning templates.

The CeNN output is watermarked with the original image. The uniqueness of the watermarked image is analyzed by calculating the Structural Dissimilarity (SDIM) Index of the watermarked image, z_j, in comparison with the watermarked image without variation (ideal case), z_I (Table III). The SDIM can be calculated using the equation of Structural Similarity (SSIM) Index [37]. The equation for SDIM can be written as follows:

$$SDIM=\frac{1}{2}-\frac{\left(2\mu_{z_j}\mu_{z_I}+C_1\right)\left(2\sigma_{z_jz_I}+C_2\right)}{2\left(\mu_{z_j}^2+\mu_{z_I}^2+C_1\right)\left(\sigma_{z_j}^2+\sigma_{z_I}^2+C_2\right)} \quad (6)$$

where μ_{z_j/z_I} denotes the mean of z_j/z_I, σ_{z_j/z_I} denotes the variance of z_j/z_I and $\sigma_{z_jz_I}$ denotes the covariance of z_j and z_I. Here, we use $C_1=0$ and $C_2=0$ as in [37]. $SDIM\rightarrow 1$ implies

TABLE IV
WATERMARKING TECHNIQUES: A COMPARISON

Method	Image (Size: 512×512)	PSNR (dB)
DCT based watermarking [38]	Lena Image	47.18
DFT based watermarking [39]	Lena Image	53.3
DWT based watermarking [40]	Lena Image	50.90
SVD based watermarking [41]	Lena Image	42.63
CeNN-PUF Smoothing template	Lena Image	59.78
	Face2comics dataset	63.59
CeNN-PUF Edge detection template	Lena Image	59.76
	Face2comics dataset	63.00
CeNN-PUF Half Toning Template	Lena Image	62.80
	Face2comics dataset	63.71

DCT: Discrete Cosine Transform, DFT: Discrete Fourier Transform, DWT: Discrete Wavelet Transform and SVD: Singular Value Decomposition.

strong dissimilarity and $SDIM \rightarrow 0$ implies weak dissimilarity. Because of the device variability, CeNN output shows variations with the three considered cloning templates for smoothing with binary output, edge detection, and halftoning. The same pattern of variations is captured with respect to watermarked image output. The result also shows that the dissimilarity index is maximum for $(R_{ON} - \xi R_{ON}, R_{OFF} - \xi R_{OFF})$ and $(R_{ON} + \xi R_{ON}, R_{OFF} + \xi R_{OFF})$ in comparison with other two cases. The same pattern is repeated for cloning templates T1, T2, T3 and all values of ξ. This is because $(R_{ON} - \xi R_{ON}, R_{OFF} - \xi R_{OFF})$ and $(R_{ON} + \xi R_{ON}, R_{OFF} + \xi R_{OFF})$ show the minimum and maximum bound of resistive state change for ϵ % variation.

Table IV compares the proposed CeNN-PUF watermarking with existing various watermarking techniques [38], [39], [40], [41]. The greyscale Lena image of size 512×512 is used for PSNR comparison. The PSNR is calculated as $10 \log_{10}(max_I/MSE)$, where max_I is the maximum pixel value of the original image and MSE is the cumulative squared error between the watermarked image and the original image [11]. Table IV shows that CeNN PUF has improved PSNR value compared with other watermarking techniques in the literature. The watermarked image quality of CeNN-PUF is better and combines the advantage of PUF security, implemented in the spatial domain.

a) Hardware Implementation: The practical demonstration of CeNN-PUF is done using myRIO FPGA. The myRIO hardware consists of ARM microcontroller and Xilinx Zynq-7010 (FPGA processor). The FPGA target is programmed using LABVIEW FPGA, a software add-on model to the LABVIEW graphical software development environment. The input image, feedback template, and feedforward template are stored in the FIFO memory with depths $M \times N$, 9 and 9 respectively. The stored feedback and feedforward template values are the conductance state values of memristor crossbar arrays. The resistance variations $(R_{ON} \pm \xi R_{ON}, R_{OFF} \pm \xi R_{OFF})$ are stored in FIFO for generating the PUF feature images. The input data is read from the FIFO for CeNN computation in generating feature masks. The output feature image is embedded into the original image for generating a PUF watermarked image. Table V shows the relative quantization error of FPGA output in comparison with software results. The error can be calculated as $\frac{|Ys_l - Yh_l|}{Ys_l}$, where Ys_l and Yh_l denote the software and hardware responses. The

TABLE V
AVERAGE RELATIVE ERROR, RE_{avg}, OF HARDWARE OUTPUT IN COMPARISON WITH SOFTWARE RESULTS WITH VARYING FIFO INTEGER LENGTH. T1: SMOOTHING WITH BINARY OUTPUT TEMPLATE, T2: EDGE DETECTION TEMPLATE T3: HALF TONING TEMPLATE

Template	Integer Word Length				
	2 bits	4 bits	8 bits	12 bits	16 bits
T1	0.46	0.34	0.23	0.13	0.03
T2	0.43	0.22	0.11	0.06	0.01
T3	0.45	0.28	0.17	0.09	0.03

T1: Smoothing with binary output template, T2: Edge detection template T3: Half Toning Template

table shows the relative output error for varying FIFO integer word lengths on myRIO. The experiment is repeated for various cloning templates, T1: Smoothing with binary output template, T2: Edge detection template and T3: Half Toning Template. The results show that 16-bit representation shows comparable performance with software results with minimum error. Hence the FIFOs for FPGA implementation are defined for a word length of 16 bits.

B. PUF Evaluation

The quality of CeNN-PUF is evaluated in terms of uniqueness, uniformity, randomness, reliability and unpredictability. These figures of merits are analyzed in the following experiments.

Uniqueness: Uniqueness measures how much the responses generated by one CeNN-PUF are different from the other. For CeNN-PUF, the uniqueness of the watermarked images is analyzed by calculating the average inter-Hamming Distance (HD). Let z_u and z_v be two different CeNN-PUF watermarked responses with the same input image, the uniqueness is calculated as [42]:

$$Uniqueness = \frac{2}{m(m-1)} \sum_{u=1}^{m-1} \sum_{v=u+1}^{m} \frac{HD(z_u, z_v)}{MN} \tag{7}$$

where the function $HD(z_u, z_v)$ evaluates the Hamming distance and m denotes the total number of responses. Table VI gives the uniqueness measure estimated for the proposed CeNN-PUF and compares it with existing works in the literature. The uniqueness here is estimated by simulating the CRPs generated from a large number of PUF instances in MATLAB. The 10,000 PUF instances with 120 CRPs for each instance are generated using images in the face2comics dataset. Figure 6(a) shows the frequency distribution of the HD of generated PUF instances. The values in Table VI are based on these CRPs generated for T1, T2 and T3 templates. The uniqueness values calculated using (7) are 49.64%, 50.77%, and 49.35%, which are very close to the ideal value of 50%.

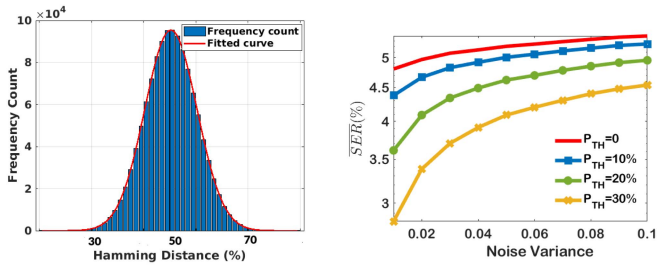
Uniformity: Uniformity is the indicator to estimate the similarity of symbols in PUF responses. Uniformity is computed using the average of the hamming weight of each response, z_u, using the formula [6]:

$$Uniformity = \frac{1}{M \times N} \sum_{m=1}^{M} \sum_{n=1}^{N} z_{u,m,n} \tag{8}$$

TABLE VI
PUF PERFORMANCE COMPARISON

PUF	Uniqueness(%) (Ideal 50%)	Uniformity (%) (Ideal 50%)	Randomness(%) (Ideal 100%)	Reliability (%), Ideal 100%	
				Temperature Variation	Supply Voltage variation
Arbiter-PUF [43]	7.2	55.7	84.7	99.8	
Ref [44]	47.0	40-60	75.7	-	
Ref [42]	49.37	-	-	88 ($P_{TH}=0$), 99.1 ($P_{TH}=30$)	90 ($P_{TH}=0$), 99 ($P_{TH}=30$)
Ref [45]	38.0	54	89.3	75.8	
SBC-PUF [6]	48.1	50.3	98.9	99.9	
CeNN-PUF					
T1	49.64	49.5	98.8	*97.5 ($P_{TH}=0$), *99.5 ($P_{TH}=30$)	**90 ($P_{TH}=0$), **99 ($P_{TH}=30$)
T2	50.77	52	94.4	*96.8 ($P_{TH}=0$), *99.1 ($P_{TH}=30$)	**92 ($P_{TH}=0$), **99.2 ($P_{TH}=30$)
T3	49.35	49.2	97.5	*97.4 ($P_{TH}=0$), *99.5 ($P_{TH}=30$)	**91 ($P_{TH}=0$), **99.1 ($P_{TH}=30$)
				* measured at 330K	**measured at 2.2V

T1: Smoothing with binary output template, T2: Edge detection template T3: Half Toning Template.



(a) (b)

Figure 6. (a) Frequency distribution of the simulated Hamming Distances for different CeNN-PUF instances and (b) Mean of Symbol error rate (SER) with noise for T1: Smoothing with binary output template.

TABLE VII
SDIM AND PSNR OF PUF WATERMARKED IMAGES WITH NOISE FOR DIFFERENT CLONING TEMPLATES

Variance σ^2	SDIM			PSNR (dB)		
	T1	T2	T3	T1	T2	T3
$(R_{ON}+\xi R_{ON}, R_{OFF}+\xi R_{OFF}), \xi = 0.2$						
0.01	0.553	0.294	0.769	39.15	39.12	39.88
0.20	0.583	0.336	0.791	32.44	32.38	33.39
0.40	0.603	0.346	0.811	26.04	26.02	26.09
0.60	0.631	0.367	0.839	18.06	18.08	19.15
0.80	0.653	0.379	0.854	13.78	13.46	8.83
1	0.685	0.394	0.873	11.46	11.51	2.90

T1: Smoothing with binary output template, T2: Edge detection template T3: Half Toning Template.

Table VI shows the uniformity values of CeNN-PUF in comparison with other works in the literature. The results show that the uniformity values vary with image processing applications but are almost close to the ideal value of 50%.

Randomness: Entropy is a statistical measure used to characterize the randomness of PUF. The randomness of CeNN-PUF can be computed as [6]:

$$Randomness = -\sum_{n} p_n \log_2 p_n \tag{9}$$

where p_n denotes the entropy probability of each pixel in the PUF feature image. Table VI shows the randomness of proposed CeNN-PUF for different cloning templates.

Reliability: The CeNN-PUF system parameters also vary with PVT variations. The section evaluates the reliability of the proposed system against different environmental changes, typically temperature, supply voltage and noise. The reliability measures the stability of the proposed CeNN-PUF system under different operating conditions. For noisy environments, the statistic mean of the Symbol Error Rate ($\overline{SER}$) is taken as the evaluation parameter. We consider the responses in the noisy region to be unstable as in [42]. The responses with values between $-P_{TH}$ and $+P_{TH}$ are stable and others are discarded for evaluating the SER, where $\pm P_{TH}$ is the threshold value which can be determined based on the target fabrication process. $\overline{SER}$ can be calculated as:

$$\overline{SER} = \frac{1}{MN}\sum_{k=1}^{MN} Pr\left(|u_{in}| - |P_{TH}| > 0\right) \tag{10}$$

i.e., an error occurs when the responses lying in the range $-P_{TH}$ and $+P_{TH}$ are discarded. The reliability measures the stability of the CeNN-PUF system under different operating conditions [42]. The reliability of the proposed system against temperature and supply voltage variations is directly measured by comparing the ideal case reference response with the responses taken by external parameter variations [42].

The watermarked image may be introduced by noise because of various external factors. This noise is introduced in real time because of errors in data transfers. Table VII shows the evaluation of the proposed watermarked image in terms of SDIM and Peak-Signal-to-Noise-Ratio (PSNR) with variation in noise variance. The T1, T2, and T3 cases are presented for the maximum variation case $(R_{ON}+\xi R_{ON}, R_{OFF}+\xi R_{OFF})$ with $\xi = 20\%$. As the noise variance increases dissimilarity index increases but the PSNR reduces considerably. The result shows that the uniqueness of the images in terms of SDIM is maintained in the presence of noise.

In addition to $SDIM$ and $PSNR$, $\overline{SER}$ is also a parameter for the evaluation of reliability. Figure 6(b) shows the $\overline{SER}$ of CeNN PUF against noise. The figure shows the average error in symbol detection with various threshold values, P_{TH} under noise. The figure shows that $\overline{SER}$ decreases with increasing $|P_{TH}|$ values. i.e, when the noise margin $|P_{TH}|$ increases, the symbol detection rate increases and $\overline{SER}$ reduces. From Table VII and Figure 6(b), it is understood that there should be a tradeoff between $SDIM$, $PSNR$ and $\overline{SER}$ in determining the performance of CeNN-PUF for various applications.

The reliability measurement of CeNN-PUF against temperature and supply voltage variations are evaluated and presented in Figs. 7 and (a) respectively for the T1 template (Smoothing

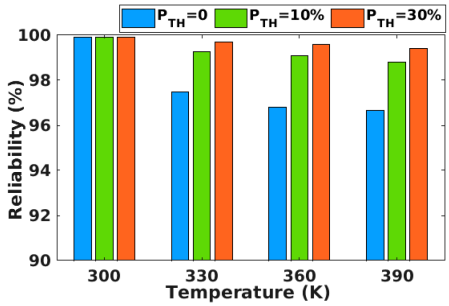
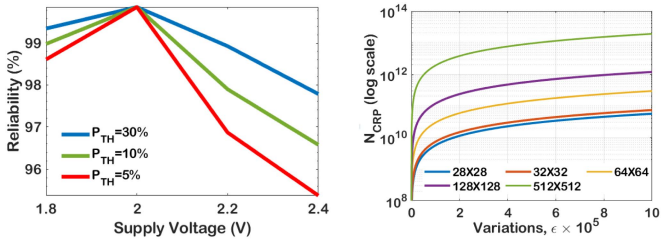


Figure 7. Reliability of CeNN-PUF against temperature variations for T1: Smoothing with binary output template.



(a) (b)

Figure 8. (a) The measured average reliability of CeNN-PUF against voltage variations for T1: Smoothing with binary output template, (b) Number of CRPs with resistance variation, ϵ.

with binary output). Reliability shows whether the system is reproducible or stable under different operating conditions and is measured. For the analysis, the operating temperature and supply voltage are increased by varying the behavioral model of Stanford-PKU SPICE RRAM. The working temperature is varied from 300 K to 400 K which is the critical temperature of the Stanford-PKU SPICE RRAM model. The response at 300 K is the reference. The supply voltage is varied from 1.8 V to 2.4 V for analysis and the nominal supply read voltage is 2 V. Here the reliability is calculated by comparing the response image at different temperatures or supply voltage variation to the reference image [42]. The results show that the reliability value peaks at reference values, at 300 K for Figure 7 and 2 V for Figure 8(b), and decreases when moving away from the reference values. The results also show that reliability increases with an increase in P_{TH} values. Table VI compares the reliability of CeNN-PUF with other works in the literature. For the proposed CeNN-PUF method, the reliability is measured at a temperature variation of 330 K and supply voltage variation at 2.2 V. The results show that CeNN-PUF is prone to voltage supply and temperature fluctuations but the effect can be minimized by increasing $|P_{TH}|$ values.

C. ML Attack Analysis

This section presents the ML attack analysis of the proposed CeNN-PUF. Figure 8(b) shows the possible number of responses with resistance variations, ϵ. The memristor can have the values $R_r(m,n;k,l) \pm \epsilon(m,n;k,l)$, where $\epsilon(m,n;k,l) = |R_r(m,n;k,l) - R_I(m,n;k,l)|$, where R_r and R_I are the real resistance state value and ideal value respectively. ϵ will be different for different types of devices. The figure shows the variation with various CeNN rectangular grid sizes. Increasing ϵ and grid sizes exponentially increase the number of challenge-response pairs. The CeNN grid size is dependent on the input image size. The massive aggregation of cells in CeNN causes an exponential increase in the count of responses.

TABLE VIII
CMA-ES ATTACK MODEL ANALYSIS FOR CENN-PUF: TRAINING TIME AND ACCURACY FOR DIFFERENT DATASETS

MNIST			CIFAR10		
#CRPs	Training Time (hours)	Accuracy (%)	#CRPs	Training Time (hours)	Accuracy (%)
1 × S	0.2	98.8	1 × T	0.3	96.9
5 × S	0.97	97.5	5 × T	3.1	96.1
10 × S	2	97.8	10 × T	4.25	95.4
30 × S	3.95	97.1	30 × T	9.5	95.6

$S = 60000$ for MNIST and $T = 50000$ for CIFAR10.

According to the CMA-ES attack model, the attacker will generate random PUF instances and check the model accuracies to find the nearest model for CeNN-PUF [30]. The attacker will generate the number of responses equal to the number of resistance variation possibilities of memristors in a CeNN crossbar array. The dataset is then used to train a neural network model to find the nearest model for CeNN-PUF. Here we use the MNIST and CIFAR10 datasets to demonstrate ML attack modelling. The MNIST dataset contains 60,000 28×28 grayscale training images and CIFAR10 with 50,000 32×32 color training images. Different PUF images are generated using the training images. The training time for MNIST and CIFAR10 with varying dataset sizes are shown in Table VIII. The Convolutional Neural Network (CNN) architecture, with convolution layers, mean pooling and two dense layers, is used for classification. For MNIST, the convolutional layer consists of 28 filters with size 3×3 and for CIFAR10 3 convolutional layers with filter sizes 32, 64 and 64 respectively. Table VIII shows the time for training the model for around 98% and 96% accuracy for MNIST and CIFAR respectively.

According to Table VIII, the projected training time for MNIST and CIFAR datasets for 10^{10} responses is $\sim 10^7$ days and $\sim 10^8$ days respectively. The training time again increases with an increase in the number of training dataset sizes. A larger training dataset may decrease the accuracy since the likelihood of overfitting the data is increased. Hence as the training dataset increases the accuracy may reduce as shown in Table VIII. This makes the proposed PUF more resilient to ML attacks.

Additional discussion: The proposed method shows the capability to generate device-specific feature images. It ensures the authenticity during data transfers, that prevents identity forgery while registering the PUF devices. However, in situations when modification occur after watermarking there are still some open challenges. The CeNN-PUF method struggles to reliably verify the watermark if compression or cropping is applied after the watermark is generated. How to ensure watermark integrity post-watermarking modified images is an open challenge. However, this is less of an issue in on-chip hardware based machine to

machine systems where image manipulations are difficult to make.

In can be noted that PUF-based data can be used for addressing the issues of tampered image detection in case of scaling, compression, and other transformations. This idea can be further explored to create a CeNN-PUF framework that is tolerant to post-watermarking alterations. There could be other approaches that can be further explored to enhance the reliability of proposed CeNN-PUF system, including integrating with techniques such as (1) comparing image hashes (based on PUF features) from the original and transformed images, (2) embedding redundant PUF-derived watermarks in multiple regions that has limited possibility of tampering, (3) degraded watermark can be attempted to be recovered using machine learning on side-channel data (e.g., metadata), and (4) the watermarks can be embedded at multiple resolutions so that even after compression at least one resolution survives - here hierarchical embedding structure with coarse watermarks can be used for global transformations (e.g., scaling) and fine-grained watermarks can be used for localized alterations (e.g., compression artifacts).

V. Conclusion

A memristor-crossbar based PUF image watermarking system is proposed in this paper. The proposed method integrates memristive-crossbar cellular neural network architecture for the generation of PUF feature images. The proposed method is capable of ensuring data security and device authentication by generating device-specific feature masks for image watermarking. The proposed crossbar based CeNN architecture put forward various possibilities in the field of analog PUF designs. The possibility of varying the cloning template makes the CeNN architecture flexible for various target applications. The proposed scheme covers various PUF characteristics like uniqueness, randomness, and reliability against different environmental changes. The security of the CeNN-PUF against ML attacks is analyzed. The CeNN-PUF hardware architecture can simultaneously perform image processing tasks and PUF image generation thus reducing the overheard on additional circuitry for ensuring security.

References

[1] G. Yansong, R. D. C. Ranasinghe, S. F. Al-Sarawi, O. Kavehei, and D. Abbott, "Memristive crypto primitive for building highly secure physical unclonable functions," *Sci. Rep.*, vol. 5, 2015, Art. no. 12785.

[2] M. Uddin, M. B. Majumder, and G. S. Rose, "Robustness analysis of a memristive crossbar PUF against modeling attacks," *IEEE Trans. Nanotechnol.*, vol. 16, no. 3, pp. 396–405, May 2017.

[3] G. S. Rose, N. McDonald, L.-K. Yan, and B. Wysocki, "A write-time based memristive PUF for hardware security applications," in *Proc. 2013 IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2013, pp. 830–833.

[4] G. S. Rose and C. A. Meade, "Performance analysis of a memristive crossbar PUF design," in *Proc. 52nd ACM/EDAC/IEEE Des. Automat. Conf.*, 2015, pp. 1–6.

[5] O. Kavehei, C. Hosung, D. Ranasinghe, and S. Skafidas, "mrPUF: A memristive device based physical unclonable function," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, Feb. 2012, pp. 595–615.

[6] D. Kim et al., "Selected bit-line current PUF: Implementation of hardware security primitive based on a memristor crossbar array," *IEEE Access*, vol. 9, pp. 120901–120910, 2021.

[7] J. Mathew, R. S. Chakraborty, D. P. Sahoo, Y. Yang, and D. K. Pradhan, "A novel memristor-based hardware security primitive," *ACM Trans. Embedded Comput. Syst.*, vol. 14, no. 3, Apr. 2015, Art. no. 60.

[8] A. Mazady, M. T. Rahman, D. Forte, and M. Anwar, "Memristor PUF—A security primitive: Theory and experiment," *IEEE Trans. Emerg. Sel. Topics Circuits Syst.*, vol. 5, no. 2, pp. 222–229, Jun. 2015.

[9] R. John, N. Shah, and S. e. a. Vishwanath, "Halide perovskite memristors as flexible and reconfigurable physical unclonable functions," *Nature Commun.*, vol. 12, no. 3681, 2021.

[10] M. I. Khan, S. Ali, A. A. Ikram, and A. Bermak, "Optimization of memristive crossbar array for physical unclonable function applications," *IEEE Access*, vol. 9, pp. 84480–84489, 2021.

[11] K. Sehra et al., "Robust and secure digital image watermarking technique using Arnold transform and memristive chaotic oscillators," *IEEE Access*, vol. 9, pp. 72465–72483, 2021.

[12] L. Zhang, X. Fong, C.-H. Chang, Z. H. Kong, and K. Roy, "Optimizating emerging nonvolatile memories for dual-mode applications: Data storage and key generator," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 34, no. 7, pp. 1176–1187, Jul. 2015.

[13] T. Addabbo, A. Fort, M. Di, L. MarcoPancioni, and V. Vignoli, "Physically unclonable functions derived from cellular neural networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 12, pp. 3205–3214, Dec. 2013.

[14] H. Takalo, A. Ahmadi, M. Mirhassani, and M. Ahmadi, "Analog cellular neural network for application in physical unclonable functions," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2016, pp. 2635–2638.

[15] H. Li et al., "Variation-aware, reliability-emphasized design and optimization of RRAM using SPICE model," in *Proc. 2015 Des. Automat. Test Europe Conf. Exhib.*, 2015, pp. 1425–1430.

[16] X. Guan, S. Yu, and H.-S. P. Wong, "A SPICE compact model of metal oxide resistive switching memory with variations," *IEEE Electron Device Lett.*, vol. 33, no. 10, pp. 1405–1407, Oct. 2012.

[17] A. Babaei and G. Schiele, "Physical unclonable functions in the Internet of Things: State of the art and open challenges," *Sensors*, vol. 19, no. 14, 2019, Art. no. 3208.

[18] J. Zhang and G. Qu, "Physical unclonable function-based key sharing via machine learning for IoT security," *IEEE Trans. Ind. Electron.*, vol. 67, no. 8, pp. 7025–7033, Aug. 2020.

[19] S. Bang, B. Sheu, and E. Chou, "A hardware annealing method for optimal solutions on cellular neural networks," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, no. 6, pp. 409–421, Jun. 1996.

[20] S. Bang, B. Sheu, and T.-Y. Wu, "Optimal solutions for cellular neural networks by paralleled hardware annealing," *IEEE Trans. Neural Netw.*, vol. 7, no. 2, pp. 440–454, Mar. 1996.

[21] X. Gao, C. Deng, X. Li, and D. Tao, "Geometric distortion insensitive image watermarking in affine covariant regions," *IEEE Trans. Syst., Man, Cybern. C. Appl. Rev.*, vol. 40, no. 3, pp. 278–286, May 2010.

[22] L. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits Syst.*, vol. CS-35, no. 10, pp. 1257–1272, Oct. 1988.

[23] Q. Lou, I. Palit, T. Li, A. Horvath, M. Niemier, and X. S. Hu, "Application-level studies of cellular neural network-based hardware accelerators," Feb. 2019, *arXiv:1903.06649*.

[24] S. Duan, X. Hu, Z. Dong, L. Wang, and P. Mazumder, "Memristor-based cellular nonlinear/neural network: Design, analysis, and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1202–1213, Jun. 2015.

[25] M. Itoh and L. Chua, "Memristor cellular automata and memristor discrete-time cellular neural networks," in *Memristor Networks*, A. Adamatzky and L. Chua, Eds. Cham: Springer, 2014, pp. 649–713.

[26] A. Gasparetto, K. Xie, Y. Yang, Y. Xin, and G. Xia, "Cellular neural network-based methods for distributed network intrusion detection," in *Mathematical Problems in Engineering*, London, U.K.: Hindawi Publishing Corporation, 2015, pp. 1–10.

[27] O. Krestinskaya, B. Choubey, and A. P. James, "Memristive GAN in analog," *Sci. Rep.*, vol. 10, no. 1, 2020, Art. no. 5838.

[28] M. Di Marco, M. Forti, and L. Pancioni, "Memristor standard cellular neural networks computing in the flux–charge domain," *Neural Netw.*, vol. 93, pp. 152–164, 2017.

[29] C.-H. Chang, Y. Zheng, and L. Zhang, "A retrospective and a look forward: Fifteen years of physical unclonable function advancement," *IEEE Circuits Syst. Mag.*, vol. 17, no. 3, pp. 32–62, 2017.

[30] G. T. Becker, "The gap between promise and reality: On the insecurity of XOR arbiter PUFs," in *Proc. Int. Workshop Cryptographic Hardware Embedded Syst.*, 2015, pp. 535–555.

[31] J. Shi, Y. Lu, and J. Zhang, "Approximation attacks on strong PUFs," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2138–2151, Oct. 2020.
[32] A. P. James and L. O. Chua, "Variability-aware memristive crossbars—A tutorial," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 6, pp. 2570–2574, Jun. 2022.
[33] C. Xiu, R. Zhou, and Y. Liu, "New chaotic memristive cellular neural network and its application in secure communication system," *Chaos, Solitons Fractals*, vol. 141, 2020, Art. no. 110316.
[34] B. Xu, H. Lin, and G. Wang, "Hidden multistability in a memristor-based cellular neural network," *Adv. Math. Phys.*, vol. 2020, no. 1, 2020, Art. no. 9708649.
[35] *22 nm PTM model for metal gate high-K CMOS: V2.0*, 2007. [Online]. Available: http://ptm.asu.edu/
[36] M. Itoh and L. O. Chua, "Memristor cellular automata and memristor discrete-time cellular neural networks," *Int. J. Bifurcation Chaos*, vol. 19, no. 11, pp. 3605–3656, 2009.
[37] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals Syst. Comput.*, 2003, vol. 2, 2003, pp. 1398–1402.
[38] A. Abdulrahman and S. Ozturk, "A novel hybrid DCT and DWT based robust watermarking algorithm for color images," *Multimed Tools Appl*, vol. 78, pp. 17027–17049, 2019.
[39] A. Poljicak, L. Mandic, and D. Agic, "Discrete Fourier transform-based watermarking method with an optimal implementation radius," *J. Electron. Imag.*, vol. 20, no. 3, 2011, Art. no. 033008.
[40] P. Garg and R. Kishore, "Performance comparison of various watermarking techniques," *Multimed Tools Appl*, vol. 79, pp. 25921–25967, 2020.
[41] P. Garg and R. R. Kishore, "Secured and multi optimized image watermarking using SVD and entropy and prearranged embedding locations in transform domain," *J. Discrete Math. Sci. Cryptogr.*, vol. 23, no. 1, pp. 73–82, 2020.
[42] Y. Cao, L. Zhang, S. S. Zalivaka, C.-H. Chang, and S. Chen, "CMOS image sensor based physical unclonable function for coherent sensor-level authentication," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 11, pp. 2629–2640, Nov. 2015.
[43] Y. Hori, T. Yoshida, T. Katashita, and A. Satoh, "Quantitative and statistical performance evaluation of arbiter physical unclonable functions on fpgas," in *Proc. 2010 Int. Conf. Reconfigurable Comput. FPGAs*, 2010, pp. 298–303.
[44] A. Chen, "Reconfigurable physical unclonable function based on probabilistic switching of RRAM," *Electron. Lett.*, vol. 51, no. 8, pp. 615–617, 2015.
[45] G. S. Lee, G.-H. Kim, K. Kwak, D. S. Jeong, and H. Ju, "Enhanced reconfigurable physical unclonable function based on stochastic nature of multilevel cell RRAM," *IEEE Trans. Electron Devices*, vol. 66, no. 4, pp. 1717–1721, Apr. 2019.

Alex James (Senior Member, IEEE) received the PhD degee from Griffith University, Queensland, Australia. He is currently a professor and the dean external linkages with the Digital University Kerala India. His research interests include AI-neuromorphic systems (software and hardware), VLSI and image processing. He is a member of IET Vision and Imaging Network, and BCS' Fellows Technical Advisory Group (F-TAG). He was an Editorial Board Member of Information Fusion (2010–2014), and currently serving as an associate editor of IEEE Access (2017-present), Frontiers in Neuroscience (2022-present), IEEE TCAS 1: Regular Papers (2018-2023) and IEEE OJCAS (2022-present). He is currently the associate EIC of IEEE OJCAS, associate editor for IEEE TBioCAS and IEEE TCASAI. He was awarded IEEE Outstanding Researcher by IEEE Kerala Section for 2022, Kairali Scientist Award for Physical Science in 2021, Best Associate Editor for TCAS1 in 2021 and and 2024 *IEEE Transactions on Circuits and Systems Guillemin-Cauer* Best Paper Award.

Chithra Reghuvaran (Senior Member, IEEE) received the PhD degree from the National Institute of Technology Rourkela, India. She currently serves the position of research engineer with the School of Electrical and Electronic Engineering, University College Dublin. She works on Sensor Edge Intelligent computing, hardware realization of neuromorphic systems, memristive systems, and related areas. She is the recipient of the Erasmus Mundus India-Europe Action-2 HERITAGE Scholarship funded by the European Union. She has undergone a research internship under the Erasmus Student Mobility Program at Czech Technical University in Prague, Czech Republic. She has published many high-quality research articles in journals and conferences in the areas of analog memristive neural accelerator design for sensor edge intelligence, power optimization, etc.

Leon Chua (Fellow, IEEE) is widely known for his invention of the Memristor. His research has been recognized through 17 honorary doctorates from major universities in Europe and Japan, and holds seven US patents. He was a Foreign Member of the European Academy of Sciences (Academia Europaea) in 1997, and a Foreign Member of the Hungarian Academy of Sciences in 2007. He was conferred numerous prestigious awards, including the first Kirchhoff Award, the Guggenheim Fellow, the 2019 EDS Celebrated Member Prize– the highest recognition of the IEEE Electron Devices Society, and the 2020 Julius Springer Prize in applied physics. Prof. Chua was elected Confrerie des Chevaliers du Tastevin in 2000. When not immersed in Science, he relaxes by searching for Wagner's leitmotifs, musing over Kandinsky's chaos, and contemplating Wittgenstein's inner thoughts.