Neuron-PUF: Physical Unclonable Function Based on a Single Spiking Neuron

Mohamed Elshamy, Haralampos-G. Stratigopoulos

Neuron-PUF: Physical Unclonable Function Based on a Single Spiking Neuron

Mohamed Elshamy and Haralampos-G. Stratigopoulos
Sorbonne Université, CNRS, LIP6, Paris, France

***Abstract*—We propose a novel Physical Unclonable Function (PUF) concept based on a single spiking neuron. The inherent variability of the neuron results in a chip-unique analog spiking pattern that is digitized to produce a chip-unique digital key. A stability booster is also employed based on self-masking to obtain a fully stable digital key. The PUF is area and power effective since a single PUF cell is used to produce an arbitrarily sized digital key. We demonstrate PUF quality metrics close to ideal values and argue that the PUF is resilient against various physical attacks.**

I. Introduction

Physical Unclonable Functions (PUFs) are a class of hardware security primitives that find several applications [1], [2]. A PUF is a circuit that leverages statistical manufacturing variations of circuit parameters to generate a chip-unique signature. When queried with an input, referred to as *challenge*, it generates an output, referred to as *response*, that typically is a bitstring composing a digital key. Ideally, a PUF should be reliable, i.e., in the presence of temperature and voltage variations, noise, and aging the key-bits should be stable. It should also be random providing a chip-unique response, i.e., PUFs into any two different chips should give different responses for the same challenge. An additional demand is resiliency against attacks, i.e., modeling attacks that leverage machine learning and side-channel information to model the PUF behavior [3], [4] and memory attacks aiming at stealing the key [5]. PUFs are also characterized by the energy per bit and area per bit, which are especially important metrics for PUFs embedded in resource-constrained devices.

PUF applications include among others device authentication, secret key generation, hardware anti-piracy, security in Internet-of-Things (IoT) devices, etc. In device authentication, the PUF is used as a silicon biometric to generate a unique fingerprint or ID per chip [6], [7]. In secret key generation, a PUF is used to generate the key on-the-fly at power-on, thus avoiding explicit key storage [6]. In hardware anti-piracy, the PUF can be used to provide each chip an ID such that it can be traced along its lifetime for anti-counterfeiting purposes [8], [9]. In addition, a PUF can be used in the key management scheme of chip locking techniques [10]–[13]. In a resource-constrained smart IoT edge device, a PUF can be used for lightweight low-cost authentication protocols [14], [15].

The PUF concept was originally introduced in [16]–[18] and several silicon PUF implementations have been proposed since then. The most popular PUFs are delay-based PUFs, such as the arbiter [18]–[21] and ring oscillator (RO) [6], [18], [22] PUFs, and memory-based PUFs, such as the SRAM [23]–[25], flip-flop [26], [27], and latch [28], [29] PUFs. Delay-based PUFs exploit some race condition that is built-up inside the circuit, while memory-based PUFs exploit the natural tendency of a memory cell towards one of its two states. There exist several variations of those PUFs, as well as several other PUF designs based on different concepts [30]–[38].

No PUF is inherently robust and a percentage of PUF cells may generate unstable bits that should be handled accordingly. Stability boosting techniques include temporal majority voting to stabilize noisy bits, burn-in hardening to accelerate aging, self-masking of "dark" bits that are unstable across varying operating conditions, and error correcting codes (ECCs) circuits [35], [39]–[42]. For this reason, PUFs typically generate an excess number of bits which can be thereafter down-sampled to a fully stable key.

Another categorization of PUFs takes into consideration the number of challenge-response pairs (CRPs) that the PUF can support. A PUF that can support only a small number of CRPs is called a *weak* PUF, while a PUF that can support a very large number of CRPs that cannot be tried out in a reasonable time frame is called a *strong* PUF.

In this paper, we propose a novel PUF class, called *neuron-PUF*, that uses a single spiking neuron as the source of entropy. A spiking neuron produces a pulse wave whose preset duty cycle and periodicity will depend on random process variations. This analog signature is processed by a key extractor to generate the digital key. A stability-enhancement technique based on self-masking is used to drop native unstable bits identified during testing. Neuron-PUF uses a composite and multidimensional challenge which makes it a candidate for implementing a strong PUF.

Previously proposed PUFs use space redundancy, i.e., multiple PUF cells, to generate the PUF response. For example, the arbiter uses n delay path circuits built by a serial connection of multiplexers to generate an n-bit PUF response. The RO PUF consists of N ROs from which a maximum of $n = \log(N!)$ PUF response bits can be extracted. For SRAM PUFs, an n-bit PUF response can be extracted by selecting the logic state of n SRAM cells. In contrast, the proposed neuron-PUF uses a single PUF cell based on a single spiking neuron and extracts an arbitrarily long PUF response in a serialized fashion using temporal redundancy. Therefore, the proposed neuron-PUF significantly reduces area and power overheads.

The rest of the paper is structured as follows. In Section II, we discuss spiking neurons focusing on the specific spiking neuron that we employed. In Section III, we present the neuron-PUF architecture. In Section IV, we present the PUF quality

metrics used to characterize the proposed PUF. In Section V, we present the results. Section VI concludes the paper.

II. Spiking Neurons

Spiking neurons are biologically-inspired neuron models that serve as the fundamental building block of neuromorphic systems. Neuromorphic systems aim at emulating the brain functionality for efficiently solving cognitive tasks, i.e., visual recognition and motion control. A large number of neuromorphic systems have been demonstrated in the recent years [43]–[46]. Spiking Neural Networks (SNNs) constitute the third generation of neural networks aiming at bridging the gap between the biological brain and machine learning in terms of recognition speed and power consumption [47], [48].

There are several spiking neuron models of different complexities, ranging from biophysical models to phenomenological models, such as the most popular Integrate & Fire (I&F) model, which can have a hardware-friendly implementation and can still be designed to reproduce a large variety of spiking firing patterns observed in biological neurons [49]. An I&F spiking neuron receives and integrates input spikes from neurons in the previous layer via the synaptic connections. If its state reaches a certain threshold, then it fires a spike of its own that propagates to the neurons in the next layer via the synaptic connections. In addition, it resets its state so that it can fire again.

There are several hardware implementations of I&F spiking neurons of different complexities [49]. For the purpose of this work, as a proof of concept, we use the axon hillock circuit proposed in [50], whose schematic is shown in Fig. 1(a).

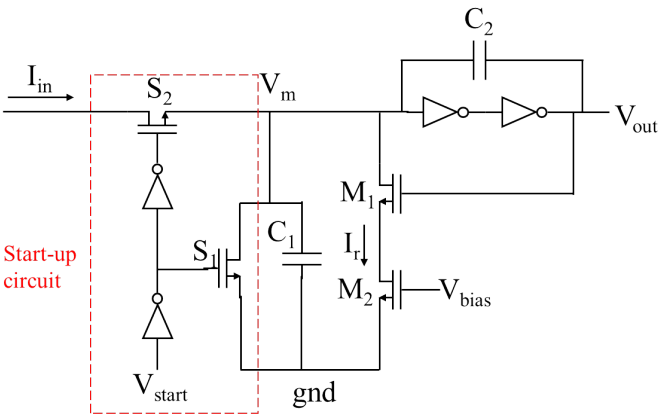
When $V_{start} = 0$, the input is disconnected and the capacitor C_1 is discharged to 0. When $V_{start} = V_{dd}$, the switch S_2 turns on and the switch S_1 turns off. The input current I_{in} starts charging the capacitor C_1, which models the membrane capacitance of the neuron's cell. When the membrane voltage V_m reaches the threshold V_ℓ of the first inverter, then the inverters switch state and the output fires a pulse, i.e., V_{out} goes to V_{dd}. The capacitor C_2 models the inherent positive feedback of the neuron cell. As V_{out} increases, V_m increases due to the positive feedback even faster. Setting a V_{bias} higher than the threshold voltage of transistor M_2 introduces a leaking mechanism that helps the neuron to reset itself once it has fired a spike. In particular, when V_{out} increases to V_{dd}, C_1 starts discharging to 0 via the current I_r flowing through transistors M_1 and M_2, until a point where V_m drops below the inverter's threshold. At this point, the output voltage goes rapidly to ground and the pulse is terminated. Then, for a constant input current the neuron starts again the integration process to fire another pulse. A simulation of this circuit is shown in Fig. 1(b).

It can be shown that the spike duration is given by

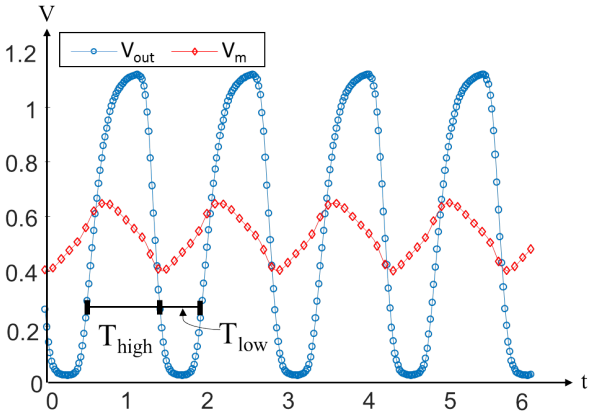
$$T_{high} \simeq \frac{C_2 \times V_{dd}}{I_r(V_{bias}) - I_{in}}, \tag{1}$$

and the time between two consecutive spikes is given by

$$T_{low} \simeq \frac{C_1 + C_2}{I_{in}} V_\ell. \tag{2}$$



(a)



(b)

Fig. 1. The axon hillock circuit: (a) schematic; (b) transient response.

III. Neuron-PUF Architecture

The architecture of the proposed neuron-PUF is illustrated in Fig. 2. It consists of the PUF core, which takes the challenge as input and produces the PUF raw response, and the stability booster, which enhances the stability of the PUF final response.

The PUF core consists of the spiking neuron, which generates an analog signature, and the key extractor, which processes the analog signature to generate the digital PUF response. In more detail, the PUF starts generating its response when V_{start} goes high. The spiking neuron integrates a current I_{in}, and produces at its output a spiking pattern in the form of a pulse wave, whose duty cycle and periodicity are defined by T_{high} and T_{low} in Eqs. (1)-(2). The key extractor consists of a linear-feedback shift register (LFSR) and an edge triggered flip-flop. The flip-flop receives its input from the LFSR and is clocked with the pulse wave, producing a serialized raw digital PUF response in the form of a bitstring.

Since the sampling is non-coherent, the raw response can be arbitrarily long. Therefore, fixed time windows on the raw response starting with any delay t_d with respect to V_{start} can be considered to extract an n_r-bit raw response. The time window equals n_r/f_s, where f_s is the system clock. Notice that the time window could also be divided into multiple intervals.

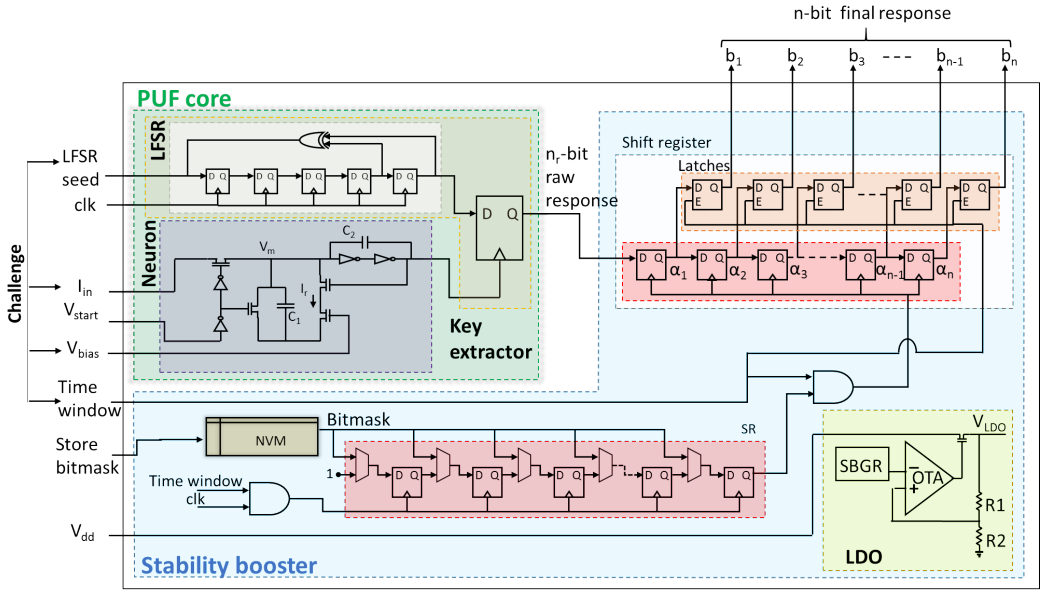


Fig. 2. Neuron-PUF architecture.

The challenge of the PUF is composed of the input current I_{in}, the bias voltage V_{bias} which controls the leaking mechanism of the neuron, the seed of the LFSR, and the delay t_d with respect to V_{start}. The input current I_{in} can be generated using a well-matched programmable current mirror controlled with a digital word of d_1 bits. Such a current mirror has d_1 mirroring branches, each composed of the mirroring transistor and a pass transistor controlled by one bit of the digital word. The bias voltage V_{bias} can be generated from a resistive ladder with equal resistors forming a voltage divider, which can be trimmed for precision. Using d_2 resistors we can program d_2 different V_{bias} values. The space of seeds is 2^{d_3}, where d_3 is the length of the LFSR. The delay t_d can take any value of d_4 clock cycles. Considering a single time window, the CRP space is equal to $2^{d_1} \cdot d_2 \cdot 2^{d_3} \cdot d_4$. The CRP space can be further increased by segmenting the time window into multiple parts. The CRP space can be made very large, which makes the neuron-PUF a good candidate for implementing a strong PUF.

The main source of entropy is process variations within the spiking neuron affecting capacitor values, leakage current I_r, and inverter threshold V_ℓ, which in turn alter the period and duty cycle of the firing pattern as shown in Eqs. (1)-(2).

The stability booster consists of a linear drop-out (LDO) regulator and a self-masking circuit. The LDO regulator stabilizes the supply for all the sub-blocks against temperature and external power supply variations. An already existing LDO inside the chip can be used to power the PUF. In our implementation, we use a standard LDO architecture composed of a sub-bandgap reference (SBGR) voltage generator, an error amplifier implemented with an operational transconductance amplifier (OTA), a power p-MOS transistor, and a feedback resistor network [51]. The bitmask is chip-specific and is computed during testing time. In particular, the chip is exercised by varying temperature and power supply and the n_r-bit PUF response is collected several times. The bitmask is a bitstring of length n_r that has 1 when the corresponding bit has been shown to be stable and 0 otherwise. The bitmask is stored inside the chip in a non-volatile memory (NVM) and loaded into a shift-register at power-up. When the time window starts, the raw PUF response is driven into a shift-register that is clocked with the bitmask. Thus, only the stable bits are serially shifted into the register, while the unstable bits are dropped. At the completion of the time window, the stable n-bit PUF response, where $n \leq n_r$, is latched using negative edge-triggered flip-flops.

The advantageous property of the proposed neuron-PUF is that it uses a single compact cell to produce an arbitrarily long key, thus it can offer significant power and area cost savings compared to existing PUFs that use one cell per key-bit.

Since the neuron-PUF generates serially the key and the key-bits are closely spaced, the key cannot be easily correlated to power traces, thus reading-out the key at run-time via side-channel analysis is unworkable. The neuron also presents complex dynamics albeit having a simple structure, i.e., the challenge is related to the derivative of the membrane potential that defines the spike firing. These two properties, combined with the composite and very large CRP space, make the neuron-PUF highly resilient against modelling attacks. Finally, the neuron-PUF is resilient to memory attacks for stealing the key. Stealing the bitmask from the NVM only reveals the position of stable bits and not the key itself, but this also requires knowing part of the challenge, i.e., the delay t_d.

IV. PUF QUALITY METRICS

There are several PUF quality metrics proposed in the literature [52], [53]. Herein, we use a distinct set of metrics that are the most vital for characterizing the reliability and randomness

of a PUF. The set of metrics includes uniformity, uniqueness (or inter-PUF variation), diffuseness, and stability (or intra-PUF variation). The first three metrics also characterize the unpredictability of the PUF against modeling attacks.

We use the following notation in describing PUF metrics:

- N is the number of PUFs evaluated on N different chips.
- n is the number of bits in the PUF response.
- m is the number of challenges of the PUF.
- r_{ijk} is bit j in the PUF response of chip i for challenge k.
- $\mathbf{R}_{ik}$ is the n-bit PUF response of chip i for challenge k, i.e., $\mathbf{R}_{ik} = [r_{i1k}, \cdots, r_{ink}]$.
- T is the number of PUF response measurements over time and different operating conditions, i.e., changes in ambient temperature and supply voltage fluctuations.
- $HD(\mathbf{R}_{ik}, \mathbf{R}_{jk})$ is the Hamming distance (HD) of the PUF responses in chips i and j for challenge k. Similarly, $HD(\mathbf{R}_{ik_1}, \mathbf{R}_{ik_2})$ is the HD of the PUF responses in chip i for two different challenges k_1 and k_2, and $HD(\mathbf{R}_{ik}^{t_1}, \mathbf{R}_{ik}^{t_2})$ is the HD of the PUF responses in chip i for challenge k for two different measurements t_1 and t_2.

Uniformity estimates how uniform the proportion of 0 and 1 is in the PUF response bits. For a given chip i and challenge k it is expressed as:

$$\text{Uniformity} = \frac{1}{n}\sum_{j=1}^{n} r_{ijk} \times 100\%. \tag{3}$$

Ideally, uniformity should be equal to 50%.

Uniqueness represents the ability of a PUF to uniquely distinguish a chip among a set of identical chips. For a given challenge k it is expressed as:

$$\text{Uniqueness} = \frac{1}{\binom{N}{2}} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{HD(\mathbf{R}_{ik}, \mathbf{R}_{jk})}{n} \times 100\% \tag{4}$$

Ideally, uniqueness should be equal to 50%.

Diffuseness measures the difference in the PUF responses when the PUF is queried with different challenges. For a given chip i it is expressed as:

$$\text{Diffuseness} = \frac{1}{\binom{m}{2}} \sum_{k_1=1}^{m-1} \sum_{k_2=k_1+1}^{m} \frac{HD(\mathbf{R}_{ik_1}, \mathbf{R}_{ik_2})}{n} \times 100\% \tag{5}$$

Ideally, diffuseness should be equal to 50%.

Stability captures the capability of the PUF to reproduce its response bits under temperature variations, power supply fluctuations, noise, and aging. For a given chip i and challenge k it is expressed as:

$$\text{Stability} = \left(1 - \frac{1}{T}\sum_{t=1}^{T} \frac{HD(\mathbf{R}_{ik}^{0}, \mathbf{R}_{ik}^{t})}{n}\right) \times 100\%, \tag{6}$$

where $\mathbf{R}_{ik}^{0}$ denotes an enrollment of the PUF response at nominal operating condition and is used as a reference. Ideally, stability should be equal to 100%.

V. Results

The neuron-PUF is designed in the 65nm CMOS technology by STMicroelectronics. For the purpose of simplicity, in our demonstration we consider that the PUF challenge is defined only based on I_{in}, i.e., V_{bias}, seed of LFSR, and delay t_d of the time window with respect to V_{start} are fixed.

For the spiking neuron we use minimum size transistors so as to increase the impact of process variations. We set $I_{in} = 14.6\mu$A and $V_{bias} = 0.6V$.

We used an LFSR with feedback polynomial $x^5 + x^4 + 1$ and seed "01110", resulting in a pattern that is repeated every 22 clock cycles. It turns out that the choice of the LSFR, i.e., length, polynomial, and seed, affects the PUF metrics. For example, increasing the LFSR length from 3 to 6 bits increases uniqueness by 10-25% and changing the seed can offer a further 4-7% improvement. The chosen LFSR results by performing such trials, but this is a quick analysis since PUF metrics close to their ideal values can be obtained in just a handful of trials.

We opt for generating a 256-bit raw PUF response. Considering a clock period of 2.56ns, this sets the time window to 0.4μs. We consider that the rising edge of the time window comes $t_d = 3$ clock cycles after V_{start} goes high.

The output of the LDO stabilizes around $V_{LDO} = 1.2$V for the nominal power supply $V_{dd} = 1.6$V. In particular, it shows a 33.4mV variation when V_{dd} varies from 1.4V to 3V and a 10mV variation when temperature varies from -55°C to 125°C. The transient response for a variation of load current from 50mA to 0mA and then from 0mA to 50mA, shows a maximum overshoot of 44.9mV and settles after 875ns, while the maximum undershoot is 53.2mV and response settles after 800ns.

Fig. 3 shows a transistor-level transient simulation of the neuron-PUF using the above settings. We plot all the relevant signals and at the bottom we show the raw PUF response and cross out the unstable bits. Bit α_1 corresponds to the output of the first flip-flop of the shift register, as shown in Fig. 2. When the bit of the bitmask is 1, α_1 is shifted by one position, i.e., $\alpha_2 = \alpha_1$, and α_1 is updated to store the bit of the current raw PUF response. When the bit of the bitmask is 0, α_1 retains its value. Thus, the first bit in the α_1 bitstring shown in Fig. 3 corresponds to bit b_n of the stable PUF response, and the last bit in the α_1 bitstring shown in Fig. 3 corresponds to bit b_1 of the stable PUF response.

A set of $N = 100$ PUF instances emulating PUFs from 100 different chips is generated by performing a Monte Carlo (MC) analysis with 100 runs, considering both mismatch and inter-die variations, and using the actual statistical process design kit (PDK) of the technology.

Uniqueness and average uniformity and diffuseness across the 100 PUF instances are calculated by simulating the PUF instances at nominal conditions of 25°C and $V_{dd} = 1.6$V. For diffuseness, we change I_{in} from 13.8μA to 15.3μA with step size of 0.15μA. Stability for each instance is calculated by changing temperature from -25°C to 100°C in steps of 25°C and V_{dd} from 1.3V to 1.9V in steps of 0.1V. Thereafter, we report average stability across all instances.

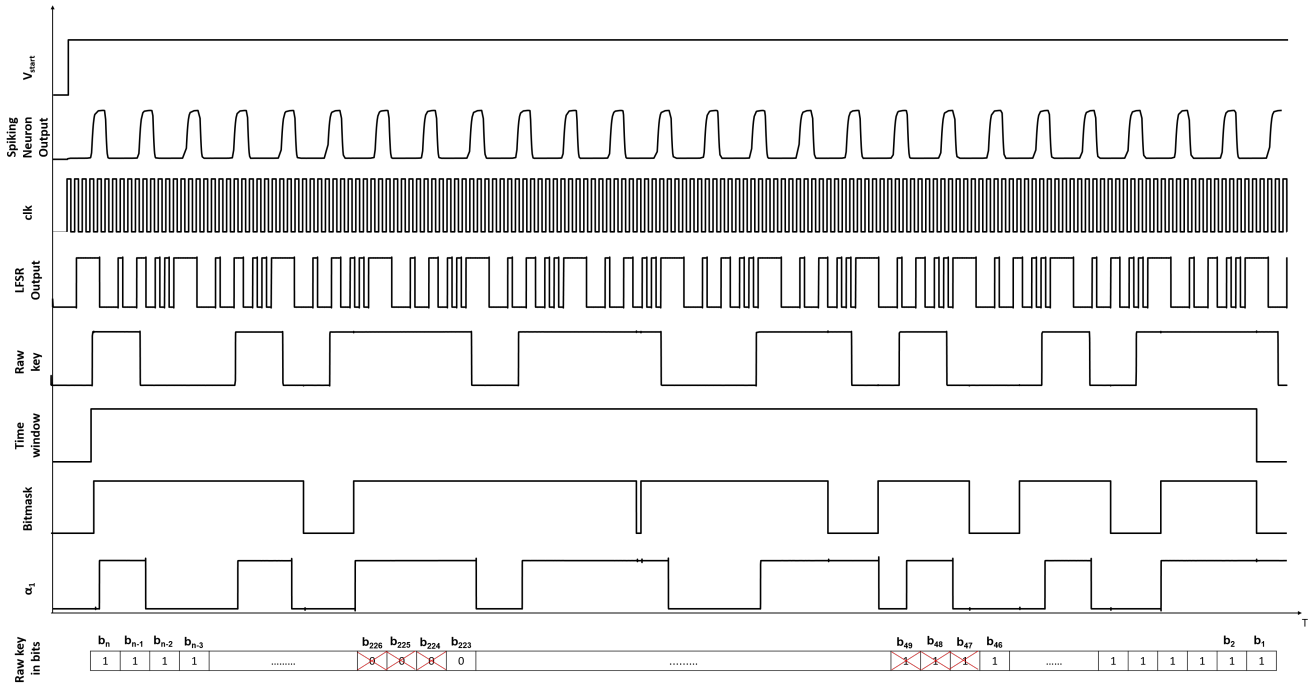


Fig. 3. Transient simulation of neuron-PUF showing relevant signals.

TABLE I
NEURON-PUF QUALITY METRICS.

	Uniformity	Uniqueness	Diffuseness	Stability (T)	Stability (V_{dd})
PUF core	53.83%	48.54%	54.07%	61%	60.51%
With LDO only	53.8	48.54%	49.43%	90.04%	96.11%
With complete stability booster	47.49%	48.42%	46.25%	100%	100%

The four PUF quality metrics are summarized in Table I considering the PUF raw response at the output of the PUF core, and the PUF response after stability boosting using the LDO only and the complete stability booster. As it can be seen, for the complete system, uniformity, uniqueness and diffuseness are 47.49%, 48.42%, and 46.25%, respectively, i.e. close to their ideal 50% value. Without the stability booster uniformity increases to 53.38% and diffuseness increases to 54.07%. Without the stability booster, the percentage of stable bits under V_{dd} and temperature variations is around 60.5% and 61%, respectively. Adding the LDO enhances the stability by around 35.6% and 29% against V_{dd} and temperature variations, respectively. With the complete stability booster in place, we obtain a fully stable 199-bit PUF response.

The PUF core consumes only 44.39nW/bit or 0.114fJ/bit, considering a clock period of 2.56ns and that there are 256 raw bits. For the same technology node, corresponding reported values for SRAM and RO PUFs are 1100fJ/bit and 474.8fJ/bit, respectively [37]. Using the stability booster, power consumption raises to 0.64μW/bit, but a stability booster is needed in all PUF architectures and can be excluded from the direct comparison.

Finally, the layout area of the PUF core is $13.4\mu\text{m} \times 20.2\mu\text{m}$. Considering that there are 199 stable bits, area per bit is 322F^2, where F=65nm is the minimum feature size. For the same technology node, corresponding reported values for SRAM and RO PUFs are 806F^2 and 39000F^2, respectively, computed as the ratio of the array area and the number of stable bits [37].

In summary, the neuron-PUF offers significant reductions in area and power overheads compared to SRAM and RO PUFs.

VI. CONCLUSIONS

We proposed neuron-PUF, a novel PUF design that uses a single spiking neuron as the source of entropy. Neuron-PUF is a single-cell PUF that uses temporal redundancy to generate a digital key of arbitrary size. It has a composite and large CRP space making it a candidate for implementing a strong PUF. Simulation results show that the neuron-PUF achieves close to ideal PUF metrics. It also has high potential for resisting modeling and memory attacks. The single-cell property results in minimum energy per bit and area per bit compared to all popular PUFs.

In terms of future work, we are planning to continue the verification and experimental validation towards proving PUF robustness and resilience to mainstream attacks. We are also planning to investigate the use of different types of spiking neurons [49] in the context of the neuron-PUF architecture.

ACKNOWLEDGMENTS

This work has been carried out in the framework of the ANR STEALTH project with N° ANR-17-CE24-0022-01.

REFERENCES

[1] C. Herder et al., "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.

[2] Y. Gao et al., "Physical unclonable functions," *Nature Electronics*, vol. 3, pp. 81–91, 2020.

[3] U. Rührmair et al., "Modeling attacks on physical unclonable functions," in *ACM Conference on Computer and Communications Security*, 2010, p. 237–249.

[4] X. Xu and W. Burleson, "Hybrid side-channel/machine-learning attacks on PUFs: A new threat?," in *Design, Automation & Test in Europe Conference Exhibition*, 2014, pp. 1–6.

[5] C. Helfmeier et al., "Cloning physically unclonable functions," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, 2013.

[6] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *ACM/IEEE Design Automation Conference (DAC)*, 2007, pp. 9–14.

[7] A. R. Sadeghi et al., "Enhancing RFID security and privacy by physically unclonable functions," in *Towards Hardware-Intrinsic Security. Information Security and Cryptography*, A. R. Sadeghi and D. Nassache (eds), Eds., pp. 281–305. Springer, Berlin, Heidelberg, 2010.

[8] J. Guajardo et al., "Brand and IP protection with physical unclonable functions," in *IEEE International Symposium on Circuits and Systems*, 2008, pp. 3186–3189.

[9] U. Guin et al., "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, 2014.

[10] J. A. Roy et al., "Ending piracy of integrated circuits," *IEEE Computer*, vol. 43, no. 10, pp. 30–38, 2010.

[11] J. Leonhard et al., "MixLock: Securing mixed-signal circuits via logic locking," in *Proc. Design, Automation & Test in Europe Conference*, 2019.

[12] M. Elshamy et al., "Securing programmable analog ICs against piracy," in *Proc. Design, Automation & Test in Europe Conference*, 2020.

[13] M. Yasin et al., *Trustworthy Hardware Design: Combinational Logic Locking Techniques*, Springer, 2020.

[14] B. Chatterjee et al., "RF-PUF: IoT security enhancement through authentication of wireless nodes using in-situ machine learning," in *Proc. IEEE International Symposium on Hardware Oriented Security and Trust*, 2018, pp. 205–208.

[15] M. J. Mahmod and U. A. Guin, "Robust, lowcost and secure authentication scheme for IoT applications," *MDPI Cryptography*, vol. 4, no. 1, pp. 8, 2020.

[16] K. Lofstrom et al., "IC identification circuit using device mismatch," in *IEEE International Solid-State Circuits Conference*, 2000, pp. 372–373.

[17] R. Pappu et al., "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.

[18] B. Gassend et al., "Silicon physical random functions," in *ACM Conference on Computer and Communications Security*, 2002, p. 148–160.

[19] D. Lim et al., "Extracting secret keys from integrated circuits," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 13, no. 10, pp. 1200–1205, 2005.

[20] E. Öztürk et al., "Towards robust low cost authentication for pervasive devices," in *IEEE International Conference on Pervasive Computing and Communications*, 2008, pp. 170–178.

[21] D. P. Sahoo et al., "A multiplexer-based arbiter PUF composition with enhanced reliability and security," *IEEE Transactions on Computers*, vol. 67, no. 3, pp. 403–417, 2018.

[22] A. Maiti et al., "A large scale characterization of RO-PUF," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, 2010, pp. 94–99.

[23] J. Guajardo et al., "FPGA intrinsic PUFs and their use for IP protection," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2007, p. 63–80.

[24] D. E. Holcomb et al., "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Transactions on Computers*, vol. 58, no. 9, pp. 1198–1210, 2009.

[25] K. Liu et al., "A 373-F^2 0.21%-native-BER EE SRAM physically unclonable function with 2-D power-gated bit cells and V_{ss} bias-based dark-bit detection," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 6, pp. 1719–1732, 2020.

[26] R. Maes et al., "Intrinsic PUFs from flip-flops on reconfigurable devices," in *3rd Benelux workshop on information and system security*, 2008.

[27] V. van der Leest et al., "Hardware intrinsic security from D flip-flops," in *Fifth ACM Workshop on Scalable Trusted Computing*, 2010, p. 53–62.

[28] S. S. Kumar et al., "Extended abstract: The butterfly PUF protecting IP on every FPGA," in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 67–70.

[29] Y. Su et al., "A digital 1.6 pJ/bit chip identification circuit using process variations," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 69–77, 2008.

[30] R. Helinski et al., "A physical unclonable function defined using power distribution system equivalent resistance variations," in *ACM/IEEE Design Automation Conference*, 2009, pp. 676–681.

[31] D. Suzuki and K. Shimizu, "The glitch PUF: A new delay-PUF architecture exploiting glitch shapes," in *Cryptographic Hardware and Embedded Systems, CHES 2010*, 2010, pp. 366–382.

[32] A. R. Krishna et al., "MECCA: A robust low-overhead PUF using embedded memory array," in *Cryptographic Hardware and Embedded Systems – CHES 2011*. 2011, pp. 407–420, Springer Berlin Heidelberg.

[33] T. Addabbo et al., "Physically unclonable functions derived from cellular neural networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 12, pp. 3205–3214, 2013.

[34] L. Bossuet et al., "A PUF based on a transient effect ring oscillator and insensitive to locking phenomenon," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 1, pp. 30–36, 2014.

[35] S. K. Mathew et al., "A 0.19pJ/b PVT-variation-tolerant hybrid physically unclonable function circuit for 100% stable secure key generation in 22nm CMOS," in *IEEE International Solid-State Circuits Conference*, 2014, pp. 278–279.

[36] Y. Gao et al., "Memristive crypto primitive for building highly secure physical unclonable functions," *Scientific Reports*, , no. 12785, 2015.

[37] A. B. Alvarez et al., "Static physically unclonable functions for secure chip identification with 1.9–5.8% native bit instability at 0.6–1V and 15 fJ/bit in 65nm," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 3, pp. 763–775, 2018.

[38] L. Yu et al., "iPUF: Interconnect PUF with self-masking circuit for performance enhancement," in *18th International Workshop on Microprocessor and SOC Test and Verification*, 2017, pp. 45–50.

[39] R. Maes et al., "PUFKY: A fully functional PUF-based cryptographic key generator," in *Cryptographic Hardware and Embedded Systems – CHES 2012*. 2012, pp. 302–319, Springer Berlin Heidelberg.

[40] M. Hiller et al., "Complementary IBS: Application specific error correction for PUFs," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, 2012.

[41] J. Delvaux et al., "Helper data algorithms for PUF-based key generation: Overview and analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 6, pp. 889–902, 2015.

[42] B. Colombier et al., "Key reconciliation protocols for error correction of silicon PUF responses," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1988–2002, 2017.

[43] S. B. Furber et al., "The SpiNNaker Project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, 2014.

[44] P. A. Merolla et al., "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[45] M. Davies et al., "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[46] L. A. Camuñas-Mesa et al., "A configurable event-driven convolutional node with rate saturation mechanism for modular convnet systems implementation," *Frontiers in Neuroscience*, vol. 12, pp. 63, 2018.

[47] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[48] L. A. Camuñas-Mesa et al., "Spiking neural networks and their memristor-CMOS hardware implementations," *Materials*, vol. 12, no. 17, pp. 2745, 2019.

[49] G. Indiveri et al., "Neuromorphic silicon neuron circuits," *Frontiers in Neuroscience*, vol. 5, 2011, Article 73.

[50] C. Mead, *Analog VLSI and Neural Systems*, Addison Wesley, 1989.

[51] M. Elshamy et al., "Hardware trojan attacks in analog/mixed-signal ICs via the test access mechanism," in *IEEE European Test Symposium*, 2020.

[52] S. Katzenbeisser et al., "PUFs: Myth, fact or busted? a security evaluation of physically unclonable functions (PUFs) cast in silicon," in *Cryptographic Hardware and Embedded Systems – CHES 2012*, Berlin, Heidelberg, 2012, pp. 283–301, Springer Berlin Heidelberg.

[53] A. Maiti et al., "A systematic method to evaluate and compare the performance of physical unclonable functions," in *Embedded Systems Design with FPGAs*, P. Athanas, D. Pnevmatikatos, and N. Sklavos, Eds., pp. 245–267. Springer, New York, NY, 2013.