Veda-PUF: A PUF based on Vedic Principles for Robust Lightweight Security for IoT

Venkata P. Yanambaka
College of Science & Engineering
Central Michigan University.
yanam1v@cmich.edu

Saraju P. Mohanty
Comp. Science & Engineering
University of North Texas.
saraju.mohanty@unt.edu

Elias Kougianos
Electrical Engineering
University of North Texas.
elias.kougianos@unt.edu

Babu K. Baniya
Computer Science & Digital Technologies
Grambling State University.
baniyab@gram.edu

Bibhudutta Rout
Dept. of Physics
University of North Texas.
bibhudutta.rout@unt.edu

Abstract—**This paper proposes a new controlled Physical Unclonable Function (PUF), Veda-PUF, which uses an algorithm for pre-processing and post-processing the input and output of PUF to increase the security of the keys generated in Internet-of-Things (IoT) devices. The key size of the PUF can be increased using the proposed protocol without compromising the integrity of the keys generated. The uniqueness of the generated keys was 50 % and the reliability of the keys generated is 99.9 % which are close to the ideal values. The proposed control algorithm also increases the uniqueness and reliability of the PUF keys after processing. This increases the number of PUF keys that can be used for various applications.**

I. Introduction

IoT devices are light weight, low power and low performance devices. The memory on the devices is also low which cannot store the data required for secure cryptographic algorithms. Lightweight security has to be implemented on these devices which can compromise the security and privacy aspect of the entire system [1, 2]. The PUF is a solution to implement lightweight security protocols in IoT architectures. It uses the manufacturing variations introduced during the fabrication of an Integrated Circuit to generate natural random numbers used for cryptographic purposes. A PUF does not store the keys but generates them on-the-fly when necessary [1, 3].

This paper proposes a new controlled PUF architecture, called "*Veda-PUF*" which uses the concepts of old Vedic chanting methods to generate the output keys for cryptographic purposes. Vedas are one of the most ancient scriptures that exist in the world. They form the basis for the Hindu Dharma [4].

II. Related Prior Research

Various architectures of PUFs have been introduced for a diverse application set [1, 5, 6]. The "things" in an IoT architecture are low-power low-performance devices which rely on the cloud for most of the post processing. They are mainly responsible for collecting the data from environment and transfer it to the cloud [7]. This makes implementing high level security in an IoT device difficult in terms of processing power.

PUF architectures also require chip area which is limited in the case of an IoT device. In some designs of PUF, the length of the key depends on the architecture, and an increased key size requires more devices in the design [7]. Controlled PUFs have been around for a long time [8]. Many designs of controlled PUFs were introduced, such as, in [9], a controlled-strong PUF was proposed using the Finite State Machine based PUF. PUF-FSM has also removed the requirement for error detection and correction mechanisms while generating the key. An MRAM based PUF was also proposed in the paper [10] which uses the unique energy tile generated by the random geometric variations of the MRAM module. These PUFs prioritize generating strong keys with the PUF modules. This paper proposes Veda-PUF, which combines the strong key generation and an increased key length.

Research on Vedas-Based cryptography is not new and has been pursued for a long time [11, 12]. Research was extensively performed to deduce various ancient texts to develop cryptographic algorithms. In [12], the authors proposed an Elliptic Curve Cryptography (ECC) based encryption and decryption method using Vedic Mathematics.

III. Novel Contributions

A. Problem Statement

Typically, the PUF key length depends on the number of devices used to design the PUF. A larger key requires a higher number of devices, in cases such as SRAM PUF or Arbiter PUF [13]. This is a challenge that needs to be addressed for a lightweight and robust security in IoT.

B. Proposed Solution

As a solution to the challenges mentioned in the previous sections, this paper proposes Veda-PUF, a Vedic chanting method based Controlled PUF with a control module. A controlled PUF is the type of PUF module where the challenges and responses are pre and post-processed for better and robust security. The following are proposed in the current paper for the Veda-PUF:

- A Controller for the PUF which controls and isolates the PUF operation.
- A Controller Algorithm to increase the length and strength of the key generated from the PUF.

IV. Ghanapatam - A Vedic Method

One important aspect of Vedas is the tone or tune with which they are recited, which is called "*Swaram*". Besides the tone, there are 11 ways of reciting the Vedas for an easier and perfect way to memorize the texts. They are Samhitha, Pada, Krama, Jata, Maala, Sikha, Rekha, Dhwaja, Danda, Rathaa, and Ghana. With time, most of the techniques were lost and only four of these are readily available to learn for students. The final form of recitation, Ghana, is considered the toughest form [4]. Ghanapatham is used in the current paper to design the proposed algorithm.

The formula to recite the Vedas in the Ghana format is as follows: consider the following array of bits in the challenge: [b_1, b_2, b_3, b_4... b_n]. In the Ghana format, every three bits are repeated in groups of 13. For example, consider the first set of three bits b_1, b_2, and b_3 and the set of 13 are represented by s_1. It can be represented as (for simplicity, the bits are divided into multiple arrays):

$$s_1 = [b_1, b_2]\,[b_2, b_1]\,[b_1, b_2, b_3]\,[b_3, b_2, b_1]\,[b_1, b_2, b_3] \tag{1}$$

As shown in Eqn. 1, the bits are repeated in the respective sequence to increase the number of challenge-bits to the PUF. A set of 3 bits is taken to process at a given point of time. Eqn. 1 shows the sequence of bits generated by considering the first three bits of the bitstream. Once the first three bits are done, the first bit is discarded and three bits from bit 2 are processed - b_2, b_3, and b_4.

At the end of the bitstream, a set of three bits could not be considered. This uses the formula for Jata form of recitation. They will be repeated in group of 6. For example, consider the last two bits, $b_(n-1)$, and b_n, and the set of repeated bits as s_n, the sequence of bits are:

$$s_n = \left[b_(n-1), b_n\right]\,\left[b_n, b_(n-1)\right]\,\left[b_(n-1), b_n\right] \tag{2}$$

V. Proposed Veda-PUF

Fig. 1 shows the architecture of the proposed controlled PUF in an IoMT module. The processor in the IoMT module is also used as a PUF Control Module. The PUF operations in the module are isolated from the rest of the application specific operations, which makes it secure from attacks. The processor communicates with the PUF module to generate the keys.

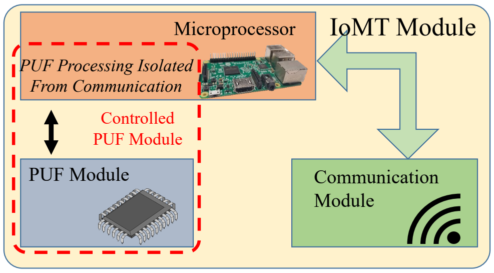


Fig. 1. Proposed Veda-PUF Architecture

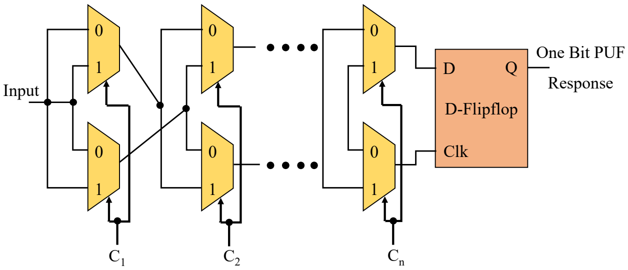


Fig. 2. Conventional One-Bit Arbiter PUF Architecture

A. Controller for Veda-PUF

Any traditional PUF can be integrated to the design, as shown in Fig. 2. The PUF module used in the design is a traditional Arbiter PUF module. Fig. 2 shows the design of a one-bit Arbiter PUF. As shown in the figure, multiplexers are connected in series in the PUF architecture. The output of a multiplexer is connected to the inputs of the next multiplexers. The design shown in Fig. 1 generates a single bit output. The output bit depends on the challenge inputs given at the select signals of the multiplexers. This can be configured on the fly based on the challenges. Dur to the manufacturing variations during the fabrication process, when the challenge is changed, the delay added to the input signal changes which changes the output at the D-Flipflop. A set of 128 such modules can generate a 128-bit key for cryptography. There are two phases in generating the keys from a PUF module, pre-processing Phase and the post-processing Phase.

B. Proposed Veda-PUF Controller Algorithm

Fig. 3 shows the proposed algorithm. As shown in the figure, the IoMT module communicates with the PUF over the isolated communication channel. A challenge is selected for the PUF module, C_1 and given as an input. This generates the Response, R_1. The response R_1 is processed in the pre-processing stage using the equations 1 and 2. This formula increases the challenge bits which gives a new challenge, PC_1. This challenge is given to the PUF to generate a response, R_2.

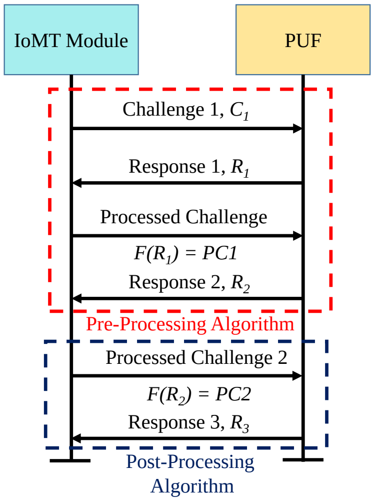


Fig. 3. Proposed Controller Algorithm for Veda-PUF

During the Post-Processing Algorithm, the generated response, R_2 is processed again using the equations 1 and 2 once again to increase the key length and strength such as, uniqueness and randomness. The obtained final key, is used as a challenge again for the PUF. This ensures the key generation is resistant against some machine learning algorithms. The output generated by the PUF, R_3 is the final key that can be used for applications, such as, device authentication.

VI. Experimental Results

A hardware prototype of the Veda-PUF was designed using Single Board Computers and Field Programmable Gate Arrays for evaluation (see Fig. 4). Not all the keys generated by a PUF module cannot be used for cryptographic purposes. The properties of PUF has to be satisfied to use the PUF keys for security. Three main properties of PUF are, Uniqueness, Reliability and Randomness.

A. Uniqueness

Uniqueness of a PUF is measured by the Hamming distance calculated between the keys. Hamming distance measures how distinct the two given

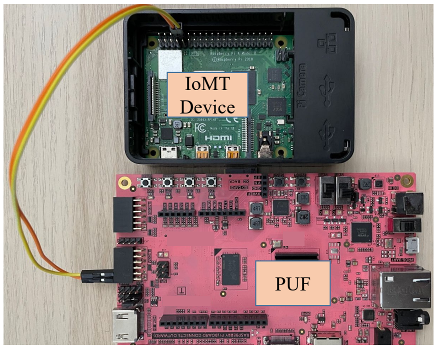


Fig. 4. Experimental Setup

keys are. The ideal value of uniqueness for a PUF is 50 % Hamming distance. A key can be used for cryptographic purposes if it has a uniqueness of 50 %.

Fig. 5 shows the uniqueness of the keys generated from the PUF module and processed using the proposed algorithm. Fig. 5a shows uniqueness of the keys generated by the PUF. Every key generated has been compared to each other in the PUF module to get the uniqueness. 1,000 keys were generated from the PUF and were compared to each other. From Fig. 5a, it can be observed that the number of occurrences that are around 50 % ideal value are 14,000. Fig. 5b shows the average uniqueness of the keys did not change but the number of keys that are around the near ideal value, 44 % - 50 % increased compared to the original number of keys. At the same time, the key length has also been increased significantly protecting the integrity of the keys.

B. Reliability

Reliability of PUF is the ability of the module to generate the same keys with no errors. Various factors affect the functioning of the module, such as aging, power supply variations, etc. Given a challenge, the PUF should produce the same response under all conditions. In the proposed algorithm, the reliability of the output key depends on the PUF module and the architecture that has been integrated into the design. The experimental setup uses an Arbiter PUF designed on an FPGA. The reliability of the PUF module used is 100 % and could generate the keys without any errors on multiple runs. The proposed algorithm or the control module in the PUF does not affect the reliability of the keys or introduces any bit flips.

C. Randomness

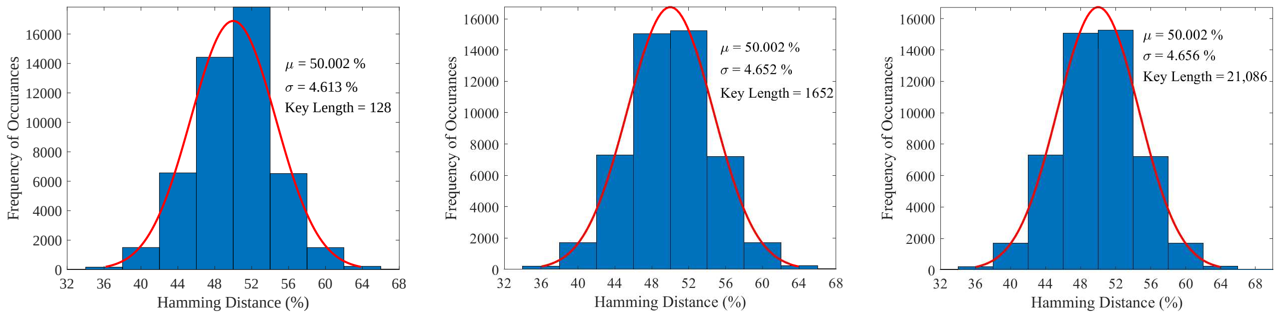
Randomness of the PUF module is the ability to generate keys that have equal number of bits. For example, a 128-bit output key must have 64 1’s and 64 0’s in the key. Fig. 6 shows the randomness of the output keys generated from the algorithm. It shows the number of zeros present in the output key. The ideal value of randomness is 50. Fig. 6a shows the randomness of the original keys. Figs. 6b and 6c shows the randomness of the keys after processing. In the original keys, most of the occurrences are concentrated when the randomness is between 52 and 56. After processing, the number of occurrences has been distributed around 50. This shows the proposed algorithm can be used to strengthen the keys generated from the PUF module.

TABLE I
CHARACTERIZATION OF THE PROPOSED VEDA-PUF.

PUF Characteristic	Original Key	Processed Key
Uniqueness		
Mean	50.002 %	50.002 %
Standard Deviation	4.613 %	4.656 %
Reliability		
Mean	99.9 %	99.9 %
Standard Deviation	0 %	0 %
Randomness		
Mean	50.292 %	50.270 %
Standard Deviation	5.739 %	5.740 %
Power Consumption	3.1 W	3.25 W

D. Resistance Against Security Attacks

Two parts of the proposed Veda-PUF are considered when analyzing the security aspects of the system. The first part, being the system, the current process, PUF key generation and the algorithm were isolated from the rest of the processes and the communication modules. This can help in isolating the keys from the attackers, even when the system has been remotely compromised. Other attacks that can pose a threat to the system are side channel attacks that can compromise the system by analyzing the keys generated from the PUF. The output has been collected from the PUF and the challenges

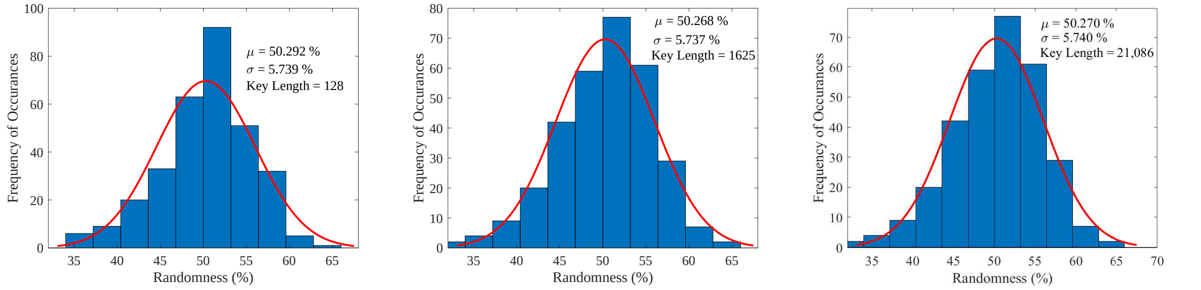


(a) Uniqueness of Original Keys

(b) Uniqueness of Processed Keys

(c) Uniqueness of Keys Processed a Second Time

Fig. 5. Uniqueness of Keys.



(a) Randomness of Original Keys

(b) Randomness of Processed Keys

(c) Randomness of Keys Processed a Second Time

Fig. 6. Randomness of Keys.

that were generated by the algorithm were not used directly. There are already many designs of PUF that are proposed to resist side-channel attacks and machine learning attacks. The proposed algorithm is independent of the PUF architecture. Integrating a side-channel attack resilient PUF can help in resisting against some of these attacks.

E. Comparison with Prior Work

Various designs of PUF were developed in the last two decades. Table II shows a comparison of the Veda-PUF with existing designs. The comparison was performed with all traditional designs of PUF besides the controlled PUF designs, as the current paper proposes a controlled PUF and an increase in key length without compromising the integrity of the keys. The PUF designs proposed in [3, 6] were Ring Oscillator PUF on an Application Specific Integrated Circuit (ASIC) and Multiplexer based Arbiter PUF. Both designs show a uniqueness and randomness of around the ideal values. But these are standalone PUF designs and the key length depends on the number of devices used in the PUF. The controlled PUFs proposed in [8, 9] have the ability to be integrated into an IoT architecture. But the designs are vulnerable to probing attacks and the length of the key is dependent on the devices as well. Compared to existing designs, the design in the current paper has near ideal uniqueness and randomness values while increasing the key length significantly.

VII. Conclusion And Future Research

This paper presents Veda-PUF, a lightweight controlled PUF architecture which uses the processor and the PUF module to generate a large key set and increase key strength. The proposed design uses Ghanapatham, a Vedic method used to recite the Vedas in a respective tune and sound. Using the proposed algorithm, the key length can be increased

TABLE II
COMPARISON WITH EXISTING DESIGNS.

Work	PUF Used	Proposed Protocol	Challenges	Uniqueness	Randomness
Nozaki et al. [3]	Ring Oscillator PUF	ASIC Evaluation	ASIC were designed and Key Length depends on the number of devices used.	64.59 %	63.86 %
Sahoo et al. [6]	Arbiter PUF	Multiplexer Based APUF	Key length depends on the number of devices used	50.01 %	49.79 %
Gassend et al. [8]	Strong PUF	Controlled PUF	Internal Probing can present the key	–	–
Gao et al. [9]	Arbiter PUF	PUF-FSM	Key length depends on the number of devices used	–	–
Beckmann et al. [14]	Logic Gates Based PUF	Simulation based Public PUF	IoT integration is challenging	–	–
The Current Paper	Arbiter PUF	Controlled PUF	–	50.002 %	50.270 %

from 128 bits to 2.5 Kilobytes saving the integrity and robustness of the key. As a future work, the proposed framework will be tested and improved for resistance against various attacks, such as machine learning and side-channel attacks.

REFERENCES

[1] V. P. Yanambaka, S. P. Mohanty, E. Kougianos, and D. Puthal, "Pmsec: Physical unclonable function-based robust and lightweight authentication in the internet of medical things," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 3, pp. 388–397, 2019.

[2] G. Dessouky, S. Zeitouni, A. Ibrahim, L. Davi, and A. Sadeghi, "CHASE: A Configurable Hardware-Assisted Security Extension for Real-Time Systems," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2019, pp. 1–8.

[3] Y. Nozaki and M. Yoshikawa, "Quantitative Performance Evaluation of PL PUF and RO PUF with ASIC Implementation," in *in Proceedings of IEEE 8th Global Conference on Consumer Electronics (GCCE)*, 2019, pp. 1127–1128.

[4] K. Suresh, *Sri Rudra Ghanam*, K. Suresh, Ed. Latha Publishers, 2002.

[5] W. Liu, Z. Lu, H. Liu, R. Min, Z. Zeng, and Z. Liu, "A novel security key generation method for sram puf based on fourier analysis," *IEEE Access*, vol. 6, pp. 49 576–49 587, 2018.

[6] D. P. Sahoo, D. Mukhopadhyay, R. S. Chakraborty, and P. H. Nguyen, "A multiplexer-based arbiter puf composition with enhanced reliability and security," *IEEE Transactions on Computers*, vol. 67, no. 3, pp. 403–417, 2018.

[7] S. P. Mohanty, V. P. Yanambaka, E. Kougianos, and D. Puthal, "PUFchain: A Hardware-Assisted Blockchain for Sustainable Simultaneous Device and Data Security in the Internet of Everything (IoE)," *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 8–16, 2020.

[8] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Controlled Physical Random Functions," in *18th Annual Computer Security Applications Conference*, 2002, pp. 149–160.

[9] Y. Gao, H. Ma, S. F. Al-Sarawi, D. Abbott, and D. C. Ranasinghe, "PUF-FSM: A Controlled Strong PUF," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 5, pp. 1104–1108, 2018.

[10] J. Das, K. Scott, S. Rajaram, D. Burgett, and S. Bhanja, "MRAM PUF: A Novel Geometry Based Magnetic PUF With Integrated CMOS," *IEEE Transactions on Nanotechnology*, vol. 14, no. 3, pp. 436–443, 2015.

[11] M. Nachtigal, H. Thapliyal, and N. Ranganathan, "Design of a reversible single precision floating point multiplier based on operand decomposition," in *10th IEEE International Conference on Nanotechnology*, 2010, pp. 233–237.

[12] S. Karthikeyan and M. Jagadeeswari, "Performance Improvement of Elliptic Curve Cryptography System Using Low Power, High Speed 16X16 Vedic Multiplier Based on Reversible Logic," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 3, pp. 4161–4170, 2021.

[13] S. Joshi, S. P. Mohanty, and E. Kougianos, "Everything You Wanted to Know about PUFs," vol. 36, no. 6, pp. 38–46, 2017.

[14] N. Beckmann and M. Potkonjak, "Hardware-Based Public-Key Cryptography with Public Physically Unclonable Functions," *International Workshop on Information Hiding*, pp. 206–220, 2009.