

A Secret Key Generation Scheme for Internet of Things using Ternary-States ReRAM-based Physical Unclonable Functions

Ashwija Reddy Korenda, Fatemeh Afghah, Bertrand Cambou

School of Informatics, Computing and Cyber Systems, Northern Arizona University, Flagstaff, AZ 86011

{ashwijakorenda,fatemeh.afghah,bertrand.cambou}@nau.edu

Abstract—Some of the main challenges towards utilizing conventional cryptographic techniques in Internet of Things (IoT) include the need for generating secret keys for such a large-scale network, distributing the generated keys to all the devices, key storage as well as the vulnerability to security attacks when an adversary gets physical access to the devices. In this paper, a novel secret key generation method is proposed for IoTs that utilize the intrinsic randomness embedded in the devices' memories introduced in the manufacturing process. A fuzzy extractor structure using serially concatenated BCH-Polar codes is proposed to generate reproducible keys from a ReRAM-based *ternary-state* Physical Unclonable Functions (PUFs) for device authentication and secret key generation. The main concern in deploying PUF-based key generation methods is the leakage of information about the secret keys from the publicly available helper data. The fuzzy extractor proposed in this paper ensures much less mutual information between the generated keys and the helper data. The experimental results show that our proposed scheme is capable of generating notably stronger keys compared to existing techniques, while utilizing a significantly lower number of helper data bits. The failure probability when a low complex Successive Cancellation decoder is implemented in the proposed fuzzy extractor structure is 10^{-8} which was further increased to 10^{-10} when a complex iterative belief propagation decoder was used.¹

Index Terms- Strong cryptographic key, Fuzzy Extractor, ReRAM, PUFs, Polar Codes.

I. INTRODUCTION

The emergence of Internet of Things (IoT) has led to a technology breakthrough, but it comes with a set of new challenges including security, radio spectrum scarcity, complexity of network, and data management. IoT nodes are prone to selfish and malicious attacks. The selfish attacks can be reduced by using a reputation based model in a property rights spectrum sharing model using coalition formation techniques [1]. Once spectrum is allocated to the IoT nodes, we have to ensure that the the information is sent to, or received from an authenticated user. Authentication is currently based on key-based cryptographic mechanisms that heavily rely on uniformly distributed, robust, reliable, and reproducible secret keys which are stored in the non-volatile memory (NVM) of the device [2]. However, these NVMs are highly vulnerable to physical attacks due to their robust electrical nature and maintaining the secrecy and integrity of these keys is a strenuous task,

¹This project is partially supported by Arizona Board of Regents under Grant # 1003330.

particularly in large-scale heterogeneous IoTs. Protection against physical attacks often involves the use of active tamper detection and prevention circuitry that requires a constant power supply [3]. Therefore, these approaches are expensive from both design area and power consumption perspectives. Hardware-based security mechanisms using Physical Unclonable Functions (PUFs), utilize the intrinsic variations introduced during the fabrication process which include the doping level of semi-conducting layers, threshold voltages, and critical dimensions, etc., to extract a “fingerprint” of the device. This can be used for identification and authentication of the device and will allow us to eliminate the need to store the secret keys in the NVM of the devices [4]–[6].

PUF-based security mechanisms rely on building unique challenge and response pairs (CRPs) for each device that can be reproduced in different conditions. Strong PUFs are expected to offer a high enough inter Hamming distance (distance between responses of two different PUFs under same conditions) which makes each device unique from the other ones, while having a low intra Hamming distance (distance between two responses of a single PUF under same conditions) to reduce the amount of False Acceptance Rate (FAR) [7]. However, the responses of PUFs are subject to variations due to temperature changes, aging, drifts, electro-magnetic interactions, and other sources of noise. To deal with such variations in the PUFs' responses in different physical conditions, several fuzzy extractors which use strong error correction coding (ECC) techniques have been proposed [8]–[13]. These conventional ECC schemes can provide an acceptable error correction capability where the PUFs are utilized for authentication purposes, since true authentication can be granted if the rate of matching responses is below a certain threshold [14]. However, these current ECC schemes deem insufficient when the PUFs are utilized for generating cryptographic keys, where even a single bit mismatch between the private keys generated from PUF responses in different conditions can invalidate the performance of the symmetrical encryption schemes such as Data Encryption Standard (DES) and Advanced Encryption Standard (AES).

The contribution of this paper is to propose a novel key generation method using ternary state ReRAM-based PUFs to generate reproducible and reliable keys using less publicly available *Helper Data*. Here, we propose a syndrome-based fuzzy extractor using serially concatenated BCH-Polar codes that can offer several

unique advantages compared to current technologies:

- Uses polar codes which can significantly reduce the mutual information between the generated key and the publicly available helper data compared to other schemes, and hence reduce the capability of attackers to reconstruct the key by using the helper data.
- Using serially concatenated BCH-Polar codes can exactly regenerate the key using helper data which is highly demanded in PUF-based key generation applications.

The rest of this paper is organized as follows: In Section II, a brief review of the state of the art PUF technologies, fuzzy extractor schemes and importance of using Polar codes in Fuzzy extractors is provided. In Section III, the proposed fuzzy extractor scheme using serially concatenated BCH-Polar Codes is provided. Section IV discusses the experimental results and the performance of the proposed method.

II. RELATED WORK

A. Memory-based PUFs

Memory-based PUFs use the memory present in a chip/device to extract a fingerprint and therefore, they can be implemented in simple IoT devices with little or no extra parts added, to allow authentication in a network. Several memory technologies have been deployed to build PUFs. SRAM-based PUF uses CMOS architecture and can be easily integrated in a system, but they give off energy while switching states that can be detected using a signal analyzer. Therefore, this type of PUF gives the hackers enough information to clone the device and extract the key [15], [16].

New IoT technologies often utilize more advanced memory devices such as Resistive Random Access Memory (ReRAM), memristive devices, and MRAM. Memristor and ReRAM rely on resistive technology to store information, where the current or the resistance of the cell when a biased voltage is applied can be used as a challenge [17], [18]. These devices have high randomness which not only occur in separate dies but also in the same die. Therefore, they can be a good candidate for a PUF. ReRAM based PUFs are faster when compared to MRAM or Flash and operate at or below noise level making them more resistant to side channel attacks without direct access to the chip. Operating power of ReRAM is around 10pJ/bit which is much lower compared to 1mJ/bit for Flash and 100pJ/bit for MRAM.

While the accessibility, easy process of challenge and response generation, and operation at or below noise level in ReRAMs are of great interest to generate ideal PUFs, these memory devices are highly sensitive to the physical conditions (e.g. temperature, operating current, and electromagnetic interference) of the test environment. Besides such environmental circumstances, other factors including aging of the device and random noise sources can result in flipping of the programmed bits one way or other during the response generation. This in turn increases the probability of false negative

authentication (FNA) when a single device is being authenticated in different conditions. Such errors can be partially corrected using ECC and fuzzy extractor schemes to allow the use of memory-based PUFs for authentication purposes. However, in key generation applications where the extracted key is used to encrypt the messages of a user, even a single bit error does not allow the decryption of the message. Thus limiting the applications of memory-based PUFs as cryptographic primitives [19]. A new concept of *ternary PUFs* was proposed in [16] to enhance the reliability of PUFs' responses by identifying three possible states of "0", "1" and "X" for each cell rather than a conventional binary assignment. In the proposed ternary PUFs, the cells that are too close to the threshold and are not reliable are blanked by an "X". This will allow reduction in CRP error compared to a binary scenario but, the CRP error is still not equal to 0. Here we utilize these ternary ReRAM-based PUFs to propose a practical solution for key generation described in Section III to allow exact reproduction of the key.

B. Fuzzy Extractors

Fuzzy extractors were proposed in [2] to convert noisy non-uniform PUF data into reproducible uniform strings. Fuzzy extractors can extract a uniformly random string S and a non-secret helper data string P from its original input W . This P will allow exact reproduction of the string S from noisy input data W' which remains close to W . A review of several constructions for fuzzy extractors are provided in [2]. Fuzzy extractor structures are based on utilizing different ECC techniques, among which BCH codes have been widely utilized [2], [8], [9], [11]–[13]. Two constructions for fuzzy extractors namely: code offset and syndrome-based constructions were proposed in [2]. In code-offset construction; helper data is generated by XORing the PUF input and the respective ECC codeword of the PUF input, while in the syndrome-based construction; helper data is generated by multiplying the PUF data with a parity matrix. Our proposed fuzzy extractor follows the syndrome-based construction, but can be extended to a code-offset construction by masking the PUF data using the ECC codewords.

C. Polar Codes

Polar codes are linear block ECCs whose construction is based on multiple recursive concatenation of short kernel codes which transforms the physical channel in to virtual outer channels whose capacity is either 0 or 1 [20]. The channels whose capacity is 1 are called *unfrozen channels* and they are used to sent information. These bits through which information is sent are called *unfrozen bits*. The channels whose capacity is 0 are called *unfrozen channels* and usually a 0 is sent through these channels. These bits through which no information is sent are called *frozen bits*.

The position where the unfrozen and frozen bits appear during channel polarization is dependent on the *Bhattacharya parameter* (measure of reliability) which is a function of the channels design parameters. Noting the

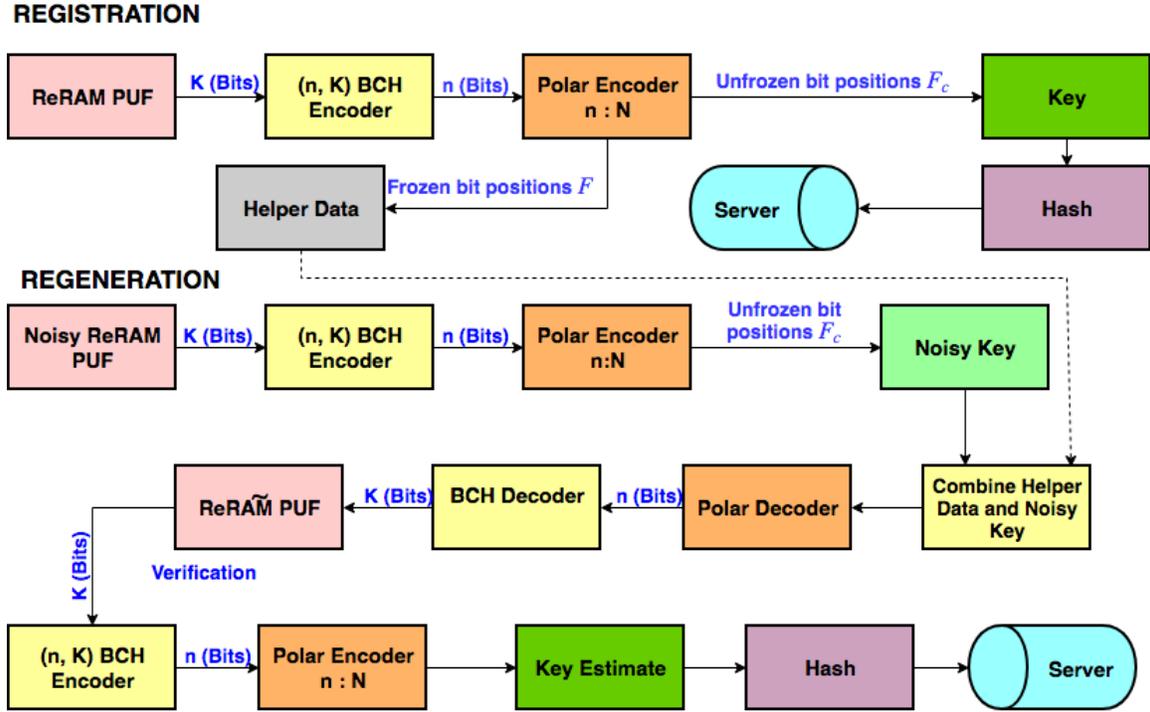


Fig. 1: Block diagram of the proposed key generation scheme for Key registration and Regeneration.

considerable error correction capability of polar codes and more importantly, the low (almost zero) mutual information between the frozen and unfrozen bits in such codes, they can be utilized in combination with fuzzy extractors for key generation purposes. In [10], a fuzzy extractor structure only based on Polar codes for binary SRAM PUF was proposed in which the key was generated using unfrozen bits of polar codes and the frozen bits were used as the helper data. This paper utilized a computationally complex successive cancellation decoder called Hash Aided SC List decoder (HASCL) to achieve a failure probability of 10^{-9} but utilized 896 bits to regenerate a 128 bit key. Thus utilizing data almost 7 times the size of the key for key regeneration.

III. PROPOSED KEY GENERATION SCHEME FOR TERNARY ReRAM-BASED PUFs

Here, we propose a secret key generation scheme using ternary ReRAM-based PUFs, to generate unique keys from noisy PUFs' data by using less helper data compared to previously proposed schemes. A fuzzy extractor using a serial concatenation of BCH and polar codes is designed to generate the *Secret Key* and *Helper Data*. Using polar codes in the proposed scheme results in reducing the level of mutual information between the generated secret key and the publicly made available helper data, since there is almost zero mutual information between the frozen bits and unfrozen bits [10]. Our proposed serially concatenated BCH-Polar model requires a lower number of helper data bits while offering a comparable failure probability compared to common fuzzy extractor schemes proposed in the literature as

shown in the Table I. This is possible as the BCH encoder also helps correct a few bits of the noisy PUF data, thus increasing the coding efficiency while having a low complexity.

The stable 0's and 1's are obtained from the ReRAM PUF data, by first dividing the ReRAM PUF data into three states "0", "1", and "X" and then discarding the memory cells which are prone to flip due to slight variations in environment (X) [21]. This will essentially extract a stable binary PUF from the ternary PUF which can be used to generate strong, reliable and reproducible cryptographic keys [22].

A. Registration

In the *Registration* phase, the ReRAM PUF data is encoded using a BCH encoder to increase the coding gain and allow reconstruction of the PUF responses with minimum error and less helper data. The BCH encoded output is then sent to a polar encoder whose output is unique for every input $U^N = X_{BCH}^N G^N$, where U^N is the polar encoded output, X_{BCH}^N is the BCH encoded PUF data, and G^N is the generator. The bits that occur in the non-frozen positions at the output of the polar encoder are hashed and stored in the server as a *secret key*, S^K and the *helper data*, P^{N-K} is obtained from the frozen bit positions of the polar encoded data. Helper data and secret key can be represented by the equations below,

$$P^{N-K} \cong X_{BCH}^N G_{\mathcal{F}}^N = C^N[\mathcal{F}] \quad (1)$$

$$S^K \cong X_{BCH}^N G_{\mathcal{F}_c}^N = C^N[\mathcal{F}_c] \quad (2)$$

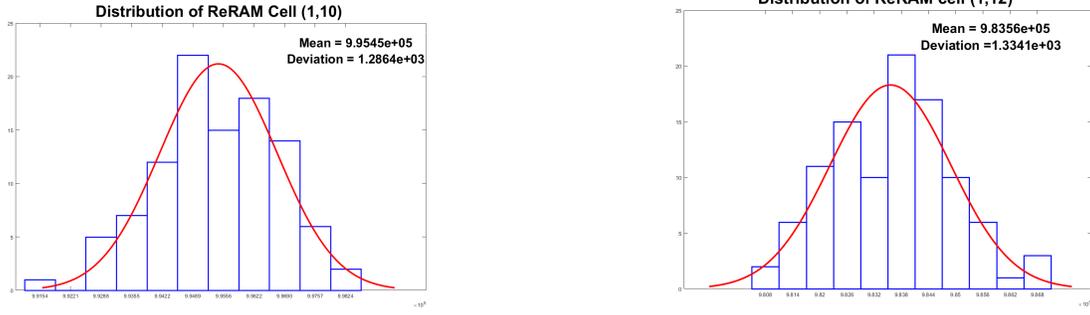


Fig. 2: Distribution of various Re-RAM Cells

where \mathcal{F} represents the frozen bit indices and \mathcal{F}_C represents the non-frozen bit indices. As the cells are assigned 0's and 1's based on their individual resistances, the cells are independent of each others behavior. An example of a Polar Encoder used in Key Generation scheme with $\{1,2,3,4,6\}$ frozen bits representing five Helper Data bits made publicly available to regenerate the three bit key using Noisy PUF data is shown in Fig. 3. In this figure, x_i represents the BCH encoded PUF bits and c_i represents the polar encoded output of x_i .

B. Key Regeneration

During *Key Regeneration* phase, a polar decoder and BCH decoder are utilized to observe the output of the BCH-Polar encoded noisy PUF along with publicly available Helper Data. The Helper data is combined with the BCH-Polar encoded noisy PUF by simply substituting the frozen bits of the noisy encoded data with the helper data bits. The Polar and BCH decoder utilizes this combined data to get a close estimate of the initial ReRAM PUF input. When a low complex SC decoder is utilized as a Polar decoder, its performance is not good enough for short block lengths, hence it can be further improved using SC list decoder, or a Belief propagation decoder. Fig. 1 summarizes the construction of *Key* and *Helper data* using ReRAM PUF data.

IV. RESULTS

In this study, we use the samples from a ReRAM PUF fabricated at Northern Arizona University's CyberSecurity laboratory. This PUF comprises of a HfO_2 active solid electrolyte chalcogenide with a Hafnium based top electrode, and a tungsten based bottom electrode. The measured parameters from each cell include: i) stepping the voltage in the positive voltage direction at constant compliance current to measure the programming voltage, V_{set} ; ii) after programming, measurement of the low resistive states R_{on} . The measurements from 254 fabricated ReRAM cells are available, where each measurement was repeated 102 times. After analyzing the data, the measurements of R_{on} were used to generate the CRPs as it offers the desired performance metrics: i) the measurement of each cell is independent and is normally distributed, and ii) the readings of resistance for different cells are also independent and normally

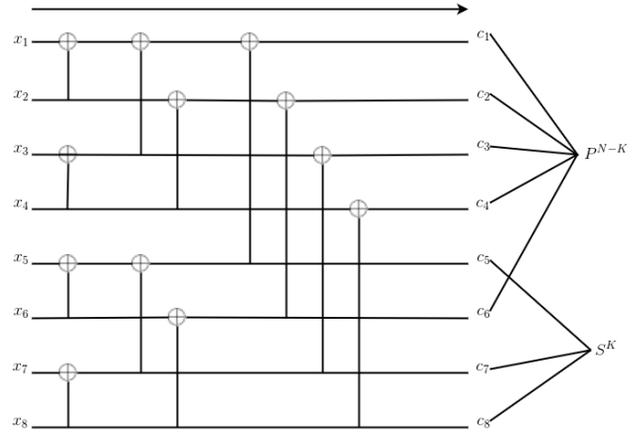


Fig. 3: Polar Encoder to select Key and Helper Data from BCH encoded ReRAM PUF.

distributed. Fig. 2 shows the independency and normal distribution of the R_{on} measurements for two randomly selected cells.

In order to determine the optimum threshold values to classify the cell responses to '0', '1', and 'X', the distribution of all the 254 cells is plotted, and then range of the response distribution is divided into three equal parts using two thresholds, $Threshold_1$ and $Threshold_2$. The cells whose resistance is below $Threshold_1$ (lies in the left most part of the histogram), are programmed with a '0'. The cells whose resistance values lie in between $Threshold_1$ and $Threshold_2$, are programmed with 'X'. The cells with the resistance values higher than $Threshold_2$ are programmed with a '1'.

The utilized ternary PUFs, exclude the unstable cells programmed with a 'X' and do not used them in generating the secure keys'. Thereby, the binary output of the generated ternary PUFs are more stable and reliable compared to a conventional binary PUF. Figure 4 shows the distribution of the 254 cells and the thresholds used to divide the data into three states.

In the Registration phase, the readings of R_{on} for the available 254 cells were mapped to '0', 'X' or '1'. A mask was used in order to extract 131 stable bits from the available 254 cells. These stable PUF bits are initially encrypted using a (255, 131) BCH encoder. Then, the

TABLE I: Comparison Of different fuzzy extractor schemes proposed in the literature.

Fuzzy Extractor Construction	Key Length	Helper Data bits	Failure Probability	Flipping probability
Reed Muller Generalized Multiple Concatenated coding [23]	128	13952	10^{-6}	15%
BCH Repetition Code [24]	128	2052	10^{-9}	13%
Generalized Concatenated (GC) Reed Muller [25]	2048	2048	$5.37 \cdot 10^{-10}$	14%
GC Reed Solomon [25]	1024	1024	$3.47 \cdot 10^{-10}$	14%
Polar Codes SC [10]	128	896	10^{-6}	15%
HA SCL Polar Codes [10]	128	896	10^{-9}	15%
Proposed Key Generation Scheme using SC decoder	250	262	10^{-8}	15%
Proposed Key Generation Scheme using BP decoder	250	262	10^{-10}	15%

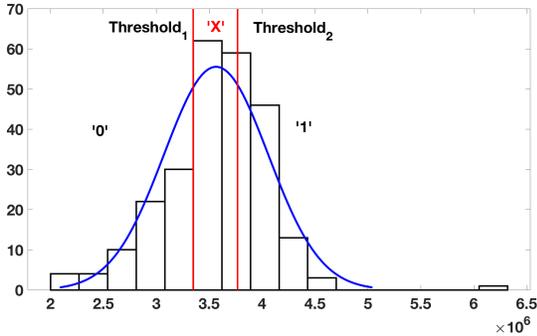


Fig. 4: Distribution of 254 ReRAM cells and extraction of stable 0's and 1's using $Threshold_1$ and $Threshold_2$

255 BCH encoded PUF data bits were encoded using a (512, 255) Polar Encoder. At the output of the Polar Encoder, the frozen bits (262 bits) were selected as *Helper Data* while the unfrozen bits were hashed using a “SHA-256” and stored in the server as the *Secret Key*. During the key regeneration phase, the repeated data measurements from 254 cells were used to generate a *noisy key* by selecting *unfrozen bits* at the output of a Polar encoder. The *Helper Data* (262 bits) that was extracted during the key generation phase, was combined with the *noisy key* and given to a Polar-BCH serial decoder to generate an estimate of the initial 131 PUF data bits. This estimate is used to regenerate the *Key* by giving it to a serially concatenated BCH-Polar encoder

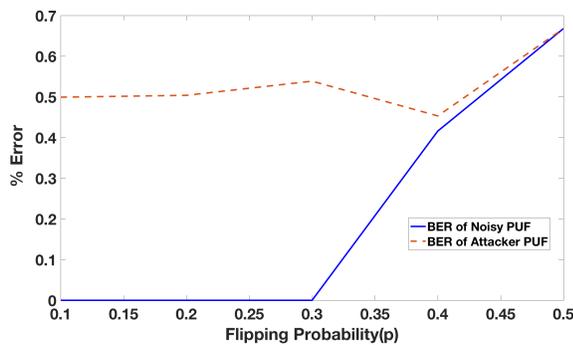


Fig. 5: Percentage error between Key generated from Noisy or Attacker PUF and Key stored in the server

and selecting the *unfrozen bits*, which is then hashed and compared against the *Key* stored in the server.

The proposed method was also tested using simulated data in order to test the efficiency of the fuzzy extractor to regenerate keys. A noisy version of the initial data with a flipping probability of 15% was used to regenerate the key. The fuzzy extractor could regenerate the key exactly with a failure probability of 10^{-8} while using a low complex Successive Cancellation decoder and 10^{-10} while using an iterative Belief Propagation decoder.

The percentage of error in regenerating the *Key* using *Helper Data* for original PUF data under different simulated flipping probabilities was compared with an Attacker PUF with an inter-Hamming distance $\geq 40\%$ and the results were plotted in Figure 6. The results indicated that, if the flipping probability is $\leq 30\%$, then the proposed key generation scheme can exactly recover the *Key* using *Helper Data*. The attacker was not successful in recovering the original PUF data by using the publicly available *Helper Data* and the percentage of error when the Attacker PUF is used to regenerate the key was about 50%.

The performance of different Polar Decoders was tested and plotted in Figure 6, which shows that the Belief Propagation Decoder has a better performance compared to the SC and SC List decoders. This increase in performance comes with an increase in time complexity in regenerating the key which is compared in Figure 7.

Table I shows the comparison between the failure probability of key regeneration using publicly available helper data utilizing different fuzzy extractor schemes proposed

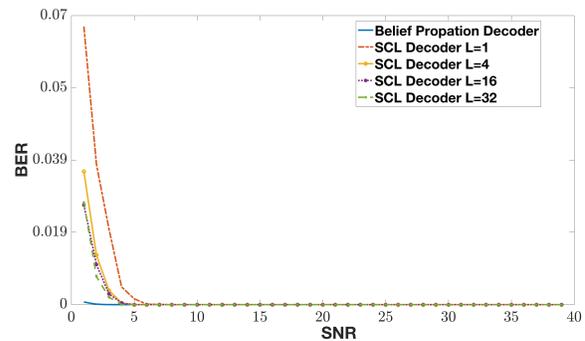


Fig. 6: BER of the Regenerated Key while using different Polar Decoders

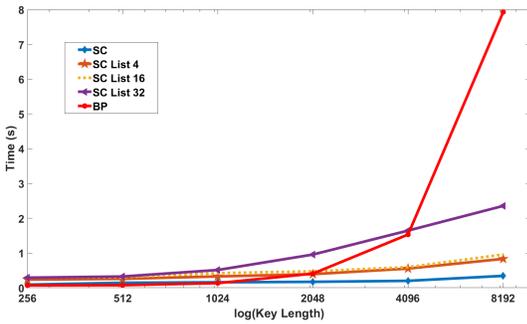


Fig. 7: Time complexity comparison of the Fuzzy extractor while using different Polar decoders.

in the literature. As seen in this table, our proposed method results in a failure probability of 10^{-8} using a low complex Successive Cancellation decoder for key regeneration with a key length of 250 bits while using 262 bits of helper data, while the competitive methods require a large number of helper bits to be able to regenerate the keys with such a low probability of mismatch. This failure probability was further reduced to 10^{-10} when a complex Belief Propagation decoder was utilized.

V. CONCLUSION

The ReRAM based PUFs are the most practical choice for authentication and key generation in IoT, as they operate at or below the systems' noise level and therefore are less vulnerable to side channel attacks compared to the alternative memory technologies. However, the current ReRAM-based PUFs present a high false negative authentication rate since the behavior of these devices can vary in different physical conditions that results in a low probability of regenerating the same response in different attempts. In this paper, we propose a secret key generation scheme for ternary state PUFs that enables reliable reconstruction of the desired secret keys utilizing a serially concatenated BCH-Polar fuzzy extractor. The experimental results show that the proposed model can offer a significantly lower probability of mismatch between the original key and the regenerated ones, while a less number of *Helper data* bits were used to extract the *Key* when compared to previously proposed fuzzy extractor techniques.

REFERENCES

- [1] A. R. Korenda, M. Zaeri-Amirani, and F. Afghah, "A hierarchical stackelberg-coalition formation game theoretic framework for cooperative spectrum leasing," in *Information Sciences and Systems (CISS), 2017 51st Annual Conference on*. IEEE, 2017, pp. 1–6.
- [2] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *International conference on the theory and applications of cryptographic techniques*. Springer, 2004, pp. 523–540.
- [3] K. Shamsi and Y. Jin, "Security of emerging non-volatile memories: Attacks and defenses," in *VLSI Test Symposium (VTS), 2016 IEEE 34th*. IEEE, 2016, pp. 1–4.
- [4] P. FremantEAU and D. Naccache, "Unforgeable identification device, identification device reader and method of identification," May 6 1999, eP Patent 0,583,709. [Online]. Available: <https://www.google.com/patents/EP0583709B1?cl=en>
- [5] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.

- [6] F. Afghah, B. Cambou, M. Abedini, and S. Zeadally, "A reram physically unclonable function (reram puf)-based approach to enhance authentication security in software defined wireless networks," *International Journal of Wireless Information Networks*, pp. 1–13, 2017.
- [7] S. Katzenbeisser, Ü. Kocabaş, V. Rozić, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, "Pufs: Myth, fact or busted? a security evaluation of physically unclonable functions (pufs) cast in silicon," *Cryptographic Hardware and Embedded Systems-CHES 2012*, pp. 283–301, 2012.
- [8] H. Kang, Y. Hori, T. Katashita, M. Hagiwara, and K. Iwamura, "Cryptographic key generation from puf data using efficient fuzzy extractors," in *Advanced Communication Technology (ICACT), 2014 16th International Conference on*. IEEE, 2014, pp. 23–26.
- [9] H. Kang, Y. Hori, T. Katashita, and M. Hagiwara, "The implementation of fuzzy extractor is not hard to do: An approach using puf data," in *Proceedings of the 30th Symposium on Cryptography and Information Security, Kyoto, Japan, 2013*, pp. 22–25.
- [10] B. Chen, T. Ignatenko, F. M. Willems, R. Maes, E. van der Sluis, and G. Selimis, "High-rate error correction schemes for sram-pufs based on polar codes," *arXiv preprint arXiv:1701.07320*, 2017.
- [11] H. Kang, Y. Hori, T. Katashita, M. Hagiwara, and K. Iwamura, "Performance analysis for puf data using fuzzy extractor," in *Ubiquitous Information Technologies and Applications*. Springer, 2014, pp. 277–284.
- [12] J. Delvaux, D. Gu, I. Verbauwhede, M. Hiller, and M.-D. M. Yu, "Efficient fuzzy extraction of puf-induced secrets: Theory and applications," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2016, pp. 412–431.
- [13] M. Taniguchi, M. Shiozaki, H. Kubo, and T. Fujino, "A stable key generation from puf responses with a fuzzy extractor for cryptographic authentications," in *Consumer Electronics (GCCE), 2013 IEEE 2nd Global Conference on*. IEEE, 2013, pp. 525–527.
- [14] B. Cambou and F. Afghah, "Physically unclonable functions with multi-states and machine learning," in *14th International Workshop on Cryptographic Architectures Embedded in Logic Devices (CryptArch)*, 2016.
- [15] C. Helfmeier, C. Boit, D. Nedospasov, S. Tajik, and J.-P. Seifert, "Physical vulnerabilities of physically unclonable functions," in *Proceedings of the conference on Design, Automation & Test in Europe*. European Design and Automation Association, 2014, p. 350.
- [16] B. Cambou and M. Orlowski, "Puf designed with resistive ram and ternary states," in *Proceedings of the 11th Annual Cyber and Information Security Research Conference*. ACM, 2016, p. 1.
- [17] P. Pavan, R. Bez, P. Olivo, and E. Zanoni, "Flash memory cells—an overview," *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1248–1271, 1997.
- [18] B. Cambou, "Random number generating systems and related methods," May 9 2016, uS Patent App. 15/150,179.
- [19] M.-D. M. Yu, D. MRaihi, R. Sowell, and S. Devadas, "Lightweight and secure puf key storage using limits of machine learning," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2011, pp. 358–373.
- [20] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.
- [21] B. Cambou, F. Afghah, D. Sonderegger, J. Taggart, H. Barnaby, and M. Kozicki, "Ag conductive bridge rams for physical unclonable functions," in *Hardware Oriented Security and Trust (HOST), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 151–151.
- [22] F. Afghah and B. F. Cambou, "Multi-state unclonable functions and related systems," May 18 2017, uS Patent App. 15/348,047.
- [23] R. Maes, P. Tuyls, and I. Verbauwhede, "Low-overhead implementation of a soft decision helper data algorithm for sram pufs," in *CHES*, vol. 9. Springer, 2009, pp. 332–347.
- [24] R. Maes, A. Van Herrewege, and I. Verbauwhede, "Pufky: A fully functional puf-based cryptographic key generator," *Cryptographic Hardware and Embedded Systems-CHES 2012*, pp. 302–319, 2012.
- [25] S. Puchinger, S. Muelich, M. Bossert, M. Hiller, and G. Sigl, "On error correction for physical unclonable functions," in *SCC 2015; 10th International ITG Conference on Systems, Communications and Coding; Proceedings of*. VDE, 2015, pp. 1–6.