PUF-COTE: A PUF Construction with Challenge Obfuscation and Throughput Enhancement

Boyapally Harishma[1], Durba Chatterjee[1], Kuheli Pratihar[1], Sayandeep Saha[1], and Debdeep Mukhopadhyay[1]

Indian Institute of Technology Kharagpur
{harishmasko, durba.chatterjee94, its.kuheli96 sayandeep.iitkgp, debdeep.mukhopadhyay}@gmail.com

Abstract. Physically Unclonable Functions (PUFs) have been a potent choice for enabling low-cost, secure communication. However, the state-of-the-art strong PUFs generate single-bit response. So, we propose PUF-COTE: a high throughput architecture based on linear feedback shift register and a strong PUF as the "base"-PUF. At the same time, we obfuscate the challenges to the "base"-PUF of the final construction. We experimentally evaluate the quality of the construction by implementing it on Artix 7 FPGAs. We evaluate the statistical quality of the responses (using NIST SP800-22 test suit and standard PUF metrics: uniformity, uniqueness, reliability, strict avalanche criterion, ML-based modelling), which is a crucial factor for cryptographic applications.

Keywords: Physically Unclonable Functions · Obfuscated Challenge · Throughput Enhancement

1 Introduction

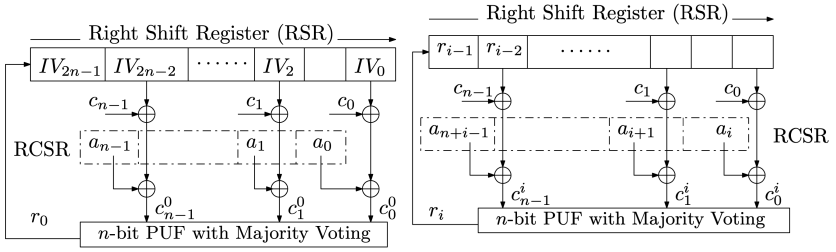
Physically Unclonable Functions (PUFs) have been established as potential candidates to solve the secret storage problem of traditional cryptography based solutions. Based on uncontrollable manufacturing process variations, PUFs generate uniformly random bit(s) as the output response (making it suitable for cryptographic applications) when queried with an input bit-string (known as a challenge) [5, 7]. The fascinating feature of a PUF is that the responses are *unpredictable* and depend solely upon the physical characteristics of the device on which it is instantiated. For the same PUFs instantiated on two different ICs, although the implemented circuit is the same, the responses are statistically independent or *unique*, thus working as a device-level fingerprint. These factors make PUF a strong candidate for on-the-fly secret generation, removing the need for highly fortified NVMs.

Based on the size of the CRP space, PUFs are classified as *strong PUFs* and *weak PUFs*. Strong PUFs usually produce a CRP space of exponential size, while weak PUFs have a much smaller CRP space. Strong PUFs are the perfect candidates for applications requiring low-cost authentication and key establishment. However, they have low throughput (typically a single-bit response corresponding to an n-bit challenge), which somewhat hinders the building of strong cryptographic protocols. They are also expected to be unclonable even if a large fraction of CRP space is exposed publicly. A well-known approach to fortify against machine learning (ML)-based modelling attacks is to enhance the randomness of the challenge-response relationship by introducing non-linearity in the design [11, 9, 14]. However, these constructions are also shown to be vulnerable to modeling attacks specific to the target construction [13, 16, 2]. An alternative strategy is to obfuscate the challenge-response behavior by hiding or transforming the raw inputs and outputs [8, 12, 9, 6] or restrict the number of CRPs that can be exposed from a given instance [4]. This class of constructions involves an input transformation network that generates internal challenges to be fed to the internal PUF, and the outputs are modified before producing the final response. In [12], a recurrence-based PUF construction is proposed that generates internal challenge using the response of the *core* PUF and the external input. The response to the internal challenge is produced as output. The feedback mechanism obscures the challenge-response relationship of the core/internal PUF. So far, to the best of our knowledge, no attack has been proposed for this construction. However, they did not demonstrate a mechanism to increase the throughput of the PUF such that it can be adopted for cryptographic applications with sufficient security.

This paper addresses these issues associated with PUFs for their eventual application in secure communication. We present PUF-COTE: a construction based on feedback shift registers. While using a Linear Feedback Shift Register (LFSR) to improve the throughput of strong PUFs has been explored previously [3], we propose PUF-COTE construction (refer to Figure 1), where the (unknown and unpredictable) response bits modify a known challenge to generate several challenges internally. The internal challenges are constructed so that they remain unknown to an adversary. In this way, we obfuscate the relation between the external challenge and the final throughput-enhanced response.

2 PUF-COTE: High Throughput Construction with Challenge Obfuscation

The common practice for increasing the throughput of a PUF is to use LFSRs [15]. The technique is to generate multiple challenges from a single challenge seed and feed them serially to the PUF. The corresponding responses are concatenated to obtain more response bits. However, in such constructions, the adversary knows the feedback polynomial. Interestingly, we found it possible to improve the construction mentioned above to develop a high-throughput strong PUF while obfuscating the challenge-response behavior. More precisely, we present an n-bit strong PUF-based architecture that enhances the through-



(a) Initial state of RSR and RCSR.

(b) State of RSR and RCSR in i-th round.

Fig. 1: PUF-COTE: Architecture to improve PUF throughput and challenge obfuscation. We describe the initial and i-th state of right shift register (RSR) and right circular shift register (RCSR).

put from 1-bit to $2n$-bits. The main idea is to replace the feedback polynomial of LFSR with a PUF. This in turn hides the relation between input challenge and the $2n$-bit output response.

The proposed PUF-COTE architecture to improve the throughput of a strong PUF is illustrated in Figure 1. It takes an n-bit challenge $C = (c_{n-1}, \ldots, c_0)$ as input and outputs a $2n$-bit response $\widetilde{R} = (r_{2n+1}, \ldots, r_2)$. Our architecture contains an n-bit *internal* PUF primitive that outputs a 1-bit response; an n-bit right circular shift register (RCSR) for supplying round-constant; and a $2n$-bit feedback right shift register (RSR) for collectively generating the $2n$ bit response. Additionally, we have a majority voting block to increase the reliability of the internal PUF. As shown in Figure 1a, in 0-th round RSR is initialized with $IV = (IV_{2n-1}, \ldots, IV_1, IV_0)$ which is denoted as S^0 and n-bit RCSR is initialized with round-constant $(a_{n-1}, \ldots, a_0)$ where, $\Sigma_{j=0}^{n-1} a_j \neq 0$. For 0-th round the input to the internal PUF denoted by $C^{int,0}$ is computed as

$$\begin{aligned} C^{int,0} &= (c^0_{n-1}, \ldots, c^0_j, \ldots, c^0_0) \\ &= \Big(\big(IV_{2n-2} \oplus c_{n-1} \oplus a_{n-1}\big), \ldots\ldots, \big(IV_{2j} \oplus c_j \oplus a_j\big), \ldots, \big(IV_0 \oplus c_0 \oplus a_0\big) \Big) \end{aligned} \tag{1}$$

Note that only even index bits of the RSR are XORed with the challenge bits and the contents of RCSR to determine the challenge to the internal PUF. The corresponding majority voted 1-bit response r_0 is then fed to the RSR changing its state as $S^1 := (r_0, IV_{2n-1}, IV_{2n-2}, \cdots, IV_2, IV_1)$ and the state of RCSR is updated to $(a_0, a_{n-1}, a_{n-2} \ldots, a_1)$. For 1-st round, the internal challenge is given as

$$\begin{aligned} C^{int,1} &= (c^1_{n-1}, \ldots, c^1_j, \ldots, c^1_0) \\ &= \Big(\big(IV_{2n-1} \oplus c_{n-1} \oplus a_0\big), \ldots, \big(IV_{2j+1} \oplus c_j \oplus a_{j+1}\big), \ldots, \big(IV_1 \oplus c_0 \oplus a_1\big) \Big) \end{aligned} \tag{2}$$

The corresponding majority voted 1-bit response r_1 is then fed to RSR changing its state as $S^2 := (r_1, r_0, IV_{2n-1}, IV_{2n-2}, \cdots, IV_3, IV_2)$ and the state of RCSR is updated to $(a_1, a_0, a_{n-1} \ldots, a_2)$.
In this way, for $i \in [2, 2n-1]$-th round (refer to Figure 1b) the internal challenge is given as

$$C^{int,i} = (c^i_{n-1}, \ldots, c^i_j, \ldots, c^i_0) = \Big(\big(r_{i-2} \oplus c_{n-1} \oplus a_{i-1}\big), \ldots, \big(IV_i \oplus c_0 \oplus a_i\big) \Big) \quad (3)$$

After $2n-1$ rounds, the corresponding majority voted 1-bit response r_{2n-1} is then fed to RSR. In $2n$-th round, the state of RSR is $S^{2n} := (r_{2n-1}, r_{2n-2}, , \ldots, IV_{2n})$, RCSR is $(a_{i-1}, a_{i-2}, \ldots, a_{i+1}, a_i)$. Then, the internal challenge is given as

$$C^{int,2n} = (c^{2n}_{n-1}, c^{2n}_{n-2}, \ldots, c^{2n}_0) = \Big(\big(r_{2n-2} \oplus c_{n-1} \oplus a_{2n-1}\big), , \ldots, \big(r_0, c_0, a_i\big) \Big) \quad (4)$$

The corresponding majority voted 1-bit response r_{2n} is fed to RSR. Thus, in $2n+1$-th round, the state of RSR is $S^{2n+1} := (r_{2n}, r_{2n-1}, \ldots, r_2, r_1)$, RCSR is $(a_0, a_{n-1}, \ldots, a_2, a_1)$ and the internal challenge is

$$\begin{aligned} C^{int,2n+1} = \Big(&\big(r_{2n-1} \oplus c_{n-1} \oplus a_0\big), \big(r_{2n-3} \oplus c_{n-2} \oplus a_{n-1}\big), \ldots \\ &\ldots, \big(r_3 \oplus c_1 \oplus a_2\big), \big(r_1 \oplus c_0 \oplus a_1\big) \Big) \end{aligned} \quad (5)$$

Finally, the corresponding majority voted 1-bit response r_{2n+1} is fed to RSR, changing its state to S^{2n+2}, which is the $2n$-bit response of our construction. It is given as:

$$\widetilde{R} = S^{2n+1} = \big(r_{2n+1}, r_{2n}, \ldots, r_3, r_2\big) \quad (6)$$

We make the following observations from the construction described in Figure 1:

1. The response bits r_0 and r_1 depend solely on the internal PUF, challenge C, state of RCSR and initialization vector IV (refer to Equations 1 and 2).

2. The response bits $r_2, \ldots, r_{2n+2}$ depend on the internal PUF, challenge C, the state of RCSR and initialization vector IV as well as the response bits r_0 or r_1 (refer to Equations 3, 4 and 6).

We obfuscate the relation between challenge (C) and response ($\widetilde{R}$) by discarding the bits r_0 and r_1 thus, hiding the challenges corresponding to the response bits r_i where $i \in [2, 2n+1]$

The independence and uniformity of the response bits for PUF-COTE is established via NIST SP800-22 Statistical Test Suite [1] in Section 3.

3 Experiments And Results

In this section we provide the hardware implementation details of PUF-COTE construction on Artix-7 FPGAs. We mainly evaluate the statistical quality of

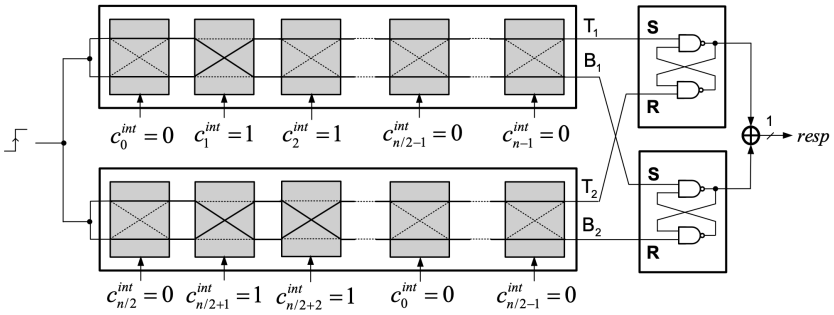


Fig. 2: 2-1 DAPUF with input transformation

the responses (using NIST SP800-22 test suite and standard metrics: uniformity, uniqueness, reliability, machine learning resistance and Strict Avalanche Criterion (SAC)) which is a key factor for cryptographic applications.

We implement the PUF-COTE construction on Xilinx Artix-7 FPGA. The construction comprises of a 64-bit PUF, a 128-bit shift register, a 64-bit LCSR and a majority voting module. The output is a 128-bit random string for a 64-bit external challenge and 128-bit harmonizing vector. We use $2-1$ Double Arbiter PUF (DAPUF) (with a input transformation) for proof-of-concept realization of the internal PUF. The schematic of the internal PUF is given in Figure 2. Unlike a classical DAPUF construction that provides the same challenge to all delay chains, in our construction, we rotate the challenge given to the lower delay chain by $n/2$ bits ($n = 64$). This input transformation ensures that on flipping a particular challenge bit, different stage delays are affected in each delay chain,

Test Name	**p-value**	**Result**
Frequency	0.743416	*PASS*
Block Frequency	0.587244	*PASS*
Cumulative Sums	0.419383	*PASS*
Runs	0.534146	*PASS*
Longest Run	0.668291	*PASS*
Rank	0.350485	*PASS*
FFT	0.383287	*PASS*
Non Overlapping Template	0.213390	*PASS*
Overlapping Template	0.911413	*PASS*
Universal	0.017912	*PASS*
Approximate Entropy	0.098524	*PASS*
Random Excursions	0.506931	*PASS*
Random Excursions Variant	0.064112	*PASS*
Serial	0.478694	*PASS*
Linear Complexity	0.161284	*PASS*

Table 1: Statistical Test Results for Randomness using NIST SP800-22 [1]

PUF Design	Uniformity	Uniqueness	Reliability (per bit)
Modified 2-1 DAPUF	45.9	32.33	97.7
PUF-COTE	47.32	39.33	98.2

Table 2: PUF Quality Metrics

thereby diminishing the influence of a challenge bit on the final response and improving the Strict Avalanche Criteria (SAC) property [14]. We would like to note that we use the modified 2-1 DAPUF for prototype implementation and the designer can choose any strong PUF design with good quality metrics.

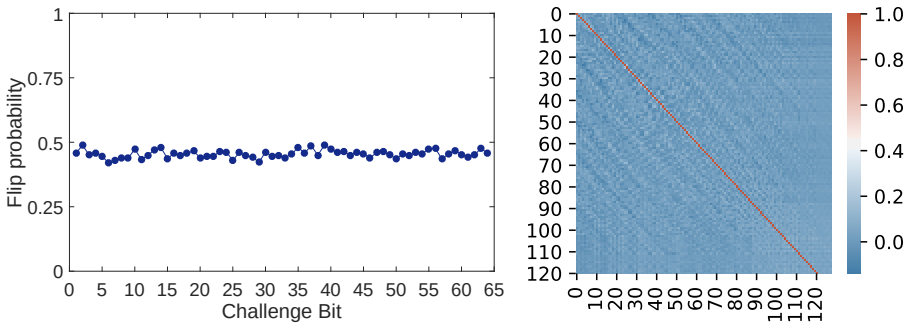
PUFs in general are not 100% reliable. We noticed the reliability of the chosen 2-1 DAPUF to be 97.7%. So, we incorporate a majority voting block along with the 2-1 DAPUF in the construction. For 99 internal majority voting, the reliability of the internal PUF is enhanced to 99.97%. However, due to the recurrent nature of the design, the unreliability of the previous response bit affects the reliability of the next response bits. To account for this, we employ an external majority voting and observed that the final 128-bit response has a reliability of 99.6%. Since, an error rate of 5% is tolerable for strong PUFs [10], our design is suitable for cryptographic applications.

We evaluate the proposed construction using standard PUF quality metrics computed over 20K CRPs, 25 external measurements obtained from 7 FPGA boards. The results are presented in Table 2.

We also evaluate the SAC property of internal PUF over a set of 1000 randomly generated challenges. Figure 3a depicts that the flip probability of a randomly chosen response bit is close to 0.5, upon flipping one challenge bit at a time. This implies that the PUF-COTE construction satisfies SAC. Figure 3b shows that the correlation for each pair of bits in the 128-bit PUF response lies in the range $[-0.024, 0.077]$. Finally, we validate the randomness of the final response using the well-known NIST SP800-22 Statistical Test Suite over 1 million CRPs. The construction passes all 15 tests as shown in Table 1 with p-value greater than 0.01 (default threshold for NIST SP800-22 test suite) thereby demonstrating good quality randomness.

We assess the unpredictability of PUF-COTE response using classical ML algorithms such as Logistic Regression (LR), Support Vector Machine (SVM) and Random Forest (RF). The models are trained using 40K CRPs using 5-fold cross-validation and test the model using 10K CRPs. We observe that each bit of the 128-bit response can be modelled with accuracy ranging $[50, 53]\%$, which is close to making a random guess. This demonstrates the robustness of our PUF-COTE construction against classical ML-attacks.

Finally, the entire implementation comprises 395 LUTs, 264 FFs and has a critical path of 1.835 ns. The time taken to generate the 128-bit response of PUF-COTE is 780μ seconds.



(a) SAC Property

(b) Pairwise Correlation of output bits

Fig. 3: SAC and Correlation Analysis Results. SAC is calculated over 1000 CRPs and correlation is computed over 50K CRPs

4 Conclusion

State-of-the-art PUF-assisted communication protocols require at least one party or some TTP to securely store the CRP database. This often becomes a usability issue as having a truly secure memory for low-end devices is challenging. In this paper, we present PUF-COTE: a high-throughput construction with challenge-obfuscation for strong PUFs. For proof-of-concept we used the 2-1 DAPUF design with an input transformation mechanism. We evaluated the SAC property and validated the randomness using NIST SP800-22 test suite. Finally, we analysed the machine learning resistance over classical ML techniques.

References

1. Bassham, L.E., Rukhin, A.L., Soto, J., Nechvatal, J.R., Smid, M.E., Barker, E.B., Leigh, S.D., Levenson, M., Vangel, M., Banks, D.L., Heckert, N.A., Dray, J.F., Vo, S.: Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications. Tech. rep., Gaithersburg, MD, USA (2010)
2. Chatterjee, D., Chatterjee, U., Mukhopadhyay, D., Hazra, A.: Sacred: An attack framework on sac resistant delay-pufs leveraging bias and reliability factors. In: 2021 58th ACM/IEEE Design Automation Conference (DAC). pp. 85–90. IEEE (2021)
3. Dubrova, E., Näslund, O., Degen, B., Gawell, A., Yu, Y.: CRC-PUF: A machine learning attack resistant lightweight PUF construction. In: 2019 IEEE European Symposium on Security and Privacy Workshops, EuroS&P Workshops 2019, Stockholm, Sweden, June 17-19, 2019. pp. 264–271. IEEE (2019). https://doi.org/10.1109/EuroSPW.2019.00036, https://doi.org/10.1109/EuroSPW.2019.00036
4. Ganji, F., Tajik, S., Stauss, P., Seifert, J.P., Tehranipoor, M., Forte, D.: Rock'n'roll pufs: crafting provably secure pufs from less secure ones (extended version). Journal of Cryptographic Engineering **11**(2), 105–118 (2021)

5. Gassend, B., Clarke, D., Van Dijk, M., Devadas, S.: Silicon physical random functions. In: Proceedings of the 9th ACM Conference on Computer and Communications Security. pp. 148–160 (2002)
6. Lalouani, W., Younis, M., Ebrahimabadi, M., Karimi, N.: Countering modeling attacks in puf-based iot security solutions. ACM Journal on Emerging Technologies in Computing Systems (JETC) **18**(3), 1–28 (2022)
7. Lim, D., Lee, J.W., Gassend, B., Suh, G.E., Van Dijk, M., Devadas, S.: Extracting secret keys from integrated circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems **13**(10), 1200–1205 (2005)
8. Majzoobi, M., Koushanfar, F., Potkonjak, M.: Lightweight secure pufs. In: Nassif, S.R., Roychowdhury, J.S. (eds.) 2008 International Conference on Computer-Aided Design, ICCAD 2008, San Jose, CA, USA, November 10-13, 2008. pp. 670–673. IEEE Computer Society (2008). https://doi.org/10.1109/ICCAD.2008.4681648, https://doi.org/10.1109/ICCAD.2008.4681648
9. Nguyen, P.H., Sahoo, D.P., Jin, C., Mahmood, K., Rührmair, U., van Dijk, M.: The interpose PUF: secure PUF design against state-of-the-art machine learning attacks. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2019**(4), 243–290 (2019). https://doi.org/10.13154/tches.v2019.i4.243-290, https://doi.org/10.13154/tches.v2019.i4.243-290
10. Rozic, V., Yang, B., Vliegen, J., Mentens, N., Verbauwhede, I.: The monte carlo PUF. In: Santambrogio, M.D., Göhringer, D., Stroobandt, D., Mentens, N., Nurmi, J. (eds.) 27th International Conference on Field Programmable Logic and Applications, FPL 2017, Ghent, Belgium, September 4-8, 2017. pp. 1–6. IEEE (2017). https://doi.org/10.23919/FPL.2017.8056780, https://doi.org/10.23919/FPL.2017.8056780
11. Sahoo, D.P., Mukhopadhyay, D., Chakraborty, R.S., Nguyen, P.H.: A multiplexer-based arbiter puf composition with enhanced reliability and security. IEEE Transactions on Computers **67**(3), 403–417 (2017)
12. Shah, N., Chatterjee, D., Sapui, B., Mukhopadhyay, D., Basu, A.: Introducing recurrence in strong pufs for enhanced machine learning attack resistance. IEEE J. Emerg. Sel. Topics Circuits Syst. **11**(2), 319–332 (2021). https://doi.org/10.1109/JETCAS.2021.3075767, https://doi.org/10.1109/JETCAS.2021.3075767
13. Shi, J., Lu, Y., Zhang, J.: Approximation attacks on strong pufs. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **39**(10), 2138–2151 (2019)
14. Siddhanti, A.A., Bodapati, S., Chattopadhyay, A., Maitra, S., Roy, D., Stanica, P.: Analysis of the strict avalanche criterion in variants of arbiter-based physically unclonable functions. Lecture Notes in Computer Science, vol. 11898, pp. 556–577 (2019). https://doi.org/10.1007/978-3-030-35423-7_28, https://doi.org/10.1007/978-3-030-35423-7_28
15. Srinivasu, B., Pudi, V., Chattopadhyay, A., Lam, K.: Colpuf : A novel configurable lfsr-based PUF. In: 2018 IEEE Asia Pacific Conference on Circuits and Systems, APCCAS 2018, Chengdu, China, October 26-30, 2018. pp. 358–361. IEEE (2018). https://doi.org/10.1109/APCCAS.2018.8605643, https://doi.org/10.1109/APCCAS.2018.8605643
16. Wisiol, N., Mühl, C., Pirnay, N., Nguyen, P.H., Margraf, M., Seifert, J.P., van Dijk, M., Rührmair, U.: Splitting the interpose puf: A novel modeling attack strategy. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 97–120 (2020)