Trustworthy Proofs for Sensor Data using FPGA based Physically Unclonable Functions

Urbi Chatterjee*, Durga Prasad Sahoo†, Debdeep Mukhopadhyay*, Rajat Subhra Chakraborty*
* Secured Embedded Architecture Laboratory, IIT Kharagpur, India, {urbi.chatterjee,debdeep,rschakraborty}@cse.iitkgp.ernet.in
† Bosch India (RBEI/ETI), dpsahoo.cs@gmail.com

***Abstract*—The Internet of Things (IoT) is envisaged to consist of billions of connected devices coupled with sensors which generate huge volumes of data enabling control-and-command in this paradigm. However, integrity of this data is of utmost concern, and is promisingly addressed leveraging the inherent unreliability of *Physically Unclonable Functions* (PUFs) w.r.t. ambient parameter variations, using the concept of *Virtual Proofs* (VPs). Advantage of these protocols is that they do not use explicit keys and aim at proving the authenticity of the sensor. Since the existing PUF-based protocols do not use the sensor data as a part of challenge (i.e. input) to PUFs, there is no guarantee of uniqueness of PUF's challenge-response behavior over multiple levels of ambient parameters. Few of these protocols needs to sequential search in the challenge-response database. To alleviate these issues, we develop a new class of authenticated sensing protocols where the sensor data is combined with the external challenge by utilizing the Strict Avalanche Criterion of the PUF. We validate the proposed protocol through actual experiments on FPGA using *Double Arbiter PUFs* (DAPUFs), which are implemented with superior uniformity, uniqueness, and reliability on Xilinx Artix-7 FPGAs. According to the FPGA-based validation, the proposed protocol with DAPUF can be effectively used to authenticate wide variations of temperature from $-20^{\circ}C$ to $80^{\circ}C$.**

***Index Terms*—Authenticated Sensing, double arbiter PUFs, FPGA, physically unclonable functions (PUFs), reliability, trustworthy proofs, virtual proofs (VPs).**

I. Introduction

The ubiquitous sensing-communicating-actuating network of wireless sensors has brought several security threats in the Internet of Things (IoTs) due to resource constraints and inadequate lightweight security protocols. Hence, the validation of authenticity of data collected from wireless sensors has become a primary concern. Physically unclonable Functions (PUFs) [1] have been witnessed as a promising unconventional hardware security primitive and appeared to act as a root-of-trust for resource constrained devices. One of the desirable properties of a PUF circuit is *reliability* which implies that a particular PUF instance should generate same response repeatedly to a given challenge. However, it can be influenced by different environmental factors. Recent works [2]–[9] have exploited the unreliability property to validate the integrity of the sensor data. Specifically, Rührmair et al. [3] proposed an authentication technique called *VP of reality*, and Gao et al. [9] proposed an authenticated sensing protocol utilizing the PUF unreliability originated from its sensitivity to ambient parameter variations. The major advantage of these protocols is that they alleviate the use of secret keys. However, these protocols have the following issues: *no guarantee of uniquenees for responses over multiple intervals of ambient parameters, storage space overhead at the verifier side, and sequential search in database*. In this work, we propose a new authenticated temperature sensing protocol which mitigates these issues. In addition, we have validated the feasibility of proposed protocol on FPGA using Double Arbiter PUF (DAPUF) [10].

The rest of the paper is organized as follows. Background and related works are introduced in Section II. In Section III, we propose the new authenticated sensing protocol. The trustworthiness proof against forgery has been discussed in Section IV. The experimental result is reported in Section V. We conclude our work and discuss the future research directions in Section VI.

II. Background

In this section, we briefly explain the two protocols of virtual proofs and their shortcomings. Note that we use the notation $[a, b]$ to denote a set of numbers $\{a, a+1, \ldots, b-1, b\}$ and $a \leq b$.

A. Protocol for Virtual Proof of Temperature [3]

Let $t_1, \ldots, t_k$ be k distinct discrete-temperature levels which span over the temperature range R_T. This protocol consists of two phases: *Setup Phase* and *Authentication Phase*. In *Setup Phase*, for each temperature level $t_i \in \{t_1, \ldots, t_k\}$, the verifier randomly generates a set of m challenges $\{C_j^i\}_{j\in[1,m]}$ and evaluates the PUF to build the corresponding response set $\{R_j^i\}_{j\in[1,m]}$. The verifier securely stores these information in a database $L = \{(C_j^i, R_j^i, t_i)\}_{i\in[1,k],j\in[1,m]}$, and deploys the PUF in the field. In *Authentication Phase*, the prover (having a temperature sensor) senses the current temperature T and send it to the verifier. Upon receiving T, and the verifier finds the near by temperature level $t \in \{t_1, \ldots, t_k\}$ for temperature T. The verifier randomly selects a set of n elements $V = \{(C_i, R_i, t_i)\}_{i\in[1,n]}$ from L where $t_i = t$, and sends the challenges $\{C_i\}_{i\in[1,n]}$ to the prover. The prover evaluates the PUF with challenges in $\{C_i\}_{i\in[1,n]}$, and sends the corresponding responses $\{R_i^*\}_{i\in[1,n]}$ to the verifier. The verifier compares $\{R_i^*\}_{i\in[1,n]}$ with the stored $\{R_i\}_{i\in[1,n]}$ in element-by-element fashion. If all values match, it accepts the

978-3-9819263-0-9/DATE18/©2018 EDAA

VP and updates the database as $L = L \setminus V$. Otherwise, it rejects.

Shortcomings. The temperature levels are defined independent of the PUF challenge-response behaviors. There might be a case where the PUF behaviors are similar for two distict temperature levels. Thus, there is no guarantee of uniqueness of challenge-response behaviors over multiple temperature intervals.

B. Protocol for Virtual Proof of Voltage [9]

This protocol consists of two phases: *Enrolment* and *Authentication*. In *Enrolment Phase*, the verifier selects p discrete voltage levels $Q_1, \ldots, Q_p$ to cover the voltage range R_V. For each applied challenge C_i, the response is evaluated as $R_i^{Q_j} = F_{PUF}(C_i, Q_j)$, which is actually the response of PUF to challenge C_i at voltage Q_j. The verifier securely stores the measured CRP in the database L. Then, the PUF-enabled device is deployed in field. Next, in *Authentication Phase*, the verifier sends a challenge C to the prover, and the PUF is evaluated with the challenge C at some unknown voltage Q_u, and the response R^{Q_u} is sent back to the verifier. The verifier compares R^{Q_u} with each recorded response $R^{Q_j}, j \in [1, p]$ for a match. If there is a match (i.e. if $R^{Q_j} = R^{Q_u}$), the voltage at the prover end is inferred to be Q_j.

Shortcomings. If the CRPs for each voltage level are not unique, then even a successful authentication might infer a wrong voltage level. This can happen when the first match for the received response is detected, the corresponding voltage is immediately declared to be the voltage level of prover. Other disadvantage is that the verifier needs to search the CRP database sequentially, and hence, it increases the authentication time.

III. Proposed Authenticated Sensing Protocol

In this section, we propose an architecture of the prover consisting of a PUF instance and a new authenticated sensing protocol for virtual proof.

A. System Model

Figure 1 depicts the architectural overview of the prover setup. Although we use this setup for temperature sensing applications; however, it is generic enough to be adapted to establish the authenticity of other types of sensors (cf. Sensor in Figure 1). The setting assumed is that in certain intervals of time the prover senses the temperature through a sensor and sends it to the verifier. Before accepting the sensor data, the verifier checks the authenticity of the data. Thus, the system does not require the VP for every value, rather for a range of temperature changes. Unlike the previous protocols, we use temperature intervels instead of some discrete values of the ambient factors. The partition of a given temperature range can be performed by observing the changes in challenge-response behavior of the PUF (cf. PUF in Figure 1). All discrete-temperature values should be part of the same interval if there is no (significant) change in PUF behavior for these temperature values with respect to a given challenge set.

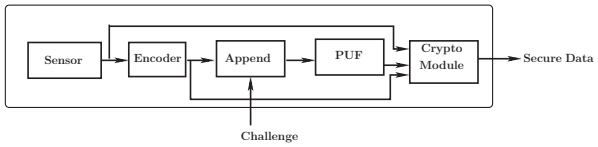

Fig. 1. Architectural overview of Prover's setting.

Then, these intervals are encoded into n-bit binary strings (cf. Encoder in Figure 1). Let us assume that the prover uses a PUF with $(m+n)$-bit challenge. In our protocol, for a given temperature interval T, the challenges are generated by combining an m-bit external challenge (sent by verifier) and an n-bit encoded value of T. Note that the definition of combining function relies on the Strict Avalanche Criterion (SAC) of the used PUF instance. In case of the DAPUF, we use Append (cf. Figure 1) function which appends encoded T to the external challenge to generate the challenge for the PUF instance.

Note that the communication between verifier and prover are encrypted (using Crypto Module) to prevent the adversary from eavesdroping the CRPs. Thus, our system is secure against man-in-the-middle attacks, and to prevent replay attack, verifier does not reuse the CRPs for multiple authentications. The following question may arise in the readers' mind: *if the cryptographic module is used, what is the need of virtual proof?* The reason is that some times the sensor might be damaged by environmental factors or modified by the adversary, and always produces wrong data. Using our protocol (to be discussed in section III-B), we can detect this kind of scenarios, but this is not possible by using only encrypted communication.

B. Authenticated Sensing Protocol

Figure 2 summarizes the steps of the proposed authenticated sensing protocol. The protocol consists of two steps: *Setup* and *Virtual Proof*. Now, we explian these steps in details.

Let us assume that the range of temperature in which the system works is from $-20°C$ to $80°C$. First, the range of temperature is partitioned into disjoint intervals as $T_1 = \{t_{-20}, t_{-19}, \ldots, t_a\}, T_2 = \{t_{a+1}, t_{a+2}, ..., t_b\}, \ldots, T_K = \{t_x, t_{x+1}, \ldots, t_{79}, t_{80}\}$. The number of intervals and the size of each interval rely on the PUF challenge-response behavior in a certain range of temperature. In our protocol, each of these intervals is encoded in n-bit binary value, for example: $T_1 : 0\ldots000$, $T_2 : 0\ldots001, \ldots, T_K : 1\ldots111$. This encoding along with specification of each interval would be stored in both the verifier and prover ends, and this information can also be kept in public.

1) Setup: For each temperature interval $T_i, i \in [1, K]$, the verifier randomly selects l bit-string of size m-bit, and to generate $(m+n)$-bit challenges for the PUF, these bit-strings are appended with the n-bit encoded value of T_i. Next, the verifier collects the responses to these challenges by evaluating the PUF instance for temperature interval T_i. So, the verifier maintains a CRP database $L = \{(C_j, R_j^i, T_i)\}_{i \in [1,K], j \in [1,l]}$. The space complexity of the CRP database L is $\mathcal{O}(K*l*(m+$

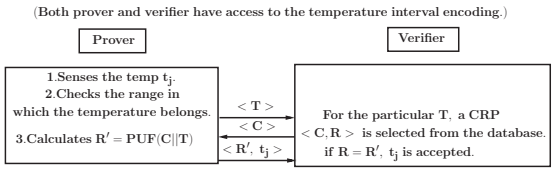

Fig. 2. The proposed trustworthy virtual proof protocol. All $< message >$ are sent to the insecure communication network after encryption.

$n + r)$), where r is the length of each response in bits. Now, the prover device can be deployed in an insecure network.

2) Virtual Proof: During the virtual proof, the following steps are executed in the verifier and prover sides:

1) When the prover node wants to send some sensed temperature data t_j to the verifier, it first finds out the interval in which the temperature falls, and send encoded n-bit value of T to the verifier.
2) The verifier then randomly chooses a challenge C of m-bit from the database for that particular temperature interval, and sends the challenge to the prover.
3) The prover appends the encoded value of T to C, and applies $(m + n)$-bit challenge to the PUF to get the response R'. The prover sends the actual sensed data t_j and R' to the verifier after encryption.
4) The verifier checks for a match between the stored response R and received response R'. If the match happens, then it accepts the sensed data. Otherwise, it rejects it.

The major advantages of the proposed protocol are as follows. The verifier can use the same set of m-bit external challenges for different temperature intervals as the appended n-bit encoded temperature values ensures the unique responses for each interval. Other advantage is that the proposed VP of temperature does not solely depend on the unreliability of the PUF, it also depends on the sensor data which ensures the uniqueness of responses for a given temperature interval.

IV. Trustworthiness of the VP Against Forgery

Now, we discuss the adversary model and security analysis of the proposed protocol.

A. Adversarial Model

We assume the adversary can have access to the communication channel and can tamper the channel with malicious data. But she can not control the environmental changes in the system. The purpose of the adversary is to authenticate a temperature value to the verifier on behalf of the legitimate nodes, without possession of the PUF instance so that it can set an undesirable effect on the system.

B. Security Analysis

In our analysis, we discuss the security of the authentication proof of sensor data w.r.t. four types of forgery attacks: *Existential Forgery*, *Selective forgery*, *Universal Forgery* and *Total Break*.

1) Existential forgery: Let us assume that the adversary has observed n data/proof pairs. Existential forgery is the creation of at least one data/proof pair (t_i, R'_i) where R'_i was not produced by the legitimate prover before, i.e. $i \notin [1, n]$. The VP is said to be trustworthy against this type of attacks if the responses of the PUF instances are: (a) *unique* to the temperatual interval, (b) *uniform*, i.e., the distributions of 0's and 1's are uniform, and (c) *uncorrelated*, i.e., there should be no correlation among the response bits of the PUF instance. Otherwise, the adversary can create a VP with non-negligible probability of success just by majority voting over the response set.

2) Selective forgery: Selective forgery is the creation of a data/proof pair where the data and the challenge have been chosen by the attacker prior to the attack. The virtual proof is said to be trustworthy against this type of attacks if the responses of the PUF instances satisfy *Strict Avalanche Criterion* (SAC) property [11]. An x-bit input, 1-bit output Boolean function $f : \{0,1\}^x \longrightarrow \{0,1\}$ is said to satisfy SAC property, if the output of the function f complements with probability of one-half, whenever a single bit of the input is complemented. If a PUF design does not satisfy the SAC property for all challenge bit positions, the adversary can cleverly derive the challenge-data pairs from the available information with a large probability of success. Thus, a PUF design with poor SAC property cannot ensure the security against the selective forgery.

3) Universal Forgery and Total Break: Universal forgery is the creation of a valid proof for any given data and challenge. An adversary capable of universal forgery is able to prove a sensor data she chooses herself (as in selective forgery) or data chosen at random. This might happen if the adversary can build a mathematical model of the PUF instance. On the other hand, a proof can be totally broken if the PUF instance is physically cloned.

As the current state-of-the-art, we expect DAPUF to be a good choice for these properties on FPGAs, and in next section, we show that the experimental results also matches with the requirements.

V. Experimental Results for Trustworthy VP

We have designed a 5-4 DAPUF (cf. Fig. 3) using Xilinx ISE (`v 14.2`) CAD tool and implemented it on Xilinx Artix-7 FPGAs. The number of registers (FF), look-up tables (LUTs) and occupied Slices for this PUF design are 283, 891 and 451, respectively. The power consumption reported by Xilinx *XPower Analyzer* is 0.044W. We have evaluated the 5-4 DAPUF instances using 10×10^3 challenges on eight FPGAs with temperature variation from $-20°$C to $80°$C. To generate the reference response to a challenge, majority voting is used over 15 measurements of the same challenge. CME Nano-Bench Top Chamber is used to vary the operating temperature of FPGA. The reliability of 5-4 DAPUF across the temperature variations lies in the range $[79, 88]\%$. The uniqueness of individual output bits of 5-4 DAPUF is approximately 44.16%. The uniformity for the four output bits of 5-4 DAPUF are

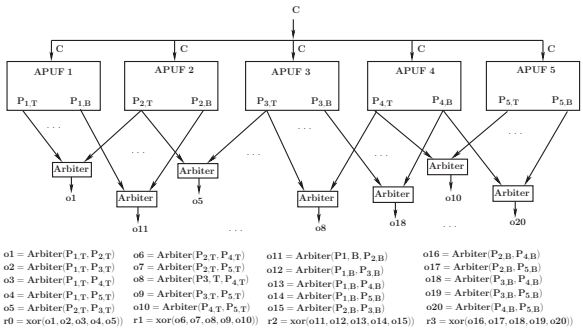

Fig. 3. Architectural overview of 5-4 DAPUF. It genreates 4-bit output, each of them depends on the outputs of five consecutive arbiters.

TABLE I
CORRELATION AMONG THE OUTPUT BITS OF A 5-4 DAPUF

correlation	1-2	1-3	1-4	2-3	2-4	3-4
Board 1	0.47	0.53	0.50	0.47	0.46	0.53
Board 2	0.52	0.53	0.57	0.45	0.51	0.56
Board 3	0.47	0.56	0.49	0.48	0.53	0.52

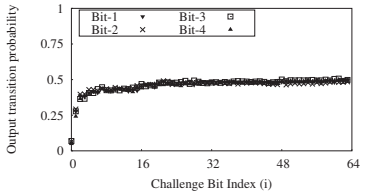

Fig. 4. SAC property of individual output bits of a 5-4 DAPUF.

44.6%, 54.9%, 43.4% and 40.9% respectively. The correlation values between the output bits of 5-4 DAPUF are reported in Table I, where ideal value of correlation is 0.5. The result implies that there is a significantly less correlation between output bits.

Since the uniformity of output bits of 5-4 DAPUF are not 0.5, the adversary can take the advantage of this bias. Ideally, the guessing probability of 4-bit response of 5-4 DAPUF to a given challenge is $(\frac{1}{2})^4$=0.0625, whereas the this probability is $(0.45 \times 0.55 \times 0.43 \times 0.41) = 0.04363$ for the present implementation of 5-4 DAPUF instances. The result implies that the proposed VP has sufficient robustness against the *existential forgery*.

In addition, we have computed the SAC property for each output bit of 5-4 DAPUF following the technique mentioned in [11], [12]. Figure 4 shows the output transition probability of each output bits of 5-4 DAPUF, when there is a flip in i-th bit of challenge. As the output transition probability of 5-4 DAPUF is very small for lower index bits of the challenge, we have appended the encoded value of a temperature interval at the end of m-bit external challenge. So even if the same m-bit external challenges are used for two different temperature intervals, where Hamming distance of the corresponding encoded values are very low, the probability that adversary predicts the response to a challenge in one interval having access to the response of the same challenge in another interval is approximately 0.5. This proves the unforgeability of the VP against *selective forgery.* Like APUF, DAPUF is also vulnerable to modeling attack [12], but the adversary needs more CRPs for DAPUF compared to an APUF modeling. We have applied the logistic regression based modeling technique of XOR APUF to build a model for each output bit of 5-4 DAPUF. The modeling accuracy of each output bit is approximately $76-79\%$ even using 2×10^5 CRPs (data complexity). Thus, the prediction accuracy of entire 4-bit response to a given challenge can be approximately $(0.79)^4 \times 100 \approx 39\%$ which is significatly less in the context of modeling attack. To achieve better modeling accuracy, the adversary needs to have access to more CRPs and computational power. Thus, the proposed VP is computationally unforgeable against mathematical cloning as well as *universal forgery*. Finally, there is no reported result on the physical cloning of DAPUF in the literature. Hence, the protocol is secure against *total break.*

VI. CONCLUSION

We have developed a PUF-based secure authenticated sensing protocol for virtual proof of the sensor data, where we have used the sensor data as a part of the input to PUF and a cryptographic module for secure communication of sensed data. We have validated the protocol using DAPUF on FPGA for virtual proof of the temperature. As a future work, we shall investigate PUF designs which are suitable for the virtual proofs of other ambient factors using the such protocols.

VII. ACKNOWLEDGEMENT

This work was supported in part by the SGDRI research grant from IIT Kharagpur, India; Information Security Education Awareness (ISEA), DeitY, India. Debdeep Mukhopadhyay would like to thank DST Swarnajayanti Fellowship for partial support.

REFERENCES

[1] D. LIM, "Extracting Secret keys from Integrated Circuits," USA, 2004.
[2] K. Rosenfeld, E. Gavas, and R. Karri, "Sensor physical unclonable functions," in *IEEE HOST*, 2010, pp. 112–117.
[3] Ulrich Rührmair and J. L. Martinez-Hurtado and Xiaolin Xu and Christian Kraeh and Christian Hilgers and Dima Kononchuk and Jonathan J. Finley and Wayne P. Burleson, "Virtual Proofs of Reality and their Physical Implementation," in *IEEE S&P*, 2015, pp. 70–85.
[4] H. Bojinov, Y. Michalevsky, G. Nakibly, and D. Boneh, "Mobile device identification via sensor fingerprinting," *CoRR*, vol. abs/1408.1416, 2014.
[5] K. C. Baby, S. Aung, and N. Schwesinger, "Finite element analysis of differential capacitive PUF sensors," in *2016 IEEE SAS*, 2016, pp. 1–6.
[6] K. Fukushima, S. Hidano, and S. Kiyomoto, "Sensor-based wearable PUF," in *ICETE 2016*, 2016, pp. 207–214.
[7] Y. Zheng, Y. Cao, and C. Chang, "A new event-driven dynamic vision sensor based physical unclonable function for camera authentication in reactive monitoring system," in *2016 IEEE AsianHOST*, 2016, pp. 1–6.
[8] Y. Cao, L. Zhang, and C. Chang, "Using image sensor PUF as root of trust for birthmarking of perceptual image hash," in *2016 IEEE AsianHOST*, 2016, pp. 1–6.
[9] H. Ma, Y. Gao, O. Kavehei, and D. C. Ranasinghe, "A PUF sensor: Securing physical measurements," in *IEEE PerCom*, 2017, pp. 648–653.
[10] T. Machida, D. Yamamoto, M. Iwamoto, and K. Sakiyama, "A new mode of operation for arbiter PUF to improve uniqueness on FPGA," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, 2014, pp. 871–878.
[11] P. H. Nguyen, D. P. Sahoo, R. S. Chakraborty, and D. Mukhopadhyay, "Security analysis of arbiter PUF and its lightweight compositions under predictability test," *ACM TODAES*, vol. 22, no. 2, pp. 20:1–20:28, 2017.
[12] D. P. Sahoo, P. H. Nguyen, R. S. Chakraborty, and D. Mukhopadhyay, "Architectural bias: a novel statistical metric to evaluate arbiter PUF variants," *IACR Cryptology ePrint Archive*, vol. 2016, p. 57, 2016.