iTPM: Exploring PUF-based Keyless TPM for Security-by-Design of Smart Electronics

Venkata K. V. V. Bathalapalli
Dept. of Computer Science and Engineering
University of North Texas.
Email: vb0194@unt.edu

Saraju P. Mohanty
Dept. of Computer Science and Engineering
University of North Texas.
Email: saraju.mohanty@unt.edu

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

Vasanth Iyer
Computer Science and Digital Technologies
Grambling State University.
Email: iyerv@gram.edu

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

***Abstract*—The scope of Smart electronics and its increasing market worldwide has made cybersecurity an important challenge. The Security-by-Design (SbD) principle, an emerging cybersecurity area, focuses on building security/privacy-enabled primitives at the design stage of an electronic system. This paper proposes a novel Physical Unclonable Function (PUF) based Trusted Platform Module (TPM) for SbD primitive. The proposed SbD primitive works by performing secure verification of the PUF key using TPM's Encryption and Decryption engine. The securely verified PUF Key is then bound to TPM using Platform Configuration Registers (PCR). PCRs in TPM facilitate a secure boot process and effective access control to TPM's Non-Volatile memory through an enhanced authorization policy. By binding PUF with PCR in TPM, a novel PUF-based access control policy can be defined, bringing in a new security ecosystem for the emerging Internet-of-Everything era. The proposed SbD approach has been experimentally validated by successfully integrating various PUF topologies with Hardware TPM.**

***Index Terms*—Security-by-Design (SbD), Trsuted Platform Module (TPM), Physical Unclonable Function (PUF), Energy-Efficient Cybersecurity, Hardware-Assisted Security, Cyber-Physical System (CPS) Internet-of-Things (IoT)**

I. Introduction

Cybersecurity is a critical challenge for connected smart electronics which are omnipresent in modern society. As a mitigation of the cybersecurity problem both hardware and software based solutions have been explored. Software-based cybersecurity solutions are more vulnerable as compared to hardware-based security primitives [1]. Resource-constrained IoT devices cannot support the computational resource requirements of software security primitives which further reduces the energy efficiency of Internet-of-Things (IoT) devices [2]–[4]. Hardware-based security solutions using TPM and PUF have proven to be much more effective as these primitives are found to be effectively integrating with IoT-based applications in Healthcare, Agriculture, and Transportation [3]–[5].

Security-by-Design (SbD) is an emerging principle that works on building security and privacy-enabled primitives at the design stage of a smart electronic system. IoT devices are deployed for realizing various smart applications [1]. The data from these heterogeneous sensors are processed in various computing paradigms like Fog, Edge, and Cloud. Edge, and Fog computing paradigms process data near the source end and are decentralized computing paradigms. Cloud is a centralized computing paradigm requiring more time for decision-making and analysis [3], [6].

Physical Unclobale Function (PUF) primitive-based security solutions have gained much prominence due to their robust nature of generating a secure cryptographic key based on process variations inside an IC [7]. PUF module, when embedded with IoT devices can generate a unique cryptographic key for that device and can guarantee integrity and authenticity [8]. TPM is a secure crypto processor that is now available in all commercial desktops, laptops, and computing systems [9]–[12]. Cryptographic key storage, RSA and AES-based encryption, decryption, and validating the integrity of a remote smart electronic system are prominent functionalities of TPM. TPMs can also facilitate secure boot for a computing platform by making use of PCR which stores device integrity credentials [13]–[15]. Fig. 1 shows the conceptual working overview of proposed iTPM which integrates PUF and TPM primitives.

The rest of the paper is organized in the following manner: Section II presents the novel contributions of the current paper. Section III presents the SbD primitive from the literature on TPM and PUF. Section IV presents a comprehensive overview of SbD and its principles. Section V presents the information and protocol overview of integrating PUF with TPM. Section VI outlines the implementation details and Section VII presents the conclusions and directions for future research.

II. Novel Contributions

To facilitate cybersecurity of Smart electronics which are IoT end-device as well as IoT edge-device, we proposed a novel SbD approach with following contributions:

- A sustainable PUF-based TPM SbD approach that works by defining a PUF-based access control policy for TPM.

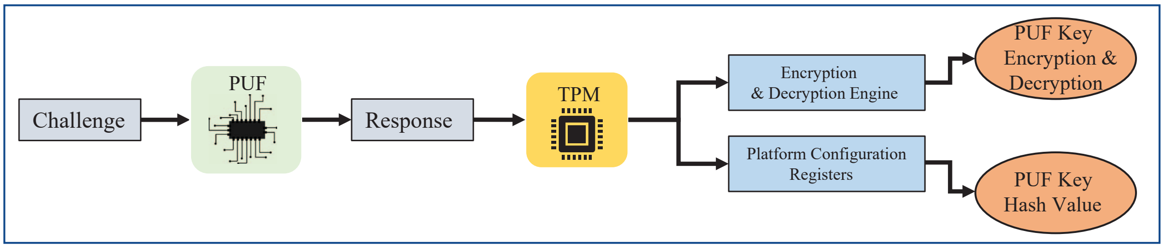

Challenge
PUF
Response
TPM
Encryption & Decryption Engine
Platform Configuration Registers
PUF Key Encryption & Decryption
PUF Key Hash Value

Fig. 1: Conceptual overview of Proposed iTPM.

- A simple, lightweight, and robust approach for integrating two hardware security primitives PUF, and TPM to achieve the objective of sustainable and secure IoT.
- A robust TPM-based PUF key verification scheme that utilizes TPM's encryption and decryption policy.
- A sustainable policy that can bind PUF with TPM's platform Configuration Registers (PCR)
- A simple Edge computing drive PUF-based keyless TPM initiative that works by binding PUF with PCR that can facilitate a secure boot process, remote attestation, and NVRAM access control in TPM.
- A novel approach that explores the true potential of proposed SbD by integrating various PUF topologies with Hardware TPM.

III. Related Prior Research

A comparative analysis of our SbD primitive with state-of-the-art research is presented in Table I. The authors in [4] proposed a sustainable TPM-based remote attestation mechanism for the security of Internet-of-Medical-Things (IoMT) where they presented a simple Client-Server model which works by enabling a hardware TPM on the server side to attest remotely an attestation Client with a shadow TPM. However, this approach does not propose a sustainable hardware security module for the Client but works on extending the TPM functionalities to Clients remotely using a Trusted Agent. A simple software-based remote attestation scheme for IoT device security is proposed in [16]. In this work trustworthiness of a client is determined by the server by comparing the hashed message authentication code values of the Client with known good values thereby performing software-based integrity verification. This work, however, does not include a hardware-assisted security solution.

A lightweight device integrity verification framework is proposed in [14]. In this work, a client-server model is adopted where Remote Attestation Server will verify the integrity of a client remotely with the objective of malware detection, and device integrity verification. In [7], [17] a novel concept of Eternal-Thing has been presented that demonstrates the tight integration of cybersecurity in IoT-end devices.

A simple distributed IoT attestation framework using Blockchain technology is presented in [13] where the IoT clients can upload device integrity credentials onto Blockchain and the attestation server performs remote attestation by extracting these values from Blockchain. This works claims to improve scalability by adopting Distributed Attestation Framework (DAN). This approach however lacks proper implementation details and information about how Block validation and device authentication is performed prior to remote attestation. A novel secure protocol for trusted cloud computing using TPM is proposed in [18]. This work proposes an enclave TPM which is a software framework for a secure cloud developed using Intel SGX technology, in comparison with the above works, our proposed iTPM is based on Hardware based TPM for a secure Edge Cloud computing environment with PUF integrated approach.

IV. Security-by-Design (SbD) Principle for Electronics Systems

Security-by-Design is an emerging phrase that emphasizes on enabling security as an inherent feature at the design level of an electronic rather than at the application level [1], [21]. The 7 principles of the Security-by-Design (SbD) approach include the following [1]:

1) *Security features should be Proactive not Reactive*: Security solutions for SbD approach should be done in a proactive fashion in anticipation that cyberscrurity issues will arise for the smart electronics instead of exploring solutions after cyberscrurity crisis takes place.
2) *Security should be Default*: Cybersecurity features of the smart electronics should be default option in the context of hardware, software, and system specifications.
3) *Security should be Embedded into Design*: The cybsecurity solutions of the smart electronics should be integrated in the design and should be builtin as if the solutions cann't be separated from the system.
4) *Security should be incorporated as a Full Functionality - PositiveSum, not Zero-Sum without trade-offs*: To facilitate effective integration with smart electronics, the SbD approach should have not tradeoffs and shouldn't have energy, battery, and performance overheads.
5) *Security-Solutions should be End-to-End Security for Lifecycle Protection*: The cybersecurity solutions should provide security in the entire life-cycle of the smart electronics, from design to deployment.
6) *Security-Solutions should have Visibility and Transparency*: The SbD approach in an Electronic system should be easily understandable and information should be visible and clear.

TABLE I: Comparative Analysis of SbD primitives from literature

Research Works	Applications	Security Mechanism	Features	Approach
eTPM [18]	Cloud Computing	Software TPM	Virtual Machine (VM) Security	Cloud Computing
RADIS [19]	IoT	NA	Distributed Service Attestation	Distributed
xTSeH [4]	IoMT (Device)	Hardware TPM	TPM based Remote Attestation	Decentralized
TPM based Sensor Security [20]	Wireless Sensor Networks (WSN)	Hardware TPM	Secure WSN	NA
IoT Remote Attestation [14]	IoT	Raspberry pi based TPM, Blockchain	Malware Detection	NA
PUFchain 4.0 [12]	IoT (Device & Data)	PUF, Hardware TPM, Tangle	Sealing PUF Key inside TPM (NVRAM)	Edge Computing
iTPM (This Work)	Smart Electronics	PUF, Hardware TPM	Securely Binding PUF with PCR	Edge Computing

7) *Security-Solutions should have Respect for Users*: The cybsecurity solutions should respect the users in terms of their safety, privacy, and convenience.

V. iTPM: PUF-based Keyless TPM

This section presents a holistic view of TPM technology and its applications for IoT security. An overview of the proposed security scheme based on integrating PUF with TPM is presented along with a brief overview of the protocol in Algorithmic and flowchart description.

PCRs in TPM store the hashes and checksum values of system configuration parameters. During the boot process in a computing platform, the BIOS and firmware check different system configuration parameters and compute the hashes of these parameters and store them inside TPM's PCR [22]. Fig. 2 presents the architecture of proposed PUF based TPM.

A. Working of Proposed iTPM

Initially, a group of PUF Keys R_n are generated by accessing the PUF module and testing it with a group of challenges C_n. These responses are evaluated to obtain PUF figures of merits. Once the keys are evaluated, a particular response R_x is selected and broadcasted to the server. As soon as the server with embedded TPM receives the key, it accesses the TPM's Encryption and Decryption engine and securely encrypts the key as R_{out} inside TPM. Fig. 3 explain the working of the TPM-based PUF Key validation process.

As shown in Algorithm 1, during verification, the server performs PUF key verification by comparing newly received PUF key R'_x for the same challenges input from the Edge node. Edge Server decrypts encrypted PUF key R_{out} inside TPM and compare it with R'_x.

Once PUF validation is done, the server access PCR inside TPM which binds the PUF key to device integrity parameters. PCR registers from 16-23 can be used to extend the PUF keys to TPM and can also be resettable. The process of binding PUF with PCR in iTPM is explained in Algorithm 2.

B. Physical Unclonable Function Topologies

Arbiter PUF is a simple and strong PUF design that works by comparing the delay variations associated with an IC. By creating two symmetric paths which are built with identical

Algorithm 1 Performing TPM Enabled PUF Validation

```
Input: PUF Challenge Input
Output: Decrypted PUF Key from TPM
Access PUF Module and extract PUF Keys
    Challenge $C_n \rightarrow$ PUF Module $\rightarrow$ Response $R_n$
Test PUF against metrics
    PUF Keys Rn $\rightarrow$ Reliability, Uniqueness, randomness.
if (PUF Keys are standard) then
    Select one of the keys $R_x$
end if
Broadcast Key to Edge Server (ES)
    $R_x \rightarrow$ Edge Server
Server access TPM
    ES $\rightarrow$ TPM
Create Primary Key in TPM to sign
    Command$\rightarrow$ Sudo tpm2_createprimary -C primary.ctx
Sign the PUF key in encrypted form
    TPM $\rightarrow$ Rx$\rightarrow$ $R_o$ut (Encrypted Form),
    Command$\rightarrow$ Sudo tpm2_rsaencrypt -c key.ctx Arbiter.enc Arbiter.dat
During authentication, ES decrypts the PUF key from TPM
    Command$\rightarrow$ Sudo tpm2_rsadecrypt -c key.ctx Arbiter.text Arbiter.enc,
    TPM Decryption $\rightarrow R_{out}$
ES recives newly extracted PUF Key from edge node
    ES $\rightarrow R'_x$
if ($Rx' = Rout$) then
    Successful Decryption of PUF key from TPM
end if
```

Algorithm 2 Binding PUF Key with PCR

```
Input: PUF Keys
Output: Hash Values
Initialize TPM
Access PUF Key Rx
    PUF Module $\rightarrow$ Rx
Access PCR
    Command $\rightarrow$ Sudo tpm2_pcrread
Obtain PCR Registers 16 & 23
Bind Arbiter PUF and XOR PUF Keys to register 16 and 23
    Command $\rightarrow$ Sudo tpm2_pcrevent 16 Arbiter.txt
    Command $\rightarrow$ Sudo tpm2_pcrevent 23 XOR.txt
Extend PCR values by calculating hash values Sha-1, and Sha-256
    Command $\rightarrow$ Sudo tpm2_pcrextend..
Read PCR values after the extension
    Command $\rightarrow$ Sudo tpm2_pcrread
```

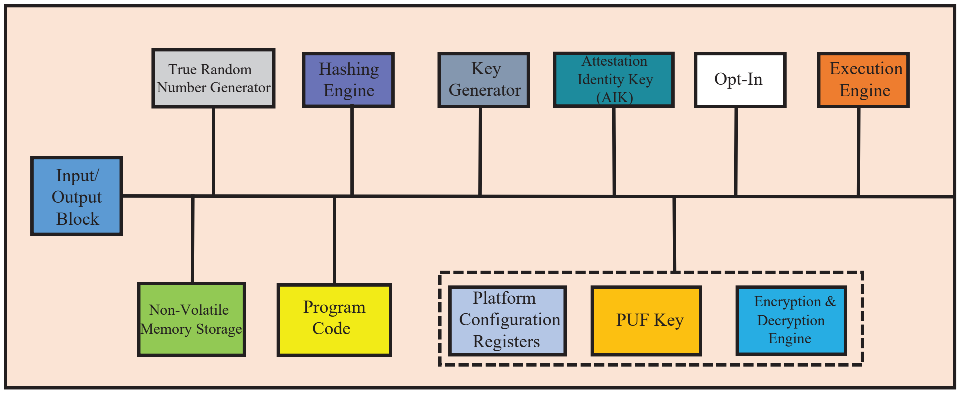

True Random Number Generator
Hashing Engine
Key Generator
Attestation Identity Key (AIK)
Opt-In
Execution Engine
Input/ Output Block
Non-Volatile Memory Storage
Program Code
Platform Configuration Registers
PUF Key
Encryption & Decryption Engine

Fig. 2: PUF-based TPM architecture.

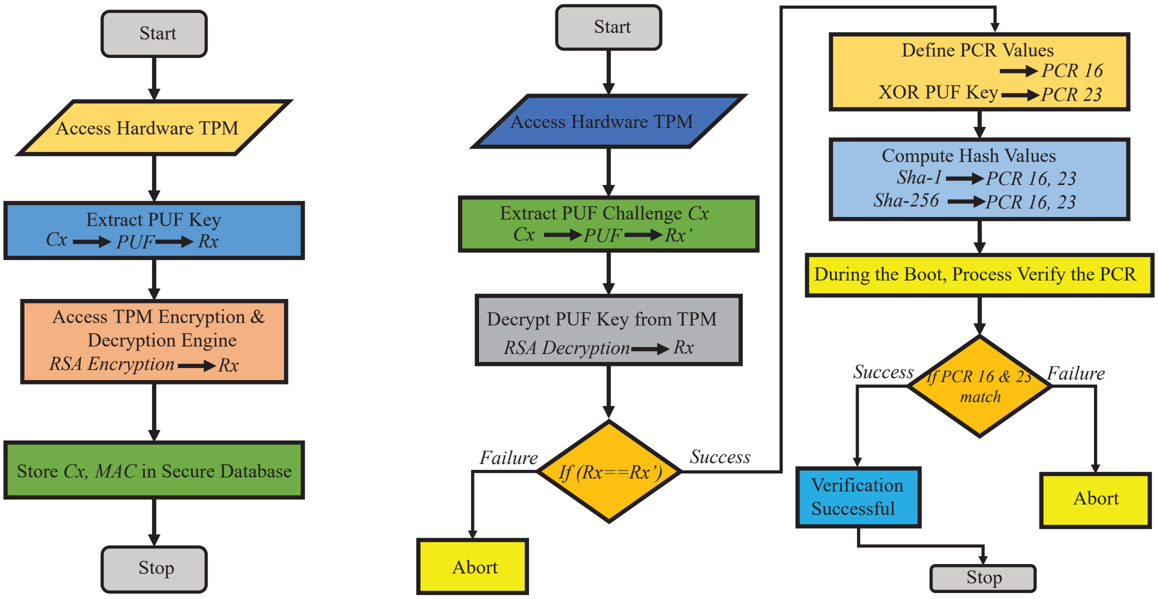


Fig. 3: Working Flow of Proposed iTPM.

digital logic elements like Multiplexer and NOT gates, a simple response output ”1 or 0” is finally obtained from D Flip flop. The challenge input to the PUF module will be a select line of the multiplexer and the response output will be the output from D Flip Flop. Fig.4 presents the topology of an Arbiter PUF [3], [23], [24].

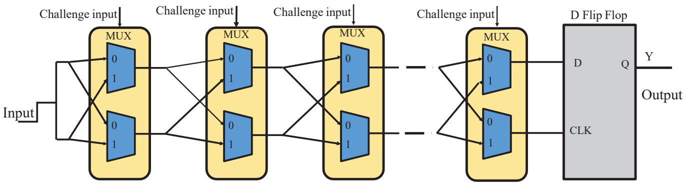


Fig. 4: Arbiter PUF Topology.

XOR PUF is another delay-based PUF design proposed to improve the resiliency of Arbiter PUF by XOR the outputs from D Flip Flops of all the Arbiter PUF instances. Performing the XOR operation on the outputs from each instance can improve the randomness of final D Flip Flop outputs [25]. The design of an XOR PUF is shown in Fig. 5.

VI. Experimental Results

Edge nodes and Edge servers have been deployed on Raspberry pi 4 2GB boards and Geek Pi TPM 2.0 module based on Infineon SLB 9670 is connected to Edge server using SPI Interface. Arbiter PUF and XOR PUF modules have been used to validate the proposed scheme. The PUF modules have been deployed on Xilinx Artix-7 Basys-3 FPGA boards using Vivado 2020.2 in Verilog Hardware description language. Universal Asynchronous Receiver Transmitter (UART) serial communication protocol is used for PUF key extraction with a baud rate of 9600. The PUF Keys from the two PUF modules have been evaluated and the Figure of merits is calculated and shown in Fig. 6.

A Geek Pi TPM 2.0 module with an embedded Infineon Chip is used as Hardware TPM for validation. It is connected to a Raspberry pi 4 board using Serial Peripheral Interface

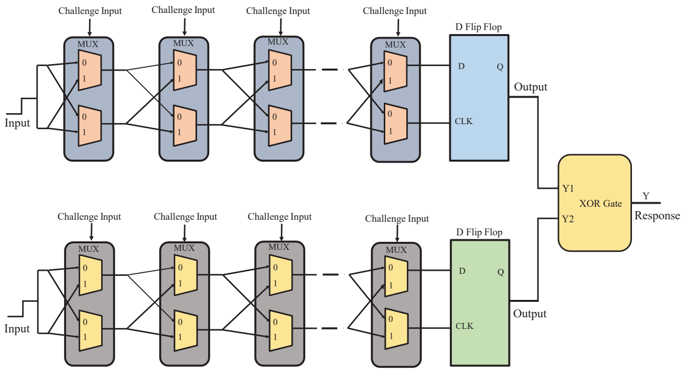
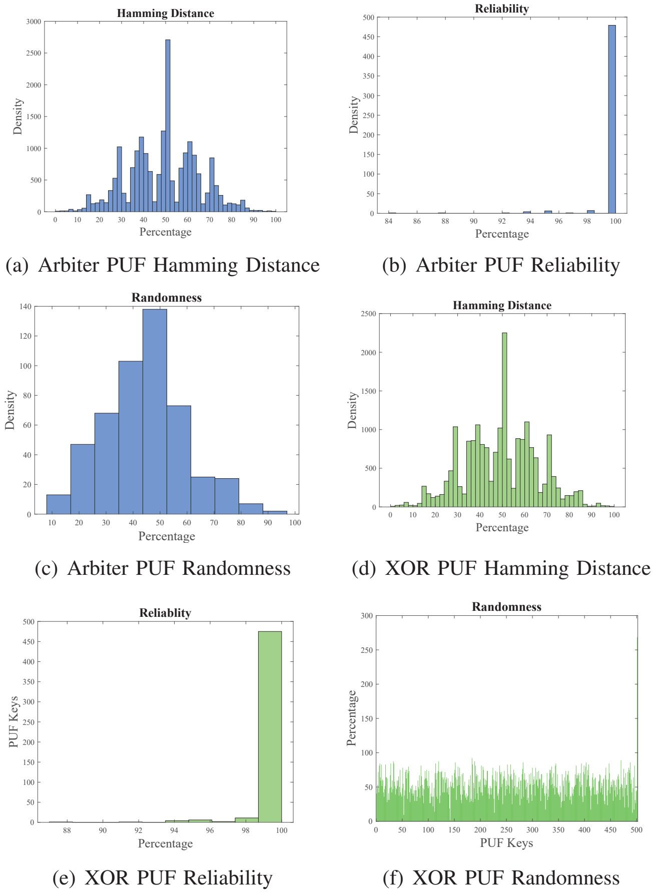


Fig. 5: XOR Arbiter PUF Topology.

(SPI). PUF modules on FPGA are connected to Pi's using Pmod ports as shown in Fig. 7.

Various 64-bit XOR Arbiter and Arbiter PUF Keys are extracted from Edge Nodes and through User Datagram Protocol (UDP), the Keys are broadcasted to the Server. TPM's RSA encryption and decryption engine securely stores the key in the encrypted form inside TPM. During verification, the PUF key in encrypted form are decrypted from TPM and compared. Table II represent the characterization of proposed iTPM.



(a) Arbiter PUF Hamming Distance

(b) Arbiter PUF Reliability

(c) Arbiter PUF Randomness

(d) XOR PUF Hamming Distance

(e) XOR PUF Reliability

(f) XOR PUF Randomness

Fig. 6: PUF Metrics.

In this work, PCR register 16 and 23 have been used. Arbiter PUF is extended to PCR 16 and XOR PUF Key is extended to PCR 23. The obtained PCR values and corresponding PUF key validation results inside TPM are shown in Fig. 8.

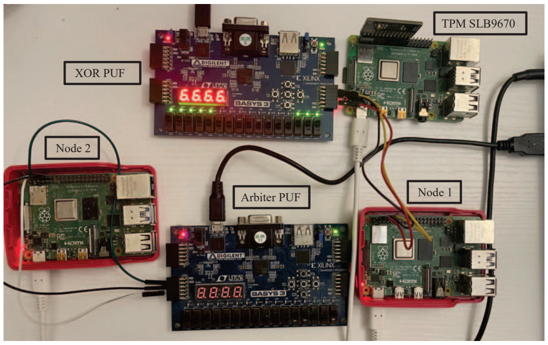


Fig. 7: Hardware Prototyping of the Proposed iTPM.

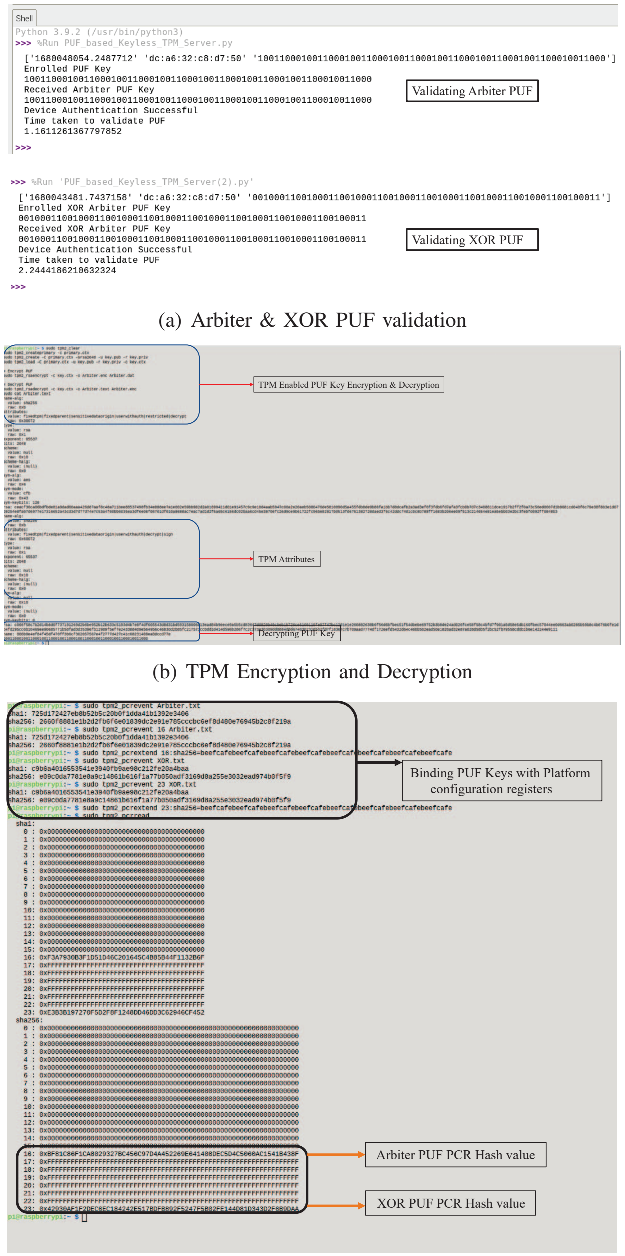
TABLE II: Evaluation of iTPM

Metrics	Results
Application	Smart Electronics
Hardware Security Mechanism	PUF-based Keyless TPM
Security Modules	TPM, PUF
PUF Modules	XOR Arbiter & Arbiter PUF
Approach	Integrating PUF with TPM's PCR
Platform Configuration Registers	16 & 23
TPM Integration	Encryption & Decryption Engine, and PCR
TPM	Hardware TPM
Hardware TPM Chip	Infineon SLB 9670
PUF	Xc7a35tcpg236-1
TPM Embedded Device	Single Board Computer
Interface	SPI, UART
Tools	tpm2-tools, tpm2-abrmd, VIVADO 2020.2
TPM Hash Algorithm	Sha-1 and Sha 256
Possible Applications	Remote Attestation, and Secure Boot Process

VII. Conclusion and Future Research

This paper presented and validated a novel SbD approach with a sustainable policy for integrating PUF and TPM by binding PUF with PCRs inside TPM. By successfully binding PUF in PCR, the PUF is made as a device integrity credential required for a secure boot process.

The proposed iTPM initiative has been experimentally substantiated and results have been presented. Further, the experimental analysis revealed that integrating PUF inside PCR could bind PUF with TPM and facilitate security at the edge level in smart Electronics applications. The proposed approach also presents the possibility of PUF-enabled secure firmware and boot process for computing systems. Extending above security protocol to various areas of Smart Electronics and improving the energy efficiency of IoT by adopting low overhead PUF-based solutions can be a part of future research.


Shell
Validating Arbiter PUF
Validating XOR PUF
TPM Enabled PUF Key Encryption & Decryption
TPM Attributes
Decrypting PUF Key
Binding PUF Keys with Platform configuration registers
Arbiter PUF PCR Hash value
XOR PUF PCR Hash value

```
Python 3.9.2 (/usr/bin/python3)
>>> %Run PUF_based_Keyless_TPM_Server.py

 ['1680048054.2487712' 'dc:a6:32:c8:d7:50' '1001100010011000100110001001100010011000100110001001100010011000']
 Enrolled PUF Key
 1001100010011000100110001001100010011000100110001001100010011000
 Received Arbiter PUF Key
 1001100010011000100110001001100010011000100110001001100010011000
 Device Authentication Successful
 Time taken to validate PUF
 1.1611261367797852

>>>


>>> %Run 'PUF_based_Keyless_TPM_Server(2).py'

 ['1680043481.7437158' 'dc:a6:32:c8:d7:50' '0010001100100011001000110010001100100011001000110010001100100011']
 Enrolled XOR Arbiter PUF Key
 0010001100100011001000110010001100100011001000110010001100100011
 Received XOR Arbiter PUF Key
 0010001100100011001000110010001100100011001000110010001100100011
 Device Authentication Successful
 Time taken to validate PUF
 2.2444186210632324

>>>
```

(a) Arbiter & XOR PUF validation

```
pi@raspberrypi:~ $ sudo tpm2_clear
sudo tpm2_createprimary -c primary.ctx
sudo tpm2_create -C primary.ctx -Grsa2048 -u key.pub -r key.priv
sudo tpm2_load -C primary.ctx -u key.pub -r key.priv -c key.ctx

# Encrypt PUF
sudo tpm2_rsaencrypt -c key.ctx -o Arbiter.enc Arbiter.dat

# Decrypt PUF
sudo tpm2_rsadecrypt -c key.ctx -o Arbiter.text Arbiter.enc
sudo cat Arbiter.text
name-alg:
  value: sha256
  raw: 0xb
attributes:
  value: fixedtpm|fixedparent|sensitivedataorigin|userwithauth|restricted|decrypt
  raw: 0x30072
type:
  value: rsa
  raw: 0x1
exponent: 65537
bits: 2048
scheme:
  value: null
  raw: 0x10
scheme-halg:
  value: (null)
  raw: 0x0
sym-alg:
  value: aes
  raw: 0x6
sym-mode:
  value: cfb
  raw: 0x43
sym-keybits: 128
rsa: [illegible]
[illegible]
name-alg:
  value: sha256
  raw: 0xb
attributes:
  value: fixedtpm|fixedparent|sensitivedataorigin|userwithauth|decrypt|sign
  raw: 0x60072
type:
  value: rsa
  raw: 0x1
exponent: 65537
bits: 2048
scheme:
  value: null
  raw: 0x10
scheme-halg:
  value: (null)
  raw: 0x0
sym-alg:
  value: null
  raw: 0x10
sym-mode:
  value: (null)
  raw: 0x0
sym-keybits: 0
rsa: [illegible]
[illegible]
name: [illegible]
1001100010011000100110001001100010011000100110001001100010011000
pi@raspberrypi:~ $ 
```

(b) TPM Encryption and Decryption

```
pi@raspberrypi:~ $ sudo tpm2_pcrevent Arbiter.txt
sha1: 725d172427eb8b52b5c20b0f1dda41b1392e3406
sha256: 2660f8881e1b2d2fb6f6e01839dc2e91e785cccbc6ef8d480e76945b2c8f219a
pi@raspberrypi:~ $ sudo tpm2_pcrevent 16 Arbiter.txt
sha1: 725d172427eb8b52b5c20b0f1dda41b1392e3406
sha256: 2660f8881e1b2d2fb6f6e01839dc2e91e785cccbc6ef8d480e76945b2c8f219a
pi@raspberrypi:~ $ sudo tpm2_pcrextend 16:sha256=beefcafebeefcafebeefcafebeefcafebeefcafebeefcafebeefcafebeefcafe
pi@raspberrypi:~ $ sudo tpm2_pcrevent XOR.txt
sha1: c9b6a4016553541e3940fb9ae98c212fe20a4baa
sha256: e09c0da7781e8a9c14861b616f1a77b050adf3169d8a255e3032ead974b0f5f9
pi@raspberrypi:~ $ sudo tpm2_pcrevent 23 XOR.txt
sha1: c9b6a4016553541e3940fb9ae98c212fe20a4baa
sha256: e09c0da7781e8a9c14861b616f1a77b050adf3169d8a255e3032ead974b0f5f9
pi@raspberrypi:~ $ sudo tpm2_pcrextend 23:sha256=beefcafebeefcafebeefcafebeefcafebeefcafebeefcafebeefcafebeefcafe
pi@raspberrypi:~ $ sudo tpm2_pcrread
  sha1:
    0 : 0x0000000000000000000000000000000000000000
    1 : 0x0000000000000000000000000000000000000000
    2 : 0x0000000000000000000000000000000000000000
    3 : 0x0000000000000000000000000000000000000000
    4 : 0x0000000000000000000000000000000000000000
    5 : 0x0000000000000000000000000000000000000000
    6 : 0x0000000000000000000000000000000000000000
    7 : 0x0000000000000000000000000000000000000000
    8 : 0x0000000000000000000000000000000000000000
    9 : 0x0000000000000000000000000000000000000000
    10: 0x0000000000000000000000000000000000000000
    11: 0x0000000000000000000000000000000000000000
    12: 0x0000000000000000000000000000000000000000
    13: 0x0000000000000000000000000000000000000000
    14: 0x0000000000000000000000000000000000000000
    15: 0x0000000000000000000000000000000000000000
    16: 0xF3A7930B3F1D51D46C201645C4B85B44F1132B6F
    17: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    18: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    19: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    20: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    21: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    22: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    23: 0xE3B3B197270F5D2F8F1248DD46DD3C62946CF452
  sha256:
    0 : 0x0000000000000000000000000000000000000000000000000000000000000000
    1 : 0x0000000000000000000000000000000000000000000000000000000000000000
    2 : 0x0000000000000000000000000000000000000000000000000000000000000000
    3 : 0x0000000000000000000000000000000000000000000000000000000000000000
    4 : 0x0000000000000000000000000000000000000000000000000000000000000000
    5 : 0x0000000000000000000000000000000000000000000000000000000000000000
    6 : 0x0000000000000000000000000000000000000000000000000000000000000000
    7 : 0x0000000000000000000000000000000000000000000000000000000000000000
    8 : 0x0000000000000000000000000000000000000000000000000000000000000000
    9 : 0x0000000000000000000000000000000000000000000000000000000000000000
    10: 0x0000000000000000000000000000000000000000000000000000000000000000
    11: 0x0000000000000000000000000000000000000000000000000000000000000000
    12: 0x0000000000000000000000000000000000000000000000000000000000000000
    13: 0x0000000000000000000000000000000000000000000000000000000000000000
    14: 0x0000000000000000000000000000000000000000000000000000000000000000
    15: 0x0000000000000000000000000000000000000000000000000000000000000000
    16: 0xBF81C86F1CA8029327BC456C97D4A452269E641408DEC5D4C5060AC1541B438F
    17: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    18: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    19: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    20: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    21: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    22: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    23: 0x42930AF1F2DEC6EC184242E517BDFB892F5247F5B02FE144D81D343D2F6B9DAA
pi@raspberrypi:~ $ 
```

(c) Binding PUF inside Platform Configuration Registers

Fig. 8: Experimental validation of iTPM.

References

[1] S. P. Mohanty, "Security and Privacy by Design is Key in the Internet of Everything (IoE) Era," *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 4–5, March 2020.

[2] D. Fu and X. Peng, "TPM-based remote attestation for Wireless Sensor Networks," *Tsinghua Science and Technology*, vol. 21, no. 3, pp. 312–321, June 2016.

[3] V. K. V. V. Bathalapalli, S. P. Mohanty, E. Kougianos, V. P. Yanambaka, B. K. Baniya, and B. Rout, "A PUF-based Approach for Sustainable Cybersecurity in Smart Agriculture," in *Proc. 19th OITS International Conference on Information Technology (OCIT)*, 2021, pp. 375–380.

[4] D. Lu, R. Han, Y. Shen, X. Dong, J. Ma, X. Du, and M. Guizani, "xTSeH: A Trusted Platform Module Sharing Scheme Towards Smart IoT-eHealth Devices," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 370–383, 2021.

[5] C. Labrado, H. Thapliyal, and S. P. Mohanty, "Fortifying Vehicular Security through Low Overhead Physically Unclonable Functions," *J. Emerg. Technol. Comput. Syst.*, vol. 18, no. 1, September 2021. [Online]. Available: https://doi.org/10.1145/3442443

[6] I. J. Gedeon, P. Snively, C. Frey, W. Almuhtadi, and S. P. Mohanty, "Privacy and Security by Design," *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 76–77, March 2020.

[7] S. K. Ram, S. R. Sahoo, B. B. Das, K. Mahapatra, and S. P. Mohanty, "Eternal-Thing: A Secure Aging-Aware Solar-Energy Harvester Thing for Sustainable IoT," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 2, pp. 320–333, April-June 2021.

[8] V. K. V. V. Bathalapalli, S. P. Mohanty, E. Kougianos, B. K. Baniya, and B. Rout, "PUFchain 2.0: Hardware-Assisted Robust Blockchain for Sustainable Simultaneous Device and Data Security in Smart Healthcare," *SN Computer Science*, vol. 3, no. 5, June 2022.

[9] A. Khan, N. Blair, C. Farnell, and H. A. Mantooth, "Integrating Trusted Platform Modules in Power Electronics," in *Proc. IEEE CyberPELS (CyberPELS)*, 2020, pp. 1–5.

[10] V. K. C. Ramesh, Y. Kim, and J.-Y. Jo, "Secure IoT Data Management in a Private Ethereum Blockchain," in *Proc. IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2020, pp. 369–375.

[11] M. Calvo and M. Beltran, "Remote Attestation as a Service for Edge-Enabled IoT," in *Proc. IEEE International Conference on Services Computing (SCC)*, 2021, pp. 329–339.

[12] V. K. V. V. Bathalapalli, S. P. Mohanty, E. Kougianos, V. Iyer, and B. Rout, "PUFchain 4.0: Integrating PUF-based TPM in Distributed Ledger for Security-by-Design of IoT," in *Proceedings of the ACM Great Lakes Symposium on VLSI (GLSVLSI)*, 2023, p. Accepted, DOI: https://doi.org/10.1145/3583781.3590206.

[13] I. R. Jenkins and S. W. Smith, "Distributed IoT Attestation via Blockchain," in *Proc. 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, 2020, pp. 798–801.

[14] K. T. Kim, J. D. Lim, and J.-N. Kim, "An IoT Device-trusted Remote Attestation Framework," in *Proc. 24th International Conference on Advanced Communication Technology (ICACT)*, 2022, pp. 218–223.

[15] S. F. J. J. Ankergård, E. Dushku, and N. Dragoni, "State-of-the-Art Software-Based Remote Attestation: Opportunities and Open Issues for Internet of Things," *Sensors*, vol. 21, no. 5, p. 1598, February 2021.

[16] S. Sundar, P. Yellai, S. S. S. Sanagapati, P. C. Pradhan, and S. K. K. R. Y, "Remote Attestation based Software Integrity of IoT devices," in *Proc. IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2019, pp. 1–4.

[17] S. K. Ram, S. R. Sahoo, B. B. Das, K. Mahapatra, and S. P. Mohanty, "Eternal-Thing 2.0: Analog-Trojan-Resilient Ripple-Less Solar Harvesting System for Sustainable IoT," *J. Emerg. Technol. Comput. Syst.*, vol. 19, no. 2, March 2023. [Online]. Available: https://doi.org/10.1145/3575800

[18] H. Sun, R. He, Y. Zhang, R. Wang, W. H. Ip, and K. L. Yung, "eTPM: A Trusted Cloud Platform Enclave TPM Scheme Based on Intel SGX Technology," *Sensors*, vol. 18, no. 11, p. 3807, November 2018.

[19] M. Conti, E. Dushku, and L. V. Mancini, "RADIS: Remote Attestation of Distributed IoT Services," in *Proc. Sixth International Conference on Software Defined Systems (SDS)*, 2019, pp. 25–32.

[20] J. Furtak and J. Chudzikiewicz, "Securing transmissions between nodes of WSN using TPM," in *Proc. Annals of Computer Science and Information Systems*, 2015, pp. 1059–1068.

[21] A. Cavoukian, "Understanding How to Implement Privacy by Design, One Step at a Time," *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 78–82, March 2020.

[22] A. J. Cabrera-Gutierrez, E. Castillo, A. Escobar-Molero, J. A. Alvarez-Bermejo, D. P. Morales, and L. Parrilla, "Integration of Hardware Security Modules and Permissioned Blockchain in Industrial IoT Networks," *IEEE Access*, vol. 10, pp. 114 331–114 345, 2022.

[23] N. N. Anandakumar, M. S. Hashmi, and M. A. Chaudhary, "Implementation of Efficient XOR Arbiter PUF on FPGA With Enhanced Uniqueness and Security," *IEEE Access*, vol. 10, pp. 129 832–129 842, 2022.

[24] P. Rojas, H. Idriss, and M. Bayoumi, "Comparative Analysis on the Scaling Properties of Arbiter-based PUFs," in *Proc. IEEE 6th World Forum on Internet of Things (WF-IoT)*, 2020, pp. 1–6.

[25] H. Li, Y. Jin, K. Han, and D. Yu, "A Lightweight XOR-PUF Structure for Resource Constrained Smart Device," in *Proc. IEEE 8th Global Conference on Consumer Electronics (GCCE)*, 2019, pp. 168–169.