SN Computer Science manuscript No.
(will be inserted by the editor)

Fortified-Edge: Secure PUF-based Authentication Mechanism for Integrated Cybersecurity in Collaborative Edge Computing

Seema G. Aarella · **Saraju P. Mohanty** · **Elias Kougianos** · **Deepak Puthal**

Received: date / Accepted: date

Abstract In the current era of collaborative edge computing (CEC), secure and efficient authentication mechanisms are essential for data integrity and privacy across distributed edge data centers (EDCs). This paper proposes a novel PUF-based (Physically Unclonable Functions) authentication framework that leverages homomorphic encryption (HE) to enable secure, lightweight, and scalable authentication. By integrating HE with PUF mechanism the proposed scheme ensures that sensitive authentication data remains encrypted throughout computation and communication, allowing the system to perform verification without direct access to the plaintext PUF responses. The proposed framework is designed to handle the collaborative and dynamic nature of edge networks, supporting real-time authentication in multi-edge scenarios while maintaining low latency and computational efficiency. Experimental results demonstrate that our system achieves high resistance to common attacks, such as eavesdropping, replay, and machine learning-based attacks while adhering to the low-latency requirements of edge networks. The algorithm is verified for safety using the AVISPA SPAN tool and the total time taken for enrollment and authentication is estimated as 5.6s and the average storage size for the encrypted keys are 96 bytes.

Keywords Collaborative Edge Computing · Load Balancing · Physical Unclonable Functions · Authentication, Certificate Authority · Cybersecurity · Smart Village · Homomorphic Encryption · Data Security

Seema G. Aarella
Dept. of Mathematics and Computer Science
Austin College, USA
E-mail: saarella@austincollege.edu

Saraju P. Mohanty
Dept. of Computer Science and Engineering
University of North Texas, USA
E-mail: saraju.mohanty@unt.edu

Elias Kougianos
Dept. of Electrical Engineering
University of North Texas, USA
E-mail: elias.kougianos@unt.edu

Deepak Puthal
Information Technology Systems and Analytics
Indian Institute of Management Bodh Gaya, India
E-mail: deepak.p@iimbg.ac.in

1 Introduction

Collaborative Edge Computing (CEC) works on the distributed model, it is established at the Fog layer that consists of multiple edge devices like Edge Data Centers (EDCs), Edge Routers, and so on. In the CEC environment, the Edge layer can store and process data. When the edge system experiences a heavy computation load, multiple edge devices work together to distribute the processing tasks, through a strategy known as load balancing. CEC enables task offloading for applications in smart villages, this demands a trusted security system to make resource sharing and information safe. Cybersecurity of the CEC components needs to be considered to protect the device, data, and communication network from threats related to cyberspace. Physically Unclonable Function (PUF) is a robust, secure, and lightweight solution for providing hardware security [26]. PUFs are used to authenticate the EDCs during load balancing in a collaborative edge computing environment. Though PUF is secure and difficult to remodel, the drawback lies in the storage and secure communication of the Challenge-Response Pairs (CRP) in a dynamic collaborative computing ecosystem.

Authentication is a necessary feature for the different layers of a smart environment. It is needed to prove the identities of the devices and to ensure confidentiality. The services and the technologies involved in a smart village are shown in the Fig. 1.

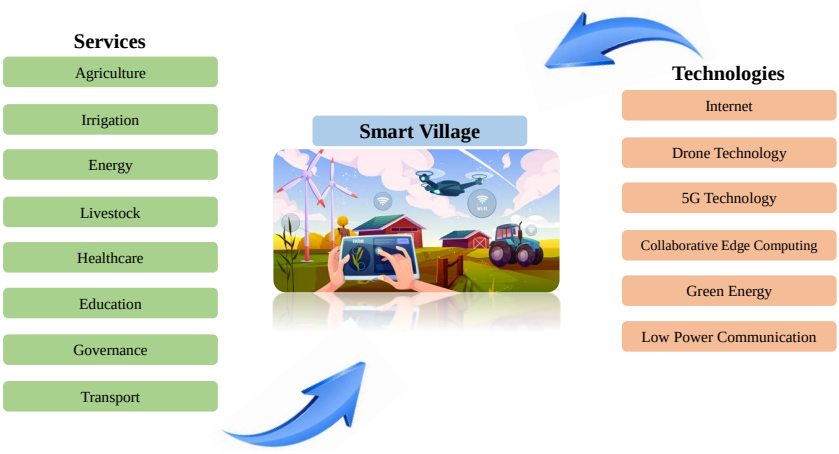

Fig. 1: Overview of Services and Enabling Technologies for Smart Village.

Cryptography is largely used in security solutions for Internet-of-Things (IoT) applications, the challenge however lies in providing a lightweight solution, in terms of generating smaller keys in resource-constrained environments significantly at the physical layer which involves devices with less processing power.

Lightweight security can be provided by hardware as well as software by using lightweight cryptographic algorithms and protocols. Cryptography is used in generating Digital Signatures, using algorithms like Advanced encryption (AES), Rivest Shamir Alderman (RSA), Elliptic Curve Cryptography(ECC), Diffie-Hellman (DH) and SHA-1/SHA-256 [54].

In the current Big Data age edge computing is seen as a scalable solution for processing data at the edge. Edge computing provides speed, reliability, security, scalability, and repeatability, with the ability to process data closer to the IoT (Internet-of-Things) devices. Edge nodes are more vulnerable compared to the cloud and hence security and privacy is an area that needs the most importance [14]. Edge computing brings cloud resources closer to the edge, with a more flexible and scalable architecture. Edge computing implementations are of three forms, a) *Cloudlet*: where a group of computers represent small data centers. b) *Fog computing*: a decentralized infrastructure of computing nodes. c) *Mobile edge computing (MEC)*: also called multiaccess edge computing, provides cloud services to mobile devices at the edge of a network.

Edge enables cooperative load sharing/offloading for effective use of computing resources and to reduce latency in time-sensitive processes. Task offloading process must be *Privacy-aware*, *Energy-aware* and *Security-aware* before collaborating, Quality of Results (QoR) is to be considered to identify acceptable results and make offloading effective [62]. The heterogeneity of the edge computing environment also makes security a challenge and calls for the need to monitor using robust monitoring tools. Appropriate monitoring tools can provide security against data breaches and physical damage to the devices. The key technical challenges related to security and privacy include ensuring protection against physical device tampering, Denial-of-Service (DoS) attacks, flooding attacks, packet forwarding attacks, and other network-based intrusions [47].

Public Key Cryptography is another technology that has been used largely in applications for authorization, authentication, data security, and privacy protection. One of the challenges in using a Public Key Cryptography system for authorization and authentication is the lack of a global Root-of-Trust (RoT), also considering the computation overhead involved in the public key cryptosystems. Any novel authentication system using cryptography should be able to address these issues [64]. Certificate Authority (CA) is a trusted resource that issues Secure Socket Layer (SSL) digital certificates which are a part of the Public Key Infrastructure(PKI). CAs help maintain trust between communicating entities over the internet [40].

Intelligent Public Key management systems are designed to move the authentication process toward the Edge by learning the requirements of the environment. such systems are designed using CA which is centrally located but can be migrated to EDC to reduce latency and cloud dependency [23]. Certificate Authority helps build a RoT between the connected devices in the environment. A centralized CA will be prone to single-point failure whereas a more distributed CA will be flexible effective security management, it has scope to generate different cryptographic keys for different authentication processes [31]. This proposed research is a continuation of the research proposed in [2], where XORArbiter PUFs are used to develop an authentication system for EDCs in CEC. The paper implements the authentication scheme where EDCs participating in load balancing authenticate each other, along with the edge server acting as an EDC verifying and authen-

ticating authority. Current research is an improvement over the former, in implementing an authentication system where data security and device security are considered.

The rest of the research paper is organized as follows; *Section 2* discusses the contributions of current paper, *Section 3* discusses about security threats in collaborative edge computing, *Section 4* discusses the Security-by-Design primitive considered for the current research , *Section 5* discusses various state-of-the-art literature related to the authentication systems, *Section 6* discusses the various cryptography for secure authentication, *Section 7* presents the SRAM PUF-based authentication scheme, *Section 8* discusses the security challenges of CRP data, *Section 9* introduces homomorphic encryption for secure collaborative edge and its technical implementation, *Section 10* explains the experimental results and analysis, *Section 11* discusses the conclusion and future research directions.

1.1 Contributions of the Current Paper

This research aims to develop a secure, sustainable, and lightweight authentication framework that leverages the unique and robust entropy source of SRAM-based PUF to derive reliable CRP data, integrated with a HE scheme for privacy-preserving operations. The proposed system is designed following the Security-by-Design (SbD) paradigm, ensuring that security, sustainability, and scalability are embedded as core architectural principles rather than post-deployment add-ons. The work introduces a PUF–HE–driven authentication protocol to enable trustworthy EDC authentication in a distributed and collaborative edge environment. The protocol comprises two fundamental phases: Enrollment phase and the Mutual Authentication phase, where both the EDC and the authentication server verify each other's legitimacy during load-balancing and task offloading operations, ensuring secure session initiation among peer edge nodes. By employing homomorphic encryption, the system enables computation on encrypted CRP data, allowing verification of PUF-based responses without exposing raw CRP data thus safeguarding against model-building and replay attacks while maintaining low communication and computational overhead. A comprehensive security evaluation of the proposed protocol has been conducted using the AVISPA/SPAN formal verification tool, supplemented by informal analysis to demonstrate resistance to common attack vectors such as man-in-the-middle, replay, and impersonation attacks. The performance metrics further confirm the efficiency and practicality of the proposed approach in resource-constrained edge environments, making it a sustainable candidate for next-generation edge computing infrastructures, including Smart Village ecosystems and collaborative IoT networks.

1.2 Research Motivation

Security and efficiency in authentication pose significant challenges in the evolving field of edge computing. With the increasing number of connected devices in the IoT and edge computing networks, ensuring authenticity, reliability, and confidentiality has become crucial. Traditional centralized authentication systems often struggle to meet the low-latency, high-security requirements of decentralized edge networks. Furthermore, edge devices typically operate under constrained infrastructure, making computationally extensive cryptographic techniques impractical for large-scale, real-time applications.

1.3 Problems Addressed

This research addresses the following key challenges:

- Centralized authentication introduces bottlenecks like single points of failure and latency during dynamic task offloading.
- Conventional cryptographic protocols impose high computational costs, unsuitable for constrained environment.
- Attacks on cyberspace threaten data privacy, integrity and confidentiality, in an untrusted communication network.

1.4 Solutions Proposed

To address these challenges, we propose a PUF-based authentication framework with homomorphic encryption, our approach involves the following solutions:

- Robust lightweight PUF-based authentication scheme that involves EDC enrollment and verification using a homomorphic encryption protocol, allowing computaions to be performed on encrypted data.
- HE ensures secure storage of secret keys by storing the keys in encrypted form, enables secure transmission and verification of the clients without having to expose the underlying plaintext.
- Secure storage of CRP in the encrypted form enhancing the data privacy and provide safety against the known attacks.
- Secure authentication operation and processing on encrypted data without decrypting it, protects the sensitive CRP data throughout the transmission.
- Secure PUF-based CA for mutual authentication enhanced with homomorphic encryption, optimized for low-latency, low-computation authentication.

The proposed system offers a scalable, privacy-preserving, and computationally efficient framework for secure communication across edge networks, making way for a secure and resilient CEC environment.

2 Security Threats in Collaborative Edge Computing

Collaborative Edge Computing is an emerging solution for integrating smart technologies into a smart village environment. CEC is a needed technology for environments where infrastructure is limited, very few EDCs will be available and located at different geographical locations, EDCs in the resource-constrained environment will not have enough processing power and they need to collaborate with other EDCs to get the task done. However, sometimes some EDCs may be idle as the demand becomes low, In such scenarios EDCs can be effectively used through task offloading. Fig. 2 shows some of the current technologies used in the security of the different layers of the IoT architecture, like the Physical Layer, Communication Layer, and Application Layer.

Smart Village is a paradigm that brings Smart City technologies to the villages but with certain limitations on infrastructure. The environment is more challenging because of a lack of computation resources, lack of energy for processing, lack of continuous and high-speed connectivity, and lack of investment. CPS is a form of IoT that typically is a distributed pool of sensors, actuators, and computing nodes, connected through various means of communication. The integration of CPS and IoT is shown in Fig. 3, various applications are developed over the CPS and IoT. CPS comes under the cyberspace-enabled system, where cybersecurity needs to be employed to protect the assets (physical and software), processes, communication, data, and information [13].

CEC is a solution to the computation challenges in a resource-constrained environment, which helps the edge nodes to transfer computational tasks to the nearest available edge node, collaborative computing can reduce the latency related to real-time processing while not requiring more processing power at one node [46].

CEC can be employed either in horizontal collaboration or vertical collaboration based on the type of service and its processing requirements [44]. These processes add multiple devices, like routers, gateways, base stations, and data centers across multiple communication channels, and security should be ensured through all the participating devices to address the technical challenges listed below:

- Reliability
- Security and Privacy
- Energy consumption
- Scalability and Mobility
- Choice of Technology
- Availability of Infrastructure for a given Technology

Any CEC model should be designed to address these challenges. Though edge computing security structure is inherited from the cloud, it is still vulnerable to external attacks, but the scope for implementing security features in fog is bigger compared to cloud [52].

Collaboration across multiple layers of IoT exposes sensitive data to unsafe communication networks, and distributed computing demands data availability at every participating node, the security challenge in such a scenario is secure data storage and communication.

2.1 Security Threats in Collaborative Edge Computing

The edge system's vulnerabilities are determined by the system's security layers and techniques employed to provide security. The security levels of the IoT system paradigm are listed in Fig. 4.

From the figure we can understand the extensive nature of a security system that needs to employ suitable protocols to ensure the complete security of an application at the edge. This research focuses on the authentication and authorization of the edge data centers, it aims to develop a security system that covers security through all the layers of the network and throughout the device or application life cycle which will be discussed further in this manuscript. The factors that contribute towards expanding the attack surface at the edge are .

Hardware Constraints: Edge computing hardware mostly has low computational power and storage capacity, making them incapable of running extensive and dedicated security systems, therefore making them more vulnerable to attacks.

Software Heterogeneities: Having no standardized regulatory system for the devices and servers operating at the edge makes it a difficult task to design a unified protection mechanism. Less robust security system, vulnerable security control system, interconnected nature of the edge devices, larger volume of connected devices, weak access control, malicious or compromised edge devices, and maintaining Quality of Service (QoS), are some of the weaknesses of the edge computing environment which can increase attack possibilities at the edge. The different types of attacks on edge computing are listed in Fig. 5.

Security at the edge is an open problem that looks at solving challenges like DDoS, side-channel attacks, malware injections, authentication, and authorization attacks that threaten privacy and security at the edge,

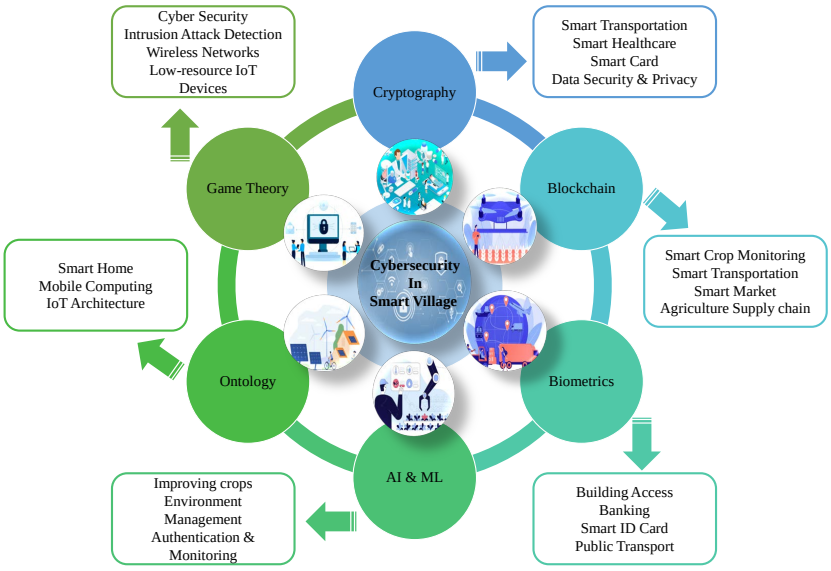

Fig. 2: Overview of Security Technologies and their Applications in Smart Village.

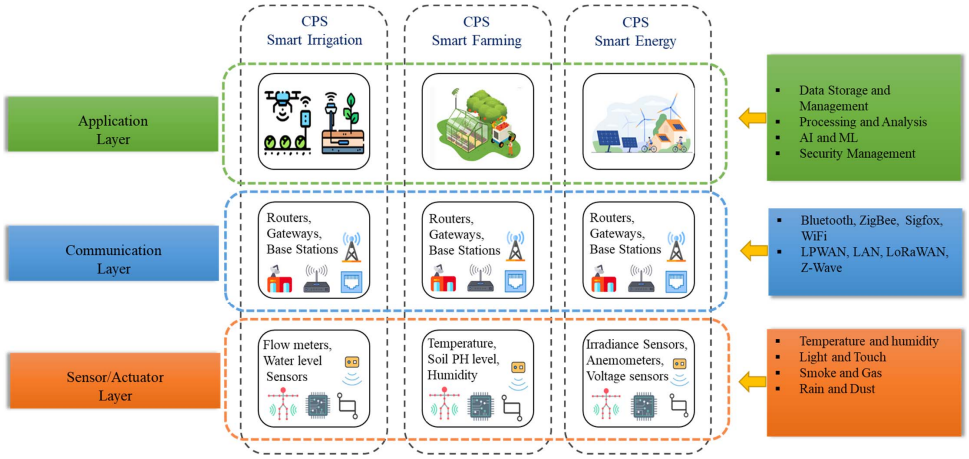

Fig. 3: Integration of CPS and IoT in Smart Village Environment.

apart from possible physical attacks, like device tampering and theft.

2.1.1 DDoS Attacks

Distributed Denial of Service Attacks on the collaborative edge computing infrastructure can have severe consequences, impacting crucial services and their performances. A DDoS attack can shut down the targeted server partially or fully, disrupting the regular flow of traffic and affecting the connected edge devices and other servers at ease. In the edge infrastructure, the edge server layer hosts the edge servers, base stations, and access points, task offloading also takes place at this layer to provide computational resources to aid the busy or low computational servers. Therefore, the distributed edge computing at this edge server layer demands a security system that can protect against adversaries. Technologies like *machine learning, deep learning, software-defined networks(SDN), blockchain, game theory*, and other statistical approaches, along with comprehensive datasets are being studied for detection and mitigation of DDoS attacks across several layers of the IoT architecture [19].

2.1.2 Side-channel Attacks

Side-channel attacks (SCA) are a class of attacks that rely on information leaked through unintended channels. These attacks focus on exploiting the vulnerabilities in the physical implementation of the systems, such as communication channels, power consumption, or electromagnetic radiation. At the edge, closer to the user end SCA becomes a concern. Some of the common side-channel attacks at the edge are *Timing attacks, Power Analysis attacks, Electromagnetic SCA, Cache attacks, Fault Injection attacks, Acoustic SCA, Temperature-based attacks, Radio Frequency Emana-*

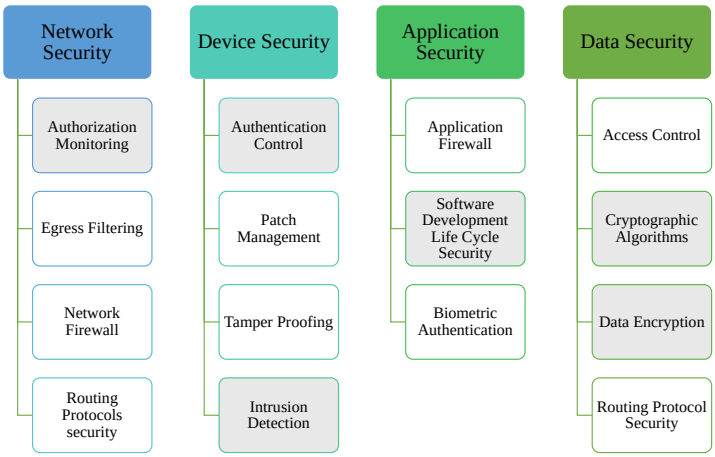

Fig. 4: Security levels at Various Security Layers of IoT Edge Paradigm.

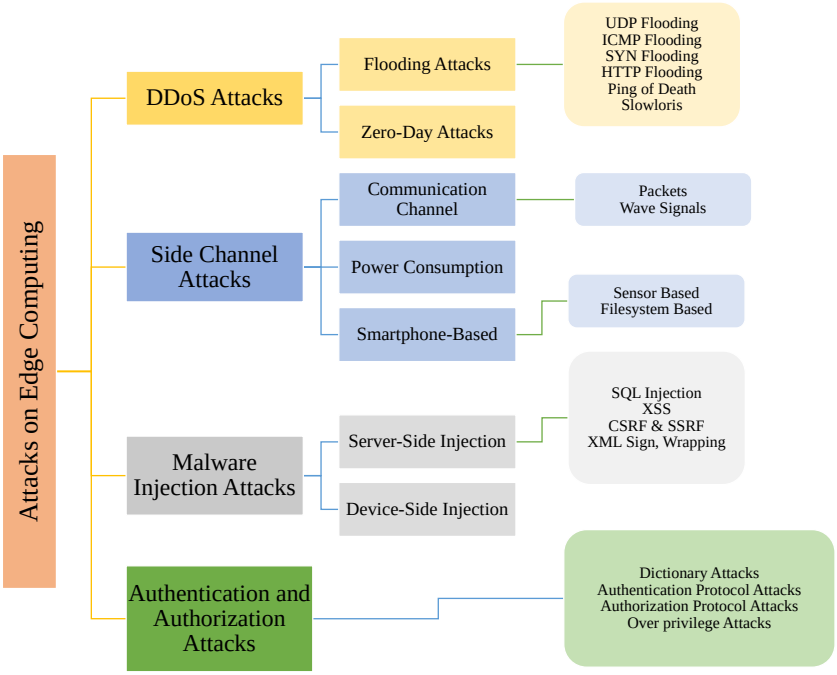

Fig. 5: Different types of attacks on edge computing.

tion attacks. To minimize the risk of SCA additional security systems such as physical security, cryptographic protocols, and algorithms, countermeasures such as secure coding and introducing noise in measurements are to be followed. Among the various side channels, the timing side channel carries the least amount of information and can be defended easily against attacks. However, power and EM side channels carry large amounts of information that require multiple blocks of countermeasures. Hardware and software countermeasures can be implemented to limit the leakage of power and electromagnetic information to the adversaries [7].

2.1.3 Malware Injection Attacks

A malware injection attack involves the attackers trying to insert malicious code or software into a system, application, or communication channels with the intent to compromise the integrity, or functionality of the system. Malware injection attacks can target various layers of the technology and infrastructure stack. Some common types of malware injection attacks are *Code injection, SQL injection, Cross-site scripting, Remote File inclusion(RFI), Command injection, Malicious payload injection, DNS spoofing, Binary injection*, and so on. Mitigating these types of attacks involves implementing, secure coding practices, security protocols and algorithms, firewalls, intrusion detection

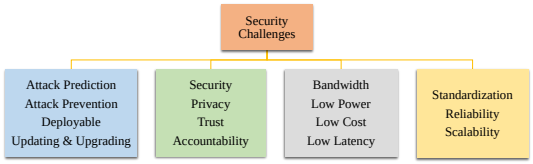

Fig. 6: Security Challenges in Smart Environments.

or prevention measures, and regular software updates to deal with evolving attacks [16].

2.1.4 Authentication and Authorization Attacks

In the data-driven industry, edge computing has shifted data storage and processing closer to the network edge. While this paradigm enables reduced latency, faster response times, and enhanced scalability, it also introduces new security challenges. Core mechanisms such as authentication and authorization, including identity verification, mutual authentication, and secure key management, remain vulnerable to diverse attacks across multiple domains, including communication networks, RFID systems, smart IoT applications, and distributed edge infrastructures [28].

In the rapidly expanding Internet of Things (IoT) ecosystem, where millions of devices are continuously integrated into the network, authentication plays a pivotal role in ensuring trust and device integrity. From a security perspective, authentication acts as the gateway for admitting a new device into the system, guaranteeing that only legitimate entities can participate in communication and computation processes [48]. To mitigate vulnerabilities in such interconnected environments, cryptographic protocols, including symmetric, asymmetric, and lightweight encryption schemes, are employed to provide secure and resilient authentication mechanisms suited to resource-constrained edge and IoT devices.

The collaborative edge computing system requires a robust authentication and authorization mechanism that implements mutual authentication of the collaborating devices, where the devices are identified and verified to ensure the safe offloading and processing of tasks. The drawback is that the processing power and resources available at the edge are less compared to the cloud, hence the demand is for lightweight security protocols for authentication and authorization.

3 Security-by-Design for Smart Villages

Cybersecurity is crucial for any IoT system to preserve the integrity of the application and to overcome the security challenges shown in Fig. 6. Data Privacy and Security, Device Security, and Network Security are crucial for any IoT application. Emerging Technologies

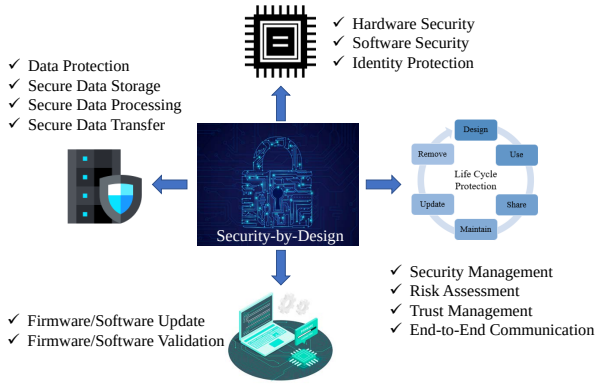

Fig. 7: Overview of Security-by-Design.

like Blockchain, Data Mining, Machine Learning, Artificial Intelligence, SDN, Biometrics, Cryptography, and other hardware and software-based security primitives are majorly used in cybersecurity.

For any environment which is mostly deprived of high-end ICT and physical infrastructure, providing sm art services is challenging. The primitives like agriculture, livestock, irrigation, healthcare, and so on have a variety of sensors and the heterogeneity of the data is higher. Cybersecurity is needed to protect the information as well as the devices. Implementing security protocols in a heterogeneous system across all platforms of communication and layers of IoT makes the process more complicated and expensive. The solution for resource-constrained environments is Security-by-Design (SbD). The key factors to be considered for providing SbD for IoT applications are listed below.

- Identifying the security objectives, requirements, risk analysis and assessment, design and implementation
- Device identification, to establish a unique identity of the device to separate it from counterfeit devices
- Secure authentication, to ensure the communicating device in the network is unique
- Data security and privacy, to recognize the need for data encryption and decryption to preserve the integrity of the data stored in the device
- End-to-End secure communication, to establish an authentic connection between the communicating devices across all layers
- Trust management and infrastructure of trust, policies to govern throughout the life-cycle of the device
- Software/Firmware updation and validation, to ensure the security systems effectiveness over time

Smart environments like smart cities face many security concerns that apply to smart villages as well. External attacks in the form of Eavesdropping, Data leakage, Hacking, IoT Device Hijacking and Ransomware,

Denial-of-Service (DoS), Side-channel attacks, Man-in-the-Middle attacks, Rouge and Counterfeit devices, Machine Learning attacks, Modelling attacks, Botnet attacks, Physical attacks etc. are always a threat to the security of the system [29].

An Intrusion Detection System(IDS) based architecture works as SbD for smart villages [6]. PUF-based security primitive is an efficient design to implement security from the early stages of design and development, which starts from device unique identification eliminating the fake or refurbished devices from the unique devices, making the system more secure and energy efficient throughout the lifecycle [46]. An overview of Security-by-Design is shown in Fig. 7.

It shows how SbD forms an integral part of security in every stage of the process or application, from device identification to lifecycle management and secure operation throughout the lifecycle. SbD incorporates seven fundamental principles [36]:

1. Cybersecurity should be proactive, not reactive
2. It should be a default feature
3. It should be embedded into the design
4. It should be end-to-end to provide lifecycle protection
5. It should be implemented as a full functionality-positive-sum, not zero-sum
6. It should have visibility and transparency
7. Cybersecurity solutions should have respect for users

Both software and hardware-based cybersecurity solutions are available, but hardware-based security satisfies the 7 principle requirements of SbD. Likewise, Hardware-Assisted Security (HAS) provides security to the electronic system as well as the information being processed by the system. HAS is a suitable security solution for many emerging CPS-based ecosystems, Internet of Medical Things (IoMT), smart transportation, Industrial Internet-of-Things (IIoT), and so on. HAS can be implemented as a special hardware security design, chip-centric design, or system-level security design, integrating security features before the IoT is formed [37]. PUF is an emerging choice for hardware-assisted security, PUF which utilizes the process variations of the nanoelectronics product is a robust cybersecurity alternative for electronic devices and complex CPS systems. PUF has been proposed for the cybersecurity of IoMT, smart grid, and secure key generation in cryptographic security systems [41].

4 Related Prior Research

Covered in this section are the related prior research involving device security and authentication methods. The research [30] uses PUF to authenticate IoT devices using the PUF CRPs, to minimize the storage space for storing all the CRPs a PUF can generate, it uses only one CRP to store for authentication by message encryption. The research [61]uses multiple PUFs to authenticate IoT devices in resource constrained environment, the generated PUFs are not stored or transmitted, and the CRP once used during authentication will not be used again hence reducing the risk of replay attack, and CRP exposure during subsequent authentication, also reduces the chances of machine learning attack because of the use of multiple PUFs. A study on optimization of both hardware and software-based PUFs using innovative Computer-aided Design (CAD) technology is proposed in the research [58]. The research explores the integration of digital PUFs into the hardware security primitive, since the analog PUFs are challenging to integrate, also the digital PUFs are more stable.

An edge-assisted decentralized authentication architecture (EADA) is proposed in the paper [59], for vehicular networks. An authentication server is involved which handles the registration and mutual authentication of the edge nodes. PUF addresses the fundamental problem of device identification which is essential for Trust Management, Root-of-Trust can be achieved using PUF-based digital signatures and authentication methods [10].

Elliptic Curve Encryption (ECC) technology-based authentication system is proposed in the research [57], for the security of smart grids. In this method, the device privacy information is encrypted with the use of random numbers and timestamps to increase security. Blockchain-based privacy-preserving authentication protocol for Vehicular Edge Computing (VEC) has been proposed in [60]. It used smart contracts issued by the certificate authority (CA) to link the public key to the user identity, which can be used for mutual authentication. A secure vault-based authentication system to authenticate the IoT devices and IoT server is discussed in [53]. This research uses a multi-key authentication mechanism where the key values keep changing, to overcome the drawbacks of single-key authentication systems.

Smart contracts and blockchain-based mutual authentication systems for inter-edge and intra-edge devices and servers are presented in the research [12], which addresses security issues like anonymity and confidentiality in the CEC environment. A Privacy-aware authentication protocol based on a combination of PUF and blockchain is proposed in the research [63], for cloud-edge multiserver IoT systems. Blockchains are used to store the Mapping Correlations (MCs) of the CRPs. In the field of Internet-of-Vehicles (IoV), technologies like blockchain, PUFs, and dynamic Proof-of-Work (dPoW) consensus algorithms are used to develop a trust management model, where PUFs are used

for the unique identity of the vehicle forming the trust element [27].

Further, a comparative table of research involving different security protocols for authentication and authorization for various applications are listed in the Table 1 along with the current research.

5 Cryptography for Secure Authentication

Cryptography plays a cruial role in secure authentication systems by safegaurding sensitive information from unauthorized access and tampering, it is privacy preserving, it enables secure storage and communication of data, and provides security against external threats that could damage the system, enhancing the confidentiality. Precisely, it is a method of storing and transmitting data in a particular form that can be read and processed by the intended entity.

The primary encryption methods used for secure authentication include public key cryptography, cipher-based hash functions, and multi-factor authentication systems.

5.1 Public Key Cryptography (PKC)

It utilizes asymmetric key pairs, public key and private key, for secure data exchange. Public keys are openly shared, private keys are confidential, and private key is used to decrypt the data encrypted using the corresponding public key, this process ensures secure communication [1].

- Algorithms used in PKC are RSA (Rivest-Shamir-Adleman), which uses factoring of large numbers to secure messages, it is widely used for secure data transmission and digital signatures. Elliptic Curve Cryptography (ECC) provides similar security but with smaller key sizes compared to RSA.
- The potential risks involved in PKC are resistance against Chosen Ciphertext Attacks (IND-CCA) and cryptanalysis techniques that are used to analyze and potentially break the public key system [15].
- Its reliance on hard mathematical problems makes it computationally intensive compared to symmetric key cryptography and makes it less suitable for encrypting large amounts of data.
- Applications of PKC include SSL/TLS (Secure Sockets Layer/Transport Layer Security) protocols which provide secure communication over networks, like the Internet and VPNs (Virtual Private Networks). Digital Signatures and Certificates, Blockchain and Cryptocurrencies, Email Encryption, Authentication and Authorization.

5.2 Symmetiric Key Cryptography (SKC)

Also known as secret-key cryptography, this method utilizes a single key for both encryption and decryption process, hence the key must be securely maintained. This method is widely used because of its simplicity, speed, and efficiency in handling large amounts of data.

- AES (Advanced Encryption Standard), is one of the popular algorithms standardized by The National Institute of Standards and Technology (NIST), it supports 128, 192, and 256-bit key sizes, providing stronger security compared to its earlier version DES (Data Encryption Standard) and Triple DES (3DES) which uses key sizes up to 56-bit which is used twice or thrice, thus the effective key length up to 168 bits.
- Blowfish and Twofish algorithms provide enhanced speed and flexibility. Blowfish is a variable length key, 64-bit block cipher which can be optimized in hardware unlike other algorithms that often use software, and it is proven to be the fastest algorithm in the batch. Twofish is a block cipher of 128 bits and key sizes up to 256 bits [17].
- Applications of SKC include data encryption, secure VPNs and connections, secure authentication, and disk and file encryption to protect data on storage devices.

5.3 Hashing Algorithms

Hashing algorithms are largely used for securely storing passwords or keys rather than encryption. It is a mathematical algorithm that plots data of random size to a bit string of fixed-length hash function, which cannot be easily reversed. It ensures an exact hash for the same data, and hash value determination is very fast. A small change in the hash value will result in different outputs.

- The Secure Hash Algorithm (SHA) family includes cryptographic functions called hash functions designed for data security, these functions consist of bitwise operations, modular additions, and compression functions. These algorithms are designed to be one-way functions, implying that once transformed it is impossible to bring back the hash values to their original data form.
- SHA-1, SHA-2, and SHA-3 are the algorithms of interest in this family, that were successively developed with increasingly strong encryption that is resistant against hackers.
- Applications of hash algorithms include digital signatures, message authentication, key derivations, pseudo-random functions, secure password storage, secure certificates, and so on.

Table 1: Comparative Table for State-of-the-Art Literature with Security Analysis.

Research	Year	Algorithm	Application	Security Pros & Cons
Puthal et al. [45]	2018	AES-based Symmetric Encryption	Authentication and Load Balancing of EDCs	**Pros:** Strong cryptographic protection, resistant to replay and MITM attacks. **Cons:** High computation cost and latency, unsuitable for resource-constrained edges.
Barbareschi et al. [9]	2019	PUF-based PHEMAP	Fog-IoT Systems	**Pros:** Hardware-level identification resists spoofing. **Cons:** Complex key management limits scalability.
Jing et al. [56]	2020	Blockchain, Smart Contracts	Smart Grid Edge Computing	**Pros:** Decentralized authentication eliminates single point of failure. **Cons:** High latency from blockchain consensus; limited for real-time edge nodes.
Hathal et al. [25]	2020	TA, TESLA	Vehicular Communication Systems	**Pros:** Provides time-based key freshness, reduces replay attacks. **Cons:** Sensitive to synchronization delays.
Li et al. [34]	2020	p-KNN	Healthcare Edge Computing	**Pros:** Privacy-preserving authentication using nearest-neighbor approach. **Cons:** Scalability and energy constraints in large sensor networks.
Zhang et al. [63]	2021	SRAM PUF, Blockchain	Multiserver Authentication in Cloud-Edge IoT	**Pros:** Resists replay and cloning attacks with immutable audit trails. **Cons:** Blockchain overhead increases with network size.
Puthal et al. [43]	2022	Decision Tree, PoAh	Data Aggregation in IoT Edge	**Pros:** Reduces false-positive trust evaluations. **Cons:** Limited ML attack resistance; lacks encryption.
Aarella et al. [2]	2022	XOR-Arbiter PUF	EDC Authentication in CEC	**Pros:** Hardware-level security, lightweight, resistant to cloning. **Cons:** Requires secure CRP storage.
Aarella et al. [3]	2023	PUF-based CA	Mutual Authentication of EDCs	**Pros:** Eliminates central database dependency; high replay resistance. **Cons:** Slightly higher setup latency.
Fugkeaw et al. [20]	2023	LightPUF, Real-Time Intrusion Detection	Fog-Assisted IIoT Systems	**Pros:** Detects rogue devices dynamically in fog-IIoT; low-latency, lightweight security layer for distributed authentication. **Cons:** Detection accuracy may degrade under dynamic fog mobility; limited testing under coordinated attack scenarios.
Tun et. al. [55]	2024	PUF, Paillier Homomorphic Encryption	IoT Device Authentication	**Pros:** Combines PUF with Paillier HE enabling verification on encrypted CRPs; strong protection against ML, replay, and impersonation attacks. **Cons:** High computational overhead for large-scale networks; performance depends on HE parameter optimization.
Assaqty et al. [8]	2025	SPUFChain (PUF, Permissioned Blockchain, ECC)	Supply Chain Management (SCM) in IoT	**Pros:** Integrates SRAM PUF with permissioned blockchain (Hyperledger Fabric) and smart contracts for lightweight, decentralized authentication; resists replay, impersonation, and MITM attacks. **Cons:** Requires Hyperledger Fabric infrastructure, which adds setup complexity; partial dependence on service provider during registration.
Rullo et al. [49]	2025	ECC-based One-Way Authentication using Stable PUF	Identification Tags, Smart Supply Chain	**Pros:** Memory-less ECC–PUF design avoids key storage, resists insider and cloning attacks; highly stable under PVT variations. **Cons:** One-way authentication limits mutual verification; ECC computations slightly heavy for ultra-low-power tags.
Fortified-Edge(Current Paper)	2025	PUF, Homomorphic Encryption (HE)	EDC Enrollment and Authentication	**Pros:** Privacy-preserving computation on encrypted CRPs; event-driven mutual authentication **Cons:** Requires HE tuning for real-time lightweight deployment.

– Though SHA has evolved strongly it is still vulnerable to brute force, dictionary attacks, and quantum attacks.

5.4 Challenge-Response Authentication Protocols (CRAPs)

CRAPs are essentially used for securing communications in networks, particularly involving drones and IoT devices. They leverage the physical-layer characteristics to authenticate devices based on their unique channel responses, enhancing security without relying on only cryptographic methods.

– Drone networks can use CRAPs to authenticate communications among the drones using the physical characteristics of the wireless channels [35]. A novel CRAPs method integrates Intelligent Reflecting Surfaces (IRS) to modify the channel environment, allowing the receiver to verify the transmitter's identity based on the altered signal characteristics [22].
– PUFs are employed in CRAPs to enable mutual authentication between IoT devices without a central server ensuring security against impersonation and eavesdropping and other adversaries by encoding the transmitted information.
– While CRAPs are robust and greatly enhance security they face challenges such as environmental variability and other potential vulnerabilities of the physical layer.

6 SRAM PUF-based Authentication Scheme

6.1 Load Balancing

When considering PUF-based authentication systems, the handling of PUFs is carefully considered. Though PUF is an ideal hardware security solution that is hard to remodel, when storing the PUFs locally in the device it becomes a risk. The PUF challenge Response Pairs stored in the cloud are highly secure due to the complex cloud security systems and policies. The EDCs at the edge rely on mutual authentication like the majority of the IoT devices. Mutual authentication is achieved using methods like (i) Symmetric Cryptography Algorithms, (ii) Public Key Infrastructure (PKI), and other mechanisms [18]. The following solutions are implemented in the PUF-based authentication scheme

1. Edge server-based architecture for EDC verification and authentication
2. Authentication scheme without cloud dependency
3. XORArbiter PUF-based authentication scheme for EDC verification and authentication, to identify the EDCs by registering them through a Client-Server authentication protocol
4. A Mutual authentication scheme for Client-Client authentication during load balancing
5. Removing with need for storing the CRP database locally at the EDC, hence reducing the risk of data compromise
6. SRAM PUF-based certificate generation to establish root-of-trust between EDCs participating in load balancing
7. Mutual authentication scheme based on certificates

Since the EDCs are capable of limited processing only, a CEC environment helps offload the processing tasks to other EDCs in the environment. The edge layer as shown in Fig. 8, helps in scaling the processing but also through load balancing calls for more secure architecture. The EDCs participating in the load balancing must be verified and authenticated to ensure data security and communication security.

6.2 Storage Space Complexity

PUFs are hardware-based lightweight security options for IoT (Internet-of-Things) devices and data centers. PUFs generate CRPs which are stored. Databases are needed for storing the PUF challenges and their responses and the device in question is authenticated based on the verification of the device's response to a given challenge. CRP database can be stored in the cloud, in real-time processing and CEC scenarios where authentication does not involve the cloud, the CRP data needs to be stored at the edge. Edge infrastructure may not always provide enough storage space and storing Databases at every EDC is a threat to data security as it can undergo data breach.

The size of the CRP data may vary considerably depending on the number of devices involved in the authentication process, storage space is dependent on the EDC infrastructure. Fig. 9 shows the storage space complexity concerning dynamic load balancing, as the number of nodes to be authenticated increases the number of CRP sets to be stored also increases. Space is not an issue in static load balancing where one EDC always transfers loads to the same other EDC.

In Dynamic load balancing the EDCs can choose any available EDC at that given time among the available EDCs, thus the need for storing more CRP sets for each of the participating EDCs, which in turn calls for larger storage space. Therefore, we need a solution where we can exploit the security of the PUFs and solve the storage space complexity.

Resource-constrained environments of the future that head towards CEC, data security and data storage are seen as an issue [44]. Such an environment needs a secure authentication system that does not need more storage space or processing power. Based on the points,

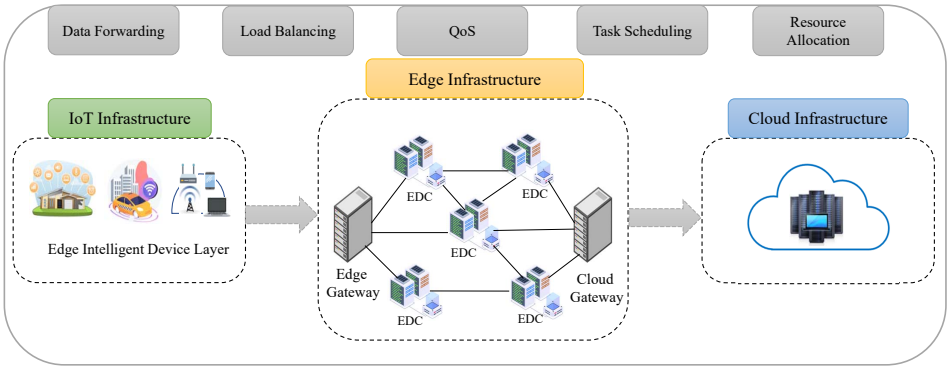

Fig. 8: Overview of Edge Data Centers and Load Balancing.

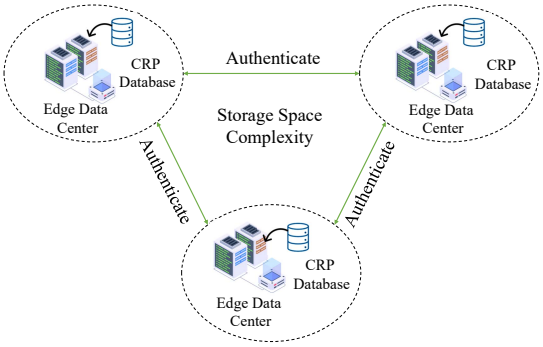

Fig. 9: Space Storage Complexity in Dynamic Load Balancing.

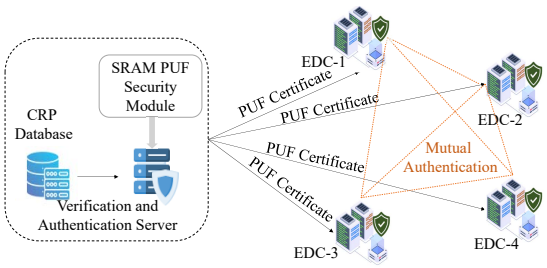

Fig. 10: Architecture of the proposed PUF based CA scheme.

the problems that are addressed in this paper are listed below.

1. Need for Lightweight, secure, and robust authentication protocol
2. At-the-Edge Authentication protocol, without cloud dependency
3. A low latency authentication protocol
4. Solving the storage space complexity when storing CRP data is involved in PUF-based authentication schemes
5. Data security of the CRP

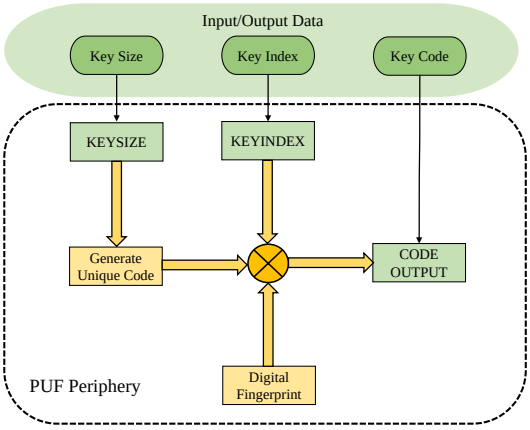

Fig. 11: SRAM PUF Key generation using Input Data and Digital Fingerprint.

6.3 PUF-based Certificate Authority

A typical PUF Certificate Authority (CA) scheme mainly focuses on removing the need for storing the CRP database locally in the EDCs, which participate in load balancing and use PUF-based security schemes for authenticating each other. Hence, we propose a scheme where a *Verification and Authentication Server* is centrally placed at the edge, making it independent of the cloud server. SRAM PUF has applications in secure key generation, device authentication, data protection, chip asset management, anti-counterfeiting, anti-cloning, supply-chain protection, and so on. The architecture of the SRAM PUF-based CA scheme is shown in Fig. 10.

SRAM PUF is used for the PUF key generation for authenticating the EDCs. The overview of the SRAM PUF key code generation is shown in Fig. 11.

The PUF keys generated from the given input data and the digital fingerprint of the SRAM PUF are the responses that are stored in the CRP database of the *Verification and Authentication Server*. The generated key code contains a 32-bit key header, which includes *Key Index*, ranging from 0-15, and *Key Size* which ranges from 64-bits to 4096-bits. The digital fingerprint is gen-

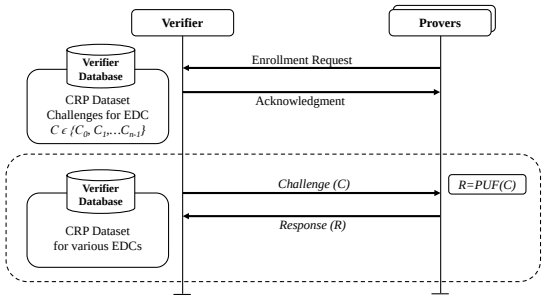

Fig. 12: Enrollment process in authenticaion protocol.

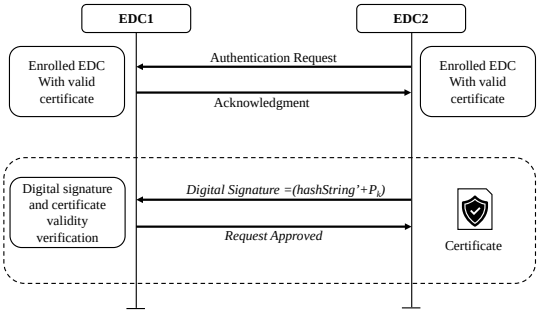

Fig. 13: Mutual authentication protocol for EDCs during load balancing.

erated when the SRAM PUF is powered up, the startup data from the SRAM PUF combined with the Activation code input generates the digital fingerprint of the SRAM PUF. This 256-bit digital fingerprint is the root key used for encryption/decryption of the user keys. The SRAM PUF bits can be extracted with less error rate using different extractor algorithms and error correction algorithms, through which stable PUF bits can be generated [32].

To mitigate the potential single point of failure in the centralized Verification and Authentication Server design, the proposed PUF-based CA architecture can be extended into a distributed and fault-tolerant model. This can be achieved by deploying multiple authentication server instances in a hierarchical or federated edge structure, where each node maintains synchronized encrypted CRP metadata, ensuring service continuity even if one node fails. Additionally, integrating the authentication server with a permissioned blockchain ledger enables decentralized trust management and tamper-evident recordkeeping of authentication transactions.

6.4 Enrollment Process of the Authentication Protocol

The PUF-based CA authentication scheme has two stages, the enrollment stage and the authentication stage. In the enrollment stage, each EDC participating in the load balancing gets to verify itself with the server. The Initial verification is done based on the CRP verification process of the scheme. The EDCs that are using PUF modules will participate in the verification process by sending the request to the server, the server sends the encrypted challenges and gets the encrypted response from the EDC, upon verification of the received response with the CRP in the database the server authorizes the EDC and enrolls it as authentic. The typical enrollment process is shown in Fig. 12. Each EDC is authorized and issued an SRAM PUF-based certificate, unique to each EDC which is generated using the parameters like EDC ID and MAC ID.

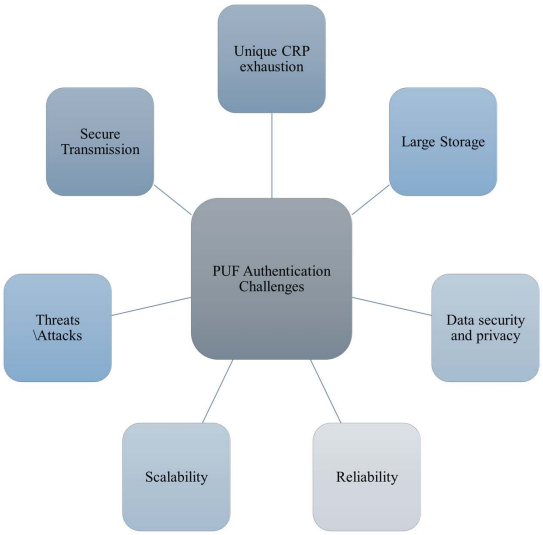

Fig. 14: Challenges of CRP in PUF-based authentication systems.

6.5 Mutual Authentication Scheme of the PUF-based CA

In the typical Mutual authentication scheme each participating EDC is pre-verified according to the enrollment protocol and issued an SRAM PUF-based certificate, during load balancing the participating EDCs will send the authentication request to the other EDC, upon receiving the acknowledgment the verifier EDC requests for the verification package, the prover EDC sends the digital signature consisting the hash string of parameters like certificate, device id, mac id, issuer id validity, and timestamp so on, encrypted with the private key. The verifier decrypts the digital signature and verifies the certificate to approve or decline the authentication request. The typical mutual authentication protocol for CA is shown in Fig. 13.

7 The CRP Data - Security Challenges

PUF-based authentication systems face challenges concerning CRP data storage and its security, as shown

in Fig. 14. An extensive CRP dataset is needed for employing the authentication protocols in an IoT environment. The demand for a unique set of CRPs for each device and the continuously increasing devices demands a huge dataset. Reuse of the CRP is not a possibility hence an exhaustion in unique CRP may arise, limiting the number of possible authentications, During a Denial-of-Service (DoS) attack the adversaries can attempt repeated authentication requests that can eventually deplete unique challenges [11]. Designing and testing any authentication system requires large and unique CRP data; this increases the storage space requirement. It becomes challenging to provide substantial memory to store sensitive CRP data on resource-constrained IoT devices [42].

CRP data being sensitive must be protected from unauthorized access, which includes securing against side-channel attacks and direct attacks on the memory where the CRPs are stored. Data breaches and data manipulation are harmful to the secure operation of the authentication system. Data manipulation can also be done through direct attacks on the PUF circuit itself, like glitch attacks or fault injection attacks, which target the input voltage or clocks in the circuit resulting in abnormal behavior of the PUF resulting in system failure, or device failure [33]. PUFs are sensitive to noise, environmental variations, any changes in temperature, voltage, electromagnetic field in the proximity or seismic activities can cause bit flips in the PUF responses causing weak PUF generation, responses with errors that tend to reduce the stability and reliability of the PUFs [51].

PUFs can become targets of replay and cloning attacks if CRPs are being re-used for multiple authentications or storing them in plaintext can be intercepted. It is difficult to achieve PUF stability as the PUF behavior is influenced by internal noise, this requires error correction mechanisms which could add the area overhead and computational overhead. A distributed environment for PUF-based authentication could be an option for the secure storage of large CRP data, however, a synchronized authentication system will fail if not properly managed. PUFs can be intercepted during transmission from the IoT device to the server, hence there is a need for secure communication which is challenging as it requires more bandwidth and energy which is challenging in a resource-constrained IoT. Managing and storing CRPs becomes increasingly difficult as the number of connected IoT devices keeps growing, creating scalability issues.

8 Homomorphic Encryption for Secure Collaborative Edge

Homomorphic Encryption (HE) is an advanced cryptographic technique that allows computations on the encrypted data without needing to decrypt it. This feature makes HE a powerful tool for secure authentication enabling data confidentiality and integrity in an environment like IoT. This property is particularly advantageous for privacy-preserving applications in IoT, cloud computing, and distributed edge computing ecosystems like CEC [38]. HE in a PUF-based authentication system enables privacy-preserving device authentication for IoT devices and EDCs, devices can send encrypted CRP to the verifier (server), where the server verifies the authenticity without revealing underlying sensitive data. This feature is particularly useful in a CEC environment where mutual authentication is needed in real-time during load balancing, the HE can operate without exposing the unique device identifiers.

There are three types of HE schemes, Partially Homomorphic Encryption (PHE), Fully Homomorphic Encryption (FHE), and Somewhat Homomorphic Encryption (SWHE). While PHE only allows certain types of computations to be performed, like addition or multiplication an unlimited number of times, the FHE allows arbitrary computations to be performed [5]. FHE is computationally expensive and cannot be used in constrained devices, but PHEs like ElGamal, Benaloh, and Paillier have reasonable computation speed [50].

The Paillier Encryption (PE) scheme is an asymmetric encryption algorithm for public key cryptography invented by Pascal Paillier in 1999. This method allows certain mathematical operations to be performed on the encrypted data without needing to decrypt it first. Its main feature is an additive homomorphism, which allows the operations of cipher texts corresponding to the addition of underlying plaintexts [39]. Paillier encryption relies on modular arithmetic and certain number-theoretic operations. An HE scheme is primarily characterized by four operations: Key generation, encryption, decryption, and evaluation [4].

Key Generation To generate the public and private keys, the following steps are performed:

Two large prime numbers are chosen p and q Compute $n = p \times q$ and $\lambda = \text{lcm}(p-1, q-1)$, where lcm denotes the least common multiple. Choose a random integer g such that $g \in \mathbb{Z}^*_{n^2}$ by checking whether $\gcd(n, L(g^\lambda \mod n^2)) = 1$ where the L functionis defined as $L(u) = \frac{u-1}{n}$ for every u from the subgroup $\mathbb{Z}^*_{n^2}$. The public key is given by (n, g), and the private key is (p, q).

Encryption For a given a message m and a randomly chosen integer r, the ciphertext c is calculated

as:

$$c = E(m) = g^m \cdot r^n \mod n^2 \tag{1}$$

Decryption To decrypt a ciphertext $c < n^2$, the decryption can be done using the private key λ as:

$$D(c) = \frac{L(c^\lambda \mod n^2)}{L(g^\lambda \mod n^2)} \mod n = m, \tag{2}$$

where the function $L(u)$ is defined as,

$$L(u) = \frac{u-1}{n} \tag{3}$$

Homomorphic Addition of Encrypted Messages Paillier encryption is homomorphic with respect to addition. If we have two encrypted messages, $E(m_1)$ and $E(m_2)$, their product corresponds to the sum of the plaintexts:

$$\begin{aligned} E(m_1) \cdot E(m_2) &= \left(g^{m_1} r_1^n \mod n^2\right) \cdot \left(g^{m_2} r_2^n \mod n^2\right) \\ &= g^{m_1+m_2} (r_1 \cdot r_2)^n \mod n^2 \\ &= E(m_1 + m_2) \end{aligned} \tag{4}$$

Additional Homomorphic Operations Additional homomorphic properties describe the cross-relations between various operations on the encrypted data and plaintexts. Equations (5) and (7) illustrate the fundamental homomorphic property of the Paillier encryption scheme, which enables mathematical operations to be performed.

$$E(m_1) \cdot E(m_2) \mod n^2 = E((m_1 + m_2) \mod n), \tag{5}$$

Equation (5) represents the additive homomorphism; multiplying two ciphertexts corresponds to the addition of their plaintext values. This property is essential in privacy-preserving computations where the aggregation of sensitive data must occur securely.

$$E(m_1) \cdot g^{m_2} \mod n^2 = E((m_1 + m_2) \mod n), \tag{6}$$

Equation (6) demonstrates that multiplying a ciphertext $E(m_1)$ with an encryption of a known plaintext g^{m_2} produces an encryption of the sum $(m_1 + m_2)$ mod n. This allows the addition of a known constant or offset to encrypted data without accessing its plaintext form.

$$E(m_1)^{m_2} \mod n^2 = E((m_1 \cdot m_2) \mod n). \tag{7}$$

Equation (7) shows the scalar multiplication property, where raising a ciphertext to a plaintext exponent results in the encryption of the product of their plaintexts. In practical terms, this enables scaling or weighting of encrypted data.

8.1 Technical Implementation

The CRP dataset from the SRAM PUF-based authentication model has been used with 1000 unique CRPs. The system involves a verifier and two clients (EDCs). The initial setup includes the enrollment phase, where the EDCs enroll themselves after verifying their authenticity with the authorizing server. The server (verifier) stores the encrypted CRP for each device (EDC client). During the load balancing when the EDCs must authenticate each other, they follow a mutual authentication protocol that will allow them to exchange the unique identifiers along with the certificate for verification and authentication. Here mutual authentication is time-sensitive hence the HE encryption is not employed as it may become computationally extensive or introduce latency. Since this CA scheme involves the verification of certificates among the enrolled devices, access control acts as a prime feature, where the certificate validity is controlled by the verifier. Hence the authentication system is designed to give prominence to the enrollment and initial verification of the EDCs participating in the load balancing. The enrollment and EDC verification process using HE is shown in the Algorithm 1.

The scheme involves a public key for encryption and a private key for decryption. The Paillier Encryption allows operations on the encrypted data ensuring that the PUF responses remain secure as they are sent from the device to the verifying server. This method enhances the secure transfer of sensitive PUF responses where the data is not exposed during the transmission and deters any interceptions or attacks. The following are the steps involved in the process:

8.1.1 Server-side Enrollment

The server enrolls the EDC by generating and storing the encrypted CRP. The enrollment phase concludes by having an encrypted record of the CRP pair $(\bar{C}_i, \bar{R}_i)$, this provides data security and is privacy-preserving against any data breaches.

- The server generates the Paillier keys pair (*public_key* and *private_key*)
- Server saves the public_key for later use
- Server generates a random challenge C_i and sends it to EDC
- The EDC uses its PUF to generate the response $R_i = \text{PUF}(C_i)$
- The EDC encrypts the response using *public_key* and sends it to server
- The server stores the encrypted CRP $(\bar{C}_i, \bar{R}_i)$, for future authentication attempts

Algorithm 1 PUF-based Authentication Using Paillier Encryption

Require: Paillier key pair (public_key, private_key) generated by server
Ensure: Device is authenticated successfully if PUF responses match

Server-side Enrollment Phase:
Server generates (public_key, private_key)
Server saves public_key to file
Server waits for device connection
Server generates a random challenge C_i
Server sends C_i to the device
Device receives C_i
Device computes the PUF response $R_i = \text{PUF}(C_i)$
Device encrypts R_i using public_key:

$$\bar{R}_i = \text{public\_key.encrypt}(R_i)$$

Device sends encrypted response $\bar{R}_i$ to the server
Server stores encrypted challenge-response pair $(\bar{C}_i, \bar{R}_i)$
Server completes the enrollment phase

Server-side Authentication Phase:
Device sends authentication request with device ID
Server retrieves stored encrypted challenge-response pair $(\bar{C}_i, \bar{R}_i)$
Server decrypts stored challenge:

$$C_i = \text{private\_key.decrypt}(\bar{C}_i)$$

Server sends C_i to the device
Device receives C_i
Device computes the PUF response $R'_i = \text{PUF}(C_i)$
Device encrypts R'_i using public_key:

$$\bar{R}'_i = \text{public\_key.encrypt}(R'_i)$$

Device sends encrypted response $\bar{R}'_i$ to the server
Server performs homomorphic subtraction:

$$\text{diff} = \bar{R}_i - \bar{R}'_i$$

Server decrypts the difference:

$$\text{decrypted\_diff} = \text{private\_key.decrypt(diff)}$$

if decrypted_diff $= 0$ **then**
    Server authenticates the device as **successful**
    Server sends success message to the device
else
    Server denies authentication (authentication failed)
    Server sends failure message to the device
end if

8.1.2 Server-side Authentication

During this phase, the server verifies the EDC's authenticity by comparing the newly generated PUF response with the stored response.

- The EDC requesting authentication sends the request to the server along with its unique device identifiers
- The server retrieves the encrypted CRP $(\bar{C}_i, \bar{R}_i)$ corresponding to the EDC ID
- the server decrypts the challenge using private_key to obtain the original challenge C_i = private_key.decrypt($\bar{C}\_i$)
- The server sends the challenge to the device, the device generates PUF response, and encrypts it using the *public_key* and sends it to the server, $\bar{R}_i$ = public_key.encrypt(R_i)
- The server performs homomorphic subtraction on the encrypted response sent by the EDC and the encrypted response stored during the enrollment phase for the EDC diff = $\bar{R}_i - \bar{R}'_i$
- If the difference is non-zero, means the PUF responses do not match, and the authentication fails
- If the difference is zero, the authentication is successful.

9 Experimental Results and Analysis

The HE protocol is tested for 20 clients that participate in the enrollment and verification process. The server handles the requests for the EDC clients. The protocol is run on the Python environment, using socket programming for clients and server communication.

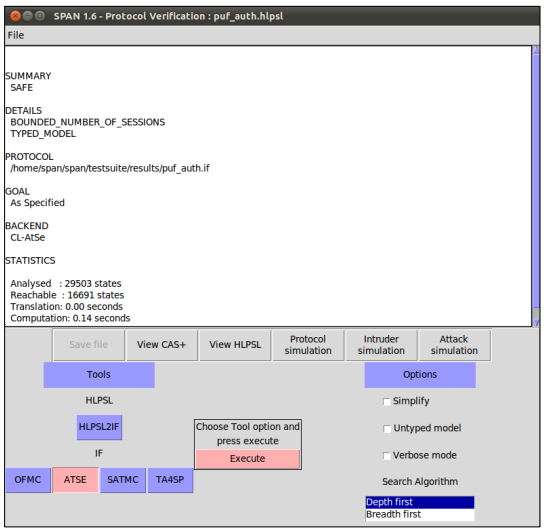

Fig. 15: Simulation result from SPAN tool for ATSE.

The HE algorithms for enrollment and device authentication are calculated the results are shown in Table 2. The authentication success rate is 1.00, which means all authentication requests were successful. The number of clients considered where 20 and multiple rounds of authentication were attempted through socket programming. The average enrollment time which includes generating and storing the encrypted challenge and response pairs is 30.45 seconds for all the 20 clients, network latency and paillier encryption may be the factors affecting the times which can be optimized further.

The average authentication time taken to complete the authentication phase for 20 clients is 44.03 seconds. The enrollment and suthentication times for single client was observed to be 6.3 seconds and 5.6 seconds. The average storage size for encrypted CRP is 96 bytes, for the 64-bit challenge and response considered in this implementation, it is a good sign for scalability. The Average communication size is 3698.9 bytes for all the client enrollment and authentication attempts done, which is again substantial, and it will be much smaller during mutual authentication.

AVISPA SPAN tool was used to evaluate protocol security. AVISPA stands for Automated Validation of Internet Security Protocols and Applications. It is an industrial-grade, push-button technology for analyzing large-scale internet protocols and applications for security [21]. Fig. 15 shows the SPAN (Security Protocol ANimator) tool interface used for verifying the current implementation. The file is written in HLPSL (High-Level Protocol Specification Language) which describes the protocol in detail. The verification tool being used is CL-AtSe (Constraint Logic-based Attack Searcher) which is a backend for analyzing security protocols, it helps find security flaws in the protocol.The protocol was classified as SAFE, meaning no vulnerabilities were detected.

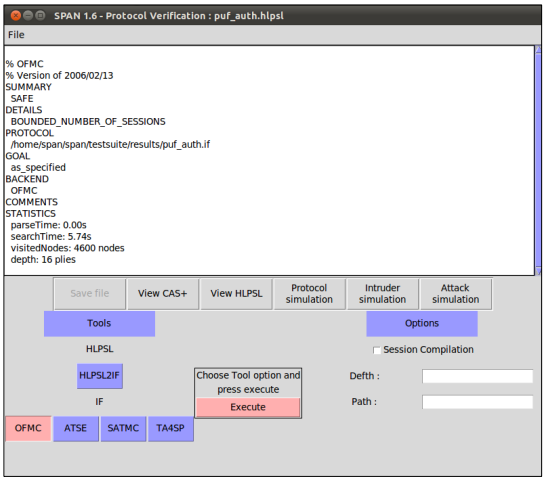

Fig. 16: Simulation result from SPAN tool for OFMC.

OFMC (On-the-fly Model Checker) shown in Fig. 16, is a backend verification tool interface, it explores possible states of the protocol by representing the actions symbolically instead of explicitly computing every possible state. This allows OFMC to detect vulnerabilities, it assumes the Dolev-Yao intruder model, a powerful adversary model. This intruder can control the network and intercept the messages but cannot break the cryptographic primitives. The result of this evaluation shows verification process is quick at 5.74 seconds, and it explored a significant number of states 4600 nodes within a manageable depth (16 plies). This suggests that the protocol is both complex and robust.

Table 3 shows the comparison of the results from state-of-the-art literature. The current scheme of the certificate-based mutual authentication process and the client-to-server verification process results show that faster speeds are achieved from the proposed scheme as compared with other SRAM PUF-based authentication systems.

9.1 Informal Security Analysis

1. Data breach: The CRPs are stored in encrypted form in the central authentication server, the EDCs participate in mutual authentication using certificates for authentication and not CRPs, hence the CRP data is more secure than storing at the local data centers. This prevents data breaches at vulnerable nodes.
2. Storage space complexity: Using certificate authority for mutual authentication reduces the need for more storage space at the nodes.
3. Eavesdropping attack: The homomorphic encryption of transmitted CRP $(\bar{C}_i, \bar{R}_i)$, ensures that the plaintext is not exposed during transmission, preserves data confidentiality, preventing unauthorized access.
4. Man-in-the-middle attacks: The gateway routers that intercept the communication verify the device using the encrypted data without exposing original data, making it difficult for the attackers to manipulate the data.
5. Impersonation attacks: The certificate authority involves data like timestamp and validity, once the validity expires even if an attacker gains access to discarded CRPs, it cannot be used to gain authorizations because to again obtain a valid certification the enrollment has to happen with the central server.
6. Physical attack: PUFs which undergo any physical manipulation will no longer be able to generate correct responses, and eventually fail in the authentication process.

10 Conclusions

The experimental results in Tables 2 and 3 demonstrate that the proposed scheme achieves a 100% authentication success rate, with an average authentication time of 5.6 seconds for a single client and 44.04 seconds for 20 clients, indicating its feasibility for real-time load-balancing scenarios. When compared with the state-of-the-art approaches, such as blockchain-based, decision-tree, and traditional PUF-based models, the proposed

Table 2: PUF-based HE Authentication System Metrics

Metric	Result (20 Clients)	Result (1 Client)
Authentication Success Rate	1.0 (100%)	1.0 (100%)
Average Enrollment Time (seconds)	30.45	6.3
Average Authentication Time (seconds)	44.03	5.6
Average Storage Size (bytes)	96	–
Average Communication Size (bytes)	3698.9	–

Table 3: Comparative Table for State-of-the-Art Literature.

Research	Algorithm	Server Authentication Time	Mutual Authentication Time
Puthal et al. [43]	Decision Tree (DT)	NA	0.6s to 0.803s
Jing et al. [56]	Blockchain, Smart Contracts	2330ms	NA
Zilong et al. [24]	SM1 and SM2	NA	NA
Barbareschi et al. [9]	PUF-based PHEMAP	NA	38.58ms
Hathal et al. [25]	TA, TESLA	NA	8600ms
Zhang et al. [63]	SRAM PUF and Blockchain	3302.9ms	991.8ms
Aarella et al. [2]	XORArbiter PUF	0.5s - 1.5s	500ms
Aarella et al. [3]	PUF-based CA	<1500ms	500ms
Current paper	Homomorphic encryption	5.6s	NA

protocol exhibits competitive authentication latency, minimal storage overhead of 96 bytes, and high scalability while preserving the confidentiality of CRP data through HE-based encryption.

The HE protocol enhances the security and privacy of CRP by storing the encrypted CRP at the authentication server, secure communication is ensured as the original CRP data is not revealed during the communication between the EDC and the server. The tests on the protocol have shown 100% success rate in multiple authentication attempts, the time for enrollment and authentication can be optimized considering the communication network. The storage space required is small and is scalable. The results from AVISPA SPAN suggest that the protocol is robust and complex in its security features.

This research is an enhancement to the work that involves SRAM PUF-based certificate generation, and the simple modification to the enrollment of devices and storage of encrypted CRP ensures data security and this model is suitable for implementing authentication schemes in CEC environment where mutual authentication is involved in real-time over different communication networks, the scheme provides the necessary security when the CRP data is transmitted through multiple network gateways in a CEC environment.

In future work, the proposed PUF–HE protocol should be rigorously evaluated against a broader range of authentication and authorization attacks, including replay, impersonation, man-in-the-middle, and side-channel attacks. Future studies will also focus on formal threat modeling and attack simulation using advanced security testing and verification tools to further validate the robustness of the system. Real-world deployment will be considered in smart village testbeds such as smart agriculture, smart healthcare, smart irrigation, and smart grid systems, to assess interoperability, scalability, and resilience across heterogeneous IoT ecosystems.

In summary, the proposed PUF–HE–based authentication framework provides a secure, privacy-preserving, and sustainable solution for next-generation collaborative edge computing. Future investigations will aim to enhance its resilience against advanced attack vectors and optimize performance for large-scale real-world implementations.

Compliance with Ethical Standards

The authors declare that they have no conflict of interest and there was no human or animal testing or participation involved in this research. All data were obtained from public domain sources.

Acknowledgments

This article is an extended version of our preliminary conference papers presented at [2] and [3].

References

1. Asymmetric Key Cryptography . pp. 109–165 (2022). DOI 10.1007/978-3-031-32959-3_4
2. Aarella, S., Mohanty, S.P., Kougianos, E., Puthal, D.: PUF Based Authentication Scheme for Edge Data Centers in Collaborative Edge Computing. In: IEEE International Symposium on Smart Electronic Systems (iSES) (IEEE-iSES). Warangal, India (2022)

3. Aarella, S.G., Mohanty, S.P., Kougianos, E., Puthal, D.: Fortified-Edge: Secure PUF Certificate Authentication Mechanism for Edge Data Centers in Collaborative Edge Computing. In: Proc. Great Lakes Symposium on VLSI, GLSVLSI '23, p. 249–254. Association for Computing Machinery, New York, NY, USA (2023). DOI 10.1145/3583781.3590249. URL https://doi.org/10.1145/3583781.3590249
4. Acar, A., Aksu, H., Uluagac, A.S., Conti, M.: A Survey on Homomorphic Encryption Schemes: Theory and Implementation. ACM Comput. Surv. **51**(4) (2018). DOI 10.1145/3214303. URL https://doi.org/10.1145/3214303
5. Aljanah, S., Zhang, N., Tay, S.W.: A Multi-factor Multi-level and Interaction based (M2I) Authentication Framework for Internet of Things (IoT) Applications. IEEE Access **PP**, 1–1 (2022). URL https://api.semanticscholar.org/CorpusID:246036035
6. Aljuhani, A., Kumar, P., Kumar, R., Jolfaei, A., Islam, A.K.M.N.: Fog Intelligence for Secure Smart Villages: Architecture, and Future Challenges. IEEE Consumer Electronics Magazine pp. 1–9 (2022). DOI 10.1109/MCE.2022.3193268
7. Ansari, M.S., Alsamhi, S.H., Qiao, Y., Ye, Y., Lee, B.: Security of Distributed Intelligence in Edge Computing: Threats and Countermeasures, pp. 95–122. Springer International Publishing, Cham (2020). DOI 10.1007/978-3-030-41110-7_6. URL https://doi.org/10.1007/978-3-030-41110-7_6
8. Assaqty, M.I.S., Gao, Y., Algarni, A.D., Khan, S., Chakraborty, M., Nur Siddiqi, H., Ahmad, S.: SPUFChain: Permissioned Blockchain Lightweight Authentication Scheme for Supply Chain Management Using PUF of IoT. IEEE Access **13**, 88662–88682 (2025). DOI 10.1109/ACCESS.2025.3566478
9. Barbareschi, M., De Benedictis, A., La Montagna, E., Mazzeo, A., Mazzocca, N.: PUF-Enabled Authentication-as-a-Service in Fog-IoT Systems. In: IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 58–63 (2019). DOI 10.1109/WETICE.2019.00020
10. Chattopadhyay, A., Lam, K.Y., Tavva, Y.: Autonomous Vehicle: Security by Design. IEEE Transactions on Intelligent Transportation Systems **22**(11), 7015–7029 (2021). DOI 10.1109/TITS.2020.3000797
11. Che, W., Martin, M., Pocklassery, G., Kajuluri, V.K., Saqib, F., Plusquellic, J.: A Privacy-Preserving, Mutual PUF-Based Authentication Protocol. Cryptography **1**(1) (2017). DOI 10.3390/cryptography1010003. URL https://www.mdpi.com/2410-387X/1/1/3
12. Cheng, G., Chen, Y., Deng, S., Gao, H., Yin, J.: A Blockchain-Based Mutual Authentication Scheme for Collaborative Edge Computing. IEEE Transactions on Computational Social Systems **9**(1), 146–158 (2022). DOI 10.1109/TCSS.2021.3056540
13. Craigen, D., Diakun-Thibault, N., Purse, R.: Defining Cybersecurity. Technology Innovation Management Review **4**, 13–21 (2014). DOI http://doi.org/10.22215/timreview/835. URL http://timreview.ca/article/835
14. Cui, L., Xie, G., Qu, Y., Gao, L., Yang, Y.: Security and Privacy in Smart Cities: Challenges and Opportunities. IEEE Access **6**, 46134–46145 (2018). DOI 10.1109/ACCESS.2018.2853985
15. Das, A., Dutta, S., Adhikari, A.: Indistinguishability against Chosen Ciphertext Verification Attack Revisited: The Complete Picture. In: W. Susilo, R. Reyhanitabar (eds.) Provable Security, pp. 104–120. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
16. Deng, X., Chen, B., Chen, X., Pei, X., Wan, S., Goudos, S.K.: A Trusted Edge Computing System Based on Intelligent Risk Detection for Smart IoT. IEEE Transactions on Industrial Informatics **20**(2), 1445–1454 (2024). DOI 10.1109/TII.2023.3245681
17. Dibas, H., Sabri, K.E.: A comprehensive performance empirical study of the symmetric algorithms:AES, 3DES, Blowfish and Twofish. In: International Conference on Information Technology (ICIT), pp. 344–349 (2021). DOI 10.1109/ICIT52682.2021.9491644
18. Farris, I., Taleb, T., Khettab, Y., Song, J.: A Survey on Emerging SDN and NFV Security Mechanisms for IoT Systems. IEEE Communications Surveys & Tutorials **21**(1), 812–837 (2019). DOI 10.1109/COMST.2018.2862350
19. Ferrag, M.A., Friha, O., Hamouda, D., Maglaras, L., Janicke, H.: Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning. IEEE Access **10**, 40281–40306 (2022). DOI 10.1109/ACCESS.2022.3165809
20. Fugkeaw, S., Changtor, A., Maneerat, T., Rattanasrisuk, P., Tangtanawirut, K.: LightPUF-IIoT: A Lightweight PUF-Based Authentication Scheme With Real-Time Detection of Rogue Devices in Fog-Assisted IIoT Data Sharing. IEEE Open Journal of the Computer Society **6**, 1438–1450 (2025). DOI 10.1109/OJCS.2025.3607984
21. Genet, T.: Span - security protocol animator and verifier (2023). URL https://people.irisa.fr/Thomas.Genet/span/. Accessed: 2023-10-30
22. Guglielmi, A.V., Crosara, L., Tomasin, S., Laurenti, N.: Physical-Layer Challenge-Response Authentication with IRS and Single-Antenna Devices pp. 560–565 (2024). DOI 10.1109/iccworkshops59551.2024.10615677
23. Haghighi, M.S., Ebrahimi, M., Garg, S., Jolfaei, A.: Intelligent Trust-Based Public-Key Management for IoT by Linking Edge Devices in a Fog Architecture. IEEE Internet of Things Journal **8**(16), 12716–12723 (2021). DOI 10.1109/JIOT.2020.3027536
24. Han, Z., Kang, C., Li, E., Wu, D.: Lightweight Encryption Authentication Technology for Interconnection of Edge Devices in PDIoT. In: IEEE 5th International Conference on Electronics Technology (ICET), pp. 1019–1025 (2022). DOI 10.1109/ICET55676.2022.9825395
25. Hathal, W., Cruickshank, H., Sun, Z., Maple, C.: Certificateless and Lightweight Authentication Scheme for Vehicular Communication Networks. IEEE Transactions on Vehicular Technology **69**(12), 16110–16125 (2020). DOI 10.1109/TVT.2020.3042431
26. Herder, C., Yu, M.D., Koushanfar, F., Devadas, S.: Physical Unclonable Functions and Applications: A Tutorial. Proceedings of the IEEE **102**(8), 1126–1141 (2014). DOI 10.1109/JPROC.2014.2320516
27. Javaid, U., Aman, M.N., Sikdar, B.: A Scalable Protocol for Driving Trust Management in Internet of Vehicles With Blockchain. IEEE Internet of Things Journal **7**(12), 11815–11829 (2020). DOI 10.1109/JIOT.2020.3002711
28. Kamarudin, N.H., Suhaimi, N.H.S., Nor Rashid, F.A., Khalid, M.N.A., Mohd Ali, F.: Exploring Authentication Paradigms in the Internet of Things: A Comprehensive Scoping Review. Symmetry **16**(2) (2024). DOI 10.3390/sym16020171. URL https://www.mdpi.com/2073-8994/16/2/171
29. Karie, N.M., Sahri, N.M., Yang, W., Valli, C., Kebande, V.R.: A Review of Security Standards and Frameworks for IoT-Based Smart Environments. IEEE Access **9**, 121975–121995 (2021). DOI 10.1109/ACCESS.2021.3109886
30. Kim, B., Yoon, S., Kang, Y., Choi, D.: PUF based IoT Device Authentication Scheme. In: International Conference on Information and Communication Technology Convergence (ICTC), pp. 1460–1462 (2019). DOI 10.1109/ICTC46691.2019.8939751
31. Kim, H., Lee, E.A.: Authentication and Authorization for the Internet of Things. IT Professional **19**(5), 27–33 (2017). DOI 10.1109/MITP.2017.3680960

32. Korenda, A.R., Afghah, F., Cambou, B., Philabaum, C.: A Proof of Concept SRAM-based Physically Unclonable Function (PUF) Key Generation Mechanism for IoT Devices. In: 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), pp. 1–8 (2019). DOI 10.1109/SAHCN.2019.8824887
33. Köylü, T., Garaffa, L., Reinbrecht, C., Zahedi, M., Hamdioui, S., Taouil, M.: Exploiting PUF Variation to Detect Fault Injection Attacks. In: 25th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), pp. 74–79 (2022). DOI 10.1109/DDECS54261.2022.9770154
34. Li, J., Cai, J., Khan, F., Rehman, A.U., Balasubramaniam, V., Sun, J., Venu, P.: A Secured Framework for SDN-Based Edge Computing in IoT-Enabled Healthcare System. IEEE Access **8**, 135479–135490 (2020). DOI 10.1109/ACCESS.2020.3011503
35. Mazzo, F., Tomasin, S., Zhang, H., Chorti, A., Poor, H.V.: Physical-Layer Challenge-Response Authentication for Drone Networks pp. 3282–3287 (2023). DOI 10.1109/globecom54140.2023.10436823
36. Mohanty, S.P.: Security and Privacy by Design is Key in the Internet of Everything (IoE) Era. IEEE Consumer Electronics Magazine **9**(2), 4–5 (2020). DOI 10.1109/MCE.2019.2954959
37. Mohanty, S.P., Plusquellic, J., Rose, G.S., Zhang, W., Michael, M.K.: Introduction to the Special Issue on Hardware-Assisted Security for Emerging Internet of Things. J. Emerg. Technol. Comput. Syst. **18**(1) (2021). DOI 10.1145/3475952. URL https://doi.org/10.1145/3475952
38. Mollakuqe, E., Parduzi, A., Rexhepi, S., Dimitrova, V., Jakupi, S., Muharremi, R., Hamiti, M., Qarkaxhija, J.: Applications of Homomorphic Encryption in Secure Computation . Open research Europe **4**, 158–158 (2024). DOI 10.12688/openreseurope.18052.1
39. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Proc. 17th International Conference on Theory and Application of Cryptographic Techniques, EUROCRYPT'99, p. 223–238. Springer-Verlag, Berlin, Heidelberg (1999)
40. Perlman, R.: An overview of PKI trust models. IEEE Network **13**(6), 38–43 (1999). DOI 10.1109/65.806987
41. Pescador, F., Mohanty, S.P.: Guest Editorial Security-by-Design for Electronic Systems. IEEE Transactions on Consumer Electronics **68**(1), 2–4 (2022). DOI 10.1109/TCE.2022.3147005
42. Prasad, K., Shah, N., Dagli, J., Mekie, J.: SDR-PUF: Sequence-Dependent Reconfigurable SRAM PUF with an Exponential CRP Space. In: 37th International Conference on VLSI Design and 2024 23rd International Conference on Embedded Systems (VLSID), pp. 535–540 (2024). DOI 10.1109/VLSID60093.2024.00095
43. Puthal, D., Damiani, E., Mohanty, S.P.: Secure and Scalable Collaborative Edge Computing using Decision Tree. In: IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 247–252 (2022). DOI 10.1109/ISVLSI54635.2022.00055
44. Puthal, D., Mohanty, S.P., Wilson, S., Choppali, U.: Collaborative Edge Computing for Smart Villages [Energy and Security]. IEEE Consumer Electronics Magazine **10**(3), 68–71 (2021). DOI 10.1109/MCE.2021.3051813
45. Puthal, D., Obaidat, M.S., Nanda, P., Prasad, M., Mohanty, S.P., Zomaya, A.Y.: Secure and Sustainable Load Balancing of Edge Data Centers in Fog Computing. IEEE Communications Magazine **56**(5), 60–65 (2018). DOI 10.1109/MCOM.2018.1700795
46. Ram, S.K., Das, B.B., Mahapatra, K., Mohanty, S.P., Choppali, U.: Energy Perspectives in IoT Driven Smart Villages and Smart Cities. IEEE Consumer Electronics Magazine **10**(3), 19–28 (2021). DOI 10.1109/MCE.2020.3023293
47. Rohan, R., Pal, D., Watanapa, B., Funilkul, S.: Emerging Paradigm of IoT Enabled Smart Villages. In: IEEE International Conference on Consumer Electronics (ICCE), pp. 1–6 (2022). DOI 10.1109/ICCE53296.2022.9730482
48. Rostampour, S., Bagheri, N., Bendavid, Y., Safkhani, M., Kumari, S., Rodrigues, J.J.P.C.: An Authentication Protocol for Next Generation of Constrained IoT Systems. IEEE Internet of Things Journal **9**(21), 21493–21504 (2022). DOI 10.1109/JIOT.2022.3184293
49. Rullo, A., Felicetti, C., Vatalaro, M., De Rose, R., Lanuzza, M., Crupi, F., Saccà, D.: PUF-Based Authentication-Oriented Architecture for Identification Tags. IEEE Transactions on Dependable and Secure Computing **22**(1), 66–83 (2025). DOI 10.1109/TDSC.2024.3387568
50. Ryu, J., Kim, K., Won, D.: A Study on Partially Homomorphic Encryption. In: 17th International Conference on Ubiquitous Information Management and Communication (IMCOM), pp. 1–4 (2023). DOI 10.1109/IMCOM56909.2023.10035630
51. Schaub, A., Danger, J.L., Guilley, S., Rioul, O.: An Improved Analysis of Reliability and Entropy for Delay PUFs. In: 21st Euromicro Conference on Digital System Design (DSD), pp. 553–560 (2018). DOI 10.1109/DSD.2018.00096
52. Sendhil, R., Amuthan, A.: A Comparative Study on security breach in Fog computing and its impact. In: International Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 247–251 (2020). DOI 10.1109/ICESC48915.2020.9155967
53. Shah, T., Venkatesan, S.: Authentication of IoT Device and IoT Server Using Secure Vaults. In: 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp. 819–824 (2018). DOI 10.1109/TrustCom/BigDataSE.2018.00117
54. Suo, H., Wan, J., Zou, C., Liu, J.: Security in the Internet of Things: A Review. In: International Conference on Computer Science and Electronics Engineering, vol. 3, pp. 648–651 (2012). DOI 10.1109/ICCSEE.2012.373
55. Tun, N.W., Mambo, M.: Secure PUF-Based Authentication Systems. Sensors **24**(16) (2024). DOI 10.3390/s24165295. URL https://www.mdpi.com/1424-8220/24/16/5295
56. Wang, J., Wu, L., Choo, K.K.R., He, D.: Blockchain-Based Anonymous Authentication With Key Management for Smart Grid Edge Computing Infrastructure. IEEE Transactions on Industrial Informatics **16**(3), 1984–1992 (2020). DOI 10.1109/TII.2019.2936278
57. Xiao, L., Cai, J., Qiu, M., Liu, M.: A Secure Identity Authentication Protocol for Edge Data in Smart Grid Environment. In: 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), pp. 188–193 (2021). DOI 10.1109/CSCloud-EdgeCom52276.2021.00042
58. Xu, T., Wendt, J.B., Potkonjak, M.: Security of IoT systems: Design challenges and opportunities. In: IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 417–423 (2014). DOI 10.1109/ICCAD.2014.7001385
59. Yang, A., Weng, J., Yang, K., Huang, C., Shen, X.: Delegating Authentication to Edge: A Decentralized Authentication Architecture for Vehicular Networks. IEEE Transactions on Intelligent Transportation Systems **23**(2), 1284–1298 (2022). DOI 10.1109/TITS.2020.3024000
60. Yang, J., Liu, J., Song, H., Liu, J., Lei, X.: Blockchain-based Conditional Privacy-Preserving Authentication Protocol with Implicit Certificates for Vehicular Edge Computing. In: 7th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), pp. 210–216 (2022). DOI 10.1109/ICCCBDA55098.2022.9778897

61. Yoon, S., Kim, B., Kang, Y.: Multiple PUF-based lightweight authentication method in the IoT. In: International Conference on Information and Communication Technology Convergence (ICTC), pp. 1198–1200 (2021). DOI 10.1109/ICTC52510.2021.9620972
62. Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., Kong, J., Jue, J.P.: All one needs to know about fog computing and related edge computing paradigms: A complete survey. Journal of Systems Architecture **98**, 289–330 (2019). DOI https://doi.org/10.1016/j.sysarc.2019.02.009
63. Zhang, Y., Li, B., Liu, B., Hu, Y., Zheng, H.: A Privacy-Aware PUFs-Based Multiserver Authentication Protocol in Cloud-Edge IoT Systems Using Blockchain. IEEE Internet of Things Journal **8**(18), 13958–13974 (2021). DOI 10.1109/JIOT.2021.3068410
64. Zhang, Z.K., Cho, M.C.Y., Wang, C.W., Hsu, C.W., Chen, C.K., Shieh, S.: IoT Security: Ongoing Challenges and Research Opportunities. In: IEEE 7th International Conference on Service-Oriented Computing and Applications, pp. 230–234 (2014). DOI 10.1109/SOCA.2014.58