IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 April 2024, accepted 6 June 2024, date of publication 26 June 2024, date of current version 8 July 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3419104*

RESEARCH ARTICLE

# DPUAUT: Secure Authentication Protocol With SmartNiC Integration for Trustworthy Communications in Intelligent Swarm Systems

**RANA ABU BAKAR[1], FRANCESCO PAOLUCCI[2], FILIPPO CUGINI[2], (Member, IEEE), PIERO CASTOLDI[1], (Senior Member, IEEE), AND JUAN JOSE VEGAS OLMOS[3]**

[1]Scuola Superiore Sant'Anna, 56124 Pisa, Italy
[2]CNIT, 56124 Pisa, Italy
[3]NVIDIA, 4000 Roskilde, Denmark

Corresponding author: Rana Abu Bakar (rana.abubakar@santannapisa.it)

This work was supported in part by European Union's Horizon Research and Innovation Programme (RIA) (SMARTEDGE) under Grant 101092908; and in part by the Department of Excellence in Robotics and Artificial Intelligence, Scuola Superiore Sant'Anna.

**ABSTRACT** Swarm devices are becoming increasingly prevalent in next-generation networks because they can efficiently handle large-scale distributed computing tasks. The traditional cloud-based frameworks are not suitable for swarm devices due to their low latency and scalability requirements. Moreover, these frameworks are vulnerable to security threats when communicating over unsecured networks. Edge computing addresses these challenges, but it suffers from several security issues, such as device authentication, access control, and privacy issues for swarm devices. Many authentication protocols have been proposed to protect from these attacks, but most of them are vulnerable to physical attacks due to the keys stored in device memory. Physical Unclonable Function (PUF)s can significantly enhance the physical security of these devices by generating a unique identifier for each device, making it difficult for attackers to physically clone or impersonate these devices. In this paper, we propose a provably secure authentication protocol using a Bluefield Data Processing Unit (DPU), enabling one-round mutual authentication while preserving anonymity and preventing physical attacks using PUF. We conduct security analyses comprehensively, which provide evidence of its strong resilience against attacks. Experimental evaluation confirms its dominance over existing solutions. The DPUAUT swarm authentication protocol has a low computational overhead of 6.161$ms$, and the communication overhead is 1052$Bits$.

**INDEX TERMS** Secure network, authentication, intelligent swarm, SmartNiC, data processing unit (DPU), physical unclonable functions (PUF).

## I. INTRODUCTION

The rise of swarm intelligence-based autonomous systems in next-generation networks has brought significant advancements to transportation. These systems effectively handle various tasks, from scheduling to real-time navigation and tracking. However, the heavy computational and communication load can limit their full potential [1]. To address these challenges, edge-centric systems have emerged as a viable solution. These systems mitigate load management issues, communication delays, and traffic overhead by deploying edge-centric servers and edge micro data centers as private cloud servers closer to swarm devices by using a DPU at both ends (e.g., Bluefield SmartNiC) [2]. Local data processing at the DPU improves latency and optimizes bandwidth utilization. However, the proximity

The associate editor coordinating the review of this manuscript and approving it for publication was Adnan M. Abu-Mahfouz.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

of private cloud servers introduces new security concerns for swarm intelligence-based autonomous systems [3], [4]. Adversaries can exploit wireless mediums to intercept, eavesdrop, tamper with, delete, or replay information, jeopardizing device security and privacy [5]. An active attacker who gains access to an open network and location can easily extract details about the device's location, messages exchanged, and time spent at the location [6]. Consequently, designing a secure and efficient authentication protocol becomes crucial to protect devices in these systems [7]. Edge-centric computing encompasses proximity, intelligence, trust, and management control by humans or machines on the edge [8]. Edge computing extends cloud-based services to the network's edge and enhances various services in edge-centric Autonomous Swarm Devices (ASD)s systems. From this viewpoint, the incorporation of privacy-preserving reputation updating (PPRU) into edge computing serves as an illustration and improves the overall effectiveness of the network [9]. Several authors have provided a comprehensive overview of security challenges, security attacks, security concerns, and potential solutions in edge computing systems [10], [11], [12], [13], [14], [15], [16], [17], [18], [19].

PUFs have been considered essential for ensuring physical security in authentication protocols. However, recent developments, such as the Xilinx-FPGA-based PUF proposed by [20], have aimed to address this need. Despite this effort, vulnerabilities in hardware-based PUFs have been identified, as highlighted by [21]. We summarized the limitations of existing state-of-the-art authentication protocols,s in Table 1. These protocols utilize various cryptographic techniques such as identity-based schemes, ECC, lattice-based encryption, and PUF in the field of edge computing. The comparison reveals the trade-offs between computational costs, resilience to physical attacks, and the security features offered by each protocol.

### A. MOTIVATIONS

Due to the capacity to handle large and distributed tasks, the swarm devices are on their way to becoming part and parcel of next-generation networks. However, owing to low latency and scalability issues, the cloud-based frameworks are not suitable for swarm devices. Edge computing can resolve these issues, but edge computing may also introduce new security threats, including vulnerability against physical attacks, weakness against forgery attacks, and tempering of the hardware. In recent times, some authentication protocols based on PUFs for edge computing-based swarm devices were proposed. However, autonomous Intelligent Swarm System (ISS) require secure and efficient authentication schemes to ensure the safety and reliability of their operations. Existing authentication schemes for edge computing systems are often insufficient in terms of security, efficiency, and attack resistance. For example, they may need to verify the physical identity of swarm devices in different environments, such as smart cities, factories, and autonomous vehicle networks. This emphasizes the ongoing challenge of achieving robust physical security in authentication systems. Therefore, this article presents a novel authentication scheme for autonomous ISS to address the mentioned limitations of existing and related schemes.

### B. CONTRIBUTION BEYOND STATE-OF-THE-ART

The research contributions of this work are as follows:

- We propose a lightweight authentication scheme that enables secure communication among devices in autonomous ISS. Our scheme incorporates PUFs to enhance security by preventing physical attacks.
- To improve the performance of the authentication protocol, we introduce a key enhancement by offloading the PUF-verifier logic from a general-purpose CPU to a Smart Network Interface Card (SmartNiC)-based DPU. This hardware acceleration approach reduces authentication latency, enhances scalability, and strengthens security.
- Our proposed scheme undergoes informal security analysis to verify its robustness against several attacks. Additionally, we conduct formal security validation and research using the Real-or-Random (RoR) formal model to assess its overall security.
- We compare the security attributes of our proposed authentication scheme with other pertinent schemes, showcasing its efficacy in delivering robust security measures.
- We have simulated the proposed scheme using the popular automated toll AVISPA to validate the automated security of the proposed scheme.
- We have performed a comparative analysis with some related schemes as per the computation, communication, and security feature provision.

Our research introduces a secure and efficient authentication scheme tailored to autonomous ISS application. Through extensive analysis and evaluation, we validate its security features and demonstrate its superior performance compared to existing schemes. Our authentication scheme provides a comprehensive solution that meets all security requirements of autonomous ISS.

### C. PAPER ORGANIZATION

The remaining article is organized as follows: In Section II, A description of the state-of-the-art authentication protocols and analysis is provided. The Section III contains the System Model and Notations. The Section IV covers the proposed protocol. The Section V provides a security analysis with simulation, formal, and informal analysis. The Section VI presents a detailed performance evaluation and analysis of security features. Finally, in the Section VII, conclusions and future research directions are provided.

## II. RELATED WORK

Various cryptographic authentication techniques have been designed for edge computing systems. However, these

**TABLE 1.** State-of-the-art authentication protocols comparison.

| State-of-the-art Protocols | Year | Methods | Limitations | Strengths |
|---|---|---|---|---|
| [14] | 2020 | Identity-based AKE scheme and One-way hash function | Lack of physical attack resilience | Ensures mutual authentication, User Anonymity |
| [22] | 2022 | ECC, One-way hash | High Computational Cost | Resilient to Physical Attack |
| [23] | 2022 | Lattice-based Encryption and Signature Algorithms | High computational cost | Provides key agreement and location privacy |
| [20] | 2023 | PUF, XOR, HASH | Impersonation attacks and key leakage attacks | Mutual authentication, FPGA-based PUF |
| [24] | 2023 | One way hash, ECC | Lack of physical attack resilient at device | Provides physical security and privacy |
| [25] | 2023 | PUF, ECC | High computation and lake of anonymity | Provides key agreement and privacy with physical security |
| [26] | 2023 | PUF, Fuzzy extractor | Vulnerable to modeling attacks | Mutual authentication, Silicon-Based PuF |

schemes indicate various limitations, such as a lack of security features and computational and communication overhead issues.

For example, Irshad et al. [27] proposed a key negotiation scheme that lacks a user revocation mechanism reported by [28]. Xiong et al. [28] proposed an enhanced key establishment scheme, which is vulnerable to impersonation and anonymity violation attacks. Jia et al. [11] designed a bilinear-based protocol susceptible to security attacks. Agilandeeswari et al. [6] proposed a scheme that uses XOR operations for key agreement and authentication in the communication between vehicles and smart grids in a social IoT environment. They also presented formal and informal analyses to prove their scheme is secure from some well-known attacks. Yang et al. [29] proposed an inefficient key agreement scheme for resource-constrained swarm devices based on Elliptic Curve Cryptography (ECC). The protocol in question involves assigning a set of pseudo-IDs and a corresponding family of secret keys to each user. This is accomplished through an Access Service Network Gateway (ASN-GW), which executes a key pre-distribution process based on elliptic curve cryptography. Unlike other schemes, no bilinear pairing is required. However, the ASN-GW must generate many pseudo-IDs for each registered user, and any mobile user must store many pseudo-IDs and their corresponding secret keys. As a result, it is not practical for mobile devices with limited storage capacity. Irshad et al. [27] proposed a protocol with a key negotiation for edge computing systems designed to be secure and efficient. The protocol uses a combination of symmetric and public key cryptography to negotiate a shared key between two devices. The protocol also includes a user revocation mechanism, allowing devices to revoke their key if it is compromised. However, the protocol lacks a user revocation mechanism. If a device's key is compromised, the attacker could still use it to impersonate the user and access their data. Xiong et al. [28] reported that Irshad et al. protocol [27] is not secure. Specifically, they claim the protocol cannot accommodate a secure user revocation. In response to this challenge, they introduced an improved key establishment scheme for swarm-based edge computing systems. They claimed that their enhanced scheme offers efficiency and resilience against potential attacks. Jia et al. [11] presented a protocol that utilizes bilinear pairing operations. Their claims regarding the scheme emphasized its robustness and efficiency, with purported attributes including perfect forward secrecy and user privacy. Li et al. [30] observed that Jia et al. protocols do not deliver the security features they initially claimed, particularly regarding resistance against impersonation and preserving user anonymity. They also suffer from ephemeral key disclosure attacks. Li et al. [31] introduced a key agreement technique based on ECC for mobile devices in edge computing systems. The scheme's primary objectives include the provision of user anonymity and perfect forward secrecy. However, vulnerabilities have been identified within the protocol by Jia et al. [11], notably concerning susceptibility to impersonation and privacy breaches. In another study, Castello et al. [32] proposed a novel authentication protocol for swarm-based intelligent systems using blockchain technology. While the paper highlights the potential benefits of blockchain in achieving secure and consensus-driven operations, it falls short in providing a critical evaluation of the limitations and challenges. The scalability issues of integrating blockchain into swarm robotics are not discussed. It should also address the computational requirements and potential performance trade-offs associated with blockchain implementation. Moreover, the paper did not explore the practical feasibility and cost-effectiveness of implementing blockchain in real-world swarm robotics applications. In another study, Ogundoyin [33] presented a certificate authentication scheme for edge computing systems. The scheme aims to provide secure communication while ensuring user anonymity and resistance against attacks. The authors claimed their proposed scheme is more efficient than other certificate authentication schemes. The scheme they proposed for secret key generation suffers from a flaw where the primary secret key of the private key generator (PKG) can be leaked through their key generation method.

Nandy et al. [34] proposed a secure authentication protocol for vehicular ad hoc networks (VANETs) in edge computing systems. The protocol is based on the combination of symmetric and asymmetric key cryptography to provide mutual authentication, confidentiality, and integrity. The authors have compared and analyzed the performance of the proposed authentication scheme with the existing authentication protocols.

In another study, Lin et al. [35] proposed a lightweight authentication protocol for autonomous vehicles based on blockchain technology. The protocol provides secure authentication and efficient data exchange between autonomous vehicles and their surrounding environment. They also provide a detailed analysis of schemes based on blockchain-based authentication solutions.

In a different approach, Miao et al. [36] proposed an enhanced authentication protocol for autonomous vehicles based on ECC. They claim that protocol provides secure and efficient authentication and key agreement between vehicles and their surrounding environment. According to a comprehensive study by Kumar et al. [37], there are several types of security threats that these systems are vulnerable to, including replay attacks, session key security attacks, physical attacks, man-in-the-middle attacks, eavesdropping attacks, lack of forward and backward secrecy, impersonation attacks, non-synchronous attacks, and lack of anonymity. Alladi et al. [38] proposed authentication scheme focuses on achieving lightweight and efficient authentication for resource-constrained vehicles. This is critical in vehicular networks where devices may have the limited processing power and energy constraints. The scheme aims to minimize computational overhead and communication costs while emphasizing lightweight design while maintaining high security. Chaudhry et al. also proposed two solutions, including a bilinear pairing-based authentication protocol for devices in distributed IoT environments [39] and device-to-device access control using Elliptic Curve Cryptography (ECC). Due to the high computational overhead of the pairing-based solution and due to the storage of certificated on SmartNiC memory, both protocols were deemed impractical for securing the communication among Swarm devices.

Liu et al. [9] have introduced a new method called Privacy-Preserving Reputation Updating (PPRU) for cloud-assisted vehicular networks. This method ensures privacy by using the ECC and Paillier algorithms. However, the Cloud Service Provider (CSP), which is honest-but-curious (HBC), can still collect and pre-process the reputation feedback in this privacy-preserving system. Which can help significantly minimize overheads on the TA side. Liu et al. [40] proposed a Lightweight Trustworthy Message Exchange (LTME) scheme that offers rich functionality but faces computation limitations due to resource constraints on UAVs. To overcome computational issues, they also presented a simple scheme called the sLTME scheme that provides similar functionality to LTME but with lower communication overhead, potentially reducing the impact on UAV network performance. The sLTME had lower TGEN values than LTME, indicating better performance, but lacked confidentiality when exchanging messages. Liu et al. [41] proposed a Privacy-Preserving Trust Management (PPTM) scheme to address the challenges associated with distributing emergency messages in Space-Air-Ground-Integrated Vehicular Networks (SAGIVN) situations. This system is reliable in this use case, widely applicable, and has many positive qualities. The PPTM scheme enables robust conditional privacy preservation and accurate trust management with minimal communication overhead. To confirm the effectiveness of the PPTM system, the authors conducted extensive theoretical analysis and simulated tests.

In a recent study, Abdel et al. [42] proposed an efficient authentication protocol for the swarm-based intelligent system for Unmanned Ariel Vehicle (UAV). The paper presents a lightweight proxy signature-based authentication mechanism for 5G Drone to Drone (D2D) communication in drone swarms. However, it lacks a detailed analysis of security vulnerabilities and does not compare its performance with existing mechanisms. The potential computational overhead and implications of leader transitions are not adequately addressed. Further research is needed to evaluate the security robustness, conduct comparative assessments, and consider leader transition's computational impact and dynamics within the swarm. The proposed mechanism is implemented in an NS-3 simulation environment and tested on a Raspberry Pi 3 device; a more thorough evaluation against other authentication schemes would provide a better understanding of its performance, scalability, and reliability in realistic scenarios.

The literature review shows that various authentication protocols and schemes have been proposed for edge computing systems and ISS in recent years. These protocols and schemes utilize cryptographic methods and tools, such as biometric authentication, blockchain, and hardware-based authentication, to ensure secure communication between devices in the swarm system. However, these existing techniques need to improve both security and efficiency. Hence, developing lightweight, secure key agreements and efficient techniques tailored for edge computing systems is paramount. However, additional research is imperative to advance the development of lightweight, efficient, and secure authentication protocols that can effectively cater to the distinctive challenges and requisites of autonomous ISS.

## III. SYSTEM MODELS

This section overviews the fundamental concepts and preliminaries for understanding the proposed authentication scheme. Details and threat model including the adversarial model, one-way hash function, SmartNiC-based PUF, and important notation utilized throughout the paper described in Table 3.

**TABLE 2.** Challenge and response table for SmartNiC-based PUF authentication protocol.

| Challenge | Response |
|---|---|
| $\mathcal{CH}_1$ | $\mathcal{R}_1 = \mathcal{P}(\mathcal{C}_1)$ |
| $\mathcal{CH}_2$ | $\mathcal{R}_2 = \mathcal{P}(\mathcal{C}_2)$ |
| $\mathcal{CH}_3$ | $\mathcal{R}_3 = \mathcal{P}(\mathcal{C}_3)$ |
| ⋮ | ⋮ |

### A. PHYSICAL UNCLONABLE FUNCTION (PUF)

A physical unclonable function is a hardware security feature that generates a unique identifier based on the physical characteristics of a device. PUFs exploit the inherent variations in manufacturing processes to create device-specific identification. These variations make it extremely difficult for attackers to clone the device or tamper with its security features. This paper proposes a new authentication protocol for swarm devices that uses a PUF to generate a unique identifier for each device. The PUF must resist physical attacks, such as side-channel attacks and tampering attacks, to prevent adversaries from extracting the PUF's response to a given challenge. Additionally, the PUF must be small and low-power, as swarm devices are often resource-constrained [43].

#### 1) PUF IMPLEMENTATIONS

A PUF is a specialized one-way function implemented within the SmartNiC BlueField-2 hardware. Our PUF implementation aligns with the guidelines set forth by Aitchison et al. [44] in their paper on integrating PUFs into ARM TrustZone Security Technology.

Below are the requirements for the PUF:

- Resistance to attacks: The PUF should resist physical attacks and machine learning attacks, which attempt to learn the PUF's response to a given challenge by observing its responses to many other challenges.
- Ease of implementation: The PUF should be easy to implement in a SmartNiC BlueField 2, with minimal impact on the performance and cost of the device.
- Improve Latency: The PUF should contribute to the reduction of latency in PUF-based swarm devices.

Key functionalities of the DPU-based PUF include:

- The DPU-based PUF uses many stages (e.g., 1000 or more).
- The DPU-based PUF is implemented in the SoC using logic synthesis and physical layout.

The SmartNiC-based PUF operates by generating a unique set of response messages, denoted as $\mathcal{R}i$, in response to a given location of challenge messages, represented as $\mathcal{CH}i$. The physical composition and dimensions of the hardware component vary, resulting in unique responses from the PUF function $\mathcal{P}_i$.

To ensure PUF security, the Hamming distance [45] between the responses of any two PUF functions $\mathcal{P}i1(.)$ and $\mathcal{P}i2(.)$, given challenges $\mathcal{CH}_{i1} \in 0, 1^k$, must exceed a threshold value $b$. In the given scenario, the variable $b$ represents the distance between the outputs of two given functions, while $\lambda$ signifies the PUF-based function output length. The symbol $\mathcal{H}_\mathcal{D}$ denotes the Hamming distance. The PUF integrated within the SmartNiC Bluefield-2 provides a secure mechanism to generate and compare these response messages, ensuring the uniqueness and reliability of the authentication process.

The PUF is used in the authentication protocol to generate a unique identifier for each swarm device. This identifier is then used to authenticate the device to the PCS. The authentication protocol is designed to resist various attacks, including physical attacks, Man-in-the-Middle (MiTM) attacks, replay attacks, and impersonation attacks. Table 2 illustrates a PUF-based example of challenge and response (challenge = CH and response = R).

Where:

- $\mathcal{CH}_i$ is a challenge generated by the PCS
- $\mathcal{P}$ is the PUF function
- $\mathcal{R}_i$ is the response generated by the ASDui

**TABLE 3.** Notations.

| Notations | Description |
|---|---|
| $\mathcal{ASDCS}$ | Autonomous Swarm Devices Computing Server |
| $\mathcal{MSK}$ | Master secret key of $\mathcal{ASDCS}$ |
| $\mathcal{ES}_j$ | $j$th Private Cloud Server (PCS) |
| $SID_j$ | Pseudonym of $\mathcal{ES}_j$ |
| $\mathcal{K}_j$ | Secret key of $\mathcal{ES}_j$ |
| $\mathcal{ASD}_i$ | $i$th Autonomous Swarm Devices |
| $ID_i$ | Unique identity of $\mathcal{ASD}_i$ |
| $\sigma_i$ | PCS Versifier |
| $TID_i$ | Temporary identity of $\mathcal{ASD}_i$ |
| $\mathcal{SN}_i$ | Sensor node of $\mathcal{ASD}_i$ |
| $i$ | Identity |
| $\mathcal{CH}_i, \mathcal{R}_i$ | Challenge-Response pair of $\mathcal{ASD}_i$ |
| $\mathcal{P}_i$ | Physical unclonable functions |
| $\mathcal{A}$ | Attacker |
| $\mathcal{ASD}$ | Autonomous Swarm Devices |
| $\mathcal{TR}_i$ | Tamper Resilient |
| $\mathcal{ASD}\int\|\|$ | Authenticated ASD |
| $G0, G1, G2, G3$ | Game 0, Game 1, Game 2, Game 3 |

### B. NOTATION

Throughout the paper, a specific notation will represent various elements, algorithms, and cryptographic primitives used in our authentication protocol presented in Table 3.

By familiarizing ourselves with these critical concepts and preliminaries, we can establish a solid foundation for understanding the subsequent sections that delve into the details of the proposed authentication scheme.

### C. SYSTEM SETUP

We are considering an ISS that consists of $\mathcal{ASD}$ devices, a Private Cloud Server (PCS), and a Autonomous Swarm Devices Computing Server (ASDCS) as described in 1. Figure 1 and Figure 2 illustrate the detailed architecture and design of the authentication-based ISS. Each ASD is equipped with a sensor node that collects data. The

ISS design is presented in our previous work [46]. In the proposed swarm authentication system, a trustworthy PCS is responsible for registration and provides the necessary information for authentication. A ASDCS positioned near the data source facilitates ASD in sharing pre-processed data with the nearest ASDCS, such as in a swarm area. This approach efficiently offloads computation tasks, reduces network bandwidth consumption, and alleviates pressure on centralized authentication systems. The collaborative efforts between the ASD and ASDCS enhance services' overall convenience and efficiency. In the registration process, the PCS employs its secret key to generate a secret key between two communicating entities. This ensures secure and authenticated communication between the entities. Registering requests through a private channel facilitates exchange between the PCS and the other two entities.

During the authentication process, mutual authentication occurs between the swarm devices and ASDCS, establishing a session key for secure communication over a public channel. One scenario is taken into consideration during the development of our protocol. In this scenario, the authentication of a new device is seeking to join the swarm. In this context, the new device's authentication process must prioritize latency considerations and the secure sharing of a security key. Following successful authentication, this security key becomes instrumental in encrypting messages exchanged among devices within the swarm. To address the demanding requirements of secure and efficient authentication in ISS, We use PUF to provide secure and efficient authentication in swarm systems. PUFs offer unpredictable electrical characteristics for generating challenges and responses. They are lightweight, energy-efficient, and can be easily scaled for a large number of devices.

In the next subsection, we will discuss offloading PUF-verifier logic to a SmartNiC-based DPU to improve authentication performance, scalability, and security. We reduce authentication latency and enhance the protocol's performance by leveraging DPU's computational power. The DPU's resources can be shared among multiple swarm devices, ensuring efficient authentication in large-scale deployments. We enhance the security of the authentication protocol by isolating the PUF verification process on the DPU.

### D. ACCELERATION OF AUTHENTICATION PROTOCOL

To enhance the performance and efficiency of the authentication protocol, we propose offloading the PUF-verifier logic from a general-purpose CPU to a SmartNiC. This offloading approach aims to achieve the following objectives: to reduce authentication latency while improving authentication scalability and intrinsic security. By offloading the PUF-verifier logic to the SmartNiC DPU, we intend to significantly reduce the time required for authenticating swarm devices before initiating data transfer to a cloud server. This reduction in authentication latency improves the system's overall responsiveness.

Scalability: We aim to support authentication for a large-scale swarm of devices, accommodating tens of thousands of machines while adhering to the memory and processing constraints imposed by the SmartNiC DPU. This ensures that the authentication protocol can scale efficiently to handle the growing number of swarm devices.

Security: Ensuring the safety and robustness of communication between swarm devices and the PUF-verifier running on the DPU is paramount. We address this challenge by implementing security measures that protect against various attacks. It is worth noting that the limited hash-based primitives and per-packet operations allowed by the SmartNiC DPU pose additional challenges in achieving robust security.

By leveraging the capabilities of the SmartNiC DPU and offloading the PUF-verifier logic, we aim to significantly improve authentication performance, scalability, and security for ISS.

### E. A ONE WAY HASH FUNCTION

The one-way and randomized secure hash algorithm, denoted as $h : (x \rightarrow y)$, possesses several important characteristics that make it a fundamental tool in modern cryptography. These characteristics are as follows:

- Message Digest Generation: The hash function takes an input message $x$ of any length and produces a fixed-length message digest $y$ as output. The output size remains constant, regardless of the input message's length.
- Computational Infeasibility of Inverse Computation: Given a hash value $y$, it is considered practically impossible to compute the inverse function $h^{-1}(y)$, which would reveal the original input message $x$. The one way property makes it extremely challenging to determine the input message from its hash value.
- Prevention of Second Preimage Attacks: For a given input message $x$, it is not feasible to find any other distinct input message $x'$ such that $h(x') = h(x)$. In other words, finding a different message that produces the same hash value is computationally infeasible. This property safeguards against the creation of additional messages with identical hash values.
- Collision Resistance: It is computationally infeasible to find any two distinct input messages $msg_1$ and $msg_2$ where $msg_1 \neq msg_2$, but their hash values collide, i.e., $h(msg_1) = h(msg_2)$. Such a pair of inputs with the same hash value is called a hash collision. A good hash function exhibits strong collision resistance, making it highly unlikely to find collisions even with significant computational resources.

  One way hash functions find wide application in various cryptographic protocols and systems is password hashing, digital signatures, integrity verification, and data authentication. They provide essential security

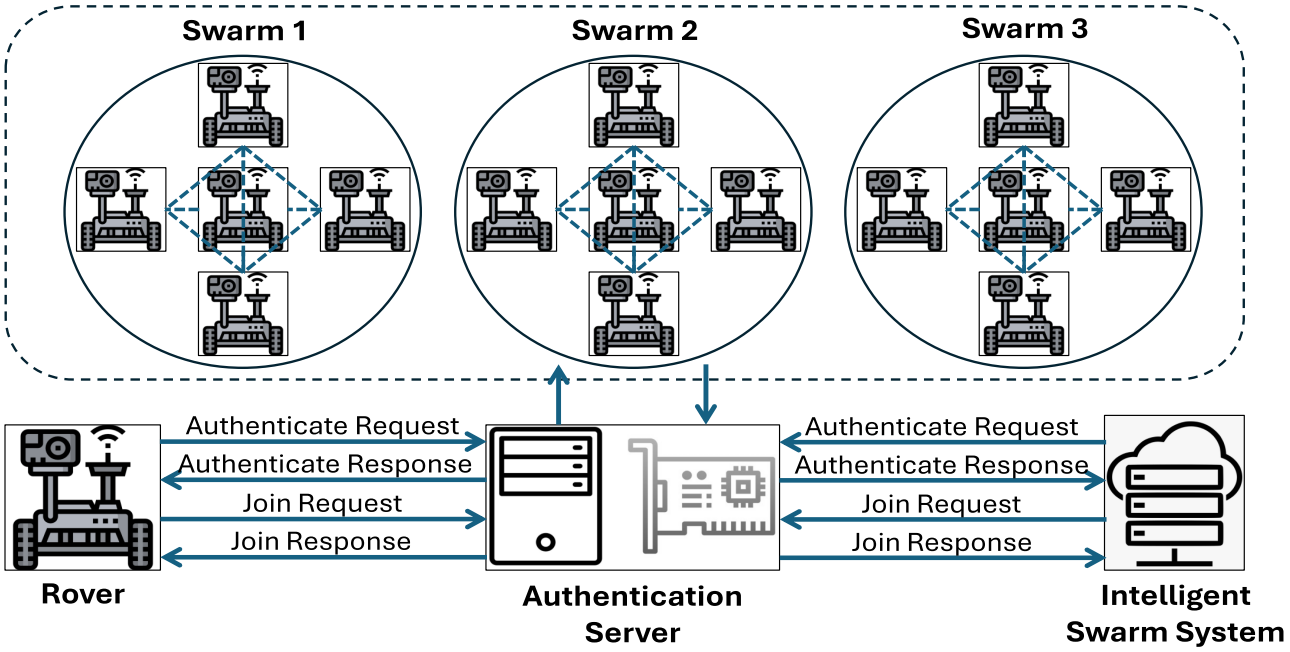



**FIGURE 1.** Autonomous swarm intelligence system.

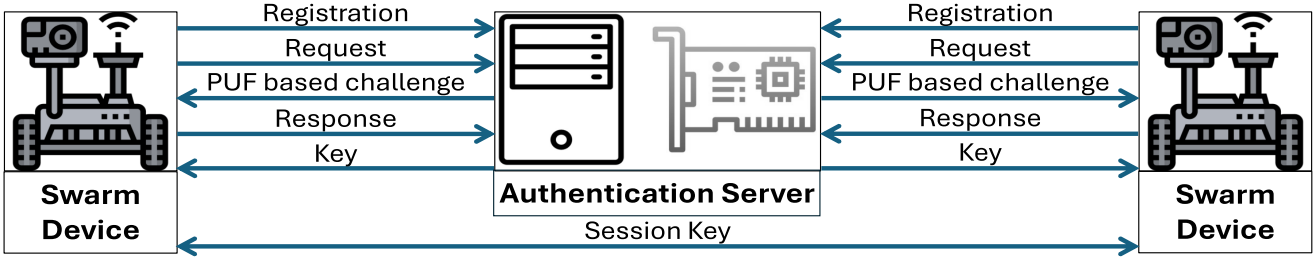



**FIGURE 2.** SmartNiC based PUF authentication protocol.

properties, such as data integrity, non-repudiation, and confidentiality, thereby ensuring the reliability and trustworthiness of digital information.

### F. THREAT MODEL

As per literature, the widely adopted "Canetti and Krawczyk's adversary model [47] stands as a de facto standard for modeling authentication schemes. We used it to assess the security of our proposed scheme for ASDs. The capabilities of the attacker $\mathcal{A}$ that we have considered in this paper are the following:

- Control over the public communication link: The adversary has full control over the communication channel used by the swarm devices, allowing them to intercept, manipulate, or modify any transmitted information. This reflects the ability of the adversary to tamper with the integrity and confidentiality of the communication.
- Extraction of data from sensor node memory: The adversary is assumed to have explicit techniques to extract data stored in the memory of an ASD sensor node. This highlights the potential vulnerability of the device's stored data and the need for protection against unauthorized access.
- Public knowledge of the Private Cloud pseudonym: The pseudonym of the private cloud, which is used in the authentication process, is supposed to be publicly known to all registered swarm devices. This assumption sets the context for the protocol design. It suggests that security measures should focus on protecting other aspects of the system, such as preventing unauthorized access to the private cloud's resources.
- Message Integrity and Modification Attack: We consider the presence of a MiTM node that attempts to tamper with the authentication messages exchanged between devices. The MiTM node may alter the content of the authentication messages or compromise their integrity, leading to potential security breaches and unauthorized access. We consider a MiTM node attempts to change the authentication message or compromise its content by compromising the Swarm device's messages.

- $\mathcal{A}$ completely controls the public channel.
- $\mathcal{A}$ can eavesdrop on, manipulate, or modify any information transmitted over the public channel. The memory of an ASD's sensor node can be accessed by $\mathcal{A}$ using explicit data extraction techniques.
- The PCS is designed to publicly available to all registered ASD devices.

## IV. PROPOSED PROTOCOL

This section will delve into the technical details of the Secure Authentication Protocol with SmartNiC Integration for Trustworthy Communications in Intelligent Swarm Systems (DPUAUT). This protocol aims to ensure secure and reliable communication within ISS by integrating SmartNiC technology. The proposed DPUAUT scheme employs key notations and definitions outlined in Table 3. Our scheme comprises four phases: Initialization, ASD registration, ASDCS registration, and authentication. The following is a detailed description of each phase involved in our proposed protocol:

### A. ISS-INITIALIZATION

In the initialization phase of the autonomous ISS, ($\mathcal{PCS}$) is responsible for initializing all the necessary parameters of the network. To accomplish this, $\mathcal{PCS}$ first selects a primary secret key $\mathcal{MSK}$ randomly from a set of prime numbers $Z_p^*$, ensuring the secrecy of the key. Next, $\mathcal{PCS}$ chooses a one way hash function. However, $\mathcal{PCS}$ securely stores $\mathcal{MSK}$, hash function $h(\cdot)$ is published and is to be used by all components of the system for cryptographic operations.

### B. DEVICE REGISTRATION PHASE

This registration phase ensures the proper identification and registration of devices within the system, establishing the necessary trust and secure communication channels between the PCS and the ASD.

1) Each device, denoted as $\mathcal{ASD}i$, initiates the registration process by sending its identity $IDi$ to the $\mathcal{PCS}$ as a registration request. The $\mathcal{PCS}$ verifies whether $ID_i$ is already registered. If it is not found in the system, the $\mathcal{PCS}$ accepts $ID_i$ as the unique identifier for the device. However, if $ID_i$ is already registered, the $\mathcal{PCS}$ requests $\mathcal{ASD}i$ to select an alternative identity $IDi$.
2) Upon receiving the registration request, the $\mathcal{PCS}$ generates a challenge message $\mathcal{CH}i$ and transmits it to $\mathcal{ASD}i$ securely over a communication link.
3) $\mathcal{ASD}i$ employs its PUF to generate a response message $\mathcal{R}i$. The response message is computed as $\mathcal{R}i \leftarrow \mathcal{P}i(\mathcal{C}i)$, where $\mathcal{P}i$ represents the PUF function applied to the challenge message $\mathcal{CH}i$. $\mathcal{ASD}i$ then sends the response message $\mathcal{R}_i$ to the $\mathcal{PCS}$.
4) Upon receiving $\mathcal{R}i$, the $\mathcal{PCS}$ verifies the digital signature $\sigma_i$ associated with $\mathcal{ASD}i$ and the registration request. If the signature is valid, the $\mathcal{PCS}$ proceeds with the registration process. Otherwise, the registration request is rejected.
5) If the digital signature is valid, the $\mathcal{PCS}$ assigns a temporary identity $TIDi$ for $\mathcal{ASD}i$. Subsequently, the $\mathcal{PCS}$ computes the key $\mathcal{K}i$ as $\mathcal{K}i = h(IDi|\mathcal{MSK})$, where $h(\cdot)$ denotes a one way hash function, $ID_i$ is the device's identity, and $\mathcal{MSK}$ represents the primary key. The $\mathcal{PCS}$ securely stores the device's registration data, including $ID_i$, $\mathcal{CH}i$, $\mathcal{R}i$, and $\sigma_i$ associated with $TID_i$ in its repository. Furthermore, the $\mathcal{PCS}$ encrypts the data using $\mathcal{MSK}$ and sends the pair $< TID_i, \mathcal{K}i >$ to $\mathcal{ASD}i$.
6) Finally, $\mathcal{ASD}i$ stores the pair $< TIDi, \mathcal{K}_i >$ in its Trusted Personal Device (TPD) for future use in the authentication protocol.

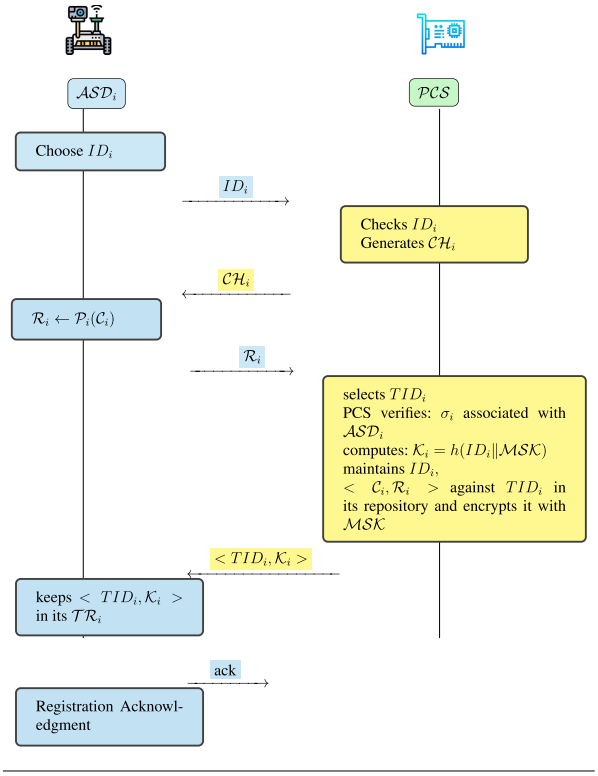



**FIGURE 3.** $\mathcal{ASD}_i$ in registration phase, $\mathcal{ASD}_i$ undergoes device authentication.

Figure 3 illustrates the registration phase of $\mathcal{ASD}_i$ (Authentication Server for Device Interface) with $\mathcal{PCS}$.

### C. AUTONOMOUS SWARM DEVICES COMPUTING SERVER REGISTRATION PHASE

The ($\mathcal{ASDCS}_j$) undergoes a registration process with the $\mathcal{PCS}$ to establish its identity and obtain the necessary credentials. This phase, performed offline, incorporates a digital signature function to ensure the authenticity and integrity of the registration. Figure 4 illustrates the registration phase of $\mathcal{ASDCS}$ with $\mathcal{PCS}$. The step-by-step procedure is as follows:

1) $\mathcal{ASDCS}_j$ sends a registration request to $\mathcal{PCS}$, indicating its intention to join the network.

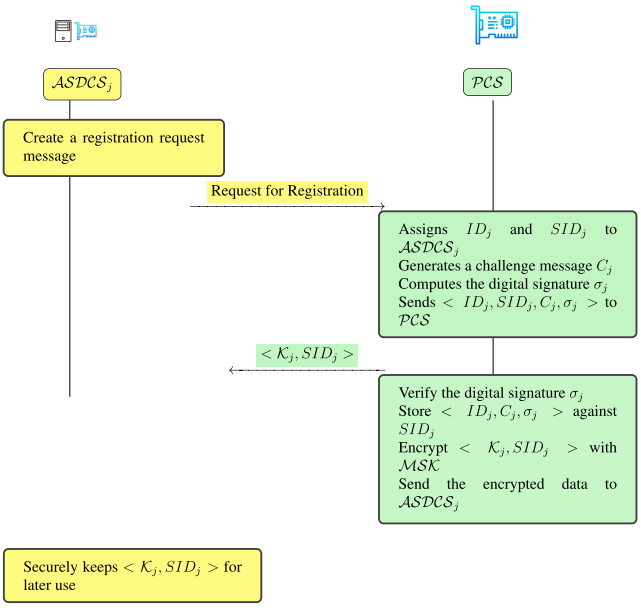



**FIGURE 4.** $\mathcal{ASDCS}_j$ **registration phase.**

2) Upon receiving a registration request from $\mathcal{ASDCS}_j$ assigned a unique identity $ID_j$ and a pseudo identity $SID_j$ by $\mathcal{PCS}$. After completing the necessary computations, the $\mathcal{PCS}$ system creates a challenge message denoted as $C_j$. This message is then transmitted to the $\mathcal{ASDCS}_j$ system for further processing.
3) $\mathcal{PCS}$ utilizes its private key to compute a digital signature $\sigma_j$ over the concatenated message $ID_j|SID_j|C_j$. The digital signature function is denoted as $Sign_{SK_j}(\cdot)$, where $SK_j$ represents the private key of $\mathcal{PCS}$. $\mathcal{PCS}$ sends the signed message $< ID_j, SID_j, C_j, \sigma_j >$.
4) $\mathcal{PCS}$ receives $< ID_j, SID_j, C_j, \sigma_j >$ and verifies the integrity and authenticity of the message. It uses the public key $PK_j$ corresponding to $SK_j$ to verify the digital signature. The verification function is denoted as $Verify_{PK_j}(\cdot)$. If the verification succeeds, $\mathcal{PCS}$ continues the registration process; otherwise, it rejects the registration request.
5) After successful verification, $\mathcal{PCS}$ computes the secret key $\mathcal{K}j$ using a one way hash function: $\mathcal{K}j = h(ID_j|\mathcal{MSK})$. The PCS stores the mapping of $\mathcal{K}j$ against $SIDj$ in its database for future reference. Furthermore, the PCS encrypts $\mathcal{K}j$ using $\mathcal{MSK}$ and sends the encrypted pair $< \mathcal{K}j, SID_j >$ to $\mathcal{ASDCS}_j$.
6) $\mathcal{PCS}j$ securely receives the pair $< \mathcal{K}j, SID_j >$ and stores it for later use in subsequent phases of the authentication protocol.

By incorporating the digital signature function, the ASDCS registration phase ensures the registration process's integrity, authenticity, and non-repudiation, providing an added layer of security to the overall authentication protocol.

### D. AUTHENTICATION PHASE

In this phase, the autonomous swarm device $\mathcal{ASD}i$ and the autonomous swarm devices computing server $\mathcal{ASDCS}j$ authenticate each other to establish a shared key for secure communication. The $\mathcal{ASD}i$ initiates the authentication process by sending a request to the Autonomous Swarm Devices Computing Server $\mathcal{ASDCS}j$. The request includes the ASD's unique identifier (UID) and a challenge message.

The $\mathcal{PCS}j$ verifies the authenticity of $\mathcal{ASD}i$ and the challenge response. If the verification is successful, $\mathcal{PCS}j$ sends a verification message to $\mathcal{ASD}i$. Subsequently, $\mathcal{ASD}i$ extracts the shared key from the verification message. This shared key is then employed for secure communication between $\mathcal{ASD}i$ and $\mathcal{PCS}j$. Both $\mathcal{ASD}i$ and $\mathcal{PCS}_j$ independently derive the shared key from the extracted key and a secret key shared with a trusted entity known as $\mathcal{PCS}$. It is emphasized that only authorized $\mathcal{ASD}\int\|$ can participate in the swarm.

#### 1) AUTHENTICATION INITIATION

- Random Values: $\mathcal{ASD}i$ generates a random nonce $ri$ and computes a timestamp $T_i$ and a digital signature $Verify_i$.
- Message Transmission: $\mathcal{ASD}i$ sends the message $(TIDi, T_i, Verify_i)$ to $\mathcal{ASDCS}_j$.

#### 2) CREDENTIAL VERIFICATION

- Message Reception: $\mathcal{ASDCS}j$ receives the message $(TIDi, T_i, Verify_i)$ from $\mathcal{ASD}_i$.
- Digital Signature Extraction: $\mathcal{ASDCS}j$ extracts the timestamp $Ti$ and the digital signature $Verify_i$ for later use.
- Credentials Verification: $\mathcal{ASDCS}j$ relays the message $(TIDi, SID_j)$ to $\mathcal{PCS}$ for verification of $\mathcal{ASD}_i$'s credentials.

#### 3) CREDENTIAL EXTRACTION AND KEY GENERATION

- Credential Retrieval: $\mathcal{PCS}$ checks the validity of $TID_i$ and extracts credentials accordingly.
- Key Retrieval and Computation: $\mathcal{PCS}$ retrieves $\mathcal{K}j$, computes $TID^n i$, and computes $\alpha$ for transmission.
- Repository Update: $\mathcal{PCS}$ updates temporary identities and transfers $\alpha$ to $\mathcal{ASDCS}_j$.

#### 4) VERIFICATION AND KEY DERIVATION

- Message Reception: $\mathcal{ASD}i$ receives $(Tj, Verify_j)$ from $\mathcal{PCS}_j$.
- Computations: $\mathcal{ASD}_i$ computes several values including $SK$.
- Verification: $\mathcal{ASD}_i$ verifies the digital signature equality for authentication.

If the equality holds true, $\mathcal{ASD}i$ accepts $SK$ as the mutually agreed-upon session key and proceeds to the next phase of the authentication protocol. However, if the equality does not hold, $\mathcal{ASD}i$ recognizes that some adversary has tampered with the messages, and the session is immediately aborted.

## V. SECURITY ANALYSIS

This section will comprehensively and rigorously evaluate our protocol, examining its performance and security features

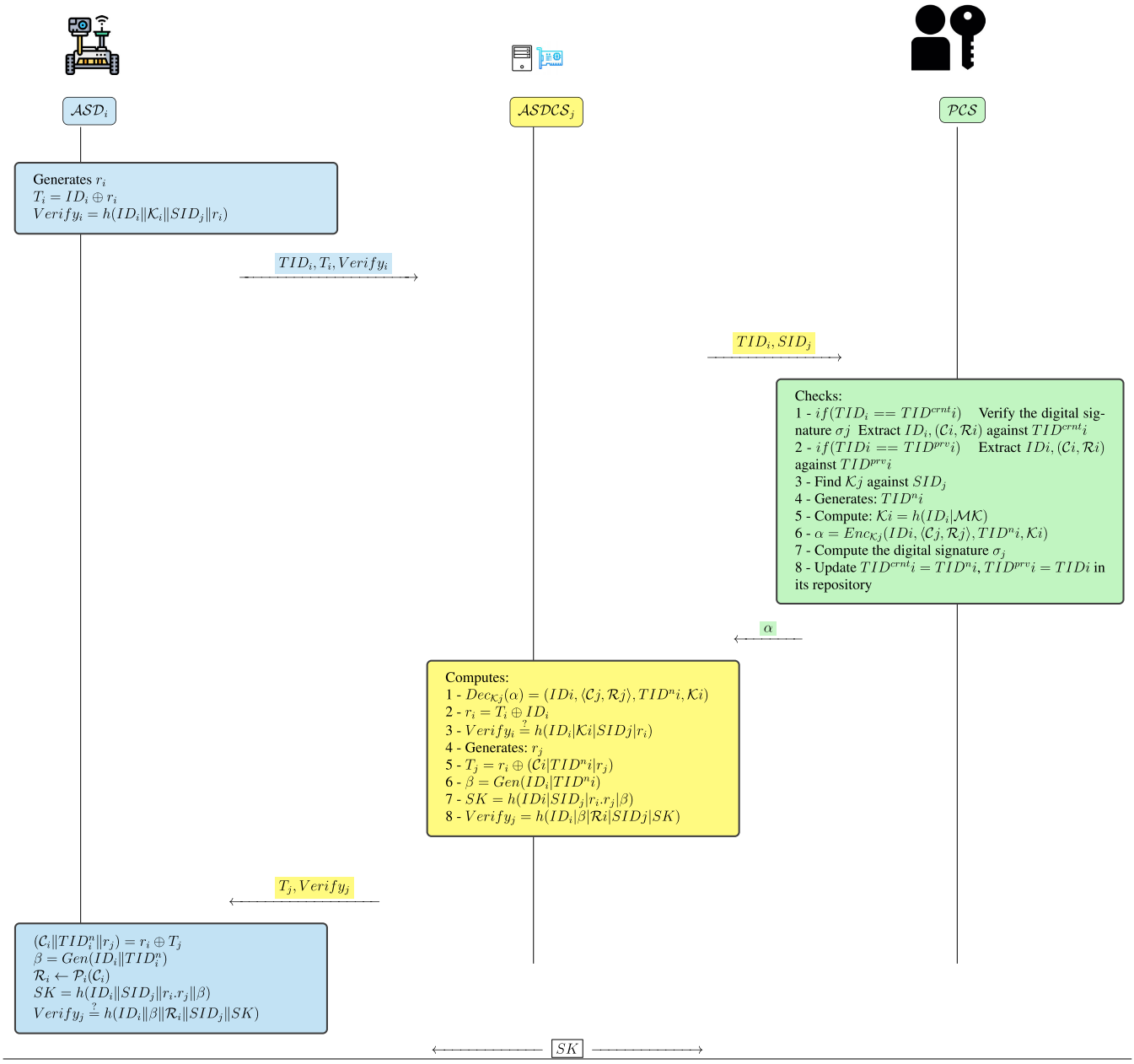



**FIGURE 5.** Authentication phase.

in predefined experimental scenarios. The evaluation will assess the protocol's strengths, weaknesses, and effectiveness using established methodologies and performance metrics. Our objective is to provide quantitative and objective insights into the performance characteristics of the protocol.

Through an informal evaluation, we can provide a qualitative assessment of the protocol's security guarantees, showcasing its strengths and detecting any areas of concern that may need further attention. This extensive analysis will supplement the formal evaluation presented in this section, providing a more comprehensive understanding of the protocol's security posture.

### A. AVISPA SIMULATION VERIFICATION-BASED ANALYSIS

This section uses the AVISPA (Automated Validation of Internet Security Protocols and Applications) tool [48] to simulate our secure protocol scheme. AVISPA is a protocol security analysis and verification tool integrating four backends (OFMC, CL-AtSe, SATMC, and TA4SP) into a single platform [6].

- On the Fly Model Checker (OFMC) OFMC is a model checker that uses abstract interpretation to analyze security protocols in a symbolic manner. It is a comprehensive and robust tool that can handle a variety of protocols and security properties. However, when

**Algorithm 1** Enhanced Mutual Authentication Protocol

**Require:** $\mathcal{ASD}_i$: ASD entity, $\mathcal{ASDCS}_j$: ASDCS entity, $\mathcal{PCS}$: PCS entity

**Ensure:** A shared session key $SK$ is established between $\mathcal{ASD}_i$ and $\mathcal{ASDCS}_j$.

1) **Authentication Initialization**
   - $\mathcal{ASD}_i$ generates a random nonce $r_i$ and a timestamp $T_i = ID_i \oplus r_i$.
   - $\mathcal{ASD}_i$ computes the digital signature $Verify_i = h(ID_i \| \mathcal{K}_i \| SID_j \| r_i)$.
   - $\mathcal{ASD}_i$ sends $\alpha = (TID_i, T_i, Verify_i)$ to $\mathcal{ASDCS}_j$.
2) **Credential Verification**
   - $\mathcal{ASDCS}_j$ receives $\alpha = (TID_i, T_i, Verify_i)$ from $\mathcal{ASD}_i$.
   - $\mathcal{ASDCS}_j$ decrypts $\alpha$ to obtain $(ID_i, \langle \mathcal{C}j, \mathcal{R}j \rangle, TID^n i, \mathcal{K}i)$.
   - $\mathcal{ASDCS}_j$ computes the signature verification key $Verify_i = h(ID_i | \mathcal{K}i | SID_j | r_i)$.
   - $\mathcal{ASDCS}_j$ verifies the digital signature $Verify_i$. If the verification fails, the protocol terminates.
3) **Credential Extraction**
   - $\mathcal{ASDCS}_j$ generates a random nonce $r_j$.
   - $\mathcal{ASDCS}_j$ computes $T_j = r_i \oplus (\mathcal{C}i | TID^n i | r_j)$.
   - $\mathcal{ASDCS}_j$ computes $\beta = Gen(ID_i \| TID^n i)$.
4) **Key Generation and Verification, and Key Derivation**
   - $\mathcal{ASDCS}_j$ computes the session key $SK = h(ID_i \| SID_j \| r_i . r_j | \beta)$.
   - $\mathcal{ASDCS}_j$ computes the digital signature $Verify_j = h(ID_i \| \beta | \mathcal{R}i \| SID_j \| SK)$.
   - $\mathcal{ASDCS}_j$ sends $\beta = (T_j, Verify_j)$ to $\mathcal{ASD}_i$.
   - $\mathcal{ASD}_i$ receives $\beta$ from $\mathcal{ASDCS}_j$.
   - $\mathcal{ASD}_i$ computes $\mathcal{CH}_i \| TID_i^n \| r_j = r_i \oplus T_j$.
   - $\mathcal{ASD}_i$ verifies the digital signature $Verify_j$. If the verification fails, the protocol terminates.
   - $\mathcal{ASD}_i$ computes $\mathcal{R}_i = \mathcal{P}_i(\mathcal{C}_i)$.
   - $\mathcal{ASD}_i$ computes the session key $SK = h(ID_i \| SID_j \| r_i . r_j \| \beta)$.
5) **Mutual Authentication Protocol Completion**
   - The enhanced mutual authentication protocol is complete.

dealing with large protocols, OFMC may become computationally intensive, which can affect its performance.

- CL-AtSe (Constraint-Logic-based Attack Searcher) CL-AtSe is a constraint solver that utilizes a technique called SAT-based attack searching to identify potential attacks on security protocols. Although it is a newly developed tool, it has proven to be effective in uncovering attacks on protocols that other tools might struggle with.
- SATMC (SAT-based Model Checker) SATMC is a model checker that employs propositional logic formulas to verify security protocols for any attacks. It utilizes a SAT solver to perform this task. SATMC is a powerful tool capable of handling large protocols. However, it can be computationally expensive.
- TA4SP (Tree Automata based on Automatic Approximations for the TA4SP is a powerful security protocol analysis tool. It has an edge over other similar tools because of its capability to handle complex protocols that are difficult to analyze. TA4SP uses regular tree languages and rewriting to approximate the knowledge of an intruder, making it a valuable tool in analyzing the security of a system. However, it is important to note that TA4SP may not be as precise as other tools and may not detect certain types of attacks.

```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/ranabest2.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 0.01s
 visitedNodes: 15 nodes
 depth: 5 plies
```

**FIGURE 6.** AVISPA OFMC result.

```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

PROTOCOL
 /home/span/span/testsuite/results/ranabest2.if

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed  : 8 states
 Reachable  : 4 states
 Translation: 0.00 seconds
 Computation: 0.00 seconds
```

**FIGURE 7.** AVISPA CL-ATSE result.

Our protocol includes three types of participants: ASDui, PCS, and SCS. We define the following roles in AVISPA to represent these participants:

- role_ADUi
- role_ASDCS
- role_PCS
- role_SN (session)
- role_environment

Each role is defined using the HLPSL specification, which includes parameters, states, and corresponding operations.

We simulated our protocol using the OFMC and CL-AtSe backends; the results are shown in Figure 6 and Figure 7, respectively. The simulation results indicate that our protocol

**TABLE 4. Queries with descriptions.**

| Query | Description |
|---|---|
| $Send(P_t, \text{msg})$ | This query models an active attack, where $\mathcal{A}$ can transmit a message msg to an instance $P_t$, and $P_t$ responds accordingly. |
| $Reveal(P_t)$ | This query allows the revelation of the shared session key $SK$ between $P_t$ and its counterpart to $\mathcal{A}$. |
| $CorruptMD(P^{t_1}_{\mathcal{ASD}_i})$ | $\mathcal{A}$ can obtain the values $\{TID_i, T_i\}$ stored in the device $\mathcal{TR}_i$ of $\mathcal{ASD}_i$. |
| $Test(P_t)$ | $\mathcal{A}$ requests $P_t$ to determine the correct value of the session key $SK$, and $P_t$ responds with a probabilistic outcome of an unbiased coin flip. |
| $Execute(P^{t_1}_{\mathcal{ASD}_i}, P^{t_2}_{\mathcal{ASDCS}_j}, P^{t_3}_{\mathcal{PCS}})$ | This query allows $\mathcal{A}$ to eavesdrop on the communication messages between $\mathcal{ASD}_i$, $\mathcal{ASDCS}_j$, and $\mathcal{PCS}$. |

satisfies the security requirements and is safe from active and passive attacks.

However, it is important to note that SATMC and TA4SP do not support the bitwise XOR operation. This is a limitation of these backends and prevents us from using them to simulate our protocol.

### B. FORMAL ANALYSIS

We have conducted a rigorous security analysis of our proposed protocol using the RoR model [49]. The RoR model allows us to prove our protocol's security under various attack scenarios, including impersonation, session key compromise, and replay attacks. In the RoR model, an adversary $\mathcal{A}$ interacts with the $t$-th instances of the involved entities: $\mathcal{ASD}_i$, $\mathcal{ASDCS}_j$, and $\mathcal{PCS}$. Thus, we have three instances: $P^{t_1}_{\mathcal{ASD}_i}$, $P^{t_2}_{\mathcal{ASDCS}_j}$, and $P^{t_3}_{\mathcal{PCS}}$, representing the $t_1$-th, $t_2$-th, and $t_3$-th instances of $\mathcal{ASD}_i$, $\mathcal{ASDCS}_j$, and $\mathcal{PCS}$, respectively. The RoR model includes several queries that simulate different types of attacks, such as *Test*, *Execute*, *Send*, *Reveal*, and *CorruptMD*. These queries allow us to analyze the protocol's security under various scenarios. Table 4 presents these queries and their descriptions.

We are introducing Theorem 1, proving that our proposed protocol, DPUAUT, is secure against the session key (SK) attack. All participants, including $\mathcal{A}$, can access the provided one-way hash function $h(\cdot)$. It is modeled as a random oracle, the Hash oracle, as described in [50]. The proof is established using the RoR model and the Oracle queries specified in Table 4.

Theorem 1 states that if there is an adversary called $\mathcal{A}$ who attacks the protocol *DPUAUT*, and the attack lasts for a polynomial time of $t$, then we can use parameters such as $qr_h$ to represent the number of *Hash* and *Send* queries, $qr_s$ to represent the number of hash queries, $l$ to represent the range of the hash function space $h(\cdot)$, and $|Hash|$ to represent the secret biometric key bits. These parameters can help us analyze the attack and understand its impact on the protocol.

The advantage of the adversary $\mathcal{A}$ in successfully obtaining the SK shared between $\mathcal{ASD}i$ and $\mathcal{PCS}j$ is bounded by:

$$Adv^{\mathcal{A}}_{DPUP}(t) \leq \frac{qr_h^2}{|Hash|} + 2\max\left\{C' \cdot qr_s^{s'}, \frac{qr_s}{2^l}\right\} \quad (1)$$

where $C'$ is a constant factor, and $s'$ represents the probability of a successful *Test* query in the RoR model.

Theorem 1 provides an upper bound on the advantage of the adversary $\mathcal{A}$ based on the number of queries made and the hash function properties employed in the protocol. It reveals that the number of *Send* queries determines the adversary's advantage, the hash function's security characteristics, and the hash function's range space.

The DPUAUT Protocol (DPUP) is secure against all the attack scenarios in the RoR model.

We prove the security of the DPUP using a series of reductions. In each reduction, we show that the advantage of an adversary $\mathcal{A}$ in winning one game is bounded by the advantage of $\mathcal{A}$ in winning another game.

***Proof:*** We provide proof following the methods outlined in the following references: [1], [51], [52].

We use four games: $G0$, $G1$, $G2$, and $G3$ to prove the security of our proposed authentication protocol in the ROR model. We assume that $Succ_j$ denotes the event that an adversary $\mathcal{A}$ is able to guess the random bit $I - bit$ in $G_j$ correctly. The advantage of $\mathcal{A}$ in winning $G_j$ is defined as $Adv^{\mathcal{A},G_j}_{DPUP} = \Pr[Succ_j]$, where $\Pr[Succ_j]$ is the probability of $Succ_j$ occurring.

- We define ***G0***, which corresponds to the actual attack carried out by advarsary $\mathcal{A}$ against the DPUP in the RoR model. During the initialization process of $G0$, a random bit is selected as the $I - bit$.
  In $G0$, the adversary $\mathcal{A}$ interacts with the instances $P^{t_1}\mathcal{ASD}i$, $P^{t_2}\mathcal{ASDCS}j$, and $P^{t_3}\mathcal{PCS}$ representing the $t_1$-th instance of $\mathcal{ASD}i$, the $t_2$-th instance of $\mathcal{ASDCS}_j$, and the $t_3$-th instance of $\mathcal{PCS}$, respectively. During the interaction, the adversary uses the Oracle queries described in Table 4.
  The advantage of the adversary $\mathcal{A}$ in $G0$ can be calculated as follows:

$$Adv^{\mathcal{A}}_{\text{DPUP}} = \left|2 \cdot Adv^{\mathcal{A}}_{\text{G0}} - 1\right| \quad (2)$$

  $G0$ represents the actual attack scenario in which the adversary $\mathcal{A}$ attempts to exploit vulnerabilities in the DPUP. The advantage calculation provides a measure of the adversary's success in obtaining the SK between $\mathcal{ASD}i$ and $\mathcal{PCS}j$ under the given conditions and queries.
- In this Game 1 ($G1$), we consider a more challenging attack scenario where the adversary $\mathcal{A}$ models the attack by executing *CorruptMD* query to tamper with the physical components of the PUF.
  In $G1$, adversary $\mathcal{A}$ utilizes the *Execute* query to intercept the following messages: $TID_i$, $T_i$, $TID_i$, $SID_j$, $\alpha$, and $T_j$, $Verify_j$, as listed in Table 4.
  After obtaining the secret key, adversary $\mathcal{A}$ models the attack by executing the *Test* and *Reveal* oracle queries to guarantee that the secret key is genuinely random and not biased towards any specific value or set of values. This validation process is crucial to guarantee the robustness and security of the cryptographic system.

The session key shared between the device with identity "i" denoted as $\mathcal{ASD}i$ and the ASDCS with identity "j" denoted as $\mathcal{ASDCS}j$ is calculated using the following formula: $SK = h(ID_i|SID_j|r_i \cdot r_j|\beta)$. Here, $ID_i$ represents the identity of the device, $SID_j$ represents the identity of the ASDCS, $r_i$ and $r_j$ are random numbers generated by the device and the ASDCS respectively, and $\beta$ is a constant value.

In order to assess the value of $SK$, it is necessary for $\mathcal{A}$ to have access to both short-term secrets, namely $(r_i, TID_i^{new} \cdot r_j)$, as well as long-term secrets, namely $(ID_i, SID_j)$. However, it's important to note that the short-term and long-term secrets are kept anonymous from $\mathcal{A}$.

By intercepting the communicated messages $TID_i, T_i$, $TID_i, SID_j$, $\alpha$, and $T_j, Verify_j$, and by corrupting the physical components of the PUF, $\mathcal{A}$ has a chance of winning $G1$. The adversary can obtain the session key $SK$ if it can compromise the short-term or long-term secrets.

However, the protocol's security is still preserved even though $\mathcal{A}$ can execute the *CorruptMD* query. This is because the short-term secrets are updated frequently, and the long-term ones are stored securely. Therefore, it is difficult for $\mathcal{A}$ to compromise the short-term or long-term secrets, even if it can corrupt the physical components of the PUF.

- ***G2***: Game 2 ($G2$) involves both *Hash* and *Send* queries. The communicated messages $TID_i, T_i$, $TID_i, SID_j$, $\alpha$, and $T_j, Verify_j$ use the one way hash function $h(.)$ to secure their computations. Given that the computations involve random values such as identities, timestamps, and secret credentials, collisions are unlikely to occur while executing *Hash* and *Send* queries. This is because one-way hash functions are designed to be difficult to invert, meaning it is difficult to find two inputs that produce the same output. Therefore, it is unlikely that the adversary $\mathcal{A}$ will be able to find a collision during the attack model execution of the *Hash* and *Send* oracle queries, even if it can intercept and modify the messages exchanged between the involved entities.

  $G2$ and $G1$ are nearly identical, with the only difference being that the oracle queries *Hash* and *Send* simulations are executed differently in $G2$. The outputs of the queries in $G2$ are compared to those in $G1$. The relationship between the advantages in $G1$ and $G2$ can be expressed as:

$$|Adv_{DPUP}^{\mathcal{A},G1} - Adv_{DPUP}^{\mathcal{A},G2}| \leq \frac{qr_h^2}{2|Hash|} \tag{3}$$

  This equation shows that the difference between the advantages of the adversary $\mathcal{A}$ in $G1$ and $G2$ is bounded by $\frac{qr_h^2}{2|Hash|}$. This bound is small because the one-way hash function $h(.)$ is assumed to be cryptographically secure. The oracle queries *Hash*, and *Send* execution does not significantly impact the security of the DPUP.

- ***G3***: Game 3 ($G3$) is a modified version of game $G2$ that includes the simulation of a query *CorruptMD* given in Table 4. When adversary $\mathcal{A}$ try lunch models attack execute the *CorruptMD* oracle query, they will have access to the secret credentials $\{TID_i, \mathcal{K}_i\}$. To execute the Oracle query, $\mathcal{A}$ is required to provide additional information, such as the physical location of the PUF or the PUF's challenge and response values, to execute this query.

  The *CorruptMD* query poses a significant challenge for $\mathcal{A}$ due to the security of the PUF, making it difficult to obtain the values of $\mathcal{TR}_i$. The inability to access these values could potentially alter the behavior of the PUF, adding further complexity to the situation. Additionally, using a secure one-way hash function makes it difficult for the adversary to find a collision during the oracle queries *Hash* and *Send* execution. Therefore, the bound on the difference between the adversary's advantages in G2 and Game 3 is small.

  This bound is expressed in the following equation:

$$|Adv_{DPUP}^{\mathcal{A},G2} - Adv_{DPUP}^{\mathcal{A},G3}| \leq \max\left\{C' \cdot qr_s^{s'}, \frac{qr_s}{2^l}\right\} \tag{4}$$

$$|Adv_{DPUP}^{\mathcal{A},G2} - Adv_{DPUP}^{\mathcal{A},G3}| \leq max\left\{C' \cdot qr_s^{s'} \frac{qr_s}{2^l}\right\} \tag{5}$$

  wheree $C'$ is a constant that depends on the difficulty of the *CorruptMD* query and the security of the PUF.

For adversary $\mathcal{A}$ to win $G3$, it must correctly predict the value of the $I$-bit when the *Test* query is run, as all queries are simulated by $\mathcal{A}$. Therefore, the advantage of $\mathcal{A}$ in $G3$ is equal to $\frac{1}{2}$.

By applying the technique of equation simplification to equations (1)-(5) and utilizing the triangular inequality, we can derive the following output:

$$\begin{aligned}\frac{1}{2} \cdot Adv_{DPUP}^{\mathcal{A}}(t) &= Adv_{DPUP}^{\mathcal{A},G0} - \frac{1}{2} \\ &= |Adv_{DPUP}^{\mathcal{A},G1} - Adv_{DPUP}^{\mathcal{A},G3}| \\ &\leq |Adv_{DPUP}^{\mathcal{A},G1} - Adv_{DPUP}^{\mathcal{A},G2}| \\ &\quad + |Adv_{DPUP}^{\mathcal{A},G2} - Adv_{DPUP}^{\mathcal{A},G3}| \\ &\leq \frac{qr_h^2}{2|Hash|} + \max\left\{C' \cdot qr_s^{s'}, \frac{qr_s}{2^l}\right\}\end{aligned} \tag{6}$$

Therefore, it follows that the equation:

$$Adv_{DPUP}^{\mathcal{A}}(t) \leq \frac{qr_h^2}{2|Hash|} + \max\left\{C' \cdot qr_s^{s'}, \frac{qr_s}{2^l}\right\}. \tag{7}$$

### C. INFORMAL SECURITY ANALYSIS

In this section, we provide our proposed protocol's informal security analysis and highlight that it's the necessary security requirements. Additionally, we discuss its resilience against common security threats, such as physical attacks, cloud computing server attacks, masquerading attacks, device anonymity and privacy, device un-traceability, de-synchronization attacks, and impersonation attacks.

### 1) RESILIENCE AGAINST PHYSICAL ATTACKS

In the context of our devised protocol, the resilience against physical attacks, wherein an adversary, denoted as $\mathcal{A}$, attempts to compromise the PUF of $\mathcal{ASD}i$, is of utmost importance. By damaging or tampering with $\mathcal{TR}_i$, the adversary aims to undermine the system's security.

However, our protocol incorporates robust measures to counter such physical attacks. Utilizing a PUF plays a crucial role in ensuring the integrity of the authentication process. When faced with attempts to damage or tamper with $\mathcal{TR}_i$, the PUF exhibits a behavior change, rendering it meaningless for the adversary. Consequently, any manipulation or impairment of $\mathcal{TR}_i$ results in the PUF failing to produce the expected output.

The significance of this lies in the fact that the successful verification of $\mathcal{ASD}i$ relies on the accurate determination of $\mathcal{R}i$, which is contingent upon the PUF producing the required output. When the PUF fails to provide the expected result due to physical attacks, the Autonomous Swarm Devices Computing Server $\mathcal{ASDCS}_j$ is equipped to identify and counter such illicit activities.

Thus, the devised protocol effectively establishes resilience against physical attacks by leveraging the inherent characteristics of the PUF. The PUF's ability to detect and respond to tampering attempts ensures the integrity and reliability of the authentication process, safeguarding the system against unauthorized access.

Our protocol's resilience against physical attacks stems from the active involvement of the PUF, which exhibits a change in behaviour in response to any attempts to compromise $\mathcal{TR}_i$. By rendering the PUF meaningless in the face of tampering, the protocol reinforces the system's security and upholds the integrity of the authentication mechanism.

### 2) MASQUERADING ATTACKS

Network entities communicate over unsecured channels during our proposed protocol's registration and authentication phase, exposing the system to various vulnerabilities. One such vulnerability is masquerade attacks, where an adversary attempts to impersonate a legitimate device or entity to gain unauthorized access. Masquerade attacks can be particularly disruptive in authentication protocols due to the inherent trust placed in the communication channels. An adversary with sufficient protocol knowledge can intercept and manipulate messages exchanged during authentication, potentially gaining access to sensitive data or compromising system integrity. In Section III-F, it was discussed that $\mathcal{A}$ possesses the proficiency to eavesdrop on messages transmitted through public channels. Once the messages are intercepted, adversary $\mathcal{A}$ can launchÂ an attack model for impersonation attacks.

- Masquerading Attack: During our proposed protocol's registration and authentication phase, a swarm device $\mathcal{ASD}i$, start a registration request $TIDi, T_i, Verify_i$ towards the Autonomous Swarm Devices Computing Server $\mathcal{ASDCS}_j$. In this attack scenario, an adversary denoted as $\mathcal{A}$ attempts to intercept and launch a swarm masquerading attack by impersonating a legitimate swarm device. However, a careful analysis reveals that such an attack would be challenging for the adversary. In the devised protocol, the security measures prevent the adversary from computing essential values required to generate a valid registration message. Specifically, the adversary lacks access to crucial information, such as the device's identity ($ID_i$), secret parameter ($\mathcal{K}i$), and session-specific random number ($ri$). The verification parameter $Verify_i$, which is computed as $h(ID_i|\mathcal{K}i|SIDj|r_i)$, remains elusive to the adversary due to the unavailability of $ID_i$, $\mathcal{K}i$, and $ri$. Similarly, the calculation of $T_i$ as $ID_i \oplus r_i$ is infeasible for the adversary, who lacks the necessary knowledge of $ID_i$ and $r_i$.Consequently, the adversary's attempts to impersonate a legitimate swarm device, $\mathcal{ASD}_i$, are thwarted by the robust security mechanisms of the devised protocol. By ensuring the secure generation and exchange of session keys and leveraging cryptographic functions, the protocol effectively mitigates swarm masquerading attacks and safeguards the integrity of the authentication process.
- Masquerading Attack on $\mathcal{PCS}$: In the context of our proposed protocol, it is essential to address the potential masquerading attack on $\mathcal{PCS}j$, wherein an adversary, denoted as $\mathcal{A}$, attempts to impersonate $\mathcal{PCS}j$ to gain unauthorized access. To successfully carry out this attack, $\mathcal{A}$ would need to generate a valid message $T_j, Verify_j$ that appears to originate from $\mathcal{PCS}j$ and is intended for $\mathcal{ASD}i$. The computation of $Verify_j$ involves several crucial parameters, including the swarm device's identity $ID_i$, the response of the PUF denoted as $\mathcal{R}i$, the pseudonym of $\mathcal{PCS}j$ denoted as $SID_j$, and the session key characterized as $SK$. To obtain the genuine value of $ID_i$, it is necessary to decrypt the encrypted message $\alpha$ using the secret key $\mathcal{K}j$ of $\mathcal{PCS}j$. However, it will be discussed in Section V-C2 that $\mathcal{A}$ does not possess any knowledge about $\mathcal{K}j$, thereby making it impossible for $\mathcal{A}$ to retrieve the actual $IDi$. Furthermore, the ingredients required for the generation of $T_j$ are also inaccessible to $\mathcal{A}$ without decrypting $\alpha$. As a result, $\mathcal{A}$ lacks the necessary information to construct a valid message $T_j, Verify_j$ that can successfully masquerade as $\mathcal{ASDCS}_j$. Therefore, our devised protocol mitigates the risk of masquerading attacks targeted at Autonomous Swarm Devices Computing Servers. The robustness of our protocol against masquerading attacks on $\mathcal{PCS}j$ can be attributed to the secure handling of sensitive parameters and the utilization of encryption mechanisms. By ensuring that critical components, such as the secret key $\mathcal{K}j$ and the decrypted message $\alpha$, remain inaccessible to unauthorized entities, the protocol establishes a secure communication

channel and effectively prevents impersonation of $\mathcal{ASDCS}_j$.

- Masquerading Attack on ASDCS: In our protocol authentication and registration phase, $\mathcal{PCS}$ transmits a message $\alpha$ to $\mathcal{ASDCS}j$. There is a security vulnerability known as impersonation, which can occur when an unauthorized entity $\mathcal{A}$ intercepts a message ($\alpha$) to impersonate a legitimate $\mathcal{ASDCS}$. This vulnerability needs to be addressed to ensure the security of the system. The message $\alpha$ contains encrypted parameters, including $ID_i$, $\langle \mathcal{C}_j, \mathcal{R}_j \rangle$, $TID_i^n$, and $\mathcal{K}_i$, which are encrypted using the secret key $\mathcal{K}_j$ of $\mathcal{ASDCS}_j$. It is practically impossible for $\mathcal{A}$ to compute the value of $\alpha = Enc_{\mathcal{K}_j}(ID_i, \langle \mathcal{C}_j, \mathcal{R}_j \rangle, TID_i^n, \mathcal{K}_i)$. Crucially, $\mathcal{A}$ lacks knowledge of $\mathcal{K}j$ for $\mathcal{ASDCS}j$, rendering the computation of $\alpha$ infeasible. Consequently, the devised protocol is due to encrypting all parameters with $\mathcal{K}j$.

### 3) DEVICE ANONYMITY AND PRIVACY

Preserving device anonymity and privacy is a crucial requirement in the design of an authentication protocol. Our devised protocol addresses this requirement by masking while exchanging the swarm device's identity $ID_i$ over the public medium. Specifically, the exchanged value is represented as $T_i = ID_i \oplus r_i$, where $r_i$ is a session-specific random number.

The masking operation ensures that the identity $ID_i$ remains concealed from any potential adversary $\mathcal{A}$. Even if $\mathcal{A}$ attempts to retrieve $ID_i$ from the exchanged value $T_i = ID_i \oplus r_i$, the lack of knowledge regarding the session-specific random number $r_i$ prevents $\mathcal{A}$ from successfully recovering $ID_i$. Consequently, the presented scheme effectively preserves device anonymity, making it difficult for an adversary to associate specific swarm devices with their corresponding identities. This enhances device privacy and strengthens the overall security of the authentication protocol.

### 4) SWARM DEVICE UNTRACEABILITY

The devised protocol ensures the untraceability of $\mathcal{ASD}i$ by incorporating session-specific random numbers $ri$ in the computation of $T_i$. As a result, each registration message $TID_i$, $T_i$, $Verify_i$ generated for a session will be unique. This uniqueness makes it impossible for $\mathcal{A}$ to determine whether registration messages from different sessions belong to the same swarm device or different ones. Consequently, our proposed protocol guarantees the untraceability of $\mathcal{ASD}_i$, preserving device privacy and preventing the tracking of swarm devices.

### 5) RESILIENCE AGAINST DESYNCHRONIZATION ATTACKS

Desynchronization attacks, also known as Denial of Service (DoS) attacks, pose a significant threat to the smooth operation of an authentication protocol. These attacks aim to disrupt the synchronization between different entities in the protocol, leading to service disruptions or system failures. To counter this attack, it is essential to implement measures that ensure regular changes in the confidential data exchanged during each session.

In our devised protocol, $\mathcal{A}$ may attempt to launch a desynchronization attack by preventing $\mathcal{ASD}i$ from receiving the message $Tj, Verify_j$ sent by $\mathcal{PCS}j$. However, our protocol mitigates such attacks by incorporating a stringent verification process. Specifically, $\mathcal{ASD}i$ verifies whether the condition $Verify_j \stackrel{?}{=} h(ID_i|\beta|\mathcal{R}i|SIDj|SK)$ holds true or not. This verification ensures the received message is authentic and consistent with the expected values.

If the verification check fails, indicating a discrepancy in the values, $\mathcal{ASD}i$ does not update $\mathcal{TID}^{new}i$, and the session is terminated. By promptly detecting and responding to potential desynchronization attacks, the devised protocol effectively safeguards against service disruptions and maintains the integrity and synchronization of the authentication process.

**TABLE 5.** Evaluation of authentication schemes based on these security properties A1: Mutual Authentication A2: Device Anonymity, A3: Provide Perfect Forward Secrecy, A4: Replay Attack, A5: Key Agreement, A6: Device Impersonation Attack, A7: Withstand De-synchronization attack, A8: Formal Security Analysis, A9: Informal Security Analysis.

| Schemes | Security Properties | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 |
| [14] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | × |
| [17] | ✓ | ✓ | × | ✓ | ✓ | ✓ | × | × | × |
| [18] | ✓ | ✓ | × | ✓ | ✓ | × | ✓ | ✓ | × |
| [22] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ |
| [20] | ✓ | × | × | × | × | ✓ | × | ✓ | × |
| [24] | ✓ | ✓ | ✓ | × | ✓ | ✓ | × | ✓ | × |
| [25] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ |
| [26] | ✓ | ✓ | × | ✓ | ✓ | ✓ | × | ✓ | ✓ |
| DPUAUT | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

## VI. PERFORMANCE AND SECURITY FEATURES EVALUATION

In this section, we provide a comprehensive performance and security features analysis of our proposed protocol with several state-of-the-art protocols [11], [14], [17], [18], [20], [22], [24], [25], [26], [27], [53]. Our scheme achieves all the security features outlined in Table 5, including mutual authentication, device anonymity, perfect forward secrecy, resistance to replay attacks, key agreement, defense against device impersonation attacks, resilience to de-synchronization attacks, formal security analysis, and informal security analysis. In our related work section, we provide a detailed analysis of why existing schemes fall short of delivering comprehensive security coverage. The performance comparison is based on key performance metrics, including computation overhead, communication overhead, and each protocol's security features. We analyze the computational requirements of each protocol to assess the efficiency of our proposed solution. Also, we look over the communication overhead incurred during the execution of the protocols, considering the number and size of messages sent between the involved entities.

**TABLE 6.** Execution time of primitive operations.

| Operation | Description | Execution Time | |
|---|---|---|---|
| | | Swarm Device | Dell Server |
| $T_{hf}$ | Hash Function | 0.096 ms | 0.001 ms |
| $T_m$ | Multiplication | 0.084 ms | 0.01 ms |
| $T_AESe-d$ | Encryption-Decryption | 0.091 ms | 0.01 ms |
| $T_b$ | Pairing Operation | 1.6 ms | 0.06 ms |

Furthermore, we evaluate the security features provided by both our state-of-the-art protocols, considering the robustness against various attacks and vulnerabilities. By conducting this comprehensive performance comparison, we aim to demonstrate the superiority of our proposed protocol in terms of efficiency and security. The analysis will help us understand the advantages and drawbacks of each protocol, enabling us to make informed conclusions about the best solution. Our proposed protocol comprises of three entities including $\mathcal{ASD}_i$, $\mathcal{ASDCS}_j$ and $\mathcal{PCS}$. Similarly, the state-of-the-art protocols [11], [14], [18], [19], [27] also deals with either two ($\mathcal{ASD}_i$, $\mathcal{ASDCS}_j$) or three entities ($\mathcal{ASD}_i$, $\mathcal{ASDCS}_j$, $\mathcal{PCS}$). In our implementation, we focused on the cryptographic operations performed at $\mathcal{ASD}i$ and $\mathcal{PCS}j$ ends. Specifically, we considered the cryptographic operations of $\mathcal{ASD}_i$ and implemented them on swarm devices. These devices are equipped with Nvidia Bluefield-2 DPU and wireless communication modules, essential components for the protocol deployment. The swarm devices used in our scenario are designed for autonomous vehicle swarm systems and are equipped with Nvidia Bluefield-2 DPUs. These DPUs provide high-performance computing capabilities and specialized hardware acceleration, particularly for cryptographic operations required by secure authentication protocols. Their robust processing power ensures the efficient execution of the authentication protocol on the swarm devices. Additionally, these swarm devices are equipped with wireless communication modules that support reliable and secure communication protocols, enabling seamless connectivity and information sharing within the swarm. On the other hand, the cryptographic operations executed at the $\mathcal{ASDCS}_j$ end are implemented on a Dell server. Our private cloud implementation employs two Dell Power Edge R760 servers, each with a 128-core Intel Xeon CPU, 256-GB RAM, 3.6 TB storage (15k RPM SAS), and two 100-Gbps Ethernet interfaces in DPU. It runs the Ubuntu 20.04 LTS 64-bit operating system. We used the PyCharm Integrated Development Environment (IDE) for software development.

To evaluate the performance of the proposed protocol, we conducted a detailed analysis of the computation overhead. We compared our protocol with several related state-of-the-art protocols, including those referenced in [14], [17], [18], [19], [22], [27], and [53]. The results, summarized in Table 7, indicate that our proposed protocol exhibits lower computation overhead compared to the state-of-the-art protocols. The computation overhead of our proposed protocol, as compared to various state-of-the-art protocols,

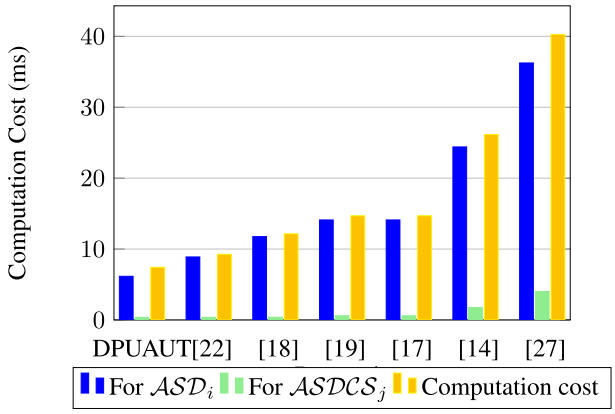



**FIGURE 8.** Computation cost comparison.

is visually compared in Figure 8. The protocol listing is presented horizontally, with the computation time associated with each protocol displayed on the vertical axis. Analysis of Figure 8 reveals a lower computation overhead at both ends, namely $\mathcal{ASD}i$ and $\mathcal{PCS}j$, in our proposed protocol compared to the corresponding ends in the state-of-the-art protocols. Additionally, the overall computation overhead of our protocol stands out as significantly less than that of the state-of-the-art protocols.

These results emphasize our proposed authentication protocol's efficiency and lightweight nature, making it a promising solution for secure authentication in autonomous vehicle swarm systems.

We have analyzed the computation overhead of our proposed protocol and other state-of-the-art protocols. To do this, we have used the computation times of primitive operations listed in Table 6. As the registration process is a singular event, our evaluation solely concentrates on cryptographic operations during authentication. This approach enables us to determine the computation overhead with precision. This analysis includes our protocol as well as those presented in previous works [14], [17], [18], [19], [22], [27], [54].

In our proposed protocol, $\mathcal{ASD}_i$ performs $3|h|$ cryptographic operations during the registration and authentication phases, resulting in a computation time of $(3 \times 2.02) \approx 6.06$ ms. Similarly, $\mathcal{ASDCS}_j$ performs $3|h| + 1|e/d|$ cryptographic operations during the registration and authentication phases, resulting in a computation time of $(3 \times 0.091) + (1 \times 0.01) \approx 0.101$ ms. Therefore, our proposed protocol's total computation cost for the registration and authentication phases is $(6.06 + 0.101) \approx 6.161$ ms.

In secure communication protocols, communication overhead refers to the number of bits required for message exchange among the entities involved. This section thoroughly compares communication overhead between our proposed protocol and the current state-of-the-art protocols [14], [18], [19], [27]. Our analysis focuses on the messages depicted in Figure 5, and our assumptions for computing the communication overhead are presented in Table 7.

**TABLE 7. Analysis of computation and communication overheads.**

| Protocols | $\mathcal{ASD}_i$ Side | $\mathcal{ASDCS}_j$ Side | Aggregated Computation Overhead | Aggregated Communication Overhead |
|---|---|---|---|---|
| DPUAUT | $3T_h \approx 6.06$ ms | $3T_h + 1T_{e/d} \approx 0.01$ ms | 6.161 ms | 1052 bits |
| [22] | $3T_h + 1T_{e/d} \approx 8.888$ ms | $3T_h + 1T_{e/d} \approx 0.373$ ms | 9.261 ms | 2145 bits |
| [18] | $5T_h \approx 11.755$ ms | $4T_h \approx 0.392$ ms | 12.147 ms | 1568 bits |
| [17] | $6T_h \approx 14.106$ ms | $6T_h \approx 0.588$ ms | 14.694 ms | 1600 bits |
| [14] | $9T_h + 2T_{pm} \approx 24.405$ ms | $6T_h + 1T_b + 1T_{pm} \approx 1.756$ ms | 26.161 ms | 1664 bits |

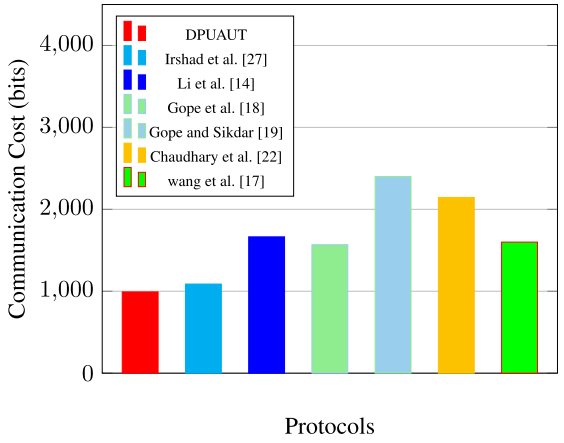



**FIGURE 9. Communication cost comparison.**

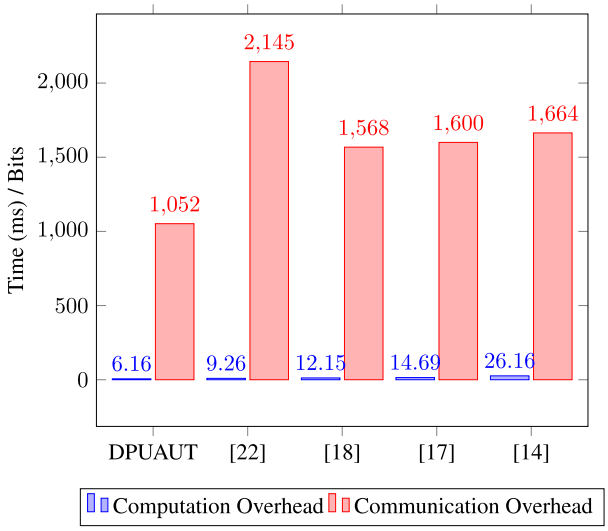



**FIGURE 10. Analysis of aggregated overheads.**

Figure 9 compares the communication overhead of our proposed and state-of-the-art protocols. The comparison chart showcases the protocols being analyzed, with the X-axis indicating the protocols and the Y-axis representing the communication overhead linked to each. Figure 9 shows that the proposed protocol outperforms all related state-of-the-art protocols regarding communication overhead. Figure 10 shows that the proposed protocol outperforms all related state-of-the-art protocols regarding aggregated computational overhead.

## VII. CONCLUSION AND FUTURE WORK

This paper presents a novel authenticated key agreement solution tailored specifically for the autonomous swarm intelligent system, aiming to ensure robust swarm device communication security between cloud servers and lightweight swarm devices. The evaluation analysis demonstrates that our protocol possesses the necessary security measures to for privacy and protection of sensitive communication within the autonomous swarm intelligent system. The protocol shows excellent computational efficiency, communication overhead, and resource utilization, reinforcing its prominence of secure key agreement protocols for the autonomous swarm intelligent system. The protocol's ability to resist cyber and physical attacks, combined with its rigorous security analysis, ensures confidence in its security guarantees. Moreover, the protocol's superior performance characteristics further solidify its position as a leading solution among recent state-of-the-art protocols. Our approach may face challenges, such as susceptibility to environmental variations affecting PUF responses, potentially causing authentication failures. We investigate error correction and noise filtering techniques to enhance PUF reliability. As future work, we plan to investigate secure multi-party computation techniques to distribute the PUF verification process across multiple DPUs. This would enhance the system's resilience against attacks targeting the DPU with different real-time applications. We are also planning to implement and evaluate the RISC-V processor-based ring oscillator circuit for PUF generation in blue-field-3. This allows the PUF circuit to be customized for each BlueField-3 DPU device. This makes it difficult for attackers to tamper with the PUF by machine learning techniques.

## REFERENCES

[1] V. O. Nyangaresi, "Privacy preserving three-factor authentication protocol for secure message forwarding in wireless body area networks," *Ad Hoc Netw.*, vol. 142, Apr. 2023, Art. no. 103117.

[2] L. Barsellotti, F. Alhamed, J. J. Vegas Olmos, F. Paolucci, P. Castoldi, and F. Cugini, "Introducing data processing units (DPU) at the edge [Invited]," in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2022, pp. 1–6.

[3] K. R. Ozyilmaz and A. Yurdakul, "Designing a blockchain-based IoT with Ethereum, swarm, and LoRa: The software solution to create high availability with minimal security risks," *IEEE Consum. Electron. Mag.*, vol. 8, no. 2, pp. 28–34, Mar. 2019.

[4] A. Mullai and K. Mani, "Enhancing the security in RSA and elliptic curve cryptography based on addition chain using simplified swarm optimization and particle swarm optimization for mobile devices," *Int. J. Inf. Technol.*, vol. 13, no. 2, pp. 551–564, Apr. 2021.

[5] L. Buttyan and J.-P. Hubaux, *Security and Cooperation in Wireless Networks: Thwarting Malicious and Selfish Behavior in the Age of Ubiquitous Computing*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[6] L. Agilandeeswari, S. Paliwal, A. Chandrakar, and M. Prabukumar, "A new lightweight conditional privacy preserving authentication and key—Agreement protocol in social Internet of Things for vehicle to smart grid networks," *Multimedia Tools Appl.*, vol. 81, no. 19, pp. 27683–27710, Aug. 2022.

[7] K. Xue, W. Meng, S. Li, D. S. L. Wei, H. Zhou, and N. Yu, "A secure and efficient access and handover authentication protocol for Internet of Things in space information networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5485–5499, Jun. 2019.

[8] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, "Edge-centric computing: Vision and challenges," *Comput. Commun. Rev.*, vol. 45, no. 5, pp. 37–42, 2015.

[9] Z. Liu, L. Wan, J. Guo, F. Huang, X. Feng, L. Wang, and J. Ma, "PPRU: A privacy-preserving reputation updating scheme for cloud-assisted vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 1, no. 2, pp. 1–16, Dec. 2024.

[10] R.-F. Liao, H. Wen, J. Wu, F. Pan, A. Xu, H. Song, F. Xie, Y. Jiang, and M. Cao, "Security enhancement for mobile edge computing through physical layer authentication," *IEEE Access*, vol. 7, pp. 116390–116401, 2019.

[11] X. Jia, D. He, N. Kumar, and K. R. Choo, "A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing," *IEEE Syst. J.*, vol. 14, no. 1, pp. 560–571, Mar. 2020.

[12] B. D. Deebak, F. Al-Turjman, and L. Mostarda, "Seamless secure anonymous authentication for cloud-based mobile edge computing," *Comput. Electr. Eng.*, vol. 87, Oct. 2020, Art. no. 106782.

[13] S. Hussain, K. Mahmood, M. K. Khan, C.-M. Chen, B. A. Alzahrani, and S. A. Chaudhry, "Designing secure and lightweight user access to drone for smart city surveillance," *Comput. Standards Inter.*, vol. 80, Mar. 2022, Art. no. 103566.

[14] Y. Li, Q. Cheng, X. Liu, and X. Li, "A secure anonymous identity-based scheme in new authentication architecture for mobile edge computing," *IEEE Syst. J.*, vol. 15, no. 1, pp. 935–946, Mar. 2021.

[15] K. Kaur, S. Garg, G. Kaddoum, M. Guizani, and D. N. K. Jayakody, "A lightweight and privacy-preserving authentication protocol for mobile edge computing," in *Proc. IEEE Global Commun. Conf.*, Dec. 2019, pp. 1–6.

[16] C. Wang, Y. Zhang, X. Chen, K. Liang, and Z. Wang, "SDN-based handover authentication scheme for mobile edge computing in cyber-physical systems," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8692–8701, Oct. 2019.

[17] Z. Wang, Y. Zhou, Z. Qiao, B. Yang, C. Gu, Y. Xu, and M. Zhang, "An anonymous and revocable authentication protocol for vehicle-to-vehicle communications," *IEEE Internet Things J.*, vol. 10, no. 6, pp. 5114–5127, Mar. 2023.

[18] P. Gope, Y. Gheraibia, S. Kabir, and B. Sikdar, "A secure IoT-based modern healthcare system with fault-tolerant decision making process," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 3, pp. 862–873, Mar. 2021.

[19] P. Gope and B. Sikdar, "An efficient privacy-preserving authenticated key agreement scheme for edge-assisted Internet of Drones," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13621–13630, Nov. 2020.

[20] S. Roy, D. Das, A. Mondal, M. H. Mahalat, B. Sen, and B. Sikdar, "PLAKE: PUF-based secure lightweight authentication and key exchange protocol for IoT," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8547–8559, May 2023.

[21] S.-W. Lee, M. Safkhani, Q. Le, O. H. Ahmed, M. Hosseinzadeh, A. M. Rahmani, and N. Bagheri, "Designing secure PUF-based authentication protocols for constrained environments," *Sci. Rep.*, vol. 13, no. 1, p. 21702, Dec. 2023.

[22] S. A. Chaudhry, K. Yahya, S. Garg, G. Kaddoum, M. M. Hassan, and Y. B. Zikria, "LAS-SG: An elliptic curve-based lightweight authentication scheme for smart grid environments," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1504–1511, Feb. 2023.

[23] S. Zhang, Y. Liu, Y. Xiao, and R. He, "A trust based adaptive privacy preserving authentication scheme for VANETs," *Veh. Commun.*, vol. 37, Oct. 2022, Art. no. 100516.

[24] Y. Ma, X. Li, W. Shi, and Q. Cheng, "STCLA: An efficient certificateless authenticated key agreement scheme for the Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 73, no. 4, pp. 4830–4841, Apr. 2024.

[25] Y. Liang, E. Luo, and Y. Liu, "Physically secure and conditional-privacy authenticated key agreement for vanets," *IEEE Trans. Veh. Technol.*, 2023.

[26] F. Zerrouki, S. Ouchani, and H. Bouarfa, "PUF-based mutual authentication and session key establishment protocol for IoT devices," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 9, pp. 12575–12593, Sep. 2023.

[27] A. Irshad, M. Sher, H. F. Ahmad, B. A. Alzahrani, S. A. Chaudhry, and R. Kumar, "An improved multi-server authentication scheme for distributed mobile cloud computing services," *KSII Trans. Internet Inf. Syst. (TIIS)*, vol. 10, no. 12, pp. 5529–5552, 2016.

[28] L. Xiong, D. Peng, T. Peng, and H. Liang, "An enhanced privacy-aware authentication scheme for distributed mobile cloud computing services," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 12, pp. 1–20, 2017.

[29] X. Yang, X. Huang, and J. K. Liu, "Efficient handover authentication with user anonymity and untraceability for mobile cloud computing," *Future Gener. Comput. Syst.*, vol. 62, pp. 190–195, Sep. 2016.

[30] X. Li, T. Chen, Q. Cheng, and J. Ma, "An efficient and authenticated key establishment scheme based on fog computing for healthcare system," *Frontiers Comput. Sci.*, vol. 16, no. 4, pp. 1–12, Aug. 2022.

[31] J. Li, W. Zhang, V. Dabra, K.-K.-R. Choo, S. Kumari, and D. Hogrefe, "AEP-PPA: An anonymous, efficient and provably-secure privacy-preserving authentication protocol for mobile services in smart cities," *J. Netw. Comput. Appl.*, vol. 134, pp. 52–61, May 2019.

[32] E. Castelló Ferrer, "The blockchain: A new framework for robotic swarm systems," in *Proc. Future Technol. Conf.*, vol. 2, 2018, pp. 1037–1058.

[33] S. O. Ogundoyin, "An autonomous lightweight conditional privacy-preserving authentication scheme with provable security for vehicular ad-hoc networks," *Int. J. Comput. Appl.*, vol. 42, no. 2, pp. 196–211, Feb. 2020.

[34] T. Nandy, M. Y. I. Idris, R. M. Noor, A. K. Das, X. Li, N. A. Ghani, and S. Bhattacharyya, "An enhanced lightweight and secured authentication protocol for vehicular ad-hoc network," *Comput. Commun.*, vol. 177, pp. 57–76, Sep. 2021.

[35] C. Lin, D. He, X. Huang, N. Kumar, and K. R. Choo, "BCPPA: A blockchain-based conditional privacy-preserving authentication protocol for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7408–7420, Dec. 2021.

[36] J. Miao, Z. Wang, X. Ning, N. Xiao, W. Cai, and R. Liu, "Practical and secure multifactor authentication protocol for autonomous vehicles in 5G," *Softw., Pract. Exper.*, vol. 52, no. 4, pp. 35–42, Apr. 2022.

[37] E. Praveen Kumar and S. Priyanka, "A comprehensive survey on hardware-assisted malware analysis and primitive techniques," *Comput. Netw.*, vol. 235, Nov. 2023, Art. no. 109967.

[38] T. Alladi, S. Chakravarty, V. Chamola, and M. Guizani, "A lightweight authentication and attestation scheme for in-transit vehicles in IoV scenario," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14188–14197, Dec. 2020.

[39] S. A. Chaudhry, M. S. Farash, N. Kumar, and M. H. Alsharif, "PFLUA-DIoT: A pairing free lightweight and unlinkable user access control scheme for distributed IoT environments," *IEEE Syst. J.*, vol. 16, no. 1, pp. 309–316, Mar. 2022.

[40] Z. Liu, J. Guo, F. Huang, D. Cai, Y. Wu, X. Chen, and K. K. Igorevich, "Lightweight trustworthy message exchange in unmanned aerial vehicle networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2144–2157, Feb. 2023.

[41] Z. Liu, J. Weng, J. Guo, J. Ma, F. Huang, H. Sun, and Y. Cheng, "PPTM: A privacy-preserving trust management scheme for emergency message dissemination in space-air-ground-integrated vehicular networks," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5943–5956, Apr. 2022.

[42] M. A. Abdel-Malek, K. Akkaya, A. Bhuyan, and A. S. Ibrahim, "A proxy signature-based swarm drone authentication with leader selection in 5G networks," *IEEE Access*, vol. 10, pp. 57485–57498, 2022.

[43] K. Goutsos, "Physical unclonability framework for the Internet of Things," Ph.D. dissertation, Fac. Sci., Agricult. Eng., Newcastle Univ., Newcastle upon Tyne, England, 2020.

[44] C. Aitchison, R. Buckle, A. Ch'ng, C. Clarke, J. Malley, and B. Halak, "On the integration of physically unclonable functions into ARM TrustZone security technology," in *Proc. Eur. Conf. Circuit Theory Design (ECCTD)*, Sep. 2020, pp. 1–4.

[45] M. Norouzi, D. J. Fleet, and R. R. Salakhutdinov, "Hamming distance metric learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–18.

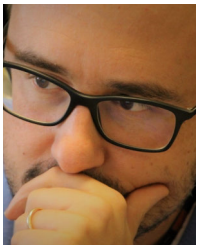
[46] P. Castoldi, A. Sgambelluri, L. Ismail, F. Paolucci, F. Cugini, and D. Bowden, "Network programmability for smart factory mobile robotics: The SmartEdge project approach," in *Proc. 23rd Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2023, pp. 1–19.

[47] M. Wazid, A. K. Das, N. Kumar, and A. V. Vasilakos, "Design of secure key management and user authentication scheme for fog computing services," *Future Gener. Comput. Syst.*, vol. 91, pp. 475–492, Feb. 2019.

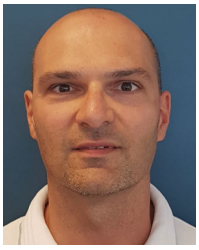
[48] AVISPA Project. (2017). *AVISPA: Automated Validation of Internet Security Protocols and Applications*. [Online]. Available: http://www.avispa-project.org/

[49] M. Abdalla, E. Bresson, O. Chevassut, B. Möller, and D. Pointcheval, "Provably secure password-based authentication in TLS," in *Proc. ACM Symp. Inf., Comput. Commun. Secur.*, Mar. 2006, pp. 35–45.
[50] C.-C. Chang and H.-D. Le, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 357–366, Jan. 2016.
[51] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "Cloud centric authentication for wearable healthcare monitoring system," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 5, pp. 942–956, Sep. 2020.
[52] M. Wazid, A. K. Das, V. Bhat K, and A. V. Vasilakos, "LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment," *J. Netw. Comput. Appl.*, vol. 150, Jan. 2020, Art. no. 102496.
[53] M. Kaveh and M. R. Mosavi, "A lightweight mutual authentication for smart grid neighborhood area network communications based on physically unclonable function," *IEEE Syst. J.*, vol. 14, no. 3, pp. 4535–4544, Sep. 2020.
[54] T. Limbasiya, M. Soni, and S. K. Mishra, "Advanced formal authentication protocol using smart cards for network applicants," *Comput. Electr. Eng.*, vol. 66, pp. 50–63, Feb. 2018.

**RANA ABU BAKAR** received the bachelor's degree in computer science from COMSATS University Islamabad, in 2015, and the master's degree in computer science from Virtual University, in 2018. He is currently pursuing the Ph.D. degree. He was a Research Scholar with Chulalongkorn Thailand, from 2020 to 2022. He has been involved in European research projects on Network Security (SmartEdge, CLEVER, DESIRE6G, and NATWORK). His research interests include software defined networking, datacenter networking, high-performance computing, confidentiality computing, homomorphic cryptography, authentication protocol development, GPU/DPU processing, network function virtualization, distributed systems, artificial intelligence and machine learning, and resilient edge platforms network security.



**FRANCESCO PAOLUCCI** is currently the Head of Research with CNIT, Pisa, Italy. His main research interests include network control plane, edge/cloud networking platforms, traffic engineering, network disaggregation, advanced monitoring/telemetry, and SDN data plane programmability. He has been involved in many industrial and European research projects on next-generation control networking (METRO-HAUL, B5G-OPEN, BRAINE, DESIRE6G, SmartEdge, and CLEVER). He is the coauthor of three IETF Internet Drafts, more than 200 publications in international journals, conference proceedings, book chapters, and filed four international patents. He is also an Associate Editor of the IEEE/OSA JOURNAL OF OPTICAL COMMUNICATIONS AND NETWORKING (JOCN) and an Executive Editor of the *Transactions on Emerging Telecommunications Technologies* (ETT).



**FILIPPO CUGINI** (Member, IEEE) is currently the Head of Research Sector with CNIT, Pisa, Italy. He is the coauthor of 14 patents and more than 300 international publications. His main research interests include theoretical and experimental studies in the field of communications and networking. In particular, the focus is on flexible optical networks, disaggregated solutions, software-defined networking (SDN), high-performance computing, photonic integrated networks, edge computing, and in-network programmability, including P4. He currently serves as a Project Coordinator of the EU-funded projects Big Data Processing and Artificial Intelligence at the Network Edge (BRAINE), Semantic Low-code Programming Tools for Edge Intelligence (SmartEdge), and Self-Managed Sustainable High-Capacity Optical Networks (SEASON).

**PIERO CASTOLDI** (Senior Member, IEEE) is currently a Full Professor in telecommunications with Scuola Superiore Sant'Anna, Pisa. He is also the Director of the TeCIP Institute (2022–2025). He teaches many courses in the area of telecommunication networks. He has been the Director of the Erasmus Mundus Master on Photonic Integrated Circuits, Sensors and Networks (PIXNET). His research interests include telecommunication networks and systems, in-networks programming, and photonic integrated networks. He is also responsible for the area of networks and services with the TeCIP Institute of Scuola Sant'Anna. He has served as Executive Editor for the *Transactions on Emerging Telecommunication Technologies* (ETT) (Wiley). He also works on technology transfer with many primary companies, such as Telecom Italia, Rete Ferroviaria Italiana, and Schindler Company.

**JUAN JOSE VEGAS OLMOS** received the B.Sc. and M.Sc. degrees in telecommunications and electronic engineering, in 2001 and 2003, respectively, the Ph.D. degree from Eindhoven University of Technology, The Netherlands, in 2006, the B.A. degree in business administration, the M.B.A. degree, and the M.A. degree in east Asian studies. He was a Research Fellow with Osaka University, Japan, from 2006 to 2008, and a Research Associate with the Central Research Laboratory, Hitachi Ltd., until 2011. He is also a Research Program Coordinator with NVIDIA Corporation in the area of software architecture.

Open Access funding provided by 'Scuola Superiore "S.Anna" di Studi Universitari e di Perfezionamento' within the CRUI CARE Agreement