BAZAM: A Blockchain-Assisted Zero-Trust Authentication in Multi-UAV Wireless Networks

Mingyue Xie, Zheng Chang, *Senior Member, IEEE*, Alain R. Ndjiongue, *Senior Member, IEEE*, Tao Chen, *Member, IEEE*, and Hongwei Li, *Fellow, IEEE*

***Abstract*—Unmanned aerial vehicles (UAVs) are vulnerable to interception and attacks when operated remotely without a unified and efficient identity authentication. Meanwhile, the openness of wireless communication environments potentially leads to data leakage and system paralysis. However, conventional authentication schemes in the UAV network are centered on the fixed trust boundary, ignoring potential internal threats and failing to flexibly respond to the dynamic requirements of UAV access and identity authentication. Additionally, UAVs are not subjected to periodic repetitive identity authentication, leading to difficulties in controlling access anomalies. Therefore, in this work, we consider a zero-trust framework for UAV network authentication, aiming to achieve UAV identity authentication through the principle of "never trust and always verify." We introduce a blockchain-assisted zero-trust authentication scheme, namely, BAZAM, designed for multi-UAV wireless networks. In this scheme, UAVs follow a key generation approach using physical unclonable functions (PUFs), and cryptographic technique helps verify registration and access requests of UAVs. The blockchain is applied to store UAVs authentication-related information in immutable storage. Through thorough security analysis and extensive evaluation, we demonstrate the effectiveness and efficiency of the proposed BAZAM.**

***Index Terms*—Authentication, blockchain, physical unclonable functions (PUFs), unmanned aerial vehicle (UAV) network, zero-trust.**

I. Introduction

A. Background

IN THE next-generation networking paradigm, unmanned aerial vehicles (UAVs)-based wireless networks, benefiting from the rapid and easily deployable characteristics [1], have been extensively applied across various domains. Due to the deployment in open environments and reliance on wireless communication channels, UAVs are susceptible to interception and attacks, such as replay attacks, impersonation attacks, and physical capture attacks [2]. These attacks have the potential to disrupt the normal operation of UAVs, allowing attackers to illicitly obtain resources from internal networks. Therefore, secure UAV authentication technologies play a pivotal role in effectively curbing adversarial attacks [3].

However, due to the characteristics of multi-UAV wireless networks, it is challenging to apply the conventional network security model and scheme directly to design authentication schemes of UAV networks [4]. On one hand, the high-speed mobility of UAVs leads to rapid changes in network conditions and connection status, necessitating the repeated verification of their identities [5]. On the other hand, an excessive trust in UAVs within the internal network makes the system difficult to effectively defend against internal malicious UAVs [6]. Moreover, traditional authentication approaches primarily relying on biometrics update, passwords, and smart cards are susceptible to theft or forgery by attackers [7], thereby enabling unauthorized access to internal network resources. Additionally, these approaches typically grant prolonged access privileges to UAVs after authentication, leading internal network resources leakage. While boundary-based multi-UAV wireless network protection can effectively thwart external attacks, it cannot prevent attacks initiated by malicious UAVs within the internal network [8]. Consequently, conventional multi-UAV wireless network architectures still face significant security challenges.

A novel network security architecture known as zero-trust [9], shows its potential to be an appropriate solution for UAV authentication. The zero-trust model posits that both the internal and external aspects of wireless networks are untrusted [10]. In other words, in the zero-trust network, UAVs are deemed untrusted until thoroughly validated and authenticated. Furthermore, the zero-trust model aims to transform traditional network architectures from being system-centric to user-centric with continuous authentication [11]. When a UAV needs wireless or other resources, its identity is initially authenticated and assessed to determine its access permissions [12]. Subsequently, the zero-trust gateway allows or denies UAVs access to service resources and monitors their access behavior in real time. The zero-trust network protection shifts from the network perimeter to the user identity itself, effectively preventing unauthorized access requests from within the network. For multi-UAV wireless networks, the zero-trust network architecture can provide a higher level of security protection, safeguarding sensitive data from unauthorized access and potential security threats. Furthermore, recognizing the fluctuating scale and mobility of

Received 2 July 2025; accepted 19 August 2025. Date of publication 25 August 2025; date of current version 7 November 2025. This work was supported in part by the NSF of Sichuan under Grant 2025YFHZ0093, and in part by the Horizon European Union under Grant 101086159 and Grant 101131117. *(Corresponding author: Zheng Chang.)*

Mingyue Xie, Zheng Chang, Tao Chen, and Hongwei Li are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: mingyueyue@std.uestc.edu.cn; zheng.chang@jyu.fi; chent@uestc.edu.cn; hongweili@uestc.edu.cn).

Alain R. Ndjiongue is with the School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg 2050, South Africa (e-mail: richard.ndjiongue@wits.ac.za).

Digital Object Identifier 10.1109/JIOT.2025.3602114

2327-4662 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Witwatersrand. Downloaded on November 07,2025 at 06:01:55 UTC from IEEE Xplore. Restrictions apply.

UAVs, blockchain technology has recently been employed to securely store the UAV-related authentication information in a distributed manner [13]. This approach helps enhance the scalability of authentication methods and facilitates their easy deployment on resource-constrained UAVs.

B. Related Work

Recently, considerable efforts have been dedicated to the research of security and threat toward UAVs. Several studies [14], [15] indicated that attackers may attack open wireless channels, potentially engaging in illicit interception of communication messages, and even manipulating the dissemination of messages through forgery and control. If attackers gain physical access to a UAV and tamper with its authentication parameters, the legitimate UAV in the network may be mistakenly identified as malicious [16]. Therefore, several cryptographic frameworks have been proposed for UAV authentication in recent years. Tan et al. [17] applied elliptic curve cryptography (ECC) to design a secure authentication scheme for UAV communications. However, the parameter size of ECC will impose significant limitations and overhead on resource-constrained UAVs. In [18], a hybrid key exchange authentication model was introduced to create an efficient, lightweight authentication model for UAV-assisted applications. In [19], a lightweight mutual authentication protocol for UAVs was designed. It is solely comprised of bitwise operations and one-way hash functions. However, it cannot prevent severe security attacks, such as "session key impersonation" and lacks "mutual authentication." Another popular approach in UAV authentication is to leverage physical layer attributes. Huang et al. [20] proposed a UAV authentication and key agreement scheme, incorporating physically unclonable function (PUF) to counteract physical capture attacks. To achieve UAV authentication with built-in privacy-preserving, Yu et al. [21] introduced a secure authentication protocol. This approach allows PUF to generate secret keys, ensuring no sensitive information is stored in memory. In [22], a PUF and fuzzy extractor-based mutual authentication mechanism was designed for UAV and ground server (GS). However, these approaches are still constrained by traditional architectures based on static authentication frameworks. Such systems grant full privileges post-initial legal/illegal validation, without mechanisms for continuous authentication.

There have been increasing interests in authentication using zero-trust. To reduce the risk of unauthorized access by malicious users, Lee et al. [23] developed a zero-trust authentication protocol based on behavioral biometric features. Ajakwe et al. [24] realized the ownership authentication and operation authorization of UAVs through the incorporation of zero-trust architecture into the blockchain and the artificial intelligence models. Ge and Zhu [25] constructed a zero-trust authentication framework based on Markov games with one-sided information, which leverages interdependent trust evaluation to automate zero-trust security. In [26], a handover and continuous authentication scheme was presented in zero-trust Internet of vehicles to satisfy higher security level. Guo et al. [27] combined an authentication model with the continuous assessment mechanism of the zero-trust architecture to enhance the safety and efficiency of UAV-assisted communications. In [28], a dynamic zero-trust federated learning framework was proposed. It employs a set of zero trust information analysis techniques to prevent malicious attacks from compromising the system.

Therefore, this article adopts a zero-trust architecture to enable continuous identity authentication for UAVs, thereby addressing the coarse-grained limitations of traditional authentication mechanisms. Moreover, a blockchain-based reputation model is proposed to identify and filter UAVs that have already been frequently and legitimately authenticated, which significantly enhances the overall authentication efficiency.

C. Motivation and Contributions

As mentioned above, existing UAV wireless networks suffer from potential security attacks. In this context, identity-based authentication is important to improve the multi-UAV network security. Taking into account the dynamics of UAV network, as well as the intranet trust and security of authentication approaches, the zero-trust model can establish a robust and adaptable UAV authentication scheme. Therefore, this article proposes a blockchain-assisted authentication scheme, called BAZAM, for multi-UAV wireless networks under the zero-trust framework. The main contributions of this article are summarized as follows.

1) We propose a blockchain-assisted UAV authentication scheme specifically designed to operate in the zero-trust network. The zero-trust network centers around the UAV identity, which is authenticated and evaluated by the software defined perimeter (SDP) controller before allowing resource access requests from a UAV.
2) We utilize the randomness and nonreplicability on the UAV hardware to generate its secret key. Based on the PUF, we present a lightweight cryptographic technique as identity authentication to design the BAZAM. The SDP controller can verify the identity and request of a UAV before sending a policy to the gateway to allow UAV access.
3) To enhance the authentication efficiency of the SDP controller in the zero-trust framework, we introduce a reputation to filter UAVs' eligibility for authentication preparation, taking into account the legitimate authentication attempts made by the UAVs.
4) We introduce a blockchain-assisted authentication service for UAVs. The reputation required for the authentication is stored in the blockchain. The untamperable record increases the reliability and security of reputation and reduces the risk of single points of failure and centralization in zero-trust network.

The remainder of this article is organized as follows. Section II presents the basic concepts. We describe the system model, threat model, security goals and adversary model in Section III. The detailed constructions of the proposed scheme are shown in Section IV. Sections V and VI demonstrates the security analysis and performance analysis. Section VII concludes the research work.

Authorized licensed use limited to: University of Witwatersrand. Downloaded on November 07,2025 at 06:01:55 UTC from IEEE Xplore. Restrictions apply.

II. Preliminaries

In this section, we introduce the following concepts for the construction of our scheme.

A. Intractable Problem

As the intractability assumptions of the hard problems is the theoretical basis for the later proof, we utilize a slight modification presented in [29] to provide a security basic for our BAZAM scheme.

Definition 1 (Modified Bilinear Diffie–Hellman (MBDH) Assumption): Given a cyclic additive group $\mathbb{G}_1$ and a cyclic multiplicative group $\mathbb{G}_2$ with the same order q, a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, and a generator G of $\mathbb{G}_1$, for a tuple $(G, aG, bG, cG, c^{-1}G)$, $z \in \mathbb{G}_2$, a, b, c are unknown integers, MBDH assumption is to compute the value of $e(G, G)^{abc}$.

Definition 2 [Modified Decisional Bilinear Diffie–Hellman (MDBDH) Assumption]: Given a cyclic additive group $\mathbb{G}_1$ and a cyclic multiplicative group $\mathbb{G}_2$ with the same order q, a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, and a generator G of $\mathbb{G}_1$, for a tuple $(G, aG, bG, cG, c^{-1}G)$, $z \in \mathbb{G}_2$, a, b, c are unknown integers, MDBDH assumption is to decide whether $z = e(G, G)^{abc}$.

B. Software Defined Perimeter

The conventional network security models typically rely on static boundary defenses, such as firewalls and virtual private networks, which lack adequate flexibility and security [30]. Therefore, the technological system in the zero-trust architecture is SDP. The typical SDP architecture consists of SDP controller, SDP client, and SDP gateway. The SDP controller is responsible for validating packet sent by the SDP client, issuing policies, managing the SDP gateway, and configuring the connection between the SDP client and SDP gateway. The SDP client, a software or agent on user devices, is tasked with communicating with the SDP controller for identity verification and access control. The SDP gateway, located at the network edge, by default denies all access requests. Only upon successful authentication of a request and the issuance of corresponding credentials and policies by the SDP controller, the SDP gateway establishes a connection with the SDP client.

III. System Model and Problem Formulation

In this section, we present the system model of BAZAM, and formulate the threat model, security goals, and adversary model.

A. System Model

In this article, we propose BAZAM, a zero-trust multi-UAV authentication scheme based on blockchain, which combines blockchain, cryptography and SDP technology to authenticate a UAV access request, so as to achieve fine-grained authentication and authorisation of UAVs and ensure the security of service resources. To demonstrate the applicability of BAZAM in zero-trust architecture, we suggest the proposed infrastructure that can be divided into five types of essential units with distinctive functionalities: 1) the key generation center (KGC); 2) the UAV; 3) the SDP controller; 4) the SDP gateway; and 5) the blockchain, as shown in Fig. 1.

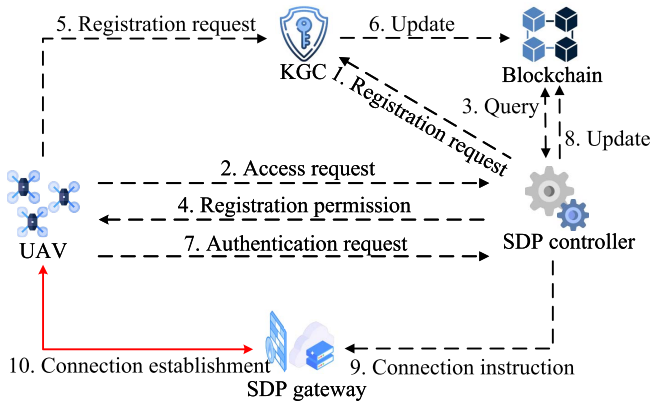

Fig. 1. System model of BAZAM.

1) *KGC*: KGC possesses robust communication and computational capabilities. It is responsible for facilitating the registration of UAVs and SDP controller in the zero-trust model, as well as assisting in the authentication process between them. The registration processes of UAVs and SDP controller with KGC occur over a secure channel. Moreover, KGC is deemed entirely trustworthy and impenetrable.
2) *UAV*: UAV assumes the role of data resource requester. Each UAV is equipped with a PUF. Upon submitting a resource access request to the SDP controller and receiving registration approval, the UAV proceeds to request its public/ private key pair from the KGC. Subsequently, the UAV utilizes its own private key and the public key of the SDP controller to encrypt and sign the single page application (SPA) packet. Then, it presents an authentication request to the SDP controller with the ciphertext. Upon successful authentication, the UAV can establish a secure connection with the SDP gateway, thereby gaining access to the resources.
3) *SDP Controller*: SDP controller is utilized to manage and control access to network resources. The primary role of the SDP controller is to dynamically determine, based on the UAV identity and packet information, that only authenticated and authorized UAV has permission to access resources. Additionally, the SDP controller considers the reputation and the corresponding thresholds of the UAV to decide whether to grant registration rights to the UAV.
4) *SDP Gateway*: When receiving a request from a UAV, the SDP gateway determines, based on the authorization policy and authentication result provided by the SDP controller, whether to allow the request. If the request is granted, the SDP gateway establishes a secure connection, enabling the UAV to securely access the required resources.
5) *Blockchain*: Blockchain is a shared distributed ledger, that consists of immutable and trustworthy information of UAVs. Blockchain is critical for SDP controller

Authorized licensed use limited to: University of Witwatersrand. Downloaded on November 07,2025 at 06:01:55 UTC from IEEE Xplore. Restrictions apply.

to process the registration requests and enforce UAV authentication.

In Fig. 1, BAZAM works with the following steps for authenticating the UAVs and establishing the connection between the UAVs and the gateway. First, the SDP controller initiates a registration request to the KGC to obtain its own key pairs. Subsequently, the UAV sends an access request to the SDP controller. The SDP controller determines the eligibility of the UAV registration by querying the UAV information on the blockchain. Upon receiving registration permission, the UAV sends a registration request to the KGC to generate a unique public/private key pair, while the KGC updates the UAV's information on the blockchain. Following this, the UAV generates packet data and sends an authentication request to the SDP controller. After verifying the packet and confirming the identity of the UAV, the SDP controller updates the UAV information on the blockchain for subsequent authentication. Upon successful authentication, the SDP gateway receives connection instruction from the SDP controller, allowing it to open corresponding ports for the UAV and grant access to the relevant service resources.

B. Threat Model

The formalization of the adversary model is based on Dolev–Yao threat framework [31]. Based on the security threats faced by UAVs in the process of data transmission, the adversary in the process of identity authentication of UAVs is defined. The adversary has the following capabilities.

When the sensitive data is exchanged over the UAV network, we assume that an adversary might employ sniffing tools to eavesdrop on the packets, thereby gaining access to the communication content. Additionally, an adversary could intercept messages sent from UAVs, forging different messages to be sent to the SDP controller, executing man-in-the-middle attack. Furthermore, an adversary may illicitly obtain secret information stored within UAVs, involving unauthorized access to confidential data stored internally. In the zero-trust network, an adversary has the capability to send messages to all entities, allowing them to impersonate UAVs within the network and send deceptive messages. However, an adversary lacks the ability to control the trusted KGC, and it cannot obtain the system's random numbers through guessing.

C. Security Goals

To fulfil the resource access requirements of multi-UAV in wide-area airspace, and to establish secure communication links between UAVs and zero-trust networks, the access authentication scheme for UAVs within the zero-trust architecture should achieve the following security goals.

Session Key Establishment: BAZAM should generate unique session keys for the authentication process. In addition, it should ensure that an adversary is unable to obtain intelligence from the captured session key.

Integrity: BAZAM should perform the verification on the packets origin, and ensure that the data are free from deliberate or inadvertent unauthorized manipulation or modification.

Forward and Backward Secrecy: Even if adversaries obtain the master key or secrets, they should not be able to learn previous session keys established between the UAV and the SDP controller. Additionally, the newly joined UAVs are prevented to learning previous session keys.

Secure Against Cyber Attacks: BAZAM should be secure against common security attacks, such as replay attack, node tampering, etc.

Secure Against Physical Attacks: The adversaries may attempt to create replicas of legitimate UAVs through cloning attacks, as well as physically tamper with UAV identifiers or data in memory. Therefore, BAZAM should be secure against UAV cloning attacks and other physical attacks, such as UAV capture and tampering.

D. Adversary Model

As proof that our BAZAM scheme satisfies confidentiality and the unforgeability, we employed the security notions in [32] to provide the security games for our BAZAM scheme.

Definition 3 [Indistinguishability Against Adaptive Chosen-Ciphertext Attack (IND-CCA2)]: Our scheme is IND-CCA2 secure if no adversary $\mathcal{A}$ possesses a nonnegligible advantage in winning the following game.

Initial: A challenger $\mathcal{C}$ initializes the system parameters and master key. Then, the parameters are forwarded to $\mathcal{A}$.

Phase 1 and 2: $\mathcal{A}$ adaptively asks to $\mathcal{C}$ the following the queries.

1) *Key Extract Queries:* $\mathcal{A}$ submits key extract queries for identities of the UAV and SDP controller. $\mathcal{C}$ sends the corresponding private keys to $\mathcal{A}$.
2) *Signcrypt Queries:* $\mathcal{A}$ chooses the identities of the UAV and SDP controller and a packet. $\mathcal{C}$ obtains the private key of the UAV for signcrypting the packet and sends it to $\mathcal{A}$.
3) *Unsigncrypt Queries:* $\mathcal{A}$ chooses the identities of the UAV and SDP controller and a ciphertext. $\mathcal{C}$ obtains the private key of the SDP controller for decrypting the ciphertext and sends it to $\mathcal{A}$.

Challenge: $\mathcal{A}$ adaptively asks to $\mathcal{C}$ the aforementioned queries. When the Phase 1 is finished, $\mathcal{A}$ generates two equal-size messages and the challenging identities of the UAV and SDP controller to $\mathcal{C}$. $\mathcal{C}$ randomly selects $\gamma \in \{0, 1\}$ to make the challenging ciphertext and forwards it to $\mathcal{A}$.

Guess: $\mathcal{A}$ outputs a guess γ' and wins this game if $\gamma' = \gamma$.

Definition 4: [Existential Unforgreability Against Chosen-Message Attack (EUF-CMA)]: Our scheme is EUF-CMA secure if no adversary $\mathcal{A}$ possesses a nonnegligible advantage in winning the following game.

Initial: A challenger $\mathcal{C}$ initializes the system parameters and master key. Then, the parameters is forwarded to $\mathcal{A}$.

Attack: $\mathcal{A}$ proposes various queries adaptively as described in the previous definition. $\mathcal{C}$ is able to answer the queries.

Forgery: $\mathcal{A}$ outputs the forged ciphertext and the identities of a UAV and an SDP controller. It is worth noting that the identities and packet cannot be queried in the attack phase. If the forged ciphertext is recognized as valid, $\mathcal{A}$ wins the game.

Authorized licensed use limited to: University of Witwatersrand. Downloaded on November 07,2025 at 06:01:55 UTC from IEEE Xplore. Restrictions apply.

TABLE I
NOTATION AND DESCRIPTION

Notations	Descriptions
ssk	System public key
spk	System master key
ID_C	The identity of SDP controller
PK_C/SK_C	The public/private key of ID_C
U_i	The i-th UAV
ID_U^i	The identity of U_i
Rep_i	The reputation of U_i
R_h/R_l	The threshold of reputation
(C_i, R_i)	The challenge–response pair of U_i
PK_U^i/SK_U^i	The public/private key of U_i
pac	An SPA packet generated by U_i
σ	The ciphertext of pac generated by U_i

IV. BAZAM: ZERO-TRUST AUTHENTICATION SCHEME

The proposed BAZAM is organized into three phases, namely, the initialize phase, the registration phase, and the UAV-SDP controller authentication phase. Table I summarizes the notations used in this article.

A. Initialization

In the proposed authentication phase, UAVs are authenticated under the coordination of SDP controller and KGC. Before starting the process, the system is assumed to have been initialized.

1) System parameters have been instantiated. Let $\mathbb{G}_1$ be a cyclic additive group generated by G, and $\mathbb{G}_2$ be a cyclic multiplicative group. $\mathbb{G}_1$ and $\mathbb{G}_2$ have the same order p, where p is a primer number of k-bits length. e: $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear map. Then, H_1, H_2, and H_3 denote the secure one-way hash functions with the definitions as follows: H_1: $\{0,1\}^* \rightarrow \mathbb{G}_1$, H_2: $\mathbb{G}_2 \rightarrow \{0,1\}^l$, and H_3: $\{0,1\}^l \times \mathbb{G}_2 \rightarrow \mathbb{Z}_p^*$, where l denotes a bit length of a message. Let Enc and Dec be the encryption and decryption algorithms of a secure symmetric cipher.
2) System public key and master key have been generated. It first chooses randomness $ssk \in \mathbb{Z}_p^*$ to compute $spk = sskG$. Then, the system selects n KGCs to participate in the registration process, and divides the ssk into n parts as $ssk = \sum_{j=1}^{n} ssk_j$. It is noticing that ssk_j is kept secretly and known only by the jth KGC.

B. SDP Controller Registration

As illustrated in Fig. 2, the public/private key pair of the SDP controller ID_C is generated by KGCs in this phase. The public key of ID_C generated by the jth KGC is as follows:

$$PK_C^j = H_1(\mathrm{ID}_C).$$

The corresponding private key is computed as

$$SK_C^j = ssk_j PK_C^j.$$

Then, the jth KGC sends $\{PK_C^j, SK_C^j\}$ to SDP controller. Note that, the public key of ID_C is finally shared with the UAVs that submit the registration request to gateway. When t private keys are received from KGCs, the public key $PK_C = \sum_{j=1}^{t} PK_C^j$ and the private key $SK_C = \sum_{j=1}^{t} SK_C^j$ are stored by SDP controller by summing all the corresponding value.

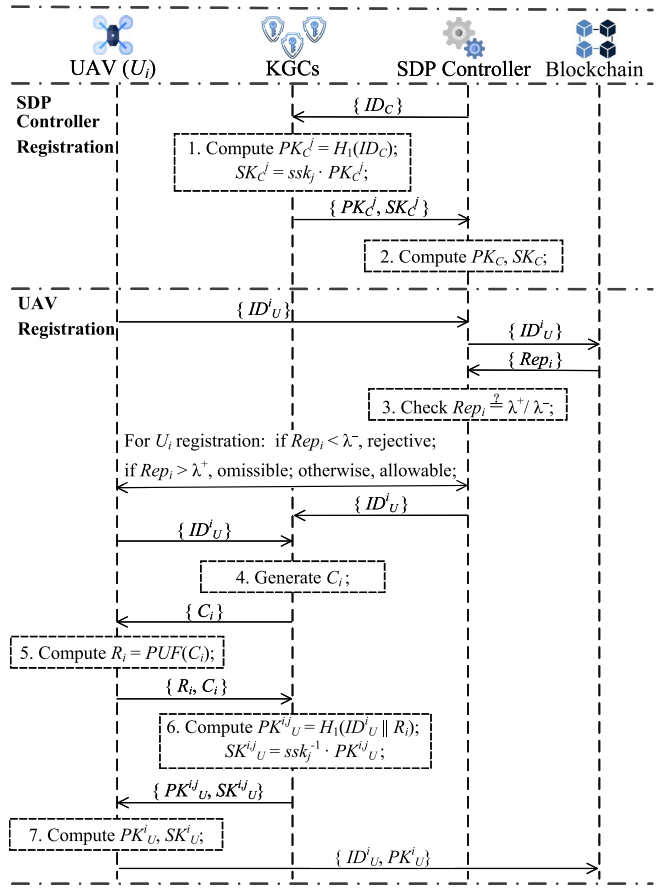

Fig. 2. Registration process of a UAV and an SDP controller.

C. UAV Registration

In the registration phase, each UAV is equipped with a PUF for generating a response output for a challenge input to it. Considering that a malicious UAV is able to request authentication from the system, it could result in a significant increase in local traffic and lead to device failures. To mitigate the occurrence of such situations, the system incorporates a reputation mechanism in the registration phase. Fig. 2 presents the registration process of a UAV. The key steps of a UAV registration process are explained as follows.

1) When UAV U_i with identity ID_U^i enters the network for the first time, it first establishes a registration request with the SDP controller via a secure and private channel.
2) Upon receiving the registration request, the SDP controller first retrieves the reputation of the UAV on the blockchain. The SDP controller will initiate a retrospective search from the latest block, proceeding backward until locating the block containing pertinent information about U_i. This approach ensures the retrieval of the latest UAV information, including its reputation and public key, etc. For U_i with its reputation Rep_i below a threshold R_l, the registration request will be rejected by the SDP controller,

Authorized licensed use limited to: University of Witwatersrand. Downloaded on November 07,2025 at 06:01:55 UTC from IEEE Xplore. Restrictions apply.

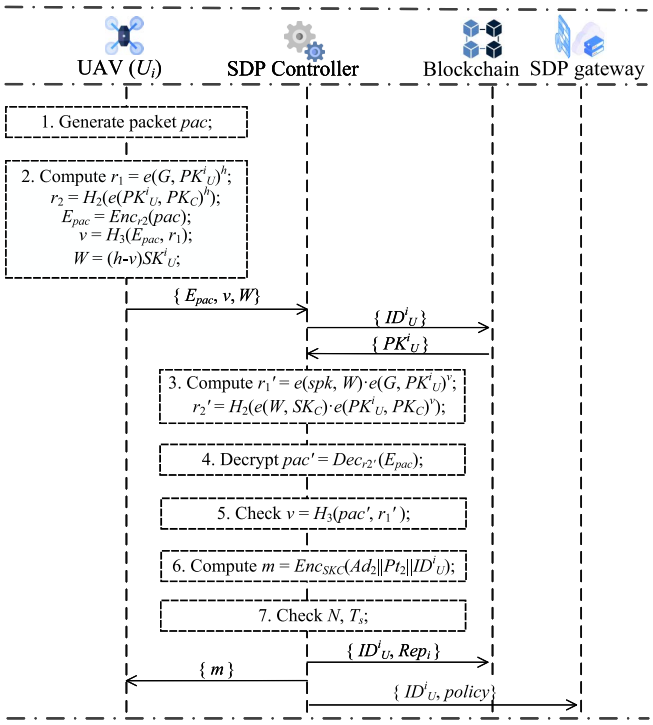

Fig. 3. UAV-SDP controller authentication process.

and registration services will not be provided. For U_i with Rep_i above a threshold R_h, it will receive a notification from the SDP controller indicating no registration is required. Otherwise, the SDP controller sends registration request to the KGCs with ID^i_U of the U_i.

3) Upon receiving the valid registration request from U_i with its identity, a random challenge C_i is generated by KGCs and sent to U_i. After receiving the C_i, based on which a response, R_i, is produced applying the PUF as

$$R_i = PUF(C_i).$$

Then, U_i sends $\{C_i, R_i\}$ to the KGC.

4) When KGCs receive R_i from U_i, the jth KGC generates the public key $PK^{i,j}_U$ for U_i as

$$PK^{i,j}_U = H_1\left(\mathrm{ID}^i_U||R_i\right).$$

Then, the corresponding private key is established as

$$SK^{i,j}_U = ssk_j^{-1} \cdot PK^{i,j}_U.$$

The KGC sends $PK^{i,j}_U$ and $SK^{i,j}_U$ to U_i.

5) When t private keys are received from KGCs, the public key $PK^i_U = \sum_{j=1}^{t} PK^{i,j}_U$ and the private key $SK^i_U = \sum_{j=1}^{t} SK^{i,j}_U$ are computed. Then, $\{\mathrm{ID}^i_U, PK^i_U\}$ is sent to blockchain.

It should be noted that the set $\{\mathrm{ID}^i_U, PK^i_U, SK^i_U\}$ is securely stored in the UAV's storage. Meantime, when a UAV registers for the first time, its reputation, denoted as Rep_i, is initialized to a value between R_l and R_h, and then uploaded to the blockchain. For UAVs with a prior registration history, their existing reputation remains unchanged during this phase. Consequently, upon completion of the registration phase, the blockchain will contain the tuple $<\mathrm{ID}^i_U, PK^i_U, Rep_i>$ for each registered UAV U_i.

D. UAV-SDP Controller Authentication

Here, we describe our scheme to achieve the authentication between UAV U_i and SDP controller under the zero trust network. Fig. 3 depicts the various authentication phase performed on U_i and SDP controller. This scheme ensures that only registered UAVs can authenticate with SDP controller before they can begin establishing a secure connection with the SDP gateway.

1) Prior to the initiation of a connection request by U_i, rigorous identity authentication must be conducted through the SDP controller.
   a) First, U_i generates an SPA packet based on its own information and requirements as

$$pac = \left(N||\mathrm{ID}^i_U||Pwd||Ts||Ver||Ad_1||Pt_1\right)$$

   where N denotes the nonce number, Pwd denotes the password, Ts denotes the timestamp, Ver denotes the protocol version, Ad_1 denotes the IP address, and Pt_1 denotes the port number.
   b) U_i selects a random number $h \in \mathbb{Z}_p^*$ to derive k_1 and k_2 as

$$r_1 = e\left(G, PK^i_U\right)^h$$
$$r_2 = H_2\left(e(PK^i_U, PK_C)^h\right).$$

   c) Based on the pac, r_1, and r_2, U_i performs the following operations to generate the ciphertext $\sigma = (E_{pac}, v, W)$ as

$$E_{pac} = Enc_{r_2}(pac)$$
$$v = H_3\left(E_{pac}, r_1\right)$$
$$W = (h - v)SK^i_U$$

   where Enc is the encryption algorithm of a secure symmetric cipher of length l.

2) U_i then sends σ as an authentication request to the SDP controller.

3) After receiving the authentication request from U_i, SDP controller executes the following steps to verify the ciphertext.
   a) It first retrieves the public key of U_i stored in the blockchain and then calculates

$$r_1' = e(spk, W) \cdot e\left(G, PK^i_U\right)^v$$
$$r_2' = H_2\left(e(W, SK_C) \cdot e(PK^i_U, PK_C)^v\right).$$

   b) The plaintext pac' is derived as

$$pac' = Dec_{r_2'}\left(E_{pac}\right)$$

   where Dec is the corresponding decryption algorithm of a secure symmetric cipher of length l.
   c) The SDP controller checks the plaintext to ensure the data integrity as

$$v = H_3\left(pac', r_1'\right).$$

Authorized licensed use limited to: University of Witwatersrand. Downloaded on November 07,2025 at 06:01:55 UTC from IEEE Xplore. Restrictions apply.

d) If the checking is successful, the ciphertext is valid and the pac' is obtained. Otherwise, the ciphertext is rejected and the authentication is failed. For a failed authentication for U_i, the SDP controller will undertake an adjustment of its reputation, decrementing it by one unit.

4) Upon the successful authentication, the SDP controller communicates the IP address Ad_2 and port Pt_2 of gateway back to U_i as

$$m = Enc_{SK_C}\left(Ad_2||Pt_2||\mathrm{ID}_U^i\right).$$

It is noteworthy that the reputation of a successfully authenticated U_i will be incremented by one unit.

5) Afterward, the SDP controller initially verifies the correctness of the Pwd of ID_U^i. Then, it meticulously scrutinizes the presence of N and Ts to ascertain the absence of replay attacks. Similar to the previous step, the reputation of a UAV failed to authenticate will decrease. Subsequently, the updated reputation Rep_i of U_i, along with its identity, will be encapsulated into a transaction, and stored in a new block.
6) The SDP controller simultaneously communicates the legitimate UAV identity and the authorization policy to the SDP gateway.
7) U_i sends a connection request and the authorization to SDP gateway with its identity. Since the UAV has been validated by the SDP controller, a secure session is established between U_i and gateway.

E. Reputation Update

The proposed BAZAM utilizes the reputation value to indicate the trustworthiness of UAVs' identities. The reputation of a UAV is quantified by a specific numerical value, reflecting the long-term performance of its authentication behavior. A higher value of the UAV corresponds to a greater number of legitimate authentication instances, indicating a higher level of trust. Without loss of generality, the initial reputation value of the UAV is set to zero. While the reputation value undergoes updates, it does not reset or zero out. Considering the need to adhere to the zero-trust principle of "never trust, always verify," every UAV is required to undergo authentication before establishing a secure connection with the SDP gateway. The reputation of a UAV evolves with each authentication attempt. Therefore, to simplify the entire authentication process, a comparison between the reputation stored in the blockchain and the predefined thresholds is employed.

For a UAV with a reputation below the threshold R_l, indicating multiple authentication failures, the SDP controller will blacklist it. Consequently, the SDP controller will reject its registration request to the KGC, preventing it from obtaining fresh key pair. Similarly, to inspire well-behaved UAVs in terms of authentication and enhance the authentication speed, a threshold R_h is set. This threshold is used to identify high-reputation UAVs, allowing them to bypass the registration process. They can simply initiate access requests to the SDP controller using the previously registered key pairs.

V. Security Analysis

This section demonstrates the security of the BAZAM using a formal security proof, an informal analysis, and a formal security verification.

A. Formal Security Analysis

Theorem 1: Confidentiality: Assume that the MDBDH assumption holds, then BAZAM is secure against adaptive chosen-ciphertext attacks.

Proof: Supposing that an adversary $\mathcal{A}$ is able to break the confidentiality of BAZAM with a nonnegligible probability ϵ, then a challenger $\mathcal{C}$ has the ability to solve the MDBDH problem with advantage ϵ' by interacting with $\mathcal{A}$. ■

Initial: $\mathcal{C}$ generates $c \in \mathbb{Z}_p^*$ and submits the system parameters to $\mathcal{A}$ with $spk = cG$.

Phase 1: $\mathcal{A}$ can issue some queries and $\mathcal{C}$ answers these queries as follows.

1) *H_1* Query: $\mathcal{C}$ keeps a list L_1 of tuples (ID_i, PK_i, s). Whenever $\mathcal{A}$ makes a query on $H_1(\mathrm{ID}_i)$, if the request has been previously queried, the corresponding value in L_1 will be returned. Otherwise, $\mathcal{C}$ chooses $s \in \mathbb{Z}_p^*$ and computes $PK_i = sG$. Afterward, (ID_i, PK_i, s) is stored in L_1 by $\mathcal{C}$ and PK_i is sent to $\mathcal{A}$. $\mathcal{C}$ chooses a extra query with random ID_n and responses $H_1(\mathrm{ID}_n) = bG$.
2) *H_2* Query: $\mathcal{C}$ keeps a list L_2 of tuples (τ, r_2). Whenever $\mathcal{A}$ makes a query on $H_2(\tau)$, if it has been previously queried, the corresponding value in L_2 will be returned. Otherwise, $\mathcal{C}$ chooses $r_2 \in \{0,1\}^l$ and returns it.
3) *H_3* Query: $\mathcal{C}$ keeps a list L_3 of tuples (E_{pac}, r_1, v). Whenever $\mathcal{A}$ makes a query on $H_3(E_{pac}, r_1)$, if the request has been previously queried, the corresponding value in L_3 will be returned. Otherwise, $\mathcal{C}$ chooses $v \in \mathbb{Z}_p^*$ and returns it.
4) *Key Extract Query:* Suppose that $\mathcal{A}$ has made the H_1 query with the UAV ID_U and SDP controller ID_C, where ID_U is the aggregate identity with challenge-response of UAV. Note that if these two identities equal to ID_n, $\mathcal{C}$ cannot calculates the private keys and declares it a failure. Then, $\mathcal{C}$ calculates $SK_U = sc^{-1}G$ and $SK_C = scG$. Finally, $\mathcal{C}$ returns SK_U and SK_C to $\mathcal{A}$.
5) *Signcrypt Query* Once upon receiving the query with ID_U^j, ID_C, and a packet pac, $\mathcal{C}$ calculates SK_U if $\mathrm{ID}_U^j \neq \mathrm{ID}_n$. Then, pac is encrypted and signed to $\mathcal{A}$. Otherwise, $\mathcal{C}$ can calculate SK_C. Then, $\mathcal{C}$ randomly selects $v \in \mathbb{Z}_p^*$ and $W \in \mathbb{G}_1$ to derive r_1, r_2, and E. If a tuple (E_{pac}, r_1, v') and $v' \neq v$ exists in L_3, then $\mathcal{C}$ selects v and W again until no such a tuple in L_3.
6) *Unsigncrypt Query:* Once upon receiving the query with ID_U^j, ID_C, and a ciphertext D_{pac}, $\mathcal{C}$ returns response to $\mathcal{A}$. For the case $\mathrm{ID}_C = \mathrm{ID}_n$, $\mathcal{C}$ returns invalid ciphertext to $\mathcal{A}$. However, if $\mathcal{A}$ has got an answer v from $\mathcal{C}$ by making H_3 query, then $\mathcal{C}$ fails. Otherwise, $\mathcal{C}$ can calculate SK_C and r_1 to check the existence of the tuple (E_{pac}, r_1, v) in L_3. If so, $\mathcal{C}$ recovers v and calculates τ. If the L_2 does not exist the tuple (τ, $\cdot$), $\mathcal{C}$ randomly selects τ' and r_2 and stores them in a tuple. Afterward, $\mathcal{C}$ derives D_{pac} and sends it to $\mathcal{A}$.

Authorized licensed use limited to: University of Witwatersrand. Downloaded on November 07,2025 at 06:01:55 UTC from IEEE Xplore. Restrictions apply.

Challenge: $\mathcal{A}$ outputs two plaintexts pac_1 and pac_2, the identities ID_U and ID_C. If ID_C is queried in the key extract query, $\mathcal{C}$ fails. Otherwise, $\mathcal{C}$ randomly selects $\beta \in \{0, 1\}$, $v \in \mathbb{Z}_p^*$. Finally, $\mathcal{C}$ sets $W^* = aG$ and returns pac_β's challenge ciphertext $\sigma^* = (E^*_{pac_\beta}, v^*, W^*)$ to $\mathcal{A}$.

Phase 2: After receiving the σ^*, $\mathcal{A}$ is allowed to make queries in the phase 1. However, it cannot make the key extract query with ID_C and the make the unsigncrypt query with ID_U and ID_C to σ^*.

Guess: $\mathcal{A}$ outputs β' as its answer. If $\beta' = \beta$, $\mathcal{C}$ outputs $R = e(G, G)^{abc}$ as the solution to the MDBDH problem.

Suppose that $\mathcal{A}$ can make maximum q_{H_i} times H_i queries ($i = 1, 2, 3$), q_{ke} times key extract queries, q_{H_3} times signcrypt queries, q_{us} times unsigncrypt queries. Since σ^* is generated randomly by $\mathcal{C}$, $\mathcal{A}$ cannot provide any advantage. However, $\mathcal{A}$ has a nonnegligible advantage ϵ to recognize the challenge is incorrect. For $\mathcal{C}$, the probability that it rejects a valid ciphertext in unsigncrypt query is $1-(2^k - 1/2^k)^{q_{us}} \leq (q_{us}/2^k)$. Then, $\mathcal{C}$ would continue the simulation in the following situations.

1) $S1$: $\mathcal{A}$ does not make an key extract query with the ID_n in the challenge phase. Then, $\Pr[S1] = (q_{H_1} - q_k/q_{H_1})$.

2) $S2$: The identities chosen by $\mathcal{A}$ satisfy $\mathrm{ID}_U \neq \mathrm{ID}_n$ and $\mathrm{ID}_C = \mathrm{ID}_n$. Then, $\Pr[S2] = (1/q_{H_1} - q_k)$.

Therefore, the probability that $\mathcal{C}$ can get the answer of MDBDH problem given by $\mathcal{A}$ is

$$\Pr[S1 \wedge S2] = \left(\frac{q_{H_1} - q_k}{q_{H_1}}\right)\left(\frac{1}{q_{H_1} - q_k}\right) = \frac{1}{q_{H_1}}$$

and $\mathcal{A}$ has a nonnegligible advantage ϵ, the probability $\mathcal{C}$ can solve the MDBDH problem is

$$Pr\left[\mathcal{C}\left(G, aG, bG, cG, c^{-1}G\right)|a, b, c \in \mathbb{Z}_p^*\right]$$
$$\geq \left[\frac{\epsilon + 1}{2}\left(1 - \frac{q_{us}}{2^k}\right) - \frac{1}{2}\right]\frac{1}{q_{H_1}} = \frac{\epsilon\left(2^k - q_{us}\right) - q_{us}}{q_{H_1} 2^{k+1}}.$$

Hence, $\mathcal{C}$ can solve the MDBDH problem with a nonnegligible probability. However, this is contrary to the Definition 2. Hence, the proposed BAZAM realizes confidentiality.

Theorem 2 (Unforgeability): Assume that the MBDH assumption holds, then BAZAM is existential unforgeability against adaptive chosen-message attacks.

Proof: Suppose that an adversary $\mathcal{A}$ is able to break the unforgeability of BAZAM with a nonnegligible probability ϵ, then a challenger $\mathcal{C}$ has the ability to solve the MBDH problem in an expected time by interacting with $\mathcal{A}$. ■

Initial: $\mathcal{C}$ submits the system parameters to $\mathcal{A}$ with $spk = sskG$, where ssk is the master key that is unknown to $\mathcal{C}$.

Attack: $\mathcal{A}$ is able to adaptively query for various oracles. $\mathcal{C}$ answers the queries as in Theorem 1.

Forgery: $\mathcal{A}$ outputs the forged the packet pac and the identity ID_U. If a valid tuple (pac, ID_U) is output, $\mathcal{C}$ employs the forking lemma: If $\mathcal{A}$ can output a valid signature tuple within time t with a advantage $\epsilon \geq [10(q_{us})(q_{us} + q_{H_3})q_{H_1}/2^k - 1]$, the expected time that $\mathcal{C}$ can get two valid signature tuples is less than $120686(2^k q_{H_3} q_{H_1} t/\epsilon(2^k - 1))$.

After obtaining two signature tuples $((pac, \mathrm{ID}_U), v, W)$ and $((pac, \mathrm{ID}_U), v', W')$, where $v \neq v'$, another machine $\mathcal{C}'$ has control over the machine obtained from $\mathcal{A}$ replacing interaction with the UAV by simulation and gains the two valid signature tuples. $\mathcal{C}'$ calculates $bc^{-1}G = (v' - v)^{-1}(W - W')$. Then, $\mathcal{C}'$ can derive $e(G, G)^{abc^{-1}} = e(aG, bc^{-1})G$.

Hence, $\mathcal{C}$ can solve the MBDH problem with a nonnegligible probability, which is contrary to the Definition 1. Therefore, the proposed BAZAM realizes unforgeability.

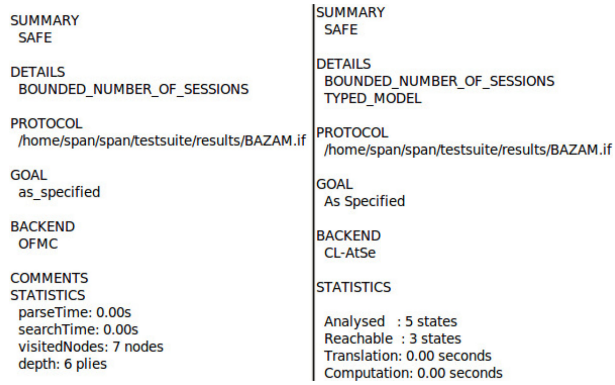

Fig. 4. AVISPA simulation results using OFMC and CL-AtSe.

B. Formal Security Verification

This section verifies the proposed BAZAM scheme using the widely known automated validation of Internet security protocols and applications (AVISPA) tool. The experimental platform utilized a 13th-generation Intel Core i5-13600KF processor with a base clock speed of 3.5 GHz, configured with 64-bit system architecture. This testing environment employed oracle VM VirtualBox virtualization software (v7.1.8) alongside the SPAN-Ubuntu-10.10-light 1 framework for cryptographic protocol implementation. Security assessments employed two formal verification tools: the on-the-fly model checker (OFMC) and the constraint-logic attack searcher (CL-AtSe) protocol analyzers for vulnerability verification.

The simulation results for the malicious intruder using CL-AtSe and OFMC back-ends are as shown in Fig. 4. Consequently, this proves that BAZAM meets the design criteria and is resilient against attacks because the simulation results are output as SAFE.

C. Informal Security Analysis

Resistance of Replay Attack: In this attack, an adversary performs a replay attack by replaying formerly received legal messages. Nevertheless, since the SPA packet generated when the UAV initiates an authentication request includes nonce number N and timestamp T_s, the SDP controller will assess whether the packet has been replayed based on T_s of the current message. Therefore, the BAZAM scheme can resist reply attacks.

Resistance of Node Tampering Attack: In this attack, even if an adversary gains physical access to a UAV and attempts to tamper with its PUF, the PUF will immediately become inoperable. Furthermore, despite such physical compromise, the attacker cannot access the sensitive $\{C_i, R_i\}$ pair because

Authorized licensed use limited to: University of Witwatersrand. Downloaded on November 07,2025 at 06:01:55 UTC from IEEE Xplore. Restrictions apply.

these critical data are not stored in the UAV's memory. Moreover, given the inherent unclonability of the PUF itself, the adversary is fundamentally unable to clone the UAV.

Resistance of Man-in-the-Middle Attack: To conduct the man-in-the-middle attack, an adversary intercepts the message and manipulates it, aiming to convince the SDP controller it was sent by the legitimate UAV. The message exchanged between the UAV and SDP controller is (E_{pac}, v, W). Since the adversary lacks access to r_1, r_2, h, and security key, they are unable to forge the exact message. Therefore, the proposed scheme effectively resists man-in-the-middle attacks.

Perfect Forward Secrecy: An adversary has the ability to successfully steal the private/public key pairs of the UAVs. It cannot obtain the keys of any previous authentication due to the response R_i and ssk corresponding to generating these keys. Consequently, the BAZAM scheme ensures perfect forward secrecy.

Perfect Backward Secrecy: In the proposed scheme, even if an adversary manages to guess the current private key, it will not compromise the security of the next authentication. This is because once the reputation value falls below the threshold R_h, the UAV is required to register with the KGC each time it submits an access request. Furthermore, even if the private key is obtained, it is difficult for adversary to compute k_2 and decrypt the plaintext.

UAV Physical Capture Attack: In the UAV registration process, suppose a UAV U_j is physically captured by an adversary, and then its identity is extracted in the memory. However, the adversary cannot derive the private/public key pair due to the unknown master key and a random nonce R_j. Moreover, there are distinct and independent for the deployed UAV since PUF challenge and response pair (C_j,R_j) are randomly generated. Hence, the revealed data does not help in obtaining the key pair. Therefore, the BAZAM can prevent UAV physical capture attacks due to its reliance on the intrinsic physical variations within the integrated circuit chip, which determines the outputs of the PUF challenge and response pairs.

Correctness: The correctness can be easily verified through the following equations:

$$\begin{aligned} r_1' &= e(spk, W) \cdot e\big(G, PK_U^i\big)^v \\ &= e\big(spk, hSK_U^i\big) \cdot e\big(spk, vSK_U^i\big)^{-1} \cdot e\big(G, PK_U^i\big)^v \\ &= e\big(spk, SK_U^i\big)^h = e\big(G, PK_U^i\big)^h = r_1 \\ r_2' &= H_2\big(e(W, SK_C) \cdot e(PK_U^i, PK_C)^v\big) \\ &= H_2\Big(e(hSK_U^i, SK_C) \cdot e(vSK_U^i, SK_C)^{-1} \cdot e(vPK_U^i, PK_C)^v\Big) \\ &= H_2\Big(e(SK_U^i, SK_C)^h\Big) = H_2\Big(e(PK_U^i, PK_C)^h\Big) = r_2. \end{aligned}$$

VI. Performance Analysis

In this section, we carry out experiments to assess the performance of BAZAM, including computation overhead, communication overhead, and storage overhead.

A. Experimental Settings

The configuration of the PC for the experiments is: Intel Core i5-13600 CPU 3.50 GHz and 32-GB physical memory. For our cryptographic primitives, we utilized the well-established pairing-based cryptography (PBC) Library, employing its Type A curve. Our experiments were fully implemented and evaluated using PBC version 0.5.14 on an Ubuntu 16.04 virtual machine running on VMware Workstation 15 Pro. Moreover, Hyperledger Fabric v1.4 are installed on PC, where Fabric runs through a docker container v20.10.7. Hyperledger Caliper is likewise configured on the same system to evaluate the latency.

B. Computational Overhead

We evaluate the computation time for various cryptographic primitives via experiments. Let T_{ad1}, T_{mu1}, T_{mu2}, T_{ex}, T_{bp}, and T_h denote the notations of point addition in $\mathbb{G}_1$, point multiplication in $\mathbb{G}_1$, point multiplication in $\mathbb{G}_2$, exponentiation operations in $\mathbb{G}_2$, bilinear pairing, and hash operations, respectively. In this platform, we executed each cryptographic primitive for 1000 times and considered the average run time for each cryptographic primitives from these 1000 runs. Then, we have $T_{ad1} \approx 0.005$ ms, $T_{mu1} \approx 0.926$ ms, $T_{mu2} \approx 0.003$ ms, $T_{ex} \approx 0.098$ ms, $T_{bp} \approx 0.757$ ms, and $T_h \approx 0.003$ ms.

We have compared the proposed scheme with state-of-the-art schemes [20], [33], [34], [35], [36]. The specific computational cost of each entity and total computational cost in different schemes are shown in Table II. Specifically, In scheme [20], two types of authentication schemes are presented, that is, UAV-2-ground control station (GCS) and UAV-2-UAV. Considering the authentication process, it highlights the computational cost of BAZAM and other schemes. In BAZAM, UAV and SDP controller need to perform $2T_{bp} + 2T_h + T_{mu1} + 2T_{ex}$ and $4T_{bp} + 2T_h + 2T_{ex}$ operations, respectively, which yields a total cost of 5.872 ms. In BAZAM, the UAV sends its resource access request packet to the SDP controller for cryptographic processing and verification. Therefore, the most time-consuming phase is the encryption and signing processing of packets by UAV. In schemes [33], [34], the UAV-2-GCS [20][1] and the UAV-2-UAV [20][2], the UAV is required to authenticate with another UAV by sharing the secret keys. Then, more T_{mu1} operations are needed than our proposed scheme, each taking 12.992, 9.298, 14.866, and 20.427 ms, respectively. In scheme [35], two pairs of elliptic curve encryption/decryption algorithms are employed in the authentication process, with totally 7.083 ms. It is worth noting that since zero-trust networks are required to authenticate UAVs, the comparison overhead only considers a single authentication of the UAV. In scheme [36], the authenticity of broadcast messages emitted by UAV is verified by an observer. During the identity authentication process, both the signatures generated by the UAV and the ciphertext recovered by observers require 13 T_{ex} operations, incurring 9.123 ms. From the Fig. 5, it indicates that BAZAM is more computational cost-effective than the other schemes.

The computational complexity of the proposed authentication scheme is primarily determined by cryptographic operations, with the dominant costs typically arising from

Authorized licensed use limited to: University of Witwatersrand. Downloaded on November 07,2025 at 06:01:55 UTC from IEEE Xplore. Restrictions apply.

TABLE II
COMPARISON BETWEEN THE PROPOSED SCHEME AND EXISTING SCHEMES OF COMPUTATIONAL OVERHEAD

Scheme	UAV	Authentication Device
Khan *et al.* [33]	$7T_{mu1} + T_{ad1} + 3T_h \approx 6.496$ ms	$7T_{mu1} + T_{ad1} + 3T_h \approx 6.496$ ms
Huang *et al.* [20][1]	$8T_{mu1} + T_{ad1} + 9T_h \approx 7.44$ ms	$8T_{mu1} + 6T_h \approx 7.426$ ms
Huang *et al.* [20][2]	$11T_{mu1} + T_{ad1} + 7T_h \approx 10.212$ ms	$11T_{mu1} + T_{ad1} + 8T_h \approx 10.215$ ms
Chaudhry *et al.* [34]	$5T_{mu1} + 2T_{ad1} + 3T_h \approx 4.649$ ms	$5T_{mu1} + 2T_{ad1} + 3T_h \approx 4.649$ ms
Feng *et al.* [35]	$3T_{mu1} + T_{ad1} \approx 2.783$ ms	$3T_{mu1} + T_{ad1} + 2T_{bp} + T_h \approx 4.3$ ms
Wisse *et al.* [36]	$2T_{bp} + 2T_h + 13T_{ex} + 3T_{mu1} + 2T_{mu2} \approx 5.578$ ms	$3T_{bp} + 13T_{ex} \approx 3.545$ ms
Ours	$2T_{bp} + 2T_h + T_{mu1} + 2T_{ex} \approx 2.642$ ms	$4T_{bp} + 2T_h + 2T_{ex} \approx 3.23$ ms

TABLE III
COMPARISON BETWEEN THE PROPOSED SCHEME AND EXISTING SCHEMES OF COMMUNICATION OVERHEAD

Scheme	UAV	Authentication Device
Khan *et al.* [33]	$\|\mathbb{G}_1\| + \|\mathbb{Z}_p\| + \|ID\| + \|H\| + \|T\|$	$\|\mathbb{G}_1\| + \|\mathbb{Z}_p\| + \|ID\| + \|H\| + \|T\|$
Huang *et al.* [20][1]	$2\|\mathbb{G}_1\| + 2\|\mathbb{Z}_p\| + \|ID\| + 3\|H\| + 2\|T\|$	$\|\mathbb{G}_1\| + \|\mathbb{Z}_p\| + \|H\| + \|T\|$
Huang *et al.* [20] [2]	$\|\mathbb{G}_1\| + \|ID\| + \|H\| + \|T\| + \|m\|$	$\|\mathbb{G}_1\| + \|ID\| + 3\|H\| + 3\|T\| + \|m\|$
Chaudhry *et al.* [34]	$\|\mathbb{G}_1\| + \|\mathbb{Z}_p\| + \|ID\| + 2\|H\| + 2\|T\|$	$\|\mathbb{G}_1\| + \|\mathbb{Z}_p\| + \|ID\| + \|H\| + \|T\|$
Feng *et al.* [35]	$2\|\mathbb{G}_1\| + 2\|\mathbb{Z}_p\| + 2\|m\| + \|ID\| + \|E\|$	$2\|\mathbb{G}_1\| + \|\mathbb{Z}_p\|$
Wisse *et al.* [36]	$7\|\mathbb{G}_1\| + 3\|\mathbb{G}_2\| + \|m\|$	-
Ours	$\|\mathbb{G}_1\| + \|\mathbb{Z}_p\| + \|H\|$	$4\|ID\| + \|m\| + 3\|\mathbb{Z}_p\|$

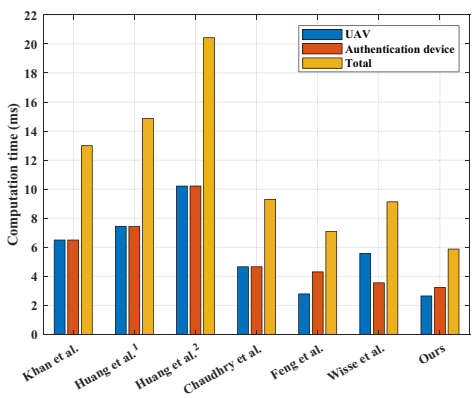

Fig. 5. Computational cost of different entities in various schemes (in ms).

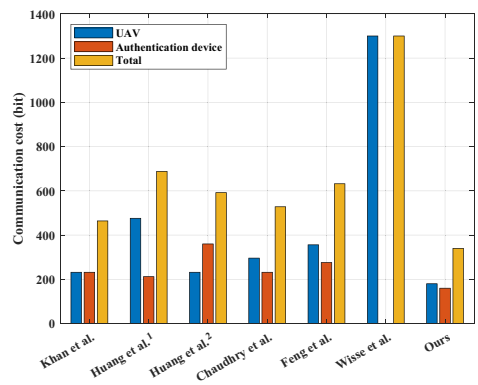

Fig. 6. Communication cost of different entities in various schemes (in bits).

bilinear pairing operations and scalar multiplication. On the UAV side, the overall complexity is $O(2T_{\text{pair}} + \log p)$, where $O(T_{\text{pair}})$ denotes the time required for a single bilinear pairing and $O(\log p)$ corresponds to the cost of scalar multiplication in the elliptic curve group, where p is the order of the elliptic curve. On the SDP controller side, the complexity is $O(4T_{\text{pair}} + \log N)$, where $O(\log N)$ accounts for the overhead of retrieving the UAV's public key or reputation from the blockchain. Therefore, the overall computational complexity of the scheme is $O(6T_{\text{pair}} + \log p + \log N)$.

C. Communication Overhead

As the length of the message varies depending on the task, only the authentication-related communication overloads are computed. As shown in Table III, to evaluate communication overheads, we consider that the sizes of the element in $\mathbb{G}_1$, element in $\mathbb{G}_2$, element in $\mathbb{Z}_p$, identity, digest of hash function, plaintext/ciphertext, and timestamp are $|\mathbb{G}_1| = 128$ bits, $|\mathbb{G}_2| = 128$ bits, $|\mathbb{Z}_p| = 20$ bits, $|ID| = 20$ bits, $|H| = 32$ bits, $|m|/|E| = 20$ bits, and $|T| = 20$ bits. "–" indicates that the entity has no associated overhead. In scheme UAV-2-GCS[20][1] and UAV-2-UAV [20][2], the interaction messages achieving the authentication and key agreement for UAVs contain identity and timestamp, etc, resulting in 688 and 592 bits. In scheme [34], the UAV sends information, including identity and timestamp, etc., for mutual authentication of 528 bits. Similarly, UAV sends a tuple consisting its identity and timestamp to the authentication device in scheme [33], with a cost of 464 bits. In scheme [35], the UAV sends ciphertext to the authentication device. If parsing of the elements of the ciphertext verifies a successful formula, then the UAV will be authenticated and the connection will be established, resulting in 632 bits. The communication overhead in scheme [36] is totally 1300 bits produced from the UAV. The UAV is required to send the signature and message to the observer. On the receiving side, the observer extracts the message and parses

Authorized licensed use limited to: University of Witwatersrand. Downloaded on November 07,2025 at 06:01:55 UTC from IEEE Xplore. Restrictions apply.

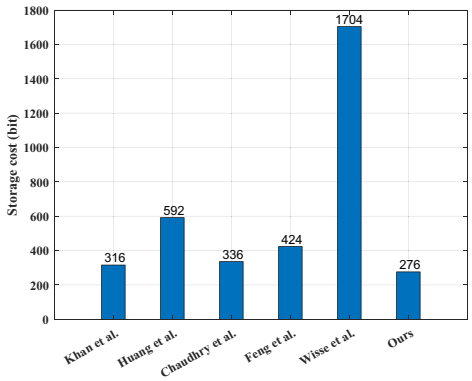

Fig. 7. Storage cost of various schemes (in bits).

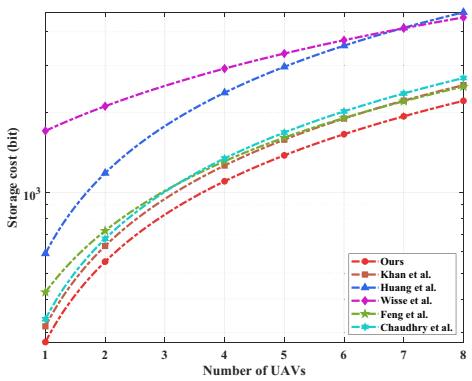

Fig. 8. Storage cost with varying UAV number (in bits).

the signature to directly authenticate the UAV and verify its contents. From the Fig. 6, the communication cost of BAZAM is similar or less to those of some related schemes. In addition, in the zero-trust network, the SDP gateway defaults to rejecting all access requests. Only after the UAV is authenticated, the SDP gateway establishes a connection with the UAV, thus minimizing the attack surface and greatly reducing the network security risk. Therefore, BAZAM provides better security and functionalities than related schemes.

D. Storage Overhead

We assess the additional storage overhead encompassing certain system security parameters and essential cryptographic keys. The storage overhead comparison is shown in Fig. 7. The UAV in BAZAM needs to store identity ID_U^i, public/ private key pair (PK_U^i, SK_U^i) in the memory. It is worthy noting that the challenge is input to generate its public/private key pair without additional storage. In addition, the reputation value is stored on the blockchain, reducing the UAV's storage overhead. In scheme [20], the UAV requires to store its previous and present identity, public and private key pair and the response of PUFs module (20 bits). The UAV in scheme [36] has a storage cost of public group key (1300 bits) and private key (404 bits) since the parameters are complex. Similarly, the UAV in scheme [35] requires to store its identity, public/private key pair, and public key of authentication device. Besides, the challenge C_i (20 bits) also is needed to be stored in the UAV's memory. In scheme [34], the UAV stores its own identity, public and private keys, and its certificate (20 bits), as well as the identity and public key of the ground station server in the memory. In scheme[33], the public/private key pair, identity, certificate (128 bits, which differs from that of scheme [34]), and signature (20 bits) are stored in the UAV's memory for authentication. The relationship between the storage overhead and the UAV number is shown in Fig. 8. We observe that with the increasing number of authenticated UAVs, due to the necessity of regenerating parameters for each UAV, the storage costs of schemes [33], [34], [35] gradually exceed those of BAZAM. Hence, in terms of the computation, storage, and communication overheads, BAZAM demonstrates strong competitiveness than the existing authentication schemes for UAVs.

E. Blockchain Latency

The blockchain delay refers to the time starting from when a UAV or authentication device submits a reading/writing request to the blockchain until that request is successfully validated or confirmed as recorded on the blockchain. We set up the consortium blockchain nodes using Docker 20.10.7 containers on the Hyperledger Fabric 1.4.1 platform, running on Ubuntu 16.04 virtual machines. Clients, which simulate UAVs and DES, operate on a Windows 11 host. Communication between the clients and the blockchain is managed via the Fabric Java SDK, with the smart contract developed in Go. To test the reading and writing latency of authentication schemes at different blockchain scales, we performed writing and querying operations based on the deployed chaincode, which consists of two cases: *C1:* one peer node and one orderer node and *C2:* four peer nodes and two orderer nodes. In scheme [20], the UAV submits one reading and one writing operations, respectively. The authentication device executes two reading and two writing operations. The latency caused in [20] is totally 24.15 (C1) and 43.8 ms (C2). In scheme [35], the UAV and authentication device perform one reading and one writing operations, respectively, resulting in a latency of about 16.1 (C1) and 29.2 ms (C2). In our proposed scheme, only one reading and one writing operation for UAV reputation query and update is required by the SDP controller, with a latency of about 8.02 (C1) and 14.6 ms (C2).

F. Energy Consumption

Since UAVs are often constrained by limited energy resources, we evaluated their energy consumption by considering two primary contributors: data transmission and cryptographic computation. To model the mobility patterns of UAVs, we used MATLAB to simulate the UAV network environment. The UAVs were configured to operate at altitudes ranging from 200 to 500 m, with a maximum flight speed of $V_{\max} = 40$ m/s. We analyzed the energy consumption under varying numbers of UAVs. The system parameters were set as follows: noise power δ^2 was assumed to be -130 dBm/Hz, transmission power P_i was set to 1 W, the available bandwidth B was 1 MHz, and the air density ρ was 1.42×10^3. As shown in Fig. 9, as the number of UAVs increases, the proposed BAZAM demonstrates lower energy consumption,

Authorized licensed use limited to: University of Witwatersrand. Downloaded on November 07,2025 at 06:01:55 UTC from IEEE Xplore. Restrictions apply.

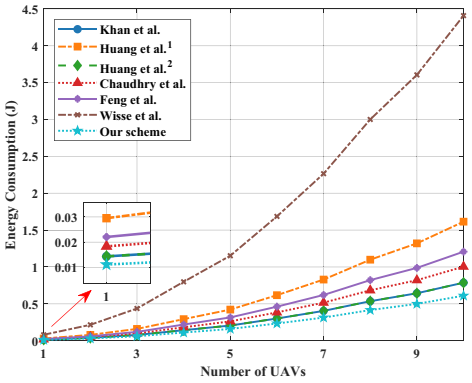

Fig. 9. Energy consumption with varying UAV number (in J).

primarily due to its reduced computational and communication overhead.

G. Discussion

BAZAM provides a high level of security by keeping keys confidential and defending against various common attacks. Compared to other schemes, BAZAM incurs relatively lower computational, communication, and storage costs. While the use of blockchain introduces additional overhead, it significantly enhances security within a zero-trust architecture. This gives BAZAM an advantage over other blockchain-based solutions. Simulations of its practical impact on UAVs show that BAZAM has lower energy consumption overhead, with its advantages becoming more pronounced as the number of UAVs increases. Overall, BAZAM demonstrates superior performance.

VII. Conclusion

In this article, we've investigated identity authentication in multi-UAV wireless networks, presenting BAZAM, a blockchain-assisted zero-trust authentication scheme. The introduction of zero-trust model emphasizes authentication and authorization at every step in the network. The blockchain is integrated to build mutual trust between different entities of the zero-trust network and to protect the UAV information from tampering. The scheme also authenticates UAVs by encrypting and signing their access requests for processing. BAZAM notably outperforms existing UAV authentication schemes in terms of computational, communication, and storage overheads.

While the proposed scheme shows promising results, scalability remains a significant challenge when applying blockchain to UAV swarm operations, especially in scenarios involving high UAV density and frequent interactions. As the number of UAVs increases, the overhead of consensus, data synchronization, and on-chain storage can grow rapidly, potentially affecting system responsiveness and efficiency. Future work will explore lightweight consensus mechanisms, blockchain sharding, and hybrid on/off-chain storage techniques as potential strategies to mitigate these issues.

References

[1] J. Li et al., "UAV-assisted microservice mobile edge computing architecture: Addressing post-disaster emergency medical rescue," *IEEE Trans. Comput.*, vol. 74, no. 8, pp. 2635–2648, Aug. 2025, doi: 10.1109/TC.2025.3566913.

[2] X. Lu et al., "Reinforcement learning-based physical cross-layer security and privacy in 6G," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 425–466, 1st Quart., 2022.

[3] M. Xie, Z. Chang, H. Li, and G. Min, "BASUV: A blockchain-enabled UAV authentication scheme for Internet of Vehicles," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 9055–9069, 2024.

[4] X. Ren et al., "A novel access and handover authentication scheme in UAV-aided satellite-terrestrial integration networks enabling 5G," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 3, pp. 3880–3899, Sep. 2023.

[5] H. Wang, X. Wang, H. Fang, and L. Hanzo, "GALAMC: Guaranteed authentication level at minimized complexity relying on intelligent collaboration," *IEEE Trans. Commun.*, vol. 71, no. 5, pp. 2916–2930, May 2023.

[6] Z. Liu et al., "Lightweight trustworthy message exchange in unmanned aerial vehicle networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2144–2157, Feb. 2023.

[7] D. Wang, Y. Cao, K.-Y. Lam, C.-H. Chi, and K.-K. R. Choo, "An accountable GAKA protocol with changeable thresholds and verifiable shares in UAVs-assisted IoVs for emergency rescue," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 4, pp. 5522–5537, Apr. 2025.

[8] O. Ceviz, S. Sen, and P. Sadioglu, "A survey of security in UAVs and FANETs: Issues, threats, analysis of attacks, and solutions," *IEEE Commun. Surveys Tuts.*, early access, Dec. 11, 2024, doi: 10.1109/COMST.2024.3515051.

[9] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," NIST, Gaithersburg, MD, USA, document NIST SP 800-207, 2020.

[10] S. Hong, L. Xu, J. Huang, H. Li, H. Hu, and G. Gu, "SysFlow: Toward a programmable zero trust framework for system security," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 2794–2809, 2023.

[11] Y. Liu et al., "Secure and scalable cross-domain data sharing in zero-trust cloud-edge-end environment based on sharding blockchain," *IEEE Trans. Depend. Secure Comput.*, vol. 21, no. 4, pp. 2603–2618, Jul./Aug. 2024.

[12] M. Tsai, S. Lee, and S. W. Shieh, "Strategy for implementing of zero trust architecture," *IEEE Trans. Rel.*, vol. 73, no. 1, pp. 93–100, Mar. 2024.

[13] M. Xie et al., "Traceability and identity protection in smart agricultural IoT system framework based on blockchains," *IEEE Trans. Depend. Secure Comput.*, early access, Apr. 29, 2025, doi: 10.1109/TDSC.2025.3565593.

[14] G. Bansal and B. Sikdar, "Secure and efficient Peer2Peer authentication-attestation protocol for UAV networks," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 9547–9557, Dec. 2024.

[15] J. Wang, Z. Jiao, J. Chen, X. Hou, T. Yang, and D. Lan, "Blockchain-aided secure access control for UAV computing networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 6, pp. 5267–5279, Nov./Dec. 2024.

[16] S. Zhu, X. Zheng, L. Liu, and H. Ma, "FPCA: Parasitic coding authentication for UAVs by FM signals," *IEEE/ACM Trans. Netw.*, vol. 32, no. 6, pp. 4570–4584, Dec. 2024.

[17] Y. Tan, J. Liu, and N. Kato, "Blockchain-based lightweight authentication for resilient UAV communications: Architecture, scheme, and future directions," *IEEE Wireless Commun.*, vol. 29, no. 3, pp. 24–31, Jun. 2022.

[18] M. Adil, H. Abulkasim, A. Farouk, and H. Song, "*R3ACWU*: A lightweight, trustworthy authentication scheme for UAV-assisted IoT applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 6, pp. 6161–6172, Jun. 2024.

[19] W. Wang, Z. Han, T. R. Gadekallu, S. Raza, J. Tanveer, and C. Su, "Lightweight blockchain-enhanced mutual authentication protocol for UAVs," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 9547–9557, Mar. 2024.

[20] K. Huang, H. Hu, and C. Lin, "BAKAS-UAV: A secure blockchain-assisted authentication and key agreement scheme for unmanned aerial vehicles networks," *IEEE Internet Things J.*, vol. 11, no. 22, pp. 36858–36883, Nov. 2024.

[21] S. Yu, A. K. Das, and Y. Park, "RLBA-UAV: A robust and lightweight blockchain-based authentication and key agreement scheme for PUF-enabled UAVs," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 12, pp. 21697–21708, Dec. 2024.

Authorized licensed use limited to: University of Witwatersrand. Downloaded on November 07,2025 at 06:01:55 UTC from IEEE Xplore. Restrictions apply.

[22] R. Karmakar, G. Kaddoum, and O. Akhrif, "A PUF and fuzzy extractor-based UAV-ground station and UAV-UAV authentication mechanism with intelligent adaptation of secure sessions," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 3858–3875, May 2023.

[23] J.-S. Lee, T.-H. Chen, C.-J. Chew, P.-Y. Wang, and Y.-Y. Fan, "Unconsciously continuous authentication protocol in zero-trust architecture based on behavioral biometrics," *IEEE Trans. Rel.*, vol. 74, no. 2, pp. 2591–2604, Jun. 2025.

[24] S. O. Ajakwe, D.-S. Kim, and J.-M. Lee, "Drone transportation system: Systematic review of security dynamics for smart mobility," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14462–14482, Aug. 2023.

[25] Y. Ge and Q. Zhu, "Gazeta: Game-theoretic zero-trust authentication for defense against lateral movement in 5G IoT networks," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 540–554, 2024.

[26] H. Fang, Y. Zhu, Y. Zhang, and X. Wang, "Decentralized edge collaboration for seamless handover authentication in zero-trust IoV," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 8760–8772, Aug. 2024.

[27] Z. Guo, J. Cao, X. Wang, Y. Zhang, B. Niu, and H. Li, "UAVA: Unmanned aerial vehicle assisted vehicular authentication scheme in edge computing networks," *IEEE Internet Things J.*, vol. 11, no. 12, pp. 22091–22106, Jun. 2024.

[28] H. Xie, Y. Wang, Y. Ding, C. Yang, H. Liang, and B. Qin, "Industrial wireless Internet zero trust model: Zero trust meets dynamic federated learning with blockchain," *IEEE Wireless Commun.*, vol. 31, no. 2, pp. 22–29, Apr. 2024.

[29] S. S. Chow, S.-M. Yiu, L. C. Hui, and K.-P. Chow, "Efficient forward and provably secure ID-based signcryption scheme with public verifiability and public ciphertext authenticity," in *Proc. 6th Int. Conf. Inf. Secur. Cryptol. (ICISC)*, Seoul, South Korea, 2004, pp. 352–369.

[30] A. Moubayed, A. Refaey, and A. Shami, "Software-defined perimeter (SDP): State of the art secure solution for modern networks," *IEEE Netw.*, vol. 33, no. 5, pp. 226–233, Sep./Oct. 2019.

[31] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[32] F. Li, H. Zhang, and T. Takagi, "Efficient signcryption for heterogeneous systems," *IEEE Syst. J.*, vol. 7, no. 3, pp. 420–429, Sep. 2013.

[33] M. A. Khan et al., "A provable and privacy-preserving authentication scheme for UAV-enabled intelligent transportation systems," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3416–3425, May 2022.

[34] S. A. Chaudhry, K. Yahya, M. Karuppiah, R. Kharel, A. K. Bashir, and Y. B. Zikria, "GCACS-IoD: A certificate based generic access control scheme for Internet of Drones," *Comput. Netw.*, vol. 191, May 2021, Art. no. 107999.

[35] C. Feng, B. Liu, Z. Guo, K. Yu, Z. Qin, and K.-K. R. Choo, "Blockchain-based cross-domain authentication for intelligent 5G-enabled Internet of Drones," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 6224–6238, Apr. 2022.

[36] E. Wisse, P. Tedeschi, S. Sciancalepore, and R. Di Pietro, "A^2RID—Anonymous direct authentication and remote identification of commercial drones," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10587–10604, Jun. 2023.

Mingyue Xie received the M.S. degree from the School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China. She is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China.

Her research interests include unmanned aerial vehicle networks, blockchain, and Internet of Things.

Zheng Chang (Senior Member, IEEE) received the B.Eng. degree from Jilin University, Changchun, China, in 2007, the M.Sc. (Tech.) degree from the Helsinki University of Technology (currently Aalto University), Espoo, Finland, in 2009, and the Ph.D degree from the University of Jyväskylä, Jyväskylä, Finland, in 2013.

Since 2008, he has been held various research positions with the Helsinki University of Technology, University of Jyvaskyla, and Magister Solutions Ltd., in Finland. He was a Visiting Researcher with Tsinghua University, Beijing, China, from June to August in 2013, and with the University of Houston, Houston, TX, USA, from April to May in 2015. He has published over 200 papers in journals and conferences. His research interests include federated learning, cloud/edge computing, UAV/vehicular networks, and green communications.

Dr. Chang has been awarded by the Ulla Tuominen Foundation, the Nokia Foundation, and the Riitta and Jorma J. Takanen Foundation for his research excellence. He has been awarded as the 2018 IEEE Communications Society best young researcher for Europe, Middle East and Africa Region and the 2021 IEEE Communications Society MMTC Outstanding Young Researcher. He received the Best Paper Awards from IEEE ICC in 2023, IEEE TCGCC, and APCC in 2017. He serves as an Editor for IEEE WIRELESS COMMUNICATIONS LETTERS, IEEE TRANSACTIONS ON MACHINE LEARNING IN COMMUNICATIONS AND NETWORKING, and *China Communications*, and a Guest Editor for IEEE NETWORK, IEEE WIRELESS COMMUNICATIONS, *IEEE Communications Magazine*, IEEE INTERNET OF THINGS JOURNAL, and IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS. He was the Best Editor of IEEE WIRELESS COMMUNICATION LETTERS AND CHINA COMMUNICATIONS in 2024, the Exemplary Reviewer of IEEE WIRELESS COMMUNICATION LETTERS in 2018. He has participated in organizing workshop and special session in Globecom' 19, WCNC'18–'24, SPAWC'19, and ISWCS'18. He also serves as Symposium/Track Co-Chair of IEEE ICC'20, Globecom'23, VTS'25S, and ICC'26, Publicity Co-Chair of IEEE Infocom'22, Workshop Co-Chair of ICCC'22 and VTS'25F, TPC Co-Chair of IEEE iThing'22, and TPC member for many IEEE major conferences, such as INFOCOM, ICC, and Globecom.

Alain R. Ndjiongue (Senior Member, IEEE) received the M.Eng. and D.Eng. degrees in electrical and electronic engineering from the University of Johannesburg, Johannesburg, South Africa, in 2013 and 2017, respectively.

He was a Senior Lecturer with the University of Johannesburg, from 2017 to 2020. He was also a Senior Researcher with the Memorial University of Newfoundland, St. John's, NL, Canada, where he was an Assistant Professor from 2022 to 2024. Since September 2024, he has been an Associate Professor with the University of Witwatersrand, Johannesburg. His research interests include digital communications, including wireless, free space optical, visible light communications, and underwater optical communications. He also researches optical reconfigurable intelligent surfaces.

Dr. Ndjiongue is an Associate Editor of IEEE COMMUNICATIONS LETTERS and IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY. He is also an active reviewer for several high-impact IEEE ComSoc journals and conferences. He was a 2017, 2018, and 2022 Exemplary Reviewer, and 2024 Exemplary Editor of IEEE Communication Letters, Optical Society of America, and is a 2019 top 1% Peer Reviewer in Essential Science Indicators Research.

Authorized licensed use limited to: University of Witwatersrand. Downloaded on November 07,2025 at 06:01:55 UTC from IEEE Xplore. Restrictions apply.

Tao Chen (Member, IEEE) directs the Intelligent Dependability Engineering for Adaptive Software Laboratory, Birmingham, U.K., conducting cutting-edge research on the intersection between AI and Software Engineering. His work has been regularly published in all major software engineering conferences/journals such as *Intelligent Control and System Engineering*, *Foundations of Software Engineering*, *Automated Software Engineering*, *ACM Transactions on Software Engineering and Methodology*, and IEEE TRANSACTIONS ON SOFTWARE ENGINEERING. He has been supported by projects worth over £1 million from external funding bodies. His research interests include performance engineering, self-adaptive systems, search-based software engineering, data-driven software engineering, and their interplay with machine learning and computational intelligence.

Dr. Chen currently serves as an Associate Editor for *ACM Transactions on Autonomous and Adaptive Systems* and a program committee member for many prestigious conferences.

Hongwei Li (Fellow, IEEE) is currently the Vice Dean and a Professor with the School of Computer Science and Engineering (School of Cyber Security), University of Electronic Science and Technology of China, Chengdu, China. His research interests include network security and applied cryptography.

Prof. Li serves as an Associate Editor for IEEE INTERNET OF THINGS JOURNAL and the Lead Guest Editor for IEEE Network, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE INTERNET OF THINGS JOURNAL. He serves/served as the Technical Symposium Co-Chair for IEEE ICC 2022, ACM TUR-C 2019, IEEE ICCC 2016, and many Technical Program Committees for international conferences, such as IEEE INFOCOM, IEEE WCNC, IEEE SmartGridComm, BO-DYNETS, and IEEE DASC. He is the Distinguished Lecturer of the IEEE Vehicular Technology Society.

Authorized licensed use limited to: University of Witwatersrand. Downloaded on November 07,2025 at 06:01:55 UTC from IEEE Xplore. Restrictions apply.