electronics

MDPI

Article

Design of Secure and Efficient Authentication Protocol for Edge Computing-Based Augmented Reality Environments

DeokKyu Kwon [1] and Youngho Park [2,*]

[1] School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Republic of Korea; kdk145@knu.ac.kr
[2] School of Electronics Engineering, Kyungpook National University, Daegu 41566, Republic of Korea
* Correspondence: parkyh@knu.ac.kr; Tel.: +82-53-950-7842

Abstract: Augmented reality (AR) is a virtual technology that integrates virtual information and objects into real environments, offering unprecedented possibilities in such fields such as architecture, education, and healthcare. Real-time communication and security protocols are critical to the successful deployment of AR applications to ensure user immersion, prevent motion sickness, and address security problems. This paper proposes a secure user-to-user (U2U) and user-to-infrastructure (U2I) authentication protocol suitable for edge computing-based AR environments. We also employ extended Chebyshev chaotic maps and physical unclonable functions to ensure security and efficiency during the authentication process. The proposed protocol initiates session keys after U2I authentication when an AR user enters the edge node area, facilitating secure U2U authentication for sharing data with nearby users. We conduct comprehensive studies of the security robustness of the proposed protocol using formal and informal analyses, including "Burrows–Abadi–Needham logic", "Real-Or-Random model", the "Scyther tool" and informal security analyses. Furthermore, we measure the performance of cryptographic primitives using the "Multiprecision Integer and Rational Arithmetic Cryptographic Library" Cryptographic SDK. We perform a comparative analysis of security features and functionality, and we conduct a computational and communication cost analysis. The results reveal that the proposed protocol can provide security and efficiency for edge computing-based AR environments, presenting the methods for seamless and secure real-time AR data exchanges for U2I and U2U communications.

Keywords: augmented reality; authentication; Chebyshev chaotic maps; edge computing; security analysis

Citation: Kwon, D.; Park, Y. Design of Secure and Efficient Authentication Protocol for Edge Computing-Based Augmented Reality Environments. *Electronics* **2024**, *13*, 551. https://doi.org/10.3390/electronics13030551

Academic Editors: Yongjun Ren and Hu Xiong

Received: 2 January 2024
Revised: 23 January 2024
Accepted: 29 January 2024
Published: 30 January 2024

1. Introduction

Augmented Reality (AR) is a computer graphics technology that overlays virtual information and objects in real environments. In contrast to traditional computer graphics, AR uses real-time data on the actual surrounding environments, including length, depth, texture, and distance. Furthermore, AR devices are equipped with high-pixel displays, eye and object-tracking systems, and sensitive cameras and sensors to process real-time information [1]. Figure 1 illustrates the simplification of AR data processing. Recently, various network environments have attempted to use AR technology with the development of mobile chipset and wireless communication technologies. For instance, Microsoft launched the HoloLens series, which can perform AR tasks, including architectural design, education, and healthcare [2]. HoloLens increases work efficiency by querying necessary data from industrial sites and using them as visual material.

The implementation of real-time big data processing is necessary to ensure user immersion and prevent motion sickness in AR technology. Furthermore, the output of information with a higher pixel density (bit rate) than typical photographs and videos is essential. However, traditional cloud computing technology is burdened by high latency,

and stand-alone technology demands high performance from end devices. Edge computing technology addresses this challenge by installing edge nodes at the network's bottom, enabling efficient communication and performance in AR environments [3,4].

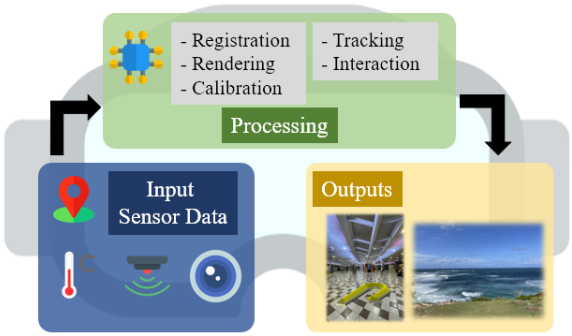

Figure 1. Simplification of general AR data processing.

In edge computing-based AR environments, edge nodes provide similar visual information-based AR services to users within the area because of their locality. Furthermore, AR environments necessitate the real-time processing of substantial data compared to existing mobile computing environments [5]. Thus, edge computing-based AR environments still have a problem, potentially leading to frequent overloading on edge nodes, which is similar to traditional cloud-based mobile computing environments. The method of sharing AR content between users involves each user downloading necessary data from nearby users, bypassing the need for AR services from the edge node [6,7]. Through this approach, AR users can receive real-time data directly from nearby users, enabling the establishment of an efficient ad hoc network. Consequently, the overall operational efficiency can be enhanced because edge nodes do not need to carry the workload for all AR users employing user-to-user (U2U) communications.

The edge-computing-based AR content-sharing scheme can be subject to security problems because the communication channels of U2U and user-to-infrastructure (U2I) communications are public and wireless environments. If messages are hijacked, deleted, or captured by attackers, infringement of the user information can occur. Moreover, stolen or lost AR devices can threaten user privacy because they store sensitive information. If user information stored in an edge node is leaked, adversaries can use it to attempt a spoofing attack. In edge computing AR environments, edge nodes can be targeted to paralysis because they are regarded as local servers. Since user data in the AR environment are completely personalized data, untraceability and anonymity must be guaranteed. Nevertheless, the user's identity must be verified in U2U and U2I communication while ensuring anonymity, and this process must be seamless and lightweight. Therefore, a secure authentication protocol is necessary for U2U and U2I communications. The security requirements and challenges for edge computing-based AR environments are as follows.

- *Authentication*: Mutual authentication is necessary to identify edge computing nodes and AR content users.
- *Privacy preserving*: Because the data that users use for AR services are based on personal information, resistance to privacy leaks is necessary.
- *Anonymity and untraceability*: Because the user's AR device utilizes sensitive data and has mobility, it must provide anonymity and untraceability in U2U and U2I communications.
- *Data access*: Sophisticated data access is required because data are generated based on the user's visual and auditory information in an edge computing-based AR environments.
- *Latency*: Since real-time communication is more important in the AR environment than in traditional environments, it must provide high performance while maintaining security.

In this paper, we propose a secure U2U and U2I authentication protocol for edge computing-based AR environments. The proposed protocol utilizes extended Chebyshev

chaotic maps to ensure security and efficiency during the authentication process. After the user enters the edge node area through U2I authentication, a session key is created, and data are shared through U2U authentication with users who have adjacent AR content. Moreover, we use physical unclonable functions (PUFs) to guarantee the privacy of edge nodes. Therefore, AR users can receive seamless and secure real-time AR data from edge nodes and surrounding users.

1.1. Motivation

The edge computing-based AR environment has the characteristic of exchanging a huge amount of data compared to the traditional mobile networks. Since the exchanged data contain sensitive information of users, it can cause serious problem when these data are encrypted in low-level security. If a lot of computational resources are consumed for data masking, delays can occur due to low data response rates, which deteriorate service quality such as motion sickness. Based on the above motivations, we designed an authentication protocol to provide seamless U2I and U2U communications considering various security threats and efficiency for edge computing-based AR environments.

1.2. Contribution

The main contributions are as follows:

- We propose a secure and efficient authentication protocol for edge computing-based AR environments. The proposed protocol considers secure U2U and U2I communications for secure edge computing. Moreover, the proposed protocol can provide an efficient communication process using extended Chebyshev chaotic maps and PUFs.
- We analyze the security robustness of the proposed protocol using formal and informal analyses, such as "Burrows–Abadi–Needham (BAN) logic [8]", "Real-Or-Random (ROR) model [9]" the "Scyther tool [10,11]", and "informal security analysis".
- We measure the performance of various cryptographic primitives using the "Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL) [12]" Cryptographic SDK. From that, we compare the security features and functionalities, and we conduct a computation and communication costs analysis of the proposed protocol and other related schemes.

2. Related Works

Edge computing-based AR environments have been being researched for a few years. In 2019, Ren et al. [13] introduced an architecture for AR environments based on edge computing technology. In Ren et al.'s AR architecture, cloud, edge, and user layers are constructed forming a hierarchy. Moreover, they present an operation mechanism to implement the edge computing-based AR technology. In 2021, Siriwardhana et al. [14] discussed AR technology that combines fifth-generation (5G) and edge computing. One of the main contributions in their paper is a division and analysis of AR environments, such as cloud, edge, localized, and hybrid-based architectures. They also discussed the requirements of security threats and solutions for AR environments. In the same year, Chen et al. [3] proposed an offloading scheme for AR edge computing environments. They considered a deep reinforcement learning model to obtain an optimized resource allocation. They demonstrated that their scheme could decrease the computation complexity and achieve a real-time offloading because this model does not consider combinatorial optimization. In 2022, Morín et al. [15] introduced a simplified AR offloading architecture using edge computing technology. Depending on the size and resource status of AR data, three AR offloading scenarios were introduced in their architecture: full, object detection and segmentation, and occlusion handling.

Recently, various service caching mechanisms have been proposed to address heavy computing tasks in AR environments. Dang et al. [16] proposed an on-device computational caching scheme for AR environments. In Dang et al.'s scheme, they presented a system model that can provide various AR services through computing caching-based device to

device (D2D) communications. In their system model, D2D communication is activated when a user device cannot access an edge server due to overloading problems. In 2023, Park et al. [17] presented an object modeling system using collaborative communication for AR streaming services. In Park et al.'s modeling system, an AR user requests a part of segments from the edge computing node (ECN) and other nearby AR users to construct a complete three-dimensional (3D) virtual object. From that, the AR user can receive a decreased end-to-end delay and check the quality of the part-segments directly.

To provide secure communications between devices, D2D authentication schemes are introduced considering various network environments. In 2019, Chen et al. [18] proposed an authentication protocol for smart grid environments. Chen et al. considered communication channels between energy-trading consumers and constructed an U2U authentication protocol using bilinear pairings. Alzahrani et al. [19] introduced a two-party authentication scheme for the Internet of Vehicles. In their system model, various autonomous devices, including sensors, cameras, and smart vehicles, are registered to certificate authority. Then, the autonomous devices exchanges the authentication messages directly using the public key infrastructure. In 2021, Pham and Dang [20] proposed a lightweight D2D authentication protocol for the Internet of Things (IoT) environment using elliptic curve cryptography (ECC). In Phan and Dang's protocol, a gateway participates in the authentication process to burden the computational overhead of ECC. In 2022, Hajian et al. [21] proposed a mutual authentication and key agreement protocol for D2D IoT deployments. In Hajian et al.'s protocol, IoT devices authenticate with each other using a direct communication link. Moreover, Hajian et al. used ECC to ensure a higher security than symmetric key-based cryptosystems. However, the above schemes [18–21] used ECC and bilinear pairings, which are unsuitable for U2U and U2I communications in edge computing-based AR environments. Moreover, several schemes used the main server as a certificate authority to complete D2D authentications, which can increase communication overhead. Because large volumes of data interact in real time in AR environments, designing computationally-efficient authentication protocols is essential. Thus, we propose a mutual authentication protocol using extended Chebyshev chaotic maps and PUFs to ensure a lightweight and secure scheme. Table 1 shows the summarized literature review of the related schemes and the proposed protocol.

Table 1. Summary of the related schemes for D2D and U2U authentications.

Year	Scheme	Contributions	Limitations
2019	[18]	• Proposed a D2D communication model in smart grids • U2U authentication and key estabilshment protocol for smart grid environments • Utilized bilinear pairings to establish shared key	• Vulnerable to impersonation, privileged insider, and ESL attacks • Requires huge computation costs using bilinear pairings
2019	[19]	• Proposed D2D network model for IoT environments • A two-party authentication scheme for the autonomous devices • Used self-certifed public keys and ECC	• Vulnerable to privileged insider attacks • Cannot guarantee user anonymity • Requires high computation costs using ECC
2021	[20]	• Proposed a layered-based network architecture for IoT deployments • Lightweight D2D authentication protocol for IoT networks • Utilized ECC to establish a session key	• Vulnerable to replay, man-in-the-middle, and impersonation attacks • The gateway must be involved in the D2D authentication process • Requires high computation costs using ECC
2022	[21]	• Proposed 6G-based network model in IoT environments • Mutual authentication and key agreement protocol for D2D IoT deployments • Utilized ECC to establish a session key	• Vulnerable to verification table leakage attacks • High computation costs using ECC
-	Proposed	○ Proposed an edge computing-based system model for AR environments ○ Lightweight and secure mutual authentication protocol for U2U and U2I communications in edge computing-based AR environments ○ Utilize Chebyshev chaotic maps and PUFs to ensure security requirements and challenges for edge computing-based AR environments	

3. Preliminaries

In this section, we introduce the backgrounds of the proposed protocol, such as the system model, adversary model, security model, extended Chebyshev chaotic maps, and PUFs.

3.1. System Model

The proposed system model consists of the AR cloud, edge computing nodes (ECNs), and AR users. Figure 2 presents the proposed system model and details as follows:

AR cloud: The AR cloud manages the proposed network system. Thus, the AR cloud registers ECN and AR users, and it stores their data in a secure database. The AR cloud has substantial data and computational resources.

ECN: The ECN is a infrastructure that performs the tasks of the AR cloud. Thus, ECNs are deployed in specific areas to provide useful AR services to service users. The ECNs communicate with the AR cloud to receive and store AR user data in the edge cache [13]. The data can be distributed to the corresponding AR users to enable real-time services. In the proposed system model, ECNs have sufficient computation and storage resources.

AR user: These users are end nodes that receive AR services using smart devices, including head-mounted display (HMD) devices. They interact with the corresponding infrastructure and users using wireless communication links. With the downloaded data, smart devices display the rendered information, and AR users can receive AR services. Thus, AR users register to the AR cloud and receive AR information from ECNs. Moreover, AR users can share their AR service data with other users (AR service user) when the ECN suffers from overloading problems.

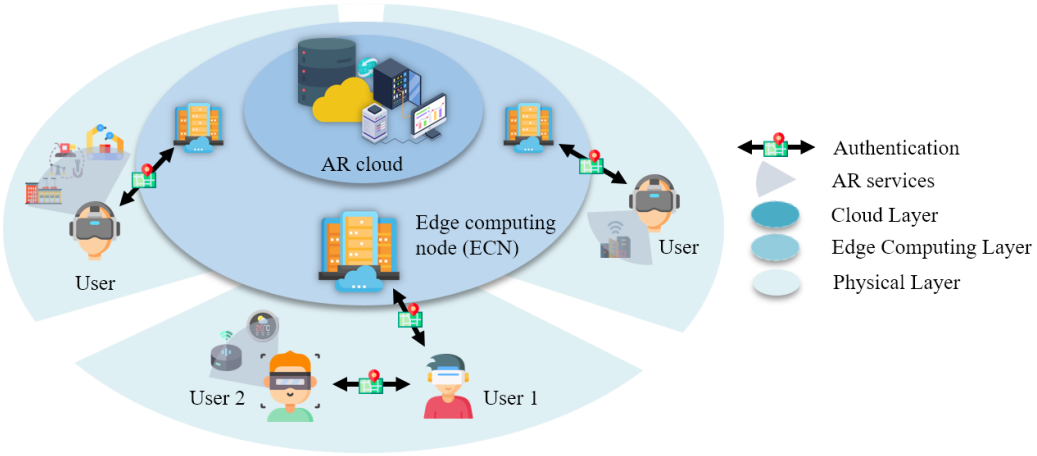

Figure 2. Proposed edge computing-based AR environments.

3.2. Adversary Model

We adopt the well-known adversary model, "Dolev–Yao (DY)" [22] and "Canetti–Krawczyk (CK)" [23] network model. In the DY network model, network participants communicate with each other through public channels. Thus, adversaries can be concerned with the messages because adversaries have authority over public networks. The adversary can handle (e.g., eavesdrop, delete, insert, capture) messages transmitted via open channels. In the CK network model, the adversary can obtain short-term (e.g., ephemeral secret parameters) or long-term (e.g., master key) parameters. Thus, the adversary can try to calculate network participants' sensitive information using ephemeral secret parameters or the master key of an AR cloud. The adversary also can obtain the legitimate AR user's secret value using power analysis attacks [24]. Therefore, the adversary can conduct various security attacks as follows:

- The adversary can try to impersonate a legitimate user [25].
- The adversary can attempt to compute the session key using secret parameters [26].
- The adversary can try to reveal the real identity or the location information of the AR user [27].
- The adversary can conduct various security threats, including insider, replay, verification table leakage, and man-in-the-middle attacks [28].

3.3. Extended Chebyshev Chaotic Maps

The extended Chebyshev chaotic maps are a cryptosystem based on the Chebyshev polynomials [29]: $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$. Note that $n \geq 2$, $n \in Z$, $x \in [-\infty, \infty]$, $T_0(x) = 1$, and $T_1(x) = x$. Moreover, the extended Chebyshev chaotic maps can be described as the following equation: $T_n(x) = cos(n * arccos(x))$. Therefore, the extended Chebyshev chaotic maps have the property of being a semi-group.

$$T_a(T_b(x)) = T_b(T_a(x)) = T_{ab}(x)$$

The extended Chebyshev chaotic maps can be utilized as a cryptosystem using the following mathematical problems:

- Extended chaotic map-based discrete logarithm problem (ECMDLP): A problem to find an integer a when x and y are given (Equation: $T_a(x) = y$).
- Extended chaotic map-based computational Diffie–Hellman (ECMCDH): A problem to calculate $T_{ab}(x)$ when $T_a(x)$ and $T_b(x)$ are given.
- Extended chaotic map-based decisional Diffie–Hellman (ECMDDH): A problem to decide $T_{ab}(x) \stackrel{?}{=} T_m$ when T_a, T_b, and T_m are given.

3.4. Physical Unclonable Functions

PUFs are the technology that makes physical replication impossible. PUFs utilize differences in the microstructure of semiconductors produced in the same manufacturing process. Thus, the same PUF-based devices output different results when the same values are input. PUFs can be utilized as a fingerprint for various devices by creating a unique security key that cannot be physically copied. In this paper, we define an equation $R = PUF(C)$ where R, $PUF(.)$, and C indicate response, PUF function, and challenge, respectively. We utilize PUF technology to preserve the security of the private key for ECNs.

3.5. Security Model

We introduce the security model of the proposed protocol for the ROR model [9]. The ROR model is a formal analysis to verify a secure session key agreement of the security protocol. We define the primitives which are used in the ROR model.

3.5.1. Participants

Participants are the network entities in the proposed protocol. Thus, we define $AR_U^{n_1}$, $AR_E^{n_2}$, and $AR_C^{n_3}$ as AR user, ECN, and AR cloud, respectively. Note that n_k ($k = 1, 2, 3$) is the instance of the participants.

3.5.2. Partnering

If two instances have session identity *sid* and the values are the same, they become partners. In the proposed protocol, it can be considered as a partnering when $AR_U^{n_1}$ and $AR_E^{n_2}$ share unique values, such as *sid*.

3.5.3. Freshness

If the adversary is unable to reveal the session key SK and thus cannot distinguish between SK and random nonce, we consider instance AR^{n_k} to be fresh.

3.5.4. Adversary

In the ROR model, the adversary can capture, delete, hijack, and insert messages over the public channel. The adversary conducts the following queries with the above abilites.

- $Execute(AR_U^{n_1}, AR_E^{n_1}, AR_C^{n_1})$: The adversary can obtain the content of messages over the public channel.
- $Send(AR^{n_k})$: The adversary actively transmits messages to a participant AR^{n_k} and receives the return message. This query can be considered as active attack.
- $Corrupt(AR_U^{n_1})$: The adversary reveals the secret values from the smart device of an AR user. This query can be considered as active attack.
- $Test(AR^{n_k})$: When the adversary conducts this query, the adversary obtains a flipped unbiased coin f. If the flipped coin shows 1 ($f = 1$), it means that the adversary cannot distinguish the session key sk and random number. Thus, sk can be considered as fresh and secure. When the flipped coin shows 0 ($f = 0$), sk is not fresh. Otherwise, the result outputs $NULL$ value ($\perp$).

4. Proposed Protocol

We introduce the proposed protocol which is designed for edge computing-based AR environments. The proposed protocol consists of six phases: Initialization, ECN registration, AR user registration, U2I authentication, AR service user search, and U2U authentication phases. Table 2 presents the explanation for each notation.

Table 2. Notations and descriptions.

Notation	Explanation
ID_{AM}	Real identity of ECN
ID_{AU}	Real identity of AR user
PW_{AU}	Password of AR user
PID_s	Pseudo-identity of AR user
TID_s	Temporary identity of AR user
r_s	Random number
t_s	Timestamp
$T_s(z)$	Chebyshev chaotic maps operator
$h(.)$	Hash function
$PUF(.)$	PUF operator
C_i	Challenge of PUF
R_i	Response of PUF
SK, K_{AR}	Session key
$\oplus$	XOR operator
$\|\|$	Concatenation operator
$\cdot$	Multiplication operator

4.1. Initialization Phase

The AR cloud selects its master key X_{AR} and a large prime number q. Then, the AR cloud selects $z \in [-\infty, \infty]$ for $T_n(z) = 2zT_{n-1}(z) - T_{n-2}(z)\ mod\ q$. From that, the AR cloud computes its public key $P_{AR} = T_{X_{AR}}(z)$ and picks a hash function $h(.)$. Finally, the AR cloud publishes $\{P_{AR}, q, z, h(.)\}$ to the network.

4.2. ECN Registration Phase

MNR1: An ECN selects its identity ID_{AM} and generates a random number R_{ECN}. Then, the ECN computes $RID_{AM} = h(ID_{AM} \parallel R_{ECN})$ and sends $\{ID_{AM}, RID_{AM}\}$ to the AR cloud via a secure channel.

MNR2: The AR cloud checks the validity of ID_{AM} and generates N_{AM} and $C_{AM} = [C_1, C_2, \ldots, C_i, \ldots, C_n]$. After that, the AR cloud computes $R_{AM} = h(N_{AM} \parallel RID_{AM})$ and sends $\{R_{AM}, C_{AM}\}$ to the ECN through a secure channel.

MNR3: The ECN computes $S_{AM} = PUF(C_{AM}) = [R_1, R_2, \ldots, R_i, \ldots, R_n]$, fuzzy extractor [30] $EH_{AM} = Gen(S_{AM}) = [(E_1, H_1), \ldots, (E_i, H_i), \ldots, (E_n, H_n)]$, $Gen(PUF(h(ID_{AM} \parallel R_{ECN}))) = (M_{ECN}, H_{ECN})$, and $P_{AM} = T_{M_{ECN}}(z)$. More-

over, the ECN stores $\{ID_{AM}, H_{ECN}, S_{AM}, H_{AM}\}$ in its secure database and sends $\{EH_{AM}, P_{AM}\}$ to the AR cloud.

MNR4: The AR cloud computes $SC_{AM} = C_{AM} \oplus h(X_{AR} \parallel ID_{AM} \parallel P_{AM})$ and $SE_{AM} = E_{AM} \oplus h(C_{AM} \parallel X_{AR} \parallel ID_{AM})$, and it stores $\{SC_{AM}, SE_{AM}, P_{AM}, ID_{AM}\}$ in its secure database.

4.3. AR User Registration Phase

AUR1: An AR user selects an identity ID_{AU} and password PW_{AU}. Then, the AR user generates R_{AU} and computes $RID_{AU} = h(ID_{AU} \parallel R_{AU})$, $RPW_{AU} = h(PW_{AU} \parallel R_{AU} \parallel ID_{AU})$, $M_{AU} = h(ID_{AU} \parallel RID_{AU} \parallel PW_{AU})$, and $P_{AU} = T_{M_{AU}}(z)$. The AR user sends $\{ID_{AU}, RID_{AU}, P_{AU}\}$ to the AR cloud via a secure channel.

AUR2: The AR cloud checks the identity of AR user ID_{AU} and generates R_{AU}, $r_E \in \{2^4, 2^8\}$. The AR cloud retrieves SC_{AMi} and SE_{AMi} from SC_{AM} and SE_{AM}, respectively. Then, the AR cloud computes $C_i = SC_{AMi} \oplus h(X_{AR} \parallel ID_{AM} \parallel P_{AM})$, $E_i = SE_{AMi} \oplus h(C_i \parallel X_{AR} \parallel ID_{AM})$, $VU_i = h(C_i \parallel E_i \parallel P_{AM})$, $PID_{AU} = h(R_{UC} \parallel RID_{AU} \parallel P_{AU})$, $SID_{AU} = ID_{AU} \oplus h(X_{AR} \parallel P_{AU} \parallel PID_{AU})$, and $SCE_{AU} = C_i \oplus h(ID_{AU} \parallel X_{AR} \parallel P_{AU})$. After that, the AR cloud stores $\{PID_{AU}, P_{AU}, SID_{AU}, SCE_{AU}\}$ in its database and returns $\{PID_{AU}, VU_i, C_i, r_E\}$ to the AR user through a secure channel.

AUR3: The AR user computes $APID_{AU} = PID_{AU} \oplus h(ID_{AU} \parallel P_{AU} \parallel PW_{AU})$, $AR_{AU} = R_{AU} \oplus h(PID_{AU} \parallel PW_{AU} \parallel RPW_{AU})$, $AVU_{AU} = VU_i \oplus h(R_{AU} \parallel RID_{AU} \parallel PW_{AU})$, $AC_{AU} = C_i \oplus h(VU_i \parallel RPW_{AU} \parallel ID_{AU})$, and $V_{AU} = h(M_{AU} \parallel PID_{AU} \parallel P_{AU} \parallel VU_i \parallel C_i) \bmod r_E$. Then, the AR user stores $\{P_{AU}, APID_{AU}, AR_{AU}, AVU_{AU}, AC_{AU}, V_{AU}, r_E\}$ in the memory.

4.4. U2I Authentication Phase

After the registration phase, the AR user enters the management region of ECN and sends an initial authentication request message to ECN. Then, the ECN and AR user perform a U2I authentication process. Figure 3 shows the U2I authentication phase, and the details are as follows.

U2I1: The AR user inputs ID_{AU} and PW_{AU}, and computes $RID_{AU} = h(ID_{AU} \parallel R_{AU})$, $PID_{AU} = APID_{AU} \oplus h(ID_{AU} \parallel P_{AU} \parallel PW_{AU})$, $R_{AU} = AR_{AU} \oplus h(PID_{AU} \parallel PW_{AU} \parallel RPW_{AU})$, $RPW_{AU} = h(PW_{AU} \parallel R_{AU} \parallel ID_{AU})$, $VU_i = AVU_{AU} \oplus h(R_{AU} \parallel RID_{AU} \parallel PW_{AU})$, $C_i = AC_{AU} \oplus h(VU_i \parallel RPW_{AU} \parallel ID_{AU})$, and $V'_{AU} = h(M_{AU} \parallel PID_{AU} \parallel P_{AU} \parallel VU_i \parallel C_i) \bmod r_E$. After checking $V'_{AU} \stackrel{?}{=} V_{AU}$, the AR user generates a random nonce r_1 and timestamp t_1, and they compute $A_1 = T_{r_1}(z)$, $A_2 = T_{r_2}(P_{AM})$, $TID_1 = PID_{AU} \oplus h(t_1 \parallel A_2 \parallel ID_{AM})$, $MC_1 = C_i \oplus h(PID_{AU} \parallel A_1 \parallel A_2)$, and $V_1 = h(VU_i \parallel C_i \parallel PID_i \parallel A_2 \parallel t_1 \parallel ID_{AM})$. The AR user generates an authentication request message $\{TID_1, A_1, MC_1, V_1, t_1\}$ and sends it to the ECN via an open channel.

U2I2: The ECN first checks $|t_1 - t_c| < \Delta t$ and computes $Rep(PUF(h(ID_{AM} \parallel R_{ECN})), H_{ECN}) = M_{ECN}$, $A'_2 = T_{M_{ECN}}(A_1)$, $PID'_{AU} = TID_1 \oplus h(t_1 \parallel A'_2 \parallel ID_{AM})$, $C'_i = MC_1 \oplus h(PID'_{AU} \parallel A_1 \parallel A'_2)$, $S'_i = PUF(C'_i)$, $E'_i = Rep(S'_i, H_i)$, $VU'_i = h(C'_i \parallel E'_i \parallel PID'_{AU} \parallel P_{AM})$, and $V'_1 = h(VU'_i \parallel C'_i \parallel PID'_{AU} \parallel A'_2 \parallel t_1 \parallel ID_{AM})$. When the result of $V'_1 \stackrel{?}{=} V_1$ is correct, the ECN generates r_2 and t_2 and computes $A_3 = T_{r_2}(z)$, $A_4 = T_{r_2}(A_1)$, a session key $SK = h(A_4 \parallel A_2 \parallel t_2 \parallel VU_i \parallel PID_{AU} \parallel ID_{AM})$, and $V_2 = h(A_4 \parallel SK \parallel t_2 \parallel VU_i \parallel A_2)$. Then, the ECN sends $\{A_3, V_2, t_2\}$ to the AR user through an open channel.

U2I3: The AR user checks $|t_2 - t_c| < \Delta t$ and computes $A'_4 = T_{r_1}(A_3)$, $SK' = h(A'_4 \parallel A_2 \parallel t_2 \parallel VU_i \parallel PID_{AU} \parallel ID_{AM})$, and $V'_2 = h(A'_4 \parallel SK' \parallel t_2 \parallel VU_i \parallel A_2)$. If $V'_2 \stackrel{?}{=} V_2$, the session key is completely agreed.

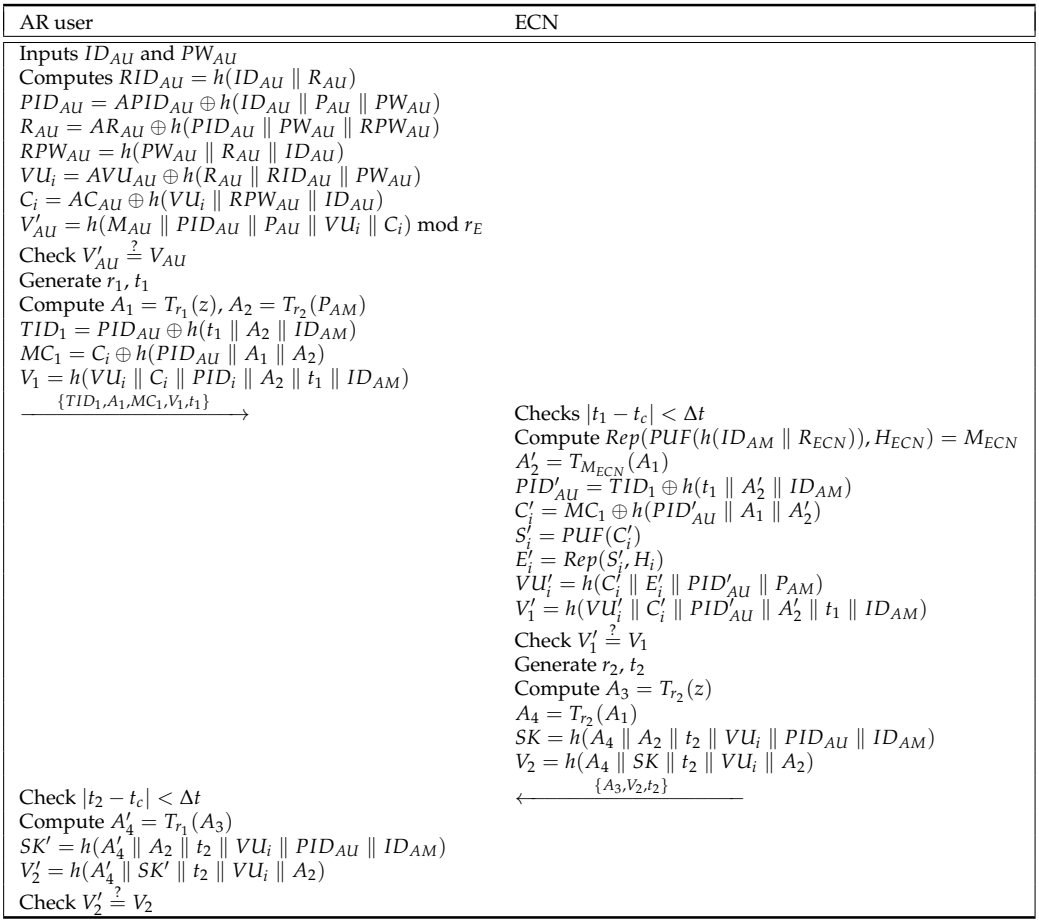

Figure 3. U2I authentication phase of the proposed scheme.

4.5. AR Service User Search Phase

After the U2I authentication phase, the AR user requests AR data from the ECN. If the ECN has a lot of work or is overloaded, it transmits search results about nearby users who own AR service data. Details are as follows.

AUS1: The AR user generates an AR request message M_{AR} and timestamp t_{AR}, and they compute $V_{AR} = h(M_{AR} \| TID_1 \| t_{AR} \| SK)$, $C_{AR} = AES(TID_1, M_{AR}, V_{AR})_{SK}$. Then, the AR user sends $\{C_{AR}, t_{AR}\}$ to the ECN.

AUS2: After checking $|t_{AR} - t_c| < \Delta t$, the ECN decrypts $(TID_1, M_{AR}, V_{AR}) = D(C_{AR})_{SK}$ and computes $V'_{AR} = h(M_{AR} \| TID_1 \| t_{AR} \| SK)$. If $V'_{AR} \stackrel{?}{=} V_{AR}$, the ECN searches a M_{AR} service user TID_M. Then, the ECN generates t_M and computes $V_M = h(SK_M \| PID_M \| M_{AR})$, $V_{AR2} = h(TID_M \| P_M \| t_M \| V_M)$, and $C_M = AES(TID_M, V_M, V_{AR2}, P_M)_{SK}$. Note that SK_M, PID_M, V_M are the session key, pseudo-identity, and verification parameter for TID_M. Finally, the ECN sends $\{C_M, t_M\}$ to the AR user via an open channel.

AUS3: The AR user checks $|t_M - t_c| < \Delta t$ and decrypts C_M to obtain $(TID_M, V_M, V_{AR2}, P_M)$. The AR user computes $V'_{AR2} = h(TID_M \| P_M \| t_M \| V_M)$ and checks $V'_{AR2} \stackrel{?}{=} V_{AR2}$.

4.6. U2U Authentication Phase

The AR user broadcasts an authentication request message. The AR service user who owns AR data inspects this message and conducts the U2U authentication process. Figure 4 indicates the U2U authentication phase, and the details are as follows.

U2U1: The AR user generates r_3, t_3 and computes $A_5 = T_{r_3}(z)$, $A_6 = T_{r_3}(P_M)$, $MC_2 = (M_{AR} \parallel P_{AU}) \oplus h(A_6 \parallel TID_M \parallel t_3 \parallel A_5)$, and $V_3 = h(M_{AR} \parallel A_6 \parallel V_M \parallel P_{AU} \parallel t_3)$. Then, the AR user sends $\{A_5, MC_2, V_3, t_3\}$ to the AR service user ID_M.

U2U2: The AR service user ID_M first checks $|t_3 - t_c| < \Delta t$ and computes $A'_6 = T_{M_M}(A_5)$, $(M'_{AR} \parallel P'_{AU}) = MC_2 \oplus h(A'_6 \parallel TID_M \parallel t_3 \parallel A_5)$, $V'_M = h(SK_M \parallel PID'_M \parallel M'_{AR})$, and $V'_3 = h(M'_{AR} \parallel A'_6 \parallel V'_M \parallel P'_{AU} \parallel t_3)$. If $V'_3 \stackrel{?}{=} V_3$, the AR service user ID_M generates r_4, t_4 and computes $A_7 = T_{r_4}(z)$, $A_8 = T_{M_M}(P_{AU})$, $A_9 = T_{r_4}(A_5)$, $K_{AR} = h(A_8 \parallel A_9 \parallel t_4 \parallel TID_M \parallel M_{AR})$, and $V_4 = h(K_{AR} \parallel A_7 \parallel A_8 \parallel A_9 \parallel t_4 \parallel M_{AR})$. Then, the AR service user sends $\{A_7, V_4, t_4\}$ to the AR user via an open channel.

U2U3: The AR user checks $|t_4 - t_c| < \Delta t$ and computes $A'_8 = T_{M_{AR}}(P_M)$, $A'_9 = T_{r_3}(A_7)$, $K'_{AR} = h(A'_8 \parallel A'_9 \parallel t_4 \parallel TID_M \parallel M_{AR})$, and $V'_4 = h(K'_{AR} \parallel A_7 \parallel A'_8 \parallel A'_9 \parallel t_4 \parallel M_{AR})$. If $V'_4 \stackrel{?}{=} V_4$, the U2U authentication phase is successful and the AR user receives the AR contents using the session key K_{AR}.

AR user	AR service user (ID_M)
Generate r_3, t_3	
Compute $A_5 = T_{r_3}(z)$	
$A_6 = T_{r_3}(P_M)$	
$MC_2 = (M_{AR} \parallel P_{AU}) \oplus h(A_6 \parallel TID_M \parallel t_3 \parallel A_5)$	
$V_3 = h(M_{AR} \parallel A_6 \parallel V_M \parallel P_{AU} \parallel t_3)$	
$\xrightarrow{\{A_5, MC_2, V_3, t_3\}}$	
	$\lvert t_3 - t_c \rvert < \Delta t$
	Computes $A'_6 = T_{M_M}(A_5)$
	$(M'_{AR} \parallel P'_{AU}) = MC_2 \oplus h(A'_6 \parallel TID_M \parallel t_3 \parallel A_5)$
	$V'_M = h(SK_M \parallel PID'_M \parallel M'_{AR})$
	$V'_3 = h(M'_{AR} \parallel A'_6 \parallel V'_M \parallel P'_{AU} \parallel t_3)$
	Check $V'_3 \stackrel{?}{=} V_3$
	Generate r_4, t_4
	Compute $A_7 = T_{r_4}(z)$
	$A_8 = T_{M_M}(P_{AU})$
	$A_9 = T_{r_4}(A_5)$
	$K_{AR} = h(A_8 \parallel A_9 \parallel t_4 \parallel TID_M \parallel M_{AR})$
	$V_4 = h(K_{AR} \parallel A_7 \parallel A_8 \parallel A_9 \parallel t_4 \parallel M_{AR})$
	$\xleftarrow{\{A_7, V_4, t_4\}}$
Check $\lvert t_4 - t_c \rvert < \Delta t$	
Compute $A'_8 = T_{M_{AR}}(P_M)$	
$A'_9 = T_{r_3}(A_7)$	
$K'_{AR} = h(A'_8 \parallel A'_9 \parallel t_4 \parallel TID_M \parallel M_{AR})$	
$V'_4 = h(K'_{AR} \parallel A_7 \parallel A'_8 \parallel A'_9 \parallel t_4 \parallel M_{AR})$	
Check $V'_4 \stackrel{?}{=} V_4$	

Figure 4. U2U authentication phase of the proposed scheme.

5. Security Analysis

To prove the security robustness of the proposed protocol, we conduct informal security analysis, using the Scyther tool, BAN logic, and the ROR model.

5.1. Informal Security Analysis

5.1.1. Replay and Man-in-the-Middle Attack

In our protocol, the network participants send messages with random numbers r_1, r_2, r_3, and r_4, and timestamps t_1, t_2, t_3, and t_4. Thus, network participants can check the freshness of messages. Although an adversary captures and sends a message from a previous session, the network participants can filter it using the timestamp and random nonce. Therefore, the proposed protocol can prevent replay and man-in-the-middle attacks.

5.1.2. AR User Impersonation Attack

Suppose that an adversary captures messages of a legitimate user and tries to impersonate the user. However, the messages transmitted through open channels are masked

in the public key of AR user P_{AU}, so the adversary must obtain the private key M_{AU}. Moreover, the adversary cannot impersonate as the AR user without knowing the pseudo-identity PID_{AU}. These parameters are encrypted in the AR user's device with the identity and password. Thus, the proposed protocol is secure against impersonation attacks.

5.1.3. Privileged Insider Attack

If an adversary is a privileged insider in our proposed network system, the adversary can reveal and captures all of registration messages, including $\{ID_{AU}, RID_{AU}, P_{AU}\}$. Moreover, suppose that the adversary obtains secret parameters of the AR user $\{P_{AU}, APID_{AU}, AR_{AU}, AVU_{AU}, AC_{AU}, V_{AU}, r_E\}$. To calculate and decrypt these parameters, the adversary must guess the password PW'_{AU} of the AR user. However, our protocol utilizes fuzzy verifier r_E, so the probability of guessing correct PW_{AU} is $2^8/|hash| \approx 1/10^{15}$. Therefore, the proposed protocol can prevent privileged insider attacks.

5.1.4. Verification Table Leakage Attack

Suppose that the adversary obtains the leaked-verification table $\{SC_{AM}, SE_{AM}, P_{AM}, ID_{AM}\}$ and $\{PID_{AU}, P_{AU}, SID_{AU}, SCE_{AU}\}$. From that, the adversary tries to reveal and attack the proposed network. However, the adversary cannot reveal any secret parameters without knowing the master key of AR cloud X_{AR}. Therefore, the proposed protocol can prevent verification table leakage attacks.

5.1.5. Ephemeral Secret Leakage Attack

If an adversary obtains the ephemeral secret parameters r_1, r_2, r_3, and r_4, the adversary tries to compute the session key $SK = h(A_4 \parallel A_2 \parallel t_2 \parallel VU_i \parallel PID_{AU} \parallel ID_{AM})$ and $K_{AR} = h(A_8 \parallel A_9 \parallel t_4 \parallel TID_M \parallel M_{AR})$. To calculate the session key, the adversary must obtain A_2, A_4, A_8, and A_9. However, this task is infeasible according to the ECMCDH problem. Thus, the proposed protocol is secure against ESL attacks.

5.1.6. Anonymity

If AR users send their real identity ID_{AU}, it can cause various security and privacy problems, including traceability. In our protocol, AR users use the temporary identities TID_k, which are used in a specific session. Thus, the adversary cannot distinguish and trace the AR user. Thus, the proposed protocol can ensure AR user anonymity.

5.1.7. Perfect Forward Secrecy

Suppose that an adversary obtains the master key X_{AR} and the public messages $\{TID_1, A_1, MC_1, V_1, t_1\}$, $\{A_3, V_2, t_2\}$, $\{C_{AR}, t_{AR}\}$, $\{C_M, t_M\}$, $\{A_5, MC_2, V_3, t_3\}$, and $\{A_7, V_4, t_4\}$. However, the adversary cannot calculate the session key SK and K_{AR} because these parameters are composed of the secret parameters of AR users and ECNs, extended chaotic maps, and ephemeral secret parameters. Thus, the proposed protocol can guarantee perfect forward secrecy.

5.1.8. Mutual Authentication

In our protocol, all messages include timestamps t_1, t_2, t_3, and t_4, and network participants check the validity of them. If these processes are successful, the network participants decrypt messages and check the integrity using V_1, V_2, V_3, and V_4. When the integrity correctness process is complete, the mutual authentication is successful. Thus, the proposed protocol can ensure mutual authentication.

5.2. Scyther Tool

In this section, we simulate the security robustness of the proposed protocol using the Scyther tool. To simulate the proposed protocol, we first convert it into "Security Protocol Description Language (SPDL)", which is the specific language for the Scyther tool. After that, the Scyther command-line tool verifies the security of the proposed protocol

using various claim events. When the simulation is complete, we can obtain the result window which indicates the security robustness of the proposed protocol. If the result window shows "OK" in the "Status" tab and "No attacks" in the "Comment" tab, we can ensure that the proposed protocol has a secure authentication process. Table 3 shows the claim events and Figure 5 indicates the result window of the proposed protocol.

Table 3. Scyther tool claim events.

Claim Event	Description
Secrecy	Integrity and confidentiality of an authentication parameter.
Alive	Checking the status whether the communication partner is active or not.
Weakagree	Checking the status whether the communication partner is the actual user or not (Simultaneously satisfying alive claim event).
Niagree	The data-set is agreed by network participants who satisfies alive and weakagree claim events.
Nisynch	The network participants conduct communications under instructions of the protocol (simultaneously satisfying alive, weak-agree, and no-agree claim events).

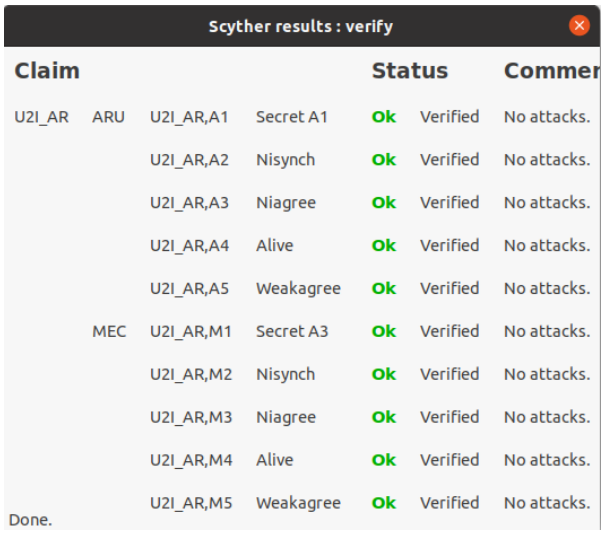

Figure 5. Scyther result window of the proposed protocol.

5.3. BAN Logic

BAN logic [8] is a formal analysis method to prove the mutual authentication of the protocol. Thus, various security protocols analyzed the property of mutual authentication using BAN logic [31–33]. To analyze the proposed protocol using BAN logic, we define basic notations and descriptions in Table 4.

Table 4. Basic notations and description.

Notation	Description
R_i, R_j	Principals
SK	Session key
A_1, A_2	Statements
$R_i \mid \equiv A_1$	R_i **believes** A_1
$R_1 \mid \sim A_1$	R_i once **said** A_1
$R_i \mapsto A_1$	R_i **controls** A_1
$R_i \lhd A_1$	R_i **receives** A_1
$\#A_1$	A_1 is **fresh**
$\{A_1\}_S$	A_1 is **encrypted** with S
$R_i \overset{KS}{\longleftrightarrow} R_j$	R_i and R_j have a shared key KS

5.3.1. Rules

In BAN logic, there are five rules. The details are outlined below.

1. Message meaning rule (MMR):

$$\frac{R_i \mid \equiv R_i \overset{KS}{\leftrightarrow} R_j, \quad R_i \lhd \{A_1\}_{SH}}{R_i \mid \equiv R_j \mid \sim A_1}$$

2. Nonce verification rule (NVR):

$$\frac{R_i \mid \equiv \#(A_1), \quad R_i \mid \equiv R_j \mid \sim A_1}{R_i \mid \equiv R_j \mid \equiv A_1}$$

3. Jurisdiction rule (JR):

$$\frac{R_i \mid \equiv R_j \mapsto A_1, \quad R_i \mid \equiv R_j \mid \equiv A_1}{R_i \mid \equiv A_1}$$

4. Belief rule (BR):

$$\frac{R_i \mid \equiv (A_1, A_2)}{R_i \mid \equiv A_1}$$

5. Freshness rule (FR):

$$\frac{R_i \mid \equiv \#(A_1)}{R_i \mid \equiv \#(A_1, A_2)}$$

5.3.2. Goals

In our protocol, a session key is established between the network participants during the authentication phase. Therefore, the goals in our protocol are as follows.

Goal 1: $ARU \mid \equiv ECN \overset{SK}{\leftrightarrow} ARU$

Goal 2: $ARU \mid \equiv ECN \mid \equiv ARU \overset{SK}{\leftrightarrow} ECN$

Goal 3: $ECN \mid \equiv ARU \overset{SK}{\leftrightarrow} ARU$

Goal 4: $ECN \mid \equiv ARU \mid \equiv ECN \overset{SK}{\leftrightarrow} ARU$

5.3.3. Idealized Forms

In our protocol, an AR user and ECN exchange $\{TID_1, A_1, MC_1, V_1, t_1\}$ and $\{A_3, V_2, t_2\}$ through open channels. Thus, we convert these messages into idealized forms.

IF 1: $ARU \rightarrow ECN : \{PID_{AU}, A_1, C_i, t_1\}_{A_2}$

IF 2: $ECN \rightarrow ARU : \{ID_{AM}, A_3, t_2\}_{A_4}$

5.3.4. Assumptions

In our protocol, assumptions are as follows.

Assumption 1. $ECN \mid \equiv \#(t_1)$.

Assumption 2. $ARU \mid \equiv \#(t_2)$.

Assumption 3. $ECN \mid \equiv ARU \overset{A_2}{\leftrightarrow} ECN$.

Assumption 4. $ARU \mid \equiv ECN \overset{A_4}{\leftrightarrow} ARU$.

5.3.5. BAN Logic Analysis

Step 1: From IF_1, we can obtain AN_1.

$$AN_1 : ECN \lhd \{PID_{AU}, A_1, C_i, t_1\}_{A_2}$$

Step 2: We can obtain AN_2 utilizing the message meaning rule and **Assumption 3**.

$$AN_2 : ECN| \equiv ARU| \sim (PID_{AU}, A_1, C_i, t_1)$$

Step 3: We can obtain AN_3 utilizing the freshness rule and **Assumption 1**.

$$AN_3 : ECN| \equiv \#(PID_{AU}, A_1, C_i, t_1)$$

Step 4: We can obtain AN_4 utilizing the nonce verification rule, AN_2, and AN_3.

$$AN_4 : ECN| \equiv ARU| \equiv (PID_{AU}, A_1, C_i, t_1)$$

Step 5: From IF_2, we can obtain AN_5.

$$AN_5 : ARU \lhd \{PID_{AM}, A_3, t_2\}_{A_4}$$

Step 6: We can obtain AN_6 utilizing the message meaning rule and **Assumption 4**.

$$AN_6 : ARU| \equiv ECN| \sim (ID_{AM}, A_3, t_2)$$

Step 7: We can obtain AN_7 utilizing the freshness rule and **Assumption 2**.

$$AN_7 : ARU| \equiv \#(ID_{AM}, A_3, t_2)$$

Step 8: We can obtain AN_8 utilizing the nonce verification rule, AN_6, and AN_7.

$$AN_8 : ARU| \equiv ECN| \equiv (ID_{AM}, A_3, t_2)$$

Step 9: In our protocol, ARU and ECN establish the session key $SK = h(A_4 \parallel A_2 \parallel t_2 \parallel VU_i \parallel PID_{AU} \parallel ID_{AM})$. Thus, we use AN_4 and AN_8 to obtain AN_9 and AN_{10}.

$$AN_9 : ARU| \equiv ECN| \equiv ARU \overset{SK}{\longleftrightarrow} ECN \quad \textbf{(Goal 2)}$$
$$AN_{10} : ECN| \equiv ARU| \equiv ECN \overset{SK}{\longleftrightarrow} ARU \quad \textbf{(Goal 4)}$$

Step 10: We can obtain AN_{11} and AN_{12} using AN_9, AN_{10}, and the jurisdiction rule.

$$AN_{11} : ARU| \equiv ECN \overset{SK}{\longleftrightarrow} ARU \quad \textbf{(Goal 1)}$$
$$AN_{12} : ECN| \equiv ARU \overset{SK}{\longleftrightarrow} ECN \quad \textbf{(Goal 3)}$$

5.4. ROR Model

In this section, we analyze the session key security of the proposed protocol using the ROR model [9]. We utlize the security model in Section 3.5 and follow the security proof of [34–36] which proved the session key security using the ROR model.

Theorem 1. *In the ROR model, the adversary tries to reveal the session key in polynomial time. Thus, let $SPD_{adv}(PT)$ be the probability that the session key security is broken in polynomial time. We also define HS, q_{hs}, PF, q_{pf} and $SPD_{adv}^{ECMDDH}(PT)$ as hash function $h(.)$'s range space, the number of hash queries, the function $PUF(.)$'s range space, the number of PUF queries, and probability to break the ECMDDH problem. C' and s' are the Zipf's parameters [37].*

$$SPD_{adv}(PT) \leq \frac{q_{hs}^2}{|HS|} + \frac{q_{pf}^2}{|PF|} + 2SPD_{adv}^{ECMDDH}(PT) + 2\{C' q_{send}^{s'}\} \quad (1)$$

We perform six games G_n $(n = 0, 1, 2, 3, 4, 5)$ to prove the security of the session key in our proposed protocol. In each game G_n, the winning probability of the adversary and the advantage is $WPA_{adv}(G_n)$ and $P[WPA_{adv}(G_n)]$, respectively.

- G_0: It is a stating game where the adversary has no information about the session key. Thus, the adversary selects a random bit. Thus, the relationship between $SPD_{adv}(PT)$ and $WPA_{adv}(G_0)$ can be expressed as follows.

$$SPD_{adv}(PT) = |2P[WPA_{adv}(G_0)] - 1| \tag{2}$$

- G_1: In this game, the adversary collects messages transmitted over public channels using the *Execute* query. Thus, the adversary obtains $\{TID_1, A_1, MC_1, V_1, t_1\}$, $\{A_3, V_2, t_2\}$, $\{A_5, MC_2, V_3, t_3\}$, and $\{A_7, V_4, t_4\}$. With this information, the adversary tries to reveal the session key $SK' = h(A'_4 \parallel A_2 \parallel t_2 \parallel VU_i \parallel PID_{AU} \parallel ID_{AM})$ and $K'_{AR} = h(A'_8 \parallel A'_9 \parallel t_4 \parallel TID_M \parallel M_{AR})$. After that, the adversary conducts a *Test* query to distinguish the session key and random number. However, the adversary cannot guess a successful session key because each parameter in the message is masked in various security parameters, including r_1, r_2, and A_2. Thus, we obtain the following equation.

$$P[WPA_{adv}(G_1)] = P[WPA_{adv}(G_0)] \tag{3}$$

- G_2: The adversary performs *Send* and hash queries. However, the adversary cannot have an advantage in this game because all of security parameters are masked in cryptographic one-way hash functions, which has resistance against hash collision problems. Using birthday paradox [38], we can obtain the following inequation.

$$|P[WPA_{adv}(G_2)] - P[WPA_{adv}(G_1)]| \leq \frac{q_{hs}^2}{|HS|} \tag{4}$$

- G_3: In this game, the adversary performs *Send* and PUF queries. As we mentioned in Section 3.4, the adversary cannot have an advantage because PUF is an anti-duplicated microstructure. Thus, we can obtain the inequation similar to (4).

$$|P[WPA_{adv}(G_3)] - P[WPA_{adv}(G_2)]| \leq \frac{q_{pf}^2}{|PUF|} \tag{5}$$

- G_4: The adversary tries to reveal the session key calculating A_2, A_4, A_6, and A_8. However, the adversary must solve the ECMDDH problem to find random numbers r_1, r_2, r_3, and r_4. However, this process is not possible in polynomial time. Thus, we obtain the following inequation.

$$|P[WPA_{adv}(G_4)] - P[WPA_{adv}(G_3)]| \leq SPD_{adv}^{ECMDDH}(PT) \tag{6}$$

- G_5: This is the final game that the adversary conducts, using the *Corrupt* query. With the revealed parameters $\{P_{AU}, APID_{AU}, AR_{AU}, AVU_{AU}, AC_{AU}, V_{AU}, r_E\}$ from the AR user's smart device, the adversary tries to guess the session key. However, the adversary cannot reveal the real identity and password of the AR user because the verification parameter V_{AU} is processed under fuzzy verifier r_E. Thus, the adversary cannot guess ID_{AU} and PW_{AU} simultaneously. As a result, we obtain the following inequation using Zipf's law [37].

$$|P[WPA_{adv}(G_5)] - P[WPA_{adv}(G_4)]| \leq \{C' q_{send}^{s'}\} \tag{7}$$

When all the games are finished, the adversary guesses a random bit.

$$P[WPA_{adv}(G_5)] = \frac{1}{2} \tag{8}$$

We derive the following equation from (2) and (3).

$$\frac{1}{2}SPD_{adv}(PT) = |P[WPA_{adv}(G_0)] - \frac{1}{2}| = |P[WPA_{adv}(G_1)] - \frac{1}{2}| \quad (9)$$

Combining Equations (8) and (9), we obtain the following:

$$\frac{1}{2}SPD_{adv}(PT) = |P[WPA_{adv}(G_1)] - P[WPA_{adv}(G_5)]| \quad (10)$$

Using the triangular inequality, we obtain the following:

$$\begin{aligned} &\frac{1}{2}SPD_{adv}(PT) = |P[WPA_{adv}(G_1)] - P[WPA_{adv}(G_5)]| \\ &\leq |P[WPA_{adv}(G_1)] - P[WPA_{adv}(G_4)]| + |P[WPA_{adv}(G_4)] - P[WPA_{adv}(G_5)]| \\ &\leq |P[WPA_{adv}(G_1)] - P[WPA_{adv}(G_2)]| + |P[WPA_{adv}(G_2)] - P[WPA_{adv}(G_3)]| \\ &|P[WPA_{adv}(G_3)] - P[WPA_{adv}(G_4)]| + |P[WPA_{adv}(G_4)] - P[WPA_{adv}(G_5)]| \\ &\leq \frac{q_{hs}^2}{2|HS|} + \frac{q_{pf}^2}{2|PF|} + SPD_{adv}^{ECMDDH}(PT) + \{C'q_{send}^{s'}\} \end{aligned} \quad (11)$$

At last, we obtain the following inequation multiplying (11) by 2. Thus, we can prove **Theorem 1**.

$$SPD_{adv}(PT) \leq \frac{q_{hs}^2}{|HS|} + \frac{q_{pf}^2}{|PF|} + 2SPD_{adv}^{ECMDDH}(PT) + 2\{C'q_{send}^{s'}\} \quad (12)$$

6. Performance Analysis

In this section, we conduct comparative studies about computational, communicational, and security features. To estimate the performance of the proposed protocol, we utilize the MIRACL testbed. We compare the performance result of the proposed protocol and the other related schemes [18–21] which are proposed for U2U and U2I authentication schemes.

6.1. Execution Time of Cryptographic Primitives Using MIRACL

The MIRACL is a C/C++ language-based cryptographic SDK that facilitates the development of cryptographic schemes. There are built-in libraries for various cryptographic primitives, including ECC, bilinear pairings, hash functions, and symmetric encryptions. Thus, we measure the execution time of each primitive using MIRACL in the desktop (11th Generation Intel(R) Core(TM) i5-11400 @ 2.60 GHz with 24.0 GB RAM) and Raspberry-PI 4 (Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit System on Chip @ 1.8 GHz, with 8 GB RAM) testbed environments. We measured the execution time for each cryptographic primitive 100 times. We defined the ECC multiplication, ECC addition, hash function, AES encryption, AES decryption, modular exponentiation, and bilinear pairing as M_{em}, M_{ea}, M_{hf}, M_{ae}, M_{ad}, M_{me}, and M_{bp}, respectively. Tables 5 and 6 show the performance results for the max, min, and the average time for each primitive using MIRACL. The testbed results show that the desktop platform has much better performance compared to Raspberry-PI 4. Thus, we apply the desktop and Raspberry-PI 4 platform to infrastructure and user devices, respectively.

Table 5. Performance results on the desktop platform using MIRACL.

Primitives	Max Time	Min Time	Average Time
M_{em}	0.539 ms	0.385 ms	0.413 ms
M_{ea}	0.009 ms	0.002 ms	0.003 ms
M_{hf}	0.004 ms	0.001 ms	0.001 ms
M_{ae}	0.002 ms	0.001 ms	0.001 ms
M_{ad}	0.001 ms	0.001 ms	0.001 ms
M_{me}	0.083 ms	0.027 ms	0.040 ms
M_{bp}	2.653 ms	2.223 ms	2.235 ms

Table 6. Performance results in Raspberry-PI 4 platform using MIRACL.

Primitives	Max Time	Min Time	Average Time
M_{em}	3.008 ms	2.152 ms	2.375 ms
M_{ea}	0.032 ms	0.026 ms	0.028 ms
M_{hf}	0.009 ms	0.004 ms	0.007 ms
M_{ae}	0.007 ms	0.004 ms	0.004 ms
M_{ad}	0.008 ms	0.004 ms	0.005 ms
M_{me}	0.233 ms	0.052 ms	0.113 ms
M_{bp}	14.232 ms	13.753 ms	13.928 ms

6.2. Comparison of Computational Cost

We compare the computational cost of the proposed protocol and the related schemes [18–21]. The computational cost of M_{em}, M_{ea}, M_{hf}, M_{ae}, M_{ad}, M_{me}, and M_{bp} can be denoted as 0.413 ms, 0.003 ms, 0.001 ms, 0.001 ms, 0.001 ms, 0.040 ms, and 2.235 ms according to Table 5, and 2.235 ms, 0.028 ms, 0.007 ms, 0.004 ms, 0.005 ms, 0.113 ms, and 13.928 ms according to Table 6. Note that the computational cost of extended Chebyshev chaotic maps M_{cm} is one-third of M_{em} [39] (0.138 ms in a desktop and 0.792 ms in the Raspberry-PI platform). In addition, the computational cost of fuzzy extractor M_f is similar to M_{em}. The results of computational cost analysis are shown in Table 7. Thus, the proposed protocol can provide a more lightweight authentication process than the related schemes [18–21].

Table 7. Comparison of computational cost.

Scheme	Phase	User	Infrastructure	Total Costs
Chen et al. [18]	U2U	$7M_{em} + 2M_{ea} + 8M_{hf} + 3M_{bp}$	-	58.521 ms
Alzahrani et al. [19]	U2U	$6M_{em} + 8M_{hf}$	-	14.306 ms
Pham et al. [20]	U2U	$6M_{em} + 7M_{hf}$	-	14.299 ms
	U2I	$5M_{em} + 5M_{hf}$	$5M_{em} + 7M_{hf}$	13.982 ms
Hajian et al. [21]	U2U	$8M_{em} + 14M_{hf}$	-	19.098 ms
Proposed	U2U	$8M_{cm} + 9M_{hf}$	-	6.396 ms
	U2I	$3M_{cm} + 12M_{hf}$	$3M_{cm} + 2M_f + 6M_{hf}$	3.704 ms

6.3. Comparison of Communcational Cost

We conduct a comparatvie study of computational cost among the proposed protocol and the related schemes [18–21]. According to [40], we define that the real identity, hash function, timestamp, ECC operation, Chebyshev polynomial, and elements in group G_1 and G_2 are 64 bits, 160 bits, 32 bits, 320 bits, 256 bits, 1024 bits, and 1024 bits, respectively. The result of communicational cost comparison is shown in Table 8. The result shows that the proposed protocol has a lower communicational overhead than the related schemes [18–21].

Table 8. Comparison of communcational cost.

Schemes	Phase	Total Communication Costs	Messages
Chen et al. [18]	U2U	1920 bits	2
Alzahrani et al. [19]	U2U	1824 bits	3
Pham et al. [20]	U2I	2560 bits	3
	U2U	3040 bits	3
Hajian et al. [21]	U2U	1344 bits	2
Proposed	U2I	1216 bits	2
	U2U	1056 bits	2

6.4. Security Features

Table 9 presents the security and functionality features of the proposed protocol and the related schemes [18–21]. We define (SEC1: Replay attacks), (SEC2: Man-in-the-middle attacks), (SEC3: Impersonation attacks), (SEC4: Privileged insider attacks), (SEC5: ESL attacks), (SEC6: Verification table leakage attacks), (SEC7: Anonymity), (SEC8: Perfect forward secrecy), and (SEC9: Mutual authentication), respectively. Therefore, the proposed protocol can provide a robust secure and lightweight communication for edge computing-based AR environments.

Table 9. Security and functionality features comparison.

Security Features	[18]	[19]	[20]	[21]	Proposed
SEC1	○	○	×	○	○
SEC2	○	○	×	○	○
SEC3	×	○	×	○	○
SEC4	×	×	○	○	○
SEC5	×	○	○	○	○
SEC6	○	○	○	×	○
SEC7	○	×	○	○	○
SEC8	○	○	○	○	○
SEC9	○	○	○	○	○

○: "Provide the security and functionality features"; ×: "Does not provide the security and functionality features".

7. Conclusions

In this paper, we proposed an authentication protocol for edge computing-based AR environments. The proposed protocol consists of U2I and U2U authentication processes considering the workload of edge nodes. Moreover, the proposed protocol can provide secure and lightweight authentication services using Chebyshev chaotic maps and PUFs. We performed various security analyses, such as the Scyther tool, BAN logic, and informal security analysis. Moreover, we conducted comparative performance analysis using MIRACL, proving the computational and communicational lightweightness against the related schemes. Thus, the proposed protocol is suitable for edge computing AR environments, and users can share AR data using a robust secure communication channel. In future work, we will implement a practical AR environment and design an improved scheme to make secure and seamless AR services.

Author Contributions: Conceptualization, D.K.; methodology, D.K.; software, D.K.; validation, D.K. and Y.P.; formal analysis, D.K.; writing—original draft preparation, D.K.; writing—review and editing, Y.P.; supervision, Y.P.; project administration, Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Education under grant 2020R1I1A3058605.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ren, P.; Qiao, X.; Huang, Y.; Liu, L.; Dustdar, S.; Chen, J. Edge-assisted distributed DNN collaborative computing approach for mobile web augmented reality in 5G networks. *IEEE Netw.* **2020**, *34*, 254–261. [CrossRef]
2. Gsaxner, C.; Li, J.; Pepe, A.; Jin, Y.; Kleesiek, J.; Schmalstieg, D.; Egger, J. The HoloLens in medicine: A systematic review and taxonomy. *Med. Image Anal.* **2023**, *85*, 102757–102785. [CrossRef] [PubMed]
3. Chen, M.; Liu, W.; Wang, T.; Liu, A.; Zeng, Z. Edge intelligence computing for mobile augmented reality with deep reinforcement learning approach. *Comput. Netw.* **2021**, *195*, 108186. [CrossRef]
4. Salman, S.M.; Sitompul, T.A.; Papadopoulos, A.V.; Nolte, T. Fog computing for augmented reality: Trends, challenges and opportunities. In Proceedings of the 2020 IEEE International Conference on Fog Computing (ICFC), Sydney, NSW, Australia, 21–24 April 2020; pp. 56–63.

5. Hossain, M.D.; Huynh, L.N.; Sultana, T.; Nguyen, T.D.; Park, J.H.; Hong, C.S.; Huh, E.N. Collaborative task offloading for overloaded mobile edge computing in small-cell networks. In Proceedings of the 2020 International Conference on Information Networking (ICOIN), Barcelona, Spain, 7–10 January 2020; pp. 717–722.
6. Apicharttrisorn, K.; Chen, J.; Sekar, V.; Rowe, A.; Krishnamurthy, S.V. Breaking edge shackles: Infrastructure-free collaborative mobile augmented reality. In Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems, Boston, MA, USA, 6–9 November 2022; pp. 1–15.
7. Ren, P.; Qiao, X.; Huang, Y.; Liu, L.; Pu, C.; Dustdar, S.; Chen, J. Edge ar x5: An edge-assisted multi-user collaborative framework for mobile web augmented reality in 5g and beyond. *IEEE Trans. Cloud Comput.* **2020**, *10*, 2521–2537. [CrossRef]
8. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst. (TOCS)* **1990**, *8*, 18–36. [CrossRef]
9. Abdalla, M.; Fouque, P.; Pointcheval, D. Password-based authenticated key exchange in the three-party setting. In *Public Key Cryptography—PKC 2005, Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, 23–26 January 2005*; Lecture Notes in Computer Science (LNCS); Springer: Berlin/Heidelberg, Germany, 2005; pp. 65–84.
10. Cremers, C.J. The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols: Tool Paper. In Proceedings of the International Conference on Computer Aided Verification, Princeton, NJ, USA, 7–14 July 2008; pp. 414–418.
11. Scyther Tool. Available online: https://people.cispa.io/cas.cremers/scyther/ (accessed on 27 December 2023).
12. MIRACL Cryptographic SDK. Available online: https://github.com/miracl/MIRACL (accessed on 27 December 2023).
13. Ren, J.; He, Y.; Huang, G.; Yu, G.; Cai, Y.; Zhang, Z. An edge-computing based architecture for mobile augmented reality. *IEEE Netw.* **2019**, *33*, 162–169. [CrossRef]
14. Siriwardhana, Y.; Porambage, P.; Liyanage, M.; Ylianttila, M. A survey on mobile augmented reality with 5G mobile edge computing: Architectures, applications, and technical aspects. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1160–1192. [CrossRef]
15. Morín, D.G.; Pérez, P.; Armada, A.G. Toward the distributed implementation of immersive augmented reality architectures on 5G networks. *IEEE Commun. Mag.* **2022**, *60*, 46–52. [CrossRef]
16. Dang, T.N.; Kim, K.; Khan, L.U.; Kazmi, S.A.; Han, Z.; Hong, C.S. On-device computational caching-enabled augmented reality for 5G and beyond: A contract-theory-based incentive mechanism. *IEEE Internet Things J.* **2021**, *8*, 17382–17394. [CrossRef]
17. Park, G.S.; Kim, R.; Song, H. Collaborative virtual 3D object modeling for mobile augmented reality streaming services over 5G networks. *IEEE Trans. Mob. Comput.* **2022**, *22*, 3855–3869. [CrossRef]
18. Chen, Y.; Martínez, J.F.; Castillejo, P.; López, L. A bilinear map pairing based authentication scheme for smart grid communications: Pauth. *IEEE Access* **2019**, *7*, 22633–22643. [CrossRef]
19. Alzahrani, B.A.; Chaudhry, S.A.; Barnawi, A.; Al-Barakati, A.; Shon, T. An anonymous device to device authentication protocol using ECC and self certified public keys usable in Internet of Things based autonomous devices. *Electronics* **2020**, *9*, 520. [CrossRef]
20. Pham, C.D.; Dang, T.K. A lightweight authentication protocol for D2D-enabled IoT systems with privacy. *Pervasive Mob. Comput.* **2021**, *74*, 101399. [CrossRef]
21. Hajian, R.; Haghighat, A.; Erfani, S.H. A secure anonymous D2D mutual authentication and key agreement protocol for IoT. *Internet Things* **2022**, *18*, 100493. [CrossRef]
22. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
23. Canetti, R.; Krawczyk, H. Universally composable notions of key exchange and secure channels. In *Advances in Cryptology—EUROCRYPT 2002, Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, 28 April–2 May 2002*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 337–351.
24. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In *Advances in Cryptology—CRYPTO'99, Proceedings of the 19th Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 1999*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 388–397.
25. Son, S.; Lee, J.; Park, Y.; Park, Y.; Das, A.K. Design of blockchain-based lightweight V2I handover authentication protocol for VANET. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 1346–1358. [CrossRef]
26. Oh, J.; Kim, M.; Park, Y.; Park, Y. A Secure Content Trading for Cross-Platform in the Metaverse with Blockchain and Searchable Encryption. *IEEE Access* **2023**, *11*, 120680–120693. [CrossRef]
27. Park, Y.; Ryu, D.; Kwon, D.; Park, Y. Provably secure mutual authentication and key agreement scheme using PUF in internet of drones deployments. *Sensors* **2023**, *23*, 2034. [CrossRef]
28. Kwon, D.; Son, S.; Park, Y.; Kim, H.; Park, Y.; Lee, S.; Jeon, Y. Design of secure handover authentication scheme for urban air mobility environments. *IEEE Access* **2022**, *10*, 42529–42541. [CrossRef]
29. Hsieh, Y.P.; Lee, K.C.; Lee, T.F.; Su, G.J. Extended chaotic-map-based user authentication and key agreement for HIPAA privacy/security regulations. *Appl. Sci.* **2022**, *12*, 5701. [CrossRef]
30. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Advances in Cryptology-EUROCRYPT 2004, Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 523–540.
31. Son, S.; Kwon, D.; Lee, S.; Jeon, Y.; Das, A.K.; Park, Y. Design of Secure and Lightweight Authentication Scheme for UAV-Enabled Intelligent Transportation Systems using Blockchain and PUF. *IEEE Access* **2023**, *11*, 60240–60253. [CrossRef]
32. Cho, Y.; Oh, J.; Kwon, D.; Son, S.; Lee, J.; Park, Y. A secure and anonymous user authentication scheme for IoT-enabled smart home environments using PUF. *IEEE Access* **2022**, *10*, 101330–101346. [CrossRef]

33. Kwon, D.K.; Yu, S.J.; Lee, J.Y.; Son, S.H.; Park, Y.H. WSN-SLAP: Secure and lightweight mutual authentication protocol for wireless sensor networks. *Sensors* **2021**, *21*, 936. [CrossRef]
34. Majumder, S.; Ray, S.; Sadhukhan, D.; Dasgupta, M.; Das, A.K.; Park, Y. ECC-EXONUM-eVOTING: A Novel Signature-Based e-Voting Scheme Using Blockchain and Zero Knowledge Property. *IEEE Open J. Commun. Soc.* **2023**, *5*, 583–598. . [CrossRef]
35. Son, S.; Oh, J.; Kwon, D.; Kim, M.; Park, K.; Park, Y. A Privacy-Preserving Authentication Scheme for a Blockchain-Based Energy Trading System. *Mathematics* **2023**, *11*, 4653. [CrossRef]
36. Park, K.; Lee, J.; Das, A.K.; Park, Y. BPPS: Blockchain-enabled privacy-preserving scheme for demand-response management in smart grid environments. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 1719–1729. [CrossRef]
37. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf's law in passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791. [CrossRef]
38. Boyko, V.; MacKenzie, P.; Patel, S. Provably secure password-authenticated key exchange using Diffie-Hellman. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Bruges, Belgium, 14–18 May 2000; pp. 156–171.
39. Abbasinezhad-Mood, D.; Ostad-Sharif, A.; Mazinani, S.M.; Nikooghadam, M. Provably secure escrow-less Chebyshev chaotic map-based key agreement protocol for vehicle to grid connections with privacy protection. *IEEE Trans. Ind. Inform.* **2020**, *16*, 7287–7294. [CrossRef]
40. Zhang, L.; Zhu, Y.; Ren, W.; Wang, Y.; Choo, K.K.R.; Xiong, N.N. An energy-efficient authentication scheme based on Chebyshev chaotic map for smart grid environments. *IEEE Internet Things J.* **2021**, *8*, 17120–17130. [CrossRef]