

## Article

# LLAKEP: A Low-Latency Authentication and Key Exchange Protocol for Energy Internet of Things in the Metaverse Era

Xin Zhang <sup>1</sup>, Xin Huang <sup>1,\*</sup>, Haotian Yin <sup>1</sup>, Jiajia Huang <sup>1</sup>, Sheng Chai <sup>1</sup>, Bin Xing <sup>1</sup>, Xiaohua Wu <sup>2</sup> and Liangbin Zhao <sup>2</sup>

<sup>1</sup> College of Data Science, Taiyuan University of Technology, Taiyuan 030024, China; zhangxin072318@163.com (X.Z.); 18875241164@163.com (H.Y.); hhhuangjiajia455@163.com (J.H.); hzltbs@163.com (S.C.); xingbin1436@link.tyut.edu.cn (B.X.)  
<sup>2</sup> China Tower, Taiyuan 030001, China; wuxh@chinatelecom.cn (X.W.); zhaolb@chinatelecom.cn (L.Z.)  
\* Correspondence: xin@huangstudio.org

**Abstract:** The authenticated key exchange (AKE) protocol can ensure secure communication between a client and a server in the electricity transaction of the Energy Internet of things (EIoT). Park proposed a two-factor authentication protocol 2PAKEP, whose computational burden of authentication is evenly shared by both sides. However, the computing capability of the client device is weaker than that of the server. Therefore, based on 2PAKEP, we propose an authentication protocol that transfers computational tasks from the client to the server. The client has fewer computing tasks in this protocol than the server, and the overall latency will be greatly reduced. Furthermore, the security of the proposed protocol is analyzed by using the ROR model and GNY logic. We verify the low-latency advantage of the proposed protocol through various comparative experiments and use it for EIoT electricity transaction systems in a Metaverse scenario.

**Keywords:** EIoT; low-latency; AKE; security analysis; ROR; GNY logic; metaverse

**MSC:** 68M12



**Citation:** Zhang, X.; Huang, X.; Yin, H.; Huang, J.; Chai, S.; Xing, B.; Wu, X.; Zhao, L. LLAKEP: A Low-Latency Authentication and Key Exchange Protocol for Energy Internet of Things in the Metaverse Era. *Mathematics* **2022**, *10*, 2545. <https://doi.org/10.3390/math10142545>

Academic Editor: Raúl M. Falcón

Received: 21 June 2022

Accepted: 20 July 2022

Published: 21 July 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Authentication schemes in the traditional Energy Internet of Things (EIoT) are generally implemented with the help of Public Key Infrastructure (PKI). To simplify the management of public-key certificates, Shamir [1] introduced the identity-based cryptography scheme (IBC). This scheme directly uses the identity to generate the public key, without certificates or public key directories.

ID-based single-factor authentication scheme is not secure [2]. Attackers can compromise this scheme by dictionary attacks [3], rainbow tables [4], or social engineering techniques [5].

Thus, researchers have proposed two-factor authentication (2FA) [6,7], which combines representative data (ID/password) with personal possession factors (i.e., smart cards or mobile phones) to provide stronger security protection.

When ID-based 2FA authentication scheme is applied in EIoT, latency becomes an issue that has not been well studied [8,9]. Especially when electricity transactions based on EIoT are realized in Metaverse in the near future, high latency will affect the validity of data information (i.e., payment information data) [10]. Therefore, compared with the traditional payment systems, the EIoT payment systems in the Metaverse should meet higher requirements regarding the latency [11].

At present, the Metaverse devices are typically virtual helmets and smart glasses. A two-factor authentication protocol using smart cards (security chips that are embedded in these devices) can enhance the security of the protocol. In related work, we found that 2PAKEP is more secure than previous protocols [12–16]. In order to satisfy the requirement

of low latency in EIoT (or future EIoT in Metaverse), we propose a low-latency ID-based two-factor authentication protocol LLAKEP. Our main contributions are summarized below.

- A low-latency ID-based two-factor authentication protocol LLAKEP has been proposed. In the case of unbalanced computing capability between the two parties of the protocol, LLAKEP reduces the computational burden on one side. Compared with 2PAKEP [17], experimental results show that LLAKEP requires less computation time and less running time;
- The security of LLAKEP is analyzed by using the ROR (Real-or-Random) model and GNY (Gong–Needham–Yahalom) logic. Analysis results show that LLAKEP achieves the security goals of an AKE protocol;
- A use case has been implemented. We applied LLAKEP to EIoT electricity transaction systems in a Metaverse scenario. Results show that LLAKEP will effectively reduce latency.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 introduces the solution methodology. Section 4 introduces the preliminaries. Section 5 proposes the LLAKEP. In Section 6, the security of the LLAKEP is analyzed. The experiment results of LLAKEP are shown in Section 7. Finally, a conclusion is summarized in Section 8.

## 2. Related Work

Das [12] designed an ID-based authentication protocol (the D protocol) using bilinear pairings. However, the D protocol is subject to forgery attacks [18]. Many improved protocols have been proposed based on D protocol [13–17]. Table 1 lists the characteristics, limitations, and disadvantages of different protocols.

**Table 1.** Comparison of the characteristics, limitations, and disadvantages of different protocols.

Protocol	Characteristics	Limitations and Disadvantages
D protocol [12]	Based on pairing and smart card	Not resistant to forgery attacks
YC protocol [13]	Based on identity	Prone to simulated attacks Cannot provide perfect forward security
YY protocol [14]	An improved ID-based mobile device key authentication scheme based on elliptic curves	Cannot provide perfect forward security
HDB protocol [15]	A key agreement remote mutual authentication protocol based on identity	Unable to resist impersonation attacks and unknown key sharing attacks Not resistant to impersonated user attacks, password changes, insider attacks, and offline password guessing attacks
QC protocol [16]	Based on elliptic curves in mobile environments	Not efficient
2PAKEP [17]	Two-factor authentication, based on identity	
LLAKEP	A low-latency ID-based two-factor authentication protocol	\

Because of the inefficiency of bilinear pairing cryptography, researchers have proposed many ID-based authentication protocols using scalar multiplication. Yang and Chang [13] proposed an authentication protocol based on ID (the YC protocol) in 2009. However, Yoon and Yoo [14] found that the YC protocol [13] is prone to simulated attacks. In addition, the YC protocol cannot provide perfect forward security. Therefore, an improved ID-based

protocol (the YY protocol) is proposed by Yoon and Yoo. The YY protocol can eliminate the defects of the YC protocol [13]. However, the YY protocol cannot provide perfect forward security. In 2012, He [15] proposed a protocol (the HDB protocol). The HDB protocol can guarantee perfect forward security. However, in 2013, Chou [19] showed that the HDB protocol [15] has defects concerning the private key verification process, and legitimate users cannot confirm whether the private key of the other party is correct. Thus, two improved security protocols (the C1 protocol, and the C2 protocol) were proposed. In 2015, Yang [20] proved that the HDB protocol [15] cannot resist simulation attacks and unknown key sharing attacks, and then Yang proposed an improved ID-based authentication key exchange protocol (the Y protocol).

However, there are some defects in the above-mentioned ID-based authenticated protocols. Their protocols have issues concerning clock synchronization and user anonymity [16]. To solve the issues, Qi and Chen [16] proposed an ID-based two-factor mutual authentication protocol with smart cards (the QC protocol). Qi and Chen claim the QC protocol is resistant to many attacks. However, in 2018, Park [17] proved that the QC protocol is not resistant to simulated user attacks, password change attacks, insider attacks, and offline password guessing attacks. Thus, Park [17] proposed an improved protocol 2PAKEP and proved that it could solve these security issues. LLAKEP uses an improved algorithm to reduce the latency of 2PAKEP. In addition, LLAKEP uses a security chip.

At present, smart cards are widely used in medical, educational, and other scenarios [21–23]. Using smart cards as an authentication factor can improve the security of system authentication. The most widely used smart cards in payment systems are mainly microprocessor chips. In addition, the Trustzone [24,25] is included in the microprocessor chip, which provides security features for smart wearable devices [26].

### 3. Solution Methodology

#### 3.1. Research Methods

We research the low latency algorithms based on 2PAKEP. Meanwhile, we use security analysis and performance analysis to verify the advantages of LLAKEP.

#### 3.2. Security Analysis Methods

First, we prove the security of LLAKEP in the ROR model. Second, we use GNY logic to prove the security of LLAKEP. Finally, we verify the security of the protocol using Prolog.

#### 3.3. Performance Analysis Methods

We use a Raspberry Pi and a laptop to simulate two communication parties. The protocol is implemented in Python. The running time and computation time of LLAKEP and other protocols are compared by experiments.

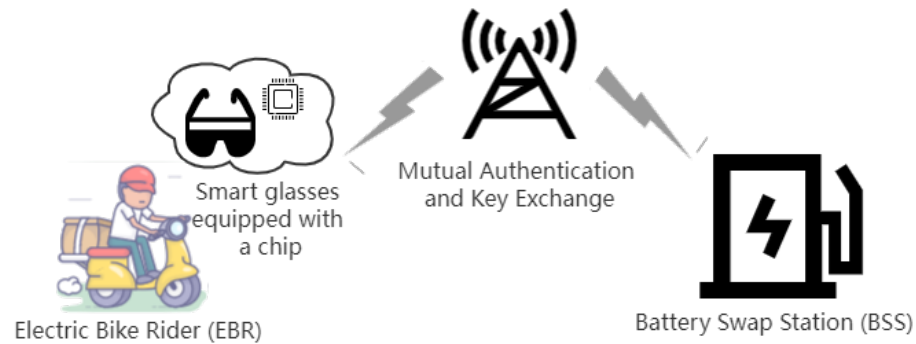
### 4. Preliminaries

The system model, ROR model, and computational assumptions are introduced in this section.

#### 4.1. System Model

In the EIoT, LLAKEP can be used to secure the key agreement for the communication of electricity transactions. A specific example is shown in Figure 1, where the electric bike rider is ready to swap his battery, and their device (smart glasses) and the battery swap station will establish a secure link through LLAKEP. The communication of transaction information, such as battery types and payment information, can then be encrypted through the session key. One thing to note is that the smart glasses in the example are the user's Metaverse interface, which implies that a "gap" in computing capabilities exists between the two ends of these common communication devices. More specifically, the smart glasses with a microprocessor have weaker computing capabilities than the battery swap station.

Before the electric bike rider uses the smart glasses to enter the Metaverse for electricity transactions, some user information needs to be stored in the memory of the smart glasses in the initial stage. Assuming that the electric bike rider has obtained a registered *microprocessor chip*, and has a *password*, and the microprocessor chip is equipped in the user's smart glass, then, as an initiator, the smart glasses authenticate with an energy device.



**Figure 1.** A typical architecture of EIoT.

#### 4.2. ROR Model

Abdalla, Fouque, and Pointcheval initially proposed the ROR model for password-based key exchange [27]. One of its significant features is that the attacker no longer has a Reveal query compared with the BPR model [28], but instead performs a simulation of a compromise caused by the misuse of a session key via the uniform Test query. This Test query can be called multiple times. Furthermore, the ROR model has been proved to be stronger than the BPR security model [27].

We introduce the primary components associated with the ROR model below.

**Participants and instances.** Let oracles  $\Pi_{EBR}^t$  and  $\Pi_{BSS}^s$  be the instances  $t$  and  $s$  of participants  $EBR$  and  $BSS$  running protocol  $\Pi$ , respectively.

**Instance state.**  $\Pi_{EBR}^t$  will be in the *accepted* state if it has received the final message according the protocol  $\Pi$ . The session identification  $sid$  of  $\Pi_{EBR}^t$  is the concatenation of exchanged messages in the session.

**Partnering.** We say that  $\Pi_{EBR}^t$  and  $\Pi_{BSS}^s$  are the partners if the following two conditions are satisfied: (1) both  $\Pi_{EBR}^t$  and  $\Pi_{BSS}^s$  are in the *accepted* state, (2)  $\Pi_{EBR}^t$  and  $\Pi_{BSS}^s$  have the same  $sid$  and mutually authenticated each other.

**Freshness.** If the session key  $SK$  of  $\Pi_{EBR}^t$  and  $\Pi_{BSS}^s$  is not compromised by a reveal query or  $EMD/EMC$  query defined below, we say  $\Pi_{EBR}^t$  and  $\Pi_{BSS}^s$  are fresh.

**Adversary.** An active adversary  $\mathcal{A}$  may intercept, delete, modify, or inject the messages over public channels by the given queries:

- $Execute(\Pi_{EBR}^t, \Pi_{BSS}^s)$  : This query models the eavesdropping attack that permits  $\mathcal{A}$  to learn the messages exchanged between  $EBR$  and  $BSS$ .
- $Send(\Pi_{EBR}^t, Msg)$  : This query models the active attack that permits  $\mathcal{A}$  to transmit a message  $Msg$  to a participant's instance  $\Pi_{EBR}^t$ .
- $EMD/EMC(\Pi_{EBR}^t)$  : This query models another active attack that permits  $\mathcal{A}$  to extract all the sensitive secret parameters stored in a mobile device ( $EMD(\Pi_{EBR}^t)$ ) or microprocessor chip ( $EMC(\Pi_{EBR}^t)$ ).
- $Test(\Pi_{EBR}^t)$  : Before the game starts, an unbiased coin  $b$  is flipped. If  $\Pi_{EBR}^t$  is fresh, this query returns the real session key  $SK$  if  $b = 1$ , or a random key in the key space of  $\Pi$  if  $b = 0$ ; otherwise, if  $\Pi_{EBR}^t$  is not fresh, this query returns the invalid symbol  $\perp$ .

We restrict  $\mathcal{A}$  to access a limited number of  $EMD/EMC(\Pi_{EBR}^t)$  queries in a formal security analysis. At the same time,  $\mathcal{A}$  is permitted to access an infinite number of  $Test(\Pi_{EBR}^t)$  queries.

**Semantic security.** Let  $\mathcal{A}$ 's guess be  $b'$ , and  $Succ$  be the winning probability in the game. A polynomial  $t$  time adversary  $\mathcal{A}$ 's advantage in breaking the semantic security of session key  $SK$  is denoted by

$$Adv_{SK}(t) = |2\Pr[Succ] - 1|.$$

**Random oracle.** We model the public one-way cryptographic hash function  $h(\cdot)$  as a random oracle (*Hash*).

#### 4.3. Computational Assumption

We use elliptic curve cryptography because it is one of the best candidates among the existing public key cryptographic techniques. Two relevant hardness assumptions are described below.

**Definition 1** (Elliptic curve discrete logarithm problem (ECDLP)). *Given an elliptic curve  $E$  over finite field  $F_p$ , and  $P, Q \in E$ , find the discrete logarithm  $d$ , such that  $Q = dP$ .*

**Definition 2** (Elliptic curve decisional Diffie–Hellman problem (ECDDHP)). *Given an elliptic curve  $E$  over finite field  $F_p$ , a generator  $P$  of  $E$ , and three random elements  $k_1P, k_2P$ , and  $k_3P$ , distinguish the triples  $(k_1P, k_2P, k_3P)$  and  $(k_1P, k_2P, k_1k_2P)$ .*

The ECDLP and ECDDHP are computationally hard problems when  $p$  is large.

### 5. The Low-Latency Protocol

In this section, we mainly introduce the process of LLAKEP. The symbols used in LLAKEP are shown in Table 2.

**Table 2.** Symbols used in LLAKEP.

Symbol	Meaning
$EBR$	Electric bike riders
$MC$	Microprocessor chip
$BSS$	Battery swap station
$\mathcal{A}$	Adversary
$ID_{EBR}$	Identity of an electric bike rider $EBR$
$PW_{EBR}$	Password of an electric bike rider $EBR$
$sk_X$	Private key of $X$
$pk_X$	Public key of $X$
$SK$	Session key
$E/F_p$	An elliptic curve $E$ over a prime finite field $F_p$ with $p$ being a large prime
$n$	Order of base point $P$
$Z_n^*$	$\{1, 2, \dots, n-1\}$
$kP$	Scalar multiplication on elliptic curves and $P$ is a base point in $E/F_p$
$A  B$	Concatenation operation between $A$ and $B$
$A \oplus B$	XOR operation between $A$ and $B$
$kdf(Msg)$	Derivate key from $Msg$
$H(Msg)$	A one-way hash function that generates $Msg$ digests
$X \dashrightarrow Y : Msg$	$X$ sends message $Msg$ to $Y$ by using a secure channel, where $X$ and $Y$ are two entities.
$X \rightarrow Y : Msg$	$X$ sends message $Msg$ to $Y$ by using a public channel

#### 5.1. Initialization Phase

This phase is performed in the battery swap station  $BSS$ . The specific process is described as follows.

**BSS-1:** BSS selects an elliptic curve  $E/F_p$  whose base point is  $P$ . Meanwhile, the order of  $p$  is set to  $n$ .

**BSS-2:** BSS generates a private key  $sk_{BSS}$  from  $Z_n^*$ , and calculates the public key  $pk_{BSS}$  by  $pk_{BSS} = sk_{BSS}P$ .

**BSS-3:** BSS chooses two hash functions (collision-resistant)  $H_1(\cdot)$  and  $H_2(\cdot)$ . At the end, BSS publishes the system parameters  $\{E/F_p, P, n, pk_{BSS}, H_1(\cdot), H_2(\cdot)\}$ .

### 5.2. User Registration Phase

Electric bike rider  $EBR$  needs to register with battery swap station  $BSS$  before swapping batteries. The registration takes place in a secure channel, and the specific process (Table 3) is described as follows.

**EBR-1:**  $EBR$  inputs the  $ID_{EBR}$  and  $PW_{EBR}$  on the smart glasses. After the input is completed, the microprocessor chip  $MC$  generates two random numbers  $a_{MC}, b_{MC}$  and calculates  $HIP = H_2(ID_{EBR}||PW_{EBR}), v = HIP \oplus a_{MC}, d = HIP \oplus b_{MC}$  and  $C = H_2(ID_{EBR}||PW_{EBR}||a_{MC})$ . Finally,  $EBR$  submits:

$$Msg1 = \{pk_{EBR}, ID_{EBR}, d\}$$

to the  $BSS$  by using a secure channel.

**BSS-2:**  $BSS$  checks whether  $H_2(ID_{EBR})$  and  $ID_{EBR}$  are valid after receiving  $Msg1$ . If they already exist in the database,  $BSS$  returns a message to  $EBR$  asking for a new  $ID$ .

**BSS-3:**  $BSS$  calculates  $l = H_1(sk_{BSS}) \oplus d \oplus H_2(sk_{BSS}||ID_{EBR})$ . After that,  $BSS$  stores  $H_2(sk_{BSS}||ID_{EBR}), ID_{EBR}$  and sends  $Msg2 = \{l\}$  to  $EBR$  by using a secure channel.

**EBR-4:** After receiving  $Msg2$ ,  $EBR$  calculates  $l' = l \oplus b_{MC} = H_1(sk_{BSS}) \oplus HIP \oplus H_2(sk_{BSS}||ID_{EBR})$  and stores  $l', v$ , and  $C$  in the microprocessor chip.

Table 3. User registration phase.

Electric Bike Riders/Microprocessor Chip (EBR/MC)	Battery Swap Station (BSS)
EBR inputs $ID_{EBR}$ and $PW_{EBR}$ MC generates $a_{MC}$ and $b_{MC}$ MC computes $HIP, v, d, C$ $Msg1 = \{pk_{EBR}, ID_{EBR}, d\}$ -->	Checks whether $H_2(ID_{EBR})$ and $ID_{EBR}$ are valid Calculates $l = H_1(sk_{BSS}) \oplus d \oplus H_2(sk_{BSS}  ID_{EBR})$ and stores $H_2(sk_{BSS}  ID_{EBR}), ID_{EBR}$ $Msg2 = \{l\}$ ←--
Calculates $l' = l \oplus b_{MC}$ Stores $l', v$ and $C$ secretly	

### 5.3. Authentication and Key Exchange (AKE) Phase

After registration, when electric bike rider  $EBR$  wants to swap batteries, he needs to send some information for identity authentication. The key algorithms of this phase are shown in Algorithms 1 and 2.  $ECC\_ScalarMul$  denotes scalar multiplication on an elliptic curve, and its computation is time-consuming.  $ECC\_Add$  represents addition on an elliptic curves, and  $ECC\_Neg$  represents negation operations on an elliptic curves. These two cryptographic operations take less time.  $kdf$  represents the key derivation function. We transferred a scalar multiplication on the  $EBR$  side in the original protocol algorithm to the  $BSS$  side. The specific process (Table 4) of the AKE phase is described as follows.



---

**Algorithm 1** EBR calculates  $SK = kdf(ID_{EBR}, SK_{EBR}, T_{MC}, T_{BSS})$

---

**Input:**  $E, r_{MC}, pk_{BSS}, R_{BSS}, sk_{EBR}, T_{MC}, T_{BSS}$

**Output:** The session key  $SK$

- 1:  $U_{EBR} = ECC\_Add(r_{MC}, sk_{EBR}, E)$
  - 2:  $R = ECC\_ScalarMul(r_{MC}, pk_{BSS}, E)$
  - 3:  $SK_{EBR} = ECC\_ScalarMul(r_{MC}, R_{BSS}, E)$
  - 4:  $SK = kdf(ID_{EBR}, SK_{EBR}, T_{MC}, T_{BSS})$
- 

---

**Algorithm 2** BSS calculates  $SK = kdf(ID_{EBR}, SK_{BSS}, T_{MC}, T_{BSS})$

---

**Input:**  $E, U_{EBR}, pk_{EBR}, sk_{BSS}, r_{BSS}, ID_{EBR}, T_{MC}, T_{BSS}$

**Output:** The temporary secret  $R$

- 1:  $temp1 = ECC\_Neg(pk_{EBR}, E)$
  - 2:  $temp2 = ECC\_ScalarMul(U_{EBR}, P, E)$
  - 3:  $R_{EBR} = ECC\_Add(temp1, temp2, E)$
  - 4:  $R = ECC\_ScalarMul(R_{EBR}, sk_{BSS}, E)$
  - 5:  $R_{BSS} = ECC\_ScalarMul(r_{BSS}, P, E)$
  - 6:  $SK_{BSS} = ECC\_ScalarMul(r_{BSS}, R_{EBR}, E)$
  - 7:  $SK = kdf(ID_{EBR}, SK_{BSS}, T_{MC}, T_{BSS})$
- 

**EBR-1:** EBR inputs  $ID_{EBR}$  and  $PW_{EBR}$  using a smart glasses. Then MC calculates  $HIP = H_2(ID_{EBR} || PW_{EBR})$ ,  $a_{MC} = v \oplus HIP$  and  $C'_{EBR} = H_2(ID_{EBR} || PW_{EBR} || a_{MC})$ . After that, EBR checks whether  $C'_{EBR}$  is equal to  $C$ . After successful verification, MC generates a random number  $r_{MC} \in Z_n^*$  and a current timestamp  $T_{MC}$ , and computes  $U_{EBR} = r_{MC} + sk_{EBR}$ ,  $R = r_{MC}pk_{BSS}$ ,  $CID_{EBR} = l' \oplus HIP = H_1(sk_{BSS}) \oplus H_2(sk_{BSS} || ID_{EBR})$  and  $Auth_{EBR} = H_2(ID_{EBR} || R || CID_{EBR} || T_{MC})$ . Then, EBR sends:

$$Msg1 = \{Auth_{EBR}, CID_{EBR}, U_{EBR}, T_{MC}\}$$

to the BSS by using a public channel.

**BSS-2:** BSS verifies whether the difference between  $T_{MC}$  and the reception time  $T_{MC}^*$  is less than the maximum transmission latency  $\Delta T$  after receiving  $Msg1$ . If it is greater than  $\Delta T$ , the protocol will stop running. Otherwise, BSS calculates  $H_2(sk_{BSS} || ID_{EBR}) = CID_{EBR} \oplus H_1(sk_{BSS})$ . After that, BSS computes  $R_{EBR} = U_{EBR}P - pk_{EBR} = r_{MC}P$  and  $R^* = sk_{BSS}R_{EBR}$ ,  $Auth_{EBR}^* = H_2(ID_{EBR} || R^* || CID_{EBR} || T_{MC})$  and checks whether  $Auth_{EBR}^*$  is equal to  $Auth_{EBR}$ . After successful verification, BSS generates a random number  $r_{BSS} \in Z_n^*$  and a current timestamp  $T_{BSS}$ . Then BSS computes  $R_{BSS} = r_{BSS}P$ ,  $SK_{BSS} = r_{BSS}R_{EBR}$  and  $Auth_{BSS} = H_2(ID_{EBR} || R^* || SK_{BSS} || T_{BSS})$ . At the end, BSS sends:

$$Msg2 = \{Auth_{BSS}, R_{BSS}, T_{BSS}\}$$

to EBR by using a public channel.

**EBR-3:** After receiving  $Msg2$ , EBR first verifies whether the difference between  $T_{BSS}$  and the reception time  $T_{BSS}^*$  is less than  $\Delta T$ . If it is greater than  $\Delta T$ , the protocol will stop running. Otherwise, EBR calculates  $SK_{EBR} = r_{MC}R_{BSS}$ ,  $Auth_{BSS}^* = H_2(ID_{EBR} || R || SK_{EBR} || T_{BSS})$ , and checks whether  $Auth_{BSS}^*$  is equal to  $Auth_{BSS}$ . After successful verification, MC generates the current timestamp  $T'_{MC}$ , and computes the session key  $SK = kdf(ID_{EBR} || SK_{EBR} || T_{MC} || T_{BSS})$ . At the end, EBR calculates  $Auth_{EB} = H_2(ID_{EBR} || R || SK || T'_{MC})$ , and EBR sends:

$$Msg3 = \{Auth_{EB}, T'_{MC}\}$$

to the BSS through a public channel.

**BSS-4:** After receiving  $Msg3$ , BSS verifies whether the difference between  $T'_{MC}$  and the reception time  $T''_{MC}$  is less than  $\Delta T$ . If it is greater than  $\Delta T$ , the protocol will stop running. Otherwise BSS computes the session key  $SK' = kdf(ID_{EBR} || SK_{BSS} || T_{MC} || T_{BSS})$ ,

$Auth_{EB}^* = H_2(ID_{EBR} || R^* || SK' || T_{MC}')$  and checks whether  $Auth_{EB}^*$  is equal to  $Auth_{EB}$ . If they are equal, the mutual authentication and session key agreement phase have successfully be completed. Finally, the same session key  $SK(= SK')$  will be store, and it will be used for secure commucations of EBR and BSS.

**Table 4.** Mutual authentication and key exchange phase.

Electric Bike Riders/Microprocessor Chip (EBR/MC)	Battery Swap Station (BSS)
EBR inputs identity $ID_{EBR}$ and password $PW_{EBR}$ MC calculates $HIP$ , $a_{MC}$ and $C'_{EBR}$ MC Checks whether $C? = C'_{EBR}$ Generates $r_{MC} \in Z_n^*$ and $T_{MC}$ Computes $U_{EBR} = r_{MC} + sk_{EBR}$ , $R = r_{MC}pk_{BSS}$ , $CID_{EBR} = l' \oplus HIP$ and $Auth_{EBR} = H_2(ID_{EBR}    R    CID_{EBR}    T_{MC})$ $Msg1 = \{Auth_{EBR}, CID_{EBR}, U_{EBR}, T_{MC}\}$	Validates the received timestamp $T_{MC}$ Computes $H_2(sk_{BSS}    ID_{EBR}) = CID_{EBR} \oplus H_1(sk_{BSS})$ Computes $R_{EBR} = U_{EBR}P - pk_{EBR} = r_{MC}P$ and $R^* = sk_{BSS}R_{EBR}$ , $Auth_{EBR}^* = H_2(ID_{EBR}    R^*    CID_{EBR}    T_{MC})$ Checks whether $Auth_{EBR}^* = Auth_{EBR}$ Generates $r_{BSS} \in Z_n^*$ and $T_{BSS}$ Computes $R_{BSS} = r_{BSS}P$ , $SK_{BSS} = r_{BSS}R_{EBR}$ $Auth_{BSS} = H_2(ID_{EBR}    R^*    SK_{BSS}    T_{BSS})$ $Msg2 = \{Auth_{BSS}, R_{BSS}, T_{BSS}\}$
Verifies the received timestamp $T_{BSS}$ Calculates $SK_{EBR} = r_{MC}R_{BSS}$ $Auth_{BSS}^* = H_2(ID_{EBR}    R    SK_{EBR}    T_{BSS})$ Checks whether $Auth_{BSS}^* = Auth_{BSS}$ Generates $T'_{MC}$ and computes $SK = kdf(ID_{EBR}    SK_{EBR}    T_{MC}    T_{BSS})$ , $Auth_{EB} = H_2(ID_{EBR}    R    SK    T'_{MC})$ $Msg3 = \{Auth_{EB}, T'_{MC}\}$	Validates the timestamp $T'_{MC}$ Calculates the session key $SK' = kdf(ID_{EBR}    SK_{BSS}    T_{MC}    T_{BSS})$ $Auth_{EB}^* = H_2(ID_{EBR}    R^*    SK'    T'_{MC})$ Checks whether $Auth_{EB}^* = Auth_{EB}$

#### 5.4. Password Change

Electric bike riders can change their password at any time. The specific process (Table 5) is described as follows.

**EBR-1:** EBR first inputs  $ID_{EBR}$  and old password  $PW_{EBR}$  through a microprocessor chip.

**MC-2:** MC computes  $HIP = H_2(ID_{EBR} || PW_{EBR})$ ,  $a_{MC} = v \oplus HIP$ . After that, MC calculates  $C' = H_2(ID_{EBR} || PW_{EBR} || a_{MC})$ , and then verifies C is equal to C' or not. If it is satisfied, MC asks EBR to input a new password.

**MC-3:** After receiving the new password, MC calculate  $HIP_{new} = H_2(ID_{EBR} || PW_{new})$ ,  $v_{new} = HIP_{new} \oplus a_{MC}$ ,  $d_{new} = HIP_{new} \oplus b_{MC}$ ,  $C_{new} = H_2(ID_{EBR} || PW_{new} || a_{MC})$  and  $l_{new} = l' \oplus HIP \oplus HIP_{new} = H_1(sk_{BSS}) \oplus HIP_{new} \oplus H_2(sk_{BSS} || ID_{EBR})$ . Finally, EBR store  $l_{new}$ ,  $v_{new}$  and  $C_{new}$  in the microprocessor chip and delete old parameters.



**Table 5.** Password change activity.

Electric Bike Riders (EBR)	Microprocessor Chip (MC)
EBR inputs $ID_{EBR}$ and $PW_{EBR}$	MC Computes $HIP = H_2(ID_{EBR}  PW_{EBR})$ , $a_{MC} = v \oplus HIP$ and $C' = H_2(ID_{EBR}  PW_{EBR}  a_{MC})$ Checks if $C = C'$
Chooses a new password $PW_{new}$	Asks EBR to input a new password  Calculate $HIP_{new} = H_2(ID_{EBR}  PW_{new})$ , $v_{new} = HIP_{new} \oplus a_{MC}$ , $C_{new} = H_2(ID_{EBR}  PW_{new}  a_{MC})$ and $l_{new} = l' \oplus HIP \oplus HIP_{new} =$ $H_1(sk_{BSS}) \oplus HIP_{new} \oplus H_2(sk_{BSS}  ID_{EBR})$ .
Stores $l_{new}$ , $v_{new}$ and $C_{new}$ , deletes old parameters	

### 5.5. Comparison of LLAKEP and Other Protocols

From the experimental results of He et al.'s scheme [15], it can be obtained that the most time spent is on the elliptic curve scalar multiplication operation, followed by the execution of a map-to-point hash function and a modular inversion operation, while the time spent on the execution of a hash operation, a dissimilarity operation, a message authentication code operation, and a key derivation function is very short. The main cryptographic operations involved in the authentication phase of the relevant protocols and LLAKEP are shown in Table 6. *Client* denotes the device with limited computing power, and *Server* denotes the device with strong computing power.

We can see that the total number of elliptic curve scalar multiplication required by LLAKEP is fewer than that of the protocols proposed in [13,14], so the total computing time of LLAKEP is less than theirs. Compared to the protocols proposed in [15–17], *Client* of LLAKEP needs to perform fewer elliptic curve scalar multiplications, which leads to the computing time being cut, thus reducing the overall latency.

**Table 6.** Comparison of computation costs.

Protocol	Client	Server
YC protocol [13]	$4\mathcal{M} + 3\mathcal{H} + \mathcal{P}$	$4\mathcal{M} + 3\mathcal{H} + \mathcal{P}$
YY protocol [14]	$4\mathcal{M} + 3\mathcal{H} + \mathcal{P}$	$4\mathcal{M} + 4\mathcal{H} + \mathcal{P}$
HDB protocol [15]	$3\mathcal{M} + 2\mathcal{H} + 2\mathcal{C}$	$3\mathcal{M} + 3\mathcal{H} + \mathcal{C} + \mathcal{I}$
QC protocol [16]	$3\mathcal{M} + 4\mathcal{H} + 3\mathcal{X} + \mathcal{K}$	$3\mathcal{M} + 4\mathcal{H} + \mathcal{X} + \mathcal{K}$
2PAKEP [17]	$3\mathcal{M} + 6\mathcal{H} + 2\mathcal{X} + \mathcal{K}$	$3\mathcal{M} + 4\mathcal{H} + 3\mathcal{X} + \mathcal{K}$
LLAKEP	$2\mathcal{M} + 6\mathcal{H} + 2\mathcal{X} + \mathcal{K}$	$4\mathcal{M} + 4\mathcal{H} + 3\mathcal{X} + \mathcal{K}$

Note:  $\mathcal{M}$ : the time for an elliptic curve point scalar multiplication operation;  $\mathcal{H}$ : the time for a hash operation;  $\mathcal{P}$ : the time for a map-to-point hash operation;  $\mathcal{X}$ : the time for a XOR operation;  $\mathcal{C}$ : the time for a message authentication code operation;  $\mathcal{I}$ : the time for executing a modular inversion operation;  $\mathcal{K}$ : the time for a key derivation function.

## 6. Security Analysis

This section proves the security of LLAKEP in the ROR model.

### 6.1. Security Proof

The security of LLAKEP in the ROR model is shown in Theorem 1.

**Theorem 1.** Let  $Adv_{LLAKEP}(t)$  be the advantage of a polynomial-time  $t$  adversary  $\mathcal{A}$  in breaking the security of LLAKEP, then

$$Adv_{LLAKEP}(t) \leq \frac{q_h^2}{|Hash|} + 2 \left( \frac{q_s}{|D|} + Adv_A^{ECDDHP}(t) \right),$$

where  $|Hash|, q_s, q_h, |D|$  and  $Adv_A^{ECDDHP}(t)$  are the number of Hash queries, the number of Send queries, the number of Hash queries, the size of password dictionary  $D$  in LLAKEP, and the advantage of  $\mathcal{A}$  in breaking the ECDDHP in time  $t$ , respectively.

**Proof.** Let  $G_j$ , where  $j = 0, 1, 2, 3, 4$ , be a sequence of games, and  $Succ_{G_j}$  be the event that an adversary  $\mathcal{A}$  wins the game  $G_j$ , the probability of which is denoted by  $\Pr[Succ_{G_j}]$ . Those five games are defined as follows:

- $G_0$  : This game models the original protocol LLAKEP in the ROR model, and an unbiased coin  $b$  is flipped. Therefore,

$$Adv_{LLAKEP}(t) = |2\Pr[Succ_{G_0}] - 1|. \quad (1)$$

- $G_1$  : This game excludes the eavesdropping attacks.  $\mathcal{A}$  may use the *Execute* query in this game, and once the instance is accepted,  $\mathcal{A}$  proceeds to the *Test* query. In LLAKEP,  $SK$  and  $SK'$  are calculated as  $SK = kdf(ID_{EBR} || SK_{EBR} || T_{MC} || T_{BSS}) = kdf(ID_{EBR} || SK_{BSS} || T_{MC} || T_{BSS}) (= SK')$ , where  $SK_{EBR} = r_{MC}R_{BSS} = r_{MC}(r_{BSS}P) = r_{BSS}(r_{MC}P) = SK_{BSS}$ . For getting the session key,  $\mathcal{A}$  needs ephemeral secrets  $\{r_{MC}, r_{BSS}\}$  and the permanent secret identity  $ID_{EBR}$ . Hence,  $\mathcal{A}$  has no advantage in winning the game  $G_1$  through eavesdropping attack. Therefore,

$$\Pr[Succ_{G_1}] = \Pr[Succ_{G_0}]. \quad (2)$$

- $G_2$  : This game models the *Send* and *Hash* queries.  $\mathcal{A}$  may mount an active attack to intercept messages  $Msg1 = \{Auth_{EBR}, CID_{EBR}, U_{EBR}, T_{MC}\}$ ,  $Msg2 = \{Auth_{BSS}, R_{BSS}, T_{BSS}\}$ , and  $Msg3 = \{Auth_{EB}, T'_{MC}\}$ . Note that all these messages involve the random nonces and the current timestamps, the only advantage  $\mathcal{A}$  can take is making the *Hash* queries to find collisions. Therefore, by the birthday paradox,

$$|\Pr[Succ_{G_2}] - \Pr[Succ_{G_1}]| \leq \frac{q_h^2}{2|Hash|}. \quad (3)$$

- $G_3$  : This game models the *EMD/EMC* query wherein  $\mathcal{A}$  can extract all the credentials  $l', v$  and  $C$  from a lost or stolen device or a microprocessor chip, where  $l' = l \oplus b_{MC} = H_1(sk_{BSS}) \oplus HIP \oplus H_2(sk_{BSS} || ID_{EBR}), v = HIP \oplus a_{MC}$  and  $C = H_2(ID_{EBR} || PW_{EBR} || a_{MC})$ . Note that since  $\mathcal{A}$  could not get the secret credentials  $a_{MC}$  and  $sk_{BSS}$  using the *Send* queries, guessing is the only way to obtain the password  $PW_{EBR}$  and identity  $ID_{EBR}$  of a registered user  $EBR$  from  $l', v$ , and  $C$ . Therefore,

$$|\Pr[Succ_{G_3}] - \Pr[Succ_{G_2}]| \leq \frac{q_s}{|D|}. \quad (4)$$

- $G_4$ : This game models an active attack. To derive the session key  $SK$  of  $EBR$  and  $BSS$  ( $SK = kdf(ID_{EBR} || SK_{EBR} || T_{MC} || T_{BSS}) = kdf(ID_{EBR} || SK_{BSS} || T_{MC} || T_{BSS} = SK')$ ),  $\mathcal{A}$  may use *Send* queries to obtain all the intercepted messages  $Msg1, Msg2$ , and  $Msg3$ , and then try to derive  $SK_{EBR} = r_{MC}R_{BSS} = r_{MC}(r_{BSS}P) = r_{BSS}(r_{MC}P) = SK_{BSS}$ . Note that  $\mathcal{A}$  can derive  $SK_{EBR} = r_{MC}R_{BSS}$  or  $SK_{BSS} = r_{BSS}(U_{EBR}P - pk_{EBR})$ . However, this problem is essentially the same as solving an ECDDHP. Therefore,

$$\begin{aligned} & |\Pr[Succ_{G_4}] - \Pr[Succ_{G_3}]| \\ & \leq Adv_A^{ECDDHP}(t). \end{aligned} \quad (5)$$

After executing the games,  $\mathcal{A}$  guesses the bit  $b$ :

$$\Pr[\text{Succ}_{G_4}] = \frac{1}{2}. \quad (6)$$

According to (1) and (2), we have:

$$\begin{aligned} \frac{1}{2} \text{Adv}_{\text{LLAKEP}}(t) &= |\Pr[\text{Succ}_{G_0}] - \frac{1}{2}| \\ &= |\Pr[\text{Succ}_{G_1}] - \frac{1}{2}|. \end{aligned} \quad (7)$$

According to (6) and (7), we have:

$$\frac{1}{2} \text{Adv}_{\text{LLAKEP}}(t) = |\Pr[\text{Succ}_{G_1}] - \Pr[\text{Succ}_{G_4}]|. \quad (8)$$

Using the triangular inequality, we have the following result:

$$\begin{aligned} &|\Pr[\text{Succ}_{G_1}] - \Pr[\text{Succ}_{G_4}]| \\ &\leq |\Pr[\text{Succ}_{G_1}] - \Pr[\text{Succ}_{G_3}]| \\ &\quad + |\Pr[\text{Succ}_{G_3}] - \Pr[\text{Succ}_{G_4}]| \\ &\leq |\Pr[\text{Succ}_{G_1}] - \Pr[\text{Succ}_{G_2}]| \\ &\quad + |\Pr[\text{Succ}_{G_2}] - \Pr[\text{Succ}_{G_3}]| \\ &\quad + |\Pr[\text{Succ}_{G_3}] - \Pr[\text{Succ}_{G_4}]| \\ &\leq \frac{q_h^2}{2|\text{Hash}|} + \leq \frac{q_s}{|D|} \\ &\quad + \text{Adv}_{\mathcal{A}}^{\text{ECDDHP}}(t). \end{aligned} \quad (9)$$

From (8) and (9), we have:

$$\begin{aligned} \frac{1}{2} \text{Adv}_{\text{LLAKEP}}(t) &\leq \frac{q_h^2}{2|\text{Hash}|} + \leq \frac{q_s}{|D|} \\ &\quad + \text{Adv}_{\mathcal{A}}^{\text{ECDDHP}}(t). \end{aligned} \quad (10)$$

Then, we obtain the required result:

$$\text{Adv}_{\text{LLAKEP}}(t) \leq \frac{q_h^2}{|\text{Hash}|} + 2 \left( \frac{q_s}{|D|} + \text{Adv}_{\mathcal{A}}^{\text{ECDDHP}}(t) \right).$$

Theorem 1 is proved.  $\square$

## 6.2. GNY Logic Proof

We introduce the symbols and meanings used in the GNY logic [29] in Table 7, and then prove the mutual authentication between electric bike rider  $EBR$  and battery swap station  $BSS$  in LLAKEP.

### 6.2.1. Protocol Paraphrase

LLAKEP consists of the following messages between  $EBR$  and  $BSS$ .

1.  $EBR \rightarrow BSS : \text{Auth}_{EBR}, \text{CID}_{EBR}, \text{U}_{EBR}, T_{MC}$
2.  $BSS \rightarrow EBR : \text{Auth}_{BSS}, R_{BSS}, T_{BSS}$
3.  $EBR \rightarrow BSS : \text{Auth}_{EB}, T'_{MC}$

### 6.2.2. Description of Protocol

The parser algorithm would describe the protocol as follows.

$$\begin{aligned}
Msg_1 &: BSS \triangleleft *Auth_{EBR}, *CID_{EBR}, *U_{EBR}, *T_{MC} \\
Msg_2 &: EBR \triangleleft *Auth_{BSS}, *R_{BSS}, *T_{BSS} \\
Msg_3 &: BSS \triangleleft *Auth_{EB}, *T'_{MC}
\end{aligned}$$

### 6.2.3. Goal

We need to show that LLAKEP achieves the following goals.

- Goal 1** :  $EBR \models \#SK$   
**Goal 2** :  $EBR \models \phi SK$   
**Goal 3** :  $EBR \models BSS \ni SK$   
**Goal 4** :  $BSS \models \#SK$   
**Goal 5** :  $BSS \models \phi SK$   
**Goal 6** :  $BSS \models EBR \ni SK$

**Table 7.** GNY Expression.

Symbol	Meaning
$(A, B)$	Conjunction of $A$ and $B$ .
$H(A)$	A one-way hash function of $A$ .
$*A$	$A$ is a not-originated-here formula.
$P \triangleleft A$	$P$ is told $A$ .
$P \ni A$	$A$ possesses, or is capable of possessing $A$ .
$P \sim A$	$P$ once said $A$ .
$P \models \#(A)$	$P$ believes that $A$ is fresh, that is, $A$ has not been used before
$P \models \mathcal{O}(A)$	$P$ can recognize $A$ , that is, $P$ has certain expectations for the content of $A$ .
$P \models P \xleftrightarrow{Key} Q$	$P$ believes that $Key$ is a suitable secret for $P$ and $Q$ .

### 6.2.4. Initialization Assumption

The initialization assumptions for  $EBR$  and  $BSS$  are as follows.

- A1** :  $EBR \models \#r_{MC}$   
**A2** :  $EBR \models \phi r_{MC}$   
**A3** :  $EBR \ni r_{MC}, pk_{BSS}, ID_{EBR}, sk_{BSS}, T_{BSS}, P$   
**A4** :  $EBR \models EBR \xleftrightarrow{ID_{EBR}} BSS$   
**A5** :  $BSS \models \#r_{BSS}, sk_{BSS}$   
**A6** :  $BSS \models \phi r_{BSS}$   
**A7** :  $BSS \ni r_{BSS}, pk_{EBR}, ID_{EBR}$   
**A8** :  $BSS \models BSS \xleftrightarrow{ID_{EBR}} EBR$

### 6.2.5. Proof

The proof of the goals are as follows.

According to rules T1 and P1, we can infer that  $EBR$  possesses  $Auth_{BSS}, R_{BSS}, T_{BSS}$ , and  $BSS$  possesses  $Auth_{EBR}, CID_{EBR}, U_{EBR}, T_{MC}, Auth_{EB}, T'_{MC}$ .

$$\begin{aligned}
& \frac{EBR \triangleleft *Auth_{BSS}, *R_{BSS}, *T_{BSS}}{EBR \triangleleft Auth_{BSS}, R_{BSS}, T_{BSS}}(T1) \\
& \frac{EBR \triangleleft Auth_{BSS}, R_{BSS}, T_{BSS}}{EBR \ni Auth_{BSS}, R_{BSS}, T_{BSS}}(P1) \\
& \frac{BSS \triangleleft *Auth_{EBR}, *CID_{EBR}, *U_{EBR}, *T_{MC}, *Auth_{EB}, *T'_{MC}}{BSS \triangleleft Auth_{EBR}, CID_{EBR}, U_{EBR}, T_{MC}, Auth_{EB}, T'_{MC}}(T1) \\
& \frac{BSS \triangleleft Auth_{EBR}, CID_{EBR}, U_{EBR}, T_{MC}, Auth_{EB}, T'_{MC}}{BSS \ni Auth_{EBR}, CID_{EBR}, U_{EBR}, T_{MC}, Auth_{EB}, T'_{MC}}(P1)
\end{aligned}$$

**Goal 1** According to A1 and the rule F1, we can infer that  $EBR$  believes that  $SK_{EBR}$  is fresh, and  $SK_{EBR} = R_{BSS} * r_{MC}$ .

$$\frac{EBR \models \sharp r_{MC}}{EBR \models \sharp R_{BSS} * r_{MC}}(F1)$$

According to the rule F1, we can infer that  $EBR$  believes that  $SK$  is fresh, and  $SK = kdf(ID_{EBR} || SK_{EBR} || T_{MC} || T_{BSS})$ . Goal 1 is proved.

$$\frac{EBR \models \sharp SK_{EBR}}{EBR \models \sharp (ID_{EBR} || SK_{EBR} || T_{MC} || T_{BSS})}(F1)$$

**Goal 2** According to A2 and the rule R1, we can infer that  $EBR$  believes that  $SK_{EBR}$  is recognizable, and  $SK_{EBR} = R_{BSS} * r_{MC}$ .

$$\frac{EBR \models \phi r_{MC}}{EBR \models \phi R_{BSS} * r_{MC}}(R1)$$

According to the rule R1, we can infer that  $EBR$  believes that  $SK$  is recognizable, and  $SK = kdf(ID_{EBR} || SK_{EBR} || T_{MC} || T_{BSS})$ . Goal 2 is proved.

$$\frac{EBR \models \phi SK_{EBR}}{EBR \models \phi (ID_{EBR} || SK_{EBR} || T_{MC} || T_{BSS})}(R1)$$

**Goal 3** According to the rule P2, we can infer that  $EBR$  possesses  $SK_{EBR}$ , and  $SK_{EBR} = R_{BSS} * r_{MC}$ .

$$\frac{EBR \ni r_{MC}, EBR \ni R_{BSS}}{EBR \ni R_{BSS} * r_{MC}}(P2)$$

According to A3 and the rule P2, we can infer that  $EBR$  possesses  $R$ , and  $R = r_{MC} * pk_{BSS}$ .

$$\frac{EBR \ni r_{MC}, EBR \ni pk_{BSS}}{EBR \ni r_{MC} * pk_{BSS}}(P2)$$

According to A3 and the rule P2, we can infer that  $EBR$  possesses  $(ID_{EBR} || R || SK_{EBR} || T_{BSS})$ .

$$\frac{EBR \ni ID_{EBR}, EBR \ni R, EBR \ni SK_{EBR}, EBR \ni T_{BSS}}{EBR \ni (ID_{EBR} || R || SK_{EBR} || T_{BSS})}(P2)$$

According to the rule F1, we can infer that  $EBR$  believes that  $R$  is fresh, and  $R = r_{MC} * pk_{BSS}$ .

$$\frac{EBR \models \sharp r_{MC}}{EBR \models \sharp r_{MC} * pk_{BSS}}(F1)$$

According to the rule F1, we can infer that  $EBR$  believes that  $(ID_{EBR} || R || SK_{EBR} || T_{BSS})$  is fresh.

$$\frac{EBR \models \sharp R}{EBR \models \sharp (ID_{EBR} || R || SK_{EBR} || T_{BSS})}(F1)$$

According to A4 and the rule I3, we can infer that  $EBR$  believes that  $BSS$  once said  $SK_{EBR}$ .

$$\frac{EBR \triangleleft * H_2(ID_{EBR} || R || SK_{EBR} || T_{BSS}), EBR \ni (ID_{EBR} || R || SK_{EBR} || T_{BSS}), EBR \models EBR \xrightarrow{ID_{EBR}} BSS, EBR \models \sharp (ID_{EBR} || R || SK_{EBR} || T_{BSS})}{EBR \models BSS \sim (ID_{EBR} || R || SK_{EBR} || T_{BSS})}(I3)$$

$$\frac{EBR \models BSS \sim (ID_{EBR} || R || SK_{EBR} || T_{BSS})}{EBR \models BSS \sim SK_{EBR}}(I7)$$

According to the rule I6, we can infer that  $EBR$  believes that  $BSS$  possesses  $SK_{EBR}$ .

$$\frac{EBR \models BSS \sim SK_{EBR}, EBR \models \#SK_{EBR}}{EBR \models BSS \ni SK_{EBR}}(I6)$$

According to the rule J6, we can infer that  $EBR$  believes that  $BSS$  possesses  $SK$ , and  $SK = kdf(ID_{EBR}||SK_{EBR}||T_{MC}||T_{BSS})$ . Goal 3 is proved.

$$\frac{EBR \models BSS \ni ID_{EBR}, EBR \models BSS \ni SK_{EBR}, EBR \models BSS \ni T_{MC}, EBR \models BSS \ni T_{BSS}}{EBR \models BSS \ni (ID_{EBR}||SK_{EBR}||T_{MC}||T_{BSS})}(J6)$$

$$\frac{EBR \models BSS \ni (ID_{EBR}||SK_{EBR}||T_{MC}||T_{BSS})}{EBR \models BSS \ni kdf(ID_{EBR}||SK_{EBR}||T_{MC}||T_{BSS})}(J6)$$

**Goal 4** According to A5 and the rule F1, we can infer that  $BSS$  believes that  $SK_{BSS}$  is fresh, and  $SK_{BSS} = R_{EBR} * r_{BSS}$ .

$$\frac{BSS \models \#r_{BSS}}{BSS \models \#R_{EBR} * r_{BSS}}(F1)$$

According to the rule F1, we can infer that  $BSS$  believes that  $SK$  is fresh, and  $SK = kdf(ID_{EBR}||SK_{BSS}||T_{MC}||T_{BSS})$ . Goal 4 is proved.

$$\frac{BSS \models \#SK_{BSS}}{BSS \models \#(ID_{EBR}||SK_{BSS}||T_{MC}||T_{BSS})}(F1)$$

**Goal 5** According to A6 and the rule R1, we can infer that  $BSS$  believes that  $SK_{BSS}$  is recognizable, and  $SK_{BSS} = R_{EBR} * r_{BSS}$ .

$$\frac{BSS \models \phi r_{BSS}}{BSS \models \#R_{EBR} * r_{BSS}}(R1)$$

According to the rule R1, we can infer that  $BSS$  believes that  $SK$  is recognizable, and  $SK = kdf(ID_{EBR}||SK_{BSS}||T_{MC}||T_{BSS})$ . Goal 5 is proved.

$$\frac{BSS \models \phi SK_{BSS}}{BSS \models \phi(ID_{EBR}||SK_{BSS}||T_{MC}||T_{BSS})}(R1)$$

**Goal 6** According to A7 and the rule P2, we can infer that  $BSS$  possesses  $R_{EBR}$  and  $R$ , and  $R_{EBR} = U_{EBR}P - pk_{EBR}$ ,  $R = sk_{BSS} * R_{EBR}$ .

$$\frac{BSS \ni P, BSS \ni U_{EBR}, BSS \ni pk_{EBR}}{BSS \ni (U_{EBR}P - pk_{EBR})}(P2)$$

$$\frac{BSS \ni sk_{BSS}, BSS \ni R_{EBR}}{BSS \ni (sk_{BSS} * R_{EBR})}(P2)$$

According to the rule P2, we can infer that  $BSS$  possesses  $SK_{BSS}$ , and  $SK_{BSS} = r_{BSS} * R_{EBR}$ .

$$\frac{BSS \ni r_{BSS}, BSS \ni R_{EBR}}{BSS \ni r_{BSS} * R_{EBR}}(F1)$$

According to A7 and the rule P2, we can infer that  $BSS$  possesses  $SK$ .

$$\frac{BSS \ni ID_{EBR}, BSS \ni SK_{BSS}, BSS \ni T_{MC}, BSS \ni T_{BSS}}{BSS \ni kdf(ID_{EBR}||SK_{BSS}||T_{MC}||T_{BSS})}(P2)$$



According to the rule P2, we can infer that  $BSS$  possesses  $(R||SK||T'_{MC})$ .

$$\frac{BSS \ni R, BSS \ni SK, BSS \ni T'_{MC}}{BSS \ni (R||SK||T'_{MC})} (P2)$$

According to the rule F1, we can infer that  $BSS$  believes that  $R$  is fresh, and  $R = sk_{BSS} * R_{EBR}$ .

$$\frac{BSS \models \#sk_{BSS}}{BSS \models \#sk_{BSS} * R_{EBR}} (F1)$$

According to the rule F1, we can infer that  $BSS$  believes that  $(R||SK||T'_{MC})$  is fresh.

$$\frac{BSS \models \#R}{BSS \models \#(R||SK||T'_{MC})} (F1)$$

According to A8 and the rule I3, we can infer that  $BSS$  believes that  $EBR$  once said  $SK$ .

$$\frac{BSS \triangleleft *H_2(ID_{EBR}||R||SK||T'_{MC}), BSS \ni (ID_{EBR}||R||SK||T'_{MC}), BSS \models EBR \xrightarrow{ID_{EBR}} BSS, BSS \models \#(ID_{EBR}||R||SK||T'_{MC})}{BSS \models EBR \sim (ID_{EBR}||R||SK||T'_{MC})} (I3)$$

$$\frac{BSS \models EBR \sim (ID_{EBR}||R||SK||T'_{MC})}{BSS \models EBR \sim SK} (I3)$$

According to the rule I6, we can infer that  $BSS$  believes that  $EBR$  possesses  $SK$ . Goal 6 is proved.

$$\frac{BSS \models EBR \sim SK, BSS \models \#SK}{BSS \models EBR \ni SK} (I6)$$

### 6.3. Formal Verification

We use Prolog to verify that our protocol achieves the session key security goals (the freshness and the recognizability of the session key, and the belief that the two authenticating parties have the session key). Prolog is a logic verification tool. Write the flow of the protocol as Prolog code, and Prolog can verify whether the protocol achieves our required security goals.

The execution results of Prolog are shown in Figure 2, and we can see that several security goals regarding the protocol returned “True”, which indicates that the LLAKEP can achieve the required security goals.

```
?- bel(ebr, fresh(idebr, rbss0*rmc, tmc, tbss)).
true .

?- bel(ebr, reco(idebr, rbss0*rmc, tmc, tbss)).
true .

?- bel(bss, fresh(idebr, rebr*rbss, tmc, tbss)).
true .

?- bel(bss, reco(idebr, rebr*rbss, tmc, tbss)).
true .

?- bel(ebr, poss(bss, kdf(idebr, rbss0*rmc, tmc, tbss))).
true .

?- bel(bss, poss(ebr, kdf(idebr, rbss*rebr, tmc, tbss))).
true .
```

Figure 2. Prolog verification results of the LLAKEP.

## 7. Performance Analysis

We mainly analyze the advantages of the LLAKEP and provide a use case of LLAKEP in this section. Furthermore, we test the computation time, the total running time and the bit rate of different protocols. The experimental environment is shown in Table 8. We use  $T_A^D$  to represent the time of running  $A$  on device  $D$ .

**Table 8.** Experiment devices and environments.

	Device	CPU	Core	RAM	Programming Language
Experiment I	Laptop	i5-8250U 1.8 GHz	4	16 GB	Python
	Laptop	i5-8250U 1.8 GHz	4	16 GB	Python
Experiment II/III/IV/V	Laptop	i5-8250U 1.8 GHz	4	16 GB	Python
	Raspberry Pi	1.2 GHz ARM	4	1 GB	Python

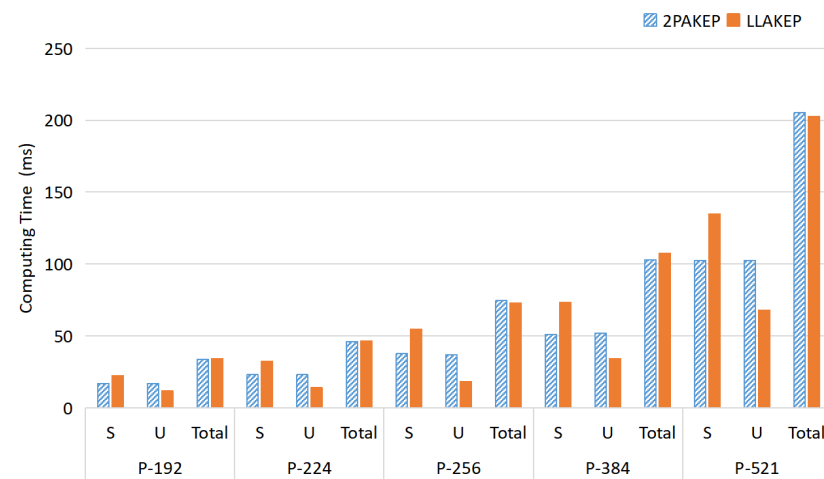
### 7.1. Experiment I

We use two identically configured laptops to represent the correspondents of the LLAKEP and test under the elliptic curves recommended by the National Institute of Standards and Technology Federal Information Processing Standard [30] (i.e., curves P-192, P-224, P-256, P-384, and P-521). From Figure 3, the following are some verified results:

For the average computing time on the EBR side:

$$T_{LLAKEP}^{EBR} < T_{2PAKEP}^{EBR}.$$

The results show that LLAKEP does reduce the computational burden on the EBR's side.



**Figure 3.** Average computing time in Experiment I. The experiment uses two identically configured laptops to represent two parties of the LLAKEP. The results are as follows: (1) the calculation time on the U side using LLAKEP is less than 2PAKEP; (2) LLAKEP does reduce the computational burden on the EBR's side.

### 7.2. Experiment II

We use a Raspberry Pi to represent the smart glasses and a laptop to represent the energy device. Smart glasses have less computing capability than laptops. We test LLAKEP under the same conditions as the elliptic curve of Experiment I. From Figure 4, the following are some verified results:

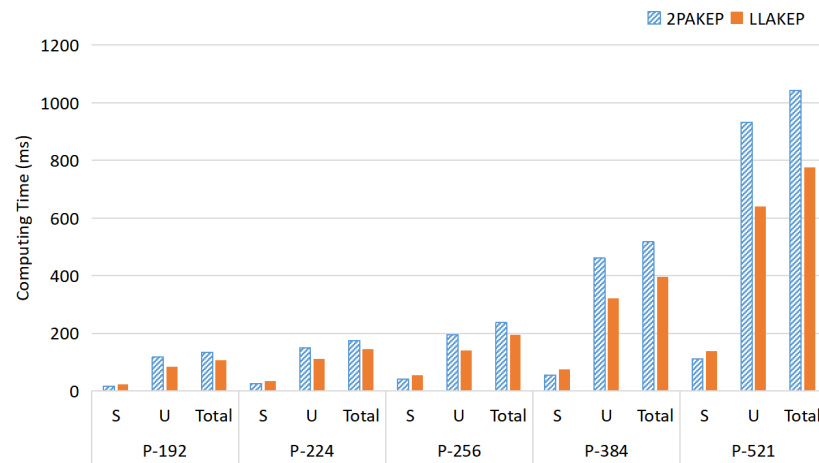
- For the average computing time on the EBR side:

$$T_{LLAKEP}^{EBR} < T_{2PAKEP}^{EBR}.$$

- For the average total computing time:

$$T_{LLAKEP} < T_{2PAKEP}.$$

It shows that the weaker device (i.e., smart glasses) in LLAKEP has shorter computation time. Further, LLAKEP has shorter total computation time compared with 2PAKEP.



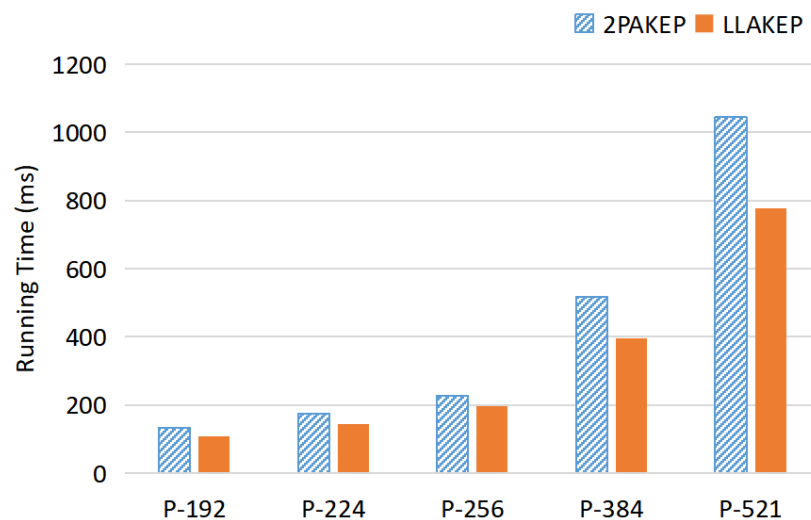
**Figure 4.** Average computing time in Experiment II. The experiment uses a Raspberry Pi to represent the smart glasses and a laptop to represent the energy device. The results are as follows: (1) the calculation time on the U side using LLAKEP is less than 2PAKEP; (2) the total calculation time of LLAKEP is less than 2PAKEP.

### 7.3. Experiment III

This experiment measures the total running time of LLAKEP on two communicating parties (a Raspberry Pi and a laptop). From Figure 5, the following are some verified results: For the average total time:

$$T_{LLAKEP} < T_{2PAKEP}.$$

The results show that LLAKEP still has shorter total running time compared with 2PAKEP.



**Figure 5.** Average total running time in Experiment III. The experiment uses a Raspberry Pi and a laptop to measure the total running time of LLAKEP. The results show that LLAKEP has shorter total running time compared with 2PAKEP.

#### 7.4. Experiment IV

We assume bits of different messages in Table 9.

**Table 9.** Bits of different messages.

Message	Number of Bits
Identity	160
Message digest	160
Nonce	160
Timestamp	160
Elliptic curve point	320

Therefore, in the authentication phase of the LLAKEP, *Msg1* needs  $(160 + 160 + 160 + 32) = 512$  bits, *Msg2* needs  $(160 + 320 + 32) = 512$  bits and *Msg3* needs  $(160 + 32) = 192$  bits. The total bits of LLAKEP is 1216 bits. Combining the total runtime of the protocol in Experiment III with the elliptic curve P-256, we can calculate the bit rate. The higher the bit rate, the faster the data transfer speed. The results are shown in Table 10.

For the bit rate *Br*:

$$Br_{LLAKEP} > Br_{2PAKEP}.$$

Therefore, the transmission latency of LLAKEP is lower.

**Table 10.** Bit rate comparison.

Protocol	Number of Bits	Bit Rate (Bit per Second)
2PAKEP [17]	1376	6048.8
LLAKEP	1216	6197.8

#### 7.5. Experiment V: Use Case Study

This section illustrates usages and advantages of LLAKEP via a use case in a battery swap cabinets scenario.

##### 7.5.1. Scenario Description

At present, there are more than 300 million electric bikes in China. In order to meet a large number of battery swap needs, China Tower has built an intelligent power exchange system. They have also deployed battery swap stations (Figure 6).



**Figure 6.** Battery swap cabinet.

In the future, with the development of the Metaverse, electric bike riders will use smart glasses to interact with battery swap cabinets. During the peak period, a large number of riders will need to authenticate and pay at the same time.

### 7.5.2. Application of LLAKEP

The following steps explain how we can use LLAKEP.

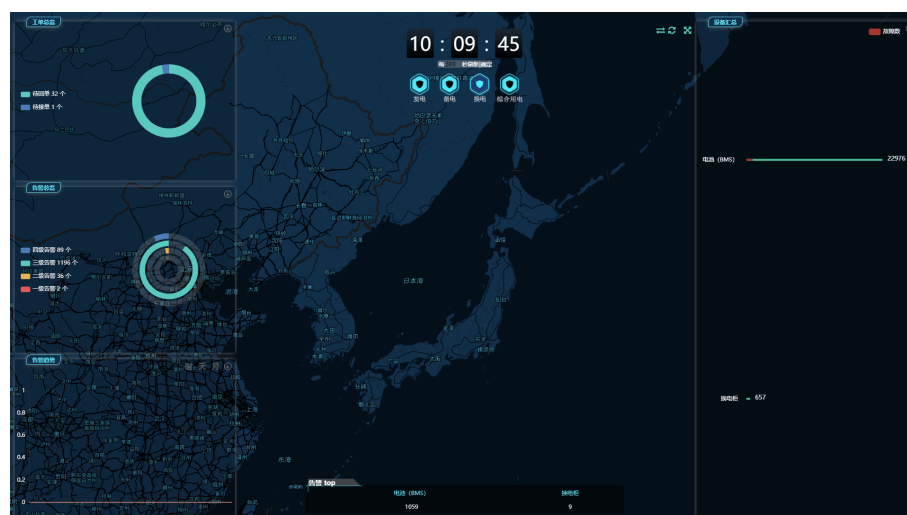
Initialization: devices *A* and *B* should support LLAKEP. Specifically, device *A* is smart glasses; device *B* is a battery swap cabinet.

Secure Handshake: suppose there are *N* smart glasses in the battery swap cabinet scenarios.

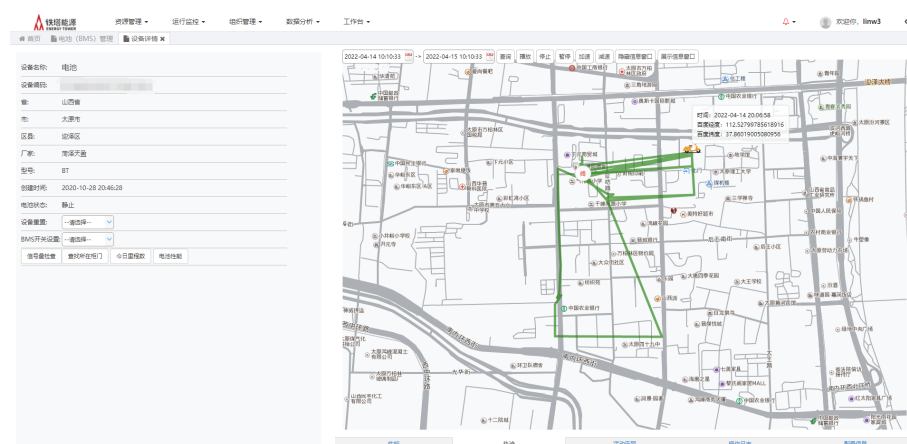
Secure Messaging: *A* and *B* use the generated session key to send the message (battery type and payment information) securely.

### 7.5.3. Advantages

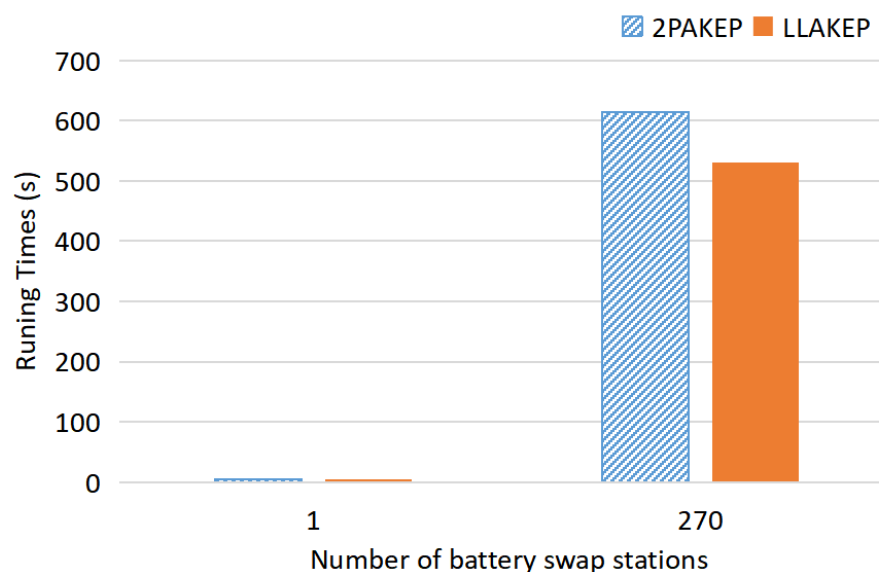
In this part, we analyze the advantages of LLAKEP. According to the statistics from the battery swap station management system (Figures 7 and 8), the number of battery swap stations in Taiyuan city is 270. One battery swap station has 10 battery swap cabinets. In the peak time, 2700 riders use smart glasses to authenticate. After successful authentication, the rider will pay for the swap of a battery. Taking P-256 as an example, Figure 9 shows the authentication protocol running time of battery swap stations in the peak time. Experiment results show that LLAKEP can reduce latency effectively.



**Figure 7.** Battery swap station management system. The number of battery swap stations in Taiyuan city can be obtained from this system.



**Figure 8.** Battery management system (BMS). The usage state of the battery can be obtained from this system.



**Figure 9.** Runing time in Experiment IV. The experiment tests the total running time of all the batteries of 270 battery swap stations in the authentication phase. The results show that the total running time of LLAKEP is significantly less than 2PAKEP.

## 8. Conclusions

This paper proposes a secure, low-latency authentication protocol LLAKEP for the EIoT. LLAKEP reduces the computational burden on weaker devices by changing the time-consuming cryptographic operations needed in the algorithms for both sides of communication. In addition, a provable security model and a logic analysis are used to analyze LLAKEP. Results show that the security of LLAKEP is guaranteed. When the computing capability of both parties is unbalanced, experimental results show that LLAKEP can reduce the computing time of the device with weaker computing capability. It can improve the efficiency of authentication. Finally in the use case, we apply LLAKEP for EIoT electricity transaction system in the Metaverse.

In the future, we will continue to optimize the low-latency algorithm, and design more low-latency AKE protocols suitable for Metaverse scenarios.

**Author Contributions:** Methodology, X.Z.; formal analysis, X.Z., H.Y. and J.H.; investigation, X.Z.; resources, S.C., B.X., X.W. and L.Z.; writing—original draft preparation, X.Z., X.H. and H.Y.; writing—review and editing, X.Z., X.H. and H.Y.; project administration, X.H.; funding acquisition, X.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Research Project Shanxi Scholarship Council of China 2021-038, and the Applied Basic Research Project of Shanxi Province No. 20210302123130.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Shamir, A. Identity-based cryptosystems and signature schemes. In *Workshop on the Theory and Application of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 1984; pp. 47–53.
- Ometov, A.; Bezzateev, S.; Mäkitalo, N.; Andreev, S.; Mikkonen, T.; Koucheryavy, Y. Multi-factor authentication: A survey. *Cryptography* **2018**, *2*, 1. [[CrossRef](#)]
- Wang, D.; Wang, P. Offline dictionary attack on password authentication schemes using smart cards. In *Information Security*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 221–237.



4. Ah Kioon, M.C.; Wang, Z.S.; Deb Das, S. Security analysis of MD5 algorithm in password storage. *Appl. Mech. Mater.* **2013**, *347*, 2706–2711. [\[CrossRef\]](#)
5. Heartfield, R.; Loukas, G. A taxonomy of attacks and a survey of defence mechanisms for semantic social engineering attacks. *ACM Comput. Surv. (CSUR)* **2015**, *48*, 1–39. [\[CrossRef\]](#)
6. Petsas, T.; Tsirantonakis, G.; Athanasopoulos, E.; Ioannidis, S. Two-factor authentication: Is the world ready? Quantifying 2FA adoption. In Proceedings of the Eighth European Workshop on System Security, Bordeaux, France, 21 April 2015; pp. 1–7.
7. Wang, D.; He, D.; Wang, P.; Chu, C.H. Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment. *IEEE Trans. Dependable Secur. Comput.* **2014**, *12*, 428–442. [\[CrossRef\]](#)
8. Jolfaei, A.; Kant, K. A lightweight integrity protection scheme for low latency smart grid applications. *Comput. Secur.* **2019**, *86*, 471–483. [\[CrossRef\]](#)
9. Mahmood, K.; Chaudhry, S.A.; Naqvi, H.; Kumari, S.; Li, X.; Sangaiah, A.K. An elliptic curve cryptography based lightweight authentication scheme for smart grid communication. *Future Gener. Comput. Syst.* **2018**, *81*, 557–565. [\[CrossRef\]](#)
10. Lee, L.H.; Braud, T.; Zhou, P.; Wang, L.; Xu, D.; Lin, Z.; Kumar, A.; Bermejo, C.; Hui, P. All one needs to know about metaverse: A complete survey on technological singularity, virtual ecosystem, and research agenda. *arXiv* **2021**, arXiv:2110.05352.
11. Ynag, Q.; Zhao, Y.; Huang, H.; Zheng, Z. Fusing Blockchain and AI with Metaverse: A Survey. *arXiv* **2022**, arXiv:2201.03201.
12. Das, M.L.; Saxena, A.; Gulati, V.P.; Phatak, D.B. A novel remote user authentication scheme using bilinear pairings. *Comput. Secur.* **2006**, *25*, 184–189. [\[CrossRef\]](#)
13. Yang, J.H.; Chang, C.C. An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. *Comput. Secur.* **2009**, *28*, 138–143. [\[CrossRef\]](#)
14. Yoon, E.J.; Yoo, K.Y. Robust id-based remote mutual authentication with key agreement scheme for mobile devices on ECC. In Proceedings of the 2009 International Conference on Computational Science and Engineering, Vancouver, BC, Canada, 29–31 August 2009; Volume 2, pp. 633–640.
15. Debiao, H.; Jianhua, C.; Jin, H. An ID-based client authentication with key agreement protocol for mobile client–Server environment on ECC with provable security. *Inf. Fusion* **2012**, *13*, 223–230. [\[CrossRef\]](#)
16. Qi, M.; Chen, J. An efficient two-party authentication key exchange protocol for mobile environment. *Int. J. Commun. Syst.* **2017**, *30*, e3341. [\[CrossRef\]](#)
17. Park, K.; Park, Y.; Park, Y.; Das, A.K. 2PAKEP: Provably secure and efficient two-party authenticated key exchange protocol for mobile environment. *IEEE Access* **2018**, *6*, 30225–30241. [\[CrossRef\]](#)
18. Goriparthi, T.; Das, M.L.; Negi, A.; Saxena, A. Cryptanalysis of recently proposed Remote User Authentication Schemes. *IACR Cryptol. ePrint Arch.* **2006**, *2006*, 28.
19. Chou, C.H.; Tsai, K.Y.; Lu, C.F. Two ID-based authenticated schemes with key agreement for mobile environments. *J. Supercomput.* **2013**, *66*, 973–988. [\[CrossRef\]](#)
20. Yang, H.; Chen, J.; Zhang, Y. An improved two-party authentication key exchange protocol for mobile environment. *Wirel. Pers. Commun.* **2015**, *85*, 1399–1409. [\[CrossRef\]](#)
21. Yang, W.; Wang, S.; Hu, J.; Zheng, G.; Chaudhry, J.; Adi, E.; Valli, C. Securing mobile healthcare data: A smart card based cancelable finger-vein bio-cryptosystem. *IEEE Access* **2018**, *6*, 36939–36947. [\[CrossRef\]](#)
22. Zheng, L.; Song, C.; Cao, N.; Li, Z.; Zhou, W.; Chen, J.; Meng, L. A new mutual authentication protocol in mobile RFID for smart campus. *IEEE Access* **2018**, *6*, 60996–61005. [\[CrossRef\]](#)
23. Shouqi, C.; Wanrong, L.; Liling, C.; Xin, H.; Zhiyong, J. An improved authentication protocol using smart cards for the Internet of Things. *IEEE Access* **2019**, *7*, 157284–157292. [\[CrossRef\]](#)
24. Zhang, Y.; Zhao, S.; Qin, Y.; Yang, B.; Feng, D. Trusttokenf: A generic security framework for mobile two-factor authentication using trustzone. In Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA, Helsinki, Finland, 20–22 August 2015; Volume 1, pp. 41–48.
25. Koutroumpouchos, N.; Ntantogian, C.; Xenakis, C. Building Trust for Smart Connected Devices: The Challenges and Pitfalls of TrustZone. *Sensors* **2021**, *21*, 520. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Brasser, F.; Kim, D.; Liebchen, C.; Ganapathy, V.; Iftode, L.; Sadeghi, A.R. Regulating arm trustzone devices in restricted spaces. In Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services, Singapore, 26–30 June 2016; pp. 413–425.
27. Abdalla, M.; Fouque, P.A.; Pointcheval, D. Password-based authenticated key exchange in the three-party setting. In Proceedings of the International Workshop on Public Key Cryptography, Les Diablerets, Switzerland, 23–26 January 2005; pp. 65–84.
28. Bellare, M.; Pointcheval, D.; Rogaway, P. Authenticated key exchange secure against dictionary attacks. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Bruges, Belgium, 14–18 May 2000; pp. 139–155.
29. Gong, L.; Needham, R.M.; Yahalom, R. Reasoning about Belief in Cryptographic Protocols. In Proceedings of the 1990 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 7–9 May 1990; pp. 234–248. [\[CrossRef\]](#)
30. Standard, S.H. National Institute of Standards and Technology (NIST) Federal Information Processing Standard (FIPS) 186-4. 2013. Available online: <https://csrc.nist.gov/publications/detail/fips/186/4/final> (accessed on 19 July 2013).