IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 29, 2021, accepted April 2, 2021, date of publication April 7, 2021, date of current version April 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3071499

Design of Secure Decentralized Car-Sharing System Using Blockchain

MYEONGHYUN KIM[1], JOONYOUNG LEE[1], KISUNG PARK[1,2], YOHAN PARK[3], KIL HOUM PARK[1,4], AND YOUNGHO PARK[1,4], (Member, IEEE)

[1]School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, South Korea
[2]Blockchain Technology Research Center, Electronics and Telecommunications Research Institute, Daejeon 34129, South Korea
[3]School of Computer Engineering, Keimyung University, Daegu 42601, South Korea
[4]School of Electronics Engineering, Kyungpook National University, Daegu 41566, South Korea

Corresponding author: Youngho Park (parkyh@knu.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1I1A3058605, and in part by the BK21 FOUR Project funded by the Ministry of Education, South Korea, under Grant 4199990113966.

ABSTRACT Car-sharing systems can solve various urban problems by providing shared vehicles to people and reducing the operation of personal vehicles. With the development of the Internet of Things, people can easily use a shared car through simple operations on their mobile devices. However, the car-sharing system has security problems. Sensitive information, such as the user's identity, location information, and access code, is transmitted through a public channel for car-sharing. Hence, an attacker can access this information for illegal purposes, making the establishment of a secure authentication protocol essential. Furthermore, the traditional car-sharing system is established on the centralized structure, so there is a single point of failure. Thus, the design of a decentralized car-sharing scheme is vital for solving the centralized problem. This study designed a decentralized car-sharing scheme using blockchain. Specifically, blockchain technology was used to provide a decentralization car-sharing service and ensure data integrity. The participant entities of the proposed system can be authenticated anonymously. The proposed car-sharing system can be secured against various attacks and provide mutual authentication using informal analysis, automated validation of internet security protocols and applications (AVISPA) simulation, and BAN logic analysis. The computation costs and communication costs of the proposed scheme were also analyzed.

INDEX TERMS Car-sharing system, blockchain, security, authentication.

I. INTRODUCTION

Car-sharing systems were introduced to help solve transportation problems in urban areas, such as traffic congestion on the road, pollution from fuel combustion [1], [2], and shortage of parking place from the increased number of vehicles. Car-sharing systems offer the benefits of private vehicle use without the costs and responsibilities of ownership to users and reduce private vehicle ownership. Rather than owning one or more vehicles, a household or business can access a fleet of shared vehicles on an as-required basis. With these advantages, car-sharing systems have proliferated. In 2019, the car-sharing market size exceeded USD 2.5 billion and is expected to surpass USD 9 billion by 2026 [3].

The associate editor coordinating the review of this manuscript and approving it for publication was SK Hafizul Islam.

The car-sharing system is classified as business models such as Business-to-Consumer (B2C) and Peer-to-Peer (P2P) car-sharing service models [4]. In the B2C service model, companies have deployed their shared cars that are rented out to users. Unlike B2C, the P2P service model is a system in which car owners convert their personal vehicles into shared cars and rent them to other users on a short-term basis [5]. In both car-sharing models, a service vendor assists the car owners and renters by acting as an intermediary and provides the resources needed to make the exchange possible, such as an online platform and customer support [6]. Under this system, users can book and lease a shared car on an online service platform using their smartphones.

The advent of car-sharing systems can alleviate transportation problems, but car-sharing systems have security

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

problems. The user operates their smartphone to lease a shared car by a simple operation on the online applications in the car-sharing system. However, because the information is transmitted through a public channel, a malicious attacker can easily eavesdrop, forge, delete, and modify the information. Unfortunately, if a digital key or code for accessing is exposed, the malicious attacker can control the shared car and steal it. Therefore, secure authentication must be guaranteed to provide a secure communication channel. Moreover, a user authentication step is essential to check that the user has the right and ability to drive a car. Users must submit their information (e.g., identity and driving license) to the service provider (e.g., sharing service company) when they request a car-sharing service. The service provider verifies that the customer has the right and ability to drive as a valid driver. After that, the user can utilize the car-sharing service through a service provider.

In a traditional car-sharing system, the user's information and service information can be stored and controlled at a centralized service server. However, a centralized server suffers from a single point of failure by a malicious attacker. For example, if the service server is compromised and all the sharing records are deleted, then the user will not obtain the previous records corresponding to the utilized car information when there is a missing item on the car. Furthermore, if the sharing records are tampered or rewritten when the user has conducted fraudulent activities during car-sharing. It is difficult to obtain the user's evidence a crime from these records. In addition, if the stored information has been leaked, it brings serious privacy issues because it is related to the user's privacy. Therefore, it is necessary to resolve the above-mentioned problems incurred from a centralized structure.

Blockchain is a network technology that keeps transactions and establishes a chain form linked by hash values [7]. Blockchain is considered a trusted distributed ledger that ensures the decentralization and integrity of information to resolve the above-mentioned problem [8], [9]. The tamper-proof and traceable features of a blockchain system ensure the auditability of data operation, thereby ensuring data security [10], [11]. This paper proposes a decentralized car-sharing system model and a secure authentication protocol using blockchain to guarantee security, integrity, and decentralization. Stations provide a place for parking and sharing a car, and they act as the service vendor for a user to authenticate. These stations maintain a blockchain to provide a decentralized car-sharing service. When a car-sharing service occurs, the station authenticates the user and stores the service information in the blockchain. Furthermore, in the proposed system, a user utilizes the pseudonym for anonymity while using the car-sharing service. Therefore, even if the stored service information is leaked to an adversary, the attacker cannot infer the user.

A. CONTRIBUTION

The main contributions of this paper are as follows:

- A secure decentralized car-sharing system was designed using blockchain where stations provide a car-sharing service for the users replacing a single service vendor, and the stations maintain the blockchain by acting as a blockchain node.
- This paper proposes a secure authentication scheme for the decentralized car-sharing system, which withstands various attacks, including impersonation and replay attacks, and provides secure mutual authentication and privacy-preserving.
- The Burrows-Abadi-Needham (BAN) logic analysis is presented to analyze whether the proposed car-sharing scheme provides secure mutual authentication.
- The automated validation of internet security protocols and applications (AVISPA) was performed to analyze man-in-the-middle (MITM) and replay attacks. The performance analysis was compared with related schemes to show that the proposed authentication scheme can be applied to the blockchain-based car-sharing system.

B. PAPER ORGANIZATION

This paper is organized as follows. Sections II and III review previous interrelated researches and relevant preliminaries, respectively. A secure decentralized model of a car-sharing system is defined in Section IV. Section V presents the proposed car-sharing system. The security of the scheme is analyzed in Section VI, and the computation and the communication costs of the proposed scheme are discussed in Section VII. Finally, this paper is concluded in Section VIII.

II. RELATED WORK

Some studies discussed the security and user privacy in car-sharing systems [12]–[14]. Vaidaya and Mouftah [12] discussed security issues and the requirements of the car-sharing system. In their article, the connected and autonomous vehicles with external connectivity have security and privacy issues, such as eavesdropping, man-in-the-middle, replay, and denial-of-service attacks. Thus, secure communication and user authentication are essential for secure car-sharing systems. They also proposed a system overview of a personal vehicle sharing system. Symeonidis *et al.* [13] specified the security and privacy requirements for a car-sharing system. They reported that entity authentication, data integrity, confidentiality, non-repudiation, and authorization are required to design a car-sharing system to mitigate security threats. Furthermore, anonymity is needed to protect the users' privacy.

Some studies proposed an authentication protocol and secure system in a car-sharing system [15]–[19]. Busold *et al.* [15] suggested an authentication protocol for car access and rights delegation using a smartphone and access token. Wei *et al.* [16] proposed a hierarchical car-sharing system. Their system consisted of three entity levels: a key generation center was the top level; owners or sharing companies were the middle level; the users were the lowest level. Each level receives a key to access the vehicle from the upper level. Therefore, the user obtains the access key from

the owners or companies and uses it to access the sharing vehicle through NFC communication. Laurent *et al.* [17] proposed an authentication protocol for a car-sharing service, which addresses privacy-preserving using a pseudonym. Park *et al.* [18] suggested an authentication method using fingerprints. In their protocol, the server is vulnerable to a DoS attack. Dmitrienko *et al.* [19] proposed a secure free-floating car-sharing system. In their system, if a user wants to reserve the car-sharing service, the user is authenticated by the car-sharing provider to obtain an access token. The user can then access the vehicle using the access token and mobile device. However, their scheme did not consider the users' privacy. Moreover, these authentication schemes for car-sharing systems suffered from a single point of failure problem and bottleneck problem because they depend on a central node to manage the data and operate the system.

Recently, the characteristics of blockchain, such as decentralization, tamper-proof, and security, have motivated researchers in security authentication. Some blockchain-based authentication schemes [20]–[22] use a blockchain to achieve secure authentication without depending on a central node. Wang *et al.* [20] designed a blockchain-based anonymous authentication and key agreement protocol for a smart grid system. Xiong *et al.* [21] proposed a blockchain-based authentication scheme for multi-server architectures. Wang *et al.* [22] proposed a blockchain-assisted handover authenticated key agreement scheme in an edge-computing environment. In these schemes [20]–[22], the authentication servers, which maintained the blockchain, authenticate the user by employing the user's information stored in the blockchain. Therefore, there is no need for support by a registration authority in the authentication phase.

From related work, there has not been a secure authentication protocol for car-sharing systems. Therefore, this paper proposes a decentralized car-sharing system model and a secure authentication protocol using blockchain.

III. PRELIMINARIES

This section introduces the adversary model and relevant mathematical preliminaries used in this paper, including the blockchain and elliptic curve cryptosystem (ECC).

A. BLOCKCHAIN

Blockchain is a distributed ledger that offers decentralization, integrity, and tamper resistance. Blockchain can be classified into three categories: public blockchain (also called permissionless blockchain), consortium blockchain, and private blockchain (also both are permissioned blockchain) [23], [24]. In a public blockchain, every node keeps the ledgers, participates in the consensus, and has the permissions for reading and writing the data. This results in the arduous task of reaching a consensus quickly and high maintenance costs. Moreover, any node in the public blockchain can join or leave the network easily without authorization; hence, an adversary can easily join. Therefore, a public blockchain is unsuitable in a car-sharing system because the car-sharing records are related to the users' privacy. On the other hand, only authorized nodes can access the blockchain in a consortium blockchain and private blockchain. A private blockchain is managed by an authorized organization, and it has centralized characteristics [25]. On the other hand, a consortium blockchain is partially private, and has efficient consensus time and maintenance costs, which is operated under an authorized group. Therefore, a consortium blockchain was used to propose a car-sharing system.

B. ELLIPTIC CURVE CRYPTOSYSTEM (ECC)

ECC is a public-key cryptosystem, which is based on elliptic curve [26], [27]. It is widely utilized to construct cryptographic protocols because of a smaller key length and the same level of security compared to other encryption methods. In an ECC, an elliptic curve is defined as $E_p(a, b)$: $y^2 = x^3 + ax + b$ over a prime finite field $\mathbb{Z}_q$, where q is a large prime, $(a, b) \in \mathbb{Z}_q$, and $4a^3 + 27b^2 \neq 0 \pmod{q}$. Let P be a point on $E_p(a, b)$. The security of ECC depends on the following intractable problems.

- Elliptic Curve Discrete Logarithm Problem (ECDLP): The finding $x \in \mathbb{Z}_q$ in probability polynomial time is negligible when given two points Q and P, where $Q = x \cdot P$.
- Elliptic Curve Decisional Diffie-Hellman Problem (ECDDHP): The finding $(x \cdot y) \cdot P$ in probability polynomial time is negligible when given three points Q, R,and P, where $Q = x \cdot P$ and $R = y \cdot P$.

C. ADVERSARY MODEL

The capabilities of the adversary are based on the Dolev-Yao (DY) attack model. The Dolev-Yao threat model [28] is widely accepted in evaluating the security of a protocol [29]–[32]. The capabilities of an adversary model can be defined in the following manner:

- An attacker can intercept, modify, forge, and delete the messages transmitted via a public channel.
- An attacker can guess either the identity or the password of a user but cannot guess both of them simultaneously.
- An attacker can steal the mobile device of a legitimate user. The attacker can then attempt a power analysis attack to extract the stored values in the device [29], [33].
- An attacker can attempt various attacks, such as impersonation, man-in-the-middle, replay attacks, etc.

IV. SYSTEM MODEL

The proposed authentication scheme for a car-sharing system was designed based on blockchain consisting of five entities: trust authority, stations, owner, vehicle, and user. A trust authority sets up the system and issues the credential and pseudo-identity to the user and the vehicle owner as a trust entity. Stations have data storage and computing and organize the consortium blockchain. The user sends the request for car-sharing to the owner through the station. After being authenticated, the user receives the access code to unlock and

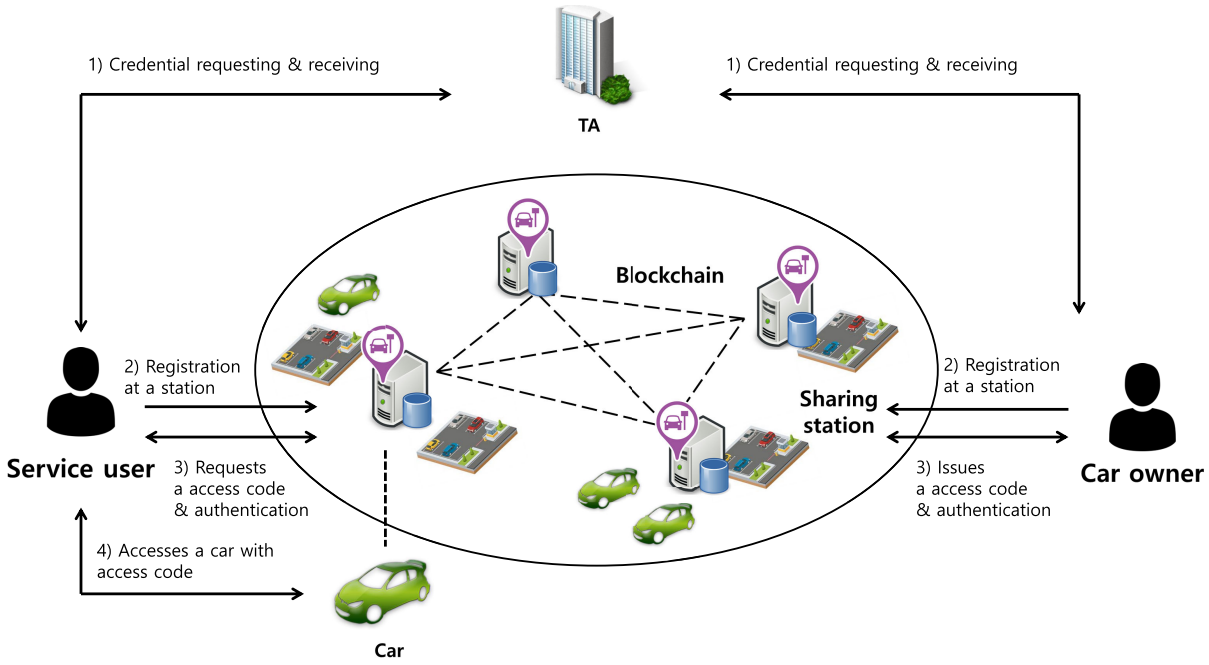

FIGURE 1. The proposed system model.

control the vehicle. The proposed system model is depicted in Figure 1.

- Trust authority: A trust authority is responsible for setting up the system, generating the keys for the stations, and issuing credentials and pseudo-identity to the user and vehicle owner. It is assumed that the trusted authority is not captured easily and is completely trustworthy. The credential proves who has a right and ability to drive, and the pseudo-identity is used in the car-sharing system to replace the real identity. When disputes occur in the car-sharing system, the trust authority exposes the identity of the malicious users based on the data stored in the blockchain.
- Stations: The station provides the car-sharing service place and platform for users and vehicle owners as an arbitrager. The station receives the user and owner's credentials for registration in the car-sharing system. The station verifies the received credentials and stores their information in the blockchain. When the station receives the user's request for a car-sharing service, the station authenticates the user using the information stored in the blockchain. It provides the car-sharing service by transmitting the information received from the vehicle owner. The station stores the provided service information in the blockchain, which can be used as the basis for the arbitration of disputes by the trusted authority.
- User: The user can use the car-sharing service through a mobile device, such as a smartphone. The user sends the request and authentication messages to the station to prove that the user is an authorized driver. The station authenticates the user based on the information stored at the blockchain. After being authenticated and obtaining the vehicle access code, the user can access the vehicle using their mobile device.
- Owner: The owner translates their vehicle to the shared vehicle by registering the information of the vehicle at the station. Once the station sends the user's request for sharing the vehicle, the owner generates the access code and transmits it to the station to distribute the access code to the user and vehicle.
- Vehicles: Vehicles are parked at the station and are ready for sharing by authorized users. There are communication modules and tamper-proofing modules in vehicles. The vehicle receives the access code through the communication modules, which it uses to check whether the user accessing it is authorized. All parameters used in vehicles are stored in a tamper-proof module for secrecy.

The communication flows on the proposed car-sharing system are depicted as follows:

1. User and owner send the real identities and licenses to TA to obtain the pseudo-identity and the credentials for registering a car-sharing system.
2. The user and owner register their pseudo identities, public keys, and information of shared car at the station to access the car-sharing service.
3. The user sends the station a request for access to a shared car using a mobile device. The station authenticates the user and notifies the request to the owner. The owner issues a code to access a shared car and sends the code to the user and car through the station.
4. The user utilizes the mobile device that stores the code to access the shared car and starts the sharing service.

TABLE 1. Symbols and their meanings.

Symbol	Description
U_i	i-th user
O_j	j-th vehicle's owner
C	The vehicle
TA	A trust authority
ST_s	s-th station
SID_s	Identity of ST_s
ID_i, PW_i	Identity and password of U_i
sk_x, PK_x	Secret key and public key of entity x
$h_1(\cdot), h_2(\cdot)$	Hash function
L	Location information
$info$	Vehicle's information
$request$	U_i's request including vehicle's information
$code$	Access code
T	Timestamp
TID	Transaction's identity
a_i, r_x, x_i, q_s, v_i	Random numbers
$\oplus$	XOR operation
$\|\|$	Concatenation operation

When the user finishes the sharing service, they park the car at the nearest station and send the return messages to the station.

V. PROPOSED SCHEME

This section presents the proposed secure authentication scheme for the car-sharing system based on blockchain. The proposed protocol includes the system setup phase, registration phase, authentication phase, and return phase. Table 1 lists the symbols used in the paper.

A. SYSTEM SETUP

Before the system, *TA* sets up the system parameters. *TA* selects large prime number p, q, an elliptic curve E_p, a base point P, two hash functions $h_1 : \{0,1\}^* \rightarrow \{0,1\}^*$, $h_2 : \{0,1\}^* \rightarrow Z_q$ and a secret key sk_{TA}. Then, the *TA* generates a public key $PK_{TA} = sk_{TA} \cdot P$ and publishes $(p, q, G, P, PK_{TA}, h_1, h_2)$ it as the system parameters. U_i and O_j are the received credential and pseudo-identity from TA before registration in the car-sharing system. These steps are executed over a secure channel. Figure 2 and 3 present the detailed process.

1) USER SETUP

- **Step 1:** U_i selects ID_i and PW_i and generates a private key and a random number sk_i, $a_i \in Z_q$. U_i then computes $PK_i = sk_i \cdot P$ and sends $\{ID_i, l, PK_i\}$ to *TA*, where l is a driving license.
- **Step 2:** After *TA* receives $\{ID_i, l, PK_i\}$, it verifies the ID_i and l. If it is valid, *TA* generates a random number $r_i \in Z_q$ and computes $PID_i = ID_i \oplus h_1(r_i \cdot PK_{TA})$, $R_i = r_i \cdot P$, $z_i = r_i + h_2(PID_i||R_i||PK_i) \cdot sk_{TA}$. Next, *TA* stores PID_i with R_i and sends $\{z_i, PID_i, R_i\}$ to U_i.
- **Step 3:** U_i calculates $HPW_i = h_1(PW_i||a_i)$, $B_i = h_1(ID_i||PW_i) \oplus a_i$, $C_i = h_1(h_1(ID_i||a_i)||HPW_i) \oplus sk_i$, $D_i = h_1(sk_i||a_i) \oplus z_i$, $E_i = h_1(sk_i||z_i||a_i)$, $HPID_i = PID_i \oplus HPW_i$ and stores $\{B_i, C_i, D_i, E_i, HPID_i, R_i\}$ in the memory of the mobile device.

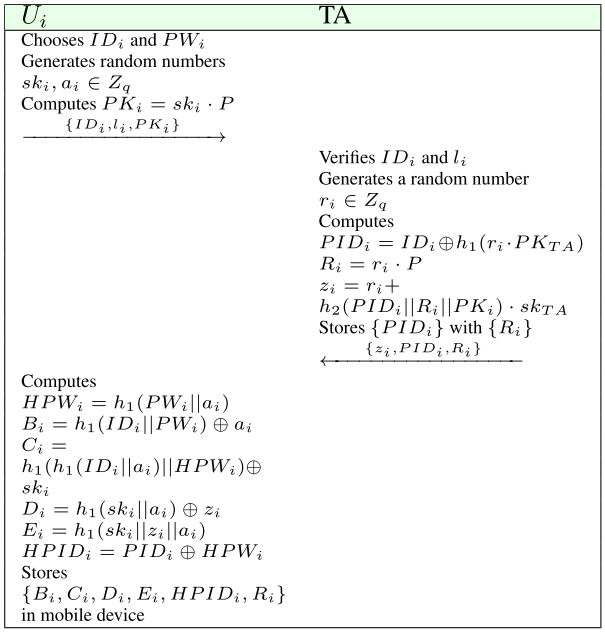

FIGURE 2. Setup phase of the user.

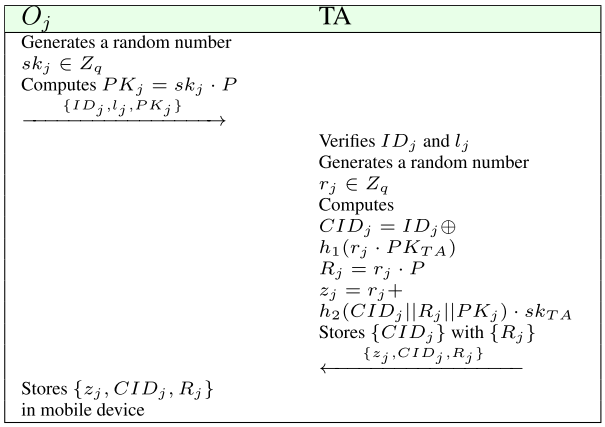

FIGURE 3. Setup phase of the owner.

2) OWNER SETUP

- **Step 1:** O_j generates a private key $sk_j \in Z_q$. Then, O_j computes $PK_j = sk_j \cdot P$ and sends $\{ID_j, l, PK_j\}$ to *TA*, where l is a driving license.
- **Step 2:** *TA* receives $\{ID_j, l, PK_j\}$, and *TA* verifies the ID_j and l. If it is valid, *TA* generates a random number $r_j \in Z_q$ and computes $CID_j = ID_j \oplus h_1(r_j \cdot PK_{TA})$, $R_j = r_j \cdot P$, $z_j = r_j + h_2(CID_j||R_j||PK_j) \cdot sk_{TA}$. And then, *TA* stores CID_j with R_j and sends $\{z_j, CID_j, R_j\}$ to O_j.
- **Step 3:** O_j stores $\{z_j, CID_j, R_j\}$ in the mobile device's memory.

B. REGISTRATION

U_i and O_j want to access the car-sharing system. They send their credentials z_i, z_j and pseudo-identities PID_i, CID_j to ST_s. ST_s checks the validity of z_i, z_j and the information of U_i and O_j in the blockchain if it is a valid credential. Figure 4 and 5 outline the detailed registration process.

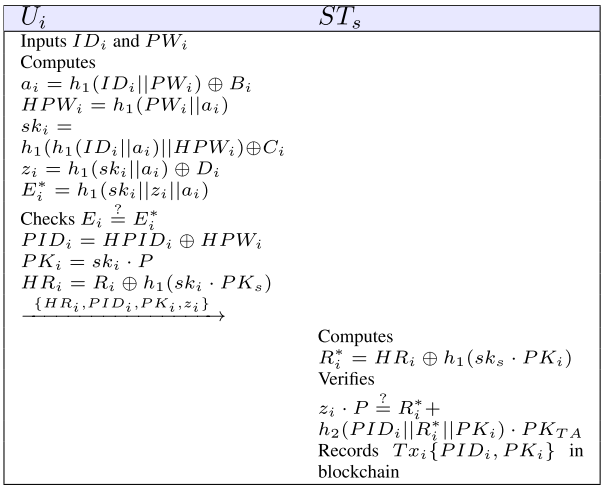

FIGURE 4. Registration phase of the user.

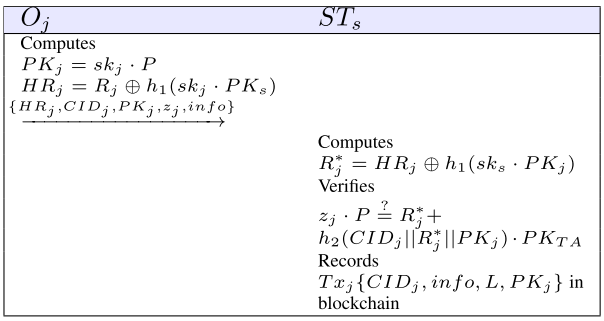

FIGURE 5. Registration phase of the owner.

1) USER REGISTRATION

- **Step 1:** U_i inputs ID_i and PW_i and calculates $a_i = h_1(ID_i||PW_i) \oplus B_i$, $HPW_i = h_1(PW_i||a_i)$, $sk_i = h_1(h_1(ID_i||a_i)||HPW_i) \oplus C_i$, $z_i = h_1(sk_i||a_i) \oplus D_i$, $E_i^* = h_1(sk_i||z_i||a_i)$. Then, U_i's mobile device checks $E_i^* \stackrel{?}{=} E_i$. If it is correct, U_i computes $PID_i = HPID_i \oplus HPW_i$, $PK_i = sk_i \cdot P$, $HR_i = R_i \oplus h_1(sk_i \cdot PK_s)$ and securely sends $\{HR_i, PID_i, PK_i, z_i\}$ to ST_s.
- **Step 2:** ST_s computes $R_i^* = HR_i \oplus h_1(sk_s \cdot PK_i)$ and verifies the U_i's credential $z_i \cdot P \stackrel{?}{=} R_i^* + h_2(PID_i||R_i^*||PK_i) \cdot PK_{TA}$. If it is valid, ST_s stores transaction including $\{PID_i, PK_i\}$ in the blockchain.

2) OWNER REGISTRATION

- **Step 1:** O_j calculates $PK_j = sk_j \cdot P$, $HR_j = R_j \oplus h_1(sk_j \cdot PK_s)$ and securely sends $\{HR_j, CID_j, PK_j, z_j, info\}$ to ST_s, where *info* is vehicle's information.
- **Step 2:** ST_s computes $R_j^* = HR_j \oplus h_1(sk_j \cdot PK_s)$ and verifies the O_j's credential $z_j \cdot P = R_j^* + h_2(CID_j||R_j^*||PK_j) \cdot PK_{TA}$. If it is valid, ST_s stores a transaction including $\{CID_j, info, L, PK_j\}$ in the blockchain. L is the location information about the car.

C. AUTHENTICATION

When U_i wants to use the car-sharing service, U_i must authenticate with the nearest ST_s. ST_s then collects the access code from O_j and sends it to U_i and the vehicle. U_i can then access the vehicle using the access code and starts the sharing service. Figure 6 summarizes the detailed authentication process.

- **Step 1:** U_i inputs ID_i and PW_i and computes $a_i = h_1(ID_i||PW_i) \oplus B_i$, $HPW_i = h_1(PW_i||a_i)$, $sk_i = h_1(h_1(ID_i||a_i)||HPW_i) \oplus C_i$, $z_i = h_1(sk_i||a_i) \oplus D_i$, $E_i^* = h_1(sk_i||z_i||a_i)$. Then, U_i's mobile device checks $E_i^* \stackrel{?}{=} E_i$. If it is correct, U_i generates a random number x_i and computes $X_i = x_i \cdot P$, $PID_i = HPID_i \oplus HPW_i$, $CPID_i = PID_i \oplus h_1(x_i \cdot PK_s)$, $Auth_{u_i} = x_i + h_2(PID_i||X_i||h_1(request)||T_1) \cdot sk_i$. Next, U_i sends messages $\{X_i, CPID_i, Auth_{u_i}, T_1, request\}$ including the request for accessing the shared car to ST_s through a public channel.
- **Step 2:** ST_s verifies the timestamp and computes $PID_i^* = CPID_i \oplus h_1(X_i \cdot sk_s)$. Next, ST_s checks that PID_i is stored in blockchain, and If so, ST_s retrieves PK_i. Then, ST_s verifies $Auth_{u_i} \cdot P \stackrel{?}{=} X_i + h_2(PID_i^*||X_i||h_1(request)||T_1) \cdot PK_i$. If it is valid, ST_s authenticate U_i and confirms *request* from U_i. And then, ST_s generates a random number q_s, computes $Q_s = q_s \cdot P$, $SM_1 = X_i \oplus h_1(q_s \cdot PK_j||T_2)$, $Auth_s = q_s + h_2(X_i||SID||Q_s||T_2) \cdot sk_s$ and sends $\{Q_s, SID, Auth_s, T_2, SM_1\}$ to O_j.
- **Step 3:** O_j checks T_2 and calculates $X_i^* = SM_1 \oplus h_1(Q_s \cdot sk_j||T_2)$. O_j verifies $Auth_s \cdot P \stackrel{?}{=} Q_s + h_2(X_i^*||SID||Q_s||T_2) \cdot PK_s$. If it is valid, O_j generates *code* for accessing to shared car and a random number y_j, and computes $CM_1 = \{code\} \oplus h_1(X_i^* \cdot y_j)$, $CM_2 = h_1(h_1(code)||X_i^* \cdot y_j||T_3)$, $CM_3 = \{code\} \oplus h_1(PK_c \cdot y_j)$, $CM_4 = h_1(h_1(code)||PK_c \cdot y_j||T_3)$, $CM_5 = h_1(h_1(code)||SID||Q_s \cdot y_j||T_3)$, $CM_6 = h_1(code) \oplus h_1(Q_s \cdot y_j)$. Thereafter, O_j sends $\{Y_j, CM_1, CM_2, CM_3, CM_4, CM_5, CM_6, T_3\}$ to ST_s.
- **Step 4:** ST_s checks T_3 and computes $h_1(code)^* = CM_6 \oplus h_1(q_s \cdot Y_j)$, $CM_5^* = h_1(h_1(code)^*||SID||q_s \cdot Y_j||T_3)$. Next, ST_s checks that $CM_5^* \stackrel{?}{=} CM_5$ is valid, and if so, ST_s stores car-sharing transaction including $\{PID_i, CID_j, h_1(code)^*\}$ in blockchain and calculates $SM_2 = TID \oplus h_1(X_i \cdot sk_s)$, $SM_3 = h_1(h_1(code)||TID||Y_j||T_4||X_i \cdot sk_s)$, $SM_4 = h_1(h_1(code)||Y_j||T_4||PK_c \cdot sk_s)$, where *TID* is the transaction's identity. Then, ST_s sends $\{CM_1, CM_2, SM_2, SM_3, T_3, T_4, Y_j\}$ to U_i and sends $\{CM_3, CM_4, SM_4, T_3, T_4, Y_j\}$ to car C.
- **Step 5:** U_i computes $\{code\} = CM_1 \oplus h_1(x_i \cdot Y_j)$, $CM_2^* = h_1(h_1(code)||x_i \cdot Y_j||T_3)$ and checks $CM_2^* \stackrel{?}{=} CM_2$. If it is valid, U_i calculates $TID^* = SM_2 \oplus h_1(X_i \cdot sk_s)$, $SM_3^* = h_1(h_1(code)||TID^*||Y_j||T_4||x_i \cdot PK_s)$. U_i then checks that $SM_3^* \stackrel{?}{=} SM_3$ is correct, and if so, U_i stores $\{TID, code\}$. And C computes $\{code\} = CM_3 \oplus h_1(sk_c \cdot Y_j)$, $CM_4^* = h_1(h_1(code)||sk_c \cdot Y_j||T_3)$ and verifies $CM_4^* \stackrel{?}{=} CM_4$ is correct. If it is correct, C computes

U_i	ST_s	O_j
Inputs ID_i and PW_i Computes $a_i = h_1(ID_i \|\| PW_i) \oplus B_i$ $HPW_i = h_1(PW_i \|\| a_i)$ $sk_i = h_1(h_1(ID_i \|\| a_i) \|\| HPW_i) \oplus C_i$ $z_i = h_1(sk_i \|\| a_i) \oplus D_i$ $E_i^* = h_1(sk_i \|\| z_i \|\| a_i)$ Checks $E_i \stackrel{?}{=} E_i^*$ Generates a random number x_i $X_i = x_i \cdot P$ $PID_i = HPID_i \oplus HPW_i$ $CPID_i = PID_i \oplus h_1(x_i \cdot PK_s)$ $Auth_{u_i} = x_i +$ $h_2(PID_i \|\| X_i \|\| h_1(request) \|\| T_1) \cdot sk_i$ $\{X_i, CPID_i, Auth_{u_i}, T_1, request\}$ $\longrightarrow$		
	Checks ΔT_1 Computes $PID_i^* = CPID_i \oplus h_1(X_i \cdot sk_s)$ Checks $\{PID_i\}$ in blockchain and retrieves $\{PK_i\}$ Verifies $Auth_{u_i} \cdot P \stackrel{?}{=} X_i +$ $h_2(PID_i^* \|\| X_i \|\| h_1(request) \|\| T_1) \cdot PK_i$ Checks $\{request\}$ Generates a random number q_s $Q_s = q_s \cdot P$ $SM_1 = X_i \oplus h_1(q_s \cdot PK_j)$ $Auth_s = q_s + h_2(X_i \|\| SID_s \|\| Q_s \|\| T_2) \cdot sk_s$ $\{Q_s, SID_s, Auth_s, T_2, SM_1\}$ $\longrightarrow$	
		Checks ΔT_2 Computes $X_i^* = SM_1 \oplus h_1(Q_s \cdot sk_j)$ Verifies $Auth_s \cdot P \stackrel{?}{=} Q_s + h_2(X_i^* \|\| SID_s \|\| Q_s \|\| T_2) \cdot PK_s$ Generates a access code $\{code\}$ and a random number y_j $Y_j = y_j \cdot P$ $CM_1 = \{code\} \oplus h_1(X_i^* \cdot y_j)$ $CM_2 = h_1(h_1(code) \|\| X_i^* \cdot y_j \|\| T_3)$ $CM_3 = \{code\} \oplus h_1(PK_c \cdot y_j)$ $CM_4 = h_1(h_1(code) \|\| PK_c \cdot y_j \|\| T_3)$ $CM_5 = h_1(h_1(code) \|\| SID_s \|\| Q_s \cdot y_j \|\| T_3)$ $CM_6 = h_1(code) \oplus h_1(Q_s \cdot y_j)$ $\{Y_j, CM_1, CM_2, CM_3, CM_4, CM_5, CM_6, T_3\}$ $\longleftarrow$
	Checks ΔT_3 Computes $h_1(code)^* = CM_6 \oplus h_1(q_s \cdot Y_j)$ $CM_5^* = h_1(h_1(code)^* \|\| SID_s \|\| q_s \cdot Y_j \|\| T_3)$ Verifies $CM_5^* \stackrel{?}{=} CM_5$ Records $Tx_j\{PID_i, CID_j, h_1(code)\}$ in blockchain $SM_2 = TID \oplus h_1(X_i \cdot sk_s)$ $SM_3 = h_1(h_1(code) \|\| TID \|\| Y_j \|\| T_4 \|\| X_i \cdot sk_s)$ $SM_4 = h_1(h_1(code) \|\| Y_j \|\| T_4 \|\| PK_c \cdot sk_s)$ $\{CM_1, CM_2, SM_2, SM_3, T_3, T_4, Y_j\}$ $\longleftarrow$	
Computes $\{code\} = CM_1 \oplus h_1(x_i \cdot Y_j)$ $CM_2^* = h_1(h_1(code) \|\| x_i \cdot Y_j \|\| T_3)$ Checks $CM_2 \stackrel{?}{=} CM_2^*$ $TID^* = SM_2 \oplus h_1(x_i \cdot PK_s)$ $SM_3^* = h_1(h_1(code) \|\| TID^* \|\| Y_j \|\| T_4 \|\| x_i \cdot PK_s)$ Checks $SM_3 \stackrel{?}{=} SM_3^*$ Stores $\{TID, code\}$		
		$\{CM_3, CM_4, SM_4, T_3, T_4, Y_j\}$ $\longleftarrow$
(Car C) Computes $\{code\} = CM_3 \oplus h_1(Y_j \cdot sk_c)$ $CM_4^* = h_1(h_1(code) \|\| Y_j \cdot sk_c \|\| T_3)$ Checks $CM_4 \stackrel{?}{=} CM_4$ $SM_4^* = h_1(h_1(code) \|\| Y_j \|\| T_4 \|\| sk_c \cdot PK_s)$ Checks $SM_4^* \stackrel{?}{=} SM_4$ $Stores\{code\}$		

FIGURE 6. Authentication phase.

$SM_4^* = h_1(h_1(code)||Y_j||T_4||sk_c \cdot PK_s)$. C then checks that $SM_4^* \stackrel{?}{=} SM_4$ is correct, and if so, C stores the $\{code\}$. Finally, U_i and C have the same access code and U_i can access to C.

D. RETURN

U_i finishes the car-sharing service and returns the vehicle to a nearby station. When ST_s' receives a return request from U_i, ST_s' updates the C's information in the blockchain and notifies C that the sharing service has ended. C then expunges the *code*

- **Step 1:** U_i inputs ID_i and PW_i and computes $a_i = h_1(ID_i||PW_i) \oplus B_i$, $HPW_i = h_1(PW_i||a_i)$, $sk_i = h_1(h_1(ID_i||a_i)||HPW_i) \oplus C_i$, $z_i = h_1(sk_i||a_i) \oplus D_i$, $E_i^* = h_1(sk_i||z_i||a_i)$. Then, U_i's mobile device checks $E_i^* \stackrel{?}{=} E_i$. If it is correct, U_i generates a random number v_i and computes $V_i = v_i \cdot P$, $UM_1 = \{TID, h_1(code)\} \oplus h_1(v_i \cdot PK_s')$, $UM_2 = h_1(PID_i||TID||h_1(code)||v_i \cdot PK_s'||T_5)$. Next, U_i sends $\{V_i, UM_i, UM_2, T_5\}$ to ST_S'.
- **Step 2:** ST_s' checks T_5 and computes $\{TID^*, h_1(code)^*\} = UM_1 \oplus h_1(V_i \cdot sk_s')$. Next, ST_s' retrieves the transaction corresponding to TID^* from the blockchain. Then, ST_s' calculates $UM_2^* = h_1(PID_i||TID^*||h_1(code)||V_i \cdot sk_s'||T_5)$ and checks $UM_2^* \stackrel{?}{=} UM_2$. If it is correct, ST_s' computes $SM_5 = h_1(code) \oplus h_1(PK_c \cdot sk_s'||T_6)$, $SM_6 = h_1(h_1(code)||T_6||PK_c \cdot sk_s')$ and sends $\{SM_5, SM_6, T_6\}$ to C.
- **Step 3:** C checks T_6 and computes $h_1(code)^* = SM_5 \oplus h_1(PK_s' \cdot sk_c||T_6)$, $SM_6 = h_1(h_1(code)^*||T_6||PK_s' \cdot sk_c)$. Next, C checks $SM_6^* \stackrel{?}{=} SM_6$ and $h_1(code)^* \stackrel{?}{=} h_1(code)$ are valid, and if so, C revokes *code* and computes $CR_1 = h_1(code) \oplus h_1(PK_s' \cdot sk_c||T_7)$, $CR_2 = h_1(h_1(code)||T_7||PK_s' \cdot sk_c)$. After that, C sends $\{CR_1, CR_2, T_7\}$ to ST_s'.
- **Step 4:** ST_s' checks T_7 and calculates $h_1(code)^* = CR_1 \oplus h_1(PK_c \cdot sk_s'||T_7)$, $CR_2^* = h_1(h_1(code)^*||T_7||PK_c \cdot sk_s')$. Then, ST_s' verifies $CR_2^* \stackrel{?}{=} CR_2$ is valid. If it is valid, ST_s' stores transaction including $\{CID_j, info, L^*, TID_{END}\}$ in blockchain. TID_{END} is a identity of the last car-sharing transaction.

VI. SECURITY ANALYSIS

In this section, we conduct the formal security analysis using BAN-logic [34] and AVSIPA [35], [36], and informal security analysis. We then prove whether the proposed scheme is secure against malicious attacks and provides mutual authentication.

A. BAN LOGIC ANALYSIS

The BAN logic [34], which is a widely accepted formal security analysis [37]–[40], was performed to demonstrate that the proposed scheme achieves mutual authentication. This section describes the basic notations used in the BAN logic proof and presents the BAN logic postulates, the security goals, assumptions, and idealized forms. Finally, the BAN logic proof was performed to confirm the mutual authentication of the proposed scheme.

TABLE 2. Notations of the BAN logic.

Notation	Description
$P\| \equiv S$	P **believes** the statement S
$P\| \sim S$	P once **said** S
$P \Rightarrow S$	P **controls** the statement S
$P \lhd S$	P **sees** the statement S
$\#S$	The statement S is **fresh**
$< S >_F$	Formula S is **united** with formula F
$\{S\}_K$	Formula S is **encrypted** with K
$\stackrel{K}{\longrightarrow} Q$	Q has a public key K
$P \stackrel{K}{\longleftrightarrow} Q$	P and Q have **shared** key K

1) POSTULATES OF BAN LOGIC

Postulates of BAN logic are as follows.

1. Message meaning rule:

$$\frac{P \mid \equiv P \stackrel{K}{\leftrightarrow} Q, \quad P \lhd (S)_K}{P\| \equiv Q\| \sim S}$$

2. Jurisdiction rule:

$$\frac{P\| \equiv Q\| \Longrightarrow S, \quad P\| \equiv Q\| \equiv S}{P \mid \equiv S}$$

3. Nonce verification rule:

$$\frac{P\| \equiv \#(S), \quad P\| \equiv Q \mid \sim S}{P\| \equiv Q\| \equiv S}$$

4. Belief rule:

$$\frac{P \mid \equiv (S, X)}{P \mid \equiv S}$$

5. Freshness rule:

$$\frac{P \mid \equiv \#(S)}{P \mid \equiv \#(S, X)}$$

2) GOALS

The following goals are presented to prove that the proposed system achieves secure mutual authentication.

Goal 1: $ST_s| \equiv (x_i)$
Goal 2: $ST_s| \equiv U_i| \equiv (x_i)$
Goal 3: $O_j| \equiv (q_s, x_i)$
Goal 4: $O_j| \equiv ST_s| \equiv (q_s, x_i)$
Goal 5: $ST_s| \equiv h_1(code)$
Goal 6: $ST_s| \equiv O_j| \equiv (h_1(code))$
Goal 7: $U_i| \equiv (TID, h_1(code))$
Goal 8: $U_i| \equiv ST_s| \equiv (TID, h_1(code))$

3) IDEALIZED FORMS

The idealized forms are as follows.

Msg_1: $U_i \rightarrow ST_s : (x_i, PID_i, T_1) \xrightarrow[ST_s]{PK_s}$
Msg_2: $ST_s \rightarrow O_j : (q_s, x_i, SID_s, T_2) \xrightarrow[O_j]{PK_j}$
Msg_3: $O_j \rightarrow ST_s : (h_1(code), y_j, T_3)_{q_s}$
Msg_4: $ST_s \rightarrow U_i : (h_1(code), TID, y_j, T_3, T_4)_{x_i}$

4) ASSUMPTIONS

The assumptions to perform the BAN logic proof are defined as follows.

A_1: $ST_s| \equiv \#(PK_s)$
A_2: $ST_s| \equiv \#(T_1)$
A_3: $ST_s| \equiv U_i| \Rightarrow (x_i)$
A_4: $O_j| \equiv \#(PK_j)$
A_5: $O_j| \equiv \#(T_2)$
A_6: $O_j| \equiv ST_s| \Rightarrow (q_s, x_i)$
A_7: $ST_s| \equiv (O_j \overset{q_s}{\leftrightarrow} ST_s)$
A_8: $ST_s| \equiv \#(T_3)$
A_9: $ST_s| \equiv O_j| \Rightarrow (h_1(code))$
A_{10}: $U_i| \equiv (ST_s \overset{x_i}{\leftrightarrow} U_i)$
A_{11}: $U_i| \equiv \#(T_4)$
A_{12}: $U_i| \equiv ST_s| \Rightarrow (TID, h_1(code))$

5) BAN LOGIC PROOF

The BAN logic proof of the proposed protocol is as follows

Step 1: S_1 is obtained according to Msg_1.

$$S_1 : ST_s \lhd (x_i, PID_i, T_1) \xrightarrow[ST_s]{PK_s}$$

Step 2: S_2 is obtained by applying the MMR using S_1 and A_1.

$$S_2 : ST_s| \equiv U_i| \sim (x_i, PID_i, T_1)$$

Step 3: S_3 is obtained by applying the FR using A_2.

$$S_3 : ST_s| \equiv \#(x_i, PID_i, T_1)$$

Step 4: S_4 is obtained by applying the NVR using S_2 and S_3.

$$S_4 : ST_s| \equiv U_i| \equiv (x_i, PID_i, T_1) \quad (1)$$

Step 5: S_5 is obtained from S_4 and the BR.

$$S_5 : ST_s| \equiv U_i| \equiv (x_i) \quad (Goal2)$$

Step 6: S_6 is obtained from Msg_2.

$$S_6 : O_j \lhd (q_s, x_i, SID_s, T_2) \xrightarrow[O_j]{PK_j}$$

Step 7: S_7 is obtained by applying the MMR using A_4 and S_6.

$$S_7 : O_j| \equiv ST_s| \sim (q_s, x_i, SID_s, T_2)$$

Step 8: S_8 is obtained by applying the FR using A_5.

$$S_8 : O_j| \equiv \#(q_s, x_i, SID_s, T_2)$$

Step 9: S_9 is obtained by applying the NVR using S_7 and S_8.

$$S_9 : O_j| \equiv ST_s| \equiv (q_s, x_i, SID_s, T_2)$$

Step 10: S_{10} is obtained from S_9 and the BR.

$$S_{10} : O_j| \equiv ST_s| \equiv (q_s, x_i) \quad \textbf{(Goal4)}$$

Step 11: S_{11} is obtained from Msg_3.

$$S_{11} : ST_s \lhd (h_1(code), y_j, T_3)_{q_s}$$

Step 12: S_{12} is obtained by applying the MMR using A_7 and S_{11}.

$$S_{12} : ST_s| \equiv O_j| \sim (h_1(code), y_j, T_3)$$

Step 13: S_{13} is obtained by applying the FR using A_8.

$$S_{13} : ST_s| \equiv \#(h_1(code), y_j, T_3)$$

Step 14: S_{14} is obtained by applying the NVR using S_{12} and S_{13}.

$$S_{14} : ST_s| \equiv O_j| \equiv (h_1(code), y_j, T_3)$$

Step 15: S_{15} is obtained from S_{14} and the BR.

$$S_{15} : ST_s| \equiv O_j| \equiv (h_1(code)) \quad \textbf{(Goal6)}$$

Step 16: S_{16} is obtained from Msg_4.

$$S_{16} : U_i \lhd (h_1(code), TID, y_j, T_3, T_4)_{x_i}$$

Step 17: S_{17} is obtained by applying the MMR using A_{10} and S_{16}.

$$S_{17} : U_i| \equiv ST_s| \sim (h_1(code), TID, y_j, T_3, T_4)$$

Step 18: S_{18} is obtained by applying the FR using A_{11}.

$$S_{18} : U_i| \equiv \#(h_1(code), TID, y_j, T_3, T_4)$$

Step 19: S_{19} is obtained by applying the NVR using S_{17} and S_{18}.

$$S_{19} : U_i| \equiv ST_s| \equiv (h_1(code), TID, y_j, T_3, T_4)$$

Step 20: S_{20} is obtained from S_{19} and the BR.

$$S_{20} : U_i| \equiv ST_s| \equiv (TID, h_1(code)) \quad \textbf{(Goal8)}$$

Step 21: S_{21} is obtained by applying the JR using A_3 and S_5.

$$S_{21} : ST_s| \equiv (x_i) \quad \textbf{(Goal1)}$$

Step 22: S_{23} is obtained by applying the JR using A_6 and S_{10}.

$$S_{22} : O_j| \equiv (q_s, x_i) \quad \textbf{(Goal3)}$$

Step 23: S_{23} is obtained by applying the JR using A_9 and S_{15}.

$$S_{23} : ST_s| \equiv (h_1(code)) \quad \textbf{(Goal5)}$$

Step 24: S_{24} is obtained by applying the JR using A_{12} and S_{20}.

$$S_{24} : U_i| \equiv (TID, h_1(code)) \quad \textbf{(Goal7)}$$

B. AVISPA ANALYSIS

The AVISPA simulation tool [35], [36] was used to analyze that the proposed protocol is secure against replay and man-in-the-middle attacks. The AVISPA simulation tool uses High-Level Protocol Specification Language (HLPSL) [41] to implement a designed security protocol. The HLPSL is converted to ''Intermediate Format (IF)'' with the help of the HLPSL2IF translator. Four backends are associated with the AVISPA simulation tool: ''On-the-Fly Model Checker (OFMC)'', ''Constraint Logic-based Attack Searcher (CL-AtSE)'', ''Tree automata based on Automatic Approximations for Analysis of Security Protocol (TA4SP)'', and ''SAT-based Model Checker (SATMC)''. The IF is then given to one of the four backend models to produce the ''Output Format (OF)''. The OF presents the security analysis results of the protocol in few sections, which include the following: 'SUMMARY', which indicates that a protocol being 'SAFE' or 'UNSAFE'; 'DETAILS' that explains the declared result on the 'SUMMARY' section; 'PROTOCOL' that defines the HLPSL specification of the scheme in IF form; 'BACKENED', which displays the name of the backend which is used for the analysis. Figures 7, 8, 9, and 10 describe the role of user, station, owner, and trust authority nodes, respectively. Figure 11 indicates the goals and the role of the session and environment of the proposed protocol. Figure 12 presents the AVISAP simulation result of the proposed protocol using CL-AtSe and OFMC. The results under the CL-AtSe and OFMC backends show that the proposed protocol is safe. Therefore, the proposed protocol can be resilient against man-in-the-middle and replay attacks.

C. INFORMAL SECURITY ANALYSIS

Informal security analysis was performed to demonstrate that the proposed protocol prevents various attacks and supports user anonymity and mutual authentication.

1) IMPERSONATION ATTACK

A malicious adversary attempts to disguise themselves as a legitimate user by generating an authentication message $\{X_i, CPID_i, Auth_{u_i}, T_1, request\}$. However, the adversary is unable to generate the authentication message because they do not know the user's private key ski, random number xi, identity ID_i, password PW_i. Therefore, the adversary cannot generate the authentication message of a legitimate user, so the proposed scheme prevents the impersonation attack.

2) STOLEN MOBILE DEVICE ATTACK

Assume that a malicious adversary steals the mobile device of a legitimate user and can extract the stored information in the mobile device by conducting power analysis. The adversary can obtain $\{B_i, C_i, D_i, E_i, HPID_i, R_i\}$. However, the adversary cannot obtain sensitive information of a legitimate user because that information is masked with XOR and hash operations. Thus, the proposed scheme does not reveal any sensitive information if the mobile device of a legitimate user is stolen or lost.

```
role user(U,O,TA,ST : agent, SKuta, SKota, SKust, SKost : symmetric_key, H1,H2:
hash_func, SND, RCV : channel(dy))

played_by U
def=
local State: nat,
   MUL, ADD : hash_func,
   IDi,PWi,SKi,Aii, Ai,Li, Rii, PIDi, PKi, P, Ri, Zi, PKta, SKta, HPWi, Bi, Ci, Di, Ei,
HPIDi :text,
   SKj, IDj, Lj, PKj, Rjj, CIDj, Rj, Zj, PKs, SKs, INfo : text,
   HRi, HRj, Xii, Xi, CPIDi, AUTHui, REQuest, Qss, Qs, SM1, AUTHs, SIDs, T1, T2 :
text,
   Yjj, Yj, CODE, CM1, CM2, CM5, CM6, SM2, SM3, TID, T3, T4 : text
const sp1, sp2, sp3, sp4, sp5, sp6, sp7, sp8, sp9,
u_st_xi,o_u_yjj,st_u_yjj,o_st_yjj,st_o_qss: protocol_id
init State := 0
transition

%%%%%%%%%%%%%Setup phase
1. State = 0 /\ RCV(start) =|>
State' := 1 /\ PKi' := MUL(SKi.P)
      /\ SND({IDi.Li.PKi'}_SKuta)

2. State = 1 /\ RCV({ADD(Rii'.MUL(H2(xor(IDi,
H1(MUL(Rii.PKta))).MUL(Rii'.P).MUL(SKi.P)).SKta)).xor(IDi,
H1(MUL(Rii.PKta))).MUL(Rii'.P)}_SKuta) =|>
 State' := 2 /\ Aii' := new() /\ HPWi' := H1(PWi.Aii')
      /\ Bi' := xor(H1(IDi.PWi),Aii')
      /\ Ci' := xor{H1(H1(IDi.Aii').HPWi'),SKi)
      /\ Di' := xor(H1(SKi.Aii'),ADD(Rii'.MUL(H2(xor(IDi,
H1(MUL(Rii.PKta))).MUL(Rii'.P).MUL(SKi.P)).SKta)))
      /\ Ei' := H1(SKi.ADD(Rii'.MUL(H2(xor(IDi,
H1(MUL(Rii.PKta))).MUL(Rii'.P).MUL(SKi.P)).SKta)).Aii')
      /\ HPIDi' := xor(xor(IDi, H1(MUL(Rii.PKta))),HPWi')
      /\ HRi' := xor(MUL(Rii'.P),H1(MUL(SKi.PKs)))
      /\ SND({HRi'.xor(IDi,
H1(MUL(Rii.PKta))).MUL(SKi.P).ADD(Rii'.MUL(H2(xor(IDi,
H1(MUL(Rii.PKta))).MUL(Rii'.P).MUL(SKi.P)).SKta))}_SKust)
      /\ secret({PWi,SKi,Aii'}, sp1, {U})

3. State = 2 /\ RCV({xor(IDi, H1(MUL(Rii.PKta)))}_SKust) =|>
State' := 3 /\ Xii' := new() /\ T1' := new() /\ Xi' := MUL(Xii'.P)
      /\ CPIDi' := xor(xor(IDi, H1(MUL(Rii.PKta))), H1(MUL(Xii'.PKs)))
      /\ AUTHui' := ADD(Xii'.MUL(H2(xor(IDi,
H1(MUL(Rii.PKta))).Xii'.H1(REQuest).T1').SKi))
      /\ REQuest' := new()
      /\ SND(Xi'.CPIDi'.AUTHui'.T1'.REQuest')
      /\ witness(U,ST,u_st_xi,Xii')

4. State = 3 /\ RCV(xor(CODE',
H1(MUL(MUL(Xii'.P).Yjj'))).H1(H1(CODE').MUL(MUL(Xii'.P).Yjj').T3').xor(TID,H
1(MUL(MUL(Xii'.P).SKs))).H1(H1(CODE').TID.MUL(Yjj'.P).T4'.MUL(MUL(Xii'.P).S
Ks)).T3'.T4'.MUL(Yjj'.P)) =|>
State' := 4 /\ request(O,U,o_u_yjj, Yjj') /\ request(ST,U,st_u_yjj,Yjj')


end role
```

FIGURE 7. Role of the user.

3) OFFLINE PASSWORD GUESSING ATTACK

Assume that an adversary can guess the identity ID_i or password PW_i of a legitimate user. The proposed method also considers that the adversary is in possession of a legitimate user's mobile device. The adversary can obtain the stored information $\{B_i, C_i, D_i, E_i, HPID_i, R_i\}$ in the mobile device and obtain the transmitted messages $\{X_i, CPID_i, Auth_{u_i}, T_1, request\}$, $\{CM_1, CM_2, SM_2, SM_3, T_3, T_4, Y_j\}$ through public channels. However, the adversary cannot compute $a_i = h_1(ID_i||PW_i) \oplus B_i$ without guessing the correct values for ID_i and PW_i simultaneously. Thus, the adversary cannot check either ID_i or PW_i at the same time using the extracted $C_i = sk_i \oplus h_1(h_1(ID_i||a_i)||h_1(PW_i||a_i))$. Hence, the proposed scheme is not vulnerable to an offline guessing attack.

```
role station(U,O,TA,ST : agent, SKuta, SKota, SKust, SKost : symmetric_key, H1,H2:
hash_func, SND, RCV : channel(dy))

played_by ST
def=
local State: nat,
   MUL, ADD : hash_func,
   IDi,PWi,SKi,Aii, Ai,Li, Rii, PIDi, PKi, P, Ri, Zi, PKta, SKta, HPWi, Bi, Ci, Di, Ei,
HPIDi :text,
   SKj, IDj, Lj, PKj, Rjj, CIDj, Rj, Zj, PKs, SKs, INfo : text,
   HRi, HRj, Xii, Xi, CPIDi, AUTHui, REQuest, Qss, Qs, SM1, AUTHs, SIDs, T1, T2 :
text,
   Yjj, Yj, CODE, CM1, CM2, CM5, CM6, SM2, SM3, TID, T3, T4 : text
const sp1, sp2, sp3, sp4, sp5, sp6, sp7, sp8, sp9,
u_st_xi,o_u_yjj,st_u_yjj,o_st_yjj,st_o_qss: protocol_id
init State := 0
transition

1. State = 0 /\ RCV({xor(MUL(Rii'.P),H1(MUL(SKi.PKs))).xor(IDi,
H1(MUL(Rii.PKta))).MUL(SKi.P).ADD(Rii'.MUL(H2(xor(IDi,
H1(MUL(Rii.PKta))).MUL(Rii'.P).MUL(SKi.P)).SKta))}_SKust)  =|>
State' := 1 /\ SND({xor(IDi, H1(MUL(Rii.PKta)))}_SKust)
      /\ secret({SKs},sp8,{ST})

2. State = 1
/\ RCV({xor(MUL(Rjj'.P),H1(MUL(SKj.PKs))).xor(IDj,H1(MUL(Rjj'.PKta))).MUL(S
Ki.P).ADD(MUL(Rjj'.P).MUL(H2(xor(IDj,H1(MUL(Rjj'.PKta))).MUL(Rjj'.P).MUL(S
Kj.P)).SKta)).INfo}_SKost)  =|>
State' := 2 /\ secret({INfo},sp9,{O,ST})


3. State = 2 /\ RCV(MUL(Xii'.P).xor(xor(IDi, H1(MUL(Rii.PKta))),
H1(MUL(Xii'.PKs))).ADD(Xii'.MUL(H2(xor(IDi,
H1(MUL(Rii.PKta))).Xii'.H1(REQuest).T1').SKi)).T1'.REQuest')  =|>
State' := 3 /\ Qss' := new() /\ T2' := new() /\ Qs' := MUL(Qss'.P)
      /\ SM1' := xor(MUL(Xii'.P),H1(Qss'.MUL(SKj.P)))
      /\ AUTHs' := ADD(Qss'.MUL(H2((Xii'.P).SIDs.Qs'.T2').SKs))
      /\ SND(Qs'.SIDs.AUTHs'.T2'.SM1')
      /\ witness(ST,O,st_o_qss,Qss')
      /\ request(U,ST,u_st_xi,Xii')


4. State = 3 /\ RCV(MUL(Yjj'.P).xor(CODE',
H1(MUL(MUL(Xii'.P).Yjj'))).H1(H1(CODE').MUL(MUL(Xii'.P).Yjj').T3').H1(H1(CO
DE').SIDs.MUL(MUL(Qss'.P).Yjj').T3').xor(H1(CODE'),H1(MUL(MUL(Qss'.P).Yjj'))
).T3')  =|>
State' := 4 /\ T4' := new() /\ SM2' := xor(TID,H1(MUL(MUL(Xii'.P).SKs)))
      /\ SM3' := H1(H1(CODE').TID.MUL(Yjj'.P).T4'.MUL(MUL(Xii'.P).SKs))
      /\ SND(xor(CODE',
H1(MUL(MUL(Xii'.P).Yjj'))).H1(H1(CODE').MUL(MUL(Xii'.P).Yjj').T3').SM2'.SM3'.
T3'.T4'.MUL(Yjj'.P))
      /\ witness(ST,U,st_u_yjj,Yjj')
      /\ request(O,ST,o_st_yjj,Yjj')

end role
```

FIGURE 8. Role of the station.

```
role owner(U,O,TA,ST : agent, SKuta, SKota, SKust, SKost : symmetric_key, H1,H2:
hash_func, SND, RCV : channel(dy))

played_by O
def=
local State: nat,
   MUL, ADD : hash_func,
   IDi,PWi,SKi,Aii, Ai,Li, Rii, PIDi, PKi, P, Ri, Zi, PKta, SKta, HPWi, Bi, Ci, Di, Ei,
HPIDi :text,
   SKj, IDj, Lj, PKj, Rjj, CIDj, Rj, Zj, PKs, SKs, INfo : text,
   HRi, HRj, Xii, Xi, CPIDi, AUTHui, REQuest, Qss, Qs, SM1, AUTHs, SIDs, T1, T2 :
text,
   Yjj, Yj, CODE, CM1, CM2, CM5, CM6, SM2, SM3, TID, T3, T4 : text
const sp1, sp2, sp3, sp4, sp5, sp6, sp7, sp8, sp9,
u_st_xi,o_u_yjj,st_u_yjj,o_st_yjj,st_o_qss: protocol_id
init State := 0
transition

%%%%%%%%%%%%%Setup phase
1. State = 0 /\ RCV(start) =|>
State' := 1 /\ PKj' := MUL(SKj.P)
      /\ SND({IDj.Lj.PKj'}_SKota)
      /\ secret({IDj,SKj},sp5,{O})

2. State = 1
/\ RCV({ADD(Rj'.MUL(H2(xor(IDj,H1(MUL(Rjj'.PKta))).MUL(Rjj'.P).MUL(SKj.P)).
SKta)).xor(IDj,H1(MUL(Rjj'.PKta))).MUL(Rjj'.P)}_SKota)  =|>
State' := 2 /\ HRj' := xor(MUL(Rjj'.P),H1(MUL(SKj.PKs)))

/\ SND({HRj'.xor(IDj,H1(MUL(Rjj'.PKta))).MUL(SKi.P).ADD(MUL(Rjj'.P).MUL(H2
(xor(IDj,H1(MUL(Rjj'.PKta))).MUL(Rjj'.P).MUL(SKj.P)).SKta)).INfo}_SKost)

3. State = 2
/\ RCV(MUL(Qss'.P).SIDs.ADD(Qss'.MUL(H2((Xii'.P).SIDs.MUL(Qss'.P).T2').SKs))
.T2'.xor(MUL(Xii'.P),H1(Qss'.MUL(SKj.P))))  =|>
State' := 3 /\ CODE' := new() /\ T3' := new() /\ Yjj' := new()
      /\ Yj' := MUL(Yjj'.P) /\ CM1' := xor(CODE', H1(MUL(MUL(Xii'.P).Yjj')))
      /\ CM2' := H1(H1(CODE').MUL(MUL(Xii'.P).Yjj').T3')
      /\ CM5' := H1(H1(CODE').SIDs.MUL(MUL(Qss'.P).Yjj').T3')
      /\ CM6' := xor(H1(CODE'),H1(MUL(MUL(Qss'.P).Yjj')))
      /\ SND(Yj'.CM1'.CM2'.CM5'.CM6'.T3')
         /\ witness(O,ST,o_st_yjj,Yjj')
      /\ witness(O,U,o_u_yjj, Yjj')
      /\ request(ST,O,st_o_qss,Qss')

end role
```

FIGURE 9. Role of the owner.

4) REPLAY ATTACK AND MAN-IN-THE-MIDDLE ATTACK

An adversary can obtain the messages transmitted over an insecure channel among U_i, ST_s and O_j to reuse in the authentication process. However, the transmitted messages contain a timestamp that is verified by the receiver for freshness and random numbers. Furthermore, the adversary cannot obtain the random numbers. Hence, the proposed scheme is secure to replay attacks and man-in-the-middle attacks.

5) USER ANONYMITY

In the authentication phase of the proposed scheme, all the participant entities use a pseudo-identity to replace a real identity. Assume that an adversary can extract the information stored in the mobile device and intercept the transmitted messages through a public channel. However, the adversary cannot obtain the real identity of the legitimate user because the transmitted pseudo-identity is protected by random numbers x_i, XOR, and hash operations. Even if the adversary acquires the pseudo-identity, it cannot be calculated as a real identity, which is protected by the private key sk_{TA} and random number r_i. Therefore, the proposed scheme ensures the user's anonymity.

6) CONFIDENTIALITY AND INTEGRITY

An adversary can obtain the messages transmitted over an insecure channel among U_i, ST_s and O_j to obtain sensitive information, such as the user identity ID_i and access code $\{code\}$. However, this sensitive information is encrypted using ECDH keys, so the adversary needs to calculate the ECDH keys to extract information. For example, the access code $\{code\}$ is hidden in $CM_1 = \{code\} \oplus h_1(X_i^* \cdot y_j)$. To extract the $\{code\}$ from CM_1, the adversary must compute $X_i^* \cdot y_j = (x_i \cdot y_j) \cdot P$ from $X_i^* = x_i \cdot P$ and $Y_j = y_j \cdot P$. By the ECDDHP described in Section III-B, the adversary cannot calculate the ECDH key. Furthermore, the integrity of the received messages is checked using a hash function. Therefore, the protocol provides confidentiality and integrity.

7) MUTUAL AUTHENTICATION

In the authentication phase, U_i authenticates O_j by verifying CM_1, CM_2 and authenticates ST_s by verifying SM_3. ST_s conducts $Auth_{u_i} \cdot P \stackrel{?}{=} X_i + h_2(PID_i^* || X_i || h_1(request) || T_1) \cdot$

```
role authority(U,O,TA,ST : agent, SKuta, SKota, SKust, SKost : symmetric_key,
H1,H2: hash_func, SND, RCV : channel(dy))

played_by TA
def=
local State: nat,
   MUL, ADD : hash_func,
   IDi,PWi,SKi,Aii, Ai,Li, Rii, PIDi, PKi, P, Ri, Zi, PKta, SKta, HPWi, Bi, Ci, Di, Ei,
HPIDi :text,
   SKj, IDj, Lj, PKj, Rjj, CIDj, Rj, Zj, PKs, SKs, INfo : text,
   HRi, HRj, Xii, Xi, CPIDi, AUTHui, REQuest, Qss, Qs, SM1, AUTHs, SIDs, T1, T2 :
text,
   Yjj, Yj, CODE, CM1, CM2, CM5, CM6, SM2, SM3, TID, T3, T4 : text
const sp1, sp2, sp3, sp4, sp5, sp6, sp7, sp8, sp9,
u_st_xi,o_u_yjj,st_u_yjj,o_st_yjj,st_o_qss: protocol_id
init State := 0
transition


1. State = 0 /\ RCV({IDi.Li.MUL(SKi.P)}_SKuta) =|>
State' := 1 /\ Rii' := new() /\ PIDi' := xor(IDi, H1(MUL(Rii.PKta)))
      /\ Ri' := MUL(Rii'.P) /\ Zi' := ADD(Rii'.MUL(H2(PIDi'.Ri'.MUL(SKi.P)).SKta))
      /\ SND({Zi'.PIDi'.Ri'}_SKuta)
      /\ secret({SKta, Rii'},sp2,{TA})
      /\ secret({IDi}, sp3, {U,TA})
      /\ secret({PIDi'}, sp4, {U,TA,ST})

2. State = 1 /\ RCV({IDj.Lj.MUL(SKj.P)}_SKota) =|>
State' := 2 /\ Rjj' := new() /\ CIDj' := xor(IDj,H1(MUL(Rjj'.PKta)))
      /\ Rj' := MUL(Rjj'.P)
      /\ Zj' := ADD(Rj'.MUL(H2(CIDj'.Rj'.MUL(SKj.P)).SKta))
      /\ SND({Zj'.CIDj'.Rj'}_SKota)
      /\ secret({Rjj'},sp6,{TA})
      /\ secret({{CIDj'},sp7,{O,TA,ST})

end role
```

FIGURE 10. Role of the trust authority.

```
role session(U,O,TA,ST : agent, SKuta, SKota, SKust, SKost : symmetric_key, H1,H2:
hash_func)

def=
local SN1, SN2, SN3, SN4, RV1, RV2, RV3, RV4 : channel(dy)
composition
user(U, O, TA, ST, SKuta, SKota, SKust, SKost, H1, H2, SN1, RV1)
/\ owner(U, O, TA, ST, SKuta, SKota, SKust, SKost, H1, H2, SN2, RV2)
/\ authority(U, O, TA, ST, SKuta, SKota, SKust, SKost, H1, H2,SN3, RV3)
/\ station(U, O, TA, ST, SKuta, SKota, SKust, SKost, H1, H2, SN4, RV4)
end role

role environment()
def=
const u,o,ta,st : agent,
skuta, skota, skust, skost: symmetric_key,
h1,h2,mul,add: hash_func,
pki,pkj,pks,xi,qs,yj,idi,idj,sids: text,
p_mc_m1, mc_p_bj: protocol_id,
sp1,sp2,sp3,sp4: protocol_id

intruder_knowledge = {u,o,ta,st,pki,pkj,pks,xi,qs,yj,idi,idj,sids,h1,h2}
composition
session(u,o,ta,st, skuta,skota,skust,skost, h1,h2) /\ session(i,o,ta,st,
skuta,skota,skust,skost, h1,h2)
/\session(u,i,ta,st, skuta,skota,skust,skost, h1,h2) /\ session(u,o,i,st,
skuta,skota,skust,skost, h1,h2)
/\session(u,o,ta,i, skuta,skota,skust,skost, h1,h2)

end role

goal
secrecy_of sp1, sp2, sp3, sp4, sp5, sp6, sp7, sp8, sp9
authentication_on u_st_xi,o_u_yjj,st_u_yjj,o_st_yjj,st_o_qss
end goal

environment()
```

FIGURE 11. Role of session, environment and goals.

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

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

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 3 states
  Reachable  : 0 states
  Translation: 1.91 seconds
  Computation: 0.00 seconds
```

```
% OFMC

% Version of 2006/02/13

SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/car.if

GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 130.21s
  visitedNodes: 3168 nodes
  depth: 12 plies
```

FIGURE 12. AVISPA simulation results using CL-AtSe and OFMC backends.

PK_i to authenticate U_i and verifies CM_5 to authenticate O_j. O_j performs $Auth_s \cdot P \stackrel{?}{=} Q_s + h_2(X_i^*||SID||Q_s||T_2) \cdot PK_s$ to authenticate U_i and ST_s. U_i, ST_s, and O_j authenticate each other. Therefore, the proposed scheme provides mutual authentication.

VII. PERFORMANCE ANALYSIS

This section evaluates the efficiency of the proposed scheme and compares the results with a related scheme, such as Wang *et al.* [20], Xiong *et al.* [21], and Wang *et al.* [22]. The authentication phase is more frequent than other phases, so only the authentication phase was compared. The proposed scheme was compared with those of Wang *et al.* [20], Xiong *et al.* [21], and Wang *et al.* [22]because these schemes perform authentication using blockchain and similar cryptography. This comparison shows that the proposed scheme is appropriate for practical car-sharing system because it considers three party authentication, including user, station and car owner.

A. COMPUTATION ANALYSIS

The computation cost of the proposed scheme was compared with the related schemes [20]–[22]. The existing experimental result shown in [42] was used to measure the computation cost of each cryptographic operation. T_{ea}, T_{em}, T_{hash}, T_{mac} are defined as the execution time of ''point addition'', ''point multiplication'', ''hash function'', and ''MAC function'', respectively, where $T_{ea} \approx 0.081$ ms, $T_{em} \approx 13.405$ ms, $T_{hash} \approx 0.056$ ms, and $T_{mac} \approx 0.056$ ms. The exclusive-OR (XOR) operation was omitted because its execution time is negligible compared to other operations. Table 3 lists the computation costs of the proposed scheme and the related schemes. The total computation cost of the authentication phase in Wang *et al.* [20] was $10T_{em} + 3T_{ea} + 9T_{hash} \approx 134.797$ ms. The total computation cost of the authentication phase in the scheme reported by Xiong *et al.* [21] was $8T_{em} + T_{ea} + 9T_{hash} + 2T_{mac} \approx 107.937$ ms. The total computation cost of the authentication phase by Wang *et al.* [22] was $14T_{em} + 5T_{ea} + 10T_{hash} \approx 188.635$ ms. The user, station, and owner in the proposed scheme requires $3T_{em} + 13T_{hash} \approx 40.943$ms, $7T_{em} + T_{ea} + 9T_{hash} \approx 94.42$ms, and $7T_{em} + T_{ea} + 9T_{hash} \approx 94.42$ms, respectively. Comparative analysis of the computation for the user shows that the proposed authentication scheme is similar to Xiong *et al.* [21] and more efficient than Wang *et al.* [20] and Wang *et al.* [22]. A comparison of the computation cost on the station/server-side shows that the proposed scheme is slightly less than Wang *et al.* [22]. However, the computation cost was higher

TABLE 3. Computation costs for each authentication: A comparative summary.

Schemes	User	Station/Server	Owner
Wang *et al.* [20]	$4T_{em}$ + T_{ea} + $4T_{hash}$ ≈ 53.925ms	$6T_{em}$ + $2T_{ea}$ + $5T_{hash}$ ≈ 80.872ms	−
Xiong *et al.* [21]	$3T_{em}$ + $6T_{hash}$ + T_{mac} ≈ 40.607ms	$5T_{em}$ + T_{ea} + $3T_{hash}$ + T_{mac} ≈ 67.33ms	−
Wang *et al.* [22]	$6T_{em}$ + $2T_{ea}$ + $5T_{hash}$ ≈ 80.872ms	$8T_{em}$ + $3T_{ea}$ + $5T_{hash}$ ≈ 107.763ms	−
Our proposed	$3T_{em}$ + $13T_{hash}$ ≈ 40.943ms	$7T_{em}$ + T_{ea} + $9T_{hash}$ ≈ 94.42ms	$7T_{em}$ + T_{ea} + $9T_{hash}$ ≈ 94.42ms

TABLE 4. Communication costs for each authentication: A comparative summary.

Schemes	No. of messages	Communication costs
Wang *et al.* [20]	2	1472 bits
Xiong *et al.* [21]	2	1184 bits
Wang *et al.* [22]	2	1184 bits
Our proposed	4	2336 bits

than Wang *et al.* [20] and Xiong *et al.* [21] because the station authenticates the user and the owner. Overall, the proposed scheme also has certain advantages in energy consumption on the user side, which is more suitable to the user side with limited re-sources and computing power.

B. COMMUNICATION ANALYSIS

The communication cost of the proposed scheme was compared with the related schemes [20]–[22]. According to [42], it was assumed that the bit length of the identity, the hash output, the random number, the timestamp, and the elliptic curve point were 160 bits, 160 bits, 160 bits, 32 bits, and 320 bits, respectively. The bit length of the user's request information was assumed to be 160 bits. Table 4 lists the communication costs of the proposed scheme and related schemes. In Wang *et al.* [20], the communication cost of the authentication phase between the user and server was 1472 bits as $\{T, X, CT\}$ and $\{Y, V\}$. The communication cost of Xiong *et al.* [21] was 1184 bits as $\{A, pid_i, k, t_i\}$ and $\{B, w, t_j\}$. The communication cost of Wang *et al.* [22] was 1184 bits as $\{A, W_U, \sigma, T_U\}$ and $\{B, w_1, T_1\}$. In the proposed authentication phase, the exchanged messages $\{X_i, CPID_i, Auth_{u_i}, T_1, request\}$, $\{CM_1, CM_2, SM_2, SM_3, T_3, T_4, Y_j\}$ between U_i and ST_s require $(320 + 160 + 160 + 32 + 160) = 832$ bits and $(160 + 160 + 160 + 160 + 32 + 32 + 320) = 1024$ bits. Similarly, the communication cost for the exchanged messages $\{Q_s, SID_s, Auth_s, T_2, SM_1\}$, $\{Y_j, CM_1, CM_2, CM_3, CM_4, CM_5, CM_6, T_3\}$ among ST_s and O_j were $(320 + 160 + 160 + 32 + 160) = 832$ bits and $(320+160+160+160+160+160+160+32) = 1312$ bits. The total communication cost of the proposed scheme was high compared to the related schemes because an authentication phase was performed by the three parties for their car-sharing service. However, in the personal car-sharing system, people can lend their car to others and rent another personal car. A service vendor supports the process of car-sharing service for user convenience during car-sharing. However, the existing blockchain-based authentication schemes are unsuitable for a car-sharing system because these schemes are designed for the user and servers. Hence, the car owner cannot be considered in the existing authentication schemes. On the other hand, car owners can convert car use easily and reject the service request. Therefore, this study designed the blockchain-based authentication scheme for three entities in the car-sharing system.

VIII. CONCLUSION

Car-sharing systems have attracted widespread attention as an approach that alleviates the transportation problems in urban areas. However, the traditional car-sharing system is exposed to some security problems owing to the centralized system structure and communication via a public channel. This paper proposed a secure decentralized model of a car-sharing system and a secure authentication scheme to provide a decentralized sharing service for legitimate users. Blockchain was used to ensure the integrity of information of service information and provide a decentralized car-sharing service. Furthermore, a pseudonym of the user was applied in the car-sharing system to guarantee user's privacy. Thus, if the stored information is exposed to an adversary, they cannot know the user's real identity. BAN logic analysis was performed to show that the proposed protocol can provide secure mutual authentication between the user, station, and owner. In addition, the AVISPA simulation was employed to show that the proposed protocol is secure against replay and man-in-the-middle attacks. Moreover, the proposed protocol is secure against impersonation, stolen mobile devices, offline password guessing, replay, and man-in-the-middle attacks. The proposed protocol provides anonymity, confidentiality, and mutual authentication by conducting informal security analysis. The performance of the proposed protocol was compared with related schemes. The proposed protocol is efficient and can be applied in the blockchain-based car-sharing system using blockchain. In the future, a simulation will be developed to test the protocol and apply the protocol to a real car-sharing system.

REFERENCES

[1] J. Jung and Y. Koo, "Analyzing the effects of car sharing services on the reduction of greenhouse gas (GHG) emissions," *Sustainability*, vol. 10, no. 2, p. 539, Feb. 2018.

[2] F. Ferrero, G. Perboli, M. Rosano, and A. Vesco, "Car-sharing services: An annotated review," *Sustain. Cities Soc.*, vol. 37, pp. 501–518, Feb. 2018.

[3] P. W. Wadhwani and P. Saha, "Car sharing market size by model (P2P, station-based, free-floating), by business model (round trip, one way), by application (business, private), industry analysis report, regional outlook, application potential, price trend, competitive market share & forecast, 2020–2026," Global Market Insights, Pune, India, Tech. Rep., Apr. 2020. [Online]. Available: https://www.gminsights.com/industry-analysis/carsharing-market

[4] K. Münzel, L. Piscicelli, W. Boon, and K. Frenken, "Different business models–different users? Uncovering the motives and characteristics of business-to-consumer and peer-to-peer carsharing adopters in The Netherlands," *Transp. Res. D, Transp. Environ.*, vol. 73, pp. 276–306, Aug. 2019.

[5] G. H. D. A. Correia, D. R. Jorge, and D. M. Antunes, "The added value of accounting for users' flexibility and information on the potential of a station-based one-way car-sharing system: An application in Lisbon, Portugal," *J. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 299–308, Jul. 2014.

[6] S. Shaheen, N. Chan, A. Bansal, and A. Cohen, "Shared mobility: A sustainability & technologies workshop: Definitions, industry developments, and early understanding," Tech. Rep., 2015.

[7] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and C. Yang, "The blockchain as a decentralized security framework [future directions]," *IEEE Consum. Electron. Mag.*, vol. 7, no. 2, pp. 18–21, Mar. 2018.

[8] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "BlockChain: A distributed solution to automotive security and privacy," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 119–125, Dec. 2017.

[9] M. Kim, K. Park, S. Yu, J. Lee, Y. Park, S. Lee, and B. Chung, "A secure charging system for electric vehicles based on blockchain," *Sensors*, vol. 19, iss. 13, no. 3028, pp. 1–22, Jul. 2019.

[10] G. Rathee, A. Sharma, R. Iqbal, M. Aloqaily, N. Jaglan, and R. Kumar, "A blockchain framework for securing connected and autonomous vehicles," *Sensors*, vol. 19, no. 14, pp. 1–15, 2019.

[11] L. Tseng, L. Wong, S. Otoum, M. Aloqaily, and J. B. Othman, "Blockchain for managing heterogeneous Internet of Things: A perspective architecture," *IEEE Netw.*, vol. 34, no. 1, pp. 16–23, Jan. 2020.

[12] B. Vaidya and H. T. Mouftah, "Security for shared electric and automated mobility services in smart cities," *IEEE Secur. Privacy*, vol. 19, no. 1, pp. 24–33, Feb. 2021.

[13] I. Symeonidis, M. A. Mustafa, and B. Preneel, "Keyless car sharing system: A security and privacy analysis," in *Proc. IEEE Int. Smart Cities Conf. (ISC2)*, Seattle, WA, USA, Sep. 2016, pp. 1–7.

[14] C. Busold, A. Taha, C. Wachsmann, A. Dmitrienko, H. Seudié, M. Sobhani, and A.-R. Sadeghi, "Smart keys for cyber-cars: Secure smartphone-based NFC-enabled car immobilizer," in *Proc. 3rd ACM Conf. Data Appl. Secur. Privacy*, 2013, pp. 233–242.

[15] R. E. Haas and D. P. F. Moller, "Automotive connectivity, cyber attack scenarios and automotive cyber security," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT)*, May 2017, pp. 635–639.

[16] Z. Wei, Y. Yanjiang, Y. Wu, J. Weng, and R. H. Deng, "HIBS-KSharing: Hierarchical identity-based signature key sharing for automotive," *IEEE Access*, vol. 5, pp. 16314–16323, 2017.

[17] M. Laurent, J. Leneutre, S. Chabridon, and I. Laaouane, "Authenticated and privacy-preserving consent management in the Internet of Things," *Procedia Comput. Sci.*, vol. 151, pp. 256–263, Dec. 2019.

[18] S. H. Park, J. H. Kim, and M. S. Jun, "A design of secure authentication method with bio-information in the car sharing environment," in *Advances in Computer Science and Ubiquitous Computing*, J. J. Park, Y. Pan, G. Yi, and V. Loia, Eds. Singapore: Springer, 2017, pp. 205–210.

[19] A. Dmitrienko and C. Plappert, "Secure free-floating car sharing for offline cars," in *Proc. ACM Conf. Data Appl. Secur. Privacy*, 2017, pp. 349–360.

[20] J. Wang, L. Wu, K.-K.-R. Choo, and D. He, "Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1984–1992, Mar. 2020.

[21] L. Xiong, F. Li, S. Zeng, T. Peng, and Z. Liu, "A blockchain-based privacy-awareness authentication scheme with efficient revocation for multi-server architectures," *IEEE Access*, vol. 7, pp. 125840–125853, Sep. 2019.

[22] W. Wang, H. Huang, L. Xue, Q. Li, R. Malekian, and Y. Zhang, "Blockchain-assisted handover authentication for intelligent telehealth in multi-server edge computing environment," *J. Syst. Archit.*, vol. 115, May 2021, Art. no. 102024.

[23] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data*, Honolulu, HI, USA, Jun. 2017, pp. 557–564.

[24] J. Xie, H. Tang, T. Huang, F. R. Yu, R. Xie, J. Liu, and Y. Liu, "A survey of blockchain technology applied to smart cities: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2794–2830, 3rd Quart., 2019.

[25] S. Son, J. Lee, M. Kim, S. Yu, A. K. Das, and Y. Park, "Design of secure authentication protocol for cloud-assisted telecare medical information system using blockchain," *IEEE Access*, vol. 8, pp. 192177–192191, Oct. 2020.

[26] J. Cui, D. Wu, J. Zhang, Y. Xu, and H. Zhong, "An efficient authentication scheme based on semi-trusted authority in VANETs," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2972–2986, Mar. 2019.

[27] D. Chattaraj, M. Sarma, A. K. Das, N. Kumar, J. J. P. C. Rodrigues, and Y. Park, "HEAP: An efficient and fault-tolerant authentication and key exchange protocol for Hadoop-assisted big data platform," *IEEE Access*, vol. 6, pp. 75342–75382, 2018.

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

[29] S. Mandal, B. Bera, A. K. Sutrala, A. K. Das, K.-K.-R. Choo, and Y. Park, "Certificateless-signcryption-based three-factor user access control scheme for IoT environment," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3184–3197, Apr. 2020.

[30] M. Kim, S. Yu, J. Lee, Y. Park, and Y. Park, "Design of secure protocol for cloud-assisted electronic health record system using blockchain," *Sensors*, vol. 20, iss. 10, no. 2913, pp. 1–21, May 2020.

[31] J. Lee, S. Yu, M. Kim, Y. Park, S. Lee, and B. Chung, "Secure key agreement and authentication protocol for message confirmation in vehicular cloud computing," *Appl. Sci.*, vol. 10, no. 18, p. 6268, Sep. 2020.

[32] A. Kumari, V. Kumar, M. Y. Abbasi, S. Kumari, P. Chaudhary, and C.-M. Chen, "CSEF: Cloud-based secure and efficient framework for smart medical system using ECC," *IEEE Access*, vol. 8, pp. 107838–107852, 2020.

[33] S. Yu, K. Park, Y. Park, H. Kim, and Y. Park, "A lightweight three-factor authentication protocol for digital rights management system," *Peer Netw. Appl.*, vol. 13, no. 5, pp. 1340–1356, Sep. 2020.

[34] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990.

[35] AVISPA. (2020). *Automated Validation Internet Security Protocols Application*. Accessed: Mar. 2021. [Online]. Available: http://www.avispa-project.org/

[36] AVISPA. *SPAN, A Security Protocol ANimator for AVISPA*. Accessed: Mar. 2021. [Online]. Available: http://www.avispa-project.org/

[37] R. Shashidhara, S. K. Nayak, A. K. Das, and Y. Park, "On the design of lightweight and secure mutual authentication system for global roaming in resource-limited mobility networks," *IEEE Access*, vol. 9, pp. 12879–12895, 2021.

[38] D. K. Kwon, S. J. Yu, J. Y. Lee, S. H. Son, and Y. H. Park, "WSN-SLAP: Secure and lightweight mutual authentication protocol for wireless sensor networks," *Sensors*, vol. 21, no. 3, p. 936, Jan. 2021.

[39] J. Oh, S. Yu, J. Lee, S. Son, M. Kim, and Y. Park, "A secure and lightweight authentication protocol for IoT-based smart homes," *Sensors*, vol. 21, no. 4, p. 1488, Feb. 2021.

[40] Z. Ali, A. Ghani, I. Khan, S. A. Chaudhry, S. H. Islam, and D. Giri, "A robust authentication and access control protocol for securing wireless healthcare sensor networks," *J. Inf. Secur. Appl.*, vol. 52, Jun. 2020, Art. no. 102502.

[41] D. Von Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," in *Proc. 3rd APPSEM II Workshop Appl. Semantics (APPSEM)*, Frauenchiemsee, Germany, 2005, pp. 1–17.

[42] M. Wazid, P. Bagga, A. K. Das, S. Shetty, J. J. P. C. Rodrigues, and Y. H. Park, "AKM-IoV: Authenticated key management protocol in fog computing-based Internet of vehicles deployment," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8804–8817, Oct. 2019.

MYEONGHYUN KIM received the B.S. and M.S. degrees in electronics engineering from Kyungpook National University, Daegu, South Korea, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree with the School of Electronic and Electrical Engineering. His research interests include authentication, blockchain, the Internet of Things, and information security.

JOONYOUNG LEE received the B.S. and M.S. degrees in electronics engineering from Kyungpook National University, Daegu, South Korea, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree with the School of Electronic and Electrical Engineering. His research interests include authentication, the Internet of Things, and information security.

KISUNG PARK received the B.S. and M.S. degrees in electronics engineering from Kyungpook National University, Daegu, South Korea, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree with the School of Electronic and Electrical Engineering. He is currently a Researcher with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. His research interests include authentication, blockchain, anonymous credentials, decentralized identifier, the Internet of Things, post-quantum cryptography, VANET, and information security.

YOHAN PARK received the B.S., M.S., and Ph.D. degrees in electronic engineering from Kyungpook National University, Daegu, South Korea, in 2006, 2008, and 2013, respectively. He is currently an Assistant Professor with the Department of Computer Engineering, College of Engineering, Keimyung University, Daegu. His research interests include computer networks, mobile security, blockchain, and information security.

KIL HOUM PARK received the B.S. degree in electronics engineering from Kyungpook National University, Daegu, South Korea, in 1982, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 1984 and 1990, respectively. He is currently a full-time Professor with Kyungpook National University. His research interests include computer vision, image processing, electrocardiogram signal processing, and signal compression.

YOUNGHO PARK (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic engineering, Kyungpook National University, Daegu, South Korea, in 1989, 1991, and 1995, respectively. From 1996 to 2008, he was a Professor with the School of Electronics and Electrical Engineering, Sangju National University, South Korea. From 2003 to 2004, he was a Visiting Scholar with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA. He is currently a Professor with the School of Electronic and Electrical Engineering, Kyungpook National University. His research interests include computer networks, multimedia, and information security.