Privacy-Preserving ECC-Based Three-Factor Authentication Protocol for Smart Remote Vehicle Control System

Hongwei Luo[1,2(✉)], Qinyao Zhang[1], and Guoai Xu[1,2]

[1] Institute of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China
kevin.lhw@antfin.com

[2] National Engineering Laboratory of Mobile Network Security, Beijing 100876, China

Abstract. With the rapid development of auto industry and the Internet, smart remote vehicle control (SRVC) system is now showing in more and more automobile brands which allows people to remotely control vehicles through the Internet. However, SVRC's convenience brings security challenges that the related SRVC's protocol needs to be enhanced with identity authentication mechanism for both users and their vehicles, in case of illegal intrusion without identity authentication. In this paper, we first analyze Chatterjee et al.'s scheme and find that their scheme is not immune from some common attacks. Then, we design a privacy-preserving elliptic curve cryptosystem (ECC)-based three-factor authentication scheme based on the SRVC's features, which can authenticate the identities of users and vehicle, and generate a session secret key to protect the users' privacy. Security analysis shows that our protocol has many security attributes that it not only can protect users' anonymity and untraceability, but also resist many known attacks. And performance analysis shows that our protocol can run efficiently in SRVC system. Lastly, we conclude our work and give the future research direction in SVRC.

Keywords: Three-factor authentication · Privacy-preserving · Smart remote vehicle control (SRVC) · Elliptic curve cryptosystem (ECC)

1 Introduction

In the past few decades, with the upgrading of automobile control systems, the human-vehicle interaction mode has been constantly changing [1]. In the beginning, people used traditional physical car keys to open the doors and start the car. Later, car manufacturers embedded electronic control devices in the car, which enables people to control the car doors remotely by pressing the button on radio car keys.

The remote-control function of radio car keys is mainly realized by the embedded RFID chip. In the communication process between the radio key and the car electronic control device, generally, a symmetric encryption algorithm (such as AES) is used to protect the confidentiality of the message. But the work in [2] shows that these radio

B. Chen and X. Huang (Eds.): AC3 2021, LNICST 386, pp. 56–72, 2021.
https://doi.org/10.1007/978-3-030-80851-8_5

car keys are subject to relay attacks. The attacker can actively or passively control the car from a long distance through the relay device. What's more, once the long-term symmetric secret key stored in the radio key is acquired, the vehicle control right will easily be changed.

Furthermore, with the development of the Internet of Things (IoT) and the Internet of Vehicles (IoV), smart remote vehicle control (SRVC) systems are gradually appearing in much more cars. This charming system allows people to use smart devices to control the vehicle. So, users can not only send commands of opening and closing the door or starting the car through the network, but also read the real-time data of the vehicle and perform multiple operations such as controlling windows and air conditioning.

However, the convenience of remote network communication brings many challenges to the SRVC system. Since the openness of the network, attackers can initiate a variety of attacks, such as interception, modification and replay of network communication messages. Therefore, it is necessary to propose an authentication protocol, as a first line to defend the attacks talked before, for the SRVC system to realize mutual authentication and protect communication data and user privacy.

1.1 Threat Model

By the Delov-Yao adversary model [19], we suppose that the adversary can intercept, modify, delete and replay all messages transmitted via insecure public channels in the protocol. The adversary may have the ability to obtain the data stored in the smart-card or in the car enterprise servers. Besides, the insider attacker can also perform an insider attack to obtain confidential information from the enterprise database.

1.2 Related Works

In this part, we provide a review of authentication and key agreement schemes related to the SRVC system.

Since Lamport [3] proposed the first password authentication scheme, a large number of schemes have been presented to problems of different network communication models based on many cryptography primitives. Chen et al. [5] used ECC earlier in their dynamic ID-based remote mutual authentication scheme for cloud computing. Then, to get a better security property, taking the advantage of Elliptic Curve Diffie-Hellman Problem (ECDHP), many mutual authentication schemes for IoT used ECC secret keys as participants's identity certificates. In 2013 and 2014 Lai et al. [9, 10] proposed two authentication and key agreements of IoT using ECC, which provided group authentication. However, their protocols lacked location and identity privacy considerations. Chu et al. [8] offered their identity authentication scheme for IoT, which could resist several attacks in 2013. But users in their scheme need to authenticate many times in multi-server environment. Interestingly, Poranbage et al. [7] presented a two-phase authentication protocol distributed IoT applications using ECC. Their scheme is able to resist malicious users and Denial of Service (DoS) attacks, but not efficient and cannot resist node capturing attack. As an improved work, Truong et al. [15] overcame Lee et al.'s [16] limitations and proposed a dynamic ID-based user authentication scheme. And then Kumari et al. [4] analyzed Truong's scheme and found that Truong's scheme

suffers from password guessing attack and no forward secrecy. For different design scenario, Feng et al. [17] introduced a three-factor mobile multi-server authentication scheme using ECC. Alotaibi et al. [6] proposed a three-factor authentication for WSN using ECC in 2018 which can resist clone and fork attacks. Mo, Jiaqing, et al. [12] proposed a ECC-based two-factor user authentication protocol for the mobile cloud computing environment, which has reasonable computation cost and communication overhead at the mobile client side as well as the server side. Based on a comprehensive threat model, Aghili et al. [18] presented a lightweight scheme for E-Health Systems in IoT with three-factor authentication access control and ownership transfer in 2019. In 2020, Chatterjee et al. [11] proposed a robust lightweight ECC-based three-way authentication scheme for IoT in cloud. Panda et al. [13] also introduced a mutual authentication protocol for IoT environment which has relatively good security performance.

1.3 Motivation and Contribution

The existing SRVC protocol is mainly designed by automobile manufacturers and written into the intelligent vehicle control App and the in-car smart terminal. Given the non-openness of the car manufactures' code on authentication, by the only effective way of comparing the IoT model with the SRVC model, we find that SRVC system has some unique features listed below as our first contribution.

(1) The participants of protocols in SRVC system are smart device, car enterprise server and in-car smart terminal.
(2) A smart device has to communicate with the car directly for the need of privacy protection, so there shall be session key exchange phase after mutual authentication phase in a SRVC protocol.
(3) Different from the GWN and sensors in IOT protocols, the enterprise server and in-car smart terminals have stronger computing power and more stable network connection, which means more computation can be done and longer message can be transmitted to ensure the safety of protocol.
(4) The key agreement in common IoT communication is sensitive to time cost, but the SRVC is different. People open applications on smart devices and connect to the vehicles. The key agreement phase is completed at this stage. Due to a large number of computing requirements, most applications are slow when they are opened. Therefore, time performance i in the SRVC key agreement phase is not as important as in common IoT systems.
(5) Compared to IOT protocols, the security and privacy protection capabilities of SRVC protocols are more important for they are closely related to the user's life safety.

By the use of these features shown above, our second contribution is that the mutual authentication and session key agreement protocol are given special for SRVC. Specifically:

- We prove that Chatterjee et al.'s ECC-based three-way scheme of IoT [11] suffers from several attacks.

- We put forward a novel authentication protocol for SRVC system using ECC. The proposed protocol satisfies various security attributes and can also prevent the disclosure of important privacy of users.
- We conducted a security analysis of the protocol, which demonstrates that the proposed scheme can resist known attacks.
- The performance evaluation shows that our scheme can run efficiently on SRVC system.

1.4 Notations

The notations are introduced in Table 1.

Table 1. Notation of the paper

Symbols	Description
U, S, T	User, car enterprise server and smart terminal of the car
UID, SID, TID, DID	IDs of U, S, T, and the IoT node
k_U, k_{US}, k_{ST}, k_T	Public keys computed by U, S, T
d_U, d_{US}, d_{ST}, d_T	Private keys selected by U, S, T
k_D, k_N, k_G	Public keys of smart device, IoT node and gateway server in Chatterjee et al.'s scheme
d_D, d_N, d_G	Private keys of smart device, IoT node and gateway server in Chatterjee et al.'s scheme
h	One-way hash function
h_B	Bio hash function
PW, B	U's password and biometric
R_{Zp}	Random numbers chosen from the field Z_p
K_{sess}	Session key
E_p	A p-order elliptic curve group defined over $GF(q)$
G	Generate of E_p

1.5 Paper Organization

The rest of the paper is organized as follows. The review of Chatterjee et al.'s scheme is presented in Sect. 2. In Sect. 3, we analyze Chatterjee et al.'s Scheme and prove that it has security limitations. Our protocol is proposed in Sect. 4. The security analysis is discussed in Sect. 5. Section 6 shows the performance evaluation. Finally, concluding remarks and future research are discussed in Sect. 7.

2 Review of Chatterjee et al.'s Scheme

2.1 System Initialization and Registration Phase

This phase includes registration process of the user, the smart device and the IoT node.

1. User registration with smart device: User chooses his identity UID, password PW and biometric B. Then smart device stores them in the form of hashed values.
2. Smart device registration with gateway server: Smart device sends ID and publishes its public key k_U with certificate. Upon receiving the message, gateway server stores UID and publish its public key k_G.
3. IoT device registration with gateway server: IoT device sends its identity and public key certificate to the gateway server. Then the gateway server stores its ID and sends its certificate back.

2.2 Authentication and Key Agreement Phase

In this phase, the smart device, the IoT node and the gateway server authenticate each other over a public channel.

Firstly, the user inputs his ID, password and biometric into the smart device and the smart device computes hashed value to authenticate the user.

For the purpose of login, smart device first generates a nonce n_i. Then it computes $d_U \cdot k_G$ and $h(DID \oplus n_i \oplus d_U \cdot k_G)$ and sends $(DID, n_i, h(DID \oplus n_i \oplus d_U \cdot k_G))$ to the IoT node.

Upon receiving login request from the smart device, it generates a random number $p \in R_{Zp}$ and computes $h(NID \oplus d_N \cdot k_G \oplus n_i), p \oplus h(d_N \cdot k_G)$ and sends it with the login request to the IoT gateway.

The gateway server first computes $d_G \cdot k_D$, and checks if $h(DID \oplus n_i \oplus d_U \cdot k_G) = h(DID \oplus n_i \oplus d_G \cdot k_U)$, $h(NID \oplus d_N \cdot k_G \oplus n_i) = h(NID \oplus d_G \cdot k_U \oplus n_i)$. If so, the gateway server can authenticate the smart device and the IoT node. Then the server generates a nonce n_j, computes $p = p \oplus h(d_N \cdot k_G) \oplus h(d_G \cdot k_N), h\big(d_G \cdot K_D \oplus n_j\big)$, and sends $\big(n_j,\ h\big(d_G \cdot K_D \oplus n_j\big),\ p \oplus h(d_G \cdot k_D)\big)$ to the smart device.

The smart device first checks if $h\big(d_D \cdot K_G \oplus n_j\big) = h\big(d_G \cdot K_D \oplus n_j\big)$. If they are equal, the smart device generates a random number $r \in R_{Zp}$, computes p with $p \oplus h(d_G \cdot k_D) \oplus h(d_D \cdot k_G)$ and computes $SK = r \cdot p \cdot G$. Then the smart device sends $\big(n_j,\ h\big(p \cdot r \cdot n_j\big),\ r \oplus p\big)$ to the IoT node.

The IoT node now computes r with the message from the smart device and compares $h\big(r \oplus p \oplus n_j\big)$ and check the equality to the value in the message. Finally, it computes $SK = r \cdot p \cdot G$.

3 Cryptanalysis of Chatterjee et al.'s Scheme

Based on Delov-Yao threat model, since the login authentication and key agreement phase take place on the public channel, messages transmitted can be intercepted and modified by an adversary. All data stored on the gateway server is available for the

privileged insider. In the light of the aforementioned scenario, we present the security problems of Chatterjee et al.'s scheme. The listing and the discussion of various security problems of Chatterjee et al.'s scheme are given below:

1. User impersonation attack
   Since the public and private keys are generated in advance, if the adversary is able to obtain any two private keys, he can perform a user impersonation attack. Taking d_N and d_G for instance, the adversary can easily get *DID* for it is transmitted in plaintext, then he generates a nonce n_i^*, and computes $h(NID \oplus n_i^* \oplus d_G \cdot k_D)$ and sends $(DID, n_i^*, h(NID \oplus n_i^* \oplus d_G \cdot k_D))$ to the gateway server. When the gateway server receives the message, it checks the validation of the message with $h(NID \oplus n_i^* \oplus d_G \cdot k_D)$, and obviously the message can pass the check. When the IoT Node transmits message to the adversary, he is able to compute p with $d_N \cdot k_D$. Then the adversary selects a random r and computes SK and sends r to the IoT node. Finally, a session between the adversary and the IoT node is established.
2. User tracking attack
   In login authentication phase, the smart device sends *DID* to the gateway server in plain text. Although *DID* is not directly related to the user's identity, it is fixed for a specific and not updated with each session. When monitoring the public channel, the adversary can launch a tracking attack with this fixed value. Then he might collect sensitive information related to the user like user's location, user's traveling routes and so on.
3. DoS attack
   Since none of the three participants in the scheme check the freshness of messages and responds messages with calculated values. The adversary can eavesdrop a message and resend it a large number of times, making the gateway server or the IoT node out of service.

4 The Proposed Scheme

In this section, we put forward a three-factor authentication scheme for SRVC system based on ECC. It includes three kinds of participants, i.e., the user U, the car enterprise server S and the vehicle smart terminal T. In registration phase, S is responsible for storing and issuing hashed user information to U and T. In login and authentication phase, S is responsible for implement mutual authentication and negotiate a session key between U and T.

4.1 Pre-deployment Phase

When the car enterprise produces a car, it also generates the unique *TID* of the car and stores the value in the enterprise database.

S selects E_p and G. E_p is a p-order elliptic curve group defined over Galois field $GF(q)$ where q is a prime or in the binary space 2^n and G is a generator of the group E_p. S publishes the parameters (E_p, G).

4.2 User Registration Phase

This phase is depicted as Fig. 1. Below are the details.

Step1. A user U first gets the TID of his car from the automobile enterprise. Then U randomly picks a UID, a random nonce r_U and choose a password PW. U computes $h(PW||r_U)$ and send parameters $(UID,\ h(PW||r_U),\ TID)$ via a secure channel.

Step2. Upon receiving $(UID,\ h(PW||r_U), CID)$, S picks two random numbers r_S, r_T and a nonce n_0, computes $A_0 = h(UID||h(PW||r_U)) \oplus n_0$, $C_{UT} = A \oplus h(TID)$, $h(SID||r_S), h(SID||r_T), B_{US} = A \oplus h(SID||r_S), C_{US} = h(A||(SID||r_S))$, $C_{ST} = h(h(CID)||h(SID||r_T))$ and stores $(C_{UT},\ B_{US},\ C_{US},\ C_{ST})$ to its database. Next, S saves $(C_{US},\ h(SID||r_S))$ on a smart-card and hands it to the user U. Then S sends (C_{ST}, TID) to T. Both two data transmissions are carried out via secure channels.

Step3. When U receives the smart-card, he imprints his biometric data B, computes $C_1 = h(UID||PW) \oplus h(SID||r_S)$, $C_2 = h_B(B) \oplus h(UID||h(PW||r_U))$, and replaces $h(SID||r_S)$ in smart-card with C_1 and C_2. Now smart-card contains $(C_1, C_2,\ C_{US})$.

Step4. Upon the receipt of (C_{ST}, CID), T computes $h(TID)$ and stores $(h(TID), C_{ST})$ in a secure storage space.

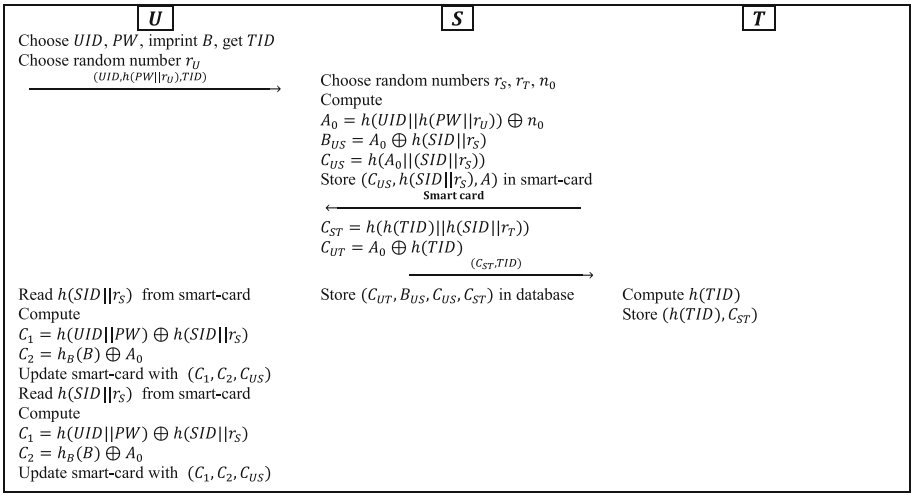

Fig. 1. Registration phase of the proposed scheme

4.3 Login and Authentication Phase

Summary of this phase is in Fig. 2 and below are details.

Step1. Firstly, U reads C_1, C_2 and Cus from the smart-card. With the input of UID', PW' and B', U computes $t_1 = C_1 \oplus h(UID'||PW')$, $t_2 = C_2 \oplus h_B(B)$. Next, U computes $C'_{sc} = h(t_2||t_1)$ and checks if $C'_{ST} = C_{ST}$. If they are equal, which means $A_i = t_2$ and $h(SID||r_S) = t_1$, U selects a random number $d_U \in {}_RZ_p$, picks a random nonce n_1 and computes $k_U = d_U \cdot G$, $D = n_1 \oplus h(SID||r_S)$, $E = h(A||h(SID||r_S)) \oplus k_U$, $V_1 = h(A||h(SID||r_S)||k_U||n_1||T_1)$ where T_1 is the current timestamp, and sends $message1 = (A,\ D,\ E, V_1, T_1)$ to S via a public channel. Otherwise U stops login.

Step2. Upon receiving the login request $message1$ from U, S first check T_1. If T_1 is fresh, U computes $t_3 = B_{US} \oplus A$, $k'_U = E \oplus h(A||t_3)$, $n'_1 = t_3 \oplus D$, $V'_1 = \left(A||t_3||k'_U\middle||n'_1\middle||T_1\right)$ and check the if $V'_1 = V_1$. Upon the equality of the two is confirmed, S picks a nonce n_2, and computes $h(TID) = C_{UT} \oplus A$, $F = h(C_{US}||n_1) \oplus h(h(TID)||C_{ST})$, $G = h(C_{ST}||n_2) \oplus h(h(TID)||C_{ST})$. *Next, S selects* $d_{ST} \in {}_RZ_p$ *and computes* $k_{ST} = d_{ST} \cdot G$, $H = h(h(C_{US}||N_1)||h(C_{ST}||n_2)) \oplus k_{ST}$ and $V_2 = h(F||G||k_{ST}||T_2)$ where T_2 is the current timestamp. Then S sends $message2 = (F, G, H, V_2, T_2)$ to T via a public channel.

Step3. When T receives $message2$ from S, T first check the freshness of T_2. If so, T computes $t_4 = F \oplus h(h(TID||C_{ST}))$, $t_5 = G \oplus h(h(TID||C_{ST}))$, $k'_{ST} = J \oplus h(t_4||t_5)$, $V'_2 = h(t_4||t_5||k'_{sc}||T_2)$ and check if $V'_2 = V_2$. If so, T selects $d_T \in {}_RZ_p$ and computes $e_{ST} = d_T \cdot k_{ST}$, $k_T = d_T \cdot G$, $I = h(h(C_{US}||n_1)||h(C_{ST}||n_2)) \oplus k_c$, $V_3 = h(I||e_{ST}||T_3)$. Then T_3 sends parameters $message3 = (I, V_3, T_3)$ back to S.

Step4. After receiving $message3$ from T and checking the freshness of T_3, S first computes $k'_T = I' \oplus h(h(C_{US}||n_1)||h(C_{ST}||n_2))$, $e'_{ST} = d_{ST} \cdot k'_T$, $V'_3 = h\left(I\middle||e'_{ST}\middle||T_3\right)$ and checks if V'_3 is equal to V_3. For the correct match, S selects $d_{US} \in {}_RZ_p$ and computes $k_{US} = d_{US} \cdot G$, $e_{US} = d_{US} \cdot k_U$. Next S picks a nonce n_3 randomly and computes $I = h(h(C_{US}||n_1)||h(C_{ST}||n_2))$, $K = n_3 \oplus I$, $L = h(n_3||I) \oplus h(k_U||e_{US}||k_{US})$, $V_4 = h(K||L||n_3||T_4)$ where T_4 is the current timestamp and sends $message4 = (K, L, V_4, T_4)$ to T via a public channel. S then computes $M = k_{US} \oplus h(SID||r_S)$, $O = h(k_U||e_{US}||k_{US}) \oplus h(k_T||e_{ST}||k_{ST})$, $Q = h(C_{US}||n_1) \oplus h(C_{ST}||n_2)$, $V5 = h(M|O||Q||e_{US}||T_5)$ and sends $message5 = (M, N, Q, V_5, T_5)$ to U via a public channel where T_5 is the current timestamp.

Step5. T receives the parameters in $message4$ from S and checks the freshness of T_4 firstly. Next, T computes $n'_3 = I \oplus K$, $h(k_U||e_{US}||k_{US}) = L \oplus h(n_3||I)$, $V'_4 = (K||L||n'_3||T_4)$ and check if $V'_4 = V_4$. If so, T computes the Session Key $K_{sess} = h(h(k_U||e_{US}||k_{US})||h(k_T||e_{ST}||k_{ST})||h(h(C_{US}|n_1)||h(C_{ST}||n_2)))$.

Step6. U first checks if T_5 is fresh upon receiving $message5$ from S. Then U compute $k'_{US} = M \oplus h(SID||r_S)$, $e'_{US} = k'_{US} \cdot d_U$, $V'_5 = h\left(M||O||Q\middle||e'_{US}\middle||T_5\right)$ and check the equality of V'_5 and V_5. If so, U computes $h(k_c||e_{sc}||k_{sc}) = O \oplus h(k_U||e_{US}||k_{US})$, $h(C_{ST}||n_2) = Q \oplus h(C_{US}||n_1)$. Finally, U computes $K_{sess} = h(h(k_U||e_{US}||k_{US})||h(k_T||e_{ST}||k_{ST})||h(h(C_{US}|n_1)||h(C_{ST}||n_2)))$.

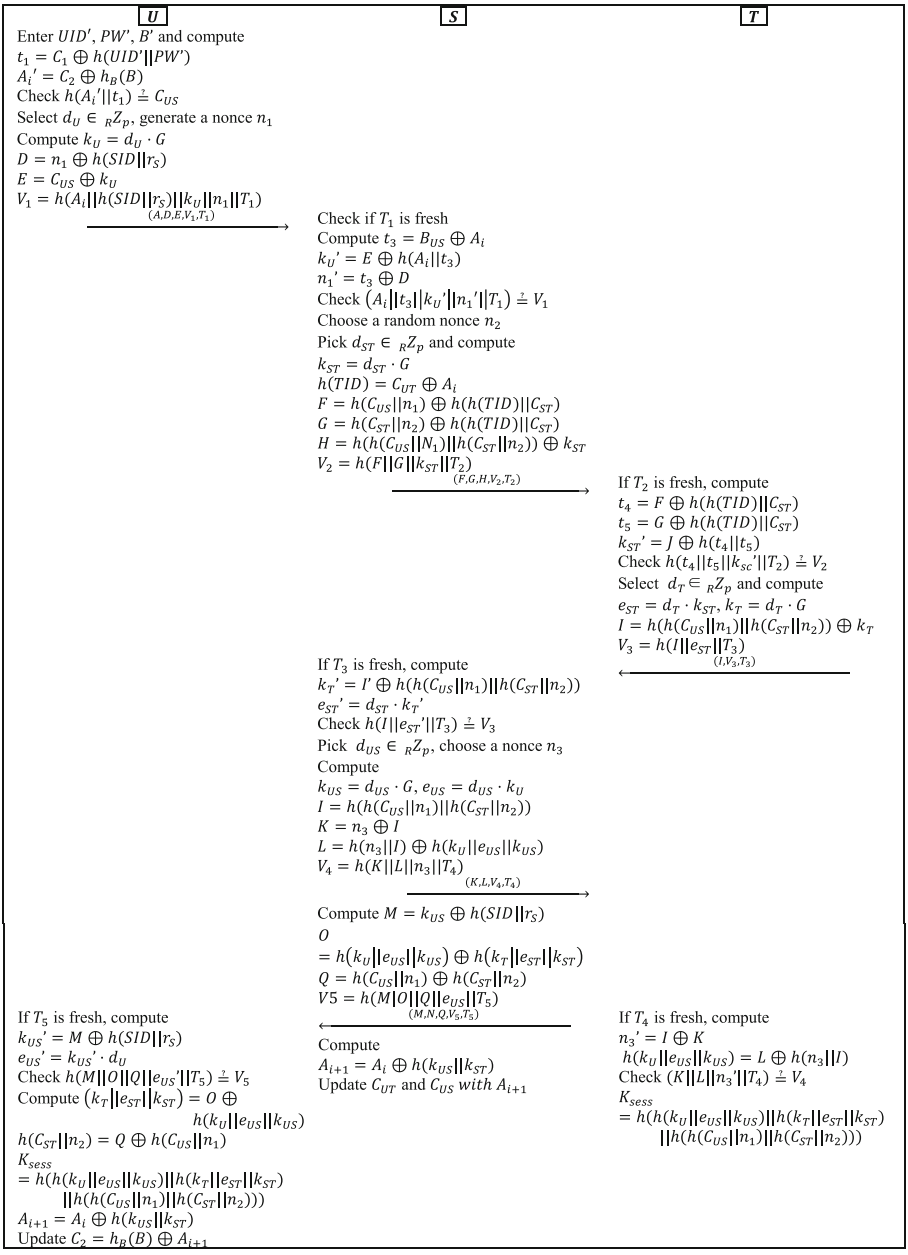

Fig. 2. The login and authentication phase of the proposed scheme

5 Security Analysis

In this section, we first show that our scheme provides mutual authentication and key agreement between the user and the car by using BAN logic. Then we demonstrate different security attributes related to our protocol.

5.1 BAN-Logic Based Proof of Authentication

In this subsection, the BAN logic is used to prove that a session key will be agreed between the user and the car node after the execution of the proposed scheme. Table 2 shows the notations used in the BAN logic.

Table 2. Notations in the BAN logic

Symbols	Discription
P, Q	Principals
X	Statements
$\#(X)$	X is fresh
$P \triangleleft X$	P sees X
$P\vert\sim X$	P once said X
$P\vert\equiv X$	P believes X
$P \overset{K}{\leftrightarrow} Q$	P and Q use the shared key K to communicate
$P \Rightarrow X$	P has jurisdiction over X
$< X >_Y$	X combined with Y
The message-meaning rule	$\frac{P\vert\equiv P \overset{K}{\leftrightarrow} Q, P\triangleleft <X>_K}{P\vert\equiv Q\vert\sim X}$ or $\frac{P\vert\equiv \overset{K}{\rightarrow} Q, P\triangleleft \{X\}_{K^{-1}}}{P\vert\equiv Q\vert\sim X}$
The freshness-conjuncatenation rule	$\frac{P\vert\equiv\#(X), P\vert\equiv Q\vert\sim X}{P\vert\equiv Q\vert\equiv X}$
The jurisdiction rule	$\frac{P\vert\equiv Q\Rightarrow X, P\vert\equiv Q\vert\equiv X}{P\vert\equiv X}$

Our scheme should be able to achieve the following goals:

G1: $\mathrm{U}\vert\equiv \left(U \overset{SK}{\leftrightarrow} \mathrm{T}\right)$

G2: $\mathrm{T}\vert\equiv \left(U \overset{SK}{\leftrightarrow} \mathrm{T}\right)$

The idealized forms of our scheme are listed below:

M1:

$\mathrm{S} \rightarrow (U \overset{h(k_U\vert\vert e_{US}\vert\vert k_{US})}{\longleftrightarrow} T,\ U \overset{h(k_T\vert\vert e_{ST}\vert\vert k_{ST})}{\longleftrightarrow} T, U \overset{h(C_{US}\vert\vert n_1)}{\longleftrightarrow} T, U \overset{h(C_{ST}\vert\vert n_2)}{\longleftrightarrow} T,\ e_{US})_{h(SID\vert r_S)}$

M2: $S \rightarrow T(\overset{h(k_U||e_{US}||k_{US})}{\longleftrightarrow} T,\ U \overset{h(k_T||e_{ST}||k_{ST})}{\longleftrightarrow} T, U \overset{h(C_{US}||n_1)}{\longleftrightarrow} T, U \overset{h(C_{ST}||n_2)}{\longleftrightarrow} T,\ V_4)_{h(n_3||I)}$

The basic initial assumptions of our scheme are as follows:

A1: $U| \equiv \left(U \overset{h(SID|r_S)}{\longleftrightarrow} S\right)$

A2: $U| \equiv \#(e_{US})$

A3: $U| \equiv S \Rightarrow \left(U \overset{h(k_U||e_{US}||k_{US})}{\longleftrightarrow} T,\ U \overset{h(k_T||e_{ST}||k_{ST})}{\longleftrightarrow} T, U \overset{h(C_{US}||n_1)}{\longleftrightarrow} T, U \overset{h(C_{ST}||n_2)}{\longleftrightarrow} T\right)$

A4: $T| \equiv \left(S \overset{h(n_3||I)}{\longleftrightarrow} T\right)$

A5: $T| \equiv \#(V_4)$

A6: $T| \equiv S \Rightarrow \left(U \overset{h(k_U||e_{US}||k_{US})}{\longleftrightarrow} T,\ U \overset{h(k_T||e_{ST}||k_{ST})}{\longleftrightarrow} T, U \overset{h(C_{US}||n_1)}{\longleftrightarrow} T, U \overset{h(C_{ST}||n_2)}{\longleftrightarrow} T\right)$

The proof process is as follows

From M1, it is easy to get the following statement:

S1:

$S \triangleleft (U \overset{h(k_U||e_{US}||k_{US})}{\longleftrightarrow} T,\ U \overset{h(k_T||e_{ST}||k_{ST})}{\longleftrightarrow} T, U \overset{h(C_{US}||n_1)}{\longleftrightarrow} T, U \overset{h(C_{ST}||n_2)}{\longleftrightarrow} T,\ e_{US})_{h(SID|r_S)}$

From S1, A1 and the message-meaning rule, we get

S2:

$U|\equiv S| \sim \left(U \overset{h(k_U||e_{US}||k_{US})}{\longleftrightarrow} T,\ U \overset{h(k_T||e_{ST}||k_{ST})}{\longleftrightarrow} T, U \overset{h(C_{US}||n_1)}{\longleftrightarrow} T, U \overset{h(C_{ST}||n_2)}{\longleftrightarrow} T,\ e_{US}\right)$

From S2, A2 and the freshness-conjuncatenation rule, we get

S3: $U|\equiv S| \equiv \left(U \overset{h(k_U||e_{US}||k_{US})}{\longleftrightarrow} T,\ U \overset{h(k_T||e_{ST}||k_{ST})}{\longleftrightarrow} T, U \overset{h(C_{US}||n_1)}{\longleftrightarrow} T, U \overset{h(C_{ST}||n_2)}{\longleftrightarrow} T\right)$

From S3, A3 and the jurisdiction rule, we get

S4: $U| \equiv \left(U \overset{h(k_U||e_{US}||k_{US})}{\longleftrightarrow} T,\ U \overset{h(k_T||e_{ST}||k_{ST})}{\longleftrightarrow} T, U \overset{h(C_{US}||n_1)}{\longleftrightarrow} T, U \overset{h(C_{ST}||n_2)}{\longleftrightarrow} T\right)$

Since $K_{sess} = h(h(k_U||e_{US}||k_{US})||h(k_T||e_{ST}||k_{ST})||h(h(C_{US}|n_1)||h(C_{ST}||n_2)))$, we get

S5: $U| \equiv \left(U \overset{K_{sess}}{\longleftrightarrow} T\right)$ (G1)

From M2, it is easy to get the following statement:

S6:

$S \triangleleft (U \overset{h(k_U||e_{US}||k_{US})}{\longleftrightarrow} T,\ U \overset{h(k_T||e_{ST}||k_{ST})}{\longleftrightarrow} T, U \overset{h(C_{US}||n_1)}{\longleftrightarrow} T, U \overset{h(C_{ST}||n_2)}{\longleftrightarrow} T,\ V_4)_{h(n_3||I)}$

From S6, A4 and the message-meaning rule, we get

S7:

$T|\equiv S| \sim \left(U \overset{h(k_U||e_{US}||k_{US})}{\longleftrightarrow} T,\ U \overset{h(k_T||e_{ST}||k_{ST})}{\longleftrightarrow} T, U \overset{h(C_{US}||n_1)}{\longleftrightarrow} T, U \overset{h(C_{ST}||n_2)}{\longleftrightarrow} T,\ V_4\right)$

From S7, A5 and the freshness-conjuncatenation rule, we get

S8: $T|\equiv S| \equiv \left(U \overset{h(k_U||e_{US}||k_{US})}{\longleftrightarrow} T,\ U \overset{h(k_T||e_{ST}||k_{ST})}{\longleftrightarrow} T, U \overset{h(C_{US}||n_1)}{\longleftrightarrow} T, U \overset{h(C_{ST}||n_2)}{\longleftrightarrow} T\right)$

From S8, A6 and the jurisdiction rule, we get

S9: $T| \equiv U \overset{h(k_U||e_{US}||k_{US})}{\longleftrightarrow} T,\ U \overset{h(k_T||e_{ST}||k_{ST})}{\longleftrightarrow} T, U \overset{h(C_{US}||n_1)}{\longleftrightarrow} T, U \overset{h(C_{ST}||n_2)}{\longleftrightarrow} T)$

Since $K_{sess} = h(h(k_U||e_{US}||k_{US})||h(k_T||e_{ST}||k_{ST})||h(h(C_{US}||n_1)||h(C_{ST}||n_2)))$, we get

S10: $T| \equiv \left(U \overset{K_{sess}}{\longleftrightarrow} T\right)$ (G2)

Through G1 and G2, we can get that both U and T believe that a session key K_{sess} is agreed between them.

5.2 Further Security Analysis

This subsection demonstrates that the proposed scheme is immune to known attacks and provides various desirable security properties.

Mutual Authentication
Our scheme has the property of mutual authentication, which means all participants in the scheme are convinced of each other. The three participants are authenticated by each other with the pre-calculated values $\{A_i, B_{US}, C_{ST}, h(TID)\}$, the randomly generated nonces $\{n_1, n_2, n_3\}$ and the temporary computed $\{e_{US}, e_{ST}\}$ associated with ECC keys, which cannot be forged by an adversary.

Session Key Agreement
U and T generate a shared session key K_{sess}. The session key is composed of ECC keys k_x, nonces n_i and values e_{US}, e_{ST} generated by Diffie-Hellman key exchange, which guarantees forward secrecy. All values mentioned above are generated randomly, which provides the resistance of known key attack and session-specific temporary information attack.

Forward Secrecy
Forward Secrecy is a feature of specific key agreement protocols that gives assurances that session keys will not be compromised even if long-term secrets used in the session key exchange are compromised. In the proposed scheme, session key is computed as $h(h(k_U\|e_{US}\|k_{US})\|\ h(k_T\|e_{ST}\|k_{ST})\|h(h(C_{US}\|n_1\|)\|h(C_{ST}||n_2)\|))$, where $\{k_x\}$ are public keys of each participants and n_1, n_2 are random nonces. No value can be retrieved for all keys and nonces are generated temporarily which means they are different from each session. Besides, due to the intractability of ECDHP, private keys of U, S and T in each session cannot be computed by the adversary.

Known Key Security
In the proposed protocol, the session key K_{sess} is computed depending on the nonces $\{n_1, n_2\}$ and ECC secret keys $\{k_U, k_{US}, k_{ST}, k_T\}$ generated by all participants. These nonces are randomly picked and keys are computed by random numbers selected in R_{Zp} which are different from session to session. Besides, the values are not transmitted in plain text so that no adversary can get the true values of them without the knowledge

of all other well protected messages. Thus, obtaining one session key does no help for computing other session keys. So, our protocol provides known-key security.

User Anonymity
Out scheme has the property of user anonymity for any adversary cannot find or compute user information from our messages. Information of U, S and T is only transmitted in plaintext form during user registration via secure channel which can be monitored by no adversary. In login and authentication phase information of the three is protected by one-way hash function. Moreover, data in any message are randomized by nonces $\{n_i\}$ in one authentication session and by elliptic curve keys $\{d_x, k_x\}$ between sessions. Therefore, any adversary has the ability to link any two sessions.

Resist User Tracking Attack
In our proposed scheme, every message transmitted via public channels are randomized by nonces n_1, n_2 and ECC keys selected from R_{Zp}. The only one value without protection is A_i. However, A_i is updated to $A_{i+1} = A_i \oplus h(k_{US}||k_{ST})$ every time the session key generated. Therefore, when the adversary eavesdrops on the delivered messages in different sessions, he cannot confirm that two messages are from a fixed user in our scheme. Hence, our scheme provides resistance to user tracking attack.

Resist Forgery and Impersonation Attack
Our scheme is able to resist forgery and impersonation attack, because when an adversary attempts to generate a legitimate login message (A_i, D, E, V_1, T_1), he does know UID, PW, B so that he cannot generate any data except a fake timestamp.

Resist Privileged Insider Attack
If a privileged insider adversary attempts to get user-related information from a database stored on the server, he might obtain $(DID,\ B_{US},\ C_{US},\ C_{ST})$. DID, B_{US}, C_{US} are all computed with A_i which updates every session. C_{ST} are also computed by one-way hash function. Even if the insider can eavesdrop all messages transmitted in a session, he cannot disclose any vital information of user.

Resist Replay Attack
In our proposed scheme, every message sent over public channel contains a timestamp T_i, which also has been hashed with other random values in V_i. Take $meesage3$ for example, if the adversary replays a $meesage3$, as the timestamp T_3 is not fresh, ultimately the authentication fails. If the adversary replaces T_3 with a fresh timestamp T_3', as the random ECC keys in each session are different, he cannot compute a correct V_3 as a consequence.

Resist Lost-Smart-Card Attack
We propose our scheme under the assumption of adversary might obtain a lost smart card. If the attacker gets (C_1, C_2, C_{US}) in the card, when he attempts to compute A, he cannot get the correct value due to lack of UID and PW. Even if the adversary obtains

UID and PW somehow, he cannot compute a legitimate $h(SID||r_S)$ because the biometric B is not stored in the smart-card, let alone compute D, E and V_1 correctly.

Resist Man-in-the-Middle Attack
We suppose that an adversary can intercept, modify and delete messages transmitted over a public channel. However, session keys are computed by U and T, and no plaintext of key information is transmitted via the public channel. Though the adversary is able to intercept any message in login and authentication phase of our scheme, he cannot compute the session key or obtain any identity information without the help of private keys $\{d_U, d_{SU}, d_{ST}, d_T\}$. Therefore, our scheme can resist man-in-the-middle-attack.

Resist Denial-of-Service Attack
This attack occurs when an attacker transmits a huge number of request messages to either U, S or T. In our proposed protocol, each message exchanged contains a timestamp. The legitimacy of each message is also checked by calculating V_n. Thus, any timed out or unauthorized message gets detected and is rejected. As a consequence, DoS attack can be well defended by our protocol.

6 Performance Evaluation

In this section, we present the performance evaluation of our scheme and compare our work with relevant schemes.

6.1 Security Features

Table 3 presents the security analysis results. According to the threat model stated in Sect. 1, the analyzed security attributes include the basic functional features such as mutual authentication, user anonymity and forward secrecy, etc. Resistance of usual like impersonation attack, insider attack and replay attack are also evaluated.

6.2 Computational Overhead

Table 4 shows the computational overhead of our scheme. Let the time to compute one hash operation, one bio hash operation, one ECC point multiplication operation be T_h, T_{h_B} and T_E, respectively. Numbers of operations done by 3 participants are given in Table 4. During calculation, we neglect the computational overhead of some lightweight operations such as xor, concatenation, comparison.

As is shown in Table 3, our proposed protocol provides registration, login, authentication and key agreement function with little computational resource. In SRVC system, the smart device, the car enterprise server and the smart terminal of car have relatively strong computing power. Therefore, our protocol works smoothly on SRVC system.

Table 3. Security features comparison. Y = Yes, N = No

Security properties	Panda et al. [13]	Mo et al. [12]	Chatterjee et al. [11]	Our scheme
Mutual authentication	Y	Y	Y	Y
Session key agreement	Y	Y	Y	Y
Forward secrecy	Y	Y	N	Y
Known key security	N	N	Y	Y
User anonymity	N	Y	N	Y
Resist user tracking Attack	N	Y	Y	Y
Resist user/server impersonation Aattack	N	N	Y	Y
Resist privileged insider attack	N	Y	Y	Y
Resist replay attack	Y	N	Y	Y
Resist lost-smart-card attack	N	N	Y	Y
Resist man-in-the-middle attack	Y	N	Y	Y
Resist denial of service attack	N	N	N	Y

Table 4. The computational overhead of our scheme

Phase	The user	The server	The terminal
Registration	$2T_h + 1T_{h_B}$	$5T_h$	$1T_h$
Login	$3T_h + 1T_{h_B} + 1T_E$	/	/
Auth and key agreement	$7T_h + 1T_E$	$14T_h + 4T_E$	$16T_h + 2T_E$
Total	$12T_h + 2T_{h_B} + 2T_E$	$19T_h + 4T_E$	$17T_h + 2T_E$

7 Conclusion and Future Work

In this paper, we first introduce the smart remote vehicle control system's features, which are similar but also has differences with IoT systems. Then based on the Delov-Yao adversary model, we analyze Chatterjee et al.'s three-way ECC-based IoT authentication protocol, and demonstrate the limitations in this protocol such as lack resistance of DoS attack, impersonation attack and user tracking attack. To give a more security for SRVC system, then we propose a Privacy-Preserving ECC-based three-factor authentication for SRVC system. Security and performance analysis show that our protocol can resist

known attacks and has better effectiveness. Finally, the performance analysis shows that our protocol can run efficiently under the system.

In the future, we will study how to extend our protocol to more application scenarios of SRVC system, such as vehicle cancellation, vehicle transaction, temporary vehicle key authorization, etc.

Acknowledgments. This research was funded by the National Key Research and Development Program of China (No. 2018YFB0803600).

References

1. Bono, S., Green, M., Stubblefield, A., Juels, A., Rubin, A.D., Szydlo, M.: Security analysis of a cryptographically-enabled RFID device. In: USENIX Security Symposium, vol. 31, pp. 1–16 (2005)
2. Francillon, A., Danev, B., Capkun, S.: Relay attacks on passive keyless entry and start systems in modern cars. In: Proceedings of the Network and Distributed System Security Symposium, Eidgenössische Technische Hochschule Zürich, Department of Computer Science (2011)
3. Lamport, L.: Password authentication with insecure communication. Commun. ACM **24**(11), 770–772 (1981)
4. Kumari, S., Li, X., Wu, F., Das, A.K., Odelu, V., Khan, M.K.: A User anonymous mutual authentication protocol. KSII Trans. Internet Inf. Syst. **10**(9), 4508–4528 (2016)
5. Chen, T.H., Yeh, H.L., Shih, W.K.: An advanced ECC dynamic id-based remote mutual authentication scheme for cloud computing. In: 2011 Fifth FTRA International Conference on Multimedia and Ubiquitous Engineering, pp. 155–159. IEEE (2011)
6. Alotaibi, M.: An enhanced symmetric cryptosystem and biometric-based anonymous user authentication and session key establishment scheme for WSN. IEEE Access **6**, 70072–70087 (2018)
7. Porambage, P., Schmitt, C., Kumar, P., Gurtov, A., Ylianttila, M.: Two-phase authentication protocol for wireless sensor networks in distributed IoT applications. In: 2014 IEEE Wireless Communications and Networking Conference, pp. 2728–2733. IEEE (2014)
8. Chu, F., Zhang, R., Ni, R., Dai, W.: An improved identity authentication scheme for internet of things in heterogeneous networking environments. In: 2013 16th International Conference on Network-Based Information Systems, pp. 589–593. IEEE (2013)
9. Lai, C., Li, H., Lu, R., Shen, X.S.: SE-AKA: a secure and efficient group authentication and key agreement protocol for LTE networks. Comput. Netw. **57**(17), 3492–3510 (2013)
10. Lai, C., Li, H., Lu, R., Jiang, R., Shen, X.: SEGR: a secure and efficient group roaming scheme for machine to machine communications between 3GPP and WiMAX networks. In: 2014 IEEE International Conference on Communications, pp. 1011–1016. IEEE (2014)
11. Chatterjee, S., Samaddar, S.G.: A robust lightweight ECC-based three-way authentication scheme for IoT in cloud. In: Elçi, A., Sa, P.K., Modi, C.N., Olague, G., Sahoo, M.N., Bakshi, S. (eds.) Smart Computing Paradigms: New Progresses and Challenges. AISC, vol. 767, pp. 101–111. Springer, Singapore (2020). https://doi.org/10.1007/978-981-13-9680-9_7
12. Mo, J., Hu, Z., Chen, H., Shen, W.: An efficient and provably secure anonymous user authentication and key agreement for mobile cloud computing. Wirel. Commun. Mob. Comput. (2019)
13. Panda, P.K., Chattopadhyay, S.: A secure mutual authentication protocol for IoT environment. J. Reliable Intell. Environ. **6**(2), 79–94 (2020). https://doi.org/10.1007/s40860-020-00098-y

14. Zhou, L., Li, X., Yeh, K.H., Su, C., Chiu, W.: Lightweight IoT-based authentication scheme in cloud computing circumstance. Future Gener. Comput. Syst. **91**, 244–251 (2019)
15. Truong, T.T., Tran, M.T., Duong, A.D.: Modified dynamic ID-based user authentication scheme resisting smart-card-theft attack. Appl. Math. Inf. Sci. **8**(3), 967 (2014)
16. Lee, Y.C.: A new dynamic id-based user authentication scheme to resist smart card theft attack. Appl. Math. Inf. Sci. **6**, 355–361 (2012)
17. Feng, Q., He, D., Zeadally, S., Wang, H.: Anonymous biometrics-based authentication scheme with key distribution for mobile multi-server environment. Future Gener. Comput. Syst. **84**, 239–251 (2018)
18. Aghili, S.F., Mala, H., Shojafar, M., Peris-Lopez, P.: LACO: lightweight three-factor authentication, access control and ownership transfer scheme for e-health systems in IoT. Future Gener. Comput. Syst. **96**, 410–424 (2019)
19. Dolev, D., Yao, A.: On the security of public key protocols. IEEE Trans. Inf. Theory **29**(2), 198–208 (1983)