

Received May 2, 2018, accepted May 31, 2018, date of publication June 5, 2018, date of current version June 26, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2844190

2PAKEP: Provably Secure and Efficient Two-Party Authenticated Key Exchange Protocol for Mobile Environment

KISUNG PARK¹, YOUNGHO PARK¹, (Member, IEEE), YOCHAN PARK²,
AND ASHOK KUMAR DAS³, (Member, IEEE)

¹School of Electronics Engineering, Kyungpook National University, Daegu 41566, South Korea

²Division of IT Convergence, Korea Nazarene University, Cheonan 31172, South Korea

³Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India

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 funded by the Ministry of Science, ICT and Future Planning under Grant 2017R1A2B1002147 and in part by the BK21 Plus Project funded by the Ministry of Education, South Korea under Grant 21A20131600011.

ABSTRACT With the increasing use of mobile devices, a secure communication and key exchange become the significant security issues in mobile environments. However, because of open network environments, mobile user can be vulnerable to various attacks. Therefore, the numerous authentication and key exchange schemes have been proposed to provide the secure communication and key exchange. Recently, Qi and Chen proposed an efficient two-party authentication key exchange protocol for mobile environments in order to overcome the security weaknesses of the previous authentication and key exchange schemes. However, we demonstrate that Qi and Chen's scheme is vulnerable to various attacks such as impersonation, offline password guessing, password change, and privileged insider attacks. We also show that Qi and Chen's scheme does not provide anonymity, efficient password change mechanism, and secure mutual authentication. In this paper, to overcome the outlined abovementioned security vulnerabilities, we propose a secure and efficient two-party authentication key exchange protocol, called 2PAKEP, that hides user's real identity from an adversary using a secret parameter. 2PAKEP also withstands various attacks, guarantees anonymity, and provides efficient password change mechanism and secure mutual authentication. In addition, we prove that 2PAKEP provides the secure mutual authentication using the broadly accepted Burrows–Abadi–Needham logic and the session key security using the formal security analysis under the widely accepted real-or-random model. Moreover, the formal security verification using the popular simulated software tool, Automated Validation of Internet Security Protocols and Applications, on 2PAKEP shows that the replay and man-in-the-middle attacks are protected. In addition, we also analyze the performance and security and functionality properties of 2PAKEP and compare these with the related existing schemes. Overall, 2PAKEP provides better security and functionality features, and also the communication and computational overheads are comparable with the related schemes. Therefore, 2PAKEP is applicable to mobile environment efficiently.

INDEX TERMS Mobile environment, authentication, key exchange, BAN logic, AVISPA, formal security.

I. INTRODUCTION

Mobile devices along with the information and communication technology (ICT) have advanced to such an extent that mobile users can freely utilize various services such as roaming, file sharing, smart healthcare, mobile banking, shopping, and payment on the go. These mobile services are convenient and improve the overall welfare of users. However, due to an open network property, an adversary could intercept, modify, replay, delete or eavesdrop the

transmitted information, and then an adversary could try to obtain sensitive user data by various attacks such as replay, masquerading, impersonation and password guessing attack. Therefore, to ensure the privacy of mobile users, secure two-party authentication and key exchange has become a very important security issue.

For the last few decades, several password based authentication key exchange protocols [1]–[6] and smart card based authentication key exchange protocols [6]–[13], [20]–[24]

have been proposed to provide privacy of the mobile users. In 2009, Yang and Chang [10] proposed an ID-based authentication scheme with smart card using Elliptic Curve Cryptography (ECC). However, Yoon and Yoo [19] found that Yang and Chang's scheme cannot prevent impersonation attack and it cannot also provide perfect forward secrecy. To overcome these security weaknesses, they proposed an enhanced authentication scheme. In 2012, He *et al.* [11] claimed that Yoon and Yoo's scheme still cannot provide perfect forward secrecy, and they then proposed an enhanced ID-based client authentication with key agreement protocol. However, in 2013, Chou *et al.* [12] showed that He *et al.*'s scheme has flaw of private key verification process. In 2015, Yang *et al.* [13] demonstrated that He *et al.*'s scheme cannot resist impersonation and unknown key share attack, and then Yang *et al.* proposed an improved two-party authentication key exchange protocol.

To overcome security weaknesses of above mentioned schemes, recently in 2017, Qi and Chen [25] proposed an efficient two-party authentication key exchange protocol for mobile environments using ECC. Qi and Chen [25] also showed that their proposed scheme can resist various attacks including man-in-the-middle, stolen verifier and replay attacks. They also claimed that their scheme can provide perfect forward secrecy, session key security, mutual authentication and anonymity. However, we demonstrate that Qi et al's scheme cannot prevent impersonation, password change, privileged-insider and offline password guessing attacks. We also show that their scheme does not provide the anonymity, secure mutual authentication, secure key agreement and efficient password change mechanism. Subsequently, we propose a secure and efficient two-party authentication key exchange protocol to solve these security vulnerabilities.

A. THREAT MODEL

For the security analysis in this paper, we present the Dolev-Yao (DY) threat model [14], which is generally used for analysis of the security of a cryptographic protocol. The assumptions of the threat model are as follows.

- Firstly, under the DY model an adversary can eavesdrop, modify, replay or delete the messages transmitted over a public channel.
- Secondly, an adversary can obtain a lost or stolen mobile device, and can then extract all the stored information in the mobile device or smart card [15], [16].
- Finally, an adversary can be a legitimate user of the system (privileged-insider) or an outsider, and that he/she can perform various attacks using obtained information [17], [18].

B. RESEARCH CONTRIBUTIONS

The contributions of this paper are listed below.

- We point out the security weaknesses of Qi and Chen's scheme and demonstrate that it is vulnerable to various attacks such as impersonation, password change, offline

password guessing and privileged-insider attacks. We also show that Qi and Chen's scheme cannot achieve secure mutual authentication, secure key agreement, efficient password change mechanism and anonymity.

- To overcome the security weaknesses of Qi and Chen's scheme, we propose a provably secure and efficient two-party authentication key exchange protocol for mobile environments. 2PAKEP prevents impersonation, password change and offline password guessing attack, and also provides secure mutual authentication, key agreement, perfect forward secrecy, session key security, efficient password change mechanism and anonymity.
- 2PAKEP has an efficient password change mechanism because mobile users can change passwords freely without the server assistance.
- We prove that 2PAKEP provides secure mutual authentication and session key security using the BAN logic [26] and formal security analysis under the ROR model, respectively. 2PAKEP is also informally (non-mathematically) analyzed to prove its security against other potential attacks. We then analyze the performance of 2PAKEP and other related existing schemes.
- The formal security verification of 2PAKEP is also done through the simulation study using the broadly-accepted AVISPA tool.

C. PAPER OUTLINE

The remainder of this paper is organized as follows. In Section II, we review the authenticated key exchange scheme of Qi and Chen [25]. In Section III, we analyze of the security weaknesses of Qi and Chen's scheme. In Section IV, to overcome security weaknesses of Qi and Chen's scheme, we propose a provably secure and efficient two-party authentication key exchange protocol for mobile environment. In Sections V and VI, we discuss the security of 2PAKEP. We compare the performance of 2PAKEP with the related existing schemes in Section VII. Finally, we conclude this paper in Section VIII.

II. REVIEW OF QI AND CHEN'S SCHEME

In this section, we review the Qi and Chen's two-party authentication key exchange protocol for mobile environment. Their scheme consists of four phases: 1) system initialization, 2) user registration, 3) mutual authentication and key exchange, and 4) password change activity. The notations used in this paper are listed in Table 1.

A. SYSTEM INITIALIZATION PHASE

Before the user registration phase, the server S has to perform system initialization phase. The system initialization process is as follows:

- Step 1:** S sets an elliptic curve E/F_p and chooses a base point P with an order n over E/F_p , where n is a large prime number.

TABLE 1. Notations.

Notation	Description
U	A mobile user
S	A server
A	An adversary
ID_U	A identity of a mobile user U
PW_U	A password of U
d_S	A long-term private key of S
Q_S	A long-term public key of S
SK	A session key between S and U
SK_{FA}	A secret key of FA
kdf	A secure one-way key derivation function
$H_1(\cdot), H_2(\cdot)$	Collision-resistant cryptographic one-way hash functions
\oplus	An exclusive-OR operation
\parallel	A concatenation operation
E/F_p	An elliptic curve E over a prime finite field F_p ; p being a large prime
P	A base point in E/F_p
kP	Elliptic curve point (scalar) multiplication of $P \in E/F_p$; k being a scalar
$A \dashrightarrow B : M$	Entity A sends message M to entity B via secure channel
$A \rightarrow B : M$	Entity A sends message M to entity B via public channel

Step 2: S chooses a long-term private key d_S from Z_n^* and computes a public key $Q_S = d_S P$.

Step 3: Finally, S selects collision-resistant one-way hash functions $H_1(\cdot)$ and $H_2(\cdot)$, and then S publishes the system parameters $\{E/F_p, P, n, Q_S, H_1(\cdot), H_2(\cdot)\}$.

B. USER REGISTRATION PHASE

If a new mobile user U wants to use various mobile services, U has to register his/her identity with the server S . The user registration phase is shown in Figure 1 and the detailed steps are as follows:

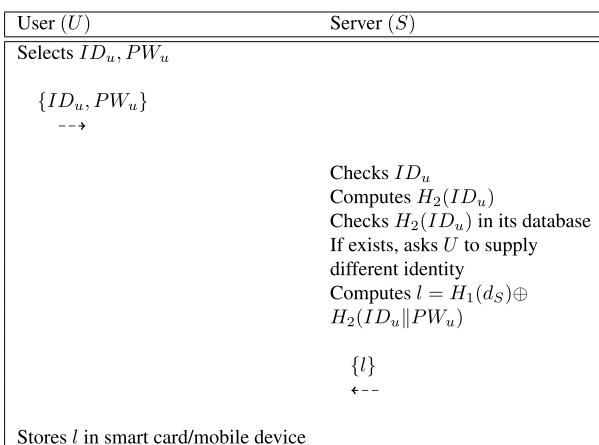


FIGURE 1. User registration phase of the Qi and Chen’s scheme.

Step 1: U submits his/her identity ID_u and password PW_u through a secure channel (e.g., in person).

Step 2: After receiving $\{ID_u, PW_u\}$, S computes $H_2(ID_u)$, and then checks whether ID_u and $H_2(ID_u)$ exist in its database. If these exist, S asks the U for a new identity.

Step 3: S computes the parameter $l = H_1(d_S) \oplus H_2(ID_u || PW_u)$ and delivers it to U through secure channel.

Step 4: Finally, U stores the secure parameter l into his/her smart card or mobile device.

C. MUTUAL AUTHENTICATION AND KEY EXCHANGE PHASE

When a mobile user U wants to access mobile services, U requires to send an authenticated key exchange request to server S . Qi and Chen’s mutual authenticated key exchange phase is given in Figure 2 and also its detailed steps are as follows:

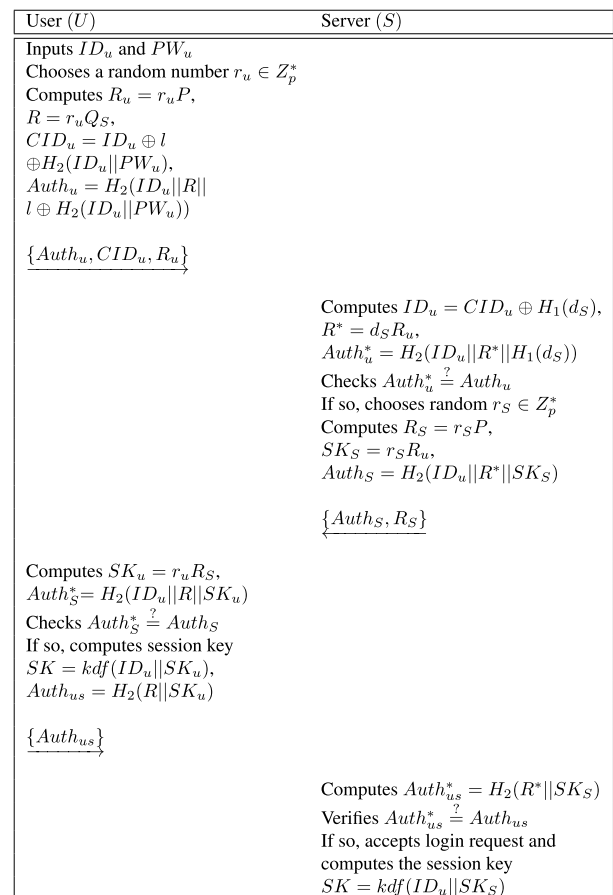


FIGURE 2. Mutual authentication key exchange phase of the Qi and Chen’s scheme.

Step 1: U inputs the identity ID_u and password PW_u , and then chooses a random number $r_u \in Z_p^*$. U also computes $R_u = r_u P$, $R = r_u Q_S$, $CID_u = ID_u \oplus l \oplus H_2(ID_u || PW_u)$ and $Auth_u = H_2(ID_u || R || l \oplus H_2(ID_u || PW_u))$. After these computations, U sends

the login request message $\{Auth_u, CID_u, R_u\}$ to S via open channel.

- Step 2:** After receiving the message $\{Auth_u, CID_u, R_u\}$, S computes $ID_u = CID_u \oplus H_1(d_S)$, $R^* = d_S R_u$ and $Auth_u^* = H_2(ID_u || R^* || H_1(d_S))$. After that S checks whether $Auth_u^* \stackrel{?}{=} Auth_u$. If this condition holds, S chooses a random number $r_S \in Z_p^*$ and computes $R_S = r_S P$, $SK_S = r_S R_u$ and $Auth_S = H_2(ID_u || R^* || SK_S)$, and then sends the authentication request message $\{Auth_S, R_S\}$ to U via public channel.
- Step 3:** Upon receiving the message $\{Auth_S, R_S\}$, U computes $SK_u = r_u R_S$, $Auth_S^* = H_2(ID_u || R || SK_u)$ and checks whether $Auth_S^* \stackrel{?}{=} Auth_S$. If the condition holds, U computes the session key $SK = kdf(ID_u || SK_u)$ and $Auth_{us} = H_2(R || SK_u)$, and sends the authentication reply message $\{Auth_{us}\}$ to S through open channel. The kdf can be considered as a keyed-hash message authentication code or hash-based message authentication code.
- Step 4:** After receiving the message $\{Auth_{us}\}$, S computes $Auth_{us}^* = H_2(R^* || SK_S)$ and checks whether $Auth_{us}^* \stackrel{?}{=} Auth_{us}$. If the condition is satisfied, S and U achieve the mutual authentication and session key agreement successfully.

D. PASSWORD CHANGE PHASE

When a mobile user U wants to change his/her password, he/she should change his/her password freely. The password change activity is shown in Figure 3 and the detailed steps of this phase are as follows:

User (U)	Server (S)
Inputs ID_u, PW_u, PW_{new} Picks a random number $r_u \in Z_p^*$ Calculates $R_u = r_u P$, $R = r_u Q_S$, $CID_u = ID_u \oplus l \oplus H_2(ID_u PW_u)$, $Auth_u = H_2(ID_u R l \oplus H_2(ID_u PW_u))$ $\{Auth_u, CID_u, R_u\}$	Calculates $ID_u = CID_u \oplus H_1(d_S)$, $R^* = d_S R_u$, $Auth_u^* = H_2(ID_u R^* H_1(d_S))$ Verifies $Auth_u^* \stackrel{?}{=} Auth_u$ If so, picks a random number $r_S \in Z_p^*$ Calculates $R_S = r_S P$, $SK_S = r_S R_u$, $Auth_S = H_2(ID_u R SK_u)$ $\{Auth_S, R_S\}$
Calculates $SK_u = r_u R_S$, $Auth_S^* = H_2(ID_u R SK_u)$ Verifies $Auth_S^* \stackrel{?}{=} Auth_S$ If so, replaces l with $l_{new} = l \oplus H_2(ID_u PW_u) \oplus H_2(ID_u PW_{new})$ in smart card/mobile device	

FIGURE 3. Password change activity of the Qi and Chen's scheme.

- Step 1:** U inputs the identity ID_u and old password PW_u and new password PW_{new} . U then chooses a

random number $r_u \in Z_p^*$. After that U computes $R_u = r_u P$, $R = r_u Q_S$, $CID_u = ID_u \oplus l \oplus H_2(ID_u || PW_u)$, $Auth_u = H_2(ID_u || R || l \oplus H_2(ID_u || PW_u))$ and sends the password change request message $\{Auth_u, CID_u, R_u\}$ to S via public channel.

- Step 2:** After receiving $\{Auth_u, CID_u, R_u\}$ from U , S computes $ID_u = CID_u \oplus H_1(d_S)$, $R^* = d_S R_u$ and $Auth_u^* = H_2(ID_u || R^* || H_1(d_S))$. S then checks whether $Auth_u^* \stackrel{?}{=} Auth_u$. If they are equal, S chooses a random number $r_S \in Z_p^*$ and calculates $R_S = r_S P$, $SK_S = r_S R_u$ and $Auth_S = H_2(ID_u || R || SK_u)$, and sends the password change reply message $\{Auth_S, R_S\}$ to U via public channel.
- Step 3:** Upon receiving $\{Auth_S, R_S\}$, U computes $SK_u = r_u R_S$, $Auth_S^* = H_2(ID_u || R || SK_u)$ and checks whether $Auth_S^* \stackrel{?}{=} Auth_S$. Finally, U replaces l with the $l_{new} = l \oplus H_2(ID_u || PW_u) \oplus H_2(ID_u || PW_{new})$ in his/her smart card or mobile device.

III. CRYPTANALYSIS OF QI AND CHEN'S SCHEME

In this section, we analyze the security weaknesses of Qi and Chen's scheme. Qi and Chen [25] claimed that their scheme is robust against various attacks, and can provide perfect forward secrecy, session key security, mutual authentication and anonymity. However, we demonstrate that Qi and Chen's scheme cannot resist the following attacks.

A. USER IMPERSONATION ATTACK

According to Section I-A, we assume that an authorized user of the system can be an adversary U_a who intercepts the transmitted messages in the previous session. Next, U_a obtains the parameter $H_1(d_S) = l \oplus H_2(ID_a || PW_a)$, where l is the parameter stored in smart card or mobile device, and ID_a and PW_a are the identity and password of U_a , respectively. Finally, U_a performs the user impersonation attack using the following steps:

- Step 1:** U_a retrieves $ID_u = CID_u \oplus H_1(d_S)$, where CID_u is the authentication request message transmitted in the previous session. U_a also chooses a random number $r_a \in Z_p^*$ and computes $R_a = r_a P$, $R = r_a Q_S$ and $Auth_a = H_2(ID_u || R || H_1(d_S))$, and then U_a sends the request message $\{Auth_a, CID_u, R_a\}$ to S .
- Step 2:** After receiving the message $\{Auth_a, CID_u, R_a\}$, S computes $ID_u = CID_u \oplus H_1(d_S)$, $R^* = d_S R$ and $Auth_a^* = H_2(ID_u || R^* || H_1(d_S))$. After that, S checks whether $Auth_a^* \stackrel{?}{=} Auth_a$. If the condition is valid, S chooses a random number $r_S \in Z_p^*$ and computes $R_S = r_S P$, $SK_S = r_S R_a$ and $Auth_S = H_2(ID_u || R^* || SK_S)$, and then sends the respond message $\{Auth_S, R_S\}$ to U .
- Step 3:** Upon receiving the respond message $\{Auth_S, R_S\}$, U_a computes $SK_a = r_a R_S$, $Auth_S^* = H_2(ID_u || R || SK_a)$ and checks whether $Auth_S^* \stackrel{?}{=} Auth_S$. If they are equal, U_a computes session key

$SK = kdf(ID_u || SK_a)$ and $Auth_{as} = H_2(R || SK_a)$, and then sends the message $\{Auth_S\}$ to S .

Step 4: After receiving the message $\{Auth_S\}$, S computes $Auth_{as}^* = H_2(R^* || SK_S)$ and checks whether $Auth_{as}^* \stackrel{?}{=} Auth_{as}$. If they are equal, S and U_a successfully achieve the mutual authentication and session key agreement.

Therefore, Qi and Chen's scheme does not withstand user impersonation attack.

B. PASSWORD CHANGE ATTACK

The password change process of Qi and Chen's scheme is similar to the mutual authentication and key exchange phase. If an adversary U_a has obtained the identity of the user ID_u , U_a can change the U 's password freely. The result of this attack shows that Qi and Chen's scheme cannot resist password change attack.

C. OFFLINE PASSWORD GUESSING ATTACK

We assume that an adversary U_a has stolen or obtained the smart card or mobile device of a legal registered user U , and then attempts to guess the correct password of U using the following steps:

Step 1: First, U_a computes $ID_u = CID_u \oplus H_1(d_S)$ and $H_2(ID_u || PW_u) = l \oplus H_1(d_S)$, where CID_u is the authentication request message transmitted in the previous session.

Step 2: Next, U_a guesses a password PW' and computes $H_2(ID_u || PW')$. Then, U_a checks whether $H_2(ID_u || PW') \stackrel{?}{=} H_2(ID_u || PW_u)$.

Step 3: If the above condition holds, U_a has guessed the password of U correctly.

Therefore, Qi and Chen's scheme is vulnerable to offline password guessing attack.

D. ANONYMITY PRESERVATION

From Section III-A, it is clear that an adversary \mathcal{A} can easily obtain user's real identity ID_u because \mathcal{A} has the parameter $H_1(d_S)$ and can also compute $ID_u = CID_u \oplus H_1(d_S)$ using $H_1(d_S)$. Hence, the user U 's real identity ID_u is revealed to \mathcal{A} , and as a result, Qi and Chen's scheme does not preserve the user anonymity property.

E. SECURE MUTUAL AUTHENTICATION AND SESSION KEY AGREEMENT

According to Section III-A, an adversary can impersonate a legitimate user successfully. Therefore, Qi and Chen's scheme cannot provide secure mutual authentication and session key agreement.

F. PRIVILEGED-INSIDER ATTACK

During the user registration phase, a registered user U submits his/her chosen identity ID_u and password PW_u to the server S . Thus, a privileged-insider user of the S being an insider attacker knows ID_u and PW_u directly.

Therefore, if U uses the same PW_u for other applications, the privileged-insider attacker can utilize this password to obtain other services for which the user U is eligible. Hence, it is clear that Qi and Chen's scheme also fails to maintain the privileged-insider attack.

IV. THE PROPOSED SCHEME

In this section, we present our provably secure and efficient two-party authenticated key exchange protocol (2PAKEP) for mobile environment that overcomes the security weaknesses of Qi and Chen's scheme discussed in Section III. 2PAKEP also consists of four phases as in the Qi and Chen's scheme, namely 1) system initialization, 2) user registration, 3) mutual authentication and key exchange, and 4) password change. It is worth noticing that the system initialization phase of 2PAKEP remains same as that for Qi and Chen's scheme.

To achieve replay attack protection and freshness, we apply the current system timestamps in 2PAKEP. For this issue, it is assumed that all the network participants (users and server) are synchronized in their respective clocks. This practical assumption has been widely applied in many authentication protocols that are recently proposed in [27]–[33].

A. USER REGISTRATION PHASE

If a mobile user U wants to access various services from the server S , U must register with S . The user registration phase of 2PAKEP is executed in a secure channel (for example, in person) that is shown in Figure 4, and the detailed steps are as follows:

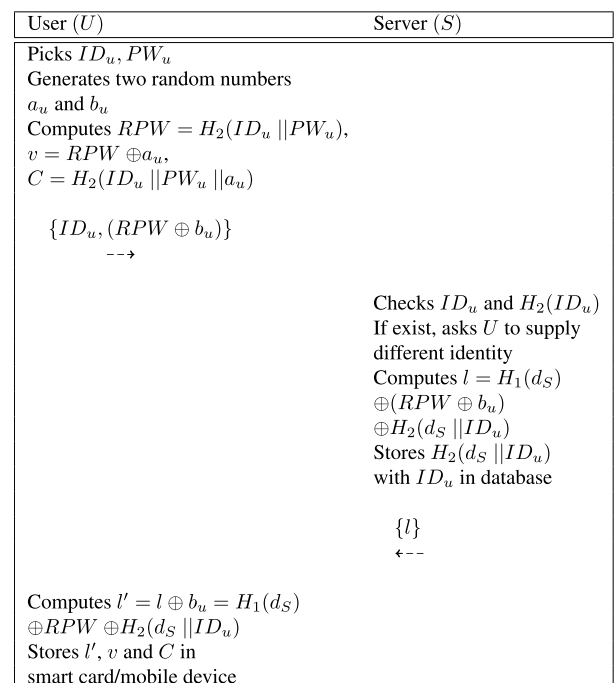


FIGURE 4. User registration phase of 2PAKEP.

- Step 1:** U picks his/her identity ID_u and password PW_u . After that, U generates two random numbers a_u and b_u , and computes $RPW = H_2(ID_u || PW_u)$, $v = RPW \oplus a_u$ and $C = H_2(ID_u || PW_u || a_u)$. Then, U submits ID_u and the masked password ($RPW \oplus b_u$) to the S through a secure channel.
- Step 2:** After receiving $\{ID_u, RPW \oplus b_u\}$, S calculates $H_2(ID_u)$, and then checks whether ID_u and $H_2(ID_u)$ exist in its own database. If they exist, S asks U to register with another new identity.
- Step 3:** S proceeds to calculate the parameter $l = H_1(d_S) \oplus (RPW \oplus b_u) \oplus H_2(d_S || ID_u)$, and then S stores $H_2(d_S || ID_u)$ with ID_u in its database and delivers the parameter l to U through a secure channel.
- Step 4:** Finally, U calculates $l' = l \oplus b_u = H_1(d_S) \oplus RPW \oplus H_2(d_S || ID_u)$, and stores the secret credentials l' , v and C in his/her smart card or mobile device.

B. MUTUAL AUTHENTICATION AND KEY EXCHANGE PHASE

When a registered mobile user U wants to access the services from the server S , U needs to send a mutual authentication key exchange request message to the S . The mutual authentication key exchange phase is briefed in Figure 5 and the detailed steps are as follows:

- Step 1:** U inputs identity ID_u and password PW_u either using smart card or mobile device, and calculates $RPW = H_2(ID_u || PW_u)$, $a_u = v \oplus RPW$ and $C'_u = H_2(ID_u || PW_u || a_u)$. Then, U checks whether $C \stackrel{?}{=} C'_u$. If they are equal, U picks a random number $r_u \in Z_p^*$ and generates the current timestamp T_u , and computes $R_u = r_u P$, $R = r_u Q_S$, $CID_u = l' \oplus RPW = H_1(d_S) \oplus H_2(d_S || ID_u)$ and $Auth_u = H_2(ID_u || R || CID_u || T_u)$. Then, U sends the login request message $Msg_1 = \{Auth_u, CID_u, R_u, T_u\}$ to the S via open channel.
- Step 2:** After receiving Msg_1 , S validates the received timestamp T_u by the verification condition $|T_u^* - T_u| < \Delta T$, where ΔT is the maximum transmission delay and T_u^* is the reception time of the message Msg_1 . If it is valid, S computes $H_2(d_S || ID_u) = CID_u \oplus H_1(d_S)$ and retrieves the real identity ID_u of U in its own database corresponding to the computed $H_2(d_S || ID_u)$. After that, S computes $R^* = d_S R_u$, $Auth_u^* = H_2(ID_u || R^* || CID_u || T_u)$ and checks whether $Auth_u^* \stackrel{?}{=} Auth_u$. If they are equal, S picks a random number $r_S \in Z_p^*$, generates the current timestamp T_S and computes $R_S = r_S P$, $SK_S = r_S R_u$ and $Auth_S = H_2(ID_u || R^* || SK_S || T_S)$. S sends the authentication request message $Msg_2 = \{Auth_S, R_S, T_S\}$ to U via open channel.
- Step 3:** Upon receiving Msg_2 , U first verifies the received timestamp T_S by the condition $|T_S^* - T_S| < \Delta T$, where the reception time of the message Msg_2 is T_S^* . If the condition is legitimate, U calculates

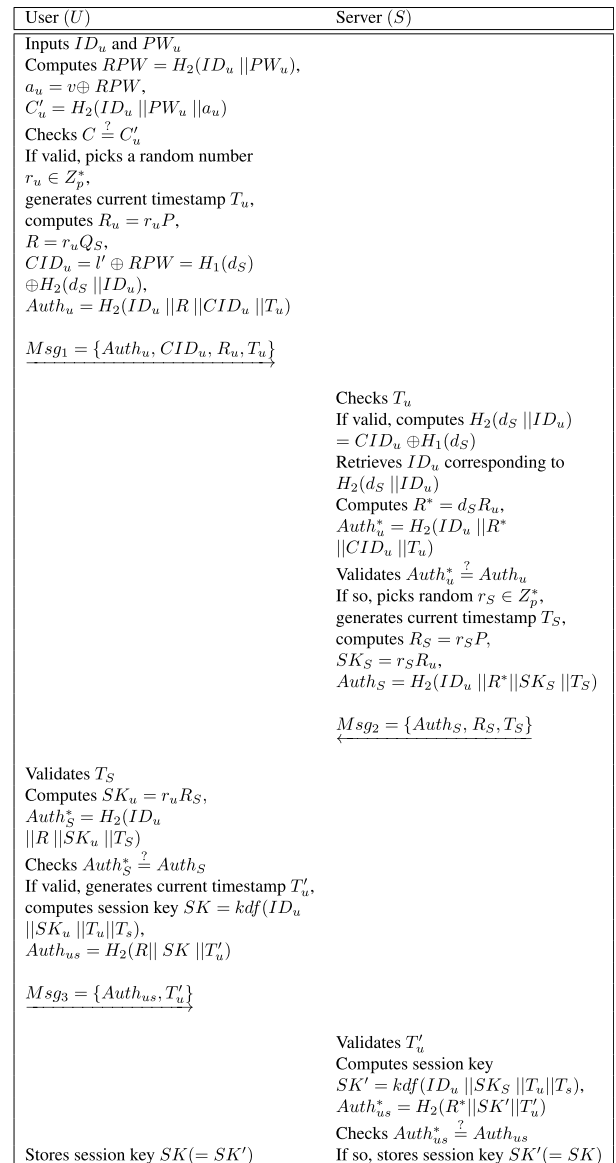


FIGURE 5. Mutual authentication key exchange phase of 2PAKEP.

$SK_u = r_u R_S$, $Auth_S^* = H_2(ID_u || R || SK_u || T_S)$, and checks whether $Auth_S^* \stackrel{?}{=} Auth_S$. If they are equal, U generates the current timestamp T'_u , and computes the session key $SK = kdf(ID_u || SK_u || T_u || T_S)$ and $Auth_{us} = H_2(R || SK || T'_u)$, and U sends the authentication reply message $Msg_3 = \{Auth_{us}, T'_u\}$ to the S through public channel.

- Step 4:** After receiving Msg_3 , S validates the timestamp T'_u with the condition $|T'_u - T'_u| < \Delta T$, where the reception time of the message Msg_3 is T'_u . After that S computes the session key $SK' = kdf(ID_u || SK_S || T_u || T_S)$, $Auth_{us}^* = H_2(R^* || SK' || T'_u)$ and checks whether $Auth_{us}^* \stackrel{?}{=} Auth_{us}$. If they are equal, S and U have successfully achieved the mutual authentication and session key agreement phase.

At the end, both S and U store the same session key $SK (= SK')$ for their secure communications.

C. PASSWORD CHANGE ACTIVITY

In 2PAKEP, a mobile user U can change his/her password freely without further involving the server S . The password change activity has the following steps:

- Step 1:** U first inputs identity ID_u and old password PW_u through the smart card or mobile device.
- Step 2:** The smart card (mobile device) computes $RPW = H_2(ID_u || PW_u)$, $a_u = v \oplus RPW$ and $C' = H_2(ID_u || PW_u || a_u)$, and then checks if $C = C'$ is satisfied. If it is valid, it asks U to enter his/her chosen new password.
- Step 3:** U chooses a new password PW_{new} and inputs it to the smart card or mobile device.
- Step 4:** The smart card (mobile device) continues to calculate $RPW_{new} = H_2(ID_u || PW_{new})$, $v_{new} = RPW_{new} \oplus a_u$, $C_{new} = H_2(ID_u || PW_{new} || a_u)$ and $l_{new} = l' \oplus RPW \oplus RPW_{new} = H_1(d_S) \oplus RPW_{new} \oplus H_2(d_S || ID_u)$. Finally, U replaces l' , v , C with the l_{new} , v_{new} and C_{new} in her/her smart card (mobile device), respectively.

Our password change activity phase is summarized in Figure 6.

User (U)	Smart card/Mobile device
Inputs identity ID_u and old password PW_u	Computes $RPW = H_2(ID_u PW_u)$, $a_u = v \oplus RPW$, $C' = H_2(ID_u PW_u a_u)$ Checks $C \stackrel{?}{=} C'$ If valid, asks U to enter his/her chosen new password
Picks new password PW_{new}	Calculates $RPW_{new} = H_2(ID_u PW_{new})$, $v_{new} = RPW_{new} \oplus a_u$, $C_{new} = H_2(ID_u PW_{new} a_u)$, $l_{new} = l' \oplus RPW \oplus RPW_{new}$ $= H_1(d_S) \oplus RPW_{new} \oplus H_2(d_S ID_u)$
Replaces l' , v , C with l_{new} , v_{new} and C_{new} in smart card (mobile device)	

FIGURE 6. Password change activity of 2PAKEP.

V. SECURITY ANALYSIS

The broadly-accepted formal methods (i.e., random oracle model) cannot capture some structural mistakes, and as a result, ensuring the soundness of authentication protocols is still an open problem [31]. Hence, it is also necessary for the security analysis informally (non-mathematical) to assure that an authentication scheme becomes secure against various known attacks with high probability. To achieve this purpose, in this section, we perform the formal security analysis using the widely-accepted Real-Or-Random (ROR) model [40] and then mutual authentication proof using the broadly-accepted

BAN logic [26] for 2PAKEP. Apart from these, we also perform the informal security analysis in order to verify the security of 2PAKEP so that the scheme will be secure with high probability. Moreover, we simulate 2PAKEP for formal security verification using the widely-used AVISPA tool [34] in Section VI.

A. FORMAL SECURITY USING ROR MODEL

This section proves 2PAKEP's session key security using the ROR model.

1) ROR MODEL

The authentication and key exchange phase of 2PAKEP involves two participants, namely, U and S . The main components related to the ROR model in 2PAKEP are briefly discussed below.

a: PARTICIPANTS

Let \mathcal{I}_U^t and \mathcal{I}_S^s be the instances t and s of U and S , respectively that are known as the *oracles*.

b: ACCEPTED STATE

If an instance \mathcal{I}^t is in an accept state after receiving the final protocol message, it will be in accepted state. The ordered concatenation of all communications (send and received messages by \mathcal{I}^t) forms the session identification (*sid*) of \mathcal{I}^t for the present session.

c: PARTNERING

Two instances \mathcal{I}^t and \mathcal{I}^s are the partners to each other once the following three criterion are satisfies concurrently: 1) \mathcal{I}^t and \mathcal{I}^s are in accepted state, 2) \mathcal{I}^t and \mathcal{I}^s mutually authenticate each other and share the same *sid*, and 3) \mathcal{I}^t and \mathcal{I}^s are mutual partners of each other.

d: FRESHNESS

If the established session key SK between U and S is not known through a reveal query *Reveal* defined below, \mathcal{I}_U^t or \mathcal{I}_S^s will be called fresh.

e: ADVERSARY

An adversary \mathcal{A} is modeled using the widely-accepted Dolev-Yao (DY) model as discussed in our defined threat model (Section I-A). Hence, \mathcal{A} can intercept, delete, modify, or even inject the messages exchanged between the involved participants U and S during the communication using the following accessed queries:

Execute($\mathcal{I}^t, \mathcal{I}^s$): The eavesdropping attack is modeled under this query that permits \mathcal{A} to intercept (read) the messages exchanged among U and S .

Send(\mathcal{I}^t, Msg): Under this query \mathcal{A} transmits a message Msg to a participant instance \mathcal{I}^t , and it also receives the response message from \mathcal{I}^t . It is modeled as an active attack.

Reveal(\mathcal{I}^t): This query reveals the session key SK created by \mathcal{I}^t (and its partner) to \mathcal{A} in the present session.

$CMD/CSC(\mathcal{I}_U^c)$: Under this corrupt mobile device or corrupt smart card query, \mathcal{A} can extract all the sensitive secret credentials stored in it. This is modeled as an active attack.

$Test(\mathcal{I}^t)$: Under this query, an unbiased coin c is flipped prior to beginning of the game. Depending on the output, the following decision is taken. \mathcal{A} executes this query and if the session key SK between U and S is fresh, \mathcal{I}^t returns SK in case $c = 1$ or a random number in case $c = 0$; otherwise, it will return a null value (\perp).

In our formal security analysis, a restriction is imposed on \mathcal{A} to access a limited number of $CMD/CSC(\mathcal{I}_U^c)$ queries, while \mathcal{A} is permitted to access an unlimited number of $Test(\mathcal{I}^t)$ queries.

f: SEMANTIC SECURITY

The indistinguishability of the actual session key SK from a random number by \mathcal{A} is essential under the semantic security. The output of $Test(\mathcal{I}^t)$ is examined for consistency against the random bit c . If \mathcal{A} 's guessed bit is c' and $Succ$ denotes the winning probability in the game, a polynomial t time adversary \mathcal{A} 's advantage in breaking the session key (SK) security of 2PAKEP is denoted and defined by $Adv_{2PAKEP}(t) = |2 \cdot Pr[Succ] - 1|$, where $Pr[\cdot]$ denotes the probability.

g: RANDOM ORACLE

2PAKEP applies the public one-way cryptographic hash function $h(\cdot)$. We model $h(\cdot)$ as a random oracle, say $Hash$.

Before proving the SK-security of 2PAKEP, we define the following computational problems. A collision-resistant one-way hash function $h(\cdot)$ is formally given below [41].

Definition 1 (Collision-Resistant One-Way Hash Function): A collision-resistant one-way hash function $h: \{0, 1\}^* \rightarrow \{0, 1\}^n$ is a deterministic function which inputs a variable length data and outputs a fixed length value, say n bits. If $Adv_{\mathcal{A}}^{HASH}(rt)$ is an adversary \mathcal{A} 's advantage in finding a hash collision, we have,

$$Adv_{\mathcal{A}}^{HASH}(rt) = Pr[(i_1, i_2) \in_R \mathcal{A} : i_1 \neq i_2, h(i_1) = h(i_2)],$$

where the pair $(i_1, i_2) \in_R \mathcal{A}$ indicates that the inputs i_1 and i_2 are randomly chosen by \mathcal{A} . An (η, rt) -adversary \mathcal{A} attacking the $h(\cdot)$'s collision resistance means that $Adv_{\mathcal{A}}^{HASH}(rt) \leq \eta$ and the runtime of \mathcal{A} is at most rt .

Let E/F_p be an elliptic curve over a finite field F_p and $P \in E_p(a, b)$ be a base point. The elliptic curve discrete logarithm problem (ECDLP) is defined as follows.

Definition 2 (Elliptic Curve Discrete Logarithm Problem (ECDLP)): Given P and $Q \in E/F_p$, to find the discrete logarithm d , where $Q = dP$ and dP is known as the elliptic curve point (scalar) multiplication, that is, $dP = P + P + \dots + P$ (d times).

The elliptic curve decisional Diffie-Hellman problem (ECDDHP) is a computational hard problem in ECC which is defined as follows.

Definition 3 (Elliptic Curve Decisional Diffie-Hellman Problem (ECDDHP)): Given a quadruple (P, k_1P, k_2P, k_3P) ,

decide whether $k_3 = k_1k_2$ or a uniform value, where $k_1, k_2, k_3 \in Z_p^*$.

The ECDLP and ECDDHP are computationally infeasible problems when p is large. To make ECDLP and ECDDHP intractable, p should be chosen at least 160-bit prime [39].

2) SECURITY PROOF

Theorem 1 proves the SK-security of 2PAKEP under the ROR model.

Theorem 1: If $Adv_{2PAKEP}(t)$ denotes the advantage function of an adversary \mathcal{A} running in polynomial-time t for breaking the SK-security of 2PAKEP, then

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

where $q_h, q_s, |Hash|, |D|$ and $Adv_{\mathcal{A}}^{ECDDHP}(t)$ are the number of $Hash$ queries, the number of $Send$ queries and the range space of the hash function $h(\cdot)$, the size of a uniformly distributed password dictionary D and the advantage of \mathcal{A} in breaking the ECDDHP in time t , respectively.

Proof: We follow the proof of this theorem as done in [31] and [33]. A sequence of five games, say G_j for $j = 0, 1, 2, 3, 4$ is needed. $Succ_{G_j}$ denotes the probability associated with the game G_j in which an adversary \mathcal{A} can win the game G_j and the advantage of \mathcal{A} in winning the game G_j is denoted and defined by $Adv_{G_j} = Pr[Succ_{G_j}]$. Each game G_j is described as follows.

- **Game G_0 :** This is the starting game in which \mathcal{A} selects the bit c . It is worth noticing that G_0 and the real protocol in the ROR model are identical to each other. Hence, we have,

$$Adv_{2PAKEP}(t) = |2 \cdot Adv_{G_0} - 1|. \quad (1)$$

- **Game G_1 :** This game implements the eavesdropping attack by \mathcal{A} . Under this game, \mathcal{A} uses $Execute$ query, and once the game is completed, \mathcal{A} makes the $Test$ query. The output of the $Test$ query is used to decide whether \mathcal{A} gets the actual session key SK or a random number. In 2PAKEP, SK is calculated by both U and S as $SK = kdf(ID_u || SK_u || T_u || T_s) = kdf(ID_u || SK_S || T_u || T_s) (= SK')$. Here, $SK_u = r_u R_S = r_u (r_S P) = r_S (r_u P) = SK_S$. In order to derive $SK (= SK')$, \mathcal{A} requires both temporal secrets r_u and r_S , and also the permanent secret ID_u . Hence, \mathcal{A} 's winning the G_1 is not increased by this eavesdropping attack. It is also worth noticing that both the games G_0 and G_1 are indistinguishable. Therefore, it follows that

$$Adv_{G_1} = Adv_{G_0}. \quad (2)$$

- **Game G_2 :** GM_2 : This game simulates the $Send$ and $Hash$ queries. This game is modeled as an active attack wherein \mathcal{A} intercepts all the messages $Msg_1 = \{Auth_u, CID_u, R_u, T_u\}$, $Msg_2 = \{Auth_S, R_S, T_S\}$ and $Msg_3 = \{Auth_{us}, T'_u\}$. Since all these messages involve the random nonces and current timestamps, there will

be no collision in hash outputs when \mathcal{A} makes the *Hash* queries (see Definition 1). Using the results from the birthday paradox, we obtain the following:

$$|Adv_{G_2} - Adv_{G_1}| \leq \frac{q_h^2}{2|Hash|}. \quad (3)$$

- *Game G₃*: This game implements the *CMD/CSC* query wherein \mathcal{A} can extract all the credentials l', v and C from the lost or stolen device or smart card of U , where $l' = l \oplus b_u = H_1(d_s) \oplus RPW \oplus H_2(d_s || ID_u)$, $v = RPW \oplus a_u$ and $C = H_2(ID_u || PW_u || a_u)$. To derive or guess the password PW_u and identity ID_u of a registered user U from l', v and C is computationally infeasible problem due to unknown secret credentials a_u and d_s using the *Send* queries. Since the games G_2 and G_3 are identical when the password guessing attack is not involved, it follows that

$$|Adv_{G_3} - Adv_{G_2}| \leq \frac{q_s}{|D|}. \quad (4)$$

- *Game G₄*: This is the final game which is modeled as an active attack. To derive the session key $SK = kdf(ID_u || SK_u || T_u || T_s) = kdf(ID_u || SK_s || T_u || T_s) (= SK')$, \mathcal{A} can use all the intercepted messages Msg_1, Msg_2 and Msg_3 , and then try to derive $SK_u = r_u R_s = r_u(r_s P) = r_s(r_u P) = SK_s$. \mathcal{A} can derive $SK_u = r_u R_s$ with the intercepted R_s or can derive $SK_s = r_s R_u$ with the intercepted R_u . However, this problem is essentially same as solving the ECDDHP (see Definition 3). Therefore, we have the following result:

$$|Adv_{G_4} - Adv_{G_3}| \leq Adv_{\mathcal{A}}^{ECDDHP}(t). \quad (5)$$

Since all the games are executed, it only remains for \mathcal{A} to guess the correct bit c . It then follows that

$$Adv_{G_4} = \frac{1}{2}. \quad (6)$$

(1) and (2) lead to the following result:

$$\begin{aligned} \frac{1}{2} Adv_{2PAKEP}(t) &= |Adv_{G_0} - \frac{1}{2}| \\ &= |Adv_{G_1} - \frac{1}{2}|. \end{aligned} \quad (7)$$

(6) and (7) also lead to the following result:

$$\frac{1}{2} Adv_{2PAKEP}(t) = |Adv_{G_1} - Adv_{G_4}|. \quad (8)$$

Using the triangular inequality, we have the following result:

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

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

$$\frac{1}{2} Adv_{2PAKEP}(t) \leq \frac{q_h^2}{2|Hash|} + \frac{q_s}{|D|} + Adv_{\mathcal{A}}^{ECDDHP}(t). \quad (10)$$

Final step is to multiply (10) on both sides by a factor of 2. After this step, rearranging the terms the required result is obtained:

$$Adv_{2PAKEP}(t) \leq \frac{q_h^2}{|Hash|} + 2\left(\frac{q_s}{|D|} + Adv_{\mathcal{A}}^{ECDDHP}(t)\right).$$

B. MUTUAL AUTHENTICATION PROOF USING BAN LOGIC

To prove that 2PAKEP achieves mutual authentication, we perform the BAN logic [26] analysis. First, we introduce the notations of the BAN logic in Table 2 and then define its logical postulates. Finally, we show that 2PAKEP provides mutual authentication among the user U and the server S .

TABLE 2. Notations of the BAN logic.

Notation	Description
A, B	Principals
C, D	Formulas
$A \equiv C$	A believes C
$\#C$	C is fresh
$A \triangleleft C$	A sees C
$A \sim C$	A once said C
$A \Rightarrow C$	A controls C
$\langle C \rangle_D$	C is combined with the formula D
$\{C\}_K$	C is encrypted by the key K
$A \stackrel{K}{\leftrightarrow} B$	A and B use the shared key K to communicate
SK	A session key used in the current session

1) LOGICAL POSTULATES OF BAN LOGIC

The logical postulates (rules) of the BAN logic are described below.

1. (Message meaning)

$$\frac{A | \equiv A \stackrel{K}{\leftrightarrow} B, \quad A \triangleleft \{C\}_K}{A | \equiv B | \sim C}$$

2. (Nonce verification)

$$\frac{A | \equiv \#(C), \quad A | \equiv B | \sim C}{A | \equiv B | \equiv C}$$

3. (Jurisdiction)

$$\frac{A | \equiv A | \implies C, \quad A | \equiv B | \equiv C}{A | \equiv C}$$

4. (Freshness)

$$\frac{A | \equiv \#(C)}{A | \equiv \#(C, D)}$$

5. (Belief)

$$\frac{A \mid \equiv (C, D)}{A \mid \equiv C.}$$

To conduct the BAN logic analysis, we first define the verification goals and idealized form of 2PAKEP. Then, we present our assumptions and demonstrate that 2PAKEP provides secure mutual authentication between U and S .

2) GOALS

The following goals are needed in 2PAKEP to prove that secure mutual authentication between U and S is achieved:

Goal 1: $U \mid \equiv (U \xleftrightarrow{SK} S)$

Goal 2: $S \mid \equiv (U \xleftrightarrow{SK} S)$

Goal 3: $U \mid \equiv S \mid \equiv (U \xleftrightarrow{SK} S)$

Goal 4: $S \mid \equiv U \mid \equiv (U \xleftrightarrow{SK} S)$

3) IDEALIZED FORMS

The idealized forms of the transmitted messages $Msg_1 = \{Auth_u, CID_u, R_u, T_u\}$, $Msg_2 = \{Auth_s, R_s, T_s\}$ and $Msg_3 = \{Auth_{us}, T'_u\}$ can be expressed as given below:

$Msg_1: U \rightarrow S: (ID_u, R, R_u, T_u)_{d_S}$

$Msg_2: S \rightarrow U: (ID_u, R^*, R_s, T_s)_{SK_S}$

$Msg_3: U \rightarrow S: (ID_u, R, T_u, T_s, T'_u)_{SK_u}$

4) ASSUMPTIONS

The following assumptions are taken into consideration:

$A_1: S \mid \equiv (U \xleftrightarrow{d_S} S)$

$A_2: S \mid \equiv \#(T_u)$

$A_3: U \mid \equiv (U \xleftrightarrow{SK_S} S)$

$A_4: U \mid \equiv \#(T_s)$

$A_5: S \mid \equiv (U \xleftrightarrow{SK_u} S)$

$A_6: S \mid \equiv \#(T'_u)$

$A_7: U \mid \equiv S \Rightarrow (U \xleftrightarrow{SK} S)$

$A_8: S \mid \equiv U \Rightarrow (U \xleftrightarrow{SK} S)$

5) PROOF USING BAN LOGIC

To achieve the above goals, we have the following steps:

Step 1: In accordance with Msg_1 , we can get:

$S_1: S \triangleleft (ID_u, R, R_u, T_u)_{d_S}$

Step 2: From S_1 and A_1 , we apply the message meaning rule to get:

$S_2: S \mid \equiv U \sim (ID_u, R, R_u, T_u)_{d_S}$

Step 3: In accordance with A_2 , we apply the freshness rule to obtain:

$S_3: S \mid \equiv \#(ID_u, R, R_u, T_u)_{d_S}$

Step 4: From S_2 and S_3 , we apply the nonce verification rule to obtain:

$S_4: S \mid \equiv U \mid \equiv (ID_u, R, R_u, T_u)_{d_S}$

Step 5: In accordance with Msg_2 , we can get:

$S_5: U \triangleleft (ID_u, R^*, R_s, T_s)_{SK_S}$

Step 6: From S_5 and A_3 , we apply the message meaning rule to obtain:

$S_6: U \mid \equiv S \sim (ID_u, R^*, R_s, T_s)_{SK_S}$

Step 7: In accordance with A_4 , we apply the freshness rule to obtain:

$S_7: U \mid \equiv \#(ID_u, R^*, R_s, T_s)_{SK_S}$

Step 8: From S_6 and S_7 , we apply the nonce verification rule to get:

$S_8: U \mid \equiv S \mid \equiv (ID_u, R^*, R_s, T_s)_{SK_S}$

Step 9: In accordance with Msg_3 , we can get:

$S_9: S \triangleleft (ID_u, R, T_u, T_s, T'_u)_{SK_u}$

Step 10: From S_5 and A_5 , we apply the message meaning rule to obtain:

$S_{10}: S \mid \equiv U \sim (ID_u, R, T_u, T_s, T'_u)_{SK_u}$

Step 11: In accordance with A_6 , we apply the freshness rule to obtain:

$S_{11}: S \mid \equiv \#(ID_u, R, T_u, T_s, T'_u)_{SK_u}$

Step 12: From to S_{10} and S_{11} , we apply the nonce verification rule to get:

$S_{12}: S \mid \equiv U \mid \equiv (ID_u, R, T_u, T_s, T'_u)_{SK_u}$

Step 13: Because of the session key $SK = kdf(ID_u || SK_S || T_u || T_s)$ and $SK = kdf(ID_u || SK_u || T_u || T_s)$, according to S_4, S_8, S_{12}, A_3 and A_5 , we can get:

$S_{13}: U \mid \equiv S \mid \equiv (U \xleftrightarrow{SK} S)$ (Goal 3)

and

$S_{14}: S \mid \equiv U \mid \equiv (U \xleftrightarrow{SK} S)$ (Goal 4)

Step 14: From S_{13} and A_7 , we apply the jurisdiction rule to obtain:

$S_{15}: U \mid \equiv (U \xleftrightarrow{SK} S)$ (Goal 1)

Step 15: From S_{14} and A_8 , we apply the jurisdiction rule to obtain:

$S_{16}: S \mid \equiv (U \xleftrightarrow{SK} S)$ (Goal 2)

Goals 1–4 prove that 2PAKEP achieves mutual authentication between U and S .

C. INFORMAL SECURITY ANALYSIS AND OTHER DISCUSSIONS

This section conducts the informal (non-mathematical) security analysis of 2PAKEP to demonstrate that it is secure against various other well-known attacks such as impersonation, password change, replay, privileged insider, and offline password guessing attacks. We also show that it ensures anonymity and mutual authentication.

1) IMPERSONATION ATTACK

For an adversary U_a trying to impersonate a legitimate user U , U_a has to know the precise real identity and password of U . According to Section III-A, if U_a obtains the hash value $H_1(d_S)$ using the parameter l , U_a cannot obtain the user's real identity ID_u because $CID_{uc} \oplus H_1(d_S)$ is $H_2(d_S || ID_u)$ in 2PAKEP. In other words, U_a cannot generate a valid login request message successfully without obtaining legitimate ID_u and PW_u . Therefore, 2PAKEP prevents impersonation attack.

2) PASSWORD CHANGE ATTACK

In the password change activity of 2PAKEP, we suppose that an adversary U_a tries to change the password of a legitimate user U . To change the password of U , U_a requires to compute $C' = H_2(ID_u || PW_u || a_u)$ correctly because the smart card checks whether C' is correct in the password change process with the stored C . However, because U_a cannot know ID_u , PW_u and a_u , U_a cannot change the password of U . For this reason, 2PAKEP resists password change attack.

3) OFFLINE PASSWORD GUESSING ATTACK

We assume that an adversary U_a could get the user's smart card and intercept previous transmitted messages. However, the parameters including password such as RPW , v , c and l' are hashed with other values. In addition, to guess the password correctly, U_a must know ID_u and a_u exactly. Therefore, due to collision-resistant property of the hash function (see Definition 1), 2PAKEP also prevents offline password guessing attack.

4) REPLAY ATTACK

If an adversary can try to reuse the transmitted messages in 2PAKEP, he/she cannot reuse messages because the transmitted messages include timestamps T . In addition, user U and server S verify the received timestamp T by the condition $|T^* - T| < \Delta T$, where the reception time of the message is T^* . Therefore, 2PAKEP resists replay attack

5) MAN-IN-THE-MIDDLE ATTACK

Suppose an adversary U_a intercepts the login request message $Msg_1 = \{Auth_u, CID_u, R_u, T_u\}$ and aims to modify this message to make another valid message, say $Msg'_1 = \{Auth_a, CID_u, R_a, T_a\}$ by generating fresh random number $r_a \in Z_p^*$ and generates the current timestamp T_a . Then, U_a can compute $R_a = r_a P$ and $R = r_a Q_S$. However, U_a cannot compute $Auth_a = H_2(ID_u || R || CID_u || T_a)$. In a similar argument, U_a cannot also modify other messages $Msg_2 = \{Auth_S, R_S, T_S\}$ and $Msg_3 = \{Auth_{us}, T'_u\}$. It is then clear that 2PAKEP resists man-in-the-middle attack.

6) PRIVILEGED INSIDER ATTACK

In the privileged insider attack, a privileged insider user of the trusted system being an attacker tries to obtain other user's

information in registration phase. This attack is considered as a critical security attack [27]–[29]. We suppose that an adversary U_a is a privileged insider of the S and becomes an insider attacker. In the registration phase of 2PAKEP, the user U computes $H_2(ID_u || PW_u) \oplus b_u$ and sends it along with ID_u to the server S . In other words, U_a cannot know the real password PW_u of the user U because $RPW \oplus b_u$ is hashed by a one-way hash function and is combined with b_u . Moreover, we also assume that after the registration process is completed, U_a has the lost/stolen smart card or mobile device, and extracts all the sensitive information l' , v and C stored in its memory, where $l' = l \oplus b_u = H_1(d_S) \oplus RPW \oplus H_2(d_S || ID_u)$, $v = RPW \oplus a_u$ and $C = H_2(ID_u || PW_u || a_u)$. Without knowing the secret credentials d_S and a_u , it is computationally infeasible to guess the correct password PW_u of the user U from l' and v . In addition, guessing the correct password PW_u of the user U from $C = H_2(ID_u || PW_u || a_u)$ is computationally infeasible without knowing the secret credential a_u . Therefore, 2PAKEP is robust to the privileged-insider attack.

7) ANONYMITY

If an adversary U_a could obtain a user's smart card and intercept previous transmitted messages, U_a cannot get the user's real identity ID_u . In the mutual authentication & key exchange phase of 2PAKEP, U sends the pseudo-identity $CID_u = l' \oplus RPW$ to the S . However, U_a cannot obtain the real identity ID_u of U because U_a cannot know the server's long-term private key d_S . In addition, S retrieves the real identity of U from the database using $H_2(d_S || ID_u)$. Therefore, 2PAKEP provides user anonymity property.

8) SECURE MUTUAL AUTHENTICATION AND SESSION KEY AGREEMENT

According to Section V-C.1, an adversary cannot generate the valid messages successfully during the authentication and key exchange phase. In addition, using the elliptic curve cryptosystem (ECC) $R_u = r_u P$ and $R_S = r_S P$ are computed, U and S check the correctness of the conditions $Auth_S = H_2(ID_u || R^* || SK_S)$ and $Auth_{us} = H_2(R || SK_u)$, respectively. Thus, 2PAKEP ensures secure mutual authentication and session key agreement.

9) EFFICIENT PASSWORD CHANGE MECHANISM

In Qi and Chen's scheme, when a mobile user U wants to change the password, he/she has to send password change request message to the server S . In other words, the computation and communication cost of this phase are similar to mutual authentication & key exchange phase in their scheme. However, in 2PAKEP, when a mobile user U wants to change the password, he/she can change the password PW_u freely without the server's assistance. First, U inserts the smart card, and then inputs the valid ID_u , PW_u and PW_{new} . Next, the smart card verifies the identity and password, and computes the parameters. Finally, because the smart card has already checked whether ID_u and PW_u are correct, the smart card changes the password successfully without any mistakes.

Therefore, the password change mechanism of 2PAKEP is more efficient than Qi and Chen's scheme, and it is achieved locally without involving the server S .

VI. FORMAL SECURITY VERIFICATION USING AVISPA TOOL: SIMULATION STUDY

This section evaluates the formal security verification of 2PAKEP using the popularly-accepted Automated Validation of Internet Security Protocols and Applications (AVISPA) tool [34]. AVISPA-based formal security verification has been attracted recently and is used in many authentication protocols to check whether a security protocol is resilient against replay and man-in-the-middle attacks [27]–[29], [31]–[33].

AVISPA has four back-ends: 1) On-the-fly Model-Checker (OFMC), 2) Constraint Logic based Attack Searcher (CL-AtSe), 3) SAT-based Model-Checker (SATMC) and 4) Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP). More detailed descriptions on these back-ends can be found in [34]. The security protocols need to be specified in HLPSL (High Level Protocols Specification Language) of AVISPA [35]. HLPSL is a role based language and contains the following roles [34], [35]:

- The basic roles denote the participating entities in the protocol.
- The composition roles denote the different scenarios involving basic roles.

In HLPSL, an intruder represented by i plays a legitimate role. The HLPSL specification of the protocol is converted to its intermediate format (IF) using the HLPSL2IF translator. After that the IF is converted to output format (OF) by feeding it to one of the four back-ends. The various sections of the OF are discussed in [35].

A. HLPSL SPECIFICATION OF 2PAKEP

2PAKEP has been implemented in HLPSL for the user registration as well as authentication and key exchange phases. The HLPSL specification for the role of the user U is provided in Figure 7. During the user registration phase, the user U , being the initiator, first receives the start signal and then generates two random numbers a_u and b_u , computes RPW and sends the registration request message $\{ID_u, (RPW \oplus b_u)\}$ to the server S through a secure channel. After that U gets the registration reply message $\{I\}$ from the S through a secure channel.

During the mutual authentication and key exchange phase, the user U generates random number r_u and current timestamp T_u , computes R_u, R, CID_u and $Auth_u$, and then sends the login request message $Msg_1 = \{Auth_u, CID_u, R_u, T_u\}$ to the S via open channel. After that U receives the authentication request message $Msg_2 = \{Auth_S, R_S, T_S\}$ from the responder S via open channel. Finally, U sends the authentication reply message $Msg_3 = \{Auth_{us}, T'_u\}$ to the S through public channel by generating current timestamp T'_u .

```

role user (U, S: agent, SKus: symmetric_key,
          Snd, Rcv: channel(dy))
% Player: the user U
played_by U
def=
local State: nat,
      IDu, PWu, RPW, Au, Bu, Ds, P: text,
      Ru, Rs, Tu, Ts, Tu1, R, Ru1: text,
      CIDu, Authu, SKu, SK, Auths: text,
% H1, H2: one-way hash functions
% F: ECC point multiplication
% Kdf: key derivation function
      H1, H2, F, Kdf: hash_func
const sp1, sp2, sp3, u_s_ru, u_s_tu,
      u_s_tu1, s_u_rs, s_u_ts: protocol_id
init State := 0
transition
% User registration phase
1. State = 0  $\wedge$  Rcv(start) =>
State' := 1  $\wedge$  RPW' := H2(IDu.PWu)
 $\wedge$  Au' := new()  $\wedge$  Bu' := new()
%% Identity IDu is shared between U and S
 $\wedge$  secret({IDu}, sp1, {U,S})
%% Password PWu is only known to U
 $\wedge$  secret({PWu}, sp2, {U})
%% Send registration request to S securely
 $\wedge$  Snd({IDu.xor(RPW'.Bu')}_SKus)
%% Receive registration reply {I} from S securely
2. State = 1  $\wedge$  Rcv({xor(xor(H1(Ds), xor(H2(IDu.PWu), Bu')),
H2(Ds.IDu))}_SKus) =>
%% Private key d_S is known to S
State' := 3  $\wedge$  secret({Ds}, sp3, {S})
%% Mutual authentication & key exchange phase
 $\wedge$  Ru' := new()  $\wedge$  Tu' := new()
 $\wedge$  Ru1' := F(Ru'.P)
 $\wedge$  R' := F(Ru'.F(Ds.P))
 $\wedge$  CIDu' := xor(H1(Ds), H2(Ds.IDu))
 $\wedge$  Authu' := H2(IDu.R'.CIDu'.Tu')
%% Send login request message Msg1 to S through open channel
 $\wedge$  Snd(Authu'.CIDu'.Ru1'.Tu')
% U has freshly generated random number r_u and timestamp Tu for S
 $\wedge$  witness(U, S, u_s_ru, Ru')
 $\wedge$  witness(U, S, u_s_tu, Tu')
%% Receive authentication request message Msg2 from S publicly
3. State = 3  $\wedge$  Rcv(H2(IDu.F(Ds.F(Ru'.P)).F(Rs'.F(Ru'.P)).Ts').
F(Rs'.P).Ts') =>
State' := 5  $\wedge$  Tu1' := new()
 $\wedge$  SKu' := F(Ru'.F(Rs'.P))
 $\wedge$  SK' := Kdf(IDu.SKu'.Tu'.Ts')
 $\wedge$  Auths' := H2(F(Ru'.F(Ds.P)).SK'.Tu1')
%% Send authentication reply message Msg3 to S publicly
 $\wedge$  Snd(Authus'.Tu1')
% U has freshly generated timestamp Tu' for S
 $\wedge$  witness(U, S, u_s_tu1, Tu1')
% U's acceptance of the values r_s and Ts generated for U by S
 $\wedge$  request(S, U, s_u_rs, Rs')
 $\wedge$  request(S, U, s_u_ts, Ts')
end role

```

FIGURE 7. Role specification for the user (U).

In Figure 7, the declarations $secret(\{IDu\}, sp1, \{U,S\})$, $secret(\{PWu\}, sp2, \{U\})$ and $secret(\{Ds\}, sp3, \{S\})$ indicate that the information ID_u is shared between both U and S , password PW_u is only known to U and the private key d_S is known to S , respectively. Here, $sp1$, $sp2$ and $sp3$ are the protocol ids. The declarations $witness(U, S, u_s_ru, Ru')$, $witness(U, S, u_s_tu, Tu')$ and $witness(U, S, u_s_tu, Tu1')$ tell that U has freshly generated the random number r_u , timestamps T_u and T'_u for S , respectively and these are

maintained by the protocol ids u_s_ru , u_s_tu and u_s_tu1 , respectively. Other declarations $request(S, U, s_u_rs, Rs')$ and $request(S, U, s_u_ts, Ts')$ mean that U 's acceptance of r_s and T_s generated for U by S , respectively. In a similar way, the role for the server S is implemented and shown in Figure 8.

```

role server (U, S: agent, SKus: symmetric_key,
            Snd, Rcv: channel(dy))
% Player: the server S
played_by S
def=
local State: nat,
    IDu, PWu, Au, Bu, Ds, L, P: text,
    Ru, Rs, Rs1, Tu, Tu1, Ts, R2, SKs, Auths: text,
    H1, H2, F, Kdf: hash_func
const sp1, sp2, sp3, u_s_ru, u_s_tu,
    u_s_tu1, s_u_rs, s_u_ts: protocol_id
init State := 0
transition
% User registration phase
1. State = 0  $\wedge$  Rcv({IDu.xor(H2(IDu.PWu),Bu')})_SKus =>
%%% Identity IDu is shared between U and S
State' := 2  $\wedge$  secret({IDu}, sp1, {U,S})
%%% Password PWu is only known to U
 $\wedge$  secret({PWu}, sp2, {U})
%%% Private key d_S is known to S
 $\wedge$  secret({Ds}, sp3, {S})
 $\wedge$  L' := xor(xor(H1(Ds), xor(H2(IDu.PWu),Bu')),
            H2(Ds.IDu))
%%% Send registration reply {1} to U securely
 $\wedge$  Snd({L'}_SKus)
%%% Mutual authentication & key exchange phase
%%% Receive login request message Msg1 from U publicly
2. State = 2  $\wedge$  Rcv(H2(IDu.F(Ru'.F(Ds.P).xor(H1(Ds),
            H2(Ds.IDu)).Tu')),xor(H1(Ds),
            H2(Ds.IDu)).F(Ru'.P).Tu') =>
State' := 4  $\wedge$  Rs' := new()  $\wedge$  Ts' := new()
 $\wedge$  Rs1' := F(Rs'.P)
 $\wedge$  R2' := F(Ds.F(Ru'.P))
 $\wedge$  SKs' := F(Rs'.F(Ru'.P))
 $\wedge$  Auths' := H2(IDu.R2'.SKs'.Ts')
%%% Send authentication request message Msg2 to U publicly
 $\wedge$  Snd(Auths'.Rs1'.Ts')
% S has freshly generated random number r_s and timestamp Ts for U
 $\wedge$  witness(S, U, s_u_rs, Rs')
 $\wedge$  witness(S, U, s_u_ts, Ts')
%%% Receive authentication reply message Msg3 from U publicly
3. State = 4  $\wedge$  Rcv(H2(F(Ru'.F(Ds.P)).Kdf(IDu.F(Ru'.F(Rs'.P)).
            Tu'.Ts')).Tu1').Tu1') =>
% S's acceptance of values r_u, Tu and Tu' generated for S by U
State' := 6  $\wedge$  request(U, S, u_s_ru, Ru')
 $\wedge$  request(U, S, u_s_tu, Tu')
 $\wedge$  request(U, S, u_s_tu1, Tu1')
end role

```

FIGURE 8. Role specification for the server (S).

Figure 9 shows the definitions for necessary roles - *session*, *goal* and *environment*. In the session segment, all the basic roles: *user* and *server* are instanced with concrete arguments. The top-level role (environment) specifies in the specification of HLPSSL, which contains the global constants and a composition of one or more sessions, where the intruder (i) plays some roles as legitimate users. The intruder also participates in the execution of protocol as a concrete session. The current version of HLPSSL supports the standard authentication and secrecy goals. In our implementation, three secrecy goals and five authentications are checked as shown in Figure 9.

```

role session (U, S: agent, SKus: symmetric_key)
def=
local Snd1, Snd2, Rcv1, Rcv2: channel (dy)
composition
    user (U, S, SKus, Snd1, Rcv1)
 $\wedge$  server (U, S, SKus, Snd2, Rcv2)
end role

role environment()
def=
const u, s: agent, skus: symmetric_key,
    h1, h2, f, kdf: hash_func, tu, tu1, ts: text,
    u_s_ru, u_s_tu, u_s_tu1, s_u_rs, s_u_ts,
    sp1, sp2, sp3: protocol_id
intruder_knowledge = {u, s, h1, h2, f, kdf, tu, ts, tu1}
composition
    session(u, s, skus)
 $\wedge$  session(i, s, skus)
 $\wedge$  session(u, i, skus)
end role
goal
%%% Confidentiality (privacy)
secrecy_of sp1, sp2, sp3
%%% Authentication
authentication_on u_s_ru, u_s_tu, u_s_tu1
authentication_on s_u_rs, s_u_ts
end goal
environment()

```

FIGURE 9. Role specification for the session, goal and environment.

B. ANALYSIS OF SIMULATION RESULTS

2PAKEP is simulated using the widely-used OFMC and CL-AtSe backends under the SPAN, the Security Protocol ANimator for AVISPA tool [36]. 2PAKEP makes use of the bitwise XOR operations. Since SATMC and TA4SP backends do not implement XOR operations at present, the simulation results of 2PAKEP under these backends become as “inconclusive,” and due to this reason, the simulation results under SATMC and TA4SP backends have been ignored in this paper.

The following three verifications are essential for 2PAKEP:

- Executability checking on non-trivial HLPSSL specifications
- Replay attack checking
- Dolev-Yao model checking

The executability check is essential to ensure that the protocol can reach to a state where a possible attack can happen, during the run of the protocol. From Figures 7 and 8, it is shown that 2PAKEP is properly translated to HLPSSL specification and it meets the design goals by ensuring the executability. 2PAKEP is simulated for the execution tests and a bounded number of sessions model checking. For replay attack checking, both OFMC and CL-AtSe backends verify if the legal agents can execute the specified protocol by performing a search of a passive intruder. Moreover, both OFMC and CL-AtSe backends verify the occurrence of any man-in-the-middle attack possible by i for the Dolev-Yao model checking.

The simulation results for both OFMC and CL-AtSe backends are reported in Figure 10. The OFMC backend takes 0.1 seconds search time, while it visits 36 nodes with a

% OFMC % Version of 2006/02/13 SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL C:\progra~1\SPAN\testsuite results\auth.if GOAL as_specified BACKEND OFMC COMMENTS STATISTICS parseTime: 0.00s searchTime: 0.10s visitedNodes: 36 nodes depth: 4 plies	SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL C:\progra~1\SPAN\testsuite results\auth.if GOAL As Specified BACKEND CL-AtSe STATISTICS Analysed : 4 states Reachable : 4 states Translation: 0.13 seconds Computation: 0.00 seconds
---	--

FIGURE 10. Analysis of simulation results using the OFMC & CL-AtSe backends.

depth of 4 plies, whereas CL-AtSe backend analyzes 4 states and it takes 0.13 seconds translation time, and all 4 states are reachable. Therefore, all verifications, such as executability checking on non-trivial HLPSP specifications, replay attack checking and Dolev-Yao model checking are fulfilled in 2PAKEP. As a result, 2PAKEP becomes safe against both replay & man-in-the-middle attacks.

VII. PERFORMANCE COMPARATIVE STUDY

This section compares the performance of our proposed scheme (2PAKEP) with other related schemes [11], [13], [25].

A. COMPARISON OF SECURITY AND FUNCTIONALITY FEATURES

Various security and functionality features of 2PAKEP are compared with other related schemes in Table 3. From this table, it is evident that the existing schemes are vulnerable to various attacks. In addition, the existing schemes cannot provide mutual authentication, anonymity and also efficient password change mechanism. Therefore, 2PAKEP provides better security and functionality features as compared to those for the other related schemes [11], [13], [25].

TABLE 3. Comparison of security and functionality features.

Property	He et al. [11]	Yang et al. [13]	Qi and Chen [25]	2PAKEP
SF_1	×	○	×	○
SF_2	○	○	×	○
SF_3	○	○	×	○
SF_4	×	×	×	○
SF_5	○	○	×	○
SF_6	○	○	○	○
SF_7	×	×	×	○
SF_8	×	○	×	○
SF_9	×	×	×	○

SF_1 : impersonation attack; SF_2 : password change attack; SF_3 : offline password guessing attack; SF_4 : privileged insider attack; SF_5 : replay attack; SF_6 : man-in-the-middle attack; SF_7 : user anonymity; SF_8 : mutual authentication; SF_9 : efficient password change facility
○: preserves the security properties; ×: does not preserve the security properties;

B. COMPARISON OF COMPUTATION OVERHEADS

We compare the computation overheads of different schemes in practical environment. We use the existing experimental results as reported in He et al.'s scheme [11]. All the cryptographic operations were implemented with a standard cryptographic library, known as MIRACLE [37]. The platform was a PIV 3-GHZ processor with 512-MB memory with Windows XP operation system. We use the following notations for analysis of computation overheads among different schemes:

- T_h : time taken for a cryptographic hash operation
- T_{XOR} : time needed for an bitwise XOR operation
- T_{ECC} : time taken for an elliptic curve point multiplication operation
- T_{MAC} : time required for a message authentication code (MAC) operation
- T_{inv} : time taken for a modular inversion operation
- T_{kdf} : time needed for a key derivation function

The execution times for different cryptographic operations are shown in Table 4. Since the bitwise XOR operation is negligible, we have ignored this operation from our comparative study on computation overheads. It is also assumed that $T_{kdf} \approx T_h$.

TABLE 4. Different cryptographic operations time (in milliseconds) [11].

Entity	T_{ECC}	T_{inv}	T_h	T_{MAC}
Server	0.83	0.13	< 0.0001	< 0.0001
Client	12.08	1.89	< 0.001	< 0.001

In Table 5, we compare the computation overheads of 2PAKEP with other related schemes [11], [13], [25] during the authentication phase based on the experimental results shown in Table 4. 2PAKEP needs $6T_h + 2T_{XOR} + 3T_{ECC} + 1T_{kdf} \approx 36.247$ ms for the user U side and $4T_h + 3T_{XOR} + 3T_{ECC} + 1T_{kdf} \approx 2.4905$ ms for the server S side. The computation overheads for other schemes for the user and server sides are also shown in Table 5. The results reported in Table 5 clearly indicate that 2PAKEP is comparable with other schemes in terms of computation overheads. Though the computation overhead required for Yang et al.'s scheme [13] is less than that for 2PAKEP, Yang et al.'s scheme is vulnerable to various attacks and also it does not support other functionality features (see Table 3).

C. COMPARISON OF COMMUNICATION OVERHEADS

Finally, we compare the communication overheads of 2PAKEP with other related schemes in Table 6 during the authentication phase. For this purpose, we assume that an identity is 160 bits, the hash output (message digest) using the SHA-1 hashing algorithm [38] is 160 bits, a random number (nonce) is 160 bits, timestamp is 32 bits and an elliptic curve point $P = (P_x, P_y)$ requires $(160 + 160) = 320$ bits assuming that an 160-bit ECC provides the same bit-security level of an 1024-bit RSA [39]; P_x and P_y are the x and y co-ordinates of P , respectively.

TABLE 5. Comparison of computation overheads.

	He et al. [11]	Yang et al. [13]	Qi and Chen [25]	2PAKEP
User	$2T_h + 2T_{MAC} + 3T_{ECC}$ ≈ 36.248 ms	$3T_h + 2T_{ECC}$ ≈ 24.163 ms	$4T_h + 3T_{XOR} + 3T_{ECC} + 1T_{kdf}$ ≈ 36.245 ms	$6T_h + 2T_{XOR} + 3T_{ECC} + 1T_{kdf}$ ≈ 36.247 ms
Server	$3T_h + 2T_{MAC} + 3T_{ECC} + 1T_{inv}$ ≈ 2.6205 ms	$4T_h + 3T_{ECC}$ ≈ 2.4904 ms	$4T_h + 1T_{XOR} + 3T_{ECC} + 1T_{kdf}$ ≈ 2.4905 ms	$4T_h + 3T_{XOR} + 3T_{ECC} + 1T_{kdf}$ ≈ 2.4905 ms

TABLE 6. Comparison of communication overheads.

Scheme	No. of messages	No. of bits
He et al. [11]	2	1344
Yang et al. [13]	3	1344
Qi and Chen [25]	3	1280
2PAKEP	3	1376

During the authentication and key exchange phase, 2PAKEP needs exchange of three messages $Msg_1 = \{Auth_u, CID_u, R_u, T_u\}$, $Msg_2 = \{Auth_s, R_s, T_s\}$ and $Msg_3 = \{Auth_{us}, T'_u\}$ which need $(160 + 160 + 320 + 32) = 672$ bits, $(160 + 320 + 32) = 512$ bits and $(160 + 32) = 192$ bits, respectively. The total communication cost of 2PAKEP is then $(672 + 512 + 192) = 1376$ bits for three messages. On the other hand, the schemes of He et al., Yang et al., and Qi and Chen need 1344 bits, 1344 bits and 1280 bits, respectively. It is also worth noticing that the communication overhead required for 2PAKEP is comparable to that for other schemes [11], [13], [25].

VIII. CONCLUDING REMARKS

In this paper, we demonstrated that Qi and Chen's scheme is vulnerable to various attacks such as impersonation, password change, offline password guessing and privileged insider attacks, and it cannot also provide anonymity and mutual authentication. To resolve these security weaknesses, we propose a secure and efficient two-party authentication key exchange protocol for mobile environment. 2PAKEP prevents various attacks because it is hiding user's real identity from an adversary using secret parameters. In addition, 2PAKEP guarantees anonymity, secure mutual authentication and efficient password change mechanism. We also proved that 2PAKEP provides secure mutual authentication between U and S using BAN logic and the session key security using ROR model through formal security analysis. 2PAKEP is secure through the formal security verification using the widely-used AVISPA simulated tool. Furthermore, we analyze the performance comparison with other related schemes and it is shown that 2PAKEP performs well in terms of security and functionality features requirements, and its performance in terms of communication and computation overheads are also comparable with other related schemes.

As a result, 2PAKEP can be applicable to mobile environment efficiently.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the Associate Editor for their valuable feedback on the paper which helped us to improve its quality and presentation.

REFERENCES

- [1] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, Nov. 1981.
- [2] S. M. Bellare and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks," in *Proc. IEEE Comput. Soc. Symp. Res. Secur. Privacy*, Oakland, CA, USA, May 1992, pp. 72–84.
- [3] M.-S. Hwang, C.-C. Chang, and Y.-L. Tang, "A simple remote user authentication scheme," *Math. Comput. Model.*, vol. 36, nos. 1–2, pp. 103–107, 2002.
- [4] S.-W. Lee, H.-S. Kim, and K.-Y. Yoo, "Improvement of Chien et al.'s remote user authentication scheme using smart cards," *Comput. Standards Interfaces*, vol. 27, no. 2, pp. 181–183, 2005.
- [5] I.-E. Liao, C.-C. Lee, and M.-S. Hwang, "A password authentication scheme over insecure networks," *J. Comput. Syst. Sci.*, vol. 72, no. 4, pp. 727–740, Jun. 2006.
- [6] J. Xu, W.-T. Zhu, and D.-G. Feng, "An improved smart card based password authentication scheme with provable security," *Comput. Standards Interfaces*, vol. 31, no. 4, pp. 723–728, 2009.
- [7] H.-Y. Chien, J.-K. Jan, and Y.-M. Tseng, "An efficient and practical solution to remote authentication: Smart card," *Comput. Secur.*, vol. 21, no. 4, pp. 372–375, 2002.
- [8] M. Peyravian and C. Jeffries, "Secure remote user access over insecure networks," *Comput. Commun.*, vol. 29, no. 5, pp. 660–667, 2006.
- [9] G. Yang, D. S. Wong, H. Wang, and X. Deng, "Two-factor mutual authentication based on smart cards and passwords," *J. Comput. Syst. Sci.*, vol. 74, no. 7, pp. 1160–1172, 2008.
- [10] J.-H. Yang and C.-C. Chang, "An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *Comput. Secur.*, vol. 28, nos. 3–4, pp. 138–143, 2009.
- [11] D. He, C. Jianhua, and H. Jin, "An ID-based client authentication with key agreement protocol for mobile client-server environment on ECC with provable security," *Inf. Fusion*, vol. 13, no. 3, pp. 223–230, 2012.
- [12] C.-H. Chou, K.-Y. Tasi, and C.-F. Lu, "Two ID-based authenticated schemes with key agreement for mobile environments," *J. Supercomput.*, vol. 66, no. 2, pp. 973–988, 2013.
- [13] H. Yang, J. Chen, and Y. Zhang, "An improved two-party authentication key exchange protocol for mobile environment," *Wireless Pers. Commun.*, vol. 85, no. 3, pp. 1399–1409, 2015.
- [14] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.
- [15] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.
- [16] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology*, vol. 1666. Berlin, Germany: Springer-Verlag, 1999, pp. 388–397.

- [17] Y. Park, S. Lee, C. Kim, and Y. Park, "Secure biometric-based authentication scheme with smart card revocation/reissue for wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 7, pp. 1–11, 2016.
- [18] J. Moon, Y. Choi, J. Jung, and D. Won, "An improvement of robust biometrics-based authentication and key agreement scheme for multi-server environments using smart cards," *PLoS ONE*, vol. 10, no. 12, pp. 1–15, 2015.
- [19] E.-J. Yoon and K.-Y. Yoo, "Robust ID-based remote mutual authentication with key agreement scheme for mobile devices on ECC," in *Proc. Int. Conf. Comput. Sci. Eng.*, Vancouver, BC, Canada, Aug. 2009, pp. 633–640.
- [20] B.-L. Chen, W.-C. Kuo, and L.-C. Woo, "Robust smart-card-based remote user password authentication scheme," *Int. J. Commun. Syst.*, vol. 27, no. 2, pp. 377–389, 2014.
- [21] Q. Jiang, J. Ma, G. Li, and X. Li, "Improvement of robust smart-card-based password authentication scheme," *Int. J. Commun. Syst.*, vol. 28, no. 2, pp. 383–393, 2015.
- [22] K. Park, Y. Park, Y. Park, A. G. Reddy, and A. K. Das, "Provably secure and efficient authentication protocol for roaming service in global mobility networks," *IEEE Access*, vol. 5, pp. 25110–25125, 2017.
- [23] C.-T. Li, "A more secure and efficient authentication scheme with roaming service and user anonymity for mobile communications," *Inf. Technol. Control*, vol. 41, no. 1, pp. 69–76, 2012.
- [24] C.-T. Li, T.-Y. Wu, C.-L. Chen, C.-C. Lee, and C.-M. Chen, "An efficient user authentication and user anonymity scheme with provably security for IoT-based medical care system," *Sensors*, vol. 17, no. 7, pp. 1482, 2017.
- [25] M. Qi and J. Chen, "An efficient two-party authentication key exchange protocol for mobile environment," *Int. J. Commun. Syst.*, vol. 30, no. 16, pp. 1–8, 2017.
- [26] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990.
- [27] A. K. Das, M. Wazid, N. Kumar, M. K. Khan, K.-K. R. Choo, and Y. Park, "Design of secure and lightweight authentication protocol for wearable devices environment," *IEEE J. Biomed. Health Informat.*, to be published, doi: [10.1109/JBHI.2017.2753464](https://doi.org/10.1109/JBHI.2017.2753464).
- [28] M. Wazid, A. K. Das, N. Kumar, M. Conti, and A. V. Vasilakos, "A novel authentication and key agreement scheme for implantable medical devices deployment," *IEEE J. Biomed. Health Informat.*, to be published, doi: [10.1109/JBHI.2017.2721545](https://doi.org/10.1109/JBHI.2017.2721545).
- [29] S. Challa et al., "Secure signature-based authenticated key establishment scheme for future IoT applications," *IEEE Access*, vol. 5, pp. 3028–3043, 2017.
- [30] S. Roy, A. K. Das, S. Chatterjee, N. Kumar, S. Chattopadhyay, and J. J. P. C. Rodrigues, "Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications," *IEEE Trans. Ind. Informat.*, to be published, doi: [10.1109/TII.2018.2824815](https://doi.org/10.1109/TII.2018.2824815).
- [31] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "Cloud centric authentication for wearable healthcare monitoring system," *IEEE Trans. Dependable Secure Comput.*, to be published, doi: [10.1109/TDSC.2018.2828306](https://doi.org/10.1109/TDSC.2018.2828306).
- [32] S. Challa, A. K. Das, P. Gope, N. Kumar, F. Wu, and A. V. Vasilakos, "Design and analysis of authenticated key agreement scheme in cloud-assisted cyber-physical systems," *Future Gener. Comput. Syst.*, to be published, doi: [10.1016/j.future.2018.04.019](https://doi.org/10.1016/j.future.2018.04.019).
- [33] A. Dua, N. Kumar, A. K. Das, and W. Susilo, "Secure message communication protocol among vehicles in smart city," *IEEE Trans. Veh. Technol.*, to be published, doi: [10.1109/TVT.2017.2780183](https://doi.org/10.1109/TVT.2017.2780183).
- [34] AVISPA. *Automated Validation of Internet Security Protocols and Applications*. Accessed: Jan. 2018. [Online]. Available: <http://www.avispa-project.org/>
- [35] D. von Oheimb, "The high-level protocol specification language hpls developed in the eu project avispa," in *Proc. 3rd APPSEM II Workshop Appl. Semantics (APPSEM)*, Frauenchiemsee, Germany, 2005, pp. 1–17.
- [36] AVISPA. *SPAN: Security Protocol Animator for AVISPA*. Accessed: Jan. 2018. [Online]. Available: <http://www.avispa-project.org/>
- [37] *MIRACL Cryptographic SDK: Multiprecision Integer and Rational Arithmetic Cryptographic Library*. Accessed: Jan. 2018. [Online]. Available: <http://github.com/miracl/MIRACL>
- [38] *Secure Hash Standard, Standard FIPS PUB 180-1*, National Institute of Standards and Technology, Apr. 1995. Accessed: Jan. 2018. [Online]. Available: <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>
- [39] R. L. Rivest, M. E. Hellman, J. C. Anderson, and J. W. Lyons, "Responses to NIST's proposal," *Commun. ACM*, vol. 35, no. 7, pp. 50–52, 1992.
- [40] M. Abdalla, P. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. 8th Int. Workshop Theory Pract. Public Key Cryptogr. (PKC)*, 2005, pp. 65–84.
- [41] P. Sarkar, "A simple and generic construction of authenticated encryption with associated data," *ACM Trans. Inf. Syst. Secur.*, vol. 13, no. 4, 2010, Art. no. 33.



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 Electronics Engineering. His research interests include authentication, computer networks, Internet of Things, post-quantum cryptography, VANET, and information security.



YOUNGHO PARK (M'17) received the B.S., M.S., and Ph.D. degrees in electronic engineering from 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, USA.

He is currently a Professor with the School of Electronics Engineering, Kyungpook National University. His research interests include computer networks, multimedia, 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 Division of IT Convergence, Information and Communication Department, Korea Nazarene University. His research interests include computer networks, mobile security, and information security.



ASHOK KUMAR DAS (M'17) received the Ph.D. degree in computer science and engineering, the M.Tech. degree in computer science and data processing, and the M.Sc. degree in mathematics from IIT Kharagpur, Kharagpur, India. He is currently an Associate Professor with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. His current research interests include cryptography, wireless sensor network

security, hierarchical access control, security in vehicular ad hoc networks, smart grid, Internet of Things (IoT), cyber-physical systems and cloud computing, and remote user authentication. He has authored over 160 papers in international journals and conferences in the abovementioned areas including 140+ reputed journal papers. Some of his research findings are published in the top-cited journals such as the IEEE Transactions on Information Forensics and Security, the IEEE Transactions on Dependable and Secure Computing, the IEEE Transactions on Smart Grid,

the IEEE Internet of Things Journal, the IEEE Transactions on Industrial Informatics, the IEEE Transactions on Vehicular Technology, the IEEE Transactions on Consumer Electronics, the IEEE Journal of Biomedical and Health Informatics (formerly, the IEEE Transactions on Information Technology in Biomedicine), the *IEEE Consumer Electronics Magazine*, the IEEE Access, the IEEE Communications Magazine, the *Future Generation Computer Systems*, the *Computers & Electrical Engineering*, the *Computer Methods and Programs in Biomedicine*, the *Computer Standards & Interfaces*, the *Computer Networks*, the *Expert Systems With Applications*, and the *Journal of Network and Computer Applications*. He received the Institute Silver Medal from IIT Kharagpur. He is on the editorial board of the *KSI Transactions on Internet and Information Systems*, the *International Journal of Internet Technology and Secured Transactions* (Inderscience), and the *Recent Advances in Communications and Networking Technology*. He is a Guest Editor of the *Computers & Electrical Engineering* (Elsevier) for the special issue on big data and IoT in e-healthcare. He served as a program committee member for many international conferences.

• • •