Efficient Authenticated Key Exchange Protocols for Large-Scale Mobile Communication Networks

Run-hua Shi(✉) and Shun Zhang

School of Computer Science and Technology, Anhui University,
Hefei City 230601, China
shirh@ahu.edu.cn, shzhang27@163.com

Abstract. For secure communications in mobile communication networks, various authenticated key exchange schemes are proposed to provide the remote client authentication and the session key establishment. In these schemes, more considerations are to reduce the costs of remote mobile clients, but not those of the server. However, the server has become a bottleneck in large-scale mobile communication networks. In this paper, in order to relieve the server's load, we presented an efficient authentication protocol with key exchange between the remote client and the server, and then generalized it to a three-party case, in which two remote clients can authenticate each other with the server's help and share a secure session key. Compared with the relevant protocols, the proposed protocols require lower computation and communication costs, and above all, dramatically reduce those of the server. Therefore, the proposed protocols are more practical and suitable for large-scale mobile communication networks.

Keywords: Elliptic curve cryptography · Authentication · Key exchange · Client-server network

1 Introduction

Secure remote client authentication with key exchange over insecure communication channel is an important issue for many applications in client-server networks, especially electronic transactions (e.g., on-line shopping, Internet banking and pay-TV). On the other hand, Mobile communication recently has become more pervasive with the popularity of mobile devices, such as smart phones, handheld game consoles, personal digital assistants, and mobile internet devices, etc. Therefore, client authentication with key exchange for mobile communication environments is becoming the focus of widely attentions.

For mobile devices, to reduce the computation loads, some authenticated key exchange schemes based on elliptic curve cryptosystem (ECC) were proposed [1–3]. As we know, security of ECC is based upon the difficulty of elliptic curve discrete logarithm problem (ECDLP) and elliptic curve Diffie-Hellman problem (ECDHP) [4, 5]. Compared with traditional public key cryptosystems (e.g., RSA [6] and ElGamal [7]), ECC offers a better performance because it achieves the same security with a

J. Wan et al. (Eds.): CloudComp 2016, SPNCE 2016, LNICST 197, pp. 204–215, 2018.
https://doi.org/10.1007/978-3-319-69605-8_19

smaller key size. For example, 160-bit ECC and 1024-bit RSA have the same security level in practice [8].

However, early remote client authentication schemes on ECC are based on public-key cryptosystem, in which the public key in the system requires the associated certificate to prove its validity and thus clients need additional computations to verify the other's certificate. To avoid the weakness, in 2009, Yang and Chang [9] proposed an identity-based remote user mutual authentication scheme for mobile users using elliptic curve cryptography. However, Yoon and Yoo [10] demonstrated that Yang and Chang's protocol is vulnerable to the impersonation attack and does not provide perfect forward secrecy, and then proposed an improved protocol. Later, He *et al.* [11] again confirmed that Yoon and Yoo's protocol does not provide perfect forward secrecy yet and fails to achieve forward secrecy. In addition, they also pointed out that a special hash function called MapToPoint function which is used to map an identity information into a point on elliptic curve is required in the previous protocols. To improve the efficiency, they presented a new remote user authentication protocol without the MapToPoint function [11]. Subsequently, there appeared more improved protocols of authenticated key establishment for client-server networks [12–16].

The above two-party authenticated key exchange (2PAKE for short) schemes can achieve two secure goals of mutual authentication and key exchange between the remote client and the server. However, these schemes are infeasible to establish a secure session key between any two remote clients in client-server networks. For this, there appeared some three-party authenticated key exchange (3PAKE for short) schemes to provide mutual authentication and key establishment between two remote clients with the help of the server. In 2009, Yang and Chang [17] proposed an efficient identity-based 3PAKE scheme to improve the security of Chen *et al.*'s scheme [18]. In 2010, Tan [19] demonstrated that Yang and Chang's 3PAKE protocol is vulnerable to the impersonation and parallel attacks, and proposed an improved scheme. However, Nose [20] pointed that Tan's scheme suffers from the impersonation attack and the man-in-the-middle attack. Later, He et al. [21] pointed out that Yang *et al.*'s scheme and Tan's scheme are also based on the public key infrastructure (PKI), and then the users need additional computations to verify the other's certificate. Therefore, He *et al.* proposed an improved identity-based 3PAKE scheme to improve these drawbacks. Furthermore, Chou *et al.* [22] again pointed out that a user cannot verify the correctness for his/her private key in these schemes mentioned above, and then proposed two authenticated key exchange schemes with private key confirmation. However, Farash and Attari [23] showed that Chou *et al.*'s 2PAKE scheme is vulnerable to the impersonation attack and the key-compromise impersonation attack, and their 3PAKE scheme is also insecure against the impersonation attack. To overcome the weaknesses, Farash and Attari presented an improved identity-based 2PAKE protocol using elliptic curves [23]. However, we found that there is still a serious security flaw in the user registration phase of the two schemes [24]: any authorized user can impersonate the server to generate the effective private key of any other unauthorized user.

In existing 2PAKE and 3PAKE schemes for mobile communication environments, the authors always find ways to reduce the computation and communication costs of the remote client/user, such that their respective schemes are feasible to mobile users with limited resources. However, when hundreds of thousands of remote users simultaneously request the server to authenticate their identities and establish the secure session keys, the server's load is very heavy. In fact, the server has becomes a bottleneck in many practical applications for large-scale client-server networks. Therefore, how to reduce the server's load in authenticated key exchange schemes is a practical and important issue.

Most current 2PAKE and 3PAKE schemes are various generalizations of Diffie-Hellman Key Exchange [25], in which two parties fair complete the same computations and communications. However, in the client-server networks, the remote client/user and the server are not peer entities, where the server is trustable. Therefore, there are redundant computations in these authentication schemes. In this paper, we exploited new methods to construct an efficient identity-based 2PAKE protocol, which is especially suitable for large-scale client-server networks. In addition, we extended the 2PAKE protocol to develop a 3PAKE protocol, which allows two remote users to share a secure session key with the server's help. Compared with other relevant protocols, our proposed protocols need lower computation and communication costs, and especially relieve the server's load.

2 Proposed Protocols

2.1 The Proposed 2PAKE Protocol

The proposed 2PAKE protocol includes three phases: Initialization, User Registration, and Mutual Authentication with Key Exchange.

Initialization

The server S generates system parameters as follows:

1. S chooses an elliptic curve equation $E_p(a,b)$ [26] defined on finite field F_p, where p be a large prime.
2. S selects a base point P with the order q over $E_p(a,b)$, where q is a large prime for the security considerations.
3. S random generates its private key $k_S \in_R \mathbb{Z}_q^*$ and computes the corresponding public key $Q_S = k_S P$.
4. In addition, the server chooses a secure hash function, $H : \{0,1\}^* \to \mathbb{Z}_q^*$.
   Then the server publishes these system parameters: $\{p, \boldsymbol{E}_p(a,b), q, \boldsymbol{P}, \boldsymbol{Q}_S, H(\cdot)\}$.

User Registration

1. The user U sends his identity, ID_U, to the server S. Then S checks the authenticity and legality of his identity.
2. After confirming the authenticity and legality of the user, S computes U's private key $k_U = (H(ID_U) \oplus k_S) + H(ID_U)[k_S + (H(ID_U) \oplus k_S)]$.
3. S computes $Q_{ID_U} = (H(ID_U) \oplus k_S)P$ and sends $\{k_U, Q_{ID_U}\}$ to U.
4. After receiving $\{k_U, Q_{ID_U}\}$, U verifies if $k_U P = Q_{ID_U} + H(ID_U)(Q_S + Q_{ID_U})$. If the equation holds, U keeps k_U in secret as his private key.

Mutual Authentication with Key Exchange

In this phase, a user U and the server S authenticate each other and establish a common session key for the later communications. This phase is divided into two rounds which are shown as follows.

Round 1

1. U randomly chooses $r_U \in \mathbb{Z}_q^*$ and $S_U \in \mathbb{Z}_p^*$, and computes $R_U = r_U P$, $V_U = S_U \oplus f_x(r_U k_U Q_S)$ and $h_U = H(ID_U||S_U||t_U)$, where t_U is a timestamp that denotes the current time. Please note that $f_x(Q)$ and $f_y(Q)$ denote the x and y coordinates of the point Q, respectively.
2. U sends $\{ID_U, R_U, V_U, h_U, t_U\}$ to the server S.

Round 2

1. After receiving $\{ID_U, R_U, V_U, h_U, t_U\}$, S verifies if t_U is valid. If t_U is not fresh, S aborts the process and sends the failed messages to U; otherwise, he continues to execute the next step.
2. S computes $k_U = (H(ID_U) \oplus k_S) + H(ID_U)[k_S + (H(ID_U) \oplus k_S)]$ and $S_U^{'} = V_U \oplus f_x(k_U k_S R_U)$, where k_S is S's private key.
3. S verifies if $h_U = H(ID_U||S_U^{'}||t_U)$. If it is true, S confirms that U is an authorized user; otherwise, he aborts the process.
4. S computes the session key $k_{SU} = H(S_U^{'}||t_U)$.
5. S computes $MAC_{k_{SU}}(t_U)$ as the response and sends $MAC_{k_{SU}}(t_U)$ to U, where $MAC_{k_{SU}}(t_U)$ denotes Message Authentication Code of the timestamp t_U by the session key k_{SU}.
6. After receiving $MAC_{k_{SU}}(t_U)$, U computes the session key $k_{US} = H(S_U||t_U)$, and then checks the integrity of $MAC_{k_{SU}}(t_U)$ by the session key k_{US}. U will quit the current session if the check produces a negative result; otherwise, U authenticates the server S and uses k_{US} as the session key with S in future communications.

2.2 The Proposed 3PAKE Protocol

Similarly, the proposed protocol includes three phases: Initialization, User Registration, and Mutual Authentication with Key Agreement. The first two phases are similar to those of the proposed 2PAKE protocol, accordingly. Here, we mainly describe the last phase as follows:

Mutual Authentication with Key Agreement

In this phase, two users A and B authenticate each other with the server S's help and negotiate a common session key for the later communications. Suppose that A and B have obtained their respective private keys, $k_A = (H(ID_A) \oplus k_S) + H(ID_A)\ [k_S + (H(ID_A) \oplus k_S)]$ and $k_B = (H(ID_B) \oplus k_S) + H(ID_B)[k_S + (H(ID_B) \oplus k_S)]$. This phase is divided into three rounds which are described in detail as follows.

Round 1

1. A randomly chooses $r_A \in \mathbb{Z}_q^*$, and computes $R_A = r_A P$, $V_A = f_x(r_A k_A Q_S)$ and $h_A = H(ID_A||V_A||t_A)$, where t_A is a timestamp that denotes the current time.
2. A sends $\{ID_A$, request$\}$ and $\{ID_A, ID_B, R_A, h_A, t_A\}$ to B and S, respectively. The message request denotes a request that A asks B to agree on a session key.

Round 2

1. After receiving $\{ID_A$, request$\}$, B randomly selects $r_B \in \mathbb{Z}_q^*$, and computes $R_B = r_B P$, $V_B = f_x(r_B k_B Q_S)$ and $h_B = H(ID_B||V_B||t_B)$, where t_B is the current timestamp.
2. B sends $\{ID_B$, response$\}$ and $\{ID_B, ID_A, R_B, h_B, t_B\}$ to A and S, respectively. The message response denotes a response that B accepts A's request.

Round 3

1. After receiving $\{ID_A, ID_B, R_A, h_A, t_A\}$ and $\{ID_B, ID_A, R_B, h_B, t_B\}$, S verifies if both t_A and t_B are valid. If t_A or t_B is not fresh, S aborts the process and sends the failed messages to the users; otherwise, he continues to perform the next step.
2. S computes $k_A = (H(ID_A) \oplus k_S) + H(ID_A)[k_S + (H(ID_A) \oplus k_S)]$ and $k_B = (H(ID_B) \oplus k_S) + H(ID_B)[k_S + (H(ID_B) \oplus k_S)]$. Furthermore, S computes $V_A^{'} = f_x(k_A k_S R_A)$ and $V_B^{'} = f_x(k_B k_S R_B)$.
3. S verifies if $h_A = H(ID_A||V_A^{'}||t_A)$ and $h_B = H(ID_B||V_B^{'}||t_B)$. If both of them are true, S confirms that A and B are all authorized users; otherwise, he/she aborts the process.
4. S computes $h_{SA} = H(ID_A||ID_B||f_x(R_B)||f_y(R_B)||V_A^{'}||t_S)$ and $h_{SB} = H(ID_B||ID_A|| f_x(R_A)||f_y(R_A)||V_B^{'}||t_S)$, where t_S is the current timestamp. S sends $\{R_B, h_{SA}, t_S\}$ and $\{R_A, h_{SB}, t_S\}$ to A and B, respectively.
5. After receiving $\{R_B, h_{SA}, t_S\}$, A verifies if t_S is valid. If t_S is not fresh, A aborts the processes; otherwise, A performs the next step.
6. A computes $h_{SA}^{'} = H(ID_A||ID_B||f_x(R_B)||f_y(R_B)||V_A||t_S)$.
7. A verifies if the equation of $h_{SA}^{'} = h_{SA}$ holds. If it holds, A believes that S is the authentic server, and further confirms that B is authenticated by S. Then he/she can obtain the session key shared between A and B by computing $k_{AB} = H(f_x(r_A R_B)||f_y(r_A R_B))$; otherwise, A rejects the transaction.
8. Similarly, after receiving $\{R_A, h_{SB}, t_S\}$, B verifies if t_S is valid. If t_S is not fresh, B aborts the processes; otherwise, B continues to execute the next step.
9. B computes $h_{SB}^{'} = H(ID_B||ID_A||f_x(R_A)||f_y(R_A)||V_B||t_S)$.
10. B verifies if the equation of $h_{SB}^{'} = h_{SB}$ holds. If it holds, B believes that S is the authentic server, and further confirms that A is authenticated by S. Then he/she can obtain the session key shared between B and A by computing $k_{BA} = H(f_x(r_B R_A)|| f_y(r_B R_A))$; otherwise, B rejects the transaction.

3 Analysis

We first analyze the security of the proposed protocols against various known cryptographic attacks. The security of our protocols relies on the difficulties of solving Elliptic Curve Discrete Logarithm (ECDL) problem (Given two points $\boldsymbol{P}$ and $\boldsymbol{Q}$ over an elliptic curve $\boldsymbol{E}_p(a,b)$, it is computationally infeasible to find an integer k such that $\boldsymbol{Q} = k \cdot \boldsymbol{P}$) and Elliptic Curve Computational Diffie-Hellman (ECCDH) problem (Given three points $\boldsymbol{P}$, $a \cdot \boldsymbol{P}$ and $b \cdot \boldsymbol{P}$ over $\boldsymbol{E}_p(a,b)$, it is computationally infeasible to compute a point $\boldsymbol{W}$ such that $\boldsymbol{W} = ab \cdot \boldsymbol{P}$). Then, we give Performance comparisons of some related protocols.

3.1 Security Analysis

In this section, we mainly analyze that the proposed protocols can withstand various related security attacks. In our scheme, since the server is trusted and further all privates are generated by the server's private key, we assume that the server's private key is secure. Otherwise, the whole system will be controlled by the attacker and thus it will not make any sense to again discuss the system security.

Theorem 1 (Replay Attack Resistance). The proposed 2PAKE and 3PAKE protocols can resist the replay attack.

Proof. In the proposed protocols, the receiver can always verify the freshness of the received messages by the freshness of the timestamp t. Furthermore, the timestamp t is embedded in the hashed message (e.g., $h_U = H(ID_U||S_U||t_U)$) by the sender, such that it can guarantee the integrity of the timestamp. Therefore, the proposed scheme can resist the replay attack.

Theorem 2 (Known-key security). The proposed 2PAKE and 3PAKE protocols satisfy the known key security. That is, an outsider cannot compute the current session key even he knows some previous session keys.

Proof. In our 2PAKE/3PAKE protocol, the session key $k_{US} = H(S_U||t_U)/k_{AB} = H(f_x(r_A r_B P)||f_y(r_A r_B P))$ is obtained by computing a secure hash function. Obviously, the session key depends on the short-term secret $S_U/(r_A,\ r_B)$, instead of the long-term secret $k_U/(k_A,\ k_B)$. Furthermore, each session has different short-term secret $S_U/(r_A,\ r_B)$, which is/are randomly generated. Thus the current session key is independent of the previous session. That is, an outsider cannot compute the current session key even he knows some previous session keys. Therefore, the known-key attack is infeasible for the proposed protocols.

Theorem 3 (Perfect forward secrecy). The proposed 2PAKE and 3PAKE protocols achieve perfect forward security. That is, the compromise of the long-term private keys of both the participating users does not affect the security of the previous session keys.

Proof. In our 2PAKE protocol, in order to successfully compute the session key $k_{US} = H(S_U||t_U)$, the most critical step is to obtain $\mathrm{k_U k_S r_U P}$ and then compute $S_U = V_U \oplus f_x(k_U k_S r_U P)$ rightly. If U's private key k_U is compromised to an attacker, it is still computationally hard for the attacker to compute $k_U k_S r_U P$ based on the difficulty of solving ECCDH Problem since he does not know k_S and r_U. Similarly, in our 3PAKE protocol, even if the private keys, k_A and k_B, of users A and B, are compromised to an attacker, it is also computationally hard for the attacker to compute $r_A r_B P$ based on the difficulty of solving ECCDH Problem since he does not know r_A and r_B. Therefore, the proposed protocols can provide perfect forward secrecy.

Theorem 4 (Key-compromise impersonation resistance). The proposed 2PAKE and 3PAKE protocols provide resistance to key-compromise impersonation attack. That is, even though the remote user's long-term private key is compromised, an adversary, who obtained the private key, cannot masquerade the other user or the server and obtain the resulting session key.

Proof. In our proposed protocols, the participant authentication mainly depends on if the sender/receiver can compute $k_U k_S r_U P$ rightly. Even if the remote user U's private key k_U is compromised to an attacker, it is still computationally hard for the attacker to compute $k_{U'} k_S r_{U'} P$ or $k_U k_S r_U P$ without knowing $\{k_S, k_{U'}\}$ or $\{k_S, r_U\}$, where $k_{U'}$ and k_S are the private keys of the other user $U^{'}$ and the server, respectively. That is, even though the remote user U's long-term private key is compromised, the attacker cannot masquerade the other user $U^{'}$ or the server to obtain the resulting session key. Therefore, the proposed protocols can resist key-compromise impersonation attack.

Theorem 5 (Unknown key-share resistance). The proposed 2PAKE and 3PAKE protocols provide resistance to unknown-key share attack.

Proof. A party A believes the key is shared with another party B, and a party C believes the key is shared with A. The above condition is called unknown key share. Our proposed 2PAKE/3PAKE schemes can obviously withstand the unknown-key share attack because the user's identity is authenticated by the server S.

Theorem 6 (Mutual authentication). The proposed 2PAKE and 3PAKE protocols achieve the property of mutual authentication.

Proof. In our 2PAKE protocol, the server S authenticates the remote user U by computing $S_U^{'} = V_U \oplus f_x(k_U k_S R_U)$ and verifying if $h_U = H(ID_U||S_U^{'}||t_U)$ holds, that is, the server S authenticates the user U by checking if he/she knows the private key,$k_U = (H(ID_U) \oplus k_S) + H(ID_U)[k_S + (H(ID_U) \oplus k_S)]$. In turn, the user U authenticates the server S by comparing the received $MAC_{k_{SU}}(t_U)$ to the result computed by him/herself, because the server S is the only one who can recover S_U from V_U and then computes the session key k_{SU} and $MAC_{k_{SU}}(t_U)$. Similarly, in our 3PAKE protocol, two users A and B authenticate the server S by checking if he can rightly compute $f_x(k_A k_S R_A)$ and $f_x(k_B k_S R_B)$ from their respective sent messages, and then authenticate each other by the help of the trusted server S who authenticates A and B by verifying their respective private keys.

In addition, our proposed protocols can provide the confirmation of the user's private key, which doesn't rely on the digital signature technology. Though the authors in References [22, 23] claimed that their protocols could provide the confirmation of the user's private key, there is a serious security flaw in their respective protocols [24]:

any authorized user can impersonate the server to generate the effective private key of any other unauthorized user, because it can't guarantee the integrity of the public information, Q_{ID_U}. In our protocol, we do not embed Q_{ID_U} into a hash function to ensure its integrity. Otherwise, it will increase the costs of the server, because he has to again compute $Q_{ID_U}(Q_{ID_U} = (H(ID_U) \oplus k_S)P)$ to obtain the user's private key in the mutual authentication phase. Here, we introduce two items of $H(ID_U) \oplus k_S$ in the equation of generating the user's private key, and ensure the equation has obvious architectural features, such that it is infeasible to modify Q_{ID_U} and successfully pass the user's check. To sum up, the good features in these two schemes are still hold in our scheme, and further we can cover the shortage of the impersonation attack.

3.2 Performance Comparisons

We have analyzed the security of the proposed protocols in the above section. Furthermore, we give security comparisons of our protocols and other related works, as shown in Tables 1 and 2.

Table 1. Security comparisons for 2PAKE protocols

	He *et al.*'s protocol [11]	Islam and Biswas's protocol [15]	Yoon *et al.*'s protocol [12]	Chou *et al.*'s protocol [22]	Farash and Attari's protocol [23]	Our 2PAKE protocol
Mutual authentication	Provided	Provided	Provided	Provided	Provided	Provided
Known-key security	Provided	Provided	Provided	Provided	Provided	Provided
Forward secrecy	Provided	Provided	Not provided	Provided	Provided	Provided
Private key confirmation	Not provided	Not provided	Not provided	Insecure	Insecure	Secure
Impersonation attack	Insecure	Secure	Secure	Insecure	Secure	Secure
Key-compromise impersonation attack	Secure	Secure	Secure	Insecure	Secure	Secure
Unknown-key share attack	Secure	Secure	Secure	Secure	Secure	Secure
User registration	Secure	Secure	Secure	Insecure	Insecure	Secure

In addition, we evaluate the performance of our proposed protocols in terms of the computation and communication costs, and list performance comparisons for 2PAKE protocols and 3PAKE protocols in Tables 3 and 4, respectively. Same as References [22, 23] we assume the timestamp length is 16-bit, the size of p used in the ECC is 160-bit, the digest message size of hash function (e.g., SHA-1) or message authentication

Table 2. Security comparisons for 3PAKE protocols

	He *et al.*'s protocol [21]	Tan's protocol [19]	Yang and Chang's protocol [17]	Chou *et al.*'s protocol [22]	Our 3PAKE protocol
Known-key security	Provided	Provided	Provided	Provided	Provided
Forward secrecy	Provided	Provided	Provided	Provided	Provided
Private key confirmation	Not provided	Not provided	Not provided	Insecure	Secure
Impersonation attack	Secure	Insecure	Insecure	Insecure	Secure
Key-compromise impersonation attack	Secure	Secure	Secure	Insecure	Secure
Unknown-key share attack	Secure	Secure	Secure	Secure	Secure
Parallel attack	Secure	Secure	Insecure	Secure	Secure
User registration	Secure	Secure	Secure	Insecure	Secure

code is 160-bit, and the identity size is 80-bit. Please note that there are some wrong evaluations in References [22, 23]: the size of a point on the ECC is 320-bit, not 160-bit, since the size of p used in the ECC is 160-bit; the size of the cipher text of the symmetric encryption/decryption is the same size of the plain text, not 128-bit. To estimate and compare the computation costs, we define the following notations: PM, PA, H, MAC, Hp, I, $E(n)$ and $D(n)$ are the time complexity of elliptic curve scalar point multiplication, elliptic curve point addition, one-way hash function, message authentication code, map-to-point hash function, modular inversion, symmetric encryption for n-bit plain text and symmetric decryption for n-bit cipher text, respectively.

According to Tables 1, 2, 3 and 4, the proposed protocols have some good advantages as follows:

(1) The proposed protocols can withstand all related security attacks.
(2) The proposed protocols are identity-based authentication protocols with key exchange using ECC.
(3) The proposed protocols can provide the confirmation of the user's private key, where the cost of the private-key confirmation is lower than that of general digital signature.
(4) The proposed protocols require lower costs in both communication and computation complexity. Especially, the costs of the server in proposed protocols are lowest in all relevant protocols.

Therefore, the proposed protocols are more practical and suitable for large-scale mobile communication networks.

Table 3. Performance comparison for 2PAKE protocols

	Communication costs	Computation costs	
		User	Server
He *et al.*'s protocol [11]	1152 bits	$3PM + 2H + 2MAC$	$3PM + 3H + 2MAC + 1I$
Islam and Biswas's protocol [15]	1440 bits	$3PM + 2PA + 4H$	$4PM + 2PA + 1Hp + 4H$
Yoon *et al.*'s protocol [12]	1072 bits	$3PM + 2PA + 5H$	$4PM + 2PA + 1HP + 5H$
Chou *et al.*'s protocol [22]	1232 bits	$3PM + 3H$	$3PM + 5H$
Farash and Attari's protocol [23]	1232 bits	$3PM + 4H$	$3PM + 6H$
Our 2PAKE protocol	896 bits	$2PM + 2H + 1MAC$	$1PM + 3H + 1MAC$

Table 4. Performance comparison for 3PAKE protocols

	Communication costs	Computation costs	
		User	Server
He *et al.*'s protocol [21]	2464 bits	$3PM + 3H$	$2PM + 6H + 2I$
Tan's protocol [19]	4224 bits	$4PM + 1E(816) + 1D(816)$	$2PM + 2E(816) + 2D(816)$
Yang and Chang's protocol [17]	3680 bits	$5PM + 1E(640) + 1D(640)$	$2PM + 2E(640) + 2D(640)$
Chou *et al.*'s protocol [22]	2464 bits	$3PM + 2H$	$2PM + 8H$
Our 3PAKE protocol	2464 bits	$3PM + 3H$	$2PM + 6H$

4 Conclusion

In this paper, we presented two efficient authentication protocols in client-server networks, where one provides mutual authentication and key exchange between the remote user and the server, and the other achieves mutual authentication and key exchange between any two remote users with the help of the server. Compared with the relevant protocols, the proposed protocols obtain higher efficiencies, and especially relieve the burden of the server. Therefore, the proposed protocols are more practical and more suitable for large-scale mobile communication networks.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China (61173187, 11301002), Natural Science Foundation of Anhui Province (1408085QF107), Talents Youth Fund of Anhui Province Universities (2013SQRL006ZD), 211 Project of Anhui University (17110099).

References

1. Abi-Char, P.E., El-Hassan, B., Mhamed, A.: A fast and secure elliptic curve based authenticated key agreement protocol for low power mobile communications. In: Proceedings of the 2007 International Conference on Next Generation Mobile Applications, Services and Technologies, pp. 235–240. IEEE, New York (2007)
2. Chen, Z.G., Song, X.X.: A distributed electronic authentication scheme based on elliptic curve. In: Proceedings of the Sixth International on Machine Learning and Cybernetics, pp. 2179–2182. IEEE, New York (2007)
3. Jiang, C., Li, B., Xu, H.: An efficient scheme for user authentication in wireless sensor networks. In: Proceedings of 21st International Conference on Advanced Information Networking and Applications Workshops, pp. 438–442. IEEE, New York (2007)
4. Miller, V.S.: Use of elliptic curves in cryptography. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 417–426. Springer, Heidelberg (1986). doi:10.1007/3-540-39799-X_31
5. Koblitz, N.: Elliptic curve cryptosystem. Math. Comput. **48**, 203–209 (1987)
6. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
7. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. IT **31**, 469–472 (1985)
8. Hankerson, D., Menezes, A., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer Professional Computing. LNCS. Springer, New York (2004). doi:10.1007/b97644
9. Yang, J.H., Chang, C.C.: An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. Comput. Secur. **28**(3–4), 138–143 (2009)
10. Yoon, E.J., Yoo, K.Y.: Robust ID-based remote mutual authentication with key agreement protocol for mobile devices on ECC. In: Proceeding of 2009 International Conference on Computational Science and Engineering, vol. 02, pp. 633–640. IEEE Computer Society, Washington, DC, USA (2009)
11. He, D., Chen, J., Hu, J.: An ID-based client authentication with key agreement protocol for mobile client–server environment on ECC with provable security. Inf. Fusion **13**(3), 223–230 (2012)
12. Yoon, E.J., Choi, S.B., Yoo, K.Y.: A secure and efficiency ID-based authenticated key agreement scheme based on elliptic curve cryptosystem for mobile devices. Int. J. Innov. Comput. Inf. Control **8**(4), 2637–2653 (2012)
13. He, D.: An efficient remote user authentication and key agreement protocol for mobile client-server environment from pairings. Ad Hoc Netw. **10**, 1009–1016 (2012)
14. Wang, D., Ma, C.G.: Cryptanalysis of a remote user authentication scheme for mobile client-server environment based on ECC. Inf. Fusion **14**(4), 498–503 (2013)
15. Islam, S.K.H., Biswas, G.P.: A more efficient and secure ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. J. Syst. Softw. **84**(11), 1892–1898 (2011)
16. Karuppiah, M., Saravanan, R.: A secure remote user mutual authentication scheme using smart cards. J. Inf. Secur. Appl. **19**, 282–294 (2014)
17. Yang, J.H., Chang, C.C.: An efficient three-party authenticated key exchange protocol using elliptic curve cryptography for mobile-commerce environments. J. Syst. Softw. **82**(9), 1497–1502 (2009)
18. Chen, T.H., Lee, W.B., Chen, H.B.: A round-and computation-efficient three-party authenticated key exchange protocol. J. Syst. Softw. **81**(9), 1581–1590 (2008)

19. Tan, Z.: An enhanced three-party authentication key exchange protocol for mobile commerce environments. J. Commun. **5**(5), 436–443 (2010)
20. Nose, P.: Security weaknesses of authenticated key agreement protocols. Inf. Process. Lett. **111**(14), 687–696 (2011)
21. He, D., Chen, Y., Chen, J.: An ID-based three-party authenticated key exchange protocol using elliptic curve cryptography for mobile-commerce environments. Comput. Eng. Comput. Sci. **38**, 2055–2061 (2013)
22. Chou, C.H., Tsai, K.Y., Lu, C.F.: Two ID-based authenticated schemes with key agreement for mobile environments. J. Supercomput. **66**(2), 973–988 (2013)
23. Farash, M.S., Attari, M.A.: A secure and efficient identity-based authenticated key exchange protocol for mobile client-server networks. J. Supercomput. **69**(1), 395–411 (2014)
24. Shi, R.H., Zhong, H., Zhang, S.: Comments on two schemes of identity-based user authentication and key agreement for mobile client-server networks. J. Supercomput. **71**(11), 4015–4018 (2015)
25. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theory IT **22**(6), 644–654 (1976)
26. Yao, A.C.C., Zhao, Y.: Privacy-preserving authenticated key-exchange over internet. IEEE Trans. Inf. Forensics Security **9**(1), 125–140 (2014)