Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Efficient and secure three-party authenticated key exchange protocol for mobile environments*

Chih-ho CHOU[1], Kuo-yu TSAI[†‡2], Tzong-chen WU[1], Kuo-hui YEH[3]

([1]*Department of Information Management, National Taiwan University of Science and Technology, Taiwan 106, Taipei*)

([2]*Department of Management Information Systems, Hwa Hsia Institute of Technology, Taiwan 235, New Taipei*)

([3]*Department of Information Management, National Dong Hwa University, Taiwan 974, Hualien*)

[†]E-mail: kytsai@cc.hwh.edu.tw

Received Sept. 22, 2012; Revision accepted Jan. 13, 2013; Crosschecked Apr. 18, 2013

Abstract: Yang and Chang (2009) proposed a three-party authenticated key exchange protocol for securing communications in mobile-commerce environments. Their protocol reduces computation and communication costs by employing elliptic curve cryptosystems. However, Tan (2010) pointed out that Yang and Chang (2009)'s protocol cannot withstand impersonation and parallel attacks, and further proposed an enhanced protocol to resist these attacks. This paper demonstrates that Tan (2010)'s approach still suffers from impersonation attacks, and presents an efficient and secure three-party authenticated key exchange protocol to overcome shown weaknesses.

Key words: Three-party, Key exchange, Authentication, Mobile environments
doi:10.1631/jzus.C1200273 **Document code:** A **CLC number:** TP393; TN929.5

1 Introduction

In mobile environments, a user can share information with others using mobile devices, such as personal digital assistants (PDAs), smart phones, and laptops. However, an attacker can jam or intercept the wireless signal to obtain private or sensitive information. Designing a secure protocol is critical to communication in mobile environments. A key exchange protocol can establish a session key for securing the subsequent communication between two communicating parties (Cagalj *et al*., 2006).

Diffie and Hellman (1976) introduced the first key exchange protocol. Since then, various key exchange protocols have been developed. A three-party key exchange (3PAKE) protocol permits three communicating entities to share a session key in public environments. Generally, 3PAKE protocols can also be divided into two types: without a server (Hölbl *et al*., 2010) and with a server (Lee *et al*., 2005; Lu and Cao, 2007; Guo *et al*., 2008; Lee and Chang, 2008; Yoon and Yoo, 2008; Padmavathy, 2010). The former is a so-called tripartite protocol, and can be used to establish a session key between three participants. The session key is used to encrypt the transmitted data for ensuring data confidentiality. In general, a 3PAKE protocol without a server is another kind of multi-party key agreement protocol, and is suitable in some practical scenarios, e.g., electronic commerce. In the latter type of protocol, two users can establish a session key with a trusted server's aid. A 3PAKE protocol can be accomplished in several applications. For example, a buyer and a seller perform an online transaction with the assistance of a trusted server.

Chen *et al*. (2008) provided a 3PAKE protocol based on Schnorr (1989)'s digital signature scheme. Their protocol achieved mutual authentication and key exchange between two communicating entities in

‡ Corresponding author

* Project (Nos. 101-2218-E-011-001, 100-2218-E-259-004-MY2, and 101-2219-E-011-004) partially supported by the Taiwan Information Security Center (TWISC), National Science Council (NSC), Taiwan

fewer communication rounds. Subsequently, Yang and Chang (2009) pointed out that Chen *et al.* (2008)'s protocol cannot withstand the stolen-verifier attack (Menezes *et al.*, 1996), and proposed a 3PAKE protocol based on elliptic curve cryptosystems (ECCs) (Miller, 1986; Koblitz, 1987). Yang and Chang (2009)'s protocol provides resistance to stolen-verifier, man-in-the-middle, and stolen-verifier attacks. In addition, their protocol lowers communication costs. However, Tan (2010) showed that Yang and Chang (2009)'s protocol was susceptible to impersonation-of-initiator, impersonation-of-responder, and parallel attacks.

1. In an impersonation-of-initiator attack, an attacker can imitate the requester to request communication with the responder.

2. In an impersonation-of-responder attack, an attacker can imitate the responder to establish communication with the requester.

3. An attacker can successfully mount a parallel attack such that two communicating entities cannot establish the same session keys.

Tan (2010) proposed an enhanced protocol to resolve the above weaknesses. In this paper, we demonstrate that Tan (2010)'s protocol is still vulnerable to impersonation attacks, and propose a secure 3PAKE protocol to support the secure communication between two mobile entities. In our proposed scheme, two mobile entities can establish a session key with the server's aid, and the session key is used for securing subsequent communication. The proposed protocol not only withstands numerous attacks, but also provides better performance.

2 Review of previous protocols

2.1 Yang and Chang (2009)'s 3PAKE protocol

Yang and Chang (2009) proposed a 3PAKE protocol to improve Chen *et al.* (2001)'s protocol. Their protocol includes three roles: a user A (called session-initiator), a user B (called session-responder), and a trusted server S, and could be divided into two phases: the initialization phase and the authenticated key exchange phase.

2.1.1 Initialization phase

The trusted server S initializes and generates system parameters, and publishes $E_p(a, b)$, $E_k(\cdot)/D_k(\cdot)$, and the point P. The notations of parameters used throughout this paper are shown in Table 1.

Table 1 Parameters and descriptions

Parameter	Description
F_p	A finite field F_p over a large odd prime $p>2^{160}$
$E_p(a, b)$	An elliptic curve equation $E_p(a, b)$: $y^2=x^3+ax+b \pmod p$ with order n over F_p, where n is a large number
$E_k(\cdot)/D_k(\cdot)$	Secure symmetric encryption/decryption algorithms (e.g., advanced encryption standard), where k is the symmetric key
P	A base point P of the order n over $E_p(a, b)$
$\mathrm{sk_S}/\mathrm{PK_S}$	The server's private/public key pair, where $\mathrm{PK_S}=\mathrm{sk_S}P$
$\mathrm{sk_{id}}/\mathrm{PK_{id}}$	User's private/public key pair, where $\mathrm{PK_{id}}=\mathrm{sk_{id}}P$. Each user needs to register to the server to derive their private/public key pair
SK	Session key between two users
Z^*_p	The reduced set of residues modulo p. Z^*_p consists of all integers in [1, 2, ..., $p-1$] that are relative prime to p

2.1.2 Authenticated key exchange phase

If user A wants to communicate with user B, both users want to authenticate each other with server S's aid and establish a session key SK. This phase is divided into three rounds as follows.

Round 1:

1. A chooses a random number $r_{\mathrm{A}}\in Z^*_p$ to compute $R_{\mathrm{A}}=r_{\mathrm{A}}\cdot\mathrm{PK_A}$, $\hat{R}_{\mathrm{A}}=r_{\mathrm{A}}\cdot\mathrm{PK_S}$, and the temporary key $K_{\mathrm{A}}=\mathrm{sk_A}\cdot\hat{R}_{\mathrm{A}}=(k_{\mathrm{A}x}, k_{\mathrm{A}y})$, where $k_{\mathrm{A}x}$ and $k_{\mathrm{A}y}$ denote x and y coordinates of K_{A} over $E_p(a, b)$, respectively.

2. A chooses a random number $w_{\mathrm{A}}\in Z^*_p$ to compute $W_{\mathrm{A}}=w_{\mathrm{A}}\cdot B$, $C_{\mathrm{A}}=E_{k_{\mathrm{A}x}}(R_{\mathrm{A}}, W_{\mathrm{A}})$, and then sends ($\mathrm{ID_A}$, request) and ($\mathrm{ID_A}$, $\mathrm{ID_B}$, C_{A}, R_{A}) to B and S, respectively. Here, 'request' means that A invites B to establish a session key.

Round 2:

1. Upon receiving the message ($\mathrm{ID_A}$, request), B chooses a random number $r_{\mathrm{B}}\in Z^*_p$ to compute $R_{\mathrm{B}}=r_{\mathrm{B}}\cdot\mathrm{PK_B}$, $\hat{R}_{\mathrm{B}}=r_{\mathrm{B}}\cdot\mathrm{PK_S}$, and the temporary key $K_{\mathrm{B}}=\mathrm{sk_B}\cdot\hat{R}_{\mathrm{B}}=(k_{\mathrm{B}x}, k_{\mathrm{B}y})$.

2. B chooses a random number $w_{\mathrm{B}}\in Z^*_p$ to compute $W_{\mathrm{B}}=w_{\mathrm{B}}\cdot B$, $C_{\mathrm{B}}=E_{k_{\mathrm{B}x}}(R_{\mathrm{B}}, W_{\mathrm{B}})$, and then sends ($\mathrm{ID_B}$, response) and ($\mathrm{ID_B}$, $\mathrm{ID_A}$, C_{B}, R_{B}) to A and S,

respectively. The 'response' means that B agrees with A's request.

Round 3:

1. Upon receiving (ID_A, ID_B, C_A, R_A) and (ID_B, ID_A, C_B, R_B), S computes two temporary keys $K_\text{A}=\text{sk}_\text{S}\cdot R_\text{A}=(k_{\text{A}x}, k_{\text{A}y})$ and $K_\text{B}=\text{sk}_\text{S}\cdot R_\text{B}=(k_{\text{B}x}, k_{\text{B}y})$, and further uses $k_{\text{A}x}$ and $k_{\text{B}x}$ to decrypt C_A and C_B to obtain (R_A, R_B) and (R_B, W_B), respectively.

2. S verifies whether the decrypted R_A and R_B are equal to R_A and R_B sent from A and B, respectively. If the verification holds, it means that both A and B are valid users, and S continues to perform Step 3. Otherwise, S terminates the protocol and returns a failure message to both users.

3. S uses the symmetric keys $k_{\text{A}x}$ and $k_{\text{B}x}$ to compute $C_\text{SA}=E_{k_{\text{A}x}}(R_\text{A}, W_\text{B})$ and $C_\text{SB}=E_{k_{\text{B}x}}(R_\text{B}, W_\text{A})$, and sends C_SA and C_SB to A and B, respectively.

4. After receiving C_SA, A decrypts C_SA to obtain (R_A, W_B). Then, A verifies if the decrypted R_A is equal to R_A computed in Round 1. If it is true, A confirms that B has been authenticated by S, and further derives the session key $\text{SK}=w_\text{A}W_\text{B}$. Otherwise, A terminates this transaction.

5. Upon receiving C_SB, B decrypts C_SB to obtain (R_A, W_B). Then, B verifies if the decrypted R_B is equal to R_B computed in Round 2. If it is true, B confirms that A has been authenticated by S, and further derives the session key $\text{SK}=w_\text{B}W_\text{A}$. Otherwise, B terminates this transaction.

2.2 Tan (2010)'s 3PAKE protocol

Tan (2010) first pointed out that Yang and Chang (2009)'s 3PAKE protocol cannot withstand impersonation and parallel attacks. Then, Tan (2010) proposed an enhanced 3PAKE protocol (Fig. 1) to improve these mentioned weaknesses. However, we find out that Tan (2010)'s protocol still cannot withstand impersonation attacks. The main reason is that the server uses only her/his own private key sk_S and the user-provided R_id to generate the temporary key K_id. Consider the impersonation-of-initiator attack scenario that an attacker C wants to impersonate the initiator A to communicate with the responder B. The attacker C performs the following steps:

1. C chooses a random number $r'_\text{A}\in Z_p^*$ to compute $R'_\text{A}=r'_\text{A}\cdot\text{PK}_\text{C}$ and the temporary key $K'_\text{A}=r'_\text{A}\cdot\text{sk}_\text{S}\cdot\text{PK}_\text{S}=(k'_{\text{A}x}, k'_{\text{A}y})$.

2. C chooses a random number $w'_\text{A}\in Z_p^*$ and current timestamp T'_A to compute $W'_\text{A}=w'_\text{A}\cdot P$, $C'_\text{AS}=E_{k_{\text{A}x}}(R'_\text{A}, W'_\text{A}, \text{ID}_\text{A}, \text{ID}_\text{B}, T'_\text{A})$, and then sends ($\text{ID}_\text{A}$, request) and ($\text{ID}_\text{A}$, C'_AS, R'_A, T_A) to B and S, respectively.

3. In Round 3, S computes $K'_\text{A}=\text{sk}_\text{S}\cdot R'_\text{A}=\text{sk}_\text{S}\cdot r'_\text{A}\cdot\text{PK}_\text{C}=r'_\text{A}\cdot\text{sk}_\text{C}\cdot\text{PK}_\text{S}=(k'_{\text{A}x}, k'_{\text{A}y})$ after receiving (ID_A, C'_AS, R'_A, T_A). Furthermore, S uses $k'_{\text{A}x}$ to decrypt C'_AS to obtain (R'_A, W'_A, ID_A, ID_B, T_A) and verifies if the decrypted R'_A and T'_A are equal to the received R'_A and T'_A, respectively. The verification will hold; it means that C successfully impersonates A and passes the authentication process.

In the impersonation-of-responder attack scenario, the attacker C can also impersonate the responder B to build a session key with the initiator A.

3 Proposed three-party key exchange (3PAKE) protocol

We propose a secure 3PAKE protocol for establishing a session key between two entities with a trusted server. To prevent impersonation attacks, the user's public key is used to generate the temporary key. The proposed protocol is divided into two phases: the initialization phase and the authenticated key exchange phase. The initialization phase of the proposed protocol is the same as the one of Yang and Chang (2009)'s protocol in Section 2.1. In addition, users A and B register with the trusted server S, and obtain $Y_\text{A}=\text{sk}_\text{A}\cdot\text{PK}_\text{S}$ and $Y_\text{B}=\text{sk}_\text{B}\cdot\text{PK}_\text{S}$, respectively. The detailed descriptions of the authenticated key exchange phase are as follows.

Round 1 (Fig. 2):

1. A chooses a random number $r_\text{A}\in Z_p^*$ to compute $R_\text{A}=r_\text{A}\text{sk}_\text{A}\cdot P+Y_\text{A}$ and the temporary key $K_\text{A}=r_\text{A}\cdot Y_\text{A}=(k_{\text{A}x}, k_{\text{A}y})$, where $k_{\text{A}x}$ and $k_{\text{A}y}$ are x and y coordinates of K_A over $E_p(a, b)$, respectively.

2. A computes the encryption message $C_\text{A}=E_{k_{\text{A}x}}(R_\text{A}, \text{ID}_\text{A}, \text{ID}_\text{B}, T_\text{A})$ and sends (ID_A, request) and (ID_A, R_A, C_A, T_A) to B and S, respectively. 'request' means that A invites B to establish a session key.

Round 2 (Fig. 3):

1. Upon receiving (ID_A, request), B chooses a random number $r_\text{B}\in Z_p^*$ to compute $R_\text{B}=r_\text{B}\text{sk}_\text{B}\cdot P+Y_\text{B}$ and the temporary key $K_\text{B}=r_\text{B}\cdot Y_\text{B}=(k_{\text{B}x}, k_{\text{B}y})$.

2. B computes the encryption message $C_\text{B}=E_{k_{\text{B}x}}(R_\text{B}, \text{ID}_\text{B}, \text{ID}_\text{A}, T_\text{B})$ and sends (ID_B, response)

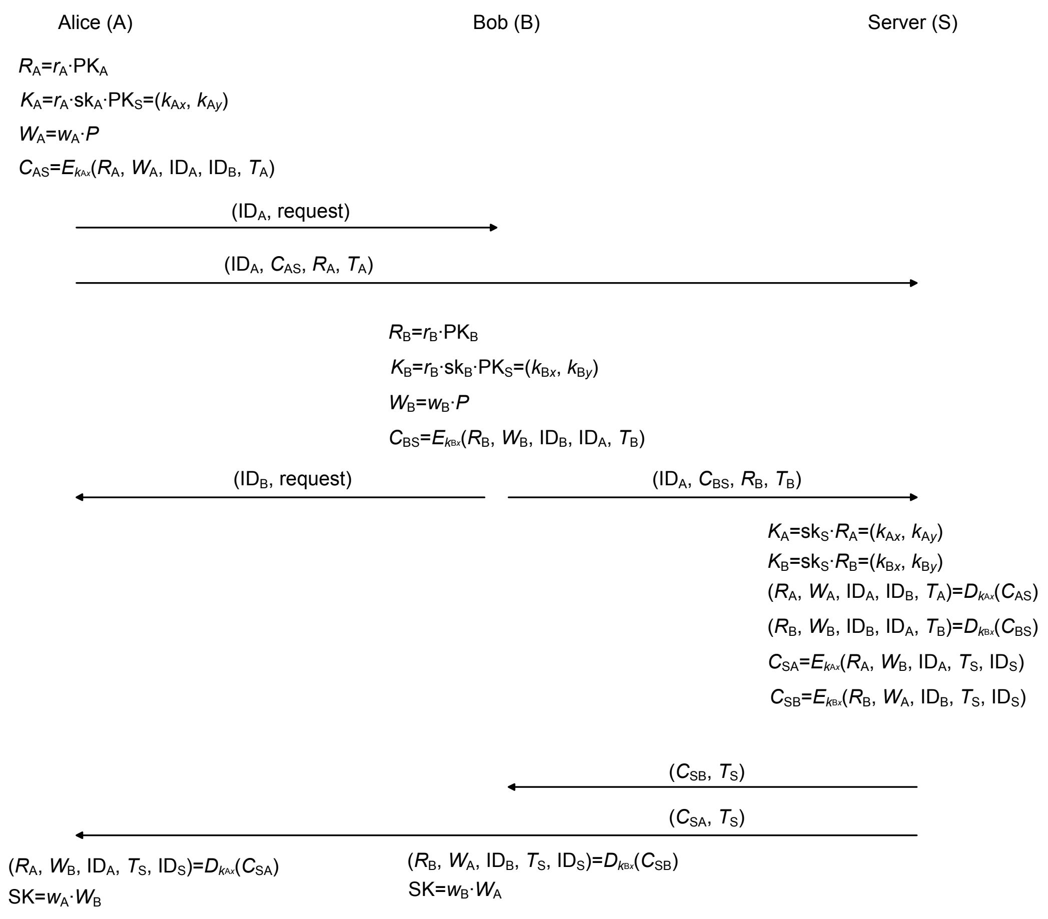

Alice (A)
Bob (B)
Server (S)
$R_A=r_A\cdot PK_A$
$K_A=r_A\cdot sk_A\cdot PK_S=(k_{Ax}, k_{Ay})$
$W_A=w_A\cdot P$
$C_{AS}=E_{k_{Ax}}(R_A, W_A, ID_A, ID_B, T_A)$
$(ID_A$, request)
(ID_A, C_{AS}, R_A, T_A)
$R_B=r_B\cdot PK_B$
$K_B=r_B\cdot sk_B\cdot PK_S=(k_{Bx}, k_{By})$
$W_B=w_B\cdot P$
$C_{BS}=E_{k_{Bx}}(R_B, W_B, ID_B, ID_A, T_B)$
$(ID_B$, request)
(ID_A, C_{BS}, R_B, T_B)
$K_A=sk_S\cdot R_A=(k_{Ax}, k_{Ay})$
$K_B=sk_S\cdot R_B=(k_{Bx}, k_{By})$
$(R_A, W_A, ID_A, ID_B, T_A)=D_{k_{Ax}}(C_{AS})$
$(R_B, W_B, ID_B, ID_A, T_B)=D_{k_{Bx}}(C_{BS})$
$C_{SA}=E_{k_{Ax}}(R_A, W_B, ID_A, T_S, ID_S)$
$C_{SB}=E_{k_{Bx}}(R_B, W_A, ID_B, T_S, ID_S)$
(C_{SB}, T_S)
(C_{SA}, T_S)
$(R_A, W_B, ID_A, T_S, ID_S)=D_{k_{Ax}}(C_{SA})$
$SK=w_A\cdot W_B$
$(R_B, W_A, ID_B, T_S, ID_S)=D_{k_{Bx}}(C_{SB})$
$SK=w_B\cdot W_A$

Fig. 1 Tan (2010)'s three-party key exchange (3PAKE) protocol

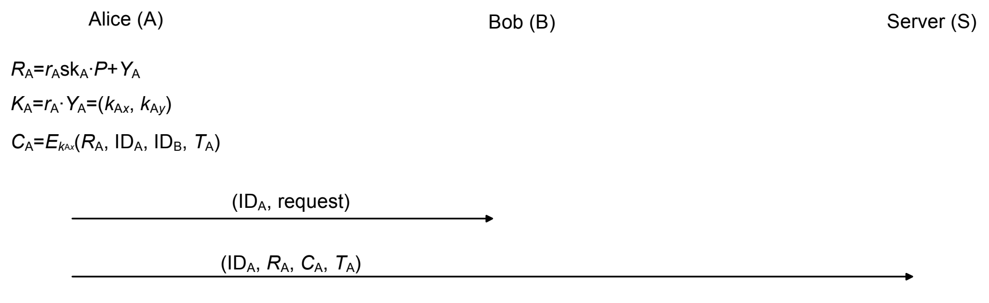

Alice (A)
Bob (B)
Server (S)
$R_A=r_A sk_A\cdot P+Y_A$
$K_A=r_A\cdot Y_A=(k_{Ax}, k_{Ay})$
$C_A=E_{k_{Ax}}(R_A, ID_A, ID_B, T_A)$
$(ID_A$, request)
(ID_A, R_A, C_A, T_A)

Fig. 2 Round 1 of the second phase in our proposed protocol

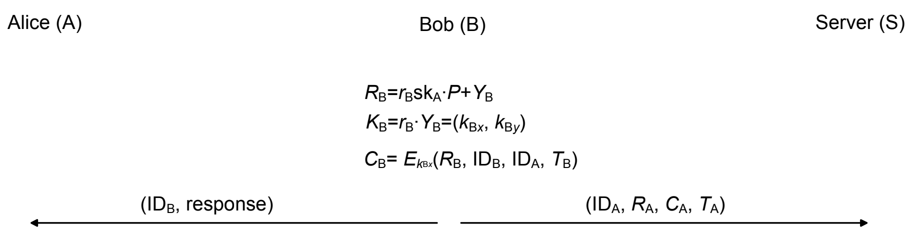

Alice (A)
Bob (B)
Server (S)
$R_B=r_B sk_A\cdot P+Y_B$
$K_B=r_B\cdot Y_B=(k_{Bx}, k_{By})$
$C_B= E_{k_{Bx}}(R_B, ID_B, ID_A, T_B)$
$(ID_B$, response)
(ID_A, R_A, C_A, T_A)

Fig. 3 Round 2 of the second phase in our proposed protocol

and ($\mathrm{ID_B}$, R_B, C_B, T_B) to A and S, respectively. 'response' means that B agrees wtih A's request.

Round 3 (Fig. 4):

1. Upon receiving ($\mathrm{ID_A}$, R_A, C_A, T_A) and ($\mathrm{ID_B}$, R_B, C_B, T_B), S computes temporary keys $K_\mathrm{A}=\mathrm{sk_S}\cdot(R_\mathrm{A}-\mathrm{sk_S}\cdot\mathrm{PK_A})=(k_{\mathrm{A}x}, k_{\mathrm{A}y})$ and $K_\mathrm{B}=\mathrm{sk_S}\cdot(R_\mathrm{B}-\mathrm{sk_S}\cdot\mathrm{PK_B})=(k_{\mathrm{B}x}, k_{\mathrm{B}y})$, and further uses $k_{\mathrm{A}x}$ and $k_{\mathrm{B}x}$ to decrypt C_A and C_B to obtain (R_A, $\mathrm{ID_A}$, $\mathrm{ID_B}$, T_A) and (R_B, $\mathrm{ID_B}$, $\mathrm{ID_A}$, T_B), respectively.

2. S performs the following verification steps: (1) The decrypted T_A and T_B are equal to the received T_A and T_B, respectively, and both of them are fresh; (2) The decrypted R_A and R_B are equal to R_A and R_B sent from A and B, respectively. If both of the verification results are true, both users are valid, and S continues to perform step 3. Otherwise, S terminates the protocol and returns a failure message to both users.

3. S uses the symmetric keys $k_{\mathrm{A}x}$ and $k_{\mathrm{B}x}$ to compute $C_\mathrm{SA}=E_{k_{\mathrm{A}x}}(R_\mathrm{A}, K_\mathrm{B}, \mathrm{ID_A}, \mathrm{ID_S}, T_\mathrm{S})$ and $C_\mathrm{SB}=E_{k_{\mathrm{B}x}}(R_\mathrm{B}, K_\mathrm{A}, \mathrm{ID_B}, \mathrm{ID_S}, T_\mathrm{S})$ and sends C_SA and C_SB to A and B, respectively.

4. After receiving C_SA, A computes $D_{k_{\mathrm{A}x}}(C_\mathrm{SA})=(R_\mathrm{A}, K_\mathrm{B}, \mathrm{ID_A}, \mathrm{ID_S}, T_\mathrm{S})$ and performs the following verification steps. A verifies (1) the decrypted T_S is equal to T_S sent from S and T_S is fresh, and (2) the decrypted R_A is equal to R_A computed in Round 1. If both of the verification results are true, A confirms that B has been authenticated by S, and computes the session key $\mathrm{SK}=r_\mathrm{A}\mathrm{sk_A}\cdot K_\mathrm{B}$. Otherwise, A rejects this transaction.

5. After receiving C_SA, B decrypts C_SB to obtain (R_B, K_A, $\mathrm{ID_B}$, $\mathrm{ID_S}$, T_S), and performs the same verification steps shown in Step 4 to verify R_B and T_S. If both of the verification results are true, B confirms that A has been authenticated by S, and further computes the session key $\mathrm{SK}=r_\mathrm{B}\mathrm{sk_B}\cdot K_\mathrm{A}$. Otherwise, B rejects this transaction.

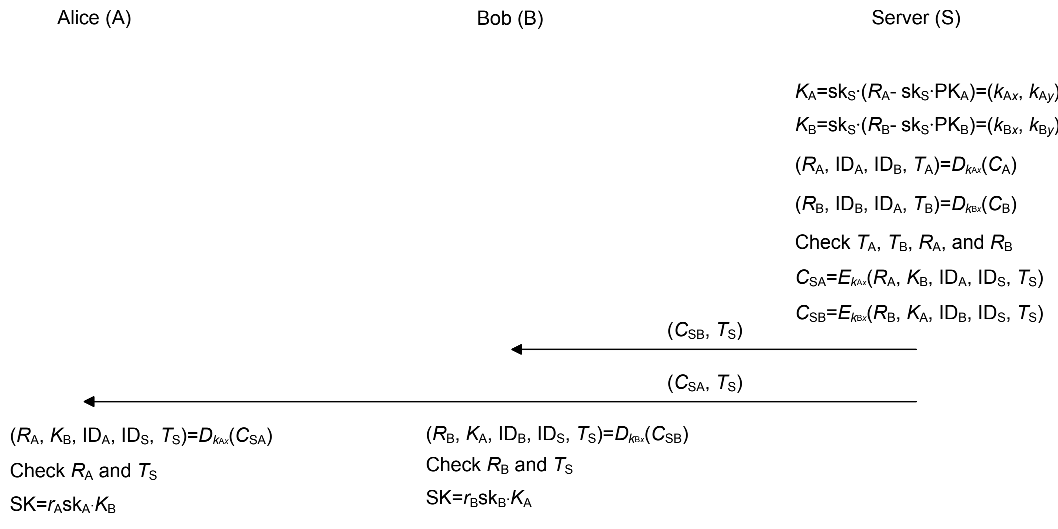
4 Security and performance analyses

4.1 Security analysis

Based on the intractability of solving the elliptic curve discrete logarithm problem (ECDLP) and elliptic curve computational Diffie-Hellman problem (ECCDHP) (Miller, 1986; Koblitz, 1987), we give formal analyses to prove that our protocol can achieve these security requirements, including known key security, forward secrecy, key-compromise impersonation, unknown key share, no key control, and replay-attack resistance. Detailed descriptions of ECDLP and ECCDHP are shown as follows.

Assumption 1 (Elliptic curve discrete logarithm problem, ECDLP) Given two points P and Q on an elliptic curve, it is computationally infeasible to find an integer r such that $Q=rP$.

Assumption 2 (Elliptic curve computational Diffie-Hellman problem, ECCDHP) Given three points P, xP, and yP on an elliptic curve, it is computationally infeasible to compute a point $xy\cdot P$.



Alice (A)
Bob (B)
Server (S)
$K_\mathrm{A}=\mathrm{sk_S}\cdot(R_\mathrm{A}-\mathrm{sk_S}\cdot\mathrm{PK_A})=(k_{\mathrm{A}x}, k_{\mathrm{A}y})$
$K_\mathrm{B}=\mathrm{sk_S}\cdot(R_\mathrm{B}-\mathrm{sk_S}\cdot\mathrm{PK_B})=(k_{\mathrm{B}x}, k_{\mathrm{B}y})$
$(R_\mathrm{A}, \mathrm{ID_A}, \mathrm{ID_B}, T_\mathrm{A})=D_{k_{\mathrm{A}x}}(C_\mathrm{A})$
$(R_\mathrm{B}, \mathrm{ID_B}, \mathrm{ID_A}, T_\mathrm{B})=D_{k_{\mathrm{B}x}}(C_\mathrm{B})$
Check T_A, T_B, R_A, and R_B
$C_\mathrm{SA}=E_{k_{\mathrm{A}x}}(R_\mathrm{A}, K_\mathrm{B}, \mathrm{ID_A}, \mathrm{ID_S}, T_\mathrm{S})$
$C_\mathrm{SB}=E_{k_{\mathrm{B}x}}(R_\mathrm{B}, K_\mathrm{A}, \mathrm{ID_B}, \mathrm{ID_S}, T_\mathrm{S})$
$(C_\mathrm{SB}, T_\mathrm{S})$
$(C_\mathrm{SA}, T_\mathrm{S})$
$(R_\mathrm{A}, K_\mathrm{B}, \mathrm{ID_A}, \mathrm{ID_S}, T_\mathrm{S})=D_{k_{\mathrm{A}x}}(C_\mathrm{SA})$
Check R_A and T_S
$\mathrm{SK}=r_\mathrm{A}\mathrm{sk_A}\cdot K_\mathrm{B}$
$(R_\mathrm{B}, K_\mathrm{A}, \mathrm{ID_B}, \mathrm{ID_S}, T_\mathrm{S})=D_{k_{\mathrm{B}x}}(C_\mathrm{SB})$
Check R_B and T_S
$\mathrm{SK}=r_\mathrm{B}\mathrm{sk_B}\cdot K_\mathrm{A}$

Fig. 4 Round 3 of the second phase in our proposed protocol

In the following, we introduce the adversary models defined by Canetti and Krawczyk (2001), and then show that the proposed protocol is secure against the adversary under the assumptions of ECDLP and ECCDHP.

Canetti and Krawczyk (2001) addressed two kinds of adversary models: the unauthenticated-links model and the authenticated-links model. Assume that two communicating entities carry out a message-driven protocol controlled by an adversary. One of the entities A with the identity $\mathrm{ID_A}$ serves as an initiator, and the other entity B with the identity $\mathrm{ID_B}$ serves as a responder. ($\mathrm{ID_A}$, $\mathrm{ID_B}$, sid, initiaotr) and ($\mathrm{ID_B}$, $\mathrm{ID_A}$, sid, responder) are the input data to the key exchange protocol associated to A and B, respectively, where sid is a session identifier. The session associated to A and the session associated to B are matching if their session identifiers are identical. Details of the adversary models are described as follows.

In the unauthenticated-links model, there is a probabilistic polynomial-time attacker, denoted by U, who can control the communication links and the schedule for all protocol events. That is, U can modify the transmitted messages, inject some messages, and re-schedule the initiation of the protocol and the subsequent message transmission in the protocol. To gain the advantage from the game, U can send the following queries to the game simulator:

Session-state reveal: U submits a party's identity and an incomplete session identifier and learns the state of the session. Note that U cannot learn any long-term secret information or master keys held by the party.

Session-key query: U submits a party's identity and a complete session identifier, and learns the session key in the intended session.

Session expiration: U submits a party's identity and a complete session identifier for letting the simulator erase the session key and related session states. This query captures the notion of perfect forward secrecy.

Party-corruption query: U decides to corrupt a party and learns all secret information or master keys of the party, and then completely controls the party. After that, the party cannot be activated.

The authenticated-links model is applicable to the case that the attacker does not have the capability to inject or modify the transmitted messages. In other words, there exists a probabilistic polynomial-time attacker V, who is restricted to delivering messages generated from one of the communicating parties to the other one.

Theorem 1 If our proposed three-party authenticated key exchange protocol is session-key secure in the authenticated-links model, then our proposed three-party authenticated key exchange protocol is also session-key secure in the unauthenticated-links model.

Proof Suppose that there is an adversary V in an authenticated-links model. Given ($\mathrm{PK_S}$, $r_\mathrm{A}Y_\mathrm{A}$, $r_\mathrm{B}Y_\mathrm{B}$), the goal of a simulator δ is to output $r_\mathrm{A}\mathrm{sk_A}r_\mathrm{B}\mathrm{sk_B}\cdot\mathrm{PK_S}$. Given a security parameter l, δ generates system parameters. After that, V can send the above queries to δ. According to the simulation, the session key is represented in the form of $\mathrm{SK}=r_\mathrm{A}\mathrm{sk_A}r_\mathrm{B}\mathrm{sk_B}\cdot\mathrm{PK_S}$. However, it is computationally infeasible to obtain $r_\mathrm{A}\mathrm{sk_A}r_\mathrm{B}\mathrm{sk_B}\cdot\mathrm{PK_S}$ for given $r_\mathrm{A}Y_\mathrm{A}$ and $r_\mathrm{B}Y_\mathrm{B}$ in accordance with Assumptions 1 and 2.

Hence, the advantage $\mathrm{Adv_V}(l)$ for V is negligible in the authenticated-links model. That is, our protocol is session-key secure in the authenticated-links model. According to the above result, the advantage $\mathrm{Adv_U}(l)$ for U is also negligible in the unauthenticated-links model. This implies that our protocol is session-key secure in the unauthenticated model.

Theorem 2 If our proposed three-party authenticated key exchange protocol is perfect forward secure in the unauthenticated-links model, then the advantage $\mathrm{Adv_V}(l)$ for an adversary is negligible in the unauthenticated model.

Proof Suppose that there is an adversary U in the unauthenticated-links model. Given a security parameter l, a simulator δ generates system parameters. According to the above models, it is assumed that U has compromised the long-term keys held by A, B, or S before the session expires. Under this assumption, U still cannot compromise the past session keys shared by A and B because it is computationally infeasible to obtain $\mathrm{SK}=r_\mathrm{A}\mathrm{sk_A}r_\mathrm{B}\mathrm{sk_B}\cdot\mathrm{PK_S}$ for given $\{\mathrm{sk_A}, \mathrm{sk_B}, \mathrm{sk_S}, K_\mathrm{A}, K_\mathrm{B}\}$ in accordance with Assumptions 1 and 2, where $K_\mathrm{A}=r_\mathrm{A}Y_\mathrm{A}$ and $K_\mathrm{B}=r_\mathrm{B}Y_\mathrm{B}$.

Hence, the advantage $\mathrm{Adv_U}(l)$ for U is negligible in the unauthenticated-links model. This implies that our protocol achieves perfect forward secrecy in the unauthenticated-links model.

Theorem 3 If our proposed three-party authenticated key exchange protocol is key-compromise impersonation resistant in the unauthenticated-links model, then the advantage $\mathrm{Adv}_{\mathrm{V}}(l)$ for an adversary is negligible in the unauthenticated model.

Proof If an attacker obtains user A's long-term secret key, he/she can certainly impersonate A. Nevertheless, a protocol is said to be resistant with key-compromise impersonation attacks if A's long-term secret key is leaked. The attacker cannot masquerade other users and obtain the resulting session key. Suppose that there is an adversary U in the unauthenticated-links model. Given a security parameter l, a simulator δ generates system parameters. According to the above models, it is assumed that U has compromised the long-term keys held by A, B, or S before the session expires. Two kinds of simulations for key-compromise impersonation attacks are as follows:

1. Suppose that one entity A's or B's private key is compromised. For given $\mathrm{sk}_{\mathrm{A}}/\mathrm{sk}_{\mathrm{B}}$, U still cannot successfully impersonate the other entity B or A to the compromised entity A or B because it is computationally infeasible to obtain K_{B} or K_{A} in accordance with Assumption 1, where $K_{\mathrm{A}}=\mathrm{sk}_{\mathrm{S}}\cdot(R_{\mathrm{A}}-\mathrm{sk}_{\mathrm{S}}\cdot\mathrm{PK}_{\mathrm{A}})$ and $K_{\mathrm{B}}=\mathrm{sk}_{\mathrm{S}}\cdot(R_{\mathrm{B}}-\mathrm{sk}_{\mathrm{S}}\cdot\mathrm{PK}_{\mathrm{B}})$. Hence, the advantage $\mathrm{Adv}_{\mathrm{U}}(l)$ for U is negligible in the unauthenticated-links model.

2. Some key exchange protocols cannot in fact be really resistant to key-compromise impersonation attacks, especially identity-based key exchange protocols. Suppose that an adversary obtains the server's long-term private key. The adversary can derive each user's private key $\mathrm{sk}_{\mathrm{A}}/\mathrm{sk}_{\mathrm{B}}$ for given $\{\mathrm{PK}_{\mathrm{A}}, \mathrm{PK}_{\mathrm{B}}\}$ due to Assumption 1. Furthermore, it is computationally infeasible to obtain $\mathrm{SK}=r_{\mathrm{A}}\mathrm{sk}_{\mathrm{A}}r_{\mathrm{B}}\mathrm{sk}_{\mathrm{B}}\cdot\mathrm{PK}_{\mathrm{S}}$ for given $\{\mathrm{sk}_{\mathrm{A}}, K_{\mathrm{A}}, K_{\mathrm{B}}\}$ due to Assumption 2. Hence, the adversary cannot successfully impersonate A or B to B or A. The advantage $\mathrm{Adv}_{\mathrm{U}}(l)$ for U is also negligible in the unauthenticated-links model. Our proposed protocol can withstand key-compromise impersonation attacks.

In the following, we discuss how our proposed protocol can achieve unknown key share, no key control, and replay-attack resistance.

Unknown key share means that two users A and B cannot be coerced into sharing a key between them when either A or B actually thinks that the key is shared with another user C. Because the transmitted messages and ciphers include users' identities, no one but A can build the session key with B, and no one but B can build the session key with A. Furthermore, our proposed protocol can withstand key-compromise impersonation attacks. A believes the session key is shared with B, and similarly B believes the session key is shared with A. Therefore, the proposed approach can withstand unknown key share attacks.

Key control means that one of the participants can determine a certain value as the session key or predict its value. Definitely, our proposed protocol achieves no key control, since the session key $\mathrm{SK}=r_{\mathrm{A}}r_{\mathrm{B}}\mathrm{PK}_{\mathrm{S}}$ is composed of r_{A} and r_{B}, which are contributed by A and B, respectively. Both users cannot decide or predict the value of the session key.

The replay attack means that an attacker simply takes a previous message and sends it again to pass the authentication. If two users A and B exchange the messages, an adversary eavesdrops and duplicates the message (ID_{A}, R_{A}, C_{A}, T_{A}). After A and B stop communications, the adversary wants to pretend to be A or B. He/She can perform the following steps:

1. The adversary wants to masquerade as A, and replays the message (ID_{A}, R_{A}, C_{A}, T_{A}) to S. Then, S checks if the decrypted T_{A} is equal to the received T_{A}, and whether T_{A} is fresh or not. If those are true, S assures that A is valid. Otherwise, S terminates the protocol and returns a failure message to both users. In this case, the timestamp T_{A} is not fresh, so the adversary cannot impersonate A successfully.

2. Similarly, the adversary cannot impersonate B by resending the message (ID_{B}, R_{B}, C_{B}, T_{B}) to S, since the message includes the timestamp T_{B}.

4.2 Performance and security achievement comparisons

According to Table 2, it is obvious to see that our proposed protocol can achieve session-key security, forward secrecy, unknown key share resistance, no key control, replay-attack resistance, and key-compromise impersonation resistance. Both of Yang and Chang (2009)'s and Tan (2010)'s protocols cannot resist against the key-compromise impersonation attack, and Yang and Chang (2009)'s protocol also cannot detect the replay attack.

We further analyze the performance of the proposed protocol. Assume the following conditions hold: (1) The size of p used in ECC is 160 bits; (2)

The encrypted message size of the symmetric encryption/ decryption algorithm (e.g., the advanced encryption standard (AES)) is 128 bits; (3) The digest message size of the hash function (e.g., secure hash algorithm 1 (SHA-1)) is 160 bits; (4) The timestamp length is 16 bits; (5) The user's identity size is 80 bits. Table 3 shows the comparisons of message sizes and computational costs of Yang and Chang (2009)'s, Tan (2010)'s, and our protocols. The total communication sizes in Yang and Chang (2009)'s protocol are 1312 bits, and the total communication size in Tan (2010)'s protocol and our proposed protocol are both 1056 bits.

In the aspect of computational costs, both Yang and Chang (2009)'s and Tan (2010)'s protocols need 12 point multiplications and 8 symmetric encryptions/ decryptions in total. Our proposed protocol requires only 10 point multiplications, 4 point additions, and 8 symmetric encryptions/decryptions. Note that point multiplications are more time-consuming than point additions on ECC (Tsaur and Chou, 2005). The computational cost for performing $P+Q$ is 2M+S+I if $P\neq Q$, which is called an addition. The computational cost for performing $P+Q$ is 2M+2S+I if $P=Q$, which is called a doubling. The symbols M, S, and I indicate a multiplication, a squaring, and an inversion in finite fields, respectively. Furthermore, if the square and multiply method (Knuth, 1981) is adopted, one point multiplication requires approximately 159 doublings and 79 additions to perform rP in the average case, where r is a 160-bit number. Consequently, Yang and Chang (2009)'s and Tan (2010)'s protocols both require 1908 doublings and 948 additions, and our protocol requires only 1590 doublings and 794 additions without considering symmetric encryptions/ decryptions.

5 Conclusions

User authentication and key exchange are essential to ensure secure communication. In a three-party authentication key exchange protocol, both communicating entities can obtain a session key with the assistance of a trusted server and establish a secure channel. This paper presents an efficient and secure three-party authenticated key exchange protocol based on ECC, which can accomplish most desired security goals and attributes, e.g., replay attack, forward secrecy, known key security, key-compromise impersonation, and unknown key share. Moreover, our proposed 3PAKE protocol is efficient and practical in mobile environments.

References

Cagalj, M., Capkun, S., Hubaux, J.P., 2006. Key agreement in peer-to-peer wireless networks. *Proc. IEEE*, **94**(2): 467-478. [doi:10.1109/JPROC.2005.862475]

Canetti, R., Krawczyk, H., 2001. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. Proc. Advances in Cryptology, p.453-474. [doi:10.1007/3-540-44987-6_28]

Chen, T.H., Lee, W.B., Chen, H.B., 2008. A round- and computation-efficient three-party authenticated key exchange protocol. *J. Syst. Softw.*, **81**(9):1581-1590. [doi:10.1016/j.jss.2007.11.720]

Table 2 Security achievement comparisons

Protocol	Session-key security	Forward secrecy	Key-compromise impersonation resistance	Unknown key share resistance	No key control	Replay-attack resistance
Yang and Chang (2009)'s	Yes	Yes	No	Yes	Yes	No
Tan (2010)'s	Yes	Yes	No	Yes	Yes	Yes
Ours	Yes	Yes	Yes	Yes	Yes	Yes

Table 3 Performance comparisons

Protocol	Message size (bit)	Computational cost			Detailed total computational cost*
		A(B)	S	Total	
Yang and Chang (2009)'s	1312	5PM+2SE	2PM+4SE	12PM+8SE	1908 doublings+948 additions
Tan (2010)'s	1056	5PM+2SE	2PM+4SE	12PM+8SE	1908 doublings+948 additions
Ours	1056	3PM+1PA+2SE	4PM+2PA+4SE	10PM+4PA+8SE	1590 doublings+794 additions

PM: elliptic curve point multiplication; PA: elliptic curve point addition; SE: symmetric encryption/decryption. * Except symmetric encryption/decryption

Diffie, W., Hellman, M., 1976. New directions in cryptography. *IEEE Trans. Inf. Theory*, **22**(6):644-654. [doi:10.1109/TIT.1976.1055638]

Guo, H., Li, Z., Mu, Y., Zhang, X., 2008. Cryptanalysis of simple three party key exchange protocol. *Comput. Secur.*, **27**(1-2):16-21. [doi:10.1016/j.cose.2008.03.001]

Hölbl, M., Welzer, T., Brumen, B., 2010. Two proposed identity-based three-party authenticated key agreement protocols from pairings. *Comput. Secur.*, **29**(2):244-252. [doi:10.1016/j.cose.2009.08.006]

Knuth, D.E., 1981. The Art of Computer Programming, Volume II: Seminumerical Algorithms (2nd Ed.). Addison-Wesley, Reading, MA.

Koblitz, N., 1987. Elliptic curve cryptosystem. *Math. Comput.*, **48**(177):203-209. [doi:10.1090/S0025-5718-1987-0866109-5]

Lee, C.C., Chang, Y.F., 2008. On security of a practical three-party key exchange protocol with round efficiency. *Inf. Technol. Control*, **37**(4):333-335.

Lee, S.W., Kim, H.S., Yoo, K.Y., 2005. Efficient verifier-based key agreement protocol for three parties without server's public key. *Appl. Math. Comput.*, **167**(2):996-1003. [doi:10.1016/j.amc.2004.06.129]

Lu, R., Cao, Z., 2007. Simple three-party key exchange protocol. *Comput. Secur.*, **26**(1):94-97. [doi:10.1016/j.cose.2006.08.005]

Menezes, A.J., Orschot, P.C., Vanstone, S.A., 1996. Handbook of Applied Cryptography. CRC Press. [doi:10.1201/9781439821916]

Miller, V.S., 1986. Use of Elliptic Curves in Cryptography. Proc. Advances in Cryptology, p.417-426. [doi:10.1007/3-540-39799-X_31]

Padmavathy, R., 2010. Improved three party EKE protocol. *Inf. Technol. Control*, **39**(3):220-226.

Schnorr, C.P., 1989. Efficient Identification and Signatures for Smart Cards. Proc. CRYPTO, p.239-252. [doi:10.1007/0-387-34805-0_22]

Tan, Z., 2010. An enhanced three-party authentication key exchange protocol for mobile commerce environments. *J. Commun.*, **5**(5):436-443. [doi:10.4304/jcm.5.5.436-443]

Tsaur, W.J., Chou, C.H., 2005. Efficient algorithms for speeding up the computations of elliptic curve cryptosystems. *Appl. Math. Comput.*, **168**(2):1045-1064. [doi:10.1016/j.amc.2004.10.010]

Yang, J.H., Chang, C.C., 2009. An efficient three-party authenticated key exchange protocol using elliptic curve cryptography for mobile-commerce environments. *J. Syst. Software*, **82**(9):1497-1502. [doi:10.1016/j.jss.2009.03.075]

Yoon, E.J., Yoo, K.Y., 2008. Improving the novel three-party encrypted key exchange protocol. *Comput. Stand. Interf.*, **30**(5):309-314. [doi:10.1016/j.csi.2007.08.018]