Wireless Pers Commun (2016) 88:337–356
DOI 10.1007/s11277-015-3123-6

An Efficient Password-Based Authenticated Key Exchange Protocol with Provable Security for Mobile Client–Client Networks

Mohammad Heydari[2] · S. Mohammad Sajad Sadough[2] · Mohammad Sabzinejad Farash[1] · Shehzad Ashraf Chaudhry[3] · Khalid Mahmood[3]

Published online: 21 November 2015

Abstract Three party password based authenticated key exchange protocol can sanction couple of clients to institute a protected session key through a server above an insecure communication link. Youn et al. (Telecommun Syst 52(2):1367–1376, 2013) proposed three-party efficient and robust authenticated key exchange scheme that incurs three rounds. They assert that their scheme is invincible against customary attacks. Moreover, they claimed the scheme is lightweight due to low communication, computation costs and incorporating authentication in three rounds. However, comprehensive analysis in this paper reveals that Youn et al.'s scheme is susceptible to impersonation attack. To overcome the security feebleness, this paper introduces a modest scheme which not only maintains round efficiency, communication and computation costs but it also offer

✉ Mohammad Sabzinejad Farash
sabzinejad@khu.ac.ir

Mohammad Heydari
m_heydari@sbu.ac.ir

S. Mohammad Sajad Sadough
s_sadough@sbu.ac.ir

Shehzad Ashraf Chaudhry
shahzad@iiu.edu.pk

Khalid Mahmood
khalid.phdcs74@iiu.edu.pk

[1] Faculty of Mathematics Sciences and Computer, Kharazmi University, Tehran, Iran

[2] Department of Electrical Engineering, Shahid Beheshti University, Tehran, Iran

[3] Department of Computer Science and Software Engineering, International Islamic University, Islamabad, Pakistan

comprehensive security to repel popular security attacks. The security of the proposed scheme is verified through random oracle model.

Keywords Authenticated key exchange protocol · Dictionary attack · Impersonation attack · Random oracle model · Provable security

1 Introduction

Wireless communication is becoming imperative due to abrupt advancement of wireless peripherals. Wireless networking appreciate open ingress of wireless amenities which leads to security trepidations amid service providers and mobile users. Wireless networks facilitate a mobile user to enjoy the preferred service offered by foreign server through roaming and instituting mutual authentication with concerning foreign agent. Consequently, session key generation and authentication among foreign agents and mobile users have become vital security concerns for wireless technology implementation. One of the possible solution is to utilize authenticated key exchange protocols (AKE) [1–5]. AKE protocols can protect the communication among foreign agents and mobile users through generating private session keys and can also provide authentication. Moreover, password-based authenticated key exchange protocol (PAKE) are considered as one of the variant of AKE protocols that empower multiple users to generate large-size session key and also provide authentication to such specific mobile users that own and share weak passwords.

Initially, two party PAKE protocols (2PAKE) were introduced that proved to be more appropriate for client–server environment [6–10]. Although it is realized that 2PAKE protocols are inapt for exhaustive client–client communication architecture because it is infeasible for a particular client to remember the password of each communicating partner present in the colossal network. Therefore, three party PAKE protocols (3PAKE) are introduced in which trustworthy server acts as an intervening agent between the communicating clients and each client is responsible to share its password with the trusted server.

Various variants of 3PAKE schemes has been developed in order to obtain a more secure and feasible solution because dictionary attack is the major threat for these protocols. In order to avoid the dictionary attack, researchers has suggested three kinds of methodologies, that are (1) Using public key of server [11–15], (2) Using symmetric cryptosystems [16–18] and (3) without utilizing public key of server and symmetric algorithms [19–33]. This paper opted the third variant of 3PAKE protocols.

Huang [23] introduced such scheme in 2009 that doesn't utilize any public key of server and also doesn't follow symmetric encryption and decryption algorithms. Later on, Yoon and Yoo [24] find out that Huang's scheme can easily be compromised by off-line password guessing and imperceptible online password attacks. Moreover, Wu et al. [25] added up that Huang's scheme is unable to resist key compromise impersonation attack and in turn presented an enhanced scheme by using public key of server.

Lee et al. [26] in 2011, presented a 3PAKE based scheme that doesn't utilize public key of server for reducing communication overhead but soon after that Chang et al. [27] modified Lee et al.'s scheme and introduced such scheme that doesn't use symmetric encryption/decryption algorithms and also doesn't public key of server. Wu et al. [28] then proved that password guessing attack is possible on Chang et al.'s scheme. Xiong et al. [15] claimed and proved that Xiong et al. [15] is insecure against key compromise impersonation attack and introduced 3PAKE based scheme that make use of public key of

server. It has been observed that most of the schemes are prone to key compromise impersonation attack therefore suitable remedy should be considered. Tso [29] also enhanced the Chang et al.'s scheme to mitigate the security weaknesses that doesn't use the public key of server and symmetric encryption/decryption algorithms.

Chien [30] introduce 3PAKE based scheme that utilizes verifiers. Chien declared that his scheme is more secure to resist security attacks specially different dictionary attacks. But Pu et al. [31] proved Chiens claim as null and void by exposing partition attack also known as offline dictionary attack. Later on, Pu et al. [31] and Liu et al. [12] both of them introduced improved schemes against Chien's scheme. Pu et al. and Liu et al. both presented the 3PAKE schemes but the difference between the both of them are that Pu et al.'s scheme uses verifiers and doesn't use public key of server and symmetric encryption/decryption algorithms whereas the Liu et al.'s scheme uses the public key of server in their scheme.

Tallapally [32] launched an anonymous key share attack over 3PAKE scheme of Huang [23] and then improved the said scheme to mitigate the security weaknesses in that. Soon after that Farash and Attari [9] pointed out that scheme of Tallapally is insecure against off-line password guessing and imperceptible online password guessing attacks.

Youn et al. [34] also introduces a robust and proficient 3PAKE scheme that doesn't use the public key of server and encryption/decryption algorithms. This paper proves that impersonation attack is likely on Youn et al.'s scheme. It is verified that any legitimate user can be masqueraded by an adversary, without having the password of the user.

Whatever is left of this paper is sorted out as follows, Sect. 2 presents the review of Youn et al.'s scheme based on 3PAKE. Impersonation attack is demonstrated and proved in Sect. 3. In Sect. 4, improved scheme is discussed. Security of the improved scheme is evaluated in Sect. 5, using random oracle model. Section 6 presents the performance comparison of improved scheme w.r.t some related schemes. Concluding remarks are stated in Sect. 7.

2 Review of Youn et al.'s 3PAKE Scheme

This section presents a brief review of Youn et al.'s 3PAKE scheme [34]. The notations used in the description of Youn et al.'s scheme are wrapped up in Table 1. In Youn et al.'s scheme, every user share his/her password pw with the remote server S. Then, S computes

Table 1 The notations

Notation	Narrative
A, B	Authentic or legal users
pw	The password of authentic user
S	Remote server
p	Sizeable prime numbers with $p = 2q + 1$
$\mathbb{G}$	A multiplicative group of order q
g	A generator of $\mathbb{G}$
$\mathbb{Z}_q^*$	The non-zero residues modulus q
$H(.)$	A hash function $H : \{0,1\} \longrightarrow \{0,1\}^k$, k referred to security parameter
$Gen_K(M)$	The message authentication code (MAC) producing algorithm that produces a tag T against message M and a key K
$Ver_K(T, M)$	MAC verification technique used set a flag F against provided T, M and K. If T is valid tag for M against key K, flag F will set to 1, otherwise F is set to 0

s^{pw}, t^{pw}, u^{pw}, v^{pw}, and save these values with identity *id* of each user. The description of this scheme, shown in Fig. 1, are as under:

Round 1 $A \longrightarrow S, B$: $\{ID_{ua}||ID_{ub}||X_1||X_2\}$

- The user A selects couple of random numbers $x_1, x_2 \in \mathbb{Z}_q^*$, calculates $X_1 = g^{x_1} s^{pw_A}$, $X_2 = g^{x_2}$ and broadcasts $\{ID_{ua}||ID_{ub}||X_1||X_2\}$ to the user B and the server S.

Round 2 $B \longrightarrow S, A$:$\{ID_{ua}||ID_{ub}||Y_1||Y_2\}$, $S \longrightarrow A$:$\{ID_S||Z_1\}$ and $S \longrightarrow A, B$: $\{ID_S||Z_1\}$, $\{ID_S||Z_2\}$

- The user B computes $Y_1 = g^{y_1} t^{pw_B}$ and $Y_2 = g^{y_2}$ for random numbers $y_1, y_2 \in \mathbb{Z}_q^*$ and transmits $\{ID_{ua}||\, ID_{ub}||\, Y_1||\, Y_2\}$ to A and S.

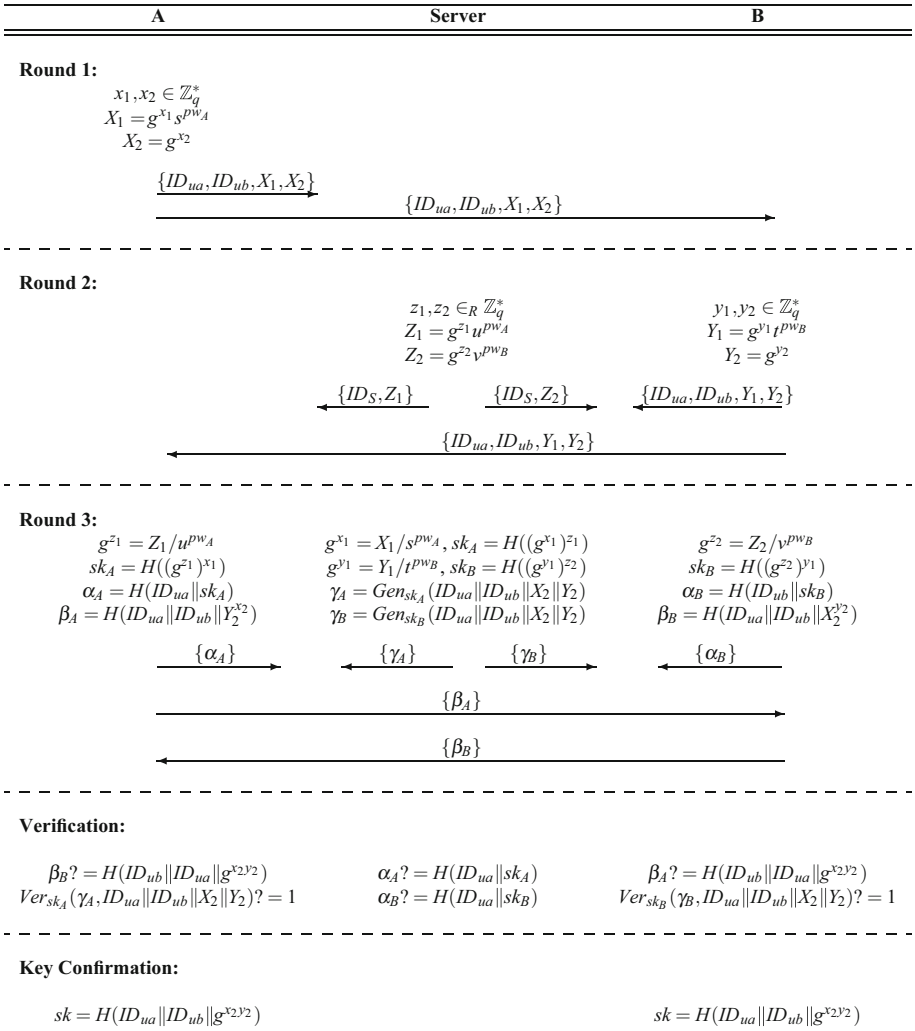

Fig. 1 Youn et al.'s 3PAKE protocol [34]

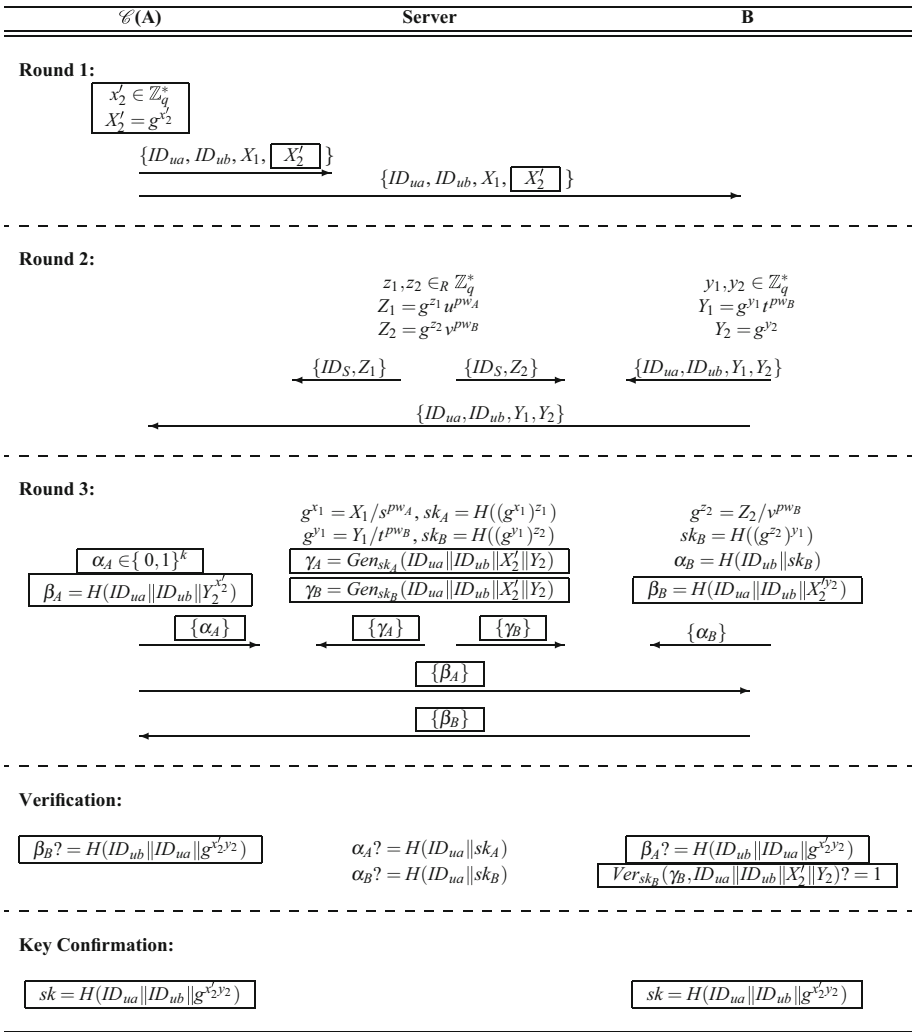

Fig. 2 Impersonation attack on Youn et al.'s scheme

- Simultaneously, S calculates $Z_1 = g^{z_1} u^{pw_A}$ and $Z_2 = g^{z_2} v^{pw_B}$ for arbitrary numbers $z_1, z_2 \in \mathbb{Z}_q^*$, and transmits $\{ID_S||Z_1\}$, $\{ID_S||Z_2\}$ to A and B, correspondingly.

Round 3 $A \longrightarrow S$: α_A, $A \longrightarrow B$: β_A, $B \longrightarrow S$: α_B, $B \longrightarrow A$: β_B, $S \longrightarrow A$: γ_A and $S \longrightarrow B$: γ_B

- A computes $g^{z_1} = Z_1/u^{pw_A}$, $sk_A = H((g^{z_1})^{x_1})$, $\alpha_A = H(ID_{ua}||sk_A)$ and $\beta_A = H(ID_{ua}||ID_{ub}||Y_2^{x_2})$, and transmits α_A, β_A to B and S, correspondingly.
- Likewise, B calculates $g^{z_2} = Z_2/v^{pw_B}$, $sk_B = H((g^{z_2})^{y_1})$, $\alpha_B = H(ID_{ub}||sk_B)$ and $\beta_B = H(ID_{ub}||ID_{ua}||X_2^{y_2})$, and sends α_B, β_B to S and A, correspondingly.
- At the same instant, S calculates $g^{x_1} = X_1/s^{pw_A}$, $sk_A = H((g^{x_1})^{z_1})$, $g^{y_1} = Y_1/t^{pw_B}$ and $sk_B = H((g^{y_1})^{z_2})$. Next, S produces couple of tags $\gamma_A = Gen_{sk_A}(ID_{ua}||ID_{ub}||X_2||Y_2)$ and $\gamma_B = Gen_{sk_B}(ID_{ua}||ID_{ub}||X_2||Y_2)$. S then sends γ_A, γ_B to A and B, correspondingly.

Verification Given data is verified by each entity that is transferred by the rest of the entities.

- If $\beta_B = H(ID_{ub}\|ID_{ua}\|g^{x_2y_2})$ or $Ver_{sk_A}(\gamma_A, ID_{ua}\|\ ID_{ub}\|\ X_2\|\ Y_2) \neq 0$, A culls the scheme.
- If $\beta_A = H(ID_{ua}\|ID_{ub}\|g^{x_2y_2})$ or $Ver_{sk_B}(\gamma_B, ID_{ua}\|\ ID_{ub}\|\ X_2\|\ Y_2) \neq 0$, B culls the scheme.
- If $\alpha_A = H(ID_{ua}\|sk_A)$ or $\alpha_B = H(ID_{ub}\|sk_B)$, S culls the scheme.

If every condition gets true, S can considers that both clients are authentic, also A and B consider that it impossible for an adversary to compromise their protocol.
Key Confirmation After successful substantiation, A and B produces a session key using $g^{x_2y_2}$ as a seed. Then session key is calculated as $sk = H(ID_{ua}\|ID_{ub}\|g^{x_2y_2})$.

3 Cryptanalysis of Youn et al.'s Scheme

As we know, the main objective of a 3PAKE protocol is to allow two users to get mutual authentication and establish a secure session key through a server over an unsafe and public channel. Although, this paper shows that Youn et al.'s scheme doesn't offer reliable authentication and session key privacy. It is also proved that any legal user can be easily masqueraded without utilizing password of user. To perform such attack, an active adversary $\mathscr{C}$ firstly obtains the exchanged messages through a previous session by eavesdropping the session, and then uses these messages for impersonating an authentic user.

Assume an active adversary $\mathscr{C}$ in an attempt to impersonate the legal user A obtains the message $\{ID_{ua}||ID_{ub}||X_1||X_2\}$ broadcasted by A in a previous session. To impersonate A, $\mathscr{C}$ completes the subsequent steps as depicted in Fig. 2:

Round 1 The adversary $\mathscr{C}$ chooses a random value $x_2' \in \mathbb{Z}_q^*$and computes

$$X_2' = g^{x_2'} \tag{1}$$

and broadcasts $\{ID_{ua}||ID_{ub}||X_1||X_2'\}$ to the user B and the server S. Note that, the parameter

$$X_1 = g^{x_1} s^{pw_A} \tag{2}$$

is computed by the user A in the previous session.
Round 2 This round is performed similar to one performed in the original protocol.

- The user B computes

$$Y_1 = g^{y_1} t^{pw_B}, \tag{3}$$

$$Y_2 = g^{y_2}, \tag{4}$$

for random numbers $y_1, y_2 \in \mathbb{Z}_q^*$,and transmits $\{ID_{ua}||\ ID_{ub}\|\ Y_1\|\ Y_2\}$ to $A$and S.

- Simultaneously, S calculates

$$Z_1 = g^{z_1} u^{pw_A}, \tag{5}$$

$$Z_2 = g^{z_2} v^{pw_B}, \tag{6}$$

for arbitrary numbers $z_1, z_2 \in \mathbb{Z}_q^*$, and transmits $\{ID_S \| Z_1\}$, $\{ID_S \| Z_2\}$ to A and B, correspondingly.

Round 3 This round is different to the original protocol in the computations performed by the adversary.

- After receiving the message $\{ID_S \| Z_1\}$, $\mathbb{C}$ computes

$$\alpha_A \in \{0,1\}^k, \tag{7}$$

$$\beta_A = H(ID_{ua} \| ID_{ub} \| Y_2^{x_2}), \tag{8}$$

and transmits α_A, β_A to S and B, correspondingly.

- Likewise, after receiving the message $\{ID_S \| Z_2\}$, B computes

$$g^{z_2} = Z_2 / v^{pw_B}, \tag{9}$$

$$sk_B = H((g^{z_2})^{y_1}), \tag{10}$$

$$\alpha_B = H(ID_{ub} \| sk_B), \tag{11}$$

$$\beta_B = H(ID_{ub} \| ID_{ua} \| X_2^{y_2}), \tag{12}$$

and transmits α_B, β_B to S and A, correspondingly.

- Simultaneously, S calculates

$$g^{x_1} = X_1 / s^{pw_A}, \tag{13}$$

$$sk_A = H((g^{x_1})^{z_1}), \tag{14}$$

$$g^{y_1} = Y_1 / t^{pw_B}, \tag{15}$$

$$sk_B = H((g^{y_1})^{z_2}). \tag{16}$$

Next, S generates couple of tags

$$\gamma_A = Gen_{sk_A}(ID_{ua} \| ID_{ub} \| X_2 \| Y_2), \tag{17}$$

$$\gamma_B = Gen_{sk_B}(ID_{ua} \| ID_{ub} \| X_2 \| Y_2). \tag{18}$$

S then transmits γ_A, γ_B to A and B, correspondingly.

Verification Given data is veried by each entity that is transferred by the rest of the entities.

- B verifies if

$$\beta_A ? = H(ID_{ua} \| ID_{ub} \| g^{x_2' y_2}) \tag{19}$$

and

$$Ver_{sk_B}(\gamma_B, ID_{ua}\|ID_{ub}\|X_2'\|Y_2)? = 1, \tag{20}$$

If they don't get true, B culls the protocol. Otherwise, B considers that their protocol cannot be compromised by any adversary.

- If

$$\alpha_A = H(ID_{ua}\|sk_A) \tag{21}$$

or

$$\alpha_B = H(ID_{ub}\|sk_B), \tag{22}$$

S culls the scheme. Otherwise, S can considers that both clients are legitimate.

Key Confirmation After successful verication, $\mathscr{C}$ and B produces a session key by using $g^{x_2'y_2}$ as a seed. Then session key is calculated as $sk = H(ID_{ua}\|ID_{ub}\|g^{x_2'y_2})$.

Proposition 1 By applying the above attack, the adversary $\mathscr{C}$ can successfully masquerade as A and establish a session key with B.

Proof According to verification phase of proposed attack, B accept the session if the Eqs. (19) and (20) hold. The Eq. (19) holds since,

$$\begin{aligned}\beta_A &= H(ID_{ua}\|ID_{ub}\|Y_2^{x_2'})\\ &= H(ID_{ua}\|ID_{ub}\|(g^{y_2})^{x_2'})\\ &= H(ID_{ua}\|ID_{ub}\|(g^{x_2'})^{y_2})\\ &= H(ID_{ua}\|ID_{ub}\|X_2'^{y_2}).\end{aligned}$$

Moreover, the Eq. (20) holds since the amount of sk_B computed by B and S are same as follows:

$$\begin{aligned}sk_B &= H((g^{y_1})^{z_2})\\ &= H((g^{z_2})^{y_1})\\ &= H(Z_2^{y_1}).\end{aligned}$$

After the successful verification, B calculates the session key sk as $H(ID_{ua}\|ID_{ub}\|X_2'^{Y_2}) = H(ID_{ua}\|ID_{ub}\|g^{x_2'y_2})$. It is obvious that, $\mathscr{C}$ can effortlessly calculate the session key as $H(ID_{ua}\|ID_{ub}\|Y_2^{x_2'}) = H(ID_{ua}\|ID_{ub}\|g^{x_2'y_2})$.

It should be noted that the verification result of the server doesn't affect the verification results of the clients, since no message is exchanged between entities after the verification phase. Although the server rejects α_A by checking the Eq. (21) since the random α_A generated by the adversary $\mathscr{C}$ is not equal to $H(ID_{ua}\|sk_A)$, B accepts γ_B and β_A by checking the Eqs. (18) and (19) and end the session by computing a session key. □

4 Our Improved Scheme

As discussed earlier in the preceding section, the main drawback of Youn et al.'s scheme is because of the fact that no message is exchanged between entities after the verification phase. Thus, by rearranging the exchanged messages between entities, the susceptibility of

Youn et al.'s scheme causing impersonation attack can be mitigated. To apply our improvement on Youn et al.'s scheme, Round 1 and Round 2 will remain without change whereas Round 3 and verification process will be changed. The details of the enhanced scheme, outlined in Fig. 3, are as under:

Round 1 $A \longrightarrow S, B$: $\{ID_{ua}||ID_{ub}||X_1||X_2\}$

- A selects two random values $x_1, x_2 \in \mathbb{Z}_q^*$ and computes $X_1 = g^{x_1} s^{pw_A}$ and $X_2 = g^{x_2}$. Then, A sends $\{ID_{ua}||ID_{ub}||X_1||X_2\}$ to B and S.

Round 2 $B \longrightarrow S, A$: $\{ID_{ua}||ID_{ub}||Y_1||Y_2\}$, $S \longrightarrow A$: $\{ID_S||Z_1\}$ and $S \longrightarrow A, B$: $\{ID_S||Z_1\}$, $\{ID_S||Z_2\}$

- Upon reception of broadcast message from A, the user B selects two random values $y_1, y_2 \in \mathbb{Z}_q^*$ and computes $Y_1 = g^{y_1} t^{pw_B}$, $Y_2 = g^{y_2}$. Then B sends $\{ID_{ua}||\, ID_{ub}||\, Y_1||\, Y_2\}$ to A and S.
- At the same time, S computes $Z_1 = g^{z_1} u^{pw_A}$ and $Z_2 = g^{z_2} v^{pw_B}$ for random numbers $z_1, z_2 \in \mathbb{Z}_q^*$, and sends $\{ID_S||Z_1\}$ and $\{ID_S||Z_2\}$ to A and B, respectively.

Round 3 $A \longrightarrow B, S$: $\{\beta_A, \alpha_A\}$, $B \longrightarrow A, S$: $\{\beta_B, \alpha_A\}$, $S \longrightarrow A$: γ_A and $S \longrightarrow B$: γ_B

- A computes $g^{z_1} = Z_1/u^{pw_A}$, $sk_A = H((g^{z_1})^{x_1})$, $\alpha_A = H(ID_{ua}||sk_A)$ and $\beta_A = H(ID_{ua}||ID_{ub}||Y_2^{x_2})$, and broadcasts α_A, β_A to S and B.

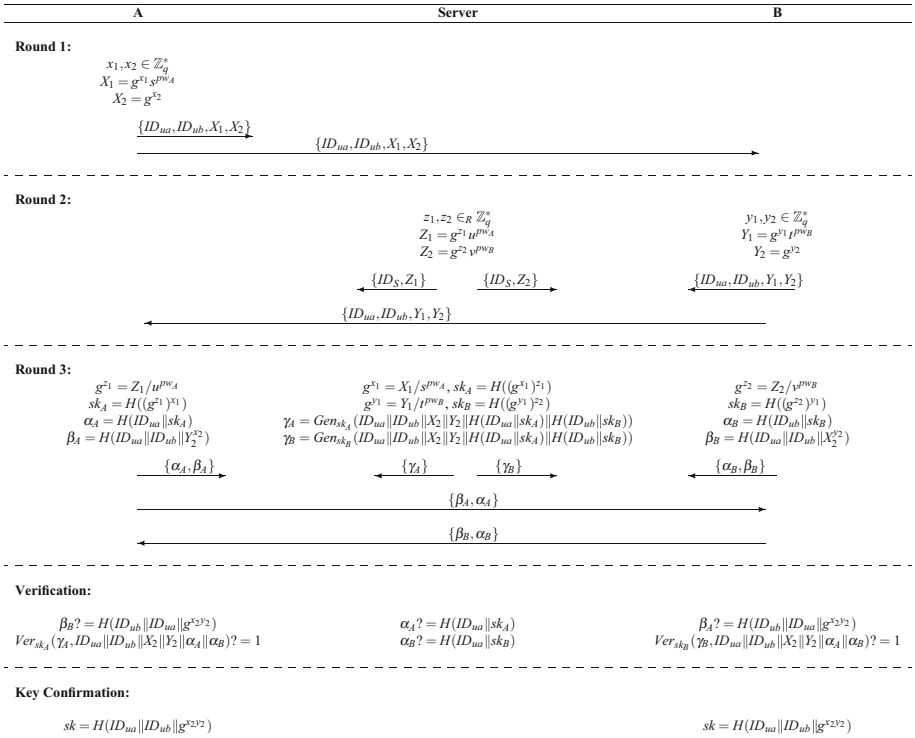

Fig. 3 The proposed improved protocol

- Likewise, B calculates $g^{z_2} = Z_2/v^{pw_B}$, $sk_B = H((g^{z_2})^{y_1})$, $\alpha_B = H(ID_{ub}\|sk_B)$ and $\beta_B = H(ID_{ub}\|ID_{ua}\|X_2^{y_2})$, and broadcasts α_B, β_B to S and B.
- At the same instance, S calculates $g^{x_1} = X_1/s^{pw_A}$, $sk_A = H((g^{x_1})^{z_1})$, $g^{y_1} = Y_1/t^{pw_B}$ and $sk_B = H((g^{y_1})^{z_2})$. Next, S produces two tags $\gamma_A = Gen_{sk_A}(ID_{ua}\|ID_{ub}\|X_2\|Y_2\|H(ID_{ua}\|sk_A)\|H(ID_{ua}\|sk_A))$ and $\gamma_B = Gen_{sk_B}(ID_{ua}\|ID_{ub}\|X_2\|Y_2\|H(ID_{ua}\|sk_A)\|H(ID_{ua}\|sk_A))$. S then transmits γ_A, γ_B to A and B, correspondingly.

Verification Each entity now verifies the legality of participants on other sides.

- If $\beta_B \neq H(ID_{ub}\|ID_{ua}\|g^{x_2y_2})$ or $Ver_{sk_A}(\gamma_A, ID_{ua}\|ID_{ub}\|X_2\|Y_2\|\alpha_A\|\alpha_B) \neq 0$, A culls the scheme.
- If $\beta_A \neq H(ID_{ua}\|ID_{ub}\|g^{x_2y_2})$ or $Ver_{sk_B}(\gamma_B, ID_{ua}\|ID_{ub}\|X_2\|Y_2\|\alpha_A\|\alpha_B) \neq 0$, B culls the scheme.
- If $\alpha_A \neq H(ID_{ua}\|sk_A)$ or $\alpha_B \neq H(ID_{ub}\|sk_B)$, S culls the scheme.

If every condition gets true, S can considers both clients as legal password owners, also A and B consider that their protocol cannot be compromised by an adversary.
Key Confirmation After successful verification,A and B produces a session key by using $g^{x_2y_2}$ as a seed. Then session key is calculated as $sk = H(ID_{ua}\|ID_{ub}\|g^{x_2y_2})$.

5 Security Analysis of Proposed Scheme

5.1 Formal Analysis

Security of proposed scheme is evaluated in this section using random oracle model. Algorithmic assumptions and formal security model that is to be used for proof is presented first.

5.1.1 Security Model

The provable security is utilized in order to evaluate the invincibility of our scheme against well known attacks. The security is verified over the proposed model of Abdalla and Pointcheval [35]. The chosen model can be summarized as follows:

Participants A 3PAKE protocol Π implemented in such a network that is comprised of interconnected participants. These participants can be either a client $U \in \mathscr{U}$ or a reliable server $S \in \mathscr{S}$. For the sake of simplicity only server $\mathscr{S}$ is considered in the network. In separate executions of protocol Π, multiple instances of each participant may exist that are also known as oracle. ith instance of U (resp. S) during single session is referred as Π_U^i (resp. Π_S^i). Each occurrence or instance Π_U^i (resp. Π_S^j) holds partner ID pid_U^i (resp:pid_S^j), a session ID and session key sid_U^i (resp:sid_S^j), sk_U^i (resp:sk_S^j) respectively. Set of identities are designated as pid_U^i (resp:pid_S^j) that are used in ith instance. Sent and received flows are designated as sid_U^i (resp:sid_S^j) using instance Π_U^i (resp. Π_S^j). An instance Π_U^i (resp. Π_S^i) is only accepted if it has session key sk_U^i (resp:sk_S^j), a session identifier and a partner identifier sid_U^i (resp:sid_S^j), pid_U^i (resp:pid_S^j) respectively. Two instances $\Pi_{U_1}^i$ and $\Pi_{U_2}^j$ are assumed *partnered* if and only if (1) both of them have accepted, (2) $pid_{U_1}^i = pid_{U_2}^j$, (3) $sid_{U_1}^i = sid_{U_2}^j$, (4) $sk_{U_1}^i = sk_{U_2}^j$.

Long-Lived Keys Every client $U \in \mathscr{U}$ keeps a password pw_U. Every server $S \in \mathscr{S}$ keeps a vector $pw_S = \langle pw_U \rangle_{U \in \mathscr{U}}$ alongside an entry for every client.

Adversary Model An adversary $\mathscr{M}$ has complete control over communication channel and network., he/she can plan and facilitate the session amid all the parties. The adversary is permitted to initiate the subsequent queries in arbitrary order:

Execute($\Pi^i_{U_1}$, $\Pi^j_{U_2}$, Π^k_S): Passive attacks can be exhibited through this query that enables an attacker to eavesdrops authentic communication among the client instances $\Pi^i_{U_1}$ and $\Pi^j_{U_2}$ and trusted server instance Π^k_S. This query generates output messages that are shared or exchanged during authentic communication of the protocol Π.

SendClient(Π^i_U, m): This query is made to capture a message and then change it, produce new, or simply onward it to the specific client instance Π^i_U. This query yields a message that would have generated from client instance against message m. Moreover, the adversary is permitted to pledge the protocol by entreating SendClient($\Pi^i_{U_1}, (U_1, Start)$).

SendServer(Π^i_S, m): The query results in an active attack over server. The adversary would use this query in order to get the message generated by server instance Π^i_S against message m.

Reveal(Π^i_U): This query initiates session key attack, and is used to steal the session key of the instance Π^i_U.

Corrupt(U): This query is used to retrieve the long-lived key pw_U for specific participant U.

Test(Π^i_U): Adversary is restricted to generate this query once for fresh oracle. Random bit $b \in \{0, 1\}$ is chosen in response to this query. Π^i_U returned the held session key if the condition $b = 1$ gets true, else uniformly selected random value is returned.

Fresh Oracle An oracle Π^i_U is considered new if and only if the following conditions become true: (1) Π^i_U has accepted; (2) Π^i_U or its partner (if exists) has not been asked a Reveal query after their acceptance, and (3) the client who has a partner instance with Π^i_U, has not been issued a Corrupt query.

*3PAKE Security*The game $Game^{3pake}(\Pi, \mathscr{M})$ models the security of a of 3PAKE protocol Π. During this game, $\mathscr{M}$ can initiate multiple queries (that are discussed earlier) to Π^i_U and Π^j_S. If $\mathscr{M}$ initiate a single test query, Test(Π^i_U), in which Π^i_U has accepted and is new, then $\mathscr{M}$ produces a single bit b' output. The major aim of $\mathscr{M}$ is to predict the single bit b accurately during test session. For extra precision, advantage of $\mathscr{M}$ can be defined as follows:

$$Adv^{3pake}_{\Pi,D}(\mathscr{M}) = |2Pr[b' = b] - 1|. \tag{23}$$

The protocol Π is known to be 3PAKE-secure if $Adv^{3pake}_{\Pi,D}(\mathscr{M})$ only insignificantly larger than $O(q_{send})/|D|$, where q_{send} represents the frequency Send queries, and $|D|$ refers to the size of the password dictionary.

5.1.2 Computational Assumption

Decisional Diffie–Hellman (DDH) consideration is defined which is utilized in the security of proposed scheme.

The DDH consideration can be specifically demarcated by couple of experiments, $Exp_{g,q}^{ddh-real}(W)$ and $Exp_{g,q}^{ddh-rand}(W)$. An adversary W is facilitated with g^u, g^v and g^{uv} in the experiment $Exp_{g,q}^{ddh-real}(W)$, and g^u, g^v and g^w in the experiment $Exp_{g,q}^{ddh-rand}(W)$, where u, v and w are yielded randomly from $\mathbb{Z}_q^*$. Delineate the benefit of W in Sacrilegious DDH consideration, $Adv_{g,q}^{ddh}(W)$, as follows:

$$Adv_{g,q}^{ddh}(W) = \max\Big\{|Pr[Exp_{g,q}^{ddh-real}(W) = 1] - Pr[Exp_{g,q}^{ddh-rand}(W) = 1]|\Big\}.$$

5.1.3 Security Proof

Theorem 1 Let D refers to dictionary of passwords with size $|D|$. Moreover, this dictionary of passwords is uniformly distributed. Let Π describes the 3PAKE scheme demarcated in Fig. 3. Consider that DDH supposition holds, Then,

$$Adv_{\Pi,D}^{3pake}(\mathscr{M}) \leq \frac{q_H^2}{2^l} + \frac{(q_s+q_e)^2}{p^2} + 2q_e \cdot Adv_{\alpha,p}^{CDDH}(W) + 2\max\left\{\frac{q_H}{p}, \frac{q_s}{|D|} + \frac{q_s}{2^l}\right\}$$

where q_s represents the frequency of Send queries; q_e represents the frequency of Execute queries; q_H represents the frequency of hash queries to H.

Proof Number of hybrid games has been presented in this proof, beginning at the factual attack G_0 and finishing up at the game G_4 without sparing any advantage for adversary. For every game $G_i (0 \leq i \leq 4)$, $Succ_i$ is defined as an event where $\mathscr{M}$ can accurately predict the bit b in the test session.

Game G_0. This game acts as the real scheme within random-oracle model. All objects A and B and reliable server S are demonstrated as real executing systems within the random-oracle model. Description of event $Succ_i$ states that an adversary can accurately predict the bit b present in the Test-query, we have

$$Adv_{\Pi,D}^{3pake}(\mathscr{M}) = 2\left|\Pr[Succ_0] - \frac{1}{2}\right|. \tag{24}$$

Game G_1. This game is alike game G_0 except hash oracle H is simulated by retaining hash list H_{List} along with entries in the form of (Inp, $Outp$). Against hash query $H(Inp)$ record (Inp, $Outp$) is present within the list H_{List}, return $Outp$. Else, selects $Outp \in \{0,1\}^l$ randomly, transmits it to $\mathscr{M}$ and enter the fresh tuple tuple (Inp, $Outp$) inside H_{List}. All instances are simulated being a real player, or the Send-query and for the Execute, SendClient, SendServer, Reveal, Corrupt and Test queries. From an adversary point of view, one can definitely observe that the game is impeccably indistinguishable as compared to real attack. Therefore,

$$\Pr[Succ_1] = \Pr[Succ_0]. \tag{25}$$

Game G_2. Inside this game, all oracles are simulated within game G_1, instead few games that contains collision on the partial transcripts (X_1, Z_1), (X_2, Y_2), (Y_1, Z_2) or on hash

values are cancelled. As per birthday paradox, $q_H^2/2^{k+1}$ is considered as the utmost probability for occurrence of collisions within outcome of H oracle, where concentrated number of queries to H are represented by q_H. Likewise, $(q_s + q_e)^2/(2p^2)$ is considered as the utmost collisions probability inside the transcript, where q_s refers towards the frequency of queries to the SendClient and SendServer oracles and q_e refers towards the frequency of queries to the Execute oracle. Therefore, we have

$$|\Pr[Succ_2] - \Pr[Succ_1]| \leq \frac{q_H^2}{2^{k+1}} + \frac{(q_s + q_e)^2}{2q^2}. \tag{26}$$

Game G_3. Inside this game, simulation of queries is modified to the SendClient oracle. At first, a session is chosen randomly that is performed by some honest clients A and B for partner instances Π_A^i and Π_B^j.

- When SendClient$(\Pi_A^i, (B, Start))$ is requested, we select arbitrary values $u \in \mathbb{Z}_q^*$, compute $X_2 = g^u$, and X_1 alike the real scheme, and return $\{ID_{ua}, ID_{ub}, X_1, X_2\}$ to $\mathscr{M}$.
- When SendClient$(\Pi_B^j, (ID_{ua}, ID_{ub}, X_1, X_2))$ is requested, we arbitrarily selects $v \in \mathbb{Z}_q^*$ and calculate $Y_2 = g^v$, Y_1 like the real protocol, and return $\{ID_{ua}, ID_{ub}, Y_1, Y_2\}$ to $\mathscr{M}$.
- When SendClient $(\Pi_A^i, (ID_{ua}, ID_{ub}, Y_1, Y_2), (ID_S, Z_1))$ is requested, we calculate g^{z_1}, sk_A and α_A alike the real scheme, and calculate $\beta_A = H(ID_{ua} \| ID_{ub} \| Y_2^u = g^{uv})$ and return $\{\alpha_A, \beta_A\}$ to $\mathscr{M}$.
- When SendClient $(\Pi_B^j, (ID_S, Z_2))$ is requested, we calculate g^{z_2}, sk_B and α_B alike the real protocol, and calculate $\beta_B = H(ID_{ua} \| ID_{ub} \| X_2^v = g^{uv})$ and return $\{\alpha_B, \beta_B\}$ to $\mathscr{M}$.

Therefore, it can be observed definitively this game is impeccably indistinguishable as compared to preceding game G_2. Thus,

$$\Pr[Succ_3] = \Pr[Succ_2]. \tag{27}$$

Game G_4. Inside this game, yet again simulation of queries is modified to the SendClient oracle for the designated session within game G_3. This time, we modify the computation method of the values β_A and β_B, so that they become self-reliant of passwords and ephemeral keys. When SendClient $(\Pi_B^j, (ID_S, Z_2))$ and SendClient $(\Pi_A^i, (ID_{ua}, ID_{ub}, Y_1, Y_2), (ID_S, Z_1))$ are asked, we set $\beta_A = \beta_B = H(ID_{ua} \| ID_{ub} \| g^w)$, where w is nominated from $\mathbb{Z}_q^*$ arbitrarily. The dissimilarity amid the game G_4 and the game G_3 is as under:

$$|\Pr[Succ_4] - \Pr[Succ_3]| \leq q_{exe} \cdot Adv_{\alpha,p}^{DDH}(W). \tag{28}$$

By considering an effective attacker $\mathscr{M}$ to discriminate G_3 and G_4, we establish a DDH solver W.

In game G_4, the parameters β_A and β_B are random and independent with passwords and ephemeral keys. Therefore, three possible cases are stated below through which an adversary differentiate the arbitrary key and the real session key:

Case 1 The queries $(ID_{ua} \| ID_{ub} \| g^w)$ from adversary to H. $q_H/2^k$ represents the occurrence likelihood of this event.

Case 2 The attacker requests SendClient query excluding SendClient(Π_B^j, m) and strongly impersonates A to B. The adversary is permitted to disclose the the static key pw_B of B, whereas he/she isn't permitted to disclose static key pw_A of A. Therefore, the adversary must require the password pw_A of A, so that A can be impersonated. The likelihood is $1 / |D|$. If adversary arbitrarily stabs to impersonate A by calculating sk_A and achieve it successfully, it will create difference but the likelihood is below $1/2^k$. As this type of sessions are utmost q_s, therefore the occurrence likelihood of this event is below $q_s/|D| + q_s/2^k$.

Case 3 The attacker requests SendClient query excluding SendClient(Π_A^i, m) and strongly impersonates B to A. Alike Case 1, the occurrence likelihood of this event is below $q_s/|D| + q_s/2^k$.

As a conclusion,

$$\Pr[Succ_4] = \frac{1}{2} + \max\left\{\frac{q_H}{2^k}, \frac{q_s}{|D|} + \frac{q_s}{2^2}\right\}. \tag{29}$$

By joining all the equations given above, the announced outcome is achieved as under:

$$\begin{aligned}
Adv_{\Pi,D}^{3pake}(\mathscr{M}) &= 2|\Pr[Succ_0] - \frac{1}{2}| \\
&= 2\left|\Pr[Succ_0] - \Pr[Succ_4] + \max\left\{\frac{q_H}{2^k}, \frac{q_s}{|D|} + \frac{q_s}{2^2}\right\}\right| \\
&\leq 2\left(|\Pr[Succ_0] - \Pr[Succ_4]| + \max\left\{\frac{q_H}{2^k}, \frac{q_s}{|D|} + \frac{q_s}{2^k}\right\}\right) \\
&\leq 2(|\Pr[Succ_1] - \Pr[Succ_2]| + |\Pr[Succ_3] - \Pr[Succ_4]| \\
&\quad + \max\left\{\frac{q_H}{2^k}, \frac{q_s}{|D|} + \frac{q_s}{2^k}\right\}\Big) \\
&\leq \frac{q_H^2}{2^k} + \frac{(q_s + q_e)^2}{p^2} + 2q_e \cdot Adv^{DDH}(W) \\
&\quad + 2\max\left\{\frac{q_H}{2^k}, \frac{q_s}{|D|} + \frac{q_s}{2^k}\right\}.
\end{aligned}$$

□

5.2 Informal Analysis and Comparisons

5.2.1 Online Password Guessing Attack

Online password guessing attack can be bifurcated into detectable attack and undetectable attack. Implication of login time intervals can help to prevent detectable attacks but this remedy doesn't work for undetectable attacks. Undetectable attacks can be prevented if every entity validate the correctness of the incoming messages. This is what the proposed scheme has implemented that during verification phase and step 4 the server and the users must validate one another, respectively. Thus it is impossible to launch undetectable online guessing attack over proposed scheme.

5.2.2 Off-Line Password Guessing Attack

Assume a passive adversary who eavesdropped the session between A, B and the server to obtain the exchanged messages. Without loosing generality, we consider the adversary

wants to guess pw_A. As it is obvious, pw_A is used to compute $X_1 = g^{x_1} s^{pw_A}$ and $Z_1 = g^{z_1} u^{pw_A}$. If the adversary predicts a password pw'_A, he/she cannot check its correctness because g^{x_1} and g^{z_1} are unknown for him/her. Besides, an active adversary also cannot guess the password. Assume an adversary who predicts a password pw'_A to compute $X'_1 = g^{x'_1} s^{pw'_A}$ and sends X'_1 to the server. To check the correctness of the predicted password, he/she has to pass the verification process in Round 4. To do so, the adversary should compute sk_A. On the other hand, if an active adversary guesses a password pw'_A to compute $Z'_1 = g^{z'_1} u^{pw'_A}$ and sends Z'_1 to A. The attacker cannot check the correctness of the guessed password because he faces to compute sk_A.

5.2.3 Replay Attack

Consider an active adversary $\mathscr{A}$ catches the message of A or B in Round 1 or Round 2 and then send it as a replay to masquerade each of them. But, $\mathscr{A}$ cannot calculate an accurate $sk_A = H(g^{x_1 z_1})$ or $sk_B = H(g^{y_1 z_2})$ and deliver α_A or α_B to S in Round 3 until he/she can accurately predict the password pw_A or pw_B and predict z_1 or z_2 from Z_1 or Z_2. Whenever $\mathscr{A}$ attempts to predict z_1 or z_2 from Z_1 or Z_2, he/she will confront the untraceable Discrete Logarithm Problem (DLP). Thus, replay attack is impossible on the proposed scheme.

5.2.4 Denning–Sacco Attack

Though an adversary $\mathscr{A}$ may attain the session key $SK = h(ID_{ua}, ID_{ub}, g^{x_2 y_2})$, for particular reasons, he/she fails to attain the secret passwords pw_A and pw_B, because SK is independent of the passwords.

5.2.5 Impersonation Attack

It is infeasible for an adversary $\mathscr{A}$ to impersonate being a server without users private password due to which, an adversary fails to calculate Z_1 and Z_2. $\mathscr{A}$ also fails to masquerade the users, because he/she cannot construct X_1 and α_A for the user A or Y_1 and α_B for the user B without having pw_A and pw_B. Thus, it can be concluded that proposed scheme is invincible against impersonation attack.

5.2.6 Modification Attack

Exchanged messages amid users and the server cannot be altered by an adversary $\mathscr{A}$. This is due to the fact that any change can easily be detected by the users and server by validating the incoming message in Round 4 and verification.

5.2.7 Known-Key Security

In this kind of attack adversary attempts to calculate the subsequent session keys on the basis of some preceding session keys. Consider an adversary $\mathscr{A}$ is aware of some preceding session key but this doesn't help him/her to compute subsequent session keys due to unique random numbers x_2, y_2 are used in every session. Thus proposed scheme gratifies the known-key security.

5.2.8 Perfect Forward Secrecy

Perfect forward secrecy can be defined as, if long-term secret keys of a single or multiple entities are exposed but the secrecy of preceding session keys remain intact. The proposed scheme gratifies the perfect forward secrecy as adversary fails to find the previous session keys because he/she doesn't have shared secret $g^{x_2y_2}$ even if he/she have pw_A and pw_B.

6 Performance and Security Comparison

We assess the competency and security of proposed scheme, compare its performance with some related 3PAKE protocols which use neither symmetric key cryptography nor server public key. Table 2 figure out the performance comparisons of proposed scheme and Pu et al.'s protocol [31] and Youn's protocol [34]. Two main operations are adopted in this analysis and they are defined as follows. *H*: The hash function operation; *M*: The modular exponentiation operation. In the proposed scheme, we suppose that the server computes and stores u^{pw_U}, v^{pw_U}, s^{pw_U} and t^{pw_U} for each user *U* in the registration phase. Thus, when the server wants to use each of these parameters, it is not needed to compute a modular exponentiation. According to this assumption, the computational cost of our protocol is same as Youn et al.'s protocol and equal to 16M + 18H.

Table 3 lists the security assessment of proposed scheme w.r.t to related schemes. The figures reveals and verify that proposed scheme is composed of many beneficial features and proved to be extra secure as compared to related schemes.

7 Conclusions

This paper briefly reviewed Youn et al.'s 3PAKE scheme with round efficiency. It is demonstrated that Youn et al.'s scheme is susceptible to impersonation attack. Then, an improved 3 PAKE scheme is proposed. The proposed scheme is having same computation

Table 2 Computation comparison

	User A	Server	User B	Total
Pu et al.'s [31]	5M + 4H	7M + 2H	5M + 4H	17M + 10H
Youn et al.'s [34]	6M + 6H	4M + 6H	6M + 6H	16M + 18H
Ours	6M + 6H	4M + 6H	6M + 6H	16M + 18H

Table 3 Security comparison

	Pu et al.'s scheme [31]	Youn et al.'s scheme [34]	Our scheme
Replay attack resists	✔	✔	✔
Impersonation attack resists	×	×	✔
Guessing attacks resists	✔	✔	✔
Denning–Sacco attack resists	✔	✔	✔
Modification attack resists	✔	✔	✔
Known-key attack resists	✔	✔	✔
Forward secrecy provides	✔	✔	✔
Provably secure provides	✔	×	✔

cost while maintaining round efficiency. The security of proposed scheme is validated in random oracle model.

References

1. Farash, M. S., Bayat, M., & Attari, M. A. (2011). Vulnerability of two multiple-key agreement protocols. *Computers & Electrical Engineering*, *37*(2), 199–204.
2. Farash, M. S., Attari, M. A., & Bayat, M. (2012). A certificateless multiple-key agreement protocol without one-way hash functions based on bilinear pairings. *IACSIT International Journal of Engineering and Technology*, *4*(3), 321–325.
3. Farash, M. S., Attari, M. A., Atani, R. E., & Jami, M. (2013). A new efficient authenticated multiple-key exchange protocol from bilinear pairings. *Computers & Electrical Engineering*, *39*(2), 530–541.
4. Farash, M. S., & Attari, M. A. (2013). Provably secure and efficient identity-based key agreement protocol for independent PKGs using ECC. *The ISC International Journal of Information Security*, *5*(1), 1–15.
5. Farash, M. S., & Attari, M. A. (2014). A pairing-free ID-based key agreement protocol with different PKGs. *International Journal of Network Security*, *16*(2), 143–148.
6. Chen, B. L., Kuo, W. C., & Wuu, L. C. (2012). A secure password-based remote user authentication scheme without smart cards. *Information Technology and Control*, *41*(1), 53–59.
7. Jiang, Q., Ma, J., Li, G., & Ma, Z. (2013). An improved password-based remote user authentication protocol without smart cards. *Information Technology and Control*, *42*(2), 150–158.
8. Bayat, M., Farash, M. S., & Movahed, A. (2010). A Novel Secure bilinear pairing based remote user authentication scheme with smart card. In *IEEE/IFIP international conference on embedded and ubiquitous computing (EUC)* (pp. 578–582).
9. Farash, M. S., & Attari, M. A. (2013). An enhanced authenticated key agreement for session initiation protocol. *Information Technology and Control*, *42*(4), 333–342.
10. Farash, M. S., & Attari, M. A. (2013). Cryptanalysis and improvement of a chaotic maps-based key agreement protocol using Chebyshev sequence membership testing. *Nonlinear Dynamics*. doi:10.1007/s11071-013-1204-1.
11. Xie, Q., Dong, N., Tan, X., Wong, D. S., & Wang, G. (2013). Improvement of a three-party password-based key exchange protocol with formal verification. *Information Technology and Control*, *42*(3), 231–237.
12. Liu, T., Pu, Q., Zhao, Y., & Wu, S. (2013). ECC-based password-authenticated key exchange in the three-party setting. *Arabian Journal for Science and Engineering*, *38*(8), 2069–2077.
13. Chien, H. Y., & Wu, T. C. (2009). Provably secure password-based three-party key exchange with optimal message steps. *Computer Journal*, *52*(6), 646–655.
14. Lee, T. F., Liu, J. L., Sung, M. J., Yang, S. B., & Chen, C. M. (2009). Communication-efficient three-party protocols for authentication and key agreement. *Computers & Mathematics with Applications*, *58*(4), 641–648.
15. Xiong, H., Chen, Y., Guan, Z., & Chen, Z. (2013). Finding and fixing vulnerabilities in several three-party password authenticated key exchange protocols without server public keys. *Information Sciences*, *235*(1), 329–340.
16. Chen, H. B., Chen, T. H., Lee, W. B., & Chang, C. C. (2008). Security enhancement for a three-party encrypted key exchange protocol against undectectable on-line password guessing attacks. *Computer Standards & Interfaces*, *30*(1–2), 95–99.
17. Zhao, J., & Gu, D. (2012). Provably secure three-party password-based authenticated key. *Information Sciences*, *184*(1), 310–323.
18. Yang, J. H., & Cao, T. J. (2012). Provably secure three-party password authenticated key exchange protocol in the standard model. *Journal of Systems and Software*, *85*(2), 340–350.
19. Kim, H. S., & Choi, J. Y. (2009). Enhanced password-based simple three-party key exchange protocol. *Computers & Electrical Engineering*, *35*(1), 107–114.
20. Nam, J., Paik, J., Kang, H. K., Kim, U. M., & Won, D. (2009). An off-line dictionary attack on a simple three-party key exchange protocol. *IEEE Communications Letters*, *13*(3), 205–207.
21. Nam, J., Paik, J., & Won, D. (2011). A security weakness in Abdalla et al.'s generic construction of a group key exchange protocol. *Information Sciences*, *181*(1), 234–238.
22. Lou, D. C., & Huang, H. F. (2010). Efficient three-party password-based key exchange scheme. *International Journal of Communication Systems*, *24*(4), 504–512.

23. Huang, H. F. (2009). A simple three-party password-based key exchange protocol. *International Journal of Communication Systems*, *22*(7), 857–862.
24. Yoon, E. J., & Yoo, K. Y. (2011). Cryptanalysis of a simple three-party password-based key exchange protocol. *International Journal of Communication Systems*, *24*(4), 532–542.
25. Wu, S., Chen, K., & Zhu, Y. (2013). Enhancements of a three-party password-based authenticated key exchange protocol. *International Arab Journal of Information Technology*, *10*(3), 215.
26. Lee, T. F., & Hwang, T. (2010). Simple password-based three-party authenticated key exchange without server public keys. *Information Sciences*, *180*(9), 1702–1714.
27. Chang, T. Y., Hwang, M. S., & Yang, W. P. (2011). A communication-efficient three-party password authenticated key exchange protocol. *Information Sciences*, *181*(1), 217–226.
28. Wu, S., Pu, Q., Wang, S., & He, D. (2012). Cryptanalysis of a communication-efficient three-party password authenticated key exchange protocol. *Information Sciences*, *215*(1), 83–96.
29. Tso, R. (2013). Security analysis and improvements of a communication-efficient three-party password authenticated key exchange protocol. *The Journal of Supercomputing*. doi:10.1007/s11227-013-0917-8.
30. Chien, H. (2011). Secure verifier-based three-party key exchange in the random oracle model. *Journal of Information Science and Engineering*, *27*(4), 1487–1501.
31. Pu, Q., Wang, J., Wu, S., & Fu, J. (2013). Secure verifier-based three-party password-authenticated key exchange. *Peer-to-Peer Networking and Applications*, *6*(1), 15–25.
32. Tallapally, S. (2012). Security enhancement on simple three-party PAKE protocol. *Information Technology and Control*, *41*(1), 15–22.
33. Farash, M. S., & Attari, M. A. (2014). An efficient and provably secure three-party password-based authenticated key exchange protocol based on Chebyshev chaotic maps. *Nonlinear Dynamics*. doi:10.1007/s11071-014-1304-6.
34. Youn, T. Y., Kang, E. S., & Lee, C. (2013). Efficient three-party key exchange protocols with round efficiency. *Telecommunication Systems*, *52*(2), 1367–1376.
35. Abdalla, M., & Pointcheval, D. (2005). Interactive Diffie–Hellman assumptions with applications to password-based authentication. In *Proceedings of FC'05, LNCS 3570* (pp. 341–356).

Mohammad Heydari received the M.Sc. degree in Electrical Engineering from Islamic Azad University, Sciences and Research Branch, Tehran, I.R. Iran. Currently, he is a Ph.D. candidate in Electrical Engineering in Shahid Beheshti University Tehran, I.R. Iran. His research interests includes Cryptography, Authentication, provable security models and lightweight Cryptography.

S. Mohammad Sajad Sadough received the B.Sc. degree in Electrical Engineering (electronics) from Shahid Beheshti University, Tehran, I.R. Iran in 2002 and the M.Sc. and the Ph.D. degrees in Electrical Engineering (telecommunication) from Paris-Sud 11 University, Orsay, France, in 2004 and 2008, respectively. From 2004 to 2007, he was jointly with the National Engineering School in Advanced Techniques (ENSTA), Paris, France, and the Laboratory of Signals and Systemes (LSS), at Supélec, Gif-sur-Yvette, France. He was a lecturer with the Depertment of Electronics and Computer Engineering (UEI), ENSTA, where his research activities were focused on improved reception schemes for ultra-wideband communication systems. From December 2007 to September 2008, he was a postdoctoral researcher with the LSS, Supélec-CNRS, where he was involved in the European research project TVMSL with Alcatel-Lucent France. Since October 2008, he has been with the Faculty of Electrical & Computer Engineering, Shahid Beheshti University, where he is currently an Assistant Professor with the Department of Telecommunication. Dr. Sadough's areas of research include signal processing, communication theory, and digital communication.

Mohammad Sabzinejad Farash received the B.Sc. degree in Electronic Engineering from Shahid Chamran College of Kerman in 2006, and the M.Sc. degree in Communication Engineering from I. Hussein University in 2009. He also received the Ph.D. degree in Cryptographic Mathematics at the Department of Mathematics and Computer Sciences of Tarbiat Moallem University in Iran in 2013. His research interests are Security Protocols and Provable Security Models.

Shehzad Ashraf Chaudhry received his M.S. Computer Science with distinction, from International Islamic University Islamabad, Pakistan in 2009 and was awarded *Gold Medal*. Currently he is working as Lecturer and Ph.D. scholar at the Department of Computer Science & Software Engineering, International Islamic University, Islamabad. He authored more than 20 scientific publications published in different international journals and proceedings, including 12 in SCI/E journals. His research interests include Lightweight Cryptography, Elliptic/Hyper Elliptic Curve Cryptography, Multimedia Security, MANETs, SIP authentication, IP Multimedia sub-system and Next Generation Networks.

Khalid Mahmood received the B.S. degree in Computer Science from Virtual University, Lahore, Pakistan and the M.S. degree in Computer Science from Riphah International University, Islamabad, Pakistan, respectively in 2007 and 2010. He is pursuing Ph.D. degree in Computer Science from International Islamic University, Islamabad, Pakistan. His research interests are in Smart Grid authentication and information security.