Mobile Networks and Applications
https://doi.org/10.1007/s11036-018-1145-5

Identity-Based User Authenticated Key Agreement Protocol for Multi-Server Environment with Anonymity

Alzubair Hassan[1] · **Anyembe Andrew Omala**[1] · **Mohamed Ali**[2] · **Chunhua Jin**[3] · **Fagen Li**[1]

Abstract

A multi-server environment is an important application paradigm in the Internet of Things (IoT). It enables a user access services from different vendors without having to go through multiple registration. The privacy of one who desires to access these services is often crucial. In order to access this service in a manner that assures user privacy, a user needs to be anonymously authenticated independent of the vendors' services. However, existing identity-based anonymous schemes are only suitable for the client-server domain. Moreover, these schemes provide conditional anonymity which presupposes that if an adversary discovers the user's private key, the identity can easily be recovered and misused. To avoid this situation, a new unconditional anonymity identity-based user authenticated key agreement scheme for IoT multi-server environment is introduced in this paper. Our protocol applies a ring signature to allow users to anonymously authenticate themselves in the severs without revealing their identities. Hence, an adversary cannot recover the user's identity even when the user's private key is known. We further provide a security proof in the random oracle model. Compared with the existing protocols, our proposed scheme is well fitting for mobile phone applications and guarantees the privacy of users in IoT multi-server domain.

Keywords Anonymous user authentication · Multi-server environment · Bilinear pairing · Random oracle model

1 Introduction

The Internet of Things (IoT) is a growing concept applied to establish a robust network of devices, all entrenched with electronics, sensors etc. that enable them to exchange and analyze data. Recently, mobile devices are widely used in our daily lives as part of IoT to receive services from servers such as in e-government, health monitoring, smart home, smart city and electronic medical applications. In the IoT, a multi-server environment provides several services which can be accessed through different wireless networks by the client.

Whenever the remote user looks forward to access services from numerous server in IoT environment, specific identity and password must be registered for each server to provide legitimate access. However, it is difficult for the user to remember the access passwords for all the servers especially when large numbers are involved. Several schemes [1–3] have been presented to address the remote user authentication in a single client-server environment. However, these schemes are only applicable in a single client-server environment [4]. Therefore, it is necessary to offer user authentication protocol that can work effectively in multi-server environment.

The concept of signature has been used to provide user authentication. In the traditional signature, the sender uses his private key to sign the message. Then, the receiver verifies the sender's signature by using the public key assigned. Therefore, the receiver has prior knowledge of the sender's identity. In a multi-server environment where IoT

✉ Fagen Li
fagenli@uestc.edu.cn

Alzubair Hassan
alzubairuofk@gmail.com

Anyembe Andrew Omala
andromala@gmail.com

Mohamed Ali
mody231279@yahoo.com

Chunhua Jin
xajch0206@163.com

[1] Center for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

[2] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

[3] The Laboratory for Internet of Things and Mobile Internet Technology of Jiangsu Province, Huaiyin Institute of Technology, Huaian 223003, China

Published online: 09 October 2018

operates, knowledge of user identity by the server poses considerable risk to the privacy. However, with the concept of ring signature [5], the sender chooses the set of users and values related to these users. Then, the sender uses his private key to connect the values in series as well as to sign the message using these values. Then, the receiver verifies the ring signature by computing the values related to the set of the users from the message and the possible users' public key. Thus, the receiver knows that this message is signed by one of the numerous independent senders. But, the receiver cannot distinguish the actual signer. The signer in a ring signature scheme randomly chooses the users in the process of signing the message and advance knowledge of users is not required. Therefore, the signature verification should be correct and should not give any information about the singer's identity.

Anonymity has the following five security levels as mentioned in [6]: (a): In level 0 (void anonymity), the identity of the users is not allowed to hide, that means there is no anonymity; (b): In level 1 (apparent anonymity), the users can provide anonymity using indirect personal information about their identities; (c): In level 2 (revocable anonymity) the personal information about a special group of users acts as trusted entities without rejecting high-level anonymity; (d): In level 3 (conditional anonymity), the users can hide their identities if they follow the policies and the rules e.g. if the users encrypt their identities, and an adversary knows the user's private keys, he/she can recover the identities of the users; (f): In level 4 (unconditional anonymity), the users can recognize a service as unconditionally anonymous e.g. if an adversary knows the user's private keys, he/she cannot recover their identities.

Indeed, user authentication, key agreement, mutual authentication and privacy-preserving should be ensured in the IoT environment to offer authentication, data confidentiality, integrity, non-repudiation, and privacy preservation. For instance, in the electronic medical systems, to keep the privacy of the clients from the medical system provider and network administrator, it is important to protect the identities and limit access to locations of the clients. Thus, we need to offer user authentication for this environment that can provide unconditional anonymity as well as non-traceability.

1.1 Related work

The identity-based cryptography (IBC) [7] was introduced to overcome the problems associated with the traditional public-key cryptography. To eliminate the complexity of the digital certificates, the IBC applies user's attributes such as email addresses or phone numbers as public keys while the private keys are created by the private key generator (PKG). Therefore, the user's keys are critical for identification and do not need to be revoked. Since then the use of IBC in the design of user authentication schemes remain active till now. In 2001, Li et al. [8] introduced a user authentication for the multi-server environment based on neural networks. In 2004, Juang [9] demonstrated that Li et al.'s protocol is not suitable for practical application because of huge communication as well as processing amount required in training the network. Therefore, he introduced an authentication protocol using a hash function and a symmetric key cryptosystem. Unfortunately, Chang and Lee [10] showed that Juang and Lee's protocol is vulnerable to off-line dictionary attack. They proposed a protocol to overcome Juang and Lee's protocol security vulnerability. Since then, Liao and Wang [11] proposed a dynamic identity-based authentication protocol for a multi-server environment. They claimed that their protocol could resist various attacks. But, Hsiang and Shih [12] found that Liao and Wang's protocol is vulnerable to an insider attack, masquerade attack, server spoofing attack, and registration center's spoofing attack. To enhance the security, they proposed an improved protocol. In 2011, Sood et al. [13] demonstrated that Hsiang and Shih's protocol cannot provide mutual authentication, and is vulnerable to masquerade attack and server spoofing attack. Also, they proposed a protocol to improve the security. Li et al. [14] pointed out that Sood et al.'s protocol is vulnerable to leak-of-verifier attack and stolen smart card attack. To overcome this weakness, they proposed an improved protocol. However, Han [15] pointed out that Li et al.'s protocol is vulnerable to the replay attack, password guessing attack, and masquerade attack. In 2013, Yoon and Yoo [16] proposed a biometric user authentication protocol with a key agreement for smart cards without a verification table to reduce the complication of the hash operation between all users. To solve the previous security weaknesses, Khan et al. [17] proposed a new dynamic identity-based authentication protocol using elliptic curve cryptography (ECC). Han and Zhu [18] proposed a new identity-based mutual authentication protocol without bilinear pairings to improve the performance. He and Wang [19] proposed a biometric-based authentication protocol for the multi-server environment using ECC. They claimed that their protocol could overcome weaknesses in previous schemes at the computation and communication costs. Shen et al. [20] found that Yoon and Yoo's protocol is not secure against three kinds of attacks and proposed an improved scheme for the multi-server environment using

biometrics and ECC. In 2017, Tseng et al. [21] proposed a user authentication and key agreement protocol based on identity-based cryptosystem. They claimed that their protocol resists to the ephemeral secret leakage (ESL) attacks in mobile multi-server environments. Furthermore, their protocol requires the lowest communication overhead. Anonymous user authentication has been studied widely. Some user anonymous protocols were presented based on smart card and biometrics [22, 23], self-certified public key cryptography [24, 25], identity-based public key cryptography [26, 27], and chaotic map [28, 29]. According to the anonymity classification in [6] none of the works mentioned above can entirely satisfy the unconditional requirement of maintaining user privacy.

Since Rivest et al. [30] introduced the ring signature scheme based on public key infrastructure (PKI), several ID-based ring signature schemes have been proposed to reduce the computational cost related to PKI. Zhang and Kim [31] introduced the first ID-based ring signature. Later, Lin and Wu [32] suggested another efficient scheme. suggested a more efficient ID-based ring signature scheme. Unfortunately, a study conducted by Awasthi and Lol [33] found that [31] and [32] have some problems and which were rectified in their proposed scheme. In a related development, Herranz and Saez [34] introduced a new scheme for anonymous subsets. Subsequently, Chow et al. [35] introduced the ID- based threshold ring signature scheme. However, introduced the ID- based threshold ring signature scheme. However, the above mentioned schemes which depends on the size of the group are associated with high computational cost. To overcome this issue Chow et al. [5] proposed a scheme that requires only two bilinear pairings and hence computationally efficient.

1.2 Contribution

This work introduces a new user authentication protocol and key agreement for multi-server environment with anonymity. To provide unconditional anonymity to IoT multi-server environment the proposed protocol uses a ring signature to allow users to anonymously authenticate themselves in the severs without revealing their identities. Though the servers recognize the client as a member of the ring, the exact identity is unknown. However, in the traditional group signature where anonymity is conditional, the possibility of knowing the client's identity can compromise the security of user privacy. Therefore, the application of our proposed scheme can assure the privacy of client for improved user confidence in a multi-server environment in which IoT operates.

1.3 Organization

This paper is organized as follows. In Section 2, preliminaries are given. In Sections 3 and 4, the proposed scheme and its security analysis are presented respectively. In Section 5, the performance analysis is provided. In Section 6, the application scenario is given. Finally, a conclusion is drawn in Section 7.

2 Preliminaries

2.1 Bilinear pairings

While $\mathbb{G}_1$ is the additive group and $\mathbb{G}_2$ is the multiplicative group of the exact prime order q, the bilinear pairing function can be illustrated as $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and P is the generator of $\mathbb{G}_1$. The bilinear pairing properties as described by [36, 37] are as follow:

1. **Bilinearity :** While $\forall a, b \in \mathbb{Z}_q^*$ and for all $Q, P \in \mathbb{G}_1$, then the bilinearity is given as $e(aQ, bP) = e(Q, P)^{ab}$
2. **Non-degeneracy:** While $Q, P \in \mathbb{G}_1$ and $1_{\mathbb{G}_2}$ is the identity of $\mathbb{G}_2$, then the non-degeneracy property is given as $e(Q, P) \neq 1_{\mathbb{G}_2}$.
3. **Computability:** The $e(Q, P)$ is processed efficiently wh-ere for all $Q, P \in \mathbb{G}_1$.

Computational Diffie-Hellman (CDH) Problem Where $\forall a, b \in \mathbb{Z}_q^*$, $(P, aP, bP) \in \mathbb{G}_1$ is given to compute abP.

Bilinear Diffie-Hellman (BDH) Problem Given $e(P, aP, bP, cP) \in \mathbb{G}_1$, where $\forall a, b, c \in \mathbb{Z}_q^*$ to compute $e(P, P)^{abc}$.

2.2 The algorithm of our protocol

Our protocol is located by *Setup* phase, *Key extract* phase, and *user authenticated key agreement* phase. The algorithm of our protocol is given as follows:

– *Setup*(1^λ): This phase is executed by the registration center (RC). RC takes a secure parameter λ as its input. RC generates a master private key x corresponding to the master public key P_{pub}, and public parameters $params$. Then, RC publishes $params$, P_{pub}, and keeps x secret.
– *Key extract*: this phase is executed by the RC. RC takes as inputs the system parameters $params$, the master private key, and the user's identity or the server's identity ID_u where $u \in \{C_i, S_j\}$. Then, the RC returns

the private key D_u and the public key Q_u to the user or the server. Upon receiving D_u, the user and the server can verify their validity.

– *User authenticated key agreement*: This phase is executed by the user and the server to authenticate from each other and to agree on a session key for use in the future communication.

2.3 Security model

The abilities of an adversary $\mathscr{A}$ and the security requirements for mutual authentication and key exchange are described in this section. know that an instance λ of a member u as been defined as Π_u^λ. The challenger $\mathscr{F}$ responds to the adversary $\mathscr{A}$ queries as follows:

1. *Setup*(1^λ): The algorithm takes as input a security parameter λ. $\mathscr{F}$ executes $Setup$ algorithm to generate a master secret key $x \in \mathbb{Z}_q^*$, a master public key P_{pub} and system parameters $params$. System parameters $params$ are delivered to adversary $\mathscr{A}$ with x remaining as secret.
2. *Probing*: Then, $\mathscr{A}$ can demonstrate polynomial limited queries in an adaptive manner:
   (a) *Extract* (ID_u) *query*: $\mathscr{A}$ could obtain the private key of other identity ID_u except the targeted identity ID_t.
   (b) *Send* (Π_u^λ, M) query: Whenever a message M is sent based on our proposed from $\mathscr{A}$ to $\mathscr{F}$, then $\mathscr{F}$ makes the computation and responds to $\mathscr{A}$.
   (c) *Reveal* (Π_u^λ, M) query: A session key sk is accepted by $\mathscr{A}$ from $\mathscr{F}$. In the case it hasn't, it replies a null.
   (d) *Corrupt* (u) query: For the aim of compromising the user's private key, a member u is created a *Corrupt* query by $\mathscr{A}$ to $\mathscr{F}$.
   (e) *Test* (Π_u^λ) query: $\mathscr{F}$ tosses a fair coin w after a single *Test* query is sent from $\mathscr{A}$. In the case where $w = 1$, the session key sk is obtained by $\mathscr{A}$. Else, a random string is received. The semantic security of sk is computed With this query.

Aftermath, $\mathscr{A}$ produces w' as estimation for w. Adv$(\mathscr{A})$ is described as the advantage of $\mathscr{A}$ mathematically as $\text{Adv}(\mathscr{A}) = |\Pr[w' = w] - 1/2|$ such that, the probability that $w' = w$ is denoted as $\Pr[w' = w]$.

3 Proposed protocol

Our protocol is located by a setup phase, a key extract phase and a user authenticated key agreement phase. We have used Chow et al.'s identity-based ring signature [5]. This paper uses the notations in Table 1. The proposed protocol's phases are illustrated as follows:

Table 1 Symbols

Notation	Explanation
λ	A security parameter
$\mathbb{G}_1$	A cyclic additive group
$\mathbb{G}_2$	A cyclic multiplicative group
q	A group's prime order of $\mathbb{G}_1$ and $\mathbb{G}_2$
P	A generator of $\mathbb{G}_1$
e	A bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$
H_i	An one way hash function, where $i = 1, 2, 3$
RC	Registration Center
x	RC's master secret key
P_{pub}	RC's master public key
u	A client's or server's identity $u \in \{C_i, S_j\}$
ID_u	participants' identity
D_u	A client's or server's private key
Q_u	A client's or server's public key
dx	Actual signer's index
ID_t	A challenged identity

3.1 Setup phase

This phase is executed by the RC as follows :

1. Take as input a security parameter λ and generate the parameters.
2. Choose two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same prime order q and the bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. P is a generator of $\mathbb{G}_1$.
3. Choose $x \in \mathbb{Z}_q^*$ as master secret key, set $P_{pub} = xP$, and select three cryptographic secure hash function $H_1 : \{0,1\}^* \to \mathbb{G}_1$, $H_2 : \{0,1\}^* \times \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{Z}_q^*$, $H_3 : \mathbb{G}_1 \times \{0,1\}^* \times \mathbb{Z}_q^* \times \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{Z}_q^*$ and $H_4 : \mathbb{G}_1 \times \{0,1\}^* \times \mathbb{Z}_q^* \times \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{Z}_q^* \to \mathbb{Z}_q^*$.
4. Publish $\{G_1, G_2, q, e, P, P_{pub}, H_1, H_2, H_3, H_4\}$ as the public parameters.

3.2 Key extract

Figure 1 depicts the key extract phase. The phase is executed by the RC as follows:

1. A user u sends his identity ID_u to the RC. Then, the RC computes $Q_u = H_1(ID_u)$ as a user's public key and

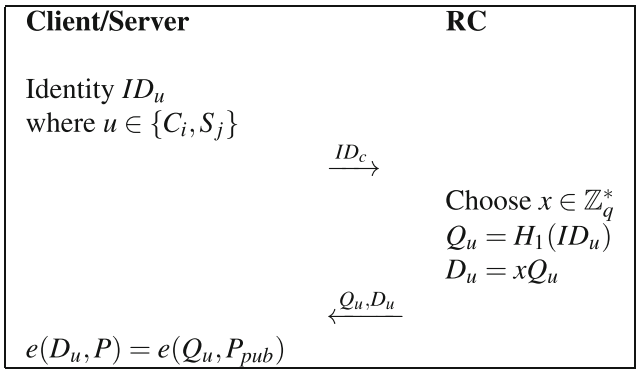

Fig. 1 Key extract phase

$D_u = xQ_u$ as a private key, where u is either a client or a server $u \in \{C_i, S_j\}$, $\forall i \in \{1, 2, 3, \ldots, n\}$ and $\forall j \in \{1, 2, 3, \ldots, y\}$.

2. RC sends the private keys to the ID_u using a secure channel, or using the proposed secure and the anonymous protocol of [38].
3. A user ID_u verifies the validity of D_u by determining if the equality $e(D_u, P) = e(Q_u, P_{pub})$ holds.

3.3 User authenticated key agreement

Here, the cooperation between the server and the client to authenticate from each other is described in Fig. 2. In this subsection our scheme offers the authentication and the key agreement. Most of the schemes [1–3] provide the authentication by using the traditional signature. Here, we have used the ring signature rather than the traditional signature to provide more features to our scheme such as unconditional anonymity and non-traceability. The procedures involved are listed as:

1. The client chooses his index dx, and $r_i \in \mathbb{Z}_q^*$. The identities of n users are denoted as $T = \{ID_1, ID_2, ID_3, \ldots, ID_n\}$ and the groups of users' identities m are denoted as $\cup\{U_i\}$, where, $1 \leq i \leq n$, $U_i = \cup\{ID_{i_j}\}$. Then, the client computes $U_i = r_i P$, $k_1 = e(r_1 P_{pub}, Q_{S_j})$, $h_i = H_2(T, U_i, k_1) \forall i \in \{1, 2, 3, \ldots, n\}$ except dx, and $U_{dx} = r_{dx} Q_{ID_{dx}} - \sum_{i \neq dx}(U_i + h_i Q_{ID_i})$. Finally, the client sends (T, U_i) to the server.
2. After (T, U_i) is received, $\alpha \in \mathbb{Z}_q^*$ is chosen by the server, computes $k_2 = e(U_1, D_{S_j})$, and $Auth = H_3(P_{pub}, T, \alpha, U_1, k_2)$. And finally, $(\alpha, Auth)$ is sent to the client by the server.
3. After $(\alpha, Auth)$ is received, the client authenticate the equality of $Auth = H_3(P_{pub}, T, \alpha, U_1, k_1)$. A common session key $sk = H_4(P_{pub}, T, \alpha, U_1, k_1, Auth)$ and $h_{dx} = H_2(T, U_{dx}, k_1)$ are computed by the client. Then, the client computes $V = (h_{dx} + r_{dx}) D_{ID_{dx}}$ and sends it to the server.
4. Upon receiving V, the server computes $h_i = H_2(T, U_i, k_2)$ to verify if $e(P_{pub}, \sum_{i=1}^{n}(U_i + h_i Q_{ID_i})) = e(P, V)$ holds. The server computes the common session key $sk = H_4(P_{pub}, T, \alpha, U_1, k_2, Auth)$.

3.4 Correctness of the proposed protocol

To ascertain that the presented scheme is correct, $e(P, V) = e(P_{pub}, \sum_{i=1}^{n}(U_i + h_i Q_{ID_i}))$ is verified, where $V = (h_{dx} +$

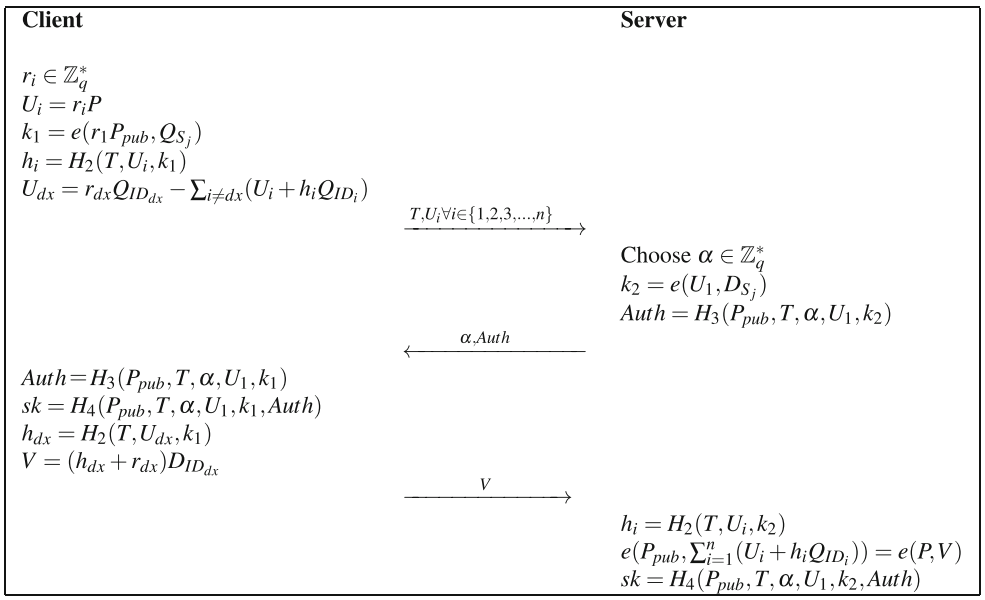

Fig. 2 User authenticated key agreement

$r_{dx})D_{ID_{dx}}$, $D_u = xQ_u$ and $U_{dx} = r_{dx}Q_{ID_{dx}} - \sum_{i \neq dx}(U_i + h_i Q_{ID_i})$ then we have

$$\begin{aligned} e(P, V) &= e(P, (h_{dx} + r_{dx})D_{ID_{dx}}) \\ &= e(P, (h_{dx} + r_{dx})xQ_{ID_{dx}} \\ &= e(xP, (h_{dx} + r_{dx})Q_{ID_{dx}} \\ &= e(P_{pub}, (h_{dx}Q_{ID_{dx}} + r_{dx}Q_{ID_{dx}}) \\ &= e\left(P_{pub}, (h_{dx}Q_{ID_{dx}} + U_{dx} + \sum_{i \neq dx}(U_i + h_i Q_{ID_i}\right) \\ &= e\left(P_{pub}, \sum_{i=1}^{n}(U_i + h_i Q_{ID_i})\right) \end{aligned}$$

4 Security analysis

This section depicts that the required security as described in Section 2 can be achieved by the proposed protocol using the random oracle model [39]. The logic of our security proof uses an approach similar to [1, 5].

4.1 Client-to-server authentication

Theorem 1 illustrates that an adversary $\mathscr{A}$ cannot represent the client to the server assuming the computational Diffe-Hellman problem is hard.

Theorem 1 *Assuming $\mathscr{A}$ having a non-negligible advantage ε exists with probability of breaking client-to-server authentication. The computational Diffe-Hellman problem is therefore solved by a challenger $\mathscr{F}$ having a non-negligible probability. Assume that at most, q_S queries to the oracle Π_S^j of the server, q_C queries to the oracle Π_C^i of the client, and q_{H_i} queries on H_i oracle $\forall i \in \{1, 2, 3\}$ are made by $\mathscr{F}$.*

Proof Suppose that our client-to-server authentication protocol is susceptible to a non-negligible ε assault advantage from $\mathscr{A}$ within a polynomial attack duration under an adaptive chosen message and identity attacks. For a chosen target identity such as Lemma 1 [40], $\mathscr{A}$ owns a non-negligible ε advantage within a polynomial duration to attack the client-to-server authentication of our scheme using adaptive chosen message attacks. This then dictates that our protocol is resistant against chosen identity attack in the random oracle model.

To prove Theorem 1, suppose that, a random instance $P, aP, bP \in \mathbb{G}_1$ with unknown $a, b \in \mathbb{Z}_q^*$ are received by $\mathscr{F}$. With $\mathscr{F}$'s main goal being to derive abP by interacting with $\mathscr{A}$. $\mathscr{A}$'s oracle queries are answered by $\mathscr{F}$ as follows :

1. *Initialization*: The algorithm $\mathscr{F}$ generates the system parameters $\{\mathbb{G}_1, \mathbb{G}_2, q, e, P, P_{pub}, H_1, H_2, H_3, , H_4\}$ where $P_{pub} = bP$ and sends them to $\mathscr{A}$. $\mathscr{F}$ picks an identity ID_t randomly as the challenge identity in this game. To avoid collision and consistency, $\mathscr{F}$ maintains five lists L_{H_1}, L_{H_2}, L_{H_3}, L_{H_4} and L_K for queries and responses. We assume H_1 query on ID_u is made first before other queries are issued.
2. H_1 query: Whenever $\mathscr{A}$ sends H_1 query on ID_u, $\mathscr{F}$ randomly chooses $y_i \in \mathbb{Z}_q^*$ and returns $Q_{C_i} = y_i P$ to $\mathscr{A}$. However, if $ID = ID_t$ returns $Q_{C_i} = y_i(aP)$ to $\mathscr{A}$. Then, $\mathscr{F}$ updates L_{H_1} with (ID_u, y_i, Q_{C_i})
3. H_2 query: When $\mathscr{A}$ submits H_2 query on (T, U_i, k), $\mathscr{F}$ randomly chooses $e_1 \in \mathbb{Z}_q^*$ and returns it to $\mathscr{A}$. $\mathscr{F}$ updates L_{H_2} with (T, U_i, k, e_1).
4. H_3 query: Whenever $\mathscr{A}$ sends H_3 query on $(P_{pub}, T, \alpha, U_i, K)$, $\mathscr{F}$ chooses $e_2 \in_R \mathbb{Z}_q^*$ and returns it back to $\mathscr{A}$. $\mathscr{F}$ updates L_{H_3} with $(P_{pub}, T, \alpha, U_i, k, e_2)$.
5. H_4 query: Whenever $\mathscr{A}$ sends H_4 query on $(P_{pub}, T, \alpha, U_i, K, Auth)$, $\mathscr{F}$ chooses $e_3 \in_R \mathbb{Z}_q^*$ and returns it back to $\mathscr{A}$. $\mathscr{F}$ updates L_{H_4} with $(P_{pub}, T, \alpha, U_i, k, Auth, e_3)$.
6. *Extract* query: Whenever $\mathscr{A}$ sends *Extract* query on ID_u requesting for the private key, $\mathscr{F}$ checks if a tuple $(ID_u, Q_{ID_u}, D_{ID_u})$ exists in L_K. If it satisfy, D_{ID_i} is returned by $\mathscr{F}$ to $\mathscr{A}$. Else, $\mathscr{F}$ searches in L_{H_1} for an entry (ID, y_i, Q_{C_i}) and executes the following:
   (a) If $ID_u = ID_t$, the private key cannot be computed since the value of a and b are unknown. $\mathscr{F}$ updates L_k with $(ID_u, y_i(aP), \perp)$. The symbol $\perp$ denotes an unknown value.
   (b) If $ID_u \neq ID_t$, $\mathscr{F}$ estimates the private key $D_{ID_u} = y_i(bP)$ and replies it to $\mathscr{A}$. $\mathscr{F}$ updates L_K with $(ID_u, y_i P, y_i(bP))$.
7. *Send* query:
   (a) When $\mathscr{A}$ submits *Send* $(\Pi_C^i$, "start ") query and chooses n user's identities $T = \cup\{ID_i\}$ where $1 \leq i \leq n$. $\mathscr{F}$ randomly chooses $U_i \in \mathbb{G}_1$, computes $k_1 = e(r_1 P_{pub}, Q_{S_j})$, chooses an index $dx \in \{1, 2, 3, \ldots, n\}$ and $z \in \mathbb{Z}_q^*$. Then, $\mathscr{F}$ computes $h_i = H_2(T, U_i, k_1) \forall i \in \{1, 2, 3, \ldots, n\}$ except dx, chooses $h'_{dx} \in \mathbb{Z}_q^*$, and computes $U_{dx} = zP - h'_{dx}Q_{ID_{dx}} - \sum_{i \neq dx}(U_i + h_i Q_{ID_i})$. Finally, $\mathscr{F}$ sends (T, U_i) to the adversary $\mathscr{A}$.
   (b) When $\mathscr{A}$ submits *Send* $(\Pi_S^j, (T, U_i))$ query to the server. If $ID_u \neq ID_t$, $\mathscr{F}$ randomly chooses $\alpha \in \mathbb{Z}_q^*$, computes $k_2 = e(U_1, D_{S_j})$, sets $Auth = H_3(P_{pub}, T, \alpha, U_1, k_2)$ and returns $(\alpha, Auth)$ to $\mathscr{A}$. Otherwise, if $ID_u = ID_t$, $\mathscr{F}$ fails and terminates.
   (c) When $\mathscr{A}$ submits *Send* $(\Pi_C^i, (\alpha, Auth))$ query to the client, If $ID \neq ID_t$, $\mathscr{F}$ verifies if $Auth = H_3(P_{pub}, T, \alpha, U_1, k_1)$ holds. If it holds,

$\mathscr{F}$ computes $sk = H_4(P_{pub}, T, \alpha, U_1, k_1, Auth)$, adds $h_{dx} = H_2(T, U_{dx}, k_1)$ to L_{H_2}, computes $V = z(bP)$ and sends V to $\mathscr{A}$. Otherwise, $ID = ID_t$, Since $\mathscr{A}$ can't satisfy $Auth'$, $\mathscr{F}$ acts correctly.

(d) When $\mathscr{A}$ submits *Send* $(\Pi_S^j, (V))$ to the server. $\mathscr{F}$ computes $h_i = H_2(T, U_i, k_2)$ to verify $e(P_{pub}, \sum_{i=1}^{n}(U_i + h_i Q_{ID_i})) = e(P, V)$. If it holds, $\mathscr{F}$ accepts, computes $sk = H_4(P_{pub}, T, \alpha, U_1, k_2, Auth)$ and terminates. Otherwise, $\mathscr{F}$ ends.

Analysis Given that, $\delta^{q_{ex}}$ is $\mathscr{F}$'s probability of succeeding all the private key extraction queries q_{ex}, as well as $(1-\delta)^p$ being the probability that $\mathscr{A}$ forges a signature that $\mathscr{F}$ won't have all the corresponding private keys engaged in it. Where p denotes the number of participants engaged with the forged signature. Then, the sum probability is given as $\delta^{q_{ex}}(1-\delta)^p$. The value of δ is obtained by maximizing this probability $q_{ex}/q_{ex}+p$ while the maximized probability is given as

$$\left(1-\frac{p}{q_{ex}+p}\right)^{q_{ex}+p}\left(\frac{p}{q_{ex}}\right)^{p}$$

With $(1 - q_{H_1}2/2^{\lambda})$ being the probability of $\mathscr{F}$ succeeding all the sign queries q_s, which is greater than $(1 - q_s q_{H_1}/2^{\lambda-1})$, the probability for $\mathscr{F}$ to success for very large q_{ex} is given as

$$\varepsilon_{\mathscr{F}} = \varepsilon_{\mathscr{A}}\left(\frac{p}{eq_{ex}}\right)^{p}\left(1-\frac{q_s q_{H_1}}{2^{\lambda-1}}\right)$$

Violation of the client-to-server authentication by $\mathscr{A}$ means a valid forgery (V, U) has been committed on ID_u. The generic ring signature schemes' Forking Lemma [34] which states that if $\varepsilon_{\mathscr{F}} \leq 7\mathscr{F}_p^{q_H}/2^{\lambda}$ as well as $\mathscr{A}$ produces a correct faked signature within a specific time $t_{\mathscr{A}}$ as in the above relationship, then another attacker that produces two signature $V = \{U_{i=1}^{n}\{Ui\}, V\}$ and $V' = \{U_{i=1}^{n}\{Ui\}, V'\}$ with not more than $\varepsilon_{\mathscr{F}}^2/66\mathscr{F}_p^{q_H}$ probability within time $2t_{\mathscr{A}}$ is computed. For all $i \in \{1, 2, \ldots, n\}$, suppose $h_i = H_2(T, U_i, k_1)$ and $h_i' = H_2(L, U_i, k_1)$, $\forall i \in \{1, 2, , n\}$ except dx, then $h_i = h_i'$. Given that $\mathscr{A}'$ is a derivation of $\mathscr{A}$, the CDHP can be solved by computing $abP = y_{dx}^{-1}(h_{dx} - h_{dx}')1(V - V')$, such that in the list L_{H_1}, y_{dx} is found by searching for ID_u. □

4.2 Key agreement

Theorem 2 illustrates our protocol achieves a key agreement under (BDH) problem.

Theorem 2 *Suppose that the value w in the Test-query can be guessed by exists $\mathscr{A}$ with a non-negligible advantage ε. Then, the BDH Problem is solved by exist $\mathscr{F}$ with a non-negligible probability. Assume that at most q_S queries to the oracle Π_S^j of the server, q_C queries to the oracle Π_C^i of the client, and q_{H_i} queries on H_i oracle $\forall i \in \{1, 2, 3\}$ are made by $\mathscr{F}$.*

Proof The toss value in a *Test* query can be perfectly guessed by $\mathscr{A}$ with probability not lower than $1/2$. $\mathscr{A}$ can obtain the correct session key with advantage $\Pr[Ocsk] \geq \varepsilon/2$ assuming it can guess the coin with ε. Where the event whereby the correct session key obtained is denoted as Osk. Let $Test(C^i)$ and $Test(S^j)$ be connoted as the success events of the oracles Π_C^i of the client and Π_S^j of the server, separately. The event which comes after break up the client-server authentication is denoted by E^{C2S}. $\mathscr{A}$ may submit *Test* query to the client and the server, then for some i and j we get this probability

$$\Pr[Ocsk \wedge Test(\Pi_S^j) \wedge E^{C2S}] + \Pr[Ocsk \wedge Test(\Pi_S^j) \wedge \neg E^{C2S}] + \Pr[Ocsk \wedge Test(\Pi_C^i)] \geq \frac{\varepsilon}{2}$$

While E^{C2S} denotes the event of the breaking client-server authentication. Then, we have this probability for some i and j as;

$$\Pr[Ocsk \wedge Test(\Pi_C^i) \wedge \neg E^{C2S}] + \Pr[Ocsk \wedge Test(\Pi_C^i)] \geq \frac{\varepsilon}{2} - \Pr_{C2S}$$

To prove Theorem 2, We assume that $\mathscr{F}$ receives random instances $P, aP, bP, cP \in \mathbb{G}_1$ with unknown $a, b, c \in \mathbb{Z}_q^*$. The goal of $\mathscr{F}$ is to derive $e(P, P)^{abc}$ by interacting with $\mathscr{A}$. $\mathscr{A}$'s queries are responded by algorithm $\mathscr{F}$ as follows :

1. *Initialization* The algorithm $\mathscr{F}$ generates the system parameters $\{\mathbb{G}_1, \mathbb{G}_2, q, e, P, P_{pub}, H_1, H_2, H_3, H_4\}$ where $P_{pub} = bP$ and sends them to $\mathscr{A}$. $\mathscr{F}$ picks an identity ID_t randomly as the challenge identity in this game. To avoid collision and consistency, $\mathscr{F}$ maintains five lists L_{H_1}, L_{H_2}, L_{H_3}, L_{H_4} and L_K for queries and responses. We assume H_1 query on ID_u is made first before other queries are issued.
2. H_1 query: Whenever $\mathscr{A}$ sends H_1 query on ID_u, $\mathscr{F}$ randomly chooses $y_i \in \mathbb{Z}_q^*$ and returns $Q_{C_i} = y_i P$ to $\mathscr{A}$. However, if $ID = ID_t$ returns $Q_{C_i} = y_i(aP)$ to $\mathscr{A}$. Then, $\mathscr{F}$ updates L_{H_1} with (ID_u, y_i, Q_{C_i})
3. H_2 query: Whenever $\mathscr{A}$ sends H_2 query on (L, U_i, k), $\mathscr{F}$ randomly chooses $e_1 \in \mathbb{Z}_q^*$ and returns it to $\mathscr{A}$. $\mathscr{F}$ updates L_{H_2} with (L, U_i, k, e_1).

4. H_3 query: Whenever $\mathscr{A}$ sends H_3 query on $(P_{pub}, L, \alpha, U_i, k)$, $\mathscr{F}$ chooses $e_2 \in_R \mathbb{Z}_q^*$ and returns it to $\mathscr{A}$. $\mathscr{F}$ updates L_{H_3} with $(P_{pub}, L, \alpha, U_i, k, e_2)$.
5. H_4 query: Whenever $\mathscr{A}$ sends H_4 *query* on $(P_{pub}, T, \alpha, U_i, K, Auth)$, $\mathscr{F}$ chooses $e_3 \in_R \mathbb{Z}_q^*$ and returns it to $\mathscr{A}$. $\mathscr{F}$ updates L_{H_4} with $(P_{pub}, T, \alpha, U_i, k, Auth, e_3)$.
6. *Extract* query: When $\mathscr{A}$ submits this query on ID_u requesting for a private key. $\mathscr{F}$ checks if a tuple $(ID_u, Q_{ID_u}, D_{ID_u})$ exists in L_K. If it satisfy, D_{ID_i} is returned by $\mathscr{F}$ to $\mathscr{A}$. Else, $\mathscr{F}$ searches in L_{H_1} for an entry (ID, y_i, Q_{C_i}) and executes the following:
   (a) If $ID_u = ID_t$, the private key cannot be computed since the value of a and b are unknown. $\mathscr{F}$ updates L_k with $(ID_u, y_i(aP), \perp)$. The symbol $\perp$ denotes an unknown value.
   (b) If $ID_u \neq ID_t$, $\mathscr{F}$ computes the private key $D_{ID_u} = y_i(bP)$ and returns it to $\mathscr{A}$. $\mathscr{F}$ updates L_K with $(ID_u, y_i P, y_i(bP))$.
7. *Send* query:
   (a) When $\mathscr{A}$ submits *Send* $(\Pi_C^i$, "start ") query and chooses n user's identities $T = \cup\{ID_i\}$ where $1 \leq i \leq n$. $\mathscr{F}$ randomly chooses $U_i \in \mathbb{G}_1$, computes $k_1 = e(r_1 P_{pub}, Q_{S_j})$, chooses an index $dx \in \{1, 2, 3, \ldots, n\}$ and $z \in \mathbb{Z}_q^*$. Then, $\mathscr{F}$ computes $h_i = H_2(T, U_i, k_1) \forall i \in \{1, 2, 3, \ldots, n\}$ except dx, chooses $h'_{dx} \in \mathbb{Z}_q^*$, and computes $U_{dx} = zP - h'_{dx} Q_{ID_{dx}} - \sum_{i \neq dx}(U_i + h_i Q_{ID_i})$. Finally, $\mathscr{F}$ sends (T, U_i) to the adversary $\mathscr{A}$.
   (b) When $\mathscr{A}$ submits *Send* $(\Pi_S^j, (T, U_i))$ query to the server. If $ID_u \neq ID_t$, $\mathscr{F}$ randomly chooses $\alpha \in \mathbb{Z}_q^*$, computes $k_2 = e(U_1, D_{S_j})$, sets $Auth = H_3(P_{pub}, T, \alpha, U_1, k_2)$ and returns $(\alpha, Auth)$ to $\mathscr{A}$. Otherwise, if $ID_u = ID_t$, $\mathscr{F}$ fails and terminates.
   (c) When $\mathscr{A}$ submits *Send* $(\Pi_C^i, (\alpha, Auth))$ query to the client, If $ID \neq ID_t$, $\mathscr{F}$ verifies if $Auth = H_3(P_{pub}, T, \alpha, U_1, k_1)$ holds. If it holds, $\mathscr{F}$ computes $sk = H_4(P_{pub}, T, \alpha, U_1, k_1, Auth)$, adds $h_{dx} = H_2(T, U_{dx}, k_1)$ to L_{H_2}, computes $V = z(bP)$ and sends V to $\mathscr{A}$. Otherwise, $ID = ID_t$, Since $\mathscr{A}$ can't satisfy $Auth'$, $\mathscr{F}$ acts correctly.
   (d) When $\mathscr{A}$ submits *Send* $(\Pi_S^j, (V))$ to the server. $\mathscr{F}$ computes $h_i = H_2(T, U_i, k_2)$ to verify $e(P_{pub}, \sum_{i=1}^n (U_i + h_i Q_{ID_i})) = e(P, V)$. If it holds, $\mathscr{F}$ accepts, computes $sk = H_4(P_{pub}, T, \alpha, U_1, k_2, Auth)$ and terminates. Otherwise, $\mathscr{F}$ ends.
8. *Corrupt* query: Whenever *Corrupt* query on ID_u is submitted by $\mathscr{A}$, D_{ID_u} is returned by $\mathscr{F}$.
9. *Reveal* query: The session key sk is returned by $\mathscr{F}$, whenever *Reveal* query is submitted by $\mathscr{A}$.
10. *Test* query: Whenever $\mathscr{A}$ send *Test* query, if the query is not asked in the session, $\mathscr{F}$ sets $U = cP$ an instance of BDHP and aborts. Else, a fair coin w is flipped by $\mathscr{F}$. If $w = 1$, a session key is returned to $\mathscr{A}$; else, a random string is returned to $\mathscr{A}$.

However, with $\mathscr{F}$ is noticed that $\exists j, Ocsk \wedge Test(\Pi_S^j) \wedge \neg E^{C2S}$ is equal to $\exists i, Ocsk \wedge Test(\Pi_C^j)$ such that $\Pr[Ocsk \wedge Test(\Pi_C^i)] \geq \frac{\varepsilon}{2} - \Pr_{C2S}$. We have the following probability by the simulation of the queries to the client

$$\Pr\left[sk = H_4(P_{pub}, L, \alpha, U_i, k_1, Auth)|_{U_1, k_1 \leftarrow \mathbb{G}_1}^{\alpha, Auth \in \mathbb{Z}_q^*}\right] \geq \frac{\varepsilon}{2} - Pr_{c2s}$$

We know that the $\Pr_{C2S}$ is negligible by Theorem 1. If ε is non-negligible, that means the $\varepsilon/2 - \Pr_{C2S}$ is a non-negligible. We assume that the adversary can calculate k_1 and k_2 with a non-negligible probability. $\mathscr{A}$ needs to calculate k_1 and k_2 to know $(Q_u = aP, P_{pub} = bP, U_1 = cP)$. To get $e(P, P)^{abc}$, which solves the difficult BDH problem, the adversary needs to computes K_1 and K_2

$$\begin{aligned} k_1 &= e(r_1 P_{pub}, Q_{S_j}) = e(r_1(bP), ap) = e(r_1 P, abP) \\ &= e(U_1, abP) = e(cP, abP) = e(P, P)^{abc} \end{aligned}$$

$$\begin{aligned} k_2 &= e(U_1, D_{S_j}) = e(cP, xQ_{S_j}) = e(cP, xaP) \\ &= e(acP, xP) = e(cP, P_{pub}) = e(acP, bP) \\ &= e(P, P)^{abc} \end{aligned}$$

According to the assumptions of the difficult problem of BDH, the probability that $\mathscr{A}$ wins the game is negligible. Hence, Our protocol provides a secure key agreement. □

4.3 Sever-to-client authentication

Theorem 3 determines that an adversary $\mathscr{A}$ cannot represent the server to communicate with the client under the BDHP.

Theorem 3 *Assuming $\mathscr{A}$ having non-negligible advantage ε exists with probability of breaking server-to-client authentication. The BDHP problem is therefore solved by a challenger $\mathscr{F}$ having a non-negligible probability. Assume that at most, q_S queries to the oracle Π_S^j of the server, q_C queries to the oracle Π_C^i of the client, and q_{H_i} queries on H_i oracle $\forall i \in \{1, 2, 3\}$ are made by $\mathscr{F}$.*

Proof Let the algorithm be displayed as presented in the proof of Theorem 2. As such, it is completely indistinguishable from our protocol except the occurrence of the event E^{C2S}. We use E^{S2C} as the event which follows

breaking up the server-to-client authentication. When the oracle accepts with a non-legitimate entity, event E^{S2C} occurs. Basically, this phenomenon happens aftermath of sending $(L, U = cP)$ and the receiving of $(\alpha, Auth)$, then, the client accepts $(\alpha, Auth)$ which is not generated by the server. In this circumstance, one of the following three conditions are occurred:

1. The adversary $\mathscr{A}$ guessed the value $Auth$ with probability less than $q_C/2^k$.
2. The value U_i occurred in another session with a probability $q_C/q \times (q_C - 1)$ less than q_C^2/q.
3. $\mathscr{A}$ asked $H_1(ID_t)$ with a probability $\Pr[(P_{pub}, L, \alpha, U_i, k_2)|P_{pub} \in_R G_1, k_2 = e(U_1, D_{S_j})]$. Then, we have

$$\Pr[E^{S2C}|\neg E^{C2S}] \leq \Pr[P_{pub}, L, \alpha, U_i, k_2]|P_{pub} \in_R G_1, k_2 = e(U_1, D_{S_j})] + \frac{q_C}{2^k} + \frac{q_C^2}{q}$$

To prove Theorem 3, $\mathscr{A}$'s queries are responded by an algorithm $\mathscr{F}$. Assuming that the instances $P, aP, bP, cP \in \mathbb{G}_1$ with unknown $a, b, c \in \mathbb{Z}_q^*$ are received by $\mathscr{F}$ randomly. The goal of $\mathscr{F}$ is to derive $e(P, P)^{abc}$ by interacting with $\mathscr{A}$. $\mathscr{F}$ picks an identity ID_t randomly as the challenged identity in this game. In this case, $\mathscr{A}$ receives $(Q_u = aP, P_{pub} = bP, U_1 = cP)$. $\mathscr{A}$ can compute $e(P, P)^{abc}$ with a non-negligible probability. To get $e(P, P)^{abc}$, which solves the BDH problem, the adversary computes

$$\begin{aligned} k_2 &= e(U_1, D_{S_j}) = e(cP, xQ_{S_j}) = e(cP, xaP) \\ &= e(acP, xP) = e(cP, P_{pub}) \\ &= e(acP, bP) = e(P, P)^{abc} \end{aligned}$$

However, $\mathscr{F}$ can use $\mathscr{A}$ to process $e(P, P)^{abc}$. $\mathscr{F}$ can solve the BDH problem with $\varepsilon' \geq \varepsilon - q_C/2^k - q_C^2/q$. Hence, the scheme provides server-to-client authentication. □

4.4 Unconditional signer ambiguity property

Theorem 4 *Our protocol has the unconditional signer ambiguity property.*

Proof To prove that the ring signature that we have used in our protocol has the unconditional signer ambiguity property. Suppose that $\cup_{i\neg dx}^{n}\{U_i\}$ and r_{dx} are randomly generated, However $\cup_{i=1}^{n}\{U_i\}$ are distributed uniformly. Considering if $V = (h_{dx} + r_{dx})D_{ID_{dx}}$, this gives details about the true signer. Then, focus is shifted on the value of $h_{dx}D_{ID_{dx}} = r_{dx}$ as h_{dx} is publicly estimable. Visibly, $r_{dx}D_{ID_{dx}}$ is associated to h_{dx}. Then, it is possible that we can estimate the $r_{dx}Q_{ID_{dx}}$ by $h_{dx} + \sum_{i\neq dx}^{n}(U_i + h_iQ_{ID_i})$. The relation between $r_{dx}D_{ID_{dx}}$ and $r_{dx}Q_{ID_{dx}}$ can be found by using the bilinearity property in this equality $e(r_{ID_{dx}}Q_{ID_{dx}}, P) = e(r_{dx}D_{ID_{dx}}, P_{pub})$. It could be enticing to check if ID_j is the true signer by examining with the condition that this equation is satisfied or not.

$$e\left(U_j + \sum_{i\neq dx}^{n}(U_i + h_iQ_{ID_i}), P_{pub}\right) = e(V, P)/e(h_jQ_{ID_j}, P_{pub})$$

Hence, the aforementioned equality might not hold when $j = dx$, as well as for all $j \in \{1, 2, \ldots, n\}$ except dx. Certainly, the prior stated equation is but similar as the equality to be looked out for in authentication phase. The following equations show that our protocol provides unconditional signer ambiguity property as follows:

$$\begin{aligned} &e\left(U_j + \sum_{i\neq dx}^{n}(U_i + h_iQ_{ID_i}), P_{pub}\right) \\ &= \left(\sum_{i\neq dx}^{n}U_j + U_{dx} + \sum_{i\neq j}^{n}(h_iQ_{ID_i}), P_{pub}\right) \\ &= e\left(\sum_{i\neq dx}^{n}U_j + r_{dx}Q_{ID_{dx}} - \sum_{i\neq dx}^{n}(U_i + h_iQ_{ID_i}) + \sum_{i\neq j}^{n}(h_iQ_{ID_i}), P_{pub}\right) \\ &= e\left(r_{dx}Q_{ID_{dx}} - \sum_{i\neq dx}^{n}(h_iQ_{ID_i}) + \sum_{i\neq j}^{n}(h_iQ_{ID_i}), P_{pub}\right) \\ &= e(r_{dx}Q_{ID_{dx}} + h_{dx}Q_{ID_{dx}} - h_jQ_{ID_j}, xP) \\ &= e(r_{dx}Q_{ID_{dx}} + h_{dx}D_{ID_{dx}} - h_jS_{ID_j}, P) \\ &= e(V - h_jS_{ID_j}, P) \\ &= e(V, P)/e(h_jS_{ID_j}, P) \\ &= e(V, P)/e(h_jQ_{ID_j}, P_{pub}) \end{aligned}$$

Table 2 Comparisons based computation and communication cost

Schemes	Computational Cost		Communication Cost
	Client	Server	
[1]	$4T_{Mu} + T_{Ad} + 3T_H$	$2T_e + 2T_{Mu} + T_{Ad} + 3T_H$	$\|ID\| + 2\|\mathbb{Z}_q^*\| + 2\|\mathbb{G}_1\|$
[41]	$2T_{Mu} + 3T_H + T_{In}$	$T_e + 5T_{Mu} + 2T_{Ad} + 5T_H$	$2\|\mathbb{Z}_q^*\| + 3\|\mathbb{G}_1\|$
[21]	$4T_{Mu} + 3T_{Ad} + 4T_H$	$4T_{Mu} + 3T_{Ad} + 4T_H$	$\|ID\| + 2\|\mathbb{Z}_q^*\| + 4\|\mathbb{G}_1\|$
Ours	$T_e + nT_{Mu} + nT_{Ad} + 4T_H$	$3T_e + 3T_H$	$\|ID\| + 2\|\mathbb{Z}_q^*\| + n\|\mathbb{G}_1\| + \|\mathbb{G}_1\|$

Table 3 Security comparisons

	[1]	[41]	[21]	Our scheme
Mutual authentication	Y	Y	Y	Y
Key agreement	Y	Y	Y	Y
Unconditional anonymity	N	N	N	Y
Un traceability	N	N	N	Y
Perfect secrecy	Y	Y	Y	Y

For each defined identity ID_j and determined n user's identities T, this dictates that the distributions of $\{\cup_{i=1}^{n}\{U_i\}, V\}$ are not only absolute but also totally consistently distributed regardless of who the true signer is. □

5 Performance analysis

This section evaluates the performance of our protocol in terms of the computation cost , the communication cost as well as the security properties. The proposed protocol is compared with Wu et al. [1], Tsai et al. [41], and Tseng et al. [21]. Some notations are assumed to evaluate the computational cost as follows:
T_e: The time of a bilinear map operation $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.
T_{Mu}: The time of a scalar multiplication operation of $\mathbb{G}_1$.
T_{In}: The time of performing a modular inversion operation.
T_{Ad}: The time of an addition operation of $\mathbb{G}_1$.
T_H: The performing time of a one-way hash function.

Table 2 gives the theoretical analysis of the computation and communication costs. Table 3 shows that our protocol gives better security properties than the other schemes by providing unconditional anonymity.

We have implemented four schemes using the java pairing-based cryptography library (JPBC) [42] for both the client and servers. The client is simulated by using Android Studio bundle version 2.2.0.0 on Honor-phone with EMUI 4.0.1 CPU Octa-core 1.5 GHz and RAM of 2.0 GB. Regarding the servers, simulation carried out by using java on computers with an Intel Core-i 3-3110 CPU dual core 2.40 and GHz 2.40 GHz and with 4 GB RAM. This study employed Type A pairings constructed from the curve $y^2 = x^3 + x$ over the field $\mathbb{F}_p$ for some prime $p = 3 \bmod 4$. The experiment involves 80, 112 and 128 bits key sizes of AES security levels [43] as shown in Table 4. To simulate our protocol, we have prepared three servers (registration server, server 1 and server 2) connected to form the multi-server and one mobile phone prepared to simulate the client (i.e., doctor, nurse or family member) as shown in Fig. 6. Here, we assume the WBANs data are transmitted and stored securely to/in the medical servers. Thus, the experiment has focused on the clients and the servers as part of the IoT environment. For each of the four schemes, the experiment was carried out 100 times and the corresponding computation time was calculated. To ascertain the computation cost for the cases of client and servers, we consider the average and the results presented in Figs. 3 and 4.

Table 4 Security level of our experiment

Security Level	Size of p	Size of q
80-bit	1024	160
112-bit	2048	224
128-bit	3072	256

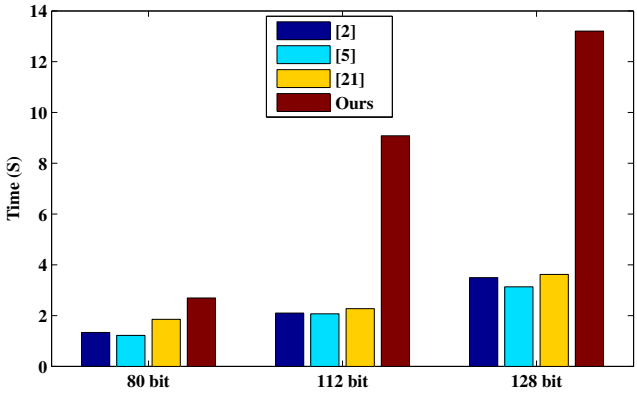

Fig. 3 The client computational time

The results show that the computational costs on both sides when our proposed protocol is adopted, exceeds the computational cost for the use of the other protocols. The higher computational cost demonstrated by the empirical results can be explained by use of the ring signature in our proposed protocol. This notwithstanding, the advantage of guaranteed privacy demonstrated by the proposed protocol makes it a preferred choice for secured privacy in multi-server IoT environment.

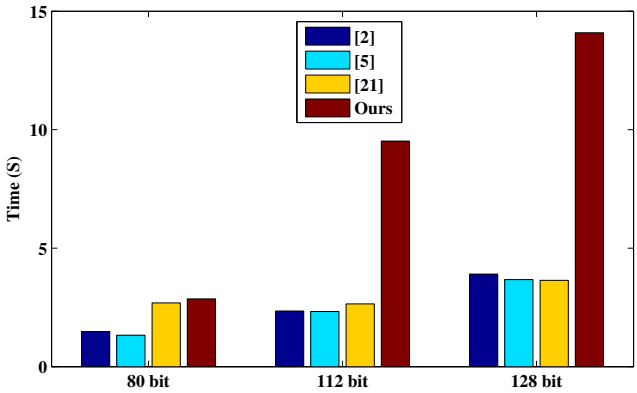

Fig. 4 The server computational time

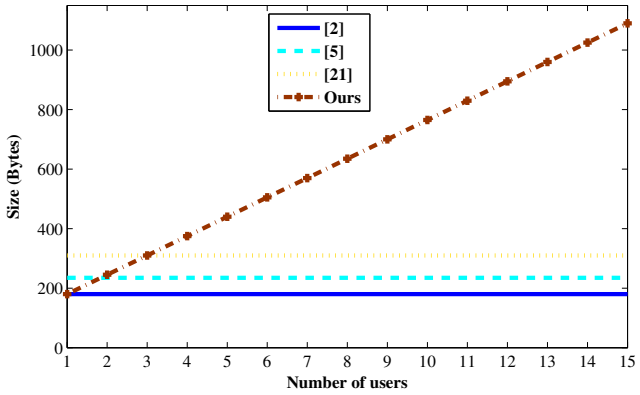

Fig. 5 Communication cost

Figure 5 shows the communication cost, by assuming that $|m| = \frac{160}{8}$ bytes and $|ID| = \frac{80}{8}$ bytes. Using an elliptic curve with $q = \frac{160}{8}$ bytes, the size of $\mathbb{G}_1$ is 1024. Using the standard compression method [44], the size of $\mathbb{G}_1$ can be reduced to 65 bytes. According to Table 2, the communication cost for each scheme is as follows.

1. Communication cost in [1] is $|ID| + 2|\mathbb{Z}_q^*| + 2|\mathbb{G}_1| = 10 + 2 \times 20 + 2 \times 65 = 180$ bytes.
2. Communication cost in [41] is $2|\mathbb{Z}_q^*| + 3|\mathbb{G}_1| = 2 \times 20 + 3 \times 65 = 235$ bytes.
3. Communication cost in [21] is $|ID| + 2|\mathbb{Z}_q^*| + 4|\mathbb{G}_1| = 10 + 2 \times 20 + 4 \times 65 = 310$ bytes.
4. Communication cost in our scheme is $|ID| + 2|\mathbb{Z}_q^*| + d|\mathbb{G}_1| + |\mathbb{G}_1| = 115 + n \times 65$ bytes.

By comparing the communication costs, our protocol has the highest cost. Indeed, The cost increases as the number of users increases.

6 Application scenario

Figure 6 shows an electronic medical application scenario in which the WBAN collects a biomedical data such as heart rate, blood pressure, etc. in real-time. A remote hospital server receives these data from a personal digital assistant (PDA) device connected to the WBAN. By using these data, the clients (doctors, nurses, scientists and family members), can access a patient's health status and related information from the medical system servers (registration server, hospital server, Health insurance server and E-government server). The medical experts can, in turn, offer clinical diagnostics according to the patient's status. When treatment is administered, relatives of patients can get the health status update. The opportunity for easy access to sensitive information made available by IoT operating in multi-server environment requires efficient scheme that preserves the privacy of system users.

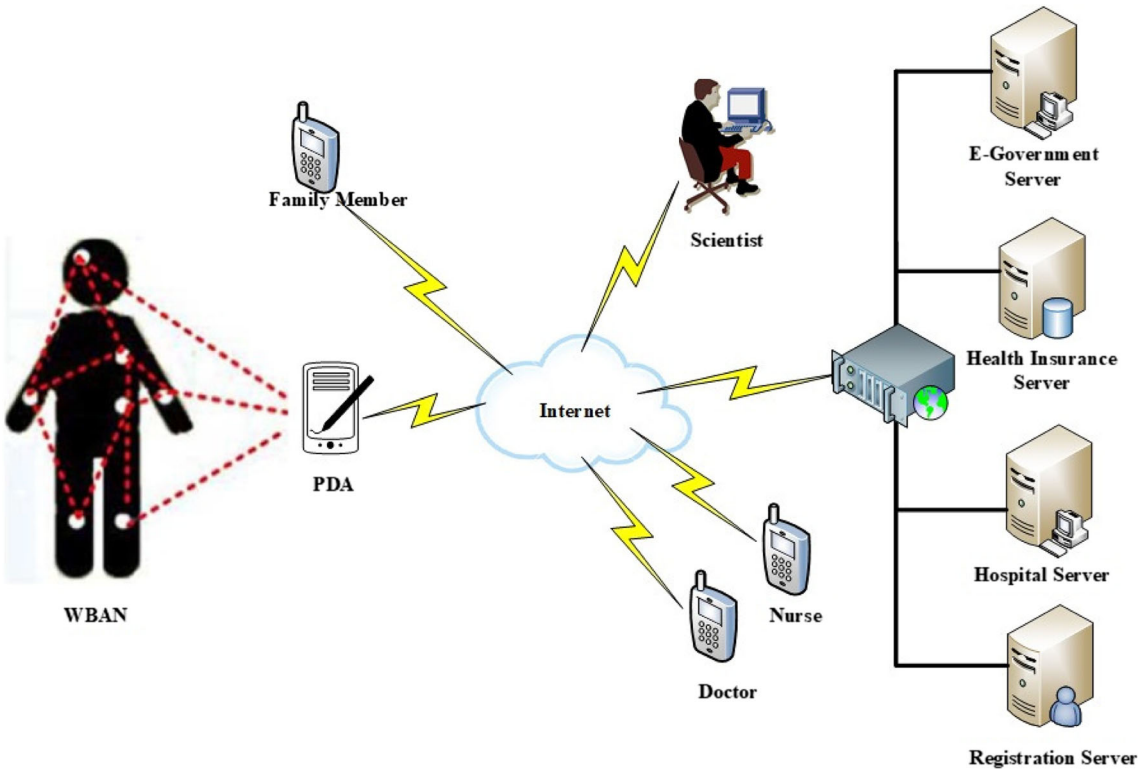

Fig. 6 An electronic medical application scenario

The transmitted data between the WBANs and the medical servers are very sensitive because they are the basis of clinical diagnostics. Situated in the context of unconditional anonymity upon which our proposal is grounded, the clients can authenticate themselves in the severs without revealing their identities to the severs unlike the other schemes. In addition, the transmitted data between clients and the servers is sensitive in that an adversary getting access to this data can be altered to the detriment of the patient in question. From the perspective of the stakeholders, privacy-preserving is one of the essential issues in the medical systems. Here, the clients interact with the multi-medical server environment and our protocol is necessary and well suited to protect their privacy for the IoT that operate in multi-severs environment as described.

For example to use our protocol in the IoT environment that described in Fig. 6, the clients are doctors, nurses, scientists and family members while the servers are registration server, hospital server, health insurance server and e-government server. The *Setup* phase is executed by the registration server to generate the public parameters and send to the clients and servers. Therefore, registration by the client and severs is required at the registration server. Then, the *Key extract* phase is executed by the clients and servers to get their private keys D_u and public keys Q_u from the registration server. Based on the proposed protocol, they evaluate their private keys to ensure that they are dealing with the appropriate registration server. The *user authenticated key agreement* phase is executed to get access to all the servers. Finally, the session key sk is computed by the clients and the servers to use in future communication.

7 Conclusion

Conclusively, this paper proposes a new anonymity ID-based user authenticated key agreement protocol for multi-server environment while its security proved in the random oracle model. Compared with the existing protocols, our model provides unconditional anonymity. The proposed protocol utilizes ring signature for the purpose of anonymously authenticating themselves in the severs without revealing their identities to the severs unlike the other schemes. Therefore, our scheme provides more secured privacy for users in IoT multi-server environment.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Grant No 61272525), the Fundamental Research Funds for the Central Universities (Grant No. ZYGX2016J081) and the Laboratory for Internet of Things and Mobile Internet Technology of Jiangsu Province (Grant No. JSWLW-2017-006).

References

1. Wu TY, Tseng YM (2010) An efficient user authentication and key exchange protocol for mobile client–server environment. Comput Netw 54(9):1520–1530
2. Debiao H, Jianhua C, Jin H (2012) An ID-based client authentication with key agreement protocol for mobile client–server environment on ECC with provable security. Information Fusion 13(3):223–230
3. He D (2012) An efficient remote user authentication and key agreement protocol for mobile client–server environment from pairings. Ad Hoc Netw 10(6):1009–1016
4. Shen H, Gao C, He D, Wu L (2015) New biometrics-based authentication scheme for multi-server environment in critical systems. J Ambient Intell Humaniz Comput 6(6):825–834
5. Chow SSM, Yiu SM, Hui LCK (2005) Efficient identity based ring signature. Springer, Berlin, pp 499–512. https://doi.org/10.1007/11496137_34
6. Pleva P (2012) A revised classification of anonymity. arXiv:1211.5613
7. Shamir A (1984) Identity-based cryptosystems and signature schemes. In: Workshop on the theory and application of cryptographic techniques. Springer, pp 47–53
8. Li LH, Lin LC, Hwang MS (2001) A remote password authentication scheme for multiserver architecture using neural networks. IEEE Trans Neural Netw 12(6):1498–1504. https://doi.org/10.1109/72.963786
9. Juang WS (2004) Efficient multi-server password authenticated key agreement using smart cards. IEEE Trans Consum Electron 50(1):251–255. https://doi.org/10.1109/TCE.2004.1277870
10. Chang CC, Lee JS (2004) An efficient and secure multi-server password authentication scheme using smart cards. In: 2004 International conference on cyberworlds, pp 417–422. https://doi.org/10.1109/CW.2004.17
11. Liao YP, Wang SS (2009) A secure dynamic id based remote user authentication scheme for multi-server environment. Computer Standards & Interfaces 31(1):24–29
12. Hsiang HC, Shih WK (2009) Improvement of the secure dynamic id based remote user authentication scheme for multi-server environment. Computer Standards & Interfaces 31(6):1118–1123
13. Sood SK, Sarje AK, Singh K (2011) A secure dynamic identity based authentication protocol for multi-server architecture. J Netw Comput Appl 34(2):609–618
14. Li X, Xiong Y, Ma J, Wang W (2012) An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. J Netw Comput Appl 35(2):763–769
15. Han W (2012) Weaknesses of a dynamic identity based authentication protocol for multi-server architecture. arXiv:1201.0883
16. Yoon EJ, Yoo KY (2013) Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. J Supercomput 63(1):235–255. https://doi.org/10.1007/s11227-010-0512-1
17. Khan MK, He D (2012) A new dynamic identity-based authentication protocol for multi-server environment using elliptic curve cryptography. Security and Communication Networks 5(11):1260–1266. https://doi.org/10.1002/sec.573
18. Han W, Zhu Z (2014) An id-based mutual authentication with key agreement protocol for multiserver environment on elliptic curve cryptosystem. Int J Commun Syst 27(8):1173–1185. https://doi.org/10.1002/dac.2405
19. He D, Wang D (2015) Robust biometrics-based authentication scheme for multiserver environment. IEEE Syst J 9(3):816–823. https://doi.org/10.1109/JSYST.2014.2301517

20. Shen H, Gao C, He D, Wu L (2015) New biometrics-based authentication scheme for multi-server environment in critical systems. J Ambient Intell Humaniz Comput 6(6):825–834. https://doi.org/10.1007/s12652-015-0305-8
21. Tseng YM, Huang SS, You ML (2017) Strongly secure ID-based authenticated key agreement protocol for mobile multi-server environments. Int J Commun Syst 30(11):e3251–n/a. https://doi.org/10.1002/dac.3251. E3251 IJCS-16-0586.R1
22. Jiang P, Wen Q, Li W, Jin Z, Zhang H (2015) An anonymous and efficient remote biometrics user authentication scheme in a multi server environment. Front Comp Sci 9(1):142–156. https://doi.org/10.1007/s11704-014-3125-7
23. Lin H, Wen F, Du C (2015) An improved anonymous multi-server authenticated key agreement scheme using smart cards and biometrics. Wirel Pers Commun 84(4):2351–2362. https://doi.org/10.1007/s11277-015-2708-4
24. Liao YP, Hsiao CM (2013) A novel multi-server remote user authentication scheme using self-certified public keys for mobile clients. Futur Gener Comput Syst 29(3):886–900
25. He D, Zeadally S, Kumar N, Wu W (2016) Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures. IEEE Trans Inf Forensics Secur 11(9):2052–2064. https://doi.org/10.1109/TIFS.2016.2573746
26. Zhu H (2015) A provable one-way authentication key agreement scheme with user anonymity for multi-server environment. KSII Trans Internet Inf Syst (TIIS) 9(2):811–829
27. Jangirala S, Mukhopadhyay S, Das AK (2017) A multi-server environment with secure and efficient remote user authentication scheme based on dynamic id using smart cards. Wirel Pers Commun 95(3):2735–2767. https://doi.org/10.1007/s11277-017-3956-2
28. Tsai JL, Lo NW (2015) A chaotic map-based anonymous multi-server authenticated key agreement protocol using smart card. Int J Commun Syst 28(13):1955–1963. https://doi.org/10.1002/dac.2829. IJCS-13-0727.R2
29. Irshad A, Sher M, Chaudhary SA, Naqvi H, Farash MS (2016) An efficient and anonymous multi-server authenticated key agreement based on chaotic map without engaging registration centre. J Supercomput 72(4):1623–1644. https://doi.org/10.1007/s11227-016-1688-9
30. Rivest RL, Shamir A, Tauman Y (2001) How to leak a secret. In: International conference on the theory and application of cryptology and information security. Springer, pp 552–565
31. Zhang F, Kim K (2002) ID-based blind signature and ring signature from pairings. In: International conference on the theory and application of cryptology and information security. Springer, pp 533–547
32. Lin CY, Wu TC (2004) An identity-based ring signature scheme from bilinear pairings. In: 18th international conference on advanced information networking and applications, 2004. AINA 2004, vol 2. IEEE, pp 182–185
33. Awasthi AK, Lal S (2005) ID-based ring signature and proxy ring signature schemes from bilinear pairings. arXiv:cs/0504097
34. Herranz J, Sáez G (2004) New identity-based ring signature schemes. In: ICICS, vol 4. Springer, pp 27–39
35. Chow SSM, Hui LCK, Yiu SM (2005) Identity based threshold ring signature. In: Park CS, Chee S (eds) Information security and cryptology – ICISC 2004. Springer, Berlin, pp 218–232
36. Boneh D, Franklin M (2001) Identity-based encryption from the weil pairing. In: Advances in cryptology – CRYPTO 2001. Springer, pp 213–229
37. Boneh D, Lynn B, Shacham H (2004) Short signatures from the weil pairing. J Cryptol 17(4):297–319. https://doi.org/10.1007/s00145-004-0314-9
38. Sui A, Chow SSM, Hui LCK, Yiu SM, Chow KP, Tsang WW, Chong CF, Pun KH, Chan HW (2005) Separable and anonymous identity-based key issuing. In: 11Th international conference on parallel and distributed systems (ICPADS'05), vol 2. pp 275–279. https://doi.org/10.1109/ICPADS.2005.263
39. Bellare M, Rogaway P (1993) Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM conference on computer and communications security, CCS '93. ACM, New York, pp 62–73. https://doi.org/10.1145/168588.168596
40. Choon JC, Hee Cheon J (2002) An identity-based signature from gap Diffie-Hellman groups. Springer, Berlin, pp 18–30. https://doi.org/10.1007/3-540-36288-6_2
41. Tsai JL, Lo NW (2015) Provably secure and efficient anonymous id-based authentication protocol for mobile devices using bilinear pairings. Wirel Pers Commun 83(2):1273–1286. https://doi.org/10.1007/s11277-015-2449-4
42. Caro AD, Iovino V (2011) JPBC: java pairing based cryptography. In: 2011 IEEE symposium on computers and communications (ISCC), pp 850–855. https://doi.org/10.1109/ISCC.2011.5983948
43. Daemen J, Rijmen V (2013) The design of Rijndael: AES-the advanced encryption standard. Springer Science & Business Media
44. Shim KA, Lee YR, Park CM (2013) EIBAS: an efficient identity-based broadcast authentication scheme in wireless sensor networks. Ad Hoc Netw 11(1):182–189