Wireless Pers Commun
DOI 10.1007/s11277-014-2264-3

# A Pairing Free Anonymous Certificateless Group Key Agreement Protocol for Dynamic Group

**Abhimanyu Kumar · Sachin Tripathi**



**Abstract** Group key agreement protocol is the primary requirement of several groupware applications like secure conferences; pay-per view, etc. which requires secure and authentic conversations among a group of participants via public networks. Protocols based on the certificateless public key cryptography (CL-PKC) are in demand because it overcomes the complex certificate management of traditional public key cryptography, as well as the key escrow problem of identity-based cryptography. Several group applications often need users anonymity also, along with their security features. However in current literature only few group key agreement protocols are available which supports user's anonymity. Further almost all GKA protocols based on CL-PKC are employs bilinear pairing in their operations. The expensive computation of pairing motivates the researchers to propose pairing free protocols based on the CL-PKC. The present paper proposes a pairing free certificateless group key agreement protocol that meets the efficiency, authenticity, and strong security with complete anonymity. The formal security validation of proposed protocol has been done by using automated validation of internet security protocols and applications tool which shows that it is unforgeable against the various attacks. The proposed protocol has the comparable performance than other existing protocols in terms of computation and communication overheads.

**Keywords** Group key agreement · Certificateless public key cryptography · ECC · Anonymity · Bilinear pairing · AVISPA

## 1 Introduction

A group key agreement scheme allows a number of members to participate in the establishment of a secret common key usually called group key with the significant contributions of each member. Group key provides confidential and authentic group conversations via open

A. Kumar (✉) · S. Tripathi
Department of Computer Science and Engineering, Indian School of Mines,
Dhanbad 826004, Jharkhand, India
e-mail: abhimanyu.arjun@hotmail.com

Published online: 22 January 2015

public networks. Several group oriented applications like teleconferences, distance learning, pay per view, collaborative workspaces, etc. often need such a key for its cryptographic operations. A number of group key agreement protocols have been proposed based on different cryptographic concepts while only few of the existing protocols, supports the privacy of the participants. Specially in most of the ID-based protocols, identity of group users are exposed, so that adversary can easily learn who belongs to the specific group while member's privacy is also one of the important issue in some of the group applications like evoting, unbiased conversations, etc.

Recently security protocols based on the certificateless public key cryptography (CL-PKC) becomes more demanded. The reason behind the popularity of CL-PKC is that it simplifies the heavy certificate management burden in the PKI-based protocols and resolves the key escrow problem in ID-based cryptosystem. The certificateless cryptosystem has been proposed by Al-Riyami and Paterson in 2003 [1] and since then many group key agreement protocols based on CL-PKC have been proposed [2–5]. However these protocols uses bilinear pairings to achieve required security goals in their operations. The bilinear pairing is a mathematical tool which maps two elements in an additive group (usually elliptic curve group) to an element of an multiplicative group of same order (usually elements in related finite field) and it is widely used in building of ID-based as well as certificateless key agreement protocols [2]. Bilinear pairing is always defined over a super singular elliptic curve group with large element size and thus it is many times more expensive operation than the scalar point multiplications in ECC. Therefore a paring free protocol based on CL-PKC is more appealing in practice. As per our literature survey none of the certificateless authenticated group key agreement (CL-AGKA) protocol provides users anonymity without pairing [2–6]. The present paper proposes a complete anonymous CL-AGKA protocol without pairing. The proposed protocol secure under the various security attributes and its performance analysis shows that it has comparable operational cost against other existing protocols. Although the security analysis of the proposed protocol has been done based on all security attributes discussed in [7], additionally the formal security validation of same is also tested using AVISPA tool which shows that it is safe against various attacks. The present paper can be considered as an extension of our recently published paper [8] which is the first pairing free AGKA under certificateless cryptosystem. However [8] is an static protocol and it is not considered the privacy of the members (user's anonymity) into account. The present technique may create an attraction for low power wireless devices such as mobile phones because applications using pairings can be hard to implement on these.

The rest of this paper is organized as follows. In Sect. 2 some related works on group key agreement protocol are discussed. The preliminaries related to proposed work are addressed in Sect. 3. Section 4 demonstrates the proposed protocol, while Sect. 5 provides its security analysis. The security test of the proposed protocol using AVISPA is illustrated in Sect. 6. Section 7 and 8 gives the security and performance comparison respectively of the proposed protocol with other existing protocols followed by a conclusion section.

## 2 Related Work

Traditional group key generation protocols such as CCEGK [9], TGDH [10], STR [11], etc. are based on the traditional public key cryptography and hence require public key infrastructure (PKI) to issue and manage the certificates for mapping the identity of an entity to their current public key. Since in group key generation algorithm authentication of the participant is an essential issues so the certificate management creates an overhead. In order to overcome

this kind of problem in 1984, Shamir [12] proposed the idea of ID based cryptosystem where the identity of a user functioned as his public key. The first ID-based authenticated group key agreement (ID-AGKA) protocol was proposed by Reddy et al. [13]. It utilises a binary tree structure and requires $log_2^n$ rounds for $n$ numbers of users. It achieves authentication with ID-based cryptosystem and thus avoids certificate managements. Since then, many ID-based group key exchange protocols [7,14–16] have been proposed and each have their own significance.

Protocols based on ID-based cryptosystem removes the certificates overheads undoubtedly, but has another drawback called key escrow problem. Although private key of all participants are generated by key generation centre (KGC) (also called Private Key Generator in ID-based cryptosystem) and distributed via secure channel, the entire system is vulnerable in case of compromised KGC. To overcome this drawback in 2003 Al-Riyami and Kenneth G. Paterson proposes the idea of certificateless cryptosystem [1]. Certificateless cryptosystem is an extension of ID-based cryptosystem in which private key generation process is splited between the KGC and the participant it self. Several certificateless authenticated group key agreement (CL-AGKA) protocols [2–6] are available in literature, and each having their own advantages and limitations e.g. Heo et al. [4] proposed a CL-AGK protocol using binary tree structure for dynamic group. It provides efficient communication and computation complexity, but does not provide perfect forward security. Teng and Chuankun Wu [2] propose a provable authenticated CL-GKA with constant rounds. [2] also present a security model for the certificateless group key agreement protocols. However one common requirement in all most all CL-AGKA protocols is bilinear pairing computations in their procedures. Pairing is one of the complex mathematical operation, always defined over a super singular elliptic curve group with large element size. So it is many times more expensive than the point multiplication operation over elliptic curve group. The burden of pairing computation motivates the researchers to design pairing free CL-AGKA protocols. Some of the two party pairing free CL-AKA protocols are recently proposed and available in current literature [17–21] but these are not suitable for multi-party group. In order to provides privacy among the group members, few anonymous AKGA protocols like [16,22] have been proposed. But Park et al. [23] shows that Wan et al.'s [16] protocol is insecure against colluding attack, and its joining/leaving technique do not guarantee forward and backward secrecy. [23] also proposes a new ID-AGKA protocol with anonymity. Gang Yao and Dengguo Feng [22] also proposes a complete anonymous GKA protocol using ID-based cryptosystem. However these anonymous protocols are based on ID-based cryptosystem and still have complex computational overheads due to involvement of bilinear pairing in their operations. More recently Wang et al. [24] proposes a location based group key agreement scheme specially for vehicular Ad Hoc network, but it still needs PKI and and requires $log_2^n$ rounds to agreed on a common key where $n$ is the number participants.

## 3 Preliminaries

This section briefly describes some predefined concepts used in proposed protocol and make familiar with some notations involved in entire paper.

### 3.1 Background of Elliptic Curve Group

Let the symbol $E/F_p$ denote an elliptic curve $E$ over a prime finite field $F_p$, defined by an equation

$$Y^2 \ mod \ p = (x^3 + ax + b) mod \ p. \tag{1}$$

where $a, b \in F_p$ and with the discriminant

$$\Delta = (4a^3 + 27b^2) mod \ p \neq 0. \tag{2}$$

The points on $E/F_p$ together with an extra point $O$ (point at infinity),forms a group $G$

$$G = \{(x, y) : x, y \in F_p \quad and \ (x, y) \in E/F_p\} \cup \{O\}. \tag{3}$$

Let the order of $G$ be $n$. $G$ is a cyclic additive group under the point addition operation '+' defined as follows:

Let $P, Q \in G, l$ be the line connecting $P$ and $Q$, and $R'$ be the third point of intersection of line $l$ with $E/F_p$. On reflecting $R'$ in $X$ axis and getting the point $R$. Define $R = P + Q$.

For any scalar $r$ the scalar point multiplication over $E/F_p$ can be computed as follows:

$$r \,.\, P = rP = P + P + \cdots + P(r \text{ times}).$$

The detail description of ECC can be found in [25,26] The following problems defined over $G$ are assumed to be intractable within polynomial time. The security of various cryptographic protocols are based on the intractability of these problems:

**Problem 1** (*Discrete Logarithm Problem (DLP) in G*) Given a generator $P$ of $G$ and an element $Q \in Z_p^*$, to find an integer $a \in Z_p^*$ such that $Q = aP$.

**Problem 2** (*Computational Diffie-Hellman (CDH) problem*) Given a generator $P$ of $G$ and $(aP, bP)$ for unknown $a, b \in_R Z_p^*$, compute $abP$.

The CDH assumption states that the probability of any polynomial-time algorithm to solve the CDH problem is negligible.

**Problem 3** (*Decisional Diffie-Hellman (DDH) Problem*) Given a generator $P$ of $G$ and $aP, bP, cP$ for unknown $a, b \in_R Z_p^*$, decide whether $c \equiv ab \ mod \ p$.

### 3.2 Certificateless Public Key Cryptography (CL-PKC)

In 2003 Al-Riyami and Paterson [1] introduced the concept of certificateless public key cryptography (CL-PKC) to overcome the key escrow limitation of the identity-based public key cryptography (ID-KC). In CL-PKC a trusted third party called key generation centre (KGC) supplies a user's partial private key. Then, the user combines the partial private key with a secret value (chosen by it self and that is unknown to the KGC) to obtain his full private key. In this way the KGC does not know the user's private keys. Then the user combines his secret value with the KGCs public parameters to compute his public key. Compared to the ID-PKC, the trust assumptions made of the trusted third party in CL-PKC are much reduced. In ID-PKC, users must trust the private key generator (PKG) not to abuse its knowledge of private keys in performing passive attacks, while in CL-PKC, users need only trust the KGC not to actively propagate false public keys. In CL-PKC a user can generate more than one pair of keys (private and public) for the same partial private key.

A CL-AKA protocol is defined by a collection of probabilistic polynomial-time algorithms as follows:

– **Setup** This algorithm is run by KGC. It takes security parameter $k$ as an input and returns a master private key $s$ and the system parameters $params$.

**Table 1** Notation table

| Notations | Meaning |
|---|---|
| $E/F_p$ | Elliptic curve over $F_p$ |
| $G$ | Additive group formed by the points on $E/F_p$ |
| $P$ | A generator of $G$ (point on EC) |
| $s$ | Master secret key of KGC (a number from $Z_p^*$) |
| $P_{pub}$ | Master public key of KGC (a point on EC) |
| $H_1, H_2$ | Cryptographic secure one way hash functions |
| $\{M\}_{\_K}$ | Symmetric encryption of the message $M$ with key $K$ |

- **Partial-Private-Key-Extract** This algorithm is also run by KGC. It takes $params, s$ and a users identity $ID_i$ as inputs, and returns the users partial private key $D_i$
- **Set-Secret-Value** This algorithm is run by the user. It takes $params$ and a users identity $ID_i$ as inputs, and returns the users secret value $x_i$.
- **Set-Private-Key** This algorithm is also run by the user. It takes $params$, a users identity $ID_i$, his partial private key $D_i$ and his secret value $x_i$ as inputs, and returns the users private key $SK_i$.
- **Set-Public-Key** This algorithm is also run by the user. It takes $params$, a users identity $ID_i$ and his secret value $x_i$ as inputs, and returns the users public key $PK_i$

The notation used in the entire paper are shown in Table 1 with their appropriate meanings.

## 4 Proposed Protocol

Like others CL-PKC based protocols the operation of proposed protocol is sequentially carried out by six algorithms. The first two algorithms are executed by the KGC and Algorithm 3–5 executed by every users while the 6th algorithm has two phases and partially executed by users as well as the KGC.

1. **Setup:** (By KGC) This algorithm take a security parameter $k \in Z^+$ as input and does the following:
   (a) Choose a $k$-bit prime $p$ and determine the tuple $\{F_p, E/F_p, G, P\}$, as discussed in Sect. 3.1
   where
   $E/F_p$: Elliptic curve over $F_p$
   $G$: Cyclic additive points group formed by points on $E/F_p$.
   $P$: Generator of $G$
   (b) Picks $s \in_R Z_p^*$ as the master private key and set its long term public key as $P_{pub} = s.P$.
   (c) For the security parameter $k$ KGC select two cryptographic secure hash function as $H_1 : \{0,1\}^* \rightarrow Z_q^*$; $H_2 : G \times G \rightarrow \{0,1\}^k$
   (d) The KGC publishes $param = \{F_p, E/F_p, G, P, P_{pub}, H_1, H_2\}$ as system parameters and secretly keeps the master key $s$.
2. **Partial-Private-Key-Extract:**
   This Algorithm Run by the KGC for every users.
   $params$ is the input for this algorithm.
   for every user $U_i$ KGC picks $r_i \in_R Z_p^*$ and calculate the following

$$\begin{cases} R_i = r_i.P, \\ h_i = H_1(ID_i \| R_i), \\ S_i = (r_i + s.h_i) mod\ p. \end{cases}$$

KGC issues user's partial private key as $D_i =< S_i, R_i >$ to the users $U_i$ having identity $ID_i$ through some secure channel.
On receiving the partial private key $D_i$ from KGC; $U_i$ can validate it by checking the Eq. 4.

$$R_i + H_1(ID_i \| R_i).P_{pub} = S_i.P \tag{4}$$

The partial private key is valid if the Eq. 4 hold and vice versa. Since

$$\begin{aligned} & R_i + H_1(ID_i \| R_i).P_{pub} \\ & = r_i.P + h_i.s.P \\ & = (r_i + s.h_i).P \\ & = S_i.P \end{aligned}$$

Note that all numeric calculations are done in $modulo\ p$.

3. **Set-Secret-Value:**
The user having identity $ID_i$ picks randomly $x_i \in_R Z_p^*$ and sets $x_i$ as his secret value.
4. **Set-Private-Key:** (By users)
The user with identity $ID_i$ takes the pair $SK_i =< x_i, S_i >$ as its private key.
5. **Set Public Key:**
The user having identity $ID_i$ calculates the followings

$$\begin{cases} P_i = (x_i + S_i).P, \\ Q_i = x_i.P \end{cases}$$

The tuple $PK_i =< P_i, Q_i >$ considered as his public key.
Note that publication of this public key must be delayed till the execution of first phase of key agreement algorithm (algorithm 6) i.e. "Member Registration for anonymity".
6. **Key-Agreement:**
Let the set of users $U = \{U_1, U_2, \ldots, U_n\}$ decided to agreed a secret group key anonymously. The entire key agreement process divided into two phases first phase concern with the registration of members necessary to achieve anonymity and it is executed among the KGC and users. While the second phase carried out the actual key establishment process among the users.
First Phase:
**Member Registration for Anonymity:**

In order to support the anonymity every users $U_i$ must do the following:

(a) randomly choose a temporary identity $TID_i$
(b) randomly choose $t_i \in_R Z_p^*$ and calculate the following:

$$\begin{cases} T_i = t_i.P_i, \\ \hat{T}_i = t_i.P_{pub}, \\ K_i = (S_i + x_i).\hat{T}_i = (K_{ix}, K_{iy}). \end{cases}$$

where $K_{ix}$ and $K_{iy}$ are $x$ and $y$ co-ordinates of $K_i$ over elliptic curve.
(c) $U_i$ send the following message to KGC:
$< ID_i, T_i, \{T_i, TID_i, Q_i\}_{\_K_{ix}} >$

Here $T_i$ and temporary identity $TID_i$ of the $U_i$ along with their public key component $Q_i$ are encrypted with $K_{ix}$ by a secure symmetric key algorithm.

On receiving the above message from every user $U_i$, $(1 \leq i \leq n)$ KGC do the following:

(a) Calculates $K_i = s.T_i = (K_{ix}, K_{iy})$, for $(1 \leq i \leq n)$.
(b) Used $K_{ix}$ to decrypt the message coming from $U_i$ and get the value of $T_i$, the temporary identity $TID_i$ and $U_i$'s public key component $Q_i$ for each user $U_i$, $(1 \leq i \leq n)$.
(c) KGC Verifies that whether the decrypted $T_i$ is same as the $T_i$ that was received from the $U_i$ or not, for all$(1 \leq i \leq n)$. If it is so the users are authentic.
(d) Now KGC calculates the following:

$$\begin{cases} h_{ti} = H_1(TID_i), \\ S_{ti} = (r_i + s.h_{ti}) mod\ p, \quad 1 \leq i \leq n. \end{cases}$$

(e) KGC send the following to every user encrypted with their $K_{ix}$:
$\{S_{ti} \| h_{t1} \| Q_1 \| h_{t2} \| Q_2 \| \ldots \| h_{tn} \| Q_n\}_{\_K_{ix}}$

Second Phase:
**Anonymous Key Establishment:**

This algorithm is run by every user $U_i$; $(1 \leq i \leq n)$ in parallel after receiving the encrypted message from KGC on executing the first phase.
$U_i$ first decrypt the message received from KGC and obtained their temporary partial secret key $S_{ti}$; hashed value of $TID_j$ and the public key component $Q_j$; $1 \leq j \leq n$ of other members. $U_i$ also find their positions on obtained list and set their subscript according to the position in the list.
Now every user $U_i$ chooses a random number $w_i \in_R Z_p^*$ and computes
$W_i = w_i.P$.
$U_i$ sends the following tuples to its two neighbours $U_{i-1}$ and $U_{i+1}$
$< h_{ti}, R_i, W_i >$
In this way $U_i$ also receives the similar tuples from $U_{i-1}$ and $U_{i+1}$.

On receiving the the above tuples from $U_{i-1}$ and $U_{i+1}$, $U_i$ first matches the value of $h_{ti}$ in the list to justify their subscript. After that $U_i$ calculates the following:

(a) $K_{i,i+1} = (x_i + S_{ti})W_{i+1} + w_i.(Q_{i+1} + R_{i+1} + h_{t(i+1)}.P_{pub})$
(b) $K'_{i,i+1} = w_i.W_{i+1}$
(c) $K_i^R = H_2(K_{i,i+1}, K'_{i,i+1})$
(d) $K_{i,i-1} = (x_i + S_{ti})W_{i-1} + w_i.(Q_{i-1} + R_{i-1} + h_{t(i-1)}.P_{pub})$
(e) $K'_{i,i-1} = w_i.W_{i-1}$
(f) $K_i^L = H_2(K_{i,i-1}, K'_{i,i-1})$
(g) $X_i = K_i^L \oplus K_i^R$

Now every user $U_i$ broadcast their $X_i$ to every other users in the network.
After getting all $X_j (j \neq i) U_i$ first verifies the received messages as follows:

$$X_1 \oplus X_2 \oplus \cdots \oplus X_n = 0 \tag{5}$$

On successful verification each user $U_i$ can calculates every unknown key components from $K_1^R, K_2^R, \ldots, K_n$ by chain XORing as follows:

$$\begin{aligned}
K_{i+1}^R &= X_{i+1} \oplus K_i^R \\
K_{i+2}^R &= X_{i+2} \oplus K_{i+1}^R \\
&\ldots \\
K_n^R &= X_n \oplus K_{n-1}^R \\
K_1^R &= X_1 \oplus K_n^R \\
&\ldots \\
K_{i-1}^R &= X_{i-1} \oplus K_{i-2}^R
\end{aligned}$$

Finally by every user the group session key is calculated as

$$K = H_1(K_1^R || K_2^R || \ldots || K_n^R)$$

Figure 1 shows the entire process of anonymous key establishment including the chain $XORing$, and session key computation.

The session key is agreed because:

$$\begin{aligned}
K_{i,i+1} &= (x_i + S_{ti})W_{i+1} + w_i.(Q_{i+1} + R_{i+1} + h_{t(i+1)}.P_{pub}) \\
&= (x_i + S_{ti})W_{i+1} + w_i.(x_{i+1}.P + r_{i+1}.P + h_{t(i+1)}.s.P) \\
&= (x_i + S_{ti})W_{i+1} + w_i.(x_{i+1} + r_{i+1} + h_{t(i+1)}.s).P \\
&= (x_i.P + S_{ti}.P).w_{i+1} + w_i.P(x_{i+1} + S_{t(i+1)}) \\
&= (Q_i + (r_i + s.h_{ti})P).w_{i+1} + W_i(x_{i+1} + S_{t(i+1)}) \\
&= (x_{i+1} + S_{t(i+1)})W_i + w_{i+1}.(Q_i + R_i + h_{ti}.P_{pub}) \\
&= K_{i+1,i}
\end{aligned}$$

Also,

$$\begin{aligned}
K'_{i,i+1} &= w_i.W_{i+1} \\
&= w_i.w_{i+1}.P \\
&= w_{i+1}W_i \\
&= K'_{i+1,i}
\end{aligned}$$

Similarly

$$K_{i,i-1} = K_{i-1,i} \quad \text{and} \quad K'_{i,i-1} = K'_{i-1,i}$$
$$K_i^R = K_{i+1}^L \quad \text{or} \quad K_i^L = K_{i-1}^R$$

### 4.1 Join Operation

Assume that the users in set $\{U_1, U_2, \ldots, U_n\}$ have shared a common session key $SK$ by performing group key agreement algorithm discussed in Sect. 4.1 and now a set of members $\{U_{n+1}, U_{n+2}, \ldots, U_m\}$ wish to join the existing group. Assume that identities of all new members are verified by KGC and all new members already sets their public/private key pair $\{< S_i, x_i >, < P_i, Q_i >\}$, and extract their temporary partial secret key $S_{ti}$ as previous. Now KGC first send the hash value of new member's temporary id $h_{ti}$; $n + 1 \leq i \leq m$. to all members $\{U_1, U_2, \ldots, U_n, U_{n+1}, \ldots, U_m\}$. If the existing group members decided to

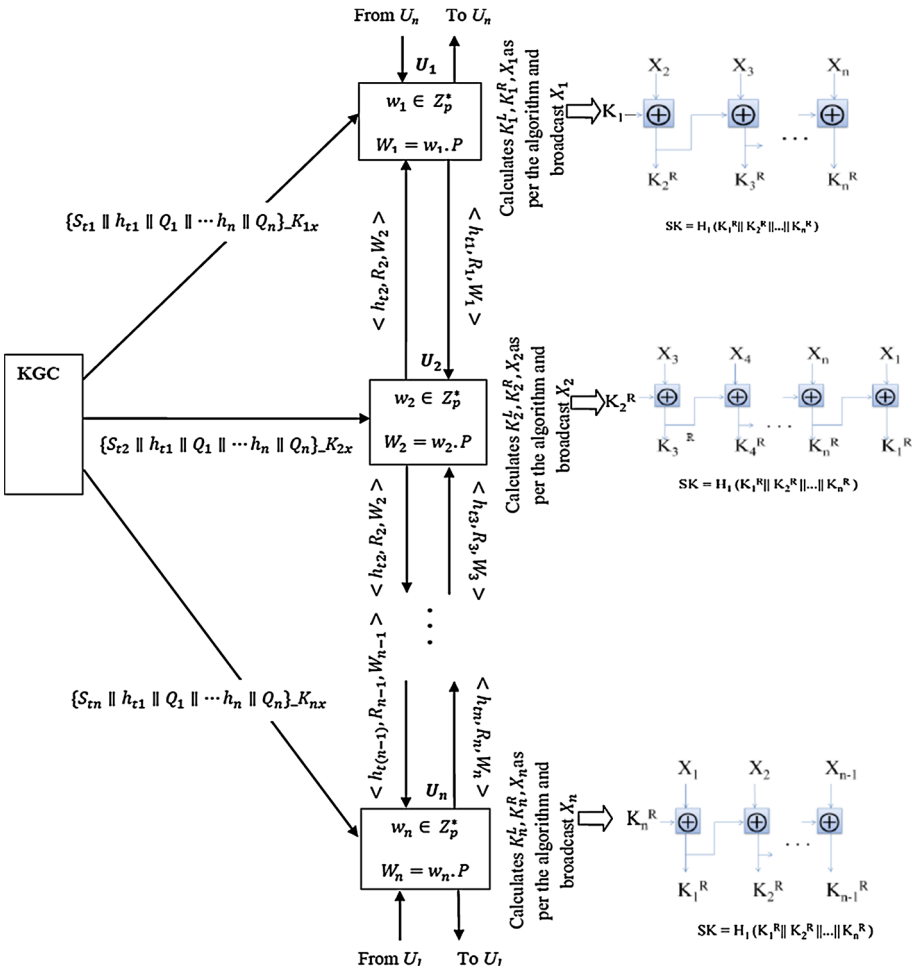



**Fig. 1** Anonymous key establishment process

allow joining the new set of members then two of them ($U_1$ and $U_n$) comes forward as the join controllers to handles the join operation. The join operation performed by the following steps:

1. $U_1$ and $U_n$ updates their public key/private key pair by choosing new secret value ($x_1$ and $x_n$).
2. $U_1$ chooses new random number $w_1 \in {}_R Zp^*$ and recalculates $W_1 = w_1.P$.
3. Similarly $U_n$ also choose a new random number $w_n \in {}_R Zp^*$ and recalculate $W_n = w_n.P$
4. $U_1$ and $U_n$ calculates two secret session keys by using their updated values. The first session key $SK_1$ calculated with the existing members i.e. $\{U_1, U_2, \ldots, U_n\}$ while the second session key $SK_2$ is calculated along with the new set of members i.e. $\{U_1, U_{n+1}, U_{n+2}, \ldots, U_m, U_n\}$ as shown in Fig. 2 para Note that to calculate $SK_1$ only the values of $K_1^R$, $K_{n-1}^R$, and $K_n^R$ have to be recalculate by the members while others $K_i^R$ $(1 \leq i \leq n; i \notin \{1, n-1, n\})$ are same.

**Fig. 2** Join operation

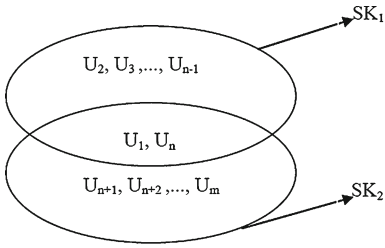



**Table 2** Cost of join operation

| Joining cost of $n'$ members | Point multiplications | Point additions | Messages | Pairings |
|---|---|---|---|---|
| By initialization of $n + n'$ members | $14(n + n')$ | $6(n + n')$ | $5(n + n')$ | 0 |
| By join procedure | $14n' + 46$ | $6n' + 21$ | $5n' + 12$ | 0 |

5. $U_1$ or $U_n$ sends $SK_1$ to all set of new members$\{U_{n+1}, U_{n+2}, \ldots, U_m\}$ and $SK_2$ to the all others existing members $\{U_2, U_3, \ldots, U_{n-1}\}$. First message is encrypted with $SK_2$ while the second message is encrypted with their old group session key $SK$.
6. Finally the new group session key calculated by the every members i.e. $(U_1, U_2, \ldots, U_n, U_{n+1}, \ldots, U_m)$ as

$$SK_{new} = H_1(SK_1||SK_2).$$

The basic idea of join operation are illustrated in Figure 2.

#### *4.1.1 Analysis of Join Procedure*

In any GKA protocol if $n$ and $n'$ are the number of current group members and joining members respectively, then a join procedure is efficient and meaningful only when the total cost of join procedure with $n'$ joining members is significantly less than the cost of group key agreement process (initialization) with $n + n'$ members for all $n' \ll n$.

The cost of above join procedure is summarized in the Table 2.

Table 2 shows that join cost of $n'$ new members requires asymptotically $O(n')$ point multiplications, additions and messages by proposed join procedure. While it requires $O(n)$ cost for the same by initialization operation and since $n' \ll n$ therefore $O(n') \ll O(n)$ Hence the proposed join procedure is an efficient one.

### 4.2 Leave Operation

If a set of members are leaving from the current group then the group session key of resulting group must be updated to provide the forward secrecy. For leave operation the present paper taken the idea of remove algorithm from [7]. Suppose $U = \{U_1, U_2 \ldots, U_n\}$ be the current group and $L = \{U_{l1}, U_{l2}, \ldots, U_{ln}\}$ is the set of leaving members, where $\{l1, l2, l3, \ldots, ln\} \subseteq \{1, 2, \ldots, n\}$ and $n' < n$. We represent the set of remaining members as $A = \{U_{a1}, U_{a2}, \ldots, U_{a(n-n')}\} = U - L$. Any one of the member from $A$ (usually $U_{a1}$) as a leave controller receives the leave request with their temporary identity and

broadcast the $h_{ti}$ of all the leaving members $L$ to $A$. The operation employs the following steps:

1. Each member $U_i \in A$, check if $U_i$ 's left member $U_{i-1}$ or/and right member $U_{i+1}$ has been leaved i.e. $U_{i-1}$ or $U_{i+1} \in L$ then $U_i$ reset their secret value $x_i \in_R Z_p^*$ and updates their private/public key pair. $U_i$ also updates their random secret $w_i$ and accordingly recalculates their $K_i^R$ and $K_i^L$ with the contribution of its neighbours (left and right) alive members. Finally $U_i$ calculates $X_{newi} = K_i^L \oplus K_i^R$ and broadcast to $A$.
2. For any other member $U_j \in A$ If the value of its $K_j^R$ needs to change since the contribution of its right member $U_{j+1}$ is changed due to Step 1(this case is occurs when $U_{j+2} \in L$). Then $U_j$ also recomputes their $K_j^R$ and accordingly recompute and broadcast its new value of $X_j$ as $X_{newj}$.
   All other members $U_k \in A$, whose neighbour(s) or contribution of neighbour(s) dose not changed, do nothing but simply broadcast their old calculated $X_k$ as its new value $X_{newk}$ i.e. $(X_{newk} = X_k)$.
3. Each member $U_i \in A$, after receiving all $X_{newj}$ $(j \neq i)$ first verifies

$$X_{newa1} \oplus X_{newa2} \oplus \cdots \oplus X_{newa(n-n)} = 0$$

   If verification is successful they can calculates the required value of $K_j^R$ as previous. Finally the new session key calculated as:

$$SK_{new} = H_1\left(K_{a1}^R || K_{a2}^R || \ldots || K_{a(n-n)}^R\right)$$

It is noted that if the number of leaving members are too large (closer to $n$) then re-initialization of group with alive members is more efficient than the leaving operation, but if $n' \ll n$ then the proposed leave procedure works efficiently. The cost of leave operation is summarized in Table 4.

## 5 Security and Privacy Analysis

In this section, we analyze some security and privacy attributes for the proposed protocol. The security attributes are taken from the [7] and privacy attributes from [22] Here we trying to give security justify the proposed protocol as follows:

**Implicit Key Authentication:** The group session key is computed by each users ephemeral and long-term private keys. So, the users are assured that no other users except the partners who have the private keys can learn the group session key.

**Known Session Key Security:** In each session, each user $U_i$ set new secret value so that their private/public key pair updated, and the generated group session key is depends up on private key of all users. The adversary that compromises one session key (if so) would not compromise other session keys, so the proposed protocol provides known session key security.

**Forward Secrecy:** If the adversary compromises one or more users partial private key $D_i =< S_i, R_i >$ somehow, he cannot further calculate anything without knowing users secret value $x_i$. Since $D_i$ is not complete private key in the certificateless cryptosystem used.

Even if the adversary compromises the users complete private key $< S_i, x_i >$ he cannot compute the value of $K_i^R$ or $K_i^L$ without knowing $w_i$. Thus if the long term private keys of one or more entities are compromised, the secrecy of previously established session key will not be affected.

**No Key-Compromise Impersonation:** Suppose that a user $U_i$'s long-term private key $< S_i, x_i >$ has been disclosed, and an adversary $E$ try to masquerade as the user $U_j$ to all other users. Since the private key is computed with the contribution of $S_i$ which is securely calculated by KGC, and the secrete value $x_i$ randomly chooses by the user itself. Thus the private keys of users are totally independent from each others. Hence the adversary may impersonate the compromised user in the subsequent protocols, but it cannot impersonate other users.

**No Key Control:** The group session key in the protocol is determined by all members long-term private keys $< S_i, x_i >$ and ephemeral secrete value neither party alone can control the outcome of the session key. No one can restrict it to lie in some predetermined value. Ephemeral private key revealing resistance. If all users ephemeral values ti have been compromised, our protocol is also secure. Because the adversary doesn't know the long-term private key of any user, he cannot compute the group session key.

**Perfect Forward Security:** Even if KGC is compromised, the private keys of users cannot be disclose this is the main characteristic of certificate less public key cryptography. Thus the previously established session keys are not compromised.

**Complete Anonymity:** In proposed protocol, both the users in the group and the adversary outside the group cannot get the real identity of the group partners.

In the first step of group register phase, each user $U_i$ encrypts his real identity and temporary identity using symmetric key $K_{ix}$ established with KGC. Thus, only KGC can decrypt this encrypted message and get the users real and temporary identity pairs. Both the other users in the group and the adversary outside the group cannot get the relation between a users real identity and temporary identity.

In the next step of group register phase, KGC encrypts one users temporary private key using the symmetric key established with that user. Thus,only legitimate group user can decrypt this encrypted message and get his temporary private key.Both the other users in the group and the adversary outside the group cannot decrypt this message to get the relation between a users real identity and temporary identity. Only the hash value of temporary identities of other users $h_{ti}$ can get the every user and in the group key agreement phase, each user in the group only uses that hash value $h_{ti}$ of others. Since both the users in the group and the adversary outside the group cannot get the relation between a users real identity and temporary identity, they cannot obtain any information of one users real identity from the messages in the protocol.

**Unlinkability:** Anonymity would be meaningless without unlinkability. The adversary can still trace an unknown user without knowing his real identity. In proposed anonymous protocol, a user can use different temporary identity in every independent execution of the protocol. Although the adversary wants to trace user information using all the temporary identity of different sessions, he cannot link them to any user information because temporary identity always change in each session and do not carry any information about the group member's identities.

## 6 Formal Security Analysis using AVISPA Tool

Recently, AVISPA tool [27] is widely used by many researchers for the automated validation of Internet security protocols and applications. The AVISPA is a push button tool designed by University of Geneva, Italy using the concept of Dolev and Yao intruder model [28], where the network is controlled by an intruder (Active and passive); however he is not allowed to crack

```
role kgc(Kc,U1,U2,U3 : agent,P,S,ID1,ID2,ID3 : message,K1,K2,K3,K : public_Key,
H,H1,H2,M,M1,A,A1 : hash_func,SND,RCV:channel(dy))

played_by Kc
def=
local
State : nat,
Ppub,S1,S2,S3,R1,R2,R3,Rr1,Rr2,Rr3,Sig1,Sig2,Sig3 : message

%knowledge(Kc)= {inv(K)}
init
State := 0

transition
1. State=0 /\ RCV(start)=|>

State' :=1   /\ Rr1':= new() /\ Rr2':= new() /\ Rr3':= new()
/\R1':= M(P,Rr1')/\ R2':= M(P,Rr2') /\ R3':= M(P,Rr3')
/\ S1':= A1(Rr1',M1(S,H(ID1))) /\ S2':= A1(Rr2',M1(S,H(ID2)))
/\ S3':= A1(Rr3',M1(S,H(ID3)))/\ Ppub':= M(P,S)
/\ Sig1':= H2(A(R1',M(Ppub',H(ID1))))
/\ Sig2':= H2(A(R2',M(Ppub',H(ID2))))
/\ Sig3':= H2(A(R3',M(Ppub',H(ID3))))

/\
SND({{R1'.S1'.Ppub'.Sig1'}_inv(K)}_K1)/\SND({{R2'.S2'.Ppub'.Sig2'}_inv(K)}_K2)
/\SND({{R3'.S3'.Ppub'.Sig3'}_inv(K)}_K3)
/\ witness(Kc,U1,u1_kgc_r1,R1')
/\ witness(Kc,U1,u1_kgc_s1,S1')
/\ witness(Kc,U1,u1_kgc_ppub,Ppub')

/\ witness(Kc,U2,u2_kgc_r2,R2')
/\ witness(Kc,U2,u2_kgc_s2,S2')
/\ witness(Kc,U2,u2_kgc_ppub,Ppub')

/\ witness(Kc,U3,u3_kgc_r3,R3')
/\ witness(Kc,U3,u3_kgc_s3,S3')
/\ witness(Kc,U3,u3_kgc_ppub,Ppub')


end role
```

**Fig. 3** Role specification of KGC in HLPSL

the underlying cryptography. The AVISPA tool supports high level protocol specification language (HLPSL) based on which the cryptographic protocols are to be implemented and analyzed. It has four model checkers/back-ends, called on-the-fly model-checker (OFMC), constraint-logic-based attack searcher (CL-AtSe), SAT-based model-checker (SATMC) and TA4SP (Tree Automata-based Protocol Analyzer). The details description about AVISPA and HLPSL can be found in [29].

The role specification of KGC and an user involved in key generation of the proposed protocol (assuming just for three users) are shown in Figs. 3 and 4 respectively. The results of OFMC & CL-AtSe are shown in Figs. 5 and 6 respectively.

```
role user1(U1,U2,U3,Kc: agent, K1,K2,K3,K:public_key,ID1,ID2,ID3 :message,
H,H1,H2,M,M1,A,A1: hash_func,P: message, SND,RCV : channel(dy))

played_by U1 def=

local

State : nat,
W3,W2,W1,S1,R1,R2,R3,X1,K12,K13,K1r,K2r,K3r,EX1,EX2,EX3,K1R,K1L,Ppub,Q1,
Q2,Q3,Ww1,Sig1 : message,
SK: symmetric_key,
IDRing: (agent.message)set

%knowledge(U1) = {inv(K1)}
init

State:= 0 /\ IDRing:= {U1.ID1,U2.ID2,U3.ID3}

transition

1. State=0 /\ RCV({{R1'.S1'.Ppub'.Sig1'}_inv(K)}_K1)
/\ Sig1'= {{M(P,S1')}_H2}_inv(K) =|>

%equal( Sig1', (M(P,S1')))
        State':=1
        /\request(U1,Kc,u1_kgc_r1,R1')
        /\ request(U1,Kc,u1_kgc_s1,S1')
        /\ request(U1,Kc,u1_kgc_ppub,Ppub')

        /\ X1':=new()/\ Ww1':=new()/\W1':= M(P,Ww1')/\ Q1':= M(P,X1') /\
        SND({U1.ID1.W1'.R1'.Q1'}_inv(K1)) /\
        %witness(U1,U5,u5_u1_t,T1'.R1')/\
        %witness(U1,U4,u4_u1_t,T1''.R1')/\
        witness(U1,U2,u2_u1_w,W1'.R1')/\
        witness(U1,U3,u3_u1_w,W1'.R1')

2.State=1 /\    RCV( {U2.ID2.W2'.R2'.Q2'}_inv(K2)) /\ in(U2.ID2,IDRing)/\
                RCV( {U3.ID3.W3'.R3'.Q3'}_inv(K3)) /\ in(U3.ID3,IDRing) =|>
State':=2 /\
request(U1,U2,u1_u2_w,W2'.R2')/\
request(U1,U3,u1_u3_w,W3'.R3')/\

K12':= A(M(W2',A1(S1,X1)),M(A(R2',Q2',M(Ppub,H(ID2))),Ww1))/\
K13':= A(M(W3',A1(S1,X1)),M(A(R3',Q3',M(Ppub,H(ID3))),Ww1))/\

K1R' := H1(K12')/\
K1L' := H1(K13')/\
EX1' := xor(K1L',K1R')/\

SND(EX1')
%/\witness(U1,U2,u2_u1_ex,EX1')/\
%witness(U1,U3,u3_u1_ex,EX1')

3. State = 2 /\ RCV(EX2')/\RCV(EX3')=|>

%%%%%/\xor(EX1,xor(EX2',EX3')) = 0
State':= 3 /\
%request(U1,U2,u1_u2_ex,EX2')/\
%request(U1,U3,u1_u3_ex,EX3')/\

K2r':= xor(EX2',K1R)/\
K3r':= xor(EX3',K2r')/\
SK':= H(K1R.K2r'.K3r')/\
secret(SK',sk,{U1,U2,U3})
end role
```

**Fig. 4** Role specification of User 1 in HLPSL

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/avispa/web-interface-computation/./tempdir/workfile7vEjL3.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.66s
  visitedNodes: 16 nodes
  depth: 4 plies
```

**Fig. 5** Simulation result of OFMC

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/avispa/web-interface-computation/./tempdir/workfileOkIMQW.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 9 states
  Reachable  : 0 states
  Translation: 1.28 seconds
  Computation: 0.00 seconds
```

**Fig. 6** Simulation result of CL-AtSe back end

## 7 Security Comparison

This section compares the proposed protocol with some other existing protocol discussed in Sect. 2 based on the various security parameters. Table 3 illustrates this security comparison. The data of Table 3 under proposed protocol is obtained from security analysis section (Sect. 5 while the same for existing protocols are taken from the referenced papers.

It may be noted that the 6th column of Table 3 shows the security parameters of backward/forward security which concern about the protection of the confidentiality of updated group key from the joining/leaving member(s) in the event of join/leave operations. The same will be not applicable (NA) in case of static protocols.

## 8 Performance Comparison

The present work compares the performance of proposed protocol with some other existing GKA protocols in terms of communication and computation costs of various group operations

**Table 3** Security comparison table

| Protocol | Authenticity of public key | User anonymity | For static group | For dynamic group | Forward & backward security | Perfect forward security |
|---|---|---|---|---|---|---|
| Wang et al. [24] | Certificate-based | No | Yes | Yes | Yes | Yes |
| Zheng et al. [9] | Certificate-based | No | Yes | Yes | Yes | No |
| Konstantinou [15] | ID-based | No | Yes | No | NA | Yes |
| Xie and He [7] | ID-based | No | Yes | Yes | Yes | Yes |
| Wan et al. [16] | ID-based | Yes | Yes | Yes | No | No |
| Yao and Feng [22] | ID-based | Yes | Yes | No | NA | Yes |
| Park et al. [23] | ID-based | Yes | Yes | Yes | Yes | Yes |
| Heo et al. [4] | Certificateless | No | Yes | Yes | Yes | No |
| Teng et al. [2] | Certificateless | No | Yes | No | NA | Yes |
| Lee et al. [5] | Certificateless | No | Yes | Yes | Yes | Yes |
| Kumar et al. [8] | Certificateless | No | Yes | No | NA | Yes |
| Proposed | Certificateless | Yes | Yes | Yes | Yes | Yes |

**Table 4** Performance comparison table

| Protocol | Group operation | Rounds | PM | PA | Pairings | Message |
|---|---|---|---|---|---|---|
| Teng et al. [2] | Initialization | 2 | $O(n)$ | 0 | $O(n^2)$ | $O(n^2)$ |
| Geng et al. [3] | Initialization | 2 | $5n$ | 0 | $2n^2$ | $4n$ |
| Cao et al. [6] | Initialization | 2 | $5n$ | $n$ | $3n$ | $2n$ |
| XIE Liyun [7] | Initialization | 2 | $n^2 + 3n$ | $n^2$ | 0 | $2n$ |
| | Join | 1 | $(n + n')^2 + 5n' + 7$ | $(n + n')^2 + n' + 2$ | 0 | $2n' + 3$ |
| Wan et al. [16] | Initialization | 3 | $3n$ | 0 | $2n$ | $4n$ |
| | Join | 1 | $(n * n')$ | 0 | $2(1 + n')$ | $7n'$ |
| | Leave | 1 | $6n'$ | 0 | $2n'$ | $7n'$ |
| Proposed | Initialization | 3 | $14n$ | $6n$ | 0 | $5n$ |
| | Join | 2 | $14n' + 46$ | $6n' + 21$ | 0 | $5n' + 12$ |
| | Leave | 1 | $14(n - n')$ | $3(n - n')$ | 0 | $(n - n') + 3$ |

and the result is showed in Table 4 (where $n$ is the number of users). Table 4 uses the following notations for various comparison parameters.

PM: number of scalar point multiplications.
PA: number of elliptic curve point additions.
Message: total number of message overheads during key agreement phase (including unicast and broadcast).
$n$: number of participants.
pairings: number of bilinear pairings needed in key agreement process (zero in case of proposed protocol)

It may be noted that protocols [2,3,6] are not dynamic (Join and Leave procedures are not exist) so only the initialization cost are tabulated in Table 4 and it is taken from their respective papers. For Xie Liyun protocol [7] the cost of Initialization are taken from the

tabulated value of [7]. While the cost of join or Leave operation are not given in its paper. So first it is calculated based on the decryption of their algorithms and tabulated in present paper for comparison. However the dynamic cost(cost of join and leave operation) of Wan's protocol [16] are described for single member join/leave in their paper. For comparison, the unit cost is multiplied by $n'$ and tabulated in Table 4. Cost of Leave operation of present paper as well as [7] are highly depends on the position of the leaving members in the current group the tabulated value of leaving cost of proposed protocol are of worst case i.e. when all alive members needs to updates their ephemeral secret and calculates their new contributions. It can be observed that overall worst case cost of leave operation is also much less than the initialization cost of $n - n'$ members.

From Table 4 it can be observed that the proposed protocol provides comparable communication and computation cost with zero pairing computations.

## 9 Conclusion

This paper proposes a complete anonymous pairing-free group key agreement protocol based on certificateless cryptosystem. Protocol also provide efficient dynamic group operations like join and leave with backward and forward security. Further the proposed protocol does not require bilinear pairing operations, and thus it provides computational benefits than the other existing protocols. It also provide complete privacy among group members during entire communications which is the key requirement of various applications. Hence the proposed protocol trying to encapsulates the advantages a pairing-free protocol, certificateless cryptosystem, and complete anonymity. Moreover the present work tested the security of the proposed protocol by using AVISPA tool and it is found secure under various attacks. In addition the proposed protocol also full fill the required security and privacy attributes of a group key generation protocol as discussed in Sect. 5. The present technique may create an attraction for low power wireless devices such as mobile phones because pairing based applications can be hard to implement on these. The protocol is also suitable in such an application which require anonymous but authentic group conversation like evoting, unbiased discussion, etc.

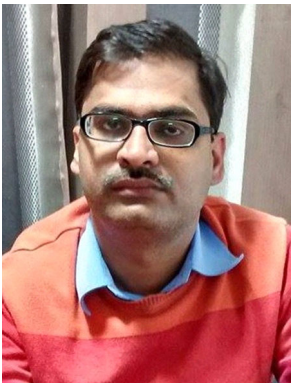
**Acknowledgments** The second author of this article is would like to thank UGC (University Grant Commission) for their partial support in this research work.

## References

1. Al-Riyami, S. S., & Paterson K. G. (2003). Certificateless public key cryptography. *2894*, 452–473.
2. Teng, J., & Wu, C. (2012). A provable authenticated certificateless group key agreement with constant rounds. *Communications and Networks, Journal of*, *14*(1), 104–110.
3. Geng, M., Zhang, F., & Gao, M. (2009). A secure certificateless authenticated group key agreement protocol. *Multimedia Information Networking and Security, 2009. MINES'09. International Conference on*, *1*, 342–346.
4. Heo S., Kim Z., & Kim K. (2007). Certificateless authenticated group key agreement protocol for dynamic groups. In *Global telecommunications conference, 2007. GLOBECOM'07. IEEE*, (pp. 464–468) Nov.
5. Lee E. J., Lee S. E., & Yoo K. Y. (2008). A certificateless authenticated group key agreement protocol providing forward secrecy. In *Proceedings of the 2008 international symposium on ubiquitous multimedia computing, UMC'08* (pp. 124–129), Washington, DC, USA, IEEE Computer Society.
6. Cao, C., Ma, J., & Moon, S. (2007). Provable efficient certificateless group key exchange protocol. *Wuhan University Journal of Natural Sciences*, *12*(1), 41–45.
7. Xie, L., & He, M. (2010). A dynamic id-based authenticated group key exchange protocol without pairings. *Wuhan University Journal of Natural Sciences*, *15*(3), 255–260.

8. Kumar, A., Tripathi, S., Jaiswal, P. (2014). A pairing free certificateless group key agreement protocol with constant round. In *Advanced Computing, Networking and Informatics-Volume 2, Volume 28 of Smart Innovation, Systems and Technologies*, (pp 341–349). Springer International Publishing.
9. Zheng, S., Manz, D., & Alves-Foss, J. (2007). A communication-computation efficient group key algorithm for large and dynamic groups. *Computer Networks*, *51*(1), 69–93.
10. Kim, Y., Perrig, A., & Tsudik, G. (2004). Tree-based group key agreement. *ACM Transactions on Information and System Security*, *7*(1), 60–96.
11. Kim, Y., Perrig, A., & Tsudik, G. (2004). Group key agreement efficient in communication. *IEEE Transactions on Computers*, *53*(7), 905–921.
12. Shamir, A. (1985). Identity-based cryptosystems and signature schemes. *Advances in Cryptology, Volume 196 of Lecture Notes in Computer Science* (pp. 47–53). Berlin Heidelberg: Springer.
13. Reddy, K. C., & Nalla, D. (2002). Identity based authenticated group key agreement protocol. In *Progress in Cryptology INDOCRYPT 2002, Volume 2551 of Lecture Notes in Computer Science* (pp 215–233). Berlin Heidelberg: Springer.
14. Choi, K. Y., Hwang, J. Y., & Lee, D. H. (2004). Efficient id-based group key agreement with bilinear maps. *Public Key Cryptography PKC 2004, volume 2947 of Lecture Notes in Computer Science* (pp. 130–144). Berlin Heidelberg: Springer.
15. Konstantinou, E. (2013). An efficient constant round id-based group key agreement protocol for ad hoc networks. *Network and System Security, volume 7873 of Lecture Notes in Computer Science* (pp. 563–574) Berlin Heidelberg: Springer.
16. Wan, Z., Ren, K., Lou, W., & Preneel, B. (2008). Anonymous id-based group key agreement for wireless networks. In *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, 2615–2620. doi:10.1109/WCNC.2008.459
17. He, D., & Chen, Y. (2011). An efficient certificateless authenticated key agreement protocol without bilinear pairings. CoRR, arXiv:1106.3898.
18. Kim, Y. J., Kang, J. G., Kim, C. H., & Kim, Y. M. (2013). An efficient pairing-free certificateless two-party authenticated key agreement protocol in the eck model. CoRR, arXiv:1304.0383.
19. Mohamed, N. A. F., Hashim, M. H. A., Bashier, E. B. M., & Hassouna M. E. H. (2012). Fully-secure and efficient pairing-free certificateless authenticated key agreement protocol. In *Internet Security (WorldCIS), 2012 World Congress on* (pp. 167–172). IEEE.
20. Farouk, A., Fouad M. M., & Abdelhafez A. A. (2014). Analysis and improvement of pairing-free certificate-less two-party authenticated key agreement protocol for grid computing. arXiv preprint arXiv:1403.2844.
21. Luo, M., & Zhao, H. (2014) An authentication and key agreement mechanism for multi-domain wireless networks using certificateless public-key cryptography. *Wireless Personal Communications*, 1–20. doi:10.1007/s11277-014-2157-5.
22. Yao, G., & Feng, D. (2010). A complete anonymous group key agreement protocol. *Networks Security Wireless Communications and Trusted Computing (NSWCTC) 2010 Second International Conference on*, *2*, 357–360.
23. Park, H., Kim, Z., & Kim, K. (2009). Forward secure id-based group key agreement protocol with anonymity. In *Emerging Security Information, Systems and Technologies, 2009. SECURWARE '09. Third International Conference on*, (274–279). IEEE. doi:10.1109/SECURWARE.2009.49.
24. Wang, E. K., Ye, Y., & Xu, X. (2014). Location-based distributed group key agreement scheme for vehicular ad hoc network. *International Journal of Distributed Sensor Networks*.http://dx.doi.org/10.1155/2014/759601.
25. Stallings, W. (2002). *Cryptography and network: Security principles and practice* (3rd ed.). New Jersey: Pearson Education.
26. Stinson, D. R. (2006). *Cryptography : theory and practice. Discrete mathematics and its applications*. Boca Raton: Chapman & Hall/CRC.
27. Viganó, L. (2006). Automated security protocol analysis with the AVISPA tool. *Electronic Notes in Theoretical Computer Science, 155*, 61–86
28. Dolev, D., & Yao, A. C. (1983). On the security of public key protocols. *Information Theory. IEEE Transactions on*, *29*(2), 198–208.
29. Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuellar, J., Drielsma, PH, Heám, P. C., Mantovani, J., Mödersheim, S., von Oheimb, D., Rusinowitch, M., Santiago, J., Turuani, M., Viganò, L., & Vigneron, L. (2005). The AVISPA tool for the automated validation of internet security protocols and applications. In *Proceedings of the 17th international conference on computer aided verification (CAV'05), volume 3576 of LNCS*. Springer.

**Abhimanyu Kumar** completed his Engineering degree in Computer Science and Engineering, from R. P. Sharma Institute of Technology, Patna (affiliated to Magadh University, Bodh Gaya, Bihar). Currenty he is pursuing Ph.D. under supervision of Dr. Sachin Tripathi in the department of Computer Science and Engineering at Indian School of Mines, Dhanbad, Jharkhand, India. His research area is group security and their applications.



**Sachin Tripathi** is an Assistant Professor in Computer Science & Engineering Department at Indian School of Mines, Dhanbad, Jharkhand, India. He received his Ph.D. in Computer Science and Engineering from the Indian School of Mines and has been teaching computer science subjects for over more than ten years. His research interest is in group security.