Anonymous ID-based Group Key Agreement Protocol without Pairing

Abhimanyu Kumar and Sachin Tripathi
(Corresponding author: Abhimanyu Kumar)

Department of Computer Science and Engineering, Indian School of Mines
Dhanbad, Jharkhand 826004, India
(Email: abhi_a1ks@yahoo.co.in & var_1285@yahoo.com)
(Received Nov. 29, 2013; revised and accepted Apr. 24 & Nov. 22, 2014)

Abstract

Secure group communication is an active area of research and its popularity is fuelled by the growing importance of group-oriented applications such as teleconferences, collaborative workspace, pay per-view etc. A number of group key agreement protocols have been proposed for these objectives. However most of the protocols have not considers the anonymity of the participants. Although in some applications the privacy of member's identity becomes more crucial and urgent especially for mobile users of a wireless network due to the open nature of radio media. The protocols having complex computations like large modular exponentiations, pairing computations, etc. are not well suited in wireless environments. Hence this paper proposes an anonymous ID-based group key agreement protocol without bilinear pairings. The proposed protocol also have anonymous join and leave procedures to facilitates the dynamic group operations. Security and performance analysis of proposed protocol shows that it provides strong security protection under different security attributes, and needs comparatively less computation and communication overheads than the other existing protocols. In addition the formal security verification of proposed protocol has been done by using AVISPA tool which shows that it is unforgeable against active and passive attacks.

Keywords: Anonymity, AVISPA, elliptic curve cryptography, group key agreement, identity-based cryptography

1 Introduction

Collaborative applications such as multimedia conferences, distributed simulations, multi-user games and replicated servers have become extremely popular during the last decades. All these applications are executed through Internet connections that in many cases should be properly secured. Moreover, wireless networks, mobile ad hoc networks and sensor networks are used extensively in many areas of interest (ranging from homes, schools and universities to inaccessible terrains, disaster places, etc.), where security is really crucial. The realization of such efficient, robust and secure environments is a challenging algorithmic and technological task. How to communicate securely over an insecure channel is a fundament problem. So that all users that participate in the particular application should be able to communicate securely and exchange information that is inaccessible to any external entity. Hence, there is a need for finding a protocol that provides such a confidential communication, termed usually as secure group communication or secure conferences. These kind of secure conferences usually achieved by symmetric key cryptography and often require an efficient group key establishment protocol. The goal of such a protocol is to establish a common secret key among the users, called group key, which can be used for data encryption and authentication among them.

Group key establishment protocols can be divided into two subcategories: the Key Transfer Protocols and the Key Agreement Protocols. During the execution of a Key Transfer Protocol an entity creates or obtains a secret value, which transmits it securely to the rest of the entities. In a Group Key Agreement (GKA) Protocol, a shared secret is derived as a function of information contributed by or associated with all the members in the group, such that no party in the group can predetermine the resulting value.

In many cases especially in wireless environments the user's anonymity also becomes more crucial and important for mobile users along with their others security issues [24]. Out of several existing group key agreement protocols based on different cryptosystems, very few of them the privacy of the users's identities are taken into account. Since the world is going wireless and ubiquitous, the privacy of the users also becomes a very challenging issue as like security because if the group member's identities are exposed to everyone including outside eavesdroppers, they can trace a mobile users, find out a specific users movement patterns etc. [24].

Hence this paper proposes an anonymous group key agreement protocol as like [24] based on ID based cryptosystem without pairing which is more suitable for wireless networks. It becomes more efficient because the relative computation cost of the pairing is many times higher than that of the scalar point multiplication over elliptic curve group. In wireless environment to construct a secure meeting session by a group of mobile users without others knowing who are in the meeting and to make sure that the users in the meeting are indeed those expected group members, the group key agreement protocol should be able to protect the user's identity from the outside eavesdroppers during the execution of the protocol. This is achieved in proposed protocol by using pseudonyms for every users and employing anonymous encryption scheme. The proposed scheme is ID-based, so it simplifies the complex certificate management of the traditional public key cryptography. Since dynamicity is a major issue for today's networks so the proposed protocol also supports all dynamic operations such as Join, Leave, Merge, to cope with dynamic membership events. The importance of group rekeying in dynamic group are summarized in [13]. The security and privacy of the proposed protocol is also analyzed in this paper and it is found that it provides strong security protection with anonymity and has relatively efficient performance in terms of communication and computation overheads than the others existing ID-based GKA protocols. Moreover the security of proposed technique is also validates by using AVISPA (Automated Validation of Internet Security Protocols and Applications) tool which shows that Protocol is safe under its different model checkers (back ends). The only limitation of the proposed work is that, it unable to achieves the complete anonymity among the legitimate members. The identities are preserved from outside adversaries only.

The rest of this paper is organize as follows: Some existing works related to the proposed work are addressed in Section 2. The preliminaries related to the proposed work such as ID-based cryptosystem, Elliptic Curve Cryptography and security attributes are discussed in Section 3. Section 4 proposes the protocol while Section 5 discussed its security analysis. Section 6 shows the security validation result of AVISPA tool. Section 7 compares the performance of proposed protocol with others followed by a conclusion section.

2 Related Work

Protocols based on the traditional public key cryptography requires Public Key Infrastructure(PKI)to issue and manage the certificates for mapping the identity of an entity to their current public key. For Group Key Agreement (GKA) Algorithms this kind of mapping usually require some efficient PKI because in group key generation algorithms authentication of participants is also one of the major issue thus needs heavy certificate management by the PKI. A number of group key algorithms including [8, 14, 20] are exist in literature which often depends on PKI for the authentication of group members. Hawang *et al.* [14] introduces the Quad Key Tree structure and trying to reduces the hight of the key tree and thus reduces the number of rounds. It uses the pairing computations for further computation along with the modular exponentiations. [20] is also a tree based GK management protocol for multicast network but it uses the hybrid key tree technique for efficiency. It uses secure locking technique based on Chinese Remainder Theorem and shows the graphical result in their paper. Instead of tree based concept Hong proposes queue based group key agreement [8] and claim that it is most suitable for heterogeneous environment. In [8] each round performs Diffie-Hellman key exchange located on the opposite side of a blind key queue. Thus only the fast member are allowed to participate in the computation of next round and improves the efficiency. Filtration of fast and slow members are done by using a FIFO queue. Although [8] is suitable for heterogeneous group but it still needed $\lceil log_2^n \rceil$ rounds for n members and, the paper not considered the authentication issues. [15] proposes a polynomial-based key management for group scenario. But latter Kamal shows some security weakness in [15] by attacks in their paper [10].

Password based GKA protocols including [7] are avoids the requirement of PKI and uses the mutual authentication. Dutta and Barua proposes an authenticated GKA protocol on password based setting [7]. In [7] users needs to shares only a low quality human memorable password among themselves to agreed upon a high quality common secrete key. This protocol require constant round but $O(n)$ modular exponentiations. Since the exponentiation cost is relatively larger than the cost of scalar point multiplication over elliptic curve, so the performance of this protocol might be poor than the protocols based on the elliptic curve.

In order to overcome the PKI burden in 1984, Shamir [18] 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 [16]. It utilized a binary tree structure and requires log_2^n rounds for n numbers of users. Since then, many ID-based group key exchange protocols [3, 12, 24, 27] have been proposed and each have their own significance.

Wan *et al.* introduces the users anonymity in the ID based GKA protocol [24] for wireless networks. This enables a group of mobile users to establish a secret meeting session without disclosing that who are in the meeting to the outside eavesdroppers. [24] also provides the dynamic membership operations (join and leave) anonymously without leaking information on who is joining/leaving the group. Although it is a constant round protocol it employs the bilinear pairing in their computation which creates overheads for the mobile users. In wireless environment nodes should have less computational burden as much as possible in order to cope energy conservation. A bilinear pairing is a mathematical tool which maps two

elements in an elliptic curve group to an element in the related finite field, and is used commonly in building ID-AKA protocols and other security schemes [2, 19, 25]. However, since the bilinear pairing is always defined over a super singular elliptic curve group with large element size, the operation time for pairings is even longer than that of RSA private key operations, which makes pairings one of the most expensive known cryptographic operation [4]. Therefore, ID-based Authenticated Group Key agreement ID-AGKA protocols without pairing may be more appealing in practice. The significance of users anonymity in ID-based cryptosystem are also justified in [17]. Moreover in wireless environment the communication round time matters. For example, in the mobile IP registration, a one-round AKA protocol is wanted to reduce the message exchange time between a foreign domain and a home domain [4]. The present paper proposes an anonymous ID-based group key agreement protocol like [24] but free from the pairing computation with more efficient performance than same.

3 Preliminaries

The basic idea of ID based cryptosystem, Elliptic Curve Cryptography and some intractable problems are addressed in this section.

3.1 ID-based Cryptosystem

The concept of ID-Based Cryptography (IBC) was proposed by Shamir in 1984 [18] to remove the transmission, verification and maintenance of public key certificates. IBC employs a user's unique identifier, e.g., e-mail address, rather than a random number, as the user's public key, and the user's corresponding private key is generated based on the user's public key by the system's trusted authority. The system's trusted authority is unique and is the establisher of the ID-based cryptosystem. It is called PKG (Private Key Generator) or KGC (Key Generate Centre) depending on whether or not the final output generated by a user is known by the authority. In ID-AKA protocols, the session key is kept secret from the authority, and thus the authority is called KGC. KGC has a secret system master key s, and the user's long-term key (the user's private key) is generated using a definite function F:

$$Private\ Key = F(s, public\ key,\ Public\ parameters).$$

In IBC, the user's private key is given to the user via a secure out-of-band channel; it is in fact the user's implicit certificate. Although such implicit certificate is known only to the user and the KGC, its validity can be verified publicly, which enables IBC to remove the public key certificate.

3.2 Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) is an approach to public key cryptography based on the algebraic structure of elliptic curves over finite fields. The use of elliptic curves in cryptography was suggested independently by Koblitz and Miller in 1985.

In ECC non-singular type of Elliptic curves over the real number are used. The elliptic curve over real numbers takes the general form as:

$$y^2 = x^3 + ax + b.$$

In cryptography, variables and coefficients of elliptic curve equation are restricted to elements in a finite field. Thus for above equation x, y are co-ordinates of $GF(p)$, and a, b are integer modulo p, satisfying

$$4a^3 + 27b^2 \neq 0 \pmod{p}.$$

(for non singular elliptic curve).

Where p is a modular prime integer which make the the EC of finite field. An elliptic curve E over $GF(p)$ consist of points (x, y) defined by above two equations,along with an additional point called O (point at infinity or zero point) in EC forms a group. The O point plays the role of identity element for EC group.

Usually an elliptic curve is defined over two types of finite fields: the prime field F_p containing p elements (prime curve) and the characteristic 2 finite field containing 2^m elements (binary curve).This paper focuses on the prime finite field as the prime curve are best suit for software applications [21].

3.2.1 Elliptic Curve Arithmetic

Cryptographic schemes based on ECC rely on scalar multiplication of elliptic curve points. Given an integer k and a point $P \in E(F_p)$, scalar multiplication is the process of adding P to itself k times. The result of this scalar multiplication is denoted $k \times P$ or kP.

Point's addition and point doubling form the basis to calculate EC scalar multiplication efficiently using the addition rule together with the double-and-add algorithm or one of its variants. The detail description of ECC (including its point addition rule) can be found in various papers including [11, 26].

The security of ECC based protocols are based on intractability of Elliptic Curve Discrete Logarithm Problem (ECDLP). ECDLP state that: Given $P, Q \in E$, find an integer $k \in Z_p^*$ such that $Q = kP$. It is relatively easy to calculate Q given k and P, but it is relatively hard to determine k given Q and P.

4 Proposed Protocol

This section describes that initially how n numbers of members agreed up on a common session key under initialization operation followed by the the join and leave procedures.

Assumptions: The following assumptions has been considered in proposed protocol. Firstly, let U=$\{U_1, U_2, \cdots, U_n\}$be the set of mobile nodes. Secondly, each group at beginning must know the identity of others group members by some sort of other mechanism. Thirdly the protocol assumes a trusted server which is responsible for private key generation for the users, called key generation centre (KGC) in the system. The subscript notation for the participants are must be considers in logical ring fashion e.g. $U_{n+1} = U_1$ and $U_0 = U_n$ in entire paper.

4.1 Initialization

This subsection illustrates that how n members $U_1, \cdots, U_n$ can establish a group key to create a secure multicast session among them. The entire group key establishment process divided in two algorithms: Algorithm 1 and Algorithm 2. Algorithm 1 is run by KGC while Algorithm 2 is to be run by every user after completion of Algorithm 1.

On completion of Algorithm 1 every user got their long term private key $< S_i, R_i >$ though some secure channel. On receiving the same every user can validate it by checking whether the following equation hold:

$$R_i + H_1(ID_i).P_{pub} = S_i. \quad (1)$$

The private key is valid if the equation holds and vice versa. Since: $R_i + H_1(ID_i).P_{pub} = r_i.P + h_i.s.P = (r_i + s.h_i).P = S_i.P$.

On successful validation of their long term private key every user $U_i; 1 \leq i \leq n$ run the Algorithm 2 in parallel to agreed on a common session key SK. The session initiator (assuming U_1 in this paper) invoked the Algorithm 2 by setting the Role as the $INITIATOR$, on the other hand rest of the users invoked the Algorithm 2 as Role = $follower$. It is also assume that the initiator already knows the identities of other users and verified their authenticity.

The encryption technique used in Step 5 of Algorithm 2 is ID based and must be anonymous as similar in [24] and Sig_1 is calculated over the the respective message by U_1 by its private key. In Step 14 user U_i wait until the receiving of X_j; $j \neq i$ broadcasted from others from 13 of Algorithm 2. On receiving all X_j, U_i verify it in Step 15 by the following equation:

$$X_1 \oplus X_2 \oplus \cdots \oplus X_n = 0. \quad (2)$$

At the time of verification in Step 15 U_i take X_i from itself instead of broadcast channel e.g U_3 take the value of $X_1, X_2, X_4, X_5, \cdots X_n$ from broadcast channel while use their own calculated value of X_3 although the value of X_3 is also available in broadcast channel, so that if an active adversary intercept and modifies some or all of the X_i's in such a way that the altered value can also satisfies Equation (2) it is easily traceable by the U_i.

Finally U_i can calculate the value of K_j; $1 \leq j \leq n$ and $j \neq i$ by applying chain $XORing$ in Step 16, and 17 of Algorithm 2 started from K_{i+1} which is equivalent to the following calculations:

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

Algorithm 1 Key Generation Algorithm (KGC)

1: Begin
2: On taking $k \in Z^+$ as the input. KGC chooses a k -bit prime p and determines the following:
$\{F_p, E/F_p, G, P\}$.
where k is the security parameter.
F_p: a prime finite field.
E/F_p: an Elliptic curve over F_p.
G: Cyclic additive group formed by points on E/F_p with an extra point O called point at infinity.
i.e. $G = \{(x, y) \in E/F_p : x, y \in F_p\} \cup \{O\}$
P: Generator of G.
3: Choose a master private key $s \in_R Z_p^*$ and compute master public key $P_{pub} = s.P$.
4: Choose two cryptographic secure hash function:
$H_1 : \{0,1\}^* \rightarrow \{0,1\}^k$, $H_2 : G \times G \rightarrow \{0,1\}^k$
5: KGC publish the tuple $\{F_p, E/F_p, G, P, H_1, H_2, P_{pub}\}$ as the public parameters and secretly keeps the master private key s.
6: **for** Every User U_i having identity ID_i; $1 \leq i \leq n$ **do**
7: Calculates $h_i = H_1(ID_i)$
8: Choose $r_i \in_R Z_p^*$ and calculates:
$\begin{cases} S_i = (r_i + s.h_i) \bmod p, \\ R_i = r_i.P \end{cases}$
9: Send U_i's long term private key as $< S_i, R_i >$ to U_i by using a secure channel.
10: **end for**
11: End

Correctness:
The correctness of the initialization operation are rely on the following relations:

$$\begin{aligned} K_i^j &= K_j^i \\ K_i^{j\prime} &= K_j^{i\prime} \end{aligned}$$

for any value of $i, j; (1 \leq \{i, j\} \leq n)$ It can be proved as follows:

$$\begin{aligned} K_i^j &= (S_i.T_j + x_i(R_j + H_1(ID_j).P_{pub}) \\ &= (r_i + s.H_1(ID_i)).x_j.P + x_i(r_j.P + H_1(ID_j).s.P) \\ &= (r_i.P + s.P.H_1(ID_i)).x_j + x_i.P(r_j + H_1(ID_j).s) \\ &= (R_i + H_1(ID_i).P_{pub}).x_j + T_i.S_j \\ &= (S_j.T_i + x_j(R_i + H_1(ID_i).P_{pub}) \\ &= K_j^i. \end{aligned}$$

Algorithm 2 Group Key generation Algorithm (Role, U)

Begin
on validating their long term private key by Equation (1)
Pick $x_i \in_R Z_p^*$
Compute $T_i = x_i P$
if $Role = INITIATOR$ **then**
  Choose a pseudonym Nym_i for every user(including itself)
  Concatenates all identities followed by their corresponding pseudonym and encrypt entire message by the public key of every other user separately and broadcast to all.
  $INITIATOR \rightarrow *$:
  $E_{id}\{ID_1||\cdots||ID_n||Nym_1||\cdots||Nym_n||Sig_1\}$
end if
On receiving the encrypted broadcast from the Initiator verify the initiator signature.
on Successful verification in previous Step U_i does a series decryption trial using the private key.
If he is successfully decrypt one cipher text and find out his identity is in the ID list in Step then look for his Nym_i chosen by the Initiator.
U_i send the following message to its immediately backward and forward neighbour with their signature which can be verifies by their pseudonym instead identity.
$U_i \rightarrow U_{i-1}, U_{i+1} :< Nym_i, T_i, R_i, Sig_i >$
In similar way receives above message from U_{i-1} and U_{i+1} and verifies their signature by the pseudonyms Nym_{i-1} and Nym_{i+1} according to list obtained from Initiator.
On Successful verification in above Step U_i calculates the following:

$$\begin{cases} K_i^{i+1} = (S_i.T_{i+1} + x_i(R_{i+1} + H_1(ID_{i+1}).P_{pub}), \\ K_i^{i+1\prime} = x_i.T_{i+1}, \\ K_i^{i-1} = (S_i.T_{i-1} + x_i(R_{i-1} + H_1(ID_{i-1}).P_{pub}), \\ K_i^{i-1\prime} = x_i.T_{i-1}, \\ {K_i}^R = H_2(K_i^{i+1}, K_i^{i+1\prime}), \\ {K_i}^L = H_2(K_i^{i-1}, K_i^{i-1\prime}), \\ X_i = {K_i}^L \oplus {K_i}^R \end{cases}$$

Broadcast X_i with their pseudonym Nym_i to all users in the network
$U_i \rightarrow * :< Nym_i, X_i >$
User U_i wait until the reception of all X_j; $1 \le j \le n$ and $j \ne i$
if $X_1 \oplus X_2 \oplus \cdots \oplus X_n = 0$ **then**
  for $j = i+1$ *to* n and $j = 1$ *to* $i-1$ **do**
    ${K_j}^R = X_j \oplus {K_{j-1}}^R$
  end for
  $SK = H_1({K_1}^R||{K_2}^R||\cdots||{K_n}^R)$
  return SK
end if
return $ERROR$
End

Similarly $K_i^{j\prime} = K_j^{i\prime}$. From above relations it is easily seen that $K_i^R = K_{i+1}^L$.

4.2 Join Operation

In present paper join operation carried out by Single join (in Section 4.2.1)(for single request) as well as mass join procedures to handle multiple join requests simultaneously (Section 4.2.2).

4.2.1 Single Join

Let U_{n+1} (a new member) send the join request to U_1 (the initiator). If U_1, the initiator of the group meeting decided that the new member U_{n+1} to join the group meeting, It execute Algorithm 3 along with the U_n as the join controllers. It is assume that U_1 knows the identity of U_{n+1} in advance and U_{n+1} is already received its long term private key pair $< S_{n+1}, R_{n+1} >$ from KGC. U_1 first inform to U_n about the joining of U_{n+1}, because U_n also have to participates in join procedure along with U_1 and U_{n+1}. Single Join can be performed by Algorithm 3.

Algorithm 3 Single Join (U, U_1, U_n, U_{n+1})

Begin
U_1 select a non used pseudonym Nym_{n+1} for U_{n+1} and broadcast the following message to all previous members encrypted with current session key:
$U_1 \rightarrow *$: $E_{SK}\{ID_{n+1}||Nym_{n+1}||SIG_1\}$
U_1 also sends the necessary information about U_n and itself to U_{n+1} required for further calculations as follows:
$U_1 \rightarrow U_{n+1}$: $E_{ID}\{ID_1||Nym_1||ID_n||Nym_n||SIG_1\}$
U_{n+1} receives the message from U_1, then he decrypt the message using his private key to receives his pseudonym selected by U_1
U_1, U_n and U_{n+1} creates a separate group key K just for three members by using Algorithm 2
U_1 broadcast K to all other members encrypted with previous group key SK.
$U_1 \rightarrow *$:$E_{SK}\{Nym_1||K\}$
All members now can calculates new group session key as:
$SK_{new} = H_1(SK||K)$
U_1 sends new group session key to U_{n+1} encrypted with K
$U_1 \rightarrow U_{n+1}$: $E_K\{Nym_1||SK_{new}\}$
End

4.2.2 Mass Join

Mass join operation can be implemented as very similar to Single join operation. Suppose that members in set $U = \{U_1, U_2, \cdots, U_n\}$ have shared a common session key SK by using Algorithm 2 and then U_1 the initiator of the group decided that some users in set $C = \{U_{n+1}, U_{n+2}, \cdots, U_{n+n'}\}$ to join U. It is assume

that U_1 knows the identities of every member of Set C. The Algorithm 4 describes the procedure of mass join.

Algorithm 4 Mass Join

1: Begin
2: First of all U_1 chooses unused pseudonyms for new user set and concatenates it with their corresponding IDs encrypts entire message with the public keys of every users and broadcast to every users as in Step 5 of Algorithm2
$U_1 \rightarrow U_{n+i}$:
$E_{id}\{ID_{n+1}||\cdots ID_{n+n'}||Nym_{n+1}||\cdots||Nym_{n+n'}||Sig_1\}$
(for $i = 1$ *to* n')
3: U_1 also sends the joining information of new set along with their ID_s and pseudonyms to all current members encrypted with current session key:
$U_1 \rightarrow *$:
$E_{SK}\{ID_{n+1}||ID_{n+2}||\cdots||ID_{n+n'}||$
$Nym_{n+1}||Nym_{n+2}||\cdots||Nym_{n+n'}||Sig_1\}$
4: All new members now create a separate group key K along with U_1 and U_n by using Algorithm 2
5: All members of set U calculates the new group session key SK_{new} as in single join operation:
$SK_{new} = H_1(SK||K)$
6: U_1 broadcast new session key to all the members of set C (new members) encrypted with K
7: End

4.3 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 [27]. Suppose $U = \{U_1, U_2 \cdots, U_n\}$ be the current group and $L = \{U_{l1}, U_{l2}, \cdots, U_{ln'}\}$ is the set of leaving members, where $\{l1, l2, \cdots, ln'\} \subseteq \{1, 2, \cdots, n\}$ and $n' < n$.We represent the set of remaining members as $A = \{U_{a1}, U_{a2}, \cdots, U_{a(n-n')}\} = U - L$. The leave operation can be carried out by Algorithm 5.

Algorithm 5 Leave Operation(U, L, A)

1: Begin
2: U_1 first broadcast set of pseudonyms Nym_i; $i \in L$ corresponds to the leaving members in U.
3: On completion of previous Step U_i; $i \in A$ know about the set L.
4: **for** Each $U_i \in A$ **do**
5: **if** $(U_{i-1} \in L)$ OR$(U_{i+1} \in L)$ **then**
6: updates their random secret x_i and accordingly recalculates their K_i^R and K_i^L with the contribution of its neighbours (left and right) alive members.
7: Finally U_i calculates $X_{newi} = K_i^L \oplus K_i^R$ and broadcast to A
8: **end if**
9: **if** $(U_{i-1} \notin L)$ AND $(U_{i+1} \notin L)$ AND $(U_{i+2} \in L)$ **then**
10: U_i recompute their K_i^R accordingly but no need to recalculate K_i^L
11: Calculate the value of X_{newi} with the contribution of newly calculated K_i^R of previous step and broad cast it in set A.
12: **end if**
13: All other members U_i;$((U_{i-1} \notin L)$ AND$(U_{i+1} \notin L)$ AND $(U_{i+2} \notin L))$ do nothing but set their $X_{newi} = X_i$ and broadcast in set A.
14: **end for**
15: Each member $U_i \in A$, after receiving all X_{newj} $(j \neq i)$ first verifies
$X_{newa1} \oplus X_{newa2} \oplus \cdots \oplus X_{newa(n-n)} = 0$
16: If above verification is success then U_i can calculates only require value of K_j^R (the updated one); $j \neq i$ by chain XOR*ing* operation as in Algorithm 2.
17: Finally the new session key calculated as:
$SK_{new} = H_1(K_{a1}^R||K_{a2}^R||\cdots||K_{a(n-n')}^R)$
18: End

5 Security Analysis

The security attributes for the proposed protocols are analyze in this section and also discussed its privacy issues. As discussed in [9], a secure authenticated group key agreement protocol should satisfies the requirements of contributiveness, message integrity, resilience against passive attack and forward/backward security for the joining/leaving operation. If an scheme is contributory, it also provides resilience against other relevant known attacks such as known key attack, key compromise impersonation attack, known session specific temporary information attack, impersonation attack, etc, as described in [9]. The security of group session key in proposed protocol relies on difficulties of ECDLP and CDHP.

Contributiveness and Group Key Secrecy: An authenticated group key agreement protocol is said to be contributory group key agreement protocol if each and every member in the group contributes in the formation of group session key. In proposed protocol each member U_i sends its T_i and R_i to its neighbour (U_{i-1}, U_{i+1}) where T_i is computed with its random secrete x_i and R_i is one of the private value received from KGC. In this way U_i agreed on two common secrets separately with its neighbours (U_{i-1} and U_{i+1}) as: $x_i.T_{i+1} = x_{i+1}.T_i$ (between U_i and U_{i+1}) and $x_i.T_{i-1} = x_{i-1}.T_i$ (between U_i and U_{i-1}) then U_i calculates ${K_i}^R$ and ${K_i}^L$ with the contribution of U_{i+1} and U_{i-1} respectively. The final group session key is computed with the help of all ${K_j}^R$ $(j = 1$ *to* $n)$ as discussed in proposed protocol of Section 4. Thus the group session key is computed by each user's ephemeral and long-term private key so the proposed protocol is contributory. In the group of n members $\{U_1, U_2, \cdots, U_n\}$, to compute ${K_j}^R$,$j = 1$ *to* n for any user U_i should know the ${K_{j-1}}^R$ and to calculate the ${K_{j-1}}^R$ they should know

the value of ${K_{j-2}}^R$ and so on this way to calculate all value ${K_j}^R$; $j = 1$ *to* n. U_i should have at least one value of ${K_j}^R$, $j \in \{1, 2, \cdots, n\}$ this is possible if and only if $U_i \in \{U_1, U_2, \cdots U_n\}$ i.e. $i \in \{1, 2, .., n\}$ means U_i is a valid group member. U_i only know the value of ${K_i}^R$ and ${K_{i-1}}^R$(since ${K_i}^L = {K_{i-1}}^R$) in advanced.

Message Integrity: In proposed protocol first every user receives pseudonyms Nym_i of every member selected from initiator member which is encrypted by an anonymous ID-based encryption scheme with their public keys and signed by initiator with a powerful signature scheme. After verifying the signature and decrypting the message every members knows the identity and their corresponding pseudonyms but an adversary cannot. All further communication between the user are done with their pseudonyms Nym_i, the receiver of the message first verifies the currently received pseudonym according to the pseudonym list in first decrypted message from initiator if the verification is successful he conclude that message is received from the expected member. Since the group member's identity is protected from outside eavesdropper, the adversary not able to know the actual communicating party. In similar way before calculating the group session key each user U_i first verifies the all pseudonyms received along with their X_j $(j \neq i)$ from others. If this is successful U_i again checks whether $X_1 \oplus X_2 \oplus \cdots \oplus X_n = 0$ hold. This is hold because X_i are calculated as $X_i = {K_i}^L \oplus {K_i}^R$ and ${K_i}^R = {K_{i+1}}^L$ this is proved in Section 4. So

$$\begin{aligned} & X_1 \oplus X_2 \oplus \cdots \oplus X_n \\ = \; & {K_1}^R \oplus {K_1}^L \oplus {K_2}^R \oplus {K_2}^L \cdots \oplus {K_n}^R \oplus {K_n}^L \\ = \; & 0. \end{aligned}$$

(note that subscript notation considered as in circular fashion i.e. $n + 1 = 1$ and $0 = n$ thus ${K_1}^L = {K_n}^R$). U_i simply abort in case of any of the above checks will fail.

No Passive Attack: The proposed protocol is secure against the passive attack under the assumption of ECDLP. That is an attacker is unable to obtain the resulting group session key by using the eavesdropping messages $(T_i, R_i, X_i)(1 \leq i \leq n)$ transmitted over the insecure network. As discussed in [9] an Authenticated group key agreement protocol is secure against the passive attack if the protocol is executed in presence of an adversary, but he cannot get success to obtain to established group session key from the eavesdropped messages exchanged between the participants. Assume that an attacker sniffing the communication channel and captures the messages (Nym_i, T_i, R_i); $(1 \leq i \leq n)$ and (Nym_i, X_i); $(1 \leq i \leq n)$ in the current session and tries to generates the group session key $K = H_1({K_1}^R || {K_2}^R || {K_3}^R || \cdots || {K_n}^R)$ of that session. Attacker is unable to do that because he cannot calculate any of the ${K_i}^R$ or ${K_i}^L (1 \leq i \leq n)$ without the knowledge of S_i and x_i. It is clear that to calculate all value of ${K_i}^R (1 \leq i \leq n)$ one should know at least one value of ${K_i}^R$ or ${K_i}^L$ along with the all other values of X_i. To calculate ${K_i}^R$ and/or ${K_i}^L$ (for any $i \leq n$) one should have to calculates the value of $K_i^{i+1} = (S_i.T_{i+1} + x_i(R_{i+1} + H_1(ID_{i+1}).P_{pub})$ and $K_i^{i+1'} = x_i.T_{i+1}$ or $K_i^{i-1} = (S_i.T_{i-1} + x_i(R_{i-1} + H_1(ID_{i-1}).P_{pub})$ and $K_i^{i-1'} = x_i.T_{i-1}$. This is not possible without the knowledge of long term private key S_i and random secret value x_i of any legitimate user U_i due to the difficulties of ECDLP and CDHP.

Forward Secrecy: The meaning of forward secrecy in any group key agreement protocol is that, on the event of leave operation the current group session key must be updated in such a way that the leaving member(s) cannot compute or trace it and then not able to access the further conversations. The proposed protocol provides the forward secrecy because even a single member is leaving but the contribution of three consecutive members is totally changed in the formation of new group key. Since this change happens due to the updating of random secret value x_i of two members U_{i-1} and U_{i+1} where U_i is the leaving member, U_i cannot trace the new contributions of members because this time the value of T_{i-1} and T_{i+1} is changed. This is achieved by leave operation of the protocol discussed in Section 4.3.

Backward Secrecy: The backward secrecy of a group key agreement protocol allows the new member(s) to join in a group and develop new group key without providing the scope for generating any previous group session key to the new members so that they cannot access the previous group conversations. The proposed protocol provides backward secrecy as the new member U_{n+1} not able to calculates previous group session key SK because it receives only the hash value of SK concatenated with K. To calculate K, new member U_{n+1} receives the new shares from U_1 and U_n which is independent from their previous contributions in SK. Same thing happens in mass join operation.

Perfect Forward Secrecy: Perfect forward secrecy represents security in case of long-term secretes compromise. In proposed protocol, perfect forward secrecy is achieved from hardness of ECDHP problem. Even if the long term secrets$\{S_i, R_i\}$ is compromised by the adversary, without the ephemeral secret x_i the adversary cannot compute ${K_i}^L$ or ${K_i}^R$ so he cannot extract the other user's ephemeral values, ${K_j}^L$ or ${K_j}^R$ and he cannot compute the session key.

No Key Control: In proposed protocol the group session key is created jointly by all legitimate group members (contributiveness is already discussed previously). So no individual member can control the key alone.

Known Session Key Security: In each session, each user U_i randomly chooses an ephemeral private key $x_i \in Z_p^*$ and the generated group session key depends on each user's ephemeral private key x_i. The adversary that compromises one session key should not compromise other session keys, so this protocol can provide known session key security.

Ephemeral Private Key Revealing Resistance: If all users ephemeral $(x_i, i = 1, 2, \cdots, n)$, have been com-

promised, 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.

Besides above security attributes, this proposed protocol is also secure in the presence of at most $(n-1)$ users controlled by the adversary without their long-term private keys. The adversary may extract the ephemeral value x_i of a user, U_i but he cannot compute the session key without any user's long-term private key.

Anonymity: The proposed protocol employs the concept of anonymity as like in [24]. In this protocol in every message exchanges the identities of the members are either encrypted so that no identity-related information is leaked or the users are identified by their pseudonyms from which impossible to infer any information by the adversary, since only the legitimate group members knows the valid Nym_i, ID_i pairs. In the first message by U_1 the identities of users and their corresponding pseudonyms are encrypted with their public key by using ID-based encryption and this encryption scheme require to be anonymous so that it is impossible to obtain any information from only the cipher text. Nym_i is selected by U_1 and obtained by U_i by decrypting that message; itself does not leak information on its identity. Since an adversary knows all these Nym_i, he may want to guess the user's identities and verifies his guess by first message. However, it is impossible to do that as the protocol uses an anonymous encryption scheme.

Unlinkability: Anonymity would be meaningless without unlikability [24]. The adversary can still trace an unknown user without knowing his real identity given only anonymity. In proposed protocols, including joining/leaving operations, different pseudonyms are uses for every user on each independent execution of the protocol. A pseudonym is never reused and cannot be used to link two different execution of the protocol.

6 Formal Security Verification Using AVISPA Tool

Recently, AVISPA tool [23] 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 [5], where the network is controlled by an intruder (Active and passive); however he is not allowed to crack 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 back-ends,namely OFMC (On-the-fly Model-Checker), CL-AtSe (Constraint-Logic-based Attack Searcher), SATMC (SAT-based Model-Checker) and TA4SP (Tree Automata-based Protocol Analyzer). The details description about AVISPA and HLPSL can be found in [1].

The initialization operation of proposed protocol is specified in HLPSL and verified using online AVISPA tool which shows that protocol is safe under different attacks. Role specification of KGC and user1 are illustrated in Figures 1 and 2, respectively. The role of other users are almost similar than that of user1. While the result under OFMC and CL-AtSe back ends are Shown in Figures 3 and 4, respectively.

Figure 1: Role specification of KGC in HLPSL

```
role kgc(Kc : agent,P,S,ID1,ID2,ID3 : text,K1,K2,K3 : public_Key,
H,H1,H2,M,A : 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 : text
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':= A(Rr1',M(S,H(ID1))) /\ S2':= A(Rr2',M(S,H(ID2)))
/\ S3':= A(Rr3',M(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
```

7 Performance Comparison

This section compares the performance of proposed protocol with some other existing ID-based GKA protocols [3, 6, 22, 24, 27] in terms of communication and computation costs. The result is showed in Table1 (where n is the number of users. The following notations are used for comparison.

- **PM:** number of Scalar point multiplications.
- **PA:** Number of elliptic curve point additions.
- **Message:** Total number of message overheads during group key generation process (including unicast and broadcast).
- n**:** number of participants.
- n'**:** number of joining or leaving participants.
- **Pairings:** number of bilinear pairing computations needed in key agreement process (zero in case of our proposal).

[3, 6, 22] protocols are not dynamic (Join and Leave procedures are not exist) so only the initialization cost are tabulated in Table 1 and it is taken from their respective papers. For Xie Liyun protocol [27] the cost of

Figure 2: Role specification of User1 in HLPSL

```
role user1(U1: agent, K1:public_key,ID1,ID2,ID3 :text,
 H1,H,H2,M,A: hash_func,P: text, SND,RCV : channel(dy))

played_by U1 def=

local
State : nat,
U2,U3 : agent,
K2,K3 : public_key,
T3,T2,T1,S1,R1,R2,R3,X1,K12,K13,K1r,K2r,K3r,
EX1,EX2,EX3,K1R,K1L,Ppub : text,
SK: symmetric_key,
IDRing: (agent.text)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) =|>

        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()/\ T1':= M(P,X1') /\
        SND({U1.ID1.T1'.R1'}_inv(K1))

        /\witness(U1,U2,u2_u1_t,T1'.R1')
        /\witness(U1,U3,u3_u1_t,T1'.R1')

2.State=1
/\ RCV( {U2.ID2.T2'.R2'}_inv(K2')) /\ in(U2.ID2,IDRing)
/\RCV( {U3.ID3.T3'.R3'}_inv(K3')) /\ in(U3.ID3,IDRing)=|>
State':=2/\
request(U1,U2,u1_u2_t,T2'.R2')/\
request(U1,U3,u1_u3_t,T3'.R3')/\

K12':= A(M(T2',S1),M(A(R2',M(Ppub,H(ID2))),X1))/\
K13':= A(M(T3',S1),M(A(R3',M(Ppub,H(ID2))),X1))/\

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
```

Figure 3: Simulation result on OFMC back end

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/avispa/web-interface-computation/./tempdir/workfile0kIMQW.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.65s
  visitedNodes: 16 nodes
  depth: 4 plies
```

Figure 4: Simulation result on CL-AtSe back end

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/avispa/web-interface-computation/./tempdir/workfile0kIMQW.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

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

Initialization are taken from the tabulated value of [27]. 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 [24] are described for single member join/leave in their paper. For comparison, the unit cost is multiplied by n' and tabulated in Table 1. Cost of Leave operation of present paper as well as [27] 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 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.

8 Conclusion

This paper proposes an anonymous pairing-free ID-based Group key agreement protocol based on the Elliptic Curve computational DiffieHellman problem. The protocol provides strong security protection including Ephemeral Private Key Revealing Resistance, forward and backward secrecy, Perfect Forward Security,etc. This is the first protocol which incorporates the user's anonymity without using the bilinear pairings. The protocol also provides efficient join and leave procedures for dynamic operations. All such operations accomplished anonymously without leaking the information on who is joining/leaving the group. In addition of security analysis phase, security of proposed protocol is also verified by the AVISPA tool which outputs safe under its different back ends. Finally the performance of the proposed technique is compared with some other existing protocols which shows that it has comparable communication and computation cost with zero pairing computation. 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.

Table 1: Comparison table

Protocol	Group Operation	PM	PA	Pairings	Message
Choi's Protocol [3]	**Initialization**	$3n$	n	$2n$	$2n$
Du's Protocol[6]	**Initialization**	$5n$	n^2+2n	$2n$	$2n$
Tang's protocol [22]	**Initialization**	$5n$	n	$3n$	$2n$
XIE Liyun protocol [27]	**Initialization**	n^2+3n	n^2	0	$2n$
	Join	$(n+n')^2+5n'+7$	$(n+n')^2+n'+2$	0	$2n'+3$
Wan *et al.* Protocol [24]	**Initialization**	$3n$	0	$2n$	$4n$
	Join	$(n*n')$	0	$2(1+n')$	$7n'$
	Leave	$6n'$	0	$2n'$	$7n'$
Proposed protocol	**Initialization**	$9n$	$4n$	0	$5n-1$
	Join	$9(n'+2)$	$4(n'+2)$	0	$5(n'+2)+2$
	Leave	$9(n-n')$	$2(n-n')$	0	$(n-n')+2$

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] A. Armando, D. Basin, Y. Boichut, et al., "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)*, LNCS 3576, pp. 281–285, Springer, 2005.

[2] S. Chang, D. S. Wong, Yi Mu, and Z. Zhang, "Certificateless threshold ring signature," *Information Sciences*, vol. 179, no. 20, pp. 3685–3696, 2009.

[3] K. Y. Choi, J. Y. Hwang, and D. H. Lee, "Efficient id-based group key agreement with bilinear maps," in *Public Key Cryptography (PKC'04)*, pp. 130–144, Springer, 2004.

[4] L. Dang, W. Kou, N. Dang, H. Li, B. Zhao, and K. Fan, "Mobile ip registration in certificateless public key infrastructure.," *IET Information Security*, vol. 1, no. 4, pp. 167–173, 2007.

[5] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[6] X. Du, Y. Wang, J. Ge, and Y. Wang, "An improved id-based authenticated group key agreement scheme," *Cryptology ePrint Archive*, Report 2003/260, 17 Dec 2003.

[7] R. Dutta and R. Barua, "Password-based encrypted group key agreement.," *International Journal of Network Security*, vol. 3, no. 1, pp. 23–34, 2006.

[8] S. Hong, "Queue-based group key agreement protocol.," *International Journal of Network Security*, vol. 9, no. 2, pp. 135–142, 2009.

[9] S. H. Islam and G. P. Biswas, "A pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile networks," *Annals of Telecommunications*, vol. 67, no. 11-12, pp. 547–558, 2012.

[10] A. A. Kamal, "Cryptanalysis of a polynomial-based key management scheme for secure group communication.," *International Journal of Network Security*, vol. 15, no. 1, pp. 68–70, 2013.

[11] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, pp. 203–209, Jan. 1987.

[12] E. Konstantinou, "An efficient constant round id-based group key agreement protocol for ad hoc networks," in *Network and System Security*, LNCS 7873, pp. 563–574, Springer, 2013.

[13] W. T. Li, C. H. Ling, and M. S. Hwang, "Group rekeying in wireless sensor networks: A survey," *International Journal of Network Security*, vol. 16, no. 6, pp. 401–410, 2014.

[14] T. C. Lin, Te-Yu Chen, C. S. Gau, and M. S. Hwang, "A key agreement for large group using bilinear maps," *Journal of Theoretical and Applied Information Technology*, vol. 49, no. 2, pp. 871–878, 2013.

[15] Y. Piao, J. Kim, U. Tariq, and M. Hong, "Polynomial-based key management for secure intra-group and inter-group communication," *Computers & Mathematics with Applications*, vol. 65, no. 9, pp. 1300–1309, 2013.

[16] K. C. Reddy and D. Nalla, "Identity based authenticated group key agreement protocol," in *Progress in Cryptology (INDOCRYPT'02)*, LNCS 2551, pp. 215–233, Springer, 2002.

[17] Y. Ren, Z. Niu, and X. Zhang, "Fully anonymous identity-based broadcast encryption without random oracles," *International Journal of Network Security*, vol. 16, no. 4, pp. 256–264, 2014.

[18] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in cryptology*, LNCS 196, pp. 47–53, Springer, 1985.

[19] Z. Shao, "Certificate-based verifiably encrypted signatures from pairings," *Information Sciences*, vol. 178, no. 10, pp. 2360–2373, 2008.

[20] R. Srinivasan, V. Vaidehi, R. Rajaraman, S. Kanagaraj, R. C. Kalimuthu, and R. Dharmaraj, "Secure group key management scheme for multicast networks," *International Journal of Network Security*, vol. 10, no. 3, pp. 205–209, 2010.

[21] W. Stallings, *Cryptography and Network Security: Principles and Practice*, Pearson Education, 3rd edition, 2002.

[22] H. Tang, L. Zhu, and Z. Zhang, "Efficient id-based two round authenticated group key agreement protocol," in *IEEE 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'08)*, pp. 1–4, 2008.

[23] L. Vigan, "Automated security protocol analysis with the avispa tool," in *Proceedings of the 21st Annual Conference on Mathematical Foundations of Programming Semantics (MFPS05)*, pp. 61–86, 2006.

[24] Z. Wan, K. Ren, W. Lou, and B. Preneel, "Anonymous id-based group key agreement for wireless networks," in *IEEE Wireless Communications and Networking Conference (WCNC'08)*, pp. 2615–2620, Mar. 2008.

[25] S. Wang, Z. Cao, K. K. R. Choo, and L. Wang, "An improved identity-based key agreement protocol and its security proof," *Information Sciences*, vol. 179, no. 3, pp. 307–318, 2009.

[26] Y. Wang, B. Ramamurthy, and X. Zou, "The performance of elliptic curve based group diffie-hellman protocols for secure group communication over ad hoc networks," in *IEEE International Conference on Communications (ICC'06)*, vol. 5, pp. 2243–2248, 2006.

[27] L. Xie and M. He, "A dynamic id-based authenticated group key exchange protocol without pairings," *Wuhan University Journal of Natural Sciences*, vol. 15, no. 3, pp. 255–260, 2010.

Abhimanyu Kumar completed his B.Sc. (Engg.) degree in Computer Science and Engineering from R. P. Sharma Institute of Technology, Patna (affiliated to Magadh University, Bodh Gaya, Bihar) in 2011. Currenty he is pursuing Ph.D. under supervision of Dr. Sachin Tripathi at Indian School of Mines, Dhanbad, 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.