Lagrange interpolation based Asymmetric Group Key Agreement Cryptosystem

Dr. R. Siva Ranjani
Associate Professor, GMRIT, Rajam
(Email: sivaranjini.r@gmrit.org)

***Abstract*-This paper is proposing a new cryptosystem termed as Asymmetric group key agreement cryptosystem, where a selected set of group users broadcast their contribution to other group members by keeping their own information secret. Each group user collects broadcast message from other group participants and derive a common group key. If any user is interested to send message to group, he decides the session key and then encrypts message using the session key. The decided session key is encrypted using the derived common group key and securely communicated to other group members. Group members will use their secret information to retrieve the session key, which is used to decrypt the received message. Also, the proposed techniques performance is compared against X.Du et al.[26], L.LU et al. [27] and J.Beak[28] protocols. Observed that the proposed technique is taking less computation time than other existing techniques.**

Keywords— Identity based, pairings, public key cryptography, Group key agreement.

I. INTRODUCTION

Now, a day's group oriented communications are playing a vital role in the organizing a video conference and group chatting. In, all these applications a set of users were identified for having a communications among them. Here, information security plays a vital role in securing the information which was shared among the group members. So, many algorithms were proposed to secure the data in group communication. Group Key Agreement (GKA) Protocols are such protocols, which allow a group of users to derive a common secret key, from which a session key can be inferred. In these protocols, all the group applicants require secure broadcasting at the network layer among the members for secure group communication. In conventional group key agreement, all the users in the group establish a common shared secret key, which is used in message encryption and decryption. In the recently developed asymmetric key agreement protocol by Wu et al. [22], all the group participants negotiate a common encryption key which is accessible to all including non group members, unlike the regular GKA. Each group participant holds his own contribution, which is used in his secret decryption key derivation. Therefore, beside the group participants, Asymmetric Group Key Agreement Protocols (AGKAP)[10,11,12] allows outsiders of the group to broadcast the cipher messages to the group participants, provided that the sender knows the negotiated public key. Because of single round efficiency, AGKAP is more preferable when compared to conventional GKA protocol. Many of the conventional GKA protocols are having more than one round for sharing a common secret key. Also, all the participants should be active while sharing the secret key. However, if the participants are located in different regions with different time zones, it is very difficult for them to be connected concurrently. But, in ASGKA, group participant need not be alive during the key sharing. In single round ASGKA, each participant has to publish their public key contribution by holding their respective secret key. During the message encryption, sender encrypts the message by using a common public key which is derived from the all group users' public key information. For decryption, participant uses his secret key.

First , Diffie and Hellman [1] proposed a solution to key agreement; later Joux [7] extended the key agreement to three parties. Many authors attempted to extend the Diffie-Hellman and Joux protocols to n participants. Burmester- Desmedt [16] protocol succeeded in extending the key agreement protocol with two rounds and irrespective on participants' count. For key agreement protocols in open networks, communication should be secure against active adversaries. But, Diffie- Hellman, Joux and Burmester- Desmedt protocols do not authenticate the communicating entities.

To add authentication, several protocols have been proposed among them, the GKA protocol [15] is based on IBPKC, which refers to Katz and Yung's result [2][4] for an authenticated version. Bresson et al. [17] formalized the first security model for group key agreement protocol, extending the group key agreement between two or three parties. Subsequently, the model was refined and modified by Dutta and Barua[3], Bresson et al. [18, 19]. Later Lei et al. [20] extended these models to define the security of IB-AAGKA protocol, later it was extended to broadcast encryption application for open networks [21]. Sivaranjini et al. [23,24,25] proposed a protocol on asymmetric group key agreement protocol. In this paper, we extended these proposed models to define an identity based asymmetric asynchronous group key agreement protocol.

In this paper, a security model for identity based authenticated asymmetric group key agreement protocol is developed. Our protocol is based on the identity based batch multi signature with batch verification [8][13][14] to generate identity based signature. Furthermore, participant identity is used in the derivation of broadcast message computations. The proposed protocol is like an authenticated group key agreement protocol with following features:

- Permits the group having any number of members without compromising the security.
- Facilitates the mutual authentication between the Group Controller and members in the group.
- Performance is compared with existing protocols.
- Allows users to broadcast public information by concealing private information. A Common group key is inferred from public information, which is received from other group members.

chapter 2, reviews of Bilinear maps and some complexity assumptions were discussed. Section 3 defines the proposed protocol block diagram, detailed description of the proposed algorithm in chapter 4, chapter 5 discusses whole about the performance evaluation and finally we concluded the work in section 6.

II. PRELIMINARIES

In this section, we put forward the definitions that we used in the discussion of the forth coming sections.

Bilinear maps: We review the basic notations of the bilinear maps [5, 6] under our proposal, Let $(G_1,+)$ and $(G_T,*)$ be two groups of prime order $q > 2k$ for a security parameter $k \in N$. A function $e: G_1 X G_1 \rightarrow G_T$ is said to be a bilinear map if it satisfies the following properties:

1. Bilinearity: $e(aP, bQ) = e(P,Q)^{ab}$, $\forall P,Q \in G_1$; $a, b \in Z$
   - $e(P + Q,R) = e(P,R)*e(Q,R)$ and (1)
   - $e(P,Q + R) = e(P,Q)*e(P,R)$ $\forall P, Q, R \in G_1$ (2)
2. Non-degeneracy: $e(P,Q) = 1$; iff $P = 1$
3. Computability: There exists a polynomial time algorithm to compute $e(P, Q)$, $\forall P,Q \in G_1$.

A bilinear map is defined as a probabilistic polynomial time algorithm (E) that takes a security parameter k and returns a uniformly random tuple (G_1, G_T, e, g, q) of bilinear parameters, where g is the generator of G_1 and e is the bilinear map.

III. PROPOSED ALGORITHM: BLOCK DIAGRAM

Proposed scheme has adopted bilinear pairings on cyclic groups. Let G_1 be an additive group and G_2 be a multiplicative group, denoted by $(G_1, +)$ and $(G_2, .)$ where both of them are cyclic and each of them is with a prime order q. Each group participant is armed with an identity and a trusted authority Group Controller (GC) is employed to issue private key to group members. After that, each user chooses a random number, and individually computes a random message to all group users, keeping his own message secret, and multicast other users messages along with the broadcast message which consists of user ID and the signature for verification. Each group user collects the broadcast and multicast messages from other participants and then adds his secret message to derive the session public and private key pairs. With this key pair, we can share a symmetric key, which is used to secure the message from unauthorized persons.

By the end of this stage, each user in the group is having a common public key and own private key. When any user wants to send any message they can encrypt using the public key, receiver decrypts it using his private key, confidentiality is provided with this approach. With this methodology, only authorized group users are allowed to decrypt the ciphertext. But, in the proposed method, a double encryption technique is used to increase the security to symmetric key, which is used to encrypt the message.

According to double encryption methodology two keys are required, one key pair is derived from above methodology, and another key is chosen then distributed by sender using Lagrange polynomial function. To achieve this, first sender decides the t (for $1 \leq t \leq n$) receivers (if t=n then sender may use broadcast approach otherwise use multicast approach), and prepares (t_1, y_1), (t_2, y_2),-------,(t_t, y_t), for them. For every receiver i, sender sets (x_i) as the root of $F_i(x) = y_i$, where the receivers identity is mapped into $t_i \in Z_q^*$ using one way hash function (H), then computes $y_i = y*Q_i$ as the personal private key of the other receiver. Where y_i is the random number chosen in Z_q^*.

The polynomial $f_i(x) = \frac{F_i(x)}{y_i}$

$$= \prod_{1 \leq j \neq i \leq t} \frac{x - t_j}{t_i - t_j}$$

$$= \begin{Bmatrix} 1 \text{ if } x = t_i \\ 0 \text{ if } x \in \{t_1, t_2 - - - -, t_t\} - \{t_i\} \end{Bmatrix}$$ is used to achieve user anonymity.

In the encryption phase, sender computes a parameter R_i for all the receivers i by using above polynomial f(x). The sender computes all the R_i's, and other parameters, double encrypted message (M) to form

the cipher text. To decrypt the cipher text, receiver decrypts cipher text using his group private key, extract all R_i′s and his x_i to compute $\lambda = f_i(x_i)$ which is y_i. Then the receiver computes the secret key using user ID_i, S_i, d_i and λ. Finally, σ is used to decrypt the ciphertext. The arrangement of the proposed technique is shown in the Figure 1:

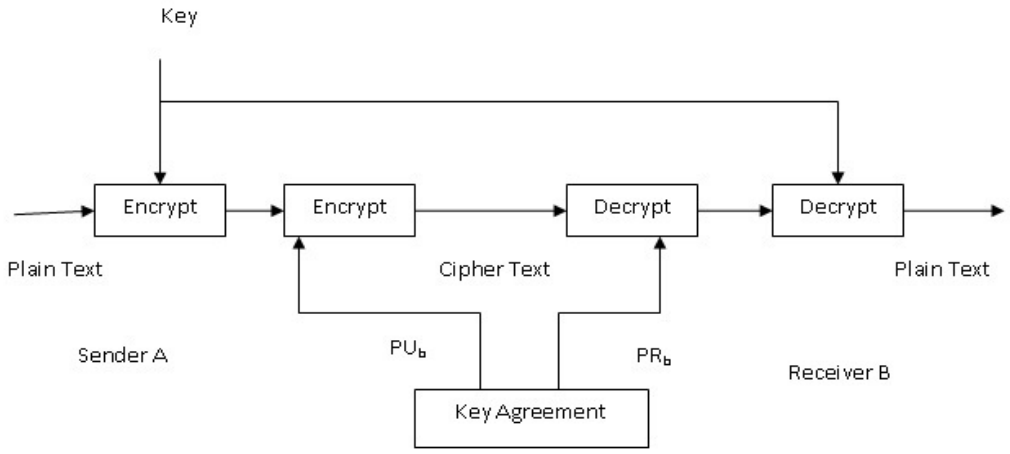

Figure 1: Identity based double key message encryption

In group communication, sender should take more care before communicating the message to other group members. Entire encryption process, which is employed on the message (M) by the sender, is shown in Figure 2. Before commencing the message encryption, sender has to decide the receivers, with whom he wants to send the message. Afterwards, a group is formulated with all the receivers and along with the sender. Later, group participants will decide a common public key (X, A), which is used for encrypting the message in group communication. Also maintains an individual private key (d_i). The public and private key pairs are used to encrypt the message.

In encryption process, initially, sender randomly choose a random symmetric key (σ), then used it for encrypting M, results encrypted message Y. We can write this as

$$Y = E_1(\sigma, M) \quad (1)$$

Again, message Y is encrypted using the group public key (X, A) to find the equivalent cipher text (C_1, C_2, C_3), can be written as

$(C_1, C_2, C_3) = E_2(PU_b, Y)$, where $PU_b = (X,A)$ (2)

Sender, sends the cipher text (C_1, C_2, C_3), encrypted symmetric key (W) along with some extra parameter (R_1, R_2, ----, R_t, U & V) to all the receivers.

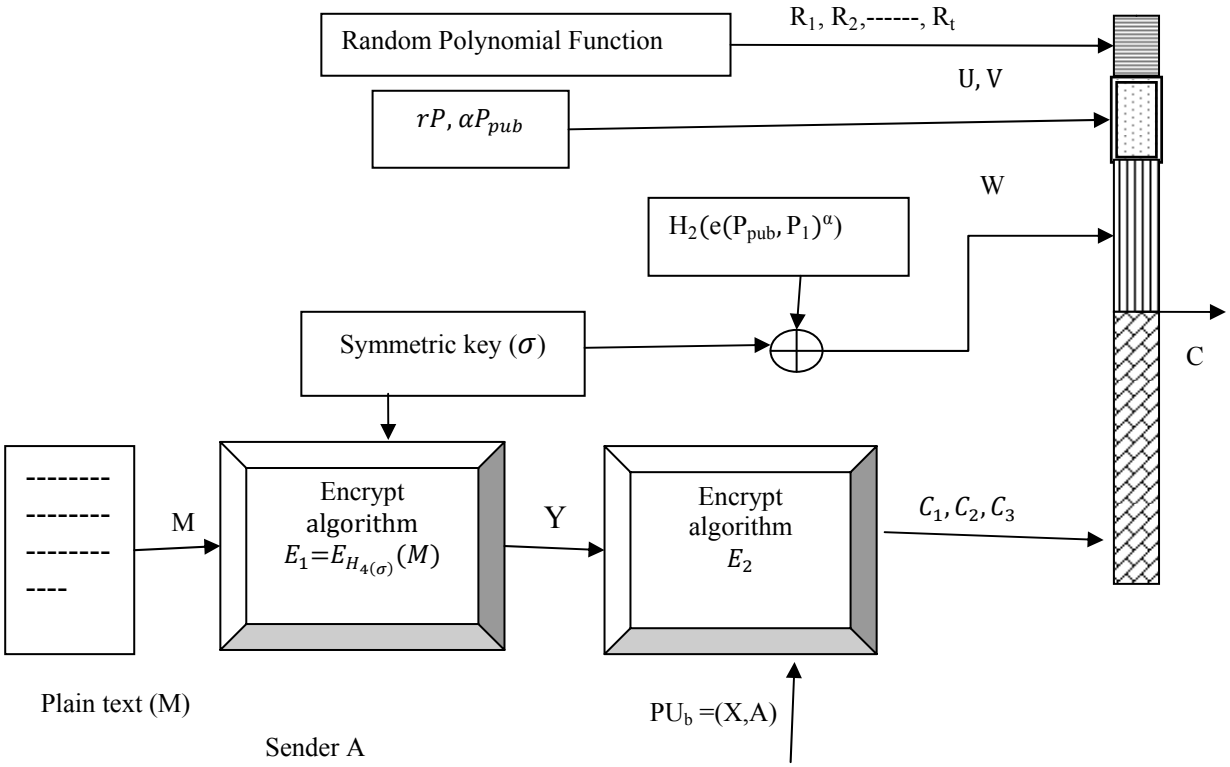

Figure 2: Message Encryption

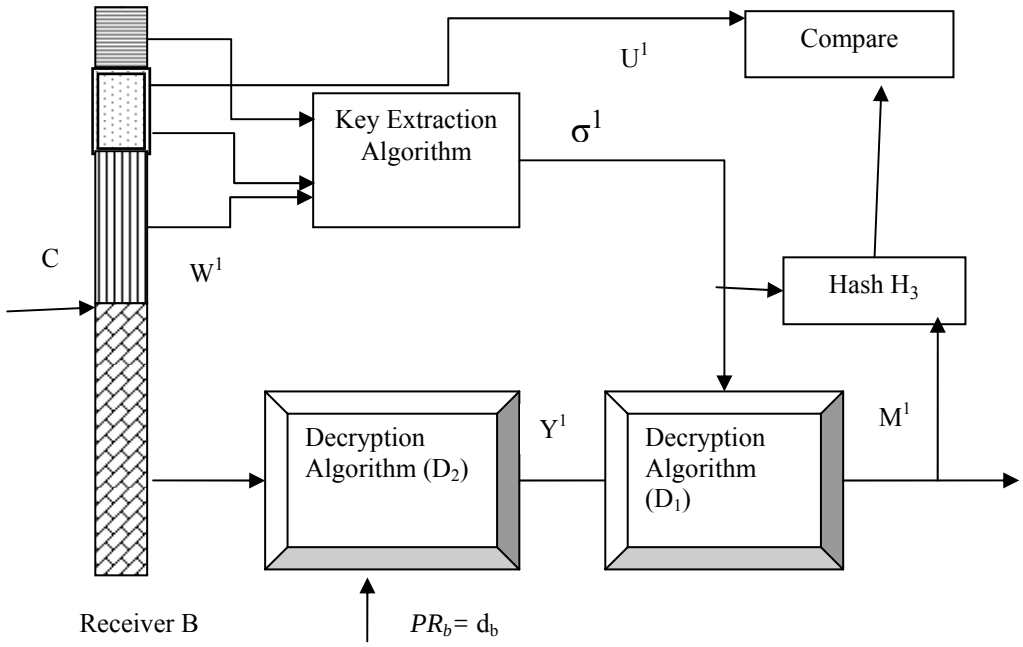

Figure 3: Message Decryption

Figure 3 shows the sequence of actions to be done by the intended receiver to decrypt the received cipher text. After receiving the ciphertext (C), receiver (B) separates the received R_1^1, R_2^1, -----,R_t^1, U^1, V^1 and W^1 parameters. A symmetric key (σ^1) is derived by using all the separated parameters.

Later, decryption algorithm (D_2) uses C_1, C_2, C_3 and receiver's private key (d_b) as the input parameters and outputs the message Y^1, the inverted transformation is written as

$Y^1 = D_2\ (d_b\ , (C_1, C_2, C_3))$ (3)

Once again, receiver applies the decryption algorithm on input parameters σ^1 and Y^1, outputs actual plaintext (M^1). The decryption algorithm is written as

$M^1 = D_1(\sigma^1,\ Y^1)$ (4)

IV. ALGORITHM FOR MESSAGE ENCRYPTION AND DECRYPTION

The entire group communication process is divided into seven phases; named as setup phase, extract phase, group key agreement phase, group encryption key generation phase, group decryption key generation phase, message encryption phase and decryption phase.

Step-1: [Group Setup]: In this step, if any user U_i is interested to participate in the group communication, initially he has to get the permission from the GC by sending his contribution to the U_0, Where U_0 is the one who collects all the user U_i's information and send them their private key information after checking their validity. The algorithm works as follows:

On the input security parameter ℓ, the GC chooses cyclic additive and multiplicative groups G_1 and G_2 with prime order q, there exist a bilinear map e: $G_1 \text{ X } G_1 \rightarrow G_2$.

1. GC chooses a random integer $k \in Z_q^*$ as the master secret key, an element $P_1 \in G_1$ at random
2. Set $P_{pub} = k\ P$, Where P is a randomly chosen generator of G_1
3. GC chooses five cryptographic one-way hash functions
   $H: \{0,1\}^* \rightarrow Z_q^*$, $H_1 : \{0,1\}^* \rightarrow G_1$
   $H_2: G_2 \rightarrow \{0,1\}^w$, $H_3: \{0,1\}^* \text{ X } \{0,1\}^w \rightarrow Z_q^*$
   $H_4 : \{0,1\}^w \rightarrow \{0,1\}^w$, for some positive integer w
4. Publish the system parameters $F=<q, P, P_1, P_{pub}, G_1, G_2, e, n, H, H_1, H_2, H_3, H_4>$ and keep the master key k secret.

Step 2: Extract: The extract algorithm that takes the master secret key k, user identity $ID_i \in \{0,1\}^*$and F as the input and generates the private key of the user is as follows:

1. Compute $Q_i = H_1(ID_i)$.
2. Output the private key $S_i = k*(P_1 + Q_i)$.

Step 3: Group key agreement: In this stage, a group of n participants generate and publishes the messages, which will be used in generation of group encryption and decryption keys. A group participant U_i, having identity ID_i and private key S_i, performs the following steps:

1. Choose a random $h_i \in Z_q^*$ and compute $x_i = g^{h_i}$
2. For $1 \le j \le n$ compute $f_j = H_1(j)$
3. For $1 \le j \le n$ compute $\Delta_{i,j} = H_1(ID_i) * f_j^{h_i}$
4. Using the private key S_i, U_i generates the signature δ_i on
   $M_i = \{ \Delta_{i,1}, \Delta_{i,2}, ----, \Delta_{i,i-1}, \Delta_{i,i+1}, -----, \Delta_{i,n}, x_i, ID_i\}$
5. Publish ($\Delta_{i,1}, \Delta_{i,2}, ----, \Delta_{i,i-1}$,null, $\Delta_{i,i+1}, -----, \Delta_{i,n}, x_i, \delta_i, ID_i$).

After completing this stage, each group participant can get the message as shown in the table 1, where $\Delta_{i,i} = H_1(ID_i) * f_i^{h_i}$ is only known to U_i, not published to any other group participants. Diagrammatic representation of step3 is shown in Figure 4.

Table 1: Message received by group participants

Required for	U_1	U_2	U_3	---	U_n	All
U_1	$\Delta_{1,1}$	$\Delta_{1,2}$	$\Delta_{1,3}$	---	$\Delta_{1,n}$	x_1, δ_1, ID_1
U_2	$\Delta_{2,1}$	$\Delta_{2,2}$	$\Delta_{2,3}$	---	$\Delta_{2,n}$	x_2, δ_2, ID_2
U_3	$\Delta_{3,1}$	$\Delta_{3,2}$	$\Delta_{3,3}$	---	$\Delta_{3,n}$	x_3, δ_3, ID_3
:	:	:	:	---	:	:
U_n	$\Delta_{n,1}$	$\Delta_{n,2}$	$\Delta_{n,3}$	---	$\Delta_{n,n}$	x_n, δ_n, ID_n
Decryption Key	d_1	d_2	d_3	---	d_n	(X,A)

Step 4: Group encryption key generation: At this stage all the n group participants will compute the group encryption key. Before calculating the group encryption key, each participant first checks the validity of n message–signature pairs (x_1, δ_1), (x_2, δ_2),----------, (x_n, δ_n).

If all the message-signature pairs are valid, then group participants calculates

$X= \prod_{i=1}^{n} x_i$ and $A= e(\prod_{i=1}^{n} H_1(ID_i),g)$

Sets (X, A) as the common group encryption key or group public key. Steps involved for group public key generation is shown in figure 5.

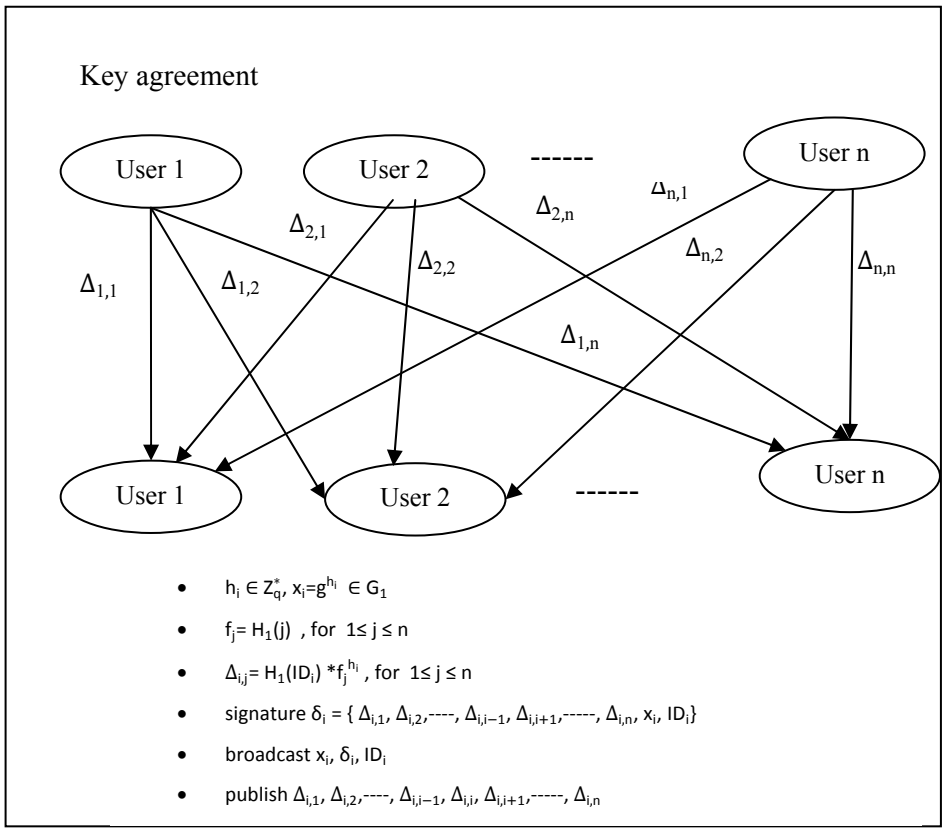

Figure 4: Key agreement by group members

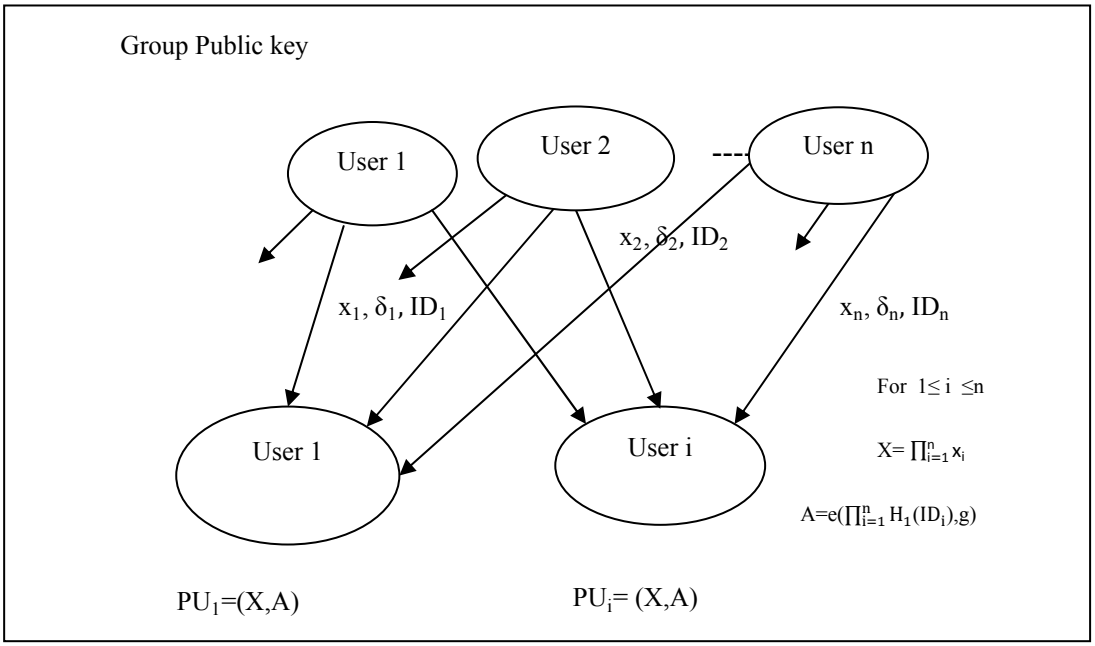

Figure 5: Derivation of common group public key

Step 5: Group decryption key derivation: Each user individually computes their private key, after checking the validity of the signatures δ_i for $1\leq i \leq n$. Participant U_i computes the decryption key $d_i = \prod_{l=1}^{n} \Delta_{l,i}$ and accepts, when all the signatures δ_1, δ_2---, $\delta_n$are valid. The sequence of operational steps for decryption key derivation is shown in figure 6.

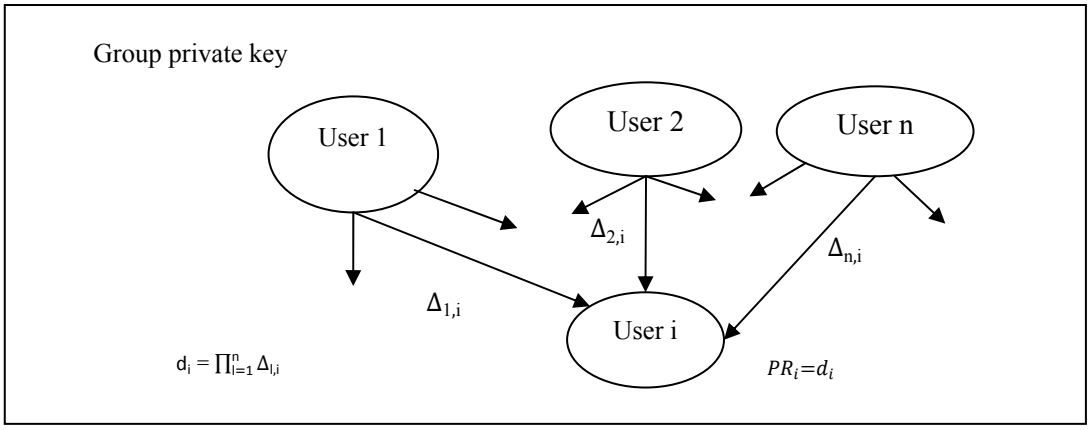

Figure 6: Group individual private key computation

Step 6: Encryption: Generation and distribution of symmetric key and message encryption will be done in this phase. When any user wants to send the message M, then he chooses a symmetric key (σ) which is used in message encryption, communicates it to other users in encrypted format using Lagrange interpolation. This stage takes the input parameters, plaintext or message (M) and select t receivers $(ID_1, ID_2 - --, ID_t)$ to whom he want to send the cipher text of M. To perform the message encryption sender executes the following steps:

1. Select a random string $\sigma \in \{0,1\}^w$ and set $r=H_3(\sigma,M)$.
2. Pick a random integer $\alpha \in Z_q^*$ and set $y=\alpha^{-1}*r \bmod q$.
3. For i=1,..,t, compute $t_i=H(ID_i)$ and $Q_i= H_1(ID_i)$.
4. For i=1,..,t, compute $f_i(x)=\prod_{1\le j\ne i\le t} \frac{x-t_j}{t_i-t_j}$
   $= a_{i,1}+a_{i,2}x+a_{i,3}x^2 +----+a_{i,t}\, x^{t-1}$, where $a_{i,1}, a_{i,2},---, a_{i,t} \in Z_q$.
5. For i=1, .. ,t compute $R_i = \sum_{j=1}^{t} a_{j,i} yQ_i$
6. Set the cipher text
   $Y=E_{H_4(\sigma)}(M)$

   $C=< R_1, R_2,------, R_t, rP, \alpha P_{pub}, \sigma \oplus H_2(e(P_{pub}, P_1)^{\alpha}), g^{\alpha}, X^{\alpha}, C_3>$, where $C_3 = Y \oplus H_2(A^{\alpha})$ and then broadcast C (cipher text) to all the recipients.

Step 7: Decryption: This algorithm is to be executed by all the t receivers to extract the symmetric key (σ), decrypt the cipher text (C) by using σ, C_1, $C_2$and C_3. Finally accepts after checking the correctness of received message.

At this stage, cipher text $C=<R_1, R_2,------, R_t, U, V,W, C_1, C_2, C_3>$, F, an receiver identity ID_i, and his private key S_i acts as inputs to decryption process. To extract the plain text (M), the receiver i performs the following tasks:

1. Compute $t_i=H(ID_i)$.
2. Find ⋋ using the following formula :
   $⋋= R_1+R_2 * t_i +----------+R_t*(t_i^{t-1} \bmod q) = yQ_i$
3. Compute $\sigma^1=W\oplus H_2(\frac{e(U,S_i)}{e(V,⋋)})$
4. Decrypt $Y^1=C_3\oplus H_2(e(d_i, C_1).e(f_i^{-1}, C_2))$
5. Find $M^1= D_{H_4(\sigma^1)}(Y^1)$
6. On the received message M^1 and computed key σ^1, receiver U_i compute $r^1= H_3(\sigma^1, M^1)$ and then test whether $U= r^1P$ or not. U_i accepts the message M^1 if it is true,otherwise rejects it.

In brief , a GC is established to run the setup, where all the users gives their identity to GC, the GC inputs the public system parameters (F), master secret key(k) and the users identity to extract and returns a private key to the user. A group key pair is derived in key agreement, where each user is choosing a random value and then multicast it to other group members by keeping his part secret. After completion of the multicasting each user can be able to compute the common group key (X, A) using the group encryption key derivation and their individual private key (d_i) derived using group decryption key derivation algorithm. This public and private key pairs becomes the asymmetric key pairs, further they are used for message encryption. When any sender wants to send message (M) to other group participants, inputs F, identities of receivers, symmetric key (σ) and plaintext message (M) to encrypt to get the cipher text. The resultant cipher text is again encrypted using the group public key and then broadcast it. After receiving the cipher text, all the receivers can input the F, cipher

text, identity, group public key, session private key and his private key to decrypt. If the participant is the intended receiver, then decrypts the cipher text and extracts plaintext message using his group private key and session key, otherwise he rejects it.

Although the proposed algorithm seems to be simple, there are many implications. Firstly, the number of users in the group is dynamic; group session key pair is also dynamic, only computed before starting the communication. This makes the intruder to stop analyzing the session key pair. When any changes to the group, join/leaving of the participant, again new session key is derived by new group participants. Secondly, after the group key pair derivation, out of the group one participant who wants to send the message to the group members, he will decide a symmetric key and conveyed to the other participants in encrypted form using group key pair. Afterwards, plaintext message is encrypted twice using symmetric key and receiver's public key, which provides the confidentiality to the message. Thirdly, proposed technique is checking the identity of the user and signature verification provides the authentication, this also stops the involvement of the intruder from joining and analyzing into the group. Finally, it also provides the mechanism to check for the correctness of the received message at the receiver side by comparing the r^1P with U.

V. PERFORMANCE ANALYSIS

The evaluation of the performance was done using JUnit test which is used for finding the response time of the algorithms. Initially the parameters to be tested to evaluate the algorithms have been determined. They are three types of entities namely the system parameters, experimental factors and simulation procedures. The proposed technique was designed and tested using PBC library on the desktop using Intel(R) Core™ i5-2400 CPU@3.10GHZ, frequency 3.09 GHz and 2.91GB RAM. The security parameter ℓ was set to be 160 bits. By changing the number of participants in the group, the tests have been performed by setting the length of the group elements in G_1 and G_2 are set to 128 and 1024 bits.

The experimental factors that are chosen here to determine the performance of algorithms and its efficiency were in terms of speed with the various sizes of data blocks. The algorithms were evaluated in terms of time required by changing the group participants from 2 to 100. All the implementations were made exact so that the results are comparable.

From the algorithm, it can be observe that, in symmetric key encryption case, sender decides and broadcasts the key. Resulting that, the key computation time is proportionately increasing with increase in group participants. In asymmetric key agreement, a matrix is formulated based on the number of participants involved in the communication, which results that the computation time is again depending on the count of participants. Hence, in the proposed algorithm the key computation increases with increase in group members. The computation cost for encrypting the key is between 16 milliseconds to 300 milliseconds. To generate symmetric and asymmetric public key, proposed algorithm requires 2n multiplications, 2n pre computation multiplications.

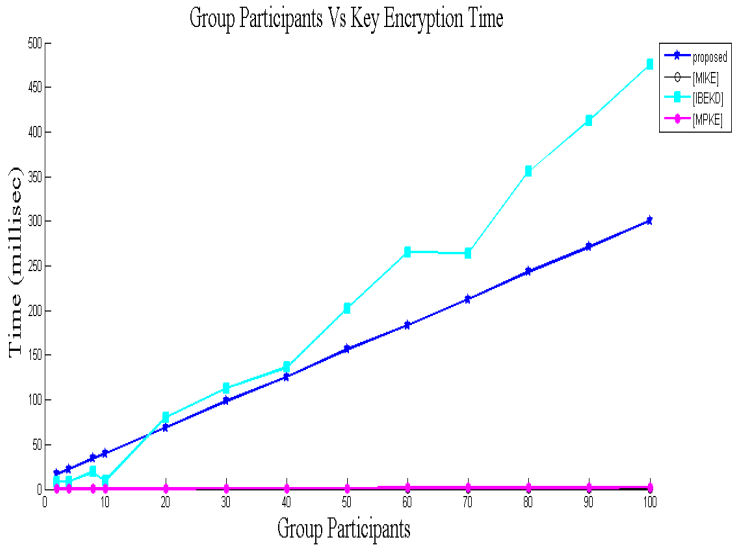

Figure 7: Computation time for key generation and distribution

Figure 7 shows the relationship between the number of group participants and execution time to generate and distribute encrypted keys in the proposed technique with other existing symmetric and asymmetric group key management protocols. After analyzing all the algorithms, we can observe that, for key encryption IBEKD technique is using 2 pairings, 1 hash function, n group multiplications, n-1 additions and 1 XOR operation , MPKE is using m+1multiplications, m group additions, where as MIKE is using XOR operations.

From the figure, we can observe that, in proposed technique, the time required for generation of encrypted key linearly increases with the increase in group participants. Because, proposed one is using two key generation algorithms, one is symmetric key generation and distribution and the other one is asymmetric based group key agreement. In asymmetric key generation a matrix is formulated to generate the key. The matrix dimension can change with respect to number of group participants. With the increase in participants, matrix formation costs, in turn increases in key generation cost. Results, more computation time in key generation and distribution. Even though, it requires more computation time, the proposed one is more secured than other existing techniques, because two keys are used in encrypting message. But MIKE, MPKE techniques requires less and constant key encryption time irrespective of the group participants.

The comparison of computation time to decrypt the key of proposed technique and other existing techniques is shown in figure 8 . In proposed technique, the decryption key operation requires n scalar group G_1 multiplication. In IBEKD, decryption key requires (n-1) additions of scalar group G_1, n scalar multiplications in G_1, two pairing computations, one hashing function and one XOR operation, results more computation cost. MPKE technique requires n+m scalar group multiplications and m-1 additions, where m is the degree of the polynomial, because of only additions and multiplications, results less computation time compared to IBEKD. From the figure , we can observe that proposed techniques requires less key decryption computation time compared to IBEKD and MPKE techniques.

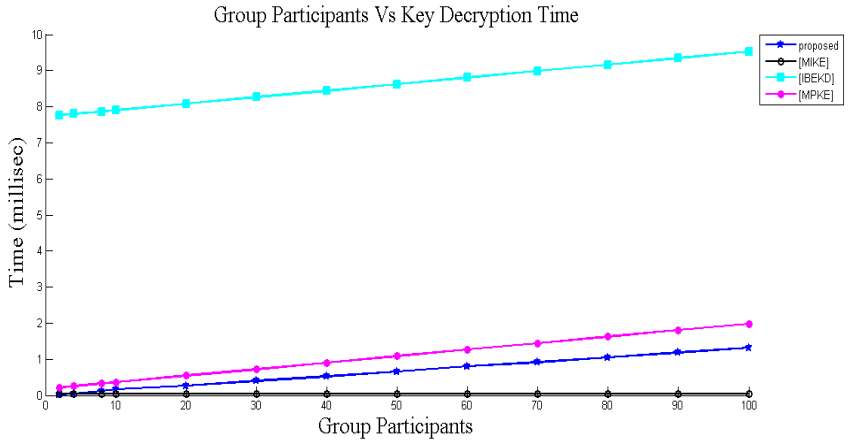

Figure 8: Comparison of computation time for Key Decryption

From the figure 9, we can observe that, proposed algorithm is taking less time than MIKE technique. To encrypt the message MIKE technique requires 2n+1 multiplication, n-1 additions, one hash function and one mapping, results more computation time. IBEKD requires 2 mappings and one XOR operation, MPKE requires 4 multiplications and one pairing. The proposed technique requires 2 XOR operations, one mapping, one hash function and three power functions. With this description proposed technique requires more computation than IBEKD and MPKE and less than MIKE.

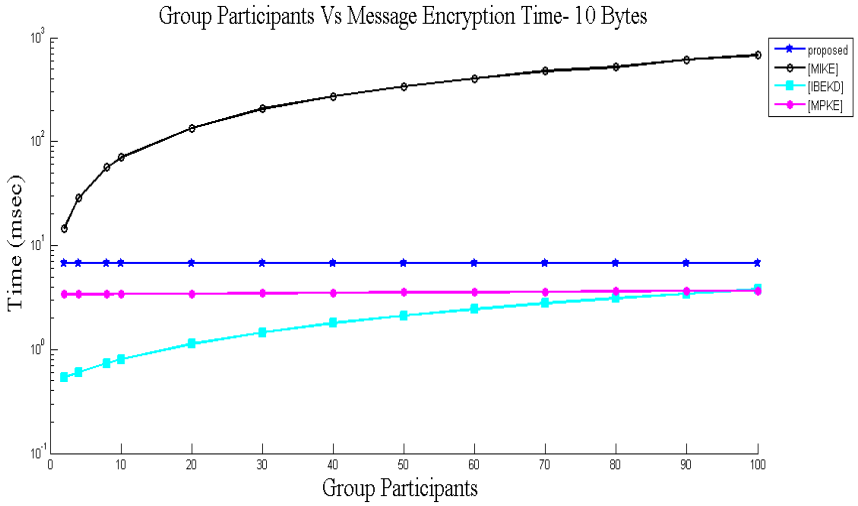

Figure 9: Comparison of computation time for message encryption

The computation time for message decryption in various approaches are shown in figure 10. From the figure we observe that, proposed technique requires constant time for message decryption, irrespective of the group size. This requires one XOR operation and two pairing functions, where as MPKE requires (n+2) multiplications and two mappings. MPKE decryption time dependent on number of participants, MIKE requires (n+1) mappings and one multiplication. Hence, it requires more computation time.

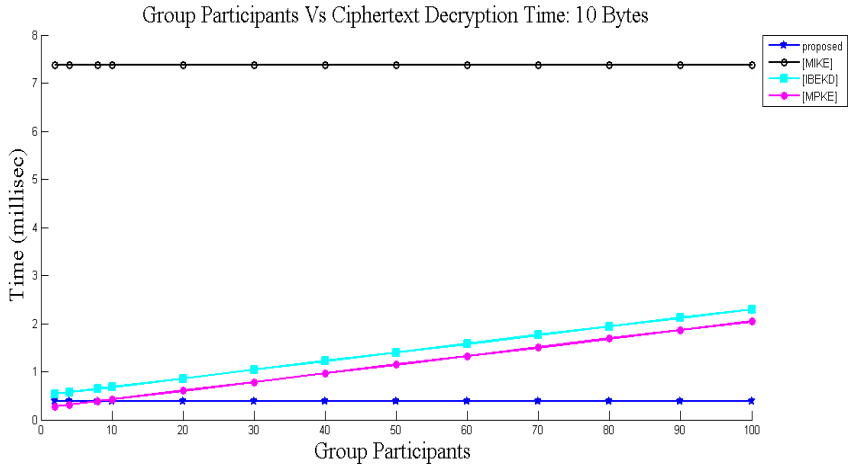

Figure 10: Comparison of computation time for message decryption

The results observed above can show that the proposed algorithm has a better performance than existing MIKE algorithm. The tests on the proposed algorithm demonstrate, authentication and confidentiality schemes at varying amounts of overhead, and also have different performance characteristics.

Afterwards, the technique is tested by varying file size from 10 bytes to 5 MB. The message encryption and decryption computation times are shown in table 2 and table 3. As already mentioned, proposed one is taking constant time in both encryption and decryption irrespective of the number of group participants

Table 2: File Encryption time

Group Participants	Time (in seconds)						
	10 bytes	1KB	10KB	256KB	512KB	1MB	5MB
2	0.006744	0.006768	0.04026	1.026208	2.156413	4.335817	21.62177
4	0.006744	0.006768	0.04026	1.026208	2.156413	4.335817	21.62177
8	0.006744	0.006768	0.04026	1.026208	2.156413	4.335817	21.62177
10	0.006744	0.006768	0.04026	1.026208	2.156413	4.335817	21.62177
20	0.006744	0.006768	0.04026	1.026208	2.156413	4.335817	21.62177
30	0.006744	0.006768	0.04026	1.026208	2.156413	4.335817	21.62177
40	0.006744	0.006768	0.04026	1.026208	2.156413	4.335817	21.62177
50	0.006744	0.006768	0.04026	1.026208	2.156413	4.335817	21.62177
60	0.006744	0.006768	0.04026	1.026208	2.156413	4.335817	21.62177
60	0.006744	0.006768	0.04026	1.026208	2.156413	4.335817	21.62177
80	0.006744	0.006768	0.04026	1.026208	2.156413	4.335817	21.62177
90	0.006744	0.006768	0.04026	1.026208	2.156413	4.335817	21.62177
100	0.006744	0.006768	0.04026	1.026208	2.156413	4.335817	21.62177

Table 3: File Decryption Time

Group Participants	Time (in seconds)						
	10 bytes	1KB	10KB	256KB	512KB	1MB	5MB
2	0.000379	0.007269	0.009107	0.066605	0.131671	0.256537	1.259191
4	0.000379	0.007269	0.009107	0.066605	0.131671	0.256537	1.259191
8	0.000379	0.007269	0.009107	0.066605	0.131671	0.256537	1.259191
10	0.000379	0.007269	0.009107	0.066605	0.131671	0.256537	1.259191
20	0.000379	0.007269	0.009107	0.066605	0.131671	0.256537	1.259191
30	0.000379	0.007269	0.009107	0.066605	0.131671	0.256537	1.259191
40	0.000379	0.007269	0.009107	0.066605	0.131671	0.256537	1.259191
50	0.000379	0.007269	0.009107	0.066605	0.131671	0.256537	1.259191
60	0.000379	0.007269	0.009107	0.066605	0.131671	0.256537	1.259191
60	0.000379	0.007269	0.009107	0.066605	0.131671	0.256537	1.259191
80	0.000379	0.007269	0.009107	0.066605	0.131671	0.256537	1.259191
90	0.000379	0.007269	0.009107	0.066605	0.131671	0.256537	1.259191
100	0.000379	0.007269	0.009107	0.066605	0.131671	0.256537	1.259191

Finally, the security properties of proposed scheme are compared with other existing schemes covered in the literature chapter. The comparisons are shown in table 4, in MIKE approach multicasting and authentication is not possible, MPKE approach is suitable for static group, authentication is not provided. But proposed one is suitable for dynamic groups, providing security services like confidentiality, integrity and authentication.

B_cast : All group members with broadcasting

M_cast:Sender can multicast to groups

S_Multicast :Sender can multicast to selected receivers

Key_mem_join : Key reuse on member join

Key_mem_leave: Key reuse on member leaving

Private_key : Single private key generator

Table 4: Security properties of proposed technique

Property	MIKE[28]	IBEKD[26]	MPKE[27]	Proposed
B_cast	√	√	√	√
M_cast	X	√	√	√
S_Multicast	X	√	√	√
Key_mem_join	√	√	X	√
Key_mem_leave	√	√	X	√
Private_key	√	√	X	√
ID-based	√	√	X	√
Confidentiality	√	√	√	√
Authentication	X	X	X	√

X: No √: YES

VI. CONCLUSION

A novel protocol was proposed based upon the algorithm, which is termed as Lagrange Interpolation based Asymmetric Group key Agreement protocol. It allows the participants in the group to derive a common encryption key, offers the key security and unknown key share properties. Evaluation shows that, the overheads of the proposed protocol are less when compared to others [26, 27, 28]. It has combined both symmetric and asymmetric key cryptographic concepts. Confidentiality is achieved by performing double key encryption on message. Integrity is provided by calculating the message digest on message and the chosen symmetric key, checking for correctness at the receiver side by comparing the computed message digest with received message digest.

REFERENCES

[1] Diffie, W., Hellman, M.: "New Directions in Cryptography". IEEE Transactions on Information Theory,1976,22(6), pp. 644-654
[2] Choi, J. Hwang, D. Lee. "Efficient ID-based Group Key Agreement with Bilinear Maps". PKC 2004, LNCS 2947, pp. 130-144, Springer, Heidelberg, 2004.
[3] R.Dutta, R.Barua,"Constant Round Dynamic Group key Agreement", ISC 2005, LNCS 3650,pp. 74-88 Springer Heidelberg, 2005
[4] J.Katz, M.Yung "Scalable Protocols for Authenticated Group Key Exchange"Crypto 2003, LNCS 2729,pp. 110-125,Springer, Heidel berg,2003.
[5] L. Zhang, B. Qin, Q. Wu, F. Zhang, "Efficient many-to-one authentication with certificateless aggregate signatures", Computer Networks ,2010,54(14),pp. 2482-2491.
[6] D. Boneh, X. Boyen, E. Goh. "Hierarchical Identity Based Encryption with Constant Size Ciphertext." EUROCRYPT 2005, LNCS 3494, pp. 440-456, Springer, Heidelberg, 2005.
[7] Joux. A" One Round Protocol for Tripartite Diffie-Hellman". Journal of Cryptology ,2004,17(4), pp. 263-276.
[8] X.Cao, W.Kou,X.Du, A pairing free identity-based authenticated key agreement protocol with minimal message exchanges, Information Science,2010,180(15),pp. 2895-2903
[9] T.Chang, M.Hawng, W.Yang, "A Communication –efficient three party password authenticated key exchange protocol, Information Science ,2011,181(1),pp. 217-226
[10] H.Guo, Z.Li, Y.Mu, X.Zhang, "Provably secure identity based authenticated key agreement protocols with malicious private key generators, Information Science,2011, 181(3),pp. 628-647.
[11] L.Zhang, F.Zhang, Q.Wu, J.Domingo-Ferrer , "Simulatable certificateless two party authenticated key agreement protocol information" Information Science, 2010, 180(15),pp. 1020-1030.
[12] Shamir,"Identity based cryptosystem and signature schemes"in proceedings of crypto'84 LNCS vol 196, Springer Verlab, 1984,pp. 47-53
[13] J.Camenish, S.Hohenberger, M.Pedersen,"Batch verification of short signatures" in proceedings of EUROCRYPT 2007, LNCS, vol 4515, Springer –verlag, 2007,pp. 246-263
[14] L.Zhang, B.Qin, Q.Wu, F.Zhang,"Efficient many-to-one authentication with certificatesless aggregate signatures"computer Networks,2010, 54(1),pp. 2482-249
[15] J. Katz, M. Yung. "Scalable Protocols for Authenticated Group Key Exchange". CRYPTO 2003, LNCS 2729, pp. 110-125, Springer, Heidelberg, 2003.
[16] M. Burmester, Y. Desmedt." A Secure and Efficient Conference Key Distribution System." EUROCRYPT 1994, LNCS 950, pp. 275-286, Springer, Heidelberg, 1995.
[17] E. Bresson, O.Chevassut, D.Pointcheval, J.Quisquater, "Provably authenticated group Diffie-Hellman key Exchange"in Proceedings ACM CCS 2001, ACM , 2001, pp. 255-264
[18] Bresson, O.Chevassut, D.Pointcheval, "Dynamic group Diffie Hellman Key exchange under standard assuptoions", in Prpceedings of EUROCRYPT 2002, LNCS, vol 2332, Springer-verlag 2002,pp. 321-336
[19] Bresson, O.Chevassut, D.Pointcheval," Provably authenticated group Diffie – Hellman key exchange", The dynamic case in Proceedings of ASIACRYPT 2001,LNCS, vol 2248 , Springer verlag, 2001,pp. 290-309.
[20] Q.Wu,Y.Mu,W.Susilo, B.Qin,J.Domingo –Ferrer, "Provably Secure one-round Identity Based Authenticated Asymmetric Group Key Agreement Protocol", Elsevier, May 2011
[21] Lei.Zhang, Q.Wu,U.G.Nicolas, , B.Qin,J.Domingo –Ferrer."Asymmetric Group key agreement Protocol for open Networks and its application to Broadcast Encryption", Elsevier, September 2011
[22] Q.Wu,Y.Mu,W.Susilo, B.Qin,J.Domingo –Ferrer."Asymmetric Group key agreement "EUROCRYPT 2009,LNCS 5479,pp. 153-170 Springer Heidelberg,2009
[23] R.Sivaranjani, D.Lalitha Bhaskari & P.S.Avadhani,"Current Trends in Group Key Management" International Journal of Advanced Computer Science & Applications, Vol-2, pp. 82-86,(Nov, 2011).
[24] R.Sivaranjani, D.Lalitha Bhaskari & P.S.Avadhani ,"Secure Message Transmission using Lagrange Polynomial Interpolation and Huffman Coding" International Journal of Computer Applications, Vol.55-No.1,(Oct 2012).
[25] R.Sivaranjani, D.Lalitha Bhaskari & P.S.Avadhani ," An Extended Identity Based Authenticated Asymmetric Group Key Agreement Protocol", International Journal of Network Security, Vol.17, No.5, PP.510-516, Sept. 2015.
[26] X. Du, Y. Wang et al., "An Id-based broadcast encryption scheme for key distribution", IEEE Trans. Broadcast., vol.51, no.2, pp.264-266, June 2005
[27] L. Lu and L. Hu, "Pairing-based multi-recipient public key encryption", Int. Conf. on Security & management, Las Vegas, Nevada, USA, pp.159–165, June 2006
[28] J. Baek, R. Safavi-Naini and W. Susilo, "Efficient multi-receiver identity based encryption and its application to broadcast encryption", Public Key Cryptography, 8th Int. Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, LNCS 3386, pp.380–397, Jan. 2005

AUTHOR PROFILE

Dr. R.Sivaranjani, is an Associate Professor in the Department of Computer Science and Engineering of GMRIT, Rajam. She has published research papers in various journals and conferences. Her areas of interest includes Data Security, Cyber forensics, Image Processing and Big Data Analytics.