Design of ID-based Contributory Key Management Scheme using Elliptic Curve Points for Broadcast Encryption

Deepa S. Kumar
Research Scholar,
Karpagam University, Coimbatore.
Associate Professor
College of Engineering Munnar

M. Abdul Rahman
Pro-Vice Chancellor
APJ Abdul Kalam- Technological University,
Thiruvananthapuram

ABSTRACT

Broadcast encryption is the process of delivering encrypted data through a secure channel, intended for multiple users, in which only the privileged users can decrypt the content. In a broadcasting system, all the intended recipients are required to be accommodated in an organized way, which is possible through an efficient key management scheme. An ideal Broadcast Encryption scheme should define a key management scheme and an encryption scheme. The potential steps in key management are key generation, a perfect revocation scheme, and a re-keying mechanism. This paper describes a key generation mechanism using Elliptic Curves. The generated key can be used as a symmetric key. The important feature of this symmetric key is that the key is constituted by the contribution from all the legitimate users so that the revocation mechanism can be simplified, but at the expense of communication overhead. The proposed method describes two approaches to communicate the symmetric key to the users or to the groups.

General Terms

Broadcast encryption, key management, re-key, elliptic curve, symmetric key

Keywords

Broadcast controller, Group Controller, Legitimate users, Data Encryption Key(DEK), Key Encryption Key(KEK), Discrete Logarithm Problem(DLP)

1. INTRODUCTION

Contributory common key generation between two parties has solved by public key cryptosystems, but extending the generation of common key share from multiple participants remains a challenging task. Broadcast system uses multiple subsets of receivers. In this paper, the BE system is implemented as a hierarchy consisting of a central Broadcast Controller(BC) on the top of the hierarchy, number of Group Controllers(GCs) and the Legitimate Users(LU) on the bottom of the hierarchy. The said hierarchical implementation yields the keying process in an organized way.

Elliptic curve has found application in cryptography in recent years because the elliptic curves over finite fields provide an enormous supply of finite abelian group. They are amenable to computation, even when large, because of their rich structure. The proposed key generation scheme describes a method to generate points from the elliptic curve, from where the x-coordinates are assigned as identity values for groups and legitimate users. One of the attractive features of elliptic curve is that when doubling and adding points, starting from the generator point, it creates a variety of randomness in coordinates. These coordinates of elliptic curves when doubled and/or added resulting new coordinates which does not keep any relation with the previous or next points generated. Also the generation of points is not a complex task. This principle has motivated the idea of assigning the x coordinate values as the identity values.

This paper proposes an ID-based group key agreement protocol with less computational overheads than the other existing protocols and free from the bilinear pairing, which is treated as a complex mathematical operation. Also the proposed key management protocol completely eliminates the key escrow problem since it does not avail any key from the Private Key Generators. Since the identities are assigned by the Broadcaster for the legitimate users at the time of registration with the key server, the need of an authentication mechanism is avoided in the proposed method. Also the necessity of an external certification authority other than the Broadcast Controller is completely ignored in this method.

The rest of this paper organized as follows. The preliminaries related to proposed work are addressed in Section II. In Section III, the state of art on group key agreement protocols and group key management requirements are described. The Section IV proposes the protocol and Section V states a Broadcast encryption scheme suitable to the method proposed. In section VI analysis of protocols are done and Section VII lists the merits and shortcoming of the proposed system followed by conclusion.

2. PRILIMINARIES

The preliminaries required to understand the proposed protocol are discussed here.

2.1 Background Of Elliptic Curve Group

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

Y2 = (x3 + ax + b)mod p. .... (1)

where a, b ∈ Fp and (4a3 + 27b2) not= 0 . ... (2)

The points on E/Fp together with an extra point O called the point at infinity forms a group

G = {(x, y) : x, y ∈ Fp and (x, y) ∈ E/Fp } ∪ {O}..... (3)

2.2 Point Addition in Elliptic curves

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

Let P, Q ∈ G , l be the line connecting P and Q , and R be the third point of intersection of line l with E/Fp . Let l be the line connecting R and O. Then P + Q is the point such that l intersects E/F p at R and O and P + Q. i.e with 2 distinct points, *P* and *Q*, addition is defined as the negation of the point resulting from the intersection of the curve, *E*, and the straight line defined by the points *P* and *Q*, giving the point, *R*.

P + Q = R

$(X_p, Y_p) + (X_q,Y_q) = (X_r,Y_r)$

Assuming, the elliptic curve, *E*, is given by $y^2 = x^3 + ax + b$, this can be calculated as:

$$x_{r=\lambda^{2}-x_{p}-x_{q}}$$

$$y_r = \lambda(x_p - x_r) - y_p$$

$$\lambda = \frac{y_q - y_p}{x_q - x_p}$$

2.3 Point Doubling in Elliptic curves

When the generator point G is known initially, the tangent to the curve, *E*, at G is calculated as

$$\lambda = \frac{3x_p^2 + a}{2y_p}$$

where *a* is from the defining equation of curve, *E*. The next point after doubling is $(x_{r,}y_r)$.

3. STATE-OF-THE-ART ON KEY MANAGEMENT

The major security concern in broadcasting is key management. Traditional group key agreement protocols [1]-[3] are based on the traditional public key cryptography and hence require public key infrastructure (PKI) to issue and manage the public key certificates, which suffers from key escrow problem. The protocols generally requires O (n) or O (log n^2) communication rounds for n number of participants. The issue of key management can be simplified by ID-based cryptosystem which overcomes the burden of heavy public key certificate managements [4]. In ID- based system user's unique identifiers itself functioned as its public key and often requires an offline trusted authority for generating their private key [5].Existing key management systems are implemented with two approaches called group key management and key distribution system [6].Group key agreement allows a group of users to negotiate a common secret key via open networks [7]. Then any member can encrypt any confidential message with the shared secret key and only the group members can decrypt. BE scheme in the literature are classified into two categories: symmetric BE and public key BE. In the symmetric key setting, a common secret key is used for encryption and decryption. In broadcasting scenario, the broadcaster has to negotiate on a common shared secret key which involves a lot of communication among the different legitimate users, broadcast controllers and group controllers etc. In the public key setting, in addition to the secret keys for each user, the broadcaster also generates a public key for all the users. Conventional methods can avail the key pairs from the Private Key Generators (PKG) which suffers from key escrow problem. From the literature there exists taxonomy of key management schemes that can be used for secure group communication.

3.1 Principles of key management:

The maintenance and the distribution of the keys (which involves re-keying also) for encryption/decryption is commonly called Group Key Management. Each membership change in the group requires re-keying and the group may be highly dynamic, the major challenge of group key management is how to assure re-keying using the minimum bandwidth overhead and without increasing the storage overhead.

3.1.1 *Group Key Management Requirements*

The group key requirements are broadly classified into four approaches viz: security requirements, QoS requirement, key server requirement and group members' resource requirement.

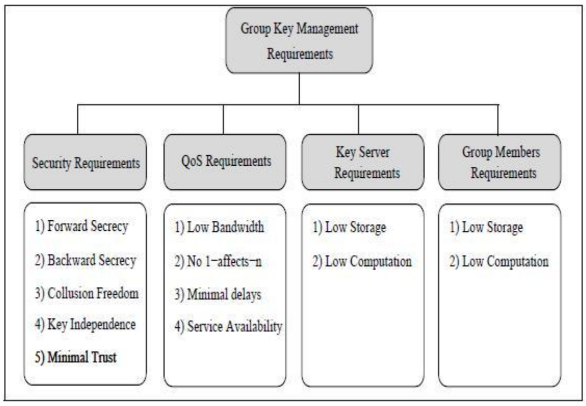


Figure 1: Taxonomy of group key management requirement[6].

3.1.1.1 *Security requirements*

1. Forward secrecy requires that the users who left the group should not have access to any future key. This ensures that a member cannot decrypt data after it leaves the group. To assure forward secrecy, a re-key of the group with a new Data Encryption Key (DEK) after each leave from the group is the ultimate solution.\
2. Backward secrecy requires that a new user that joins the session should not have access to any old key. This ensures that a member cannot decrypt data sent before it joins the group. To assure backward secrecy, a re-key of the group with a new DEK after each join to the group is the ultimate solution.
3. Collusion freedom requires that any set of unauthorized scrupulous users should not be able to deduce the current data encryption key.
4. Key independence: a protocol is said key independent if a disclosure of a key does not compromise other keys.
5. Minimal trust: the key management scheme should not place trust in a high number of entities. Otherwise, the effective deployment of the scheme would not be easy.

3.1.1.2 *Quality of service requirement:*

1. Low bandwidth overhead: the re-key of the group should not induce a high number of messages, especially for dynamic groups. Ideally, this should be independent from the group size.
2. 1-affects-n: a protocol suffers from the 1-affects-n phenomenon if a single membership change in the group affects all the other group members. This happens typically when a single membership change requires that all group members commit to a new DEK.

3. Service availability: the failure of a single entity in the key management architecture must not prevent the operation of the whole multicast session.
4. The key management scheme must not induce neither high storage of keys nor high computation overhead at the key server **or** group members.

3.1.1.3 Key server requirement:

The key server should have more storage requirement and also entitled to have much computational complexity when compared to other members in the key hierarchy. The need for the storage of older keys at the key server is obsolete in the architecture because of the usage of elliptic curve points for identity generation. The major operation for key generation at the key server involved is simple XOR operation which incurs much less computational complexity.

3.1.1.4 group members' resource requirement:

The group members store their identity values issued by the key server with a much less storage requirement. No need of any computing resources at the group members other than conferencing since the computations are done at BC and GC.

The group key management is categorized into centralized, decentralized and distributed key management schemes.

The proposed method is a hybrid of centralized and distributed key approach. In centralized approach, all the keys are controlled by the central authority, which is in turn classified into pair-wise keys, broadcast secrets and hierarchy of keys approaches. In pair-wise keys approach the re-keying incurs a lot of update messages. In broadcast secret approach by Chiou and Chen [8] introduces a secure lock: a key management protocol in which the key server requires only a single broadcast to establish a group key or to re-key the entire group in the case of leave. But complex computation is required at the server since the algorithm needs to solve the simultaneous congruences using Chineese Remainder Theorem. The third approach uses a hierarchy of keys approach whereby the aim of this approach is to reduce the re-key message updates. The paper describes, key distribution and maintenance using centralized hierarchy of keys approach. The central authority is Broadcast Controller on the top of the hierarchy who computes the shared secret key. The key independence is the utmost factor which decides the security of the system which is ensured by the central authority. Ali Miri & Behzad Malek [9] tabulated centralized group key management protocols based on Communication complexity of broadcast messages, Computation complexity to send broadcast messages, Size of update messages and one-affect-all phenomenon.

Ali Miri & Behzad Malek[10] proposed a comparison of the centralized group key management protocols based on the following factors under consideration.

Size of the broadcast group – n

Does not have the 1-affects-all effect -1-not –All

Communication complexity of broadcast messages – Communication

Computation complexity to send broadcast messages – Computation

Size of update messages - Update

A comparison of the centralized group key management protocols by Ali Miri & Behzad Malek[10] is appended in Table 1.

Table 1: comparison of group key management protocols.

Scheme	**1-Not-All**	**Communication**	**Computation**	**Update**
Secure Lock[8]	√	O(n)	O(1)	O(1)
Burmerster et al.[10]	-	O(1)	O(n)	O(n)
Perrig et al.[11]	-	O(log2n)	O(log2n)	O(log2n)
Barua et al.[12]	-	O(n)	O(n)	O(log2n)
Choi et al.[13]	-	O(1)	O(n)	O(n)
Boneh et al.[14]	√	O(1)	O(n)	O(1)
Gentry & Waters[15]	√	O(√n)	O(n)	O(1)
Behzad Malek & Ali Miri [9]	√	O(1)	O(n)	O(n)
The proposed scheme	√	O(1)	O(1)	O(C)

4. PROPOSED METHOD

The proposed system is a method of key share generation using elliptic curve and the formulation of symmetric key for Broadcast encryption. It is proposed of using the group of points on an elliptic curve defined over finite field for assigning the identities of the legitimate users [LUs] by the Broadcast Controller. The curve points are calculated, after fixing the generator point, by the BC and distributed to the GC through which the identities of LU are generated and distributed. The key share computation should be done by the GC, which consumes only very less computation power for computing XOR on all identities.

In a traditional symmetric key scheme, the choice of key is only restricted to the key size depending on the symmetric algorithm chosen. But for BE Scheme in a dynamic environment, whenever a user joins the group or when a user is revoked, the re-keying and updates should be minimal. The proposed method uses a re-key mechanism which does not affects the existing members , but the group share of the particular group which accommodates the user will be recalculated and send the update to the central authority for calculating the final key share. Also the re-keying would not have to explicitly check for key independence since the identity values are generated from the point addition calculation of elliptic curves.

In this method, when a user is revoked from the system, the identities of others will not affected, except the keyshare of the group corresponds to that particular revoked user.

4.1 Identity Assignment and Key Distribution

This approach uses the centralized approach wherein usually a central authority who manages the entire multicast groups and its memberships. At the same time, the burden of managing the group of users is under the control of Group Controllers. The GC is responsible for the generation and distribution of identities to the group of users. The GC computes the key share and unicast to the BC. Upon receiving all the keyshares from all valid groups, BC computes the final symmetric key.

The distribution of identities to groups and users and key share calculation is shown in fig 3: {BC – Broadcast Controller, GC – Group Controller , LU – Legitimate Users; ID – Identities ; KS – Key Share; IDBC – Broadcaster Identity; IDGC – Group Identity; }

$\mathrm{KS1=ID1}\oplus\mathrm{cID2}\oplus\mathrm{ID3}\oplus\mathrm{IDGC1}$

$\mathrm{KS2=ID4}\oplus\ \mathrm{ID5}\oplus\ \mathrm{ID6}\oplus\ \mathrm{ID7\ DGC2}$

$\mathrm{KS3=ID8}\oplus\ \mathrm{ID9}\oplus\ \mathrm{ID10\ ID11}\oplus\mathrm{IDGC3}$

$\mathrm{KS4=ID12}\oplus\ \mathrm{ID13}\ \oplus\mathrm{IDGC4}$

$\mathrm{KS\ =KS1}\oplus\mathrm{KS2}\oplus\mathrm{KS3}\oplus\mathrm{KS4}\oplus\mathrm{IDBC}$

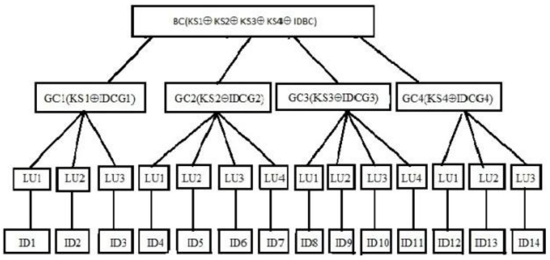


Figure 2: Identity Assignment and Key distribution in the proposed system

Usually a server sends data to a group of receivers in a single broadcast session. The security of the session is managed by two main functional entities: a Group Controller (GC) responsible for authentication, authorization and access control, and the Broadcast Controller (BC) responsible for the distribution and maintenance of keys.

4.2 Implementation of Conference Key Management

This method uses two levels of communication i.e between the BC and the Group Controllers and between the group controllers to the legitimate users within the group.

The group multicast key is generated and maintained by the BC and the Group Controller maintains the group multicast key for the group of users.

In Conference Key management system, the keys are exchanged in such a manner so that the identities of the users and group controllers must be concealed from each other. And after a round of exchange, all the users in the group receive the key.

Two types of conference keys are generated in the proposed system - Group conference key and user conference key.

4.2.1 Group conference key

Broadcast controller issues the specific group ids to all the groups. The BC computes the final keyshare by adding all the keyshares of group ids with its own identity.

The group conference key protocol is initiated by the BC by unicasting a pair of values to a particular GC whom should be designated as the leader. The pair of values includes the identity of BC (BCID) and the total number of users in the conferencing. The leader computes XOR on BCID and its GCID and set the conference count decremented by one and send the pair of values contains the computed key share and the conference count to the next GC. This process repeats until the conference count reaches to zero by a GC whom should multicast the final computed key share which is the symmetric key for decryption among the multicast group.

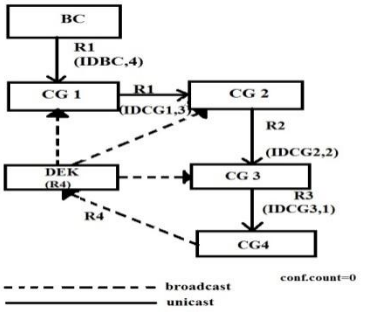


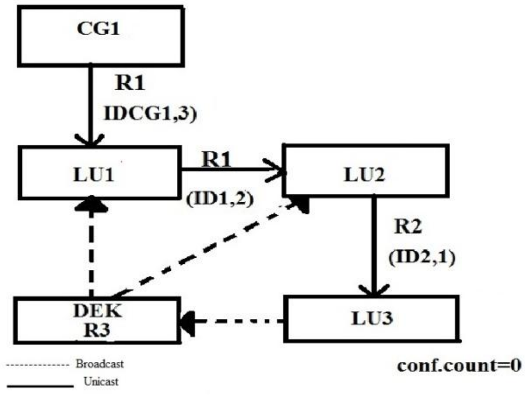
Figure 3: Group CoEnference Key Protocol

R1->IDBC $\oplus$IDCG1
R2->R1$\oplus$IDCG2
R3->R2$\oplus$IDCG3
R4->R3$\oplus$IDCG4
R4->DEK
Broadcasted by CG4 when the
Conference Count=0

4.2.2 User conference key

The user conference key protocol is initiated by the group controller(GC) which issues its id and the conference count to one leader(legitimate user). The leader computes the partial key by XORing GCID and its own id and decrement the conference count by one and sent to the next user. The next user performs XOR on the received partial key with his own id and decrement the count by one.

This process continues until the conference count reaches to zero whereby the computed key share is multicasted among the users to access the symmetric key.



R1->IDCG1⊕ID1
R2->R1⊕ID2
R3->R2⊕ID3
R3->DEK
Broadcasted by LU3 when the Conference Count=0

Figure 4 : Group Conference Key Protocol

This paper describes a method to generate a symmetric key from the contribution of all the users under the control of broadcaster and group controllers. Two levels of communication can be achieved from the broadcaster to all the group controllers and/or from the group controllers to all the members of the group. The peculiarity of this approach is that the symmetric encryption/decryption key is communicated without using traditional methods. The only computation to be performed is simple XOR operation.

In the literature, the elliptic curve is not used to generate a unique identity for the users. The point generated from the elliptic curve has the essential properties of the key such as **Forward secrecy** because users who left the group should not have access to any future key. This ensures that a member cannot decrypt data after it leaves the group.

The calculated points on the curve over finite field satisfies **Backward secrecy** requires that a new user that joins the session should not have access to any old key. This ensures that a member cannot decrypt data sent before it joins the group.

The identity values generated are **Collusion free** since that any set of unauthorized users should not be able to deduce the current symmetric encryption key.

The created symmetric key claims the feature of **Key independence** because the disclosure of a key does not compromise other keys.

1. The creation of key from the contribution of every users' identity values obtained from elliptic curves have only low bandwidth overhead because the re-key of the group should not induce a high number of messages, especially for dynamic groups. Ideally, this should be independent from the group size.
2. 1-affects-n: a protocol suffers from the 1-affects-n phenomenon if a single membership change in the group affects all the other group members. The single membership change requires that all group members commit to a new symmetric encryption key.
3. Service availability: the failure of a single entity in the key management architecture must not prevent the operation of the whole multicast session.

In the proposed system, the broadcast controller generates a random sequence using elliptic curves. The generated sequences are send to the next level in the hierarchy i.e. to group controllers from whereby another set of random sequences are created by the group controllers and communicated to the users by using the principle of point addition.

4.2.3 The method and implementation

For the implementation of our algorithm needs a sextuple (P, a, b , B , n , h) p – a large proven prime of 256 bits long for higher security.

a and b are values in the elliptic curve

B – Base point, typically the generator point (Gx, Gy)

n – The maximum points on the specific curve, order of the group which forms the cyclic ring.

Step1: the BC assigns the x coordinate of the generator point as its own id (IDBC) and start generating points from the generator point and assign the x –coordinate as the id of each group(IDGC1,IDGC2...).

Step2: The BC randomly assigns generator points to each GC. The GC starts generating points from the generator point from BC and assign the x coordinate as the id of users under the particular group.

Step3: each GC creates their own key shares (KS) by XORing all the user ids of users under the group with its group id(IDGC1) and communicate to the BC.

Step4: Upon receiving all the key shares from all GCs, the BC computes the final share by XORing all the key shares with its own id.(KS1 ⊕ KS2 ⊕ ....KSn ⊕ IDBC).

The final keyshare with the BC is declared as the symmetric encryption/decryption key for broadcasting to all the groups. The individual key shares are used as the symmetric key for the groups concerned.

4.2.4 Hierarchy of Keys Approach

The proposed method follows the strategy of hierarchy of keys approach with Broadcast Controller on the top of the hierarchy and Group Controllers in the next level and Legitimate Users in the bottom most level.

4.2.4.1 Hierarchy of broadcasting

4.2.4.2 Broadcast Controller

Broadcaster selects the generator points -Gx1, Gx2, Gx3 .... depends on the number of groups Whenever Group Controller requests for a group, the Broadcast Controller gives the generator points. The broadcast group uses separate generator points assigned by the BC depending on the number of groups-

4.2.4.3 Group Controllers

Group1: the BC perform point-double (Gx1) - group1 id is P1=(x- coordinate of the calculated point). The BC assigns another generator point (Gx2) to group1.

Group2: the BC perform point-add (Gx1,P1)- group2 id is P2=(x- coordinate of the calculated point). The BC assigns another generator point (Gx3) to group2.

G1—POINT_DOUBLE (Gx1) -P1- SELECT Gx2

G2--- (POINT ADD (Gx1,P1) -P2- SELECT Gx3

G3--- (POINT ADD (P1, P2) -P2- SELECT Gx3

(Gx2....Gxn)-generators for n groups

4.2.4.4 Individual Users

GC calculates the points from the generator point assigned by the BC.GC calculates the keyshare1 from that group and communicated to BC, likewise all other GC's calculates their own keyshares and send to Broadcast Controller.

4.2.4.5 DEK – Symmetric key generation using the points on elliptic curve

users under G1
U1ID—x-coordinateof (POINT DOUBLE (Gx2))-P11
U2ID--- x-coordinateof (POINT ADD (Gx2,P11) -P12
U3ID--- x-coordinateof (POINT ADD(P11,P12) -P13
UnID--- x-coordinateof(POINT ADD(P1n-2,P1n-1)-P1n
key share of group1—P1$\oplus$ P11$\oplus$ P12$\oplus$ P13 .... $\oplus$P1n
users under G2
U1ID—x-coordinateof(POINT DOUBLE(Gx3))-P21

U2ID--- x-coordinateof(POINT ADD(Gx3,P21)-P22

U3ID--- x-coordinateof(POINT ADD(P21,P22)-P23

UnID--- x-coordinateof(POINT ADD(P2n-2,P2n-1)-P2n

key share of group2—P2$\oplus$ P21$\oplus$ P22$\oplus$ P23 .... $\oplus$P2n

key share of group3—P3$\oplus$ P31$\oplus$ P32$\oplus$ P33 .... $\oplus$P3n

BC gets all the keyshares from all the GCs and finally calculate the final keyshare which is the symmetric encryption key or Data Encryption Key (DEK).

KS1= Key share from group1

i.e KS1= P11$\oplus$ P12$\oplus$ P13.... $\oplus$P1m

KS2 = Key share from group2

i.e. KS2 = P21$\oplus$P22$\oplus$P23.... $\oplus$P2n

final key share for single broadcast

DEK = BCID$\oplus$KS1$\oplus$KS2… $\oplus$KSn

The proposed system generates a Data Encryption Key and the conference key protocol implemented to receive the DEK effectively to multicast groups and multicast users.

1. Key Encryption Key (KEK) --- using Public key Broadcast Encryption method – Elliptic curve cryptography.

2. Data Encryption Key (DEK) --- using Symmetric key Broadcast encryption method – Elliptic curve point Generation.

DEK --- BCID$\oplus$ KS1$\oplus$ KS2...$\oplus$ KSn

In this method, the computation is not fully loaded with the central key server; instead the user identity calculation is done with the corresponding group controllers and communicated to users. Each group controller has to calculate their key share which is to be communicated to the central server and the Broadcast controller has to calculate the final key share or Data Encryption Key(DEK).

When a new member joins the system, only the corresponding group controller has to issue an identity value and execute the conference key protocol. The other users' identity remains the same, but the conference key protocol has to be re-executed for the updated key share.

When a user is revoked from the system after the expiry of its valid period, the particular group controller has to inform the identity of the user to be revoked, so that the broadcaster has to recalculate the key share of that particular group and recompute the new final key share and communicate to all. The identities of all the existing identity value remains the same, but the final symmetric key should be updated.

Either of the two different approaches can be used to communicate the DEK to the multicast group or multicast users in a group.

Approach 1: Implementation using Conference key protocol

The group conference protocol is implemented in BC to ensure group multicast. Another user conference protocol is implemented in GC to ensure user level multicast.

Approach 2: generation of Key Encryption Key(KEK)

The generation of Key Encryption Key is using the principle of Elliptic Curve Cryptography. The security of ECC is based on the assumption of hard problem analogous to discrete

logQ=kP, where Q,P belong to a prime curve.

given k,P then “easy” to compute Q given Q,P then “hard” to find k,known as the Elliptic Curve Discrete Logarithm Problem(ECDLP) provided k must be large enough.

In this approach, the DEK is encoded into a point on the elliptic curve with the value k kept as secret by the BC. The value k must be communicated to the users using the secure means of communication.

In this method, even though the number of unicast and multicast is less compared to approach1, the value k must be communicated to everyone securely. Also the value of k must be changed frequently to prevent the collusion.

In approach2, the BC encrypts the DEK using KEK and encapsulated it into the header

4.2.5 Join Protocol

Step1: On receiving the registration request along with the service request, by the LU, the BC accepts the registration request and based on the service request, the request is being forwarded to the concerned GC.

Step2: The GC accepts the request and assigns the identity to the LU. Whenever a member joins, the keyshare is re-computed.

Step3: the key share computation is initiated by the particular GC by sending IDCG to one of the LUs as a random choice.

Step4: The LU who receives the GCID invokes the conference key protocol.

Step5: When the conference reaches to zero, the corresponding LU multicast the DEK to all other LUs

End

4.2.6 Leave protocol

Step1 : when the subscription period is expired, the GC computes the new keyshare by XORing all the identities except the revoked user's identity and the new DEK is encrypted by KEK.

Step2 : the GC broadcasts the data with DEK and encrypted content is put into the payload.

Step3: the GC encrypts the DEK with KEK and encapsulated into the header.

Step4: the GC should securely communicate the KEK over a secure channel.

End.

5. BROADCAST ENCRYPTION SCHEME

is a quintuple (setup,sym_groupkeygen,sym_userkeygen, encrypt, decrypt)

Setup (): choose a large prime p during the setup phase, the seurity parameter(param) is set for the negotiated symmetric key size. It outputs the generators of the elliptic curve, which

satisfies the a and b value and the predefined set of group identities.

Sym_groupkeygen(): for the subset of groups , the corresponding the group controller generates the key values depending on the number of users in each group. It outputs the group key share (KS1,KS2...KSn).

The final key share will be computed by the BC called Data Encryption Key (DEK)

Sym_userkeygen(): it outputs a set of ids (id1,id2...idn)for all the users in a group.

Encrypt (): encrypt data using the key generated using the chosen encryption method and inserted into the payload.

Decrypt (): using the conference key protocol, the DEK is identified and decrypts the plaintext.

6. SECURITY ANALYSIS

The security analysis of the proposed scheme ensures the security of the identities issued. Also analyses the two important protocols,- join and leave protocol – backward secrecy and forward secrecy respectively. The security of the system is due to the hardness of Discrete Logarithm of Elliptic curves (ECDLP).

Theorem 1: The identities derived using the point generation of elliptic curve over finite field in the proposed protocol is indistinguishable in polynomial time from random numbers.

Proof: In elementary steps of the protocol, the identity generation is by calculating the next points from the previously derived point starting from the generator points of BC as well as the GC. Even though the generator point is known, it is difficult to guess the identities because of ECDLP.

Theorem 2: The join protocol of the proposed protocol satisfies the properties of backward security.

Proof: On receiving the join request, the broadcaster assigns the GC based on the requested service, and GC always calculate the next point on the curve for generating the identity for a new user or for a re-newing user without any collision, as the elliptic curve is chosen in such a manner from where infinitely many points can be generated. Hence there can be no collision occur during the identity generation process. An adversary is unable to guess the identity of LU , also the re-newed user is issued with new key after renewal and hence both of them are unable to guess in the proposed join protocol which ensures backward secrecy because of the property of elliptic curves.

Theorem 3: The leave protocol of the proposed system satisfies the properties of forward security.

Proof: When the subscription period of an LU expires, the group controller initiates the new approach of sending data encrypted with DEK and DEK encrypted by KEK. The encrypted DEK is encapsulated in the packet header. Because of the intractability of ECDLP , the adversary or the LU after the expiry of the period cannot guess the value of KEK , which ensures forward secrecy.

7. CONCLUSION

It is hereby summarize with some major contributions of the generation and communication of Contributory DEK.

1. The identities are generated from the principle of point addition and doubling from the generator point of elliptic curves . There is no need to maintain the identities already

assigned in this method for ensuring forward secrecy and backward secrecy. Hence storage requirement is very less.

2. Computation complexity is less since the contributory key generation involves the XOR operation only. No processing

overhead is incurred at each user, when a new member joins the group or when a new user(s) leaves the group.

3.The strength/randomness of the identity values depends on the Discrete Logarithm Problem of Elliptic curve.

4.The communication overhead is negligible since the updated DEK is communicated by the conference key protocol which involves n unicast followed by one multicast mode of communication. The conference key protocol is initiated only when the data are to be communicated to the group. For each member join/leave the updated DEK is not communicated to the LU immediately, which reduces the communication overhead.

5.Conference key protocol uses smaller key updates. Most notably, the message pair originated by the BC or the Group Controller the GC's or to the LU's do not require any data broadcast to current group members. A single round is required to compute the DEK without revealing the ID's to other members in the group.

6.The proposed method suffers from 1-affects-all phenomenon, but for every join /leave membership immediate update of re-key is not proposed here.

7.The computation complexity involved in re-keying is also negligible, since it involves only XOR operation and no communication overhead after re-keying.

The method experiences a delay to compute the contributory DEK by the receiver set and subsequently multicast the key to the members, in order to decrypt the data. Also the identities to LU’s are assigned from the x coordinate of the point on the elliptic curve, the randomness depends on the length of the generator point chosen. This scheme has chosen different approaches when a user(s) joins the system and when the user(s)leaves the system .

8. ACKNOWLEDGMENT

I would like to express my profound gratitude to Research Guide Dr.(Prof.) M Abdul Rahman for his abiding support.

9. REFERENCES

[1] Shanyu Zheng, David Manz, and Jim Alves-Foss. ***"A communication computation efficient group key algorithm for large and dynamic groups"***. Comput. Netw., 51(1):69–93, January 2007.

[2] Jim Alves-Foss. ***"An efficient secure authenticated group key exchange algorithm for large and dynamic groups".*** In IN PROC. 23rd NATIONAL INFORMATION SYSTEMS SECURITY CONFERENCE, pages 254– 266, 2000.

[3] Yongdae Kim, Adrian Perrig, and Gene Tsudik. ***"Group key agreement efficient in communication"***. IEEE Transactions on Computers, 53(7):905–921, 2004.

[4] Abhimanyu Kumar, Sachin Tripathi ,Priyanka Jaiswal ***"Design of Efficient ID-Based Group Key Agreement Protocol Suited for Pay-TV application",*** International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2015.

[5] Adi Shamir. ***"Identity-based cryptosystems and signature schemes"***. In Advances in cryptology, pages 47–53. Springer, 1985.

[6] Yacine Challal, Hamida Seba , *"**Group Key Management Protocols: A Novel Taxonomy",***International journal of Information Technology, volume 2, number 1, issn:1305-2403 , 2005

[7] Sandro Rafaeli and David Hutchison."*A **Survey of Key Management for Secure Group Communication***". ACM Computing Surveys . 35, 3, 309-329, 2003.

[8] G. H. Chiou and W. T. Chen. *"**Secure Broadcast using Secure Lock***". IEEE Transactions on Software Engineering, 15(8):929– 934, 1989.

[9] Behzad Malek , and Ali Miri, ***"Adaptively Secure Broadcast Encryption with Short Ciphertexts",*** International Journal of Network Security, Vol.14, No.2, PP. 71-79, 2012.

[10] M. Burmester and Y. Desmedt, *"**A secure and efficient conference key distribution system**,*" Advances in

[11] Cryptology: Eurocrypt'94, Springer-Verlag, LNCS 950, pp. 275–286, 1995. A. Perrig, D. Song, and J. D. Tygar, ***"ELK, a new protocol for efficient large-group key distribution,"*** IEEE Security and Privacy Symposium, pp. 247–262, 2001.

[12] R.Barua,R.Dutta, and P.Sarkar, ***"Extending JOUX protocol to multi party key agreement"*** Advances in Cryptology: INDOCRYPT'03, Springer-Verlag, LNCS 2904, pp. 205– 217, 2003.

[13] K. Y. Choi, J. Y. Hwang, and D.H.Lee, ***"Efficient ID-based group key agreement with bilinear maps,"*** Proceedings of Public Key Cryptography, Springer-Verlag, LNCS 2947, pp. 130– 144, 2004.

[14] D.Boneh, C. Gentry, and B. Waters, *"**Collusion resistant broadast encryption with short ci-phrertexts and private keys,"*** Advances in Cryptology: CRYPTO'05, Springer-Verlag, LNCS 3621, pp.258–275, 2005.

[15] C. Gentry and B. Waters, ***"Adaptive security in broadcast encryption systems (with short ciphertexts)"*** , Advances in Cryptology: EUROCRYPT'09, Springer-Verlag, LNCS5479,pp.171–188,2009

10. AUTHOR PROFILE

Deepa S Kumar , Associate Professor & HOD , Department of CSE , College of Engineering Munnar, received the M Tech Degree from Cochin University of Science & Technology..She is currently pursuing the PhD degree in Broadcast Encryption from Karpagam University, Coimbatore. She has presented papers in various National and International conferences and authored a journal in the area of Chaotic Cryptography. Her area of specialization includes cryptography, broadcast encryption schemes.

Dr. (Prof.) M. Abdul Rahiman , currently the Pro-Vice Chancellor of APJ Abdul Kalam Technological University , Kerala, was the former Director, All India Council for Technical Education (AICTE), the apex body of technical education under Ministry of Human Resource Development, Government of India which regulates Engineering, Management, Architecture, Hotel Management, Pharmacy institutions of the country. He received the Doctor of Philosophy (Ph.D.) degree in Computer Science & Engineering from Karpagam University. He obtained his Master of Technology from Kerala University in 2004, and Bachelor of Technology from Calicut University 1998. He achieved Post Graduate Diploma in Human Resource Management from Kerala University in 2006 & Master of Business Administration (MBA) in 2008.