Computationally effecient group re-keying for time sensitive applications

Deepika Rani K[1] & G. Praveen Babu[2]
JNTUH, SIT, Kukatpally, Hyderabad-500028, India.

Abstract—
Key distribution is an important problem for secure group communications. Multicast is an efficient means of distributing data in terms of resources usage. All the designated receivers or members in a multicast group share a session key. Session keys shall change dynamically to ensure both forward secrecy and backward secrecy of multicast sessions. The communication and storage complexity of multicast key distribution problem has been studied extensively. We implement a new multicast key distribution scheme whose computation complexity is significantly reduced. Instead of using conventional encryption algorithms, the scheme employs MDS codes, a class of error control codes, to distribute multicast key dynamically. This scheme drastically reduces the computation load of each group member compared to existing schemes employing traditional encryption algorithms. Easily combined with any key-tree-based schemes, this scheme provides much lower computation complexity while maintaining low and balanced communication complexity and storage complexity for secure dynamic multicast key distribution.

Keywords-distribution, multicast, MDS codes, computation complexity, erasure decoding.

1. Introduction

Key Management is one of the security services required by many Group oriented and distributed applications .In such applications data can be communicated using a secure group key, which helps in key distribution techniques. Multicast is an essential mechanism to achieve scalable information distribution for group-oriented applications. Multicast refers to communication where information is sent from one or more parties to a set of other parties in terms of resource (such as network bandwidth, server computation and I/O load) usage. In this case, information is distributed from one or more senders to a set of receivers, but not to all users of the group. The advantage of multicast is that, it enables the desired applications to service many users without overloading a network and resources in the server.

Security is provided when data is transmitting through an insecure network. Unicast security has several schemes to provide the issues which cannot be extended directly to the multicast environment. As the transmission takes place over multiple network channels, multicasting is more vulnerable than unicasting. In many applications, the multicast group membership changes dynamically, i.e., some new members are authorized to join a new multicast session while some old members should be excluded.Inorder to ensure both forward secrecy and backward secrecy ,session keys are dynamically changed The forward secrecy is maintained if an old member who has been excluded from the

current session can *not* access the communication of the current session, and the backward secrecy is guaranteed if a new member of the current session can *not* recover the communication of past sessions. This requires each session need a new key that is only known to the current session members, i.e., session keys need to be *dynamically* distributed to authorize session members. Group key management is the major issue in multicast security, which is the fundamental technology to secure group communication by generating and updating secret keys. Access control and data confidentiality can be facilitated using key management by ensuring that the keys used to encrypt group communication are shared only among the legitimate group members and only those members can access the group communication .The shared group key can be used for authentication and also for encrypting the message from a legitimate group member.Inorder to prevent these problems in secure multicast

Communication, the following two security criteria are used. Forward Secrecy: It is maintained if an old member who has been evicted should not be able to access the messages from the current and future sessions. Backward secrecy: It is guaranteed if a new member of the current session cannot recover the communication data of past sessions. The process of changing the session key and communicating the same to only the legitimate group members is called as Re keying.Group key management schemes are of three types. Centralized key management: group members trust a centralized server, referred to as the key distribution center (KDC), which generates and distributes encryption keys. Decentralized scheme: the task of KDC is divided among subgroup managers. Contributory key management schemes: Group members are trusted equally and all participate in key establishment.

In this paper, we study how a multicast group key can efficiently be distributed in computation. In this a centralized key management model is used where session keys are issued and distributed by a central group controller (GC), as it has much less communication complexity, when compared to distributed key exchange protocols. The group controller uses the communication, computation and storage resources for distributing the session key to the group members. The main problem here is how the resources can

be used to distribute the session key, which is referred to as group key distribution problem. There are two approaches that are generally used for distributing the session key to the group of n members. The first approach is that the group controller GC shares an individual key with each group member. That key is used to encrypt a new group session key. In the second approach the group controller shares an individual key with each subset of the group, which can then be used to multicast a session key to a designated. Subset of group members. This approach has less communication, computation and storage complexity when compared to the other approach.

A multicast group with large number of members uses the key-tree-based approach. This approach decomposes a large group into multiple layers of subgroups with smaller sizes. Using this approach communication complexity is reduced, but the storage and computation complexity is increased.

In this paper, the main aim is to reduce the rekeying cost. A new novel approach for computation efficient rekeying for multicast key distribution is introduced, which reduces the rekeying cost by employing a hybrid group key management scheme .It also maintains the same security level without increasing the communication and storage complexity. In this scheme, session keys are encoded using error control codes. In general encoding and decoding using error control code reduces the computation complexity. Thus, the computational complexity of key distribution can be significantly reduced.

2. The basic scheme (dynamic key distribution using maximum distance separable codes)

2.1. Maximum Distance Separable Codes

Block codes that achieve equality in Singleton bound are called **MDS (maximum distance separable) codes**. Examples of such codes include codes that have only one codeword (minimum distance n), codes that use the whole of $(F_q)^n$ (minimum distance 1).

Maximum Distance Separable (MDS) codes are a class of error control codes that meet the Singleton bound Letting GF(q) be a finite field with q elements, an (n, k) (block) error control code is then a mapping from $GF(q)^k$ to $GF(q)^n$: E(m) = c, where m = m1m2 _ _ _mk is the original message block, c = c1c2 _ _ _ cn is its code word block, and E(.) is an encoding function, with k <= n. If a decoding function D (.) exists such that D(ci1ci2 _ _ _ cik; i1; i2; _ _ _ ; ik)= m for 1 <= ij <= n and 1 <= j <= k, then this code is called an (n,k) MDS code. For an (n, k) MDS code, the k original message symbols can be recovered from any k symbols of its code word block. The process of recovering the k message symbols is called erasure decoding. All the symbols are defined over GF(q), and usually, $q = 2^{m.}$ The well-known Reed-Solomon (RS) codes are a class of widely used MDS codes. The RS codes and other MDS codes can be used to construct secret-sharing and threshold schemes.

2.2 Maximum Distance Separable Codes Algorithm

It mainly consist of three parts, they are as follows:
a) Initializing Group controller.
b) Subscribing new members.
c) Applying the procedure of Re-Keying whenever member leaves the group.

Steps for the Algorithm:

Step I : GC Initialization by constructing codeword C using MDS.
Step II : Applying One-Way Hashfunction
Step III : H(x)=y, propoerty of Hashfunction
Step IV : Subscribing new member
Step V : Ji= +ve integer ji!=jk
Step VI : Select Si
Step VII : Applying the procedure of Re-Keying whenever member leaves the group.
Step VIII: Cj=H(Si+r)
Step IX : Member j every 'n' members in the group calculates these own codeword C1 C2.....Cn

Fig.1: *MDS Code Algorithm*

3. Proposed system

In our paper we proposed four modules:
a. Group Controller.
b. Client.
c. Group Key Generation.
d. Re-Keying.

A. Group Controller

The group controller is the center of the system. It acts as the server for the system. The GC distributes session key to group members, which can then be used to multicast messages to a designated subset of group members.

Multicasting is a process of sending a message to a selected group. Internet applications, such as online games, newscast, stock quotes, multiparty conferences, and military communications can benefit from secure multicast communications. In most of these applications, users typically receive identical information from a single or multiple senders. Hence, grouping these users into a single multicast group and providing a common session encryption key to all of them will reduce the number of message units to be encrypted by the senders. Various types of data communication are broadcast, Multicast, group communication.

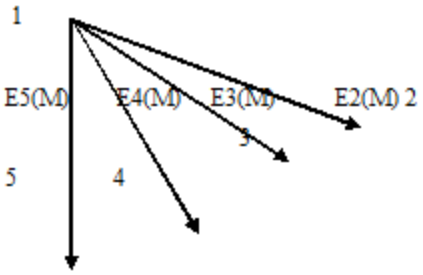

Fig.2: Transmission of the message M through 4 point-to-point connections

Fig.2 shows the transmission of message m to four point to point connections. Here node number 1 is the service provider. Nodes 2,3,4,5 are the receiving nodes. Nodes 2,3,4,5 are receiving the same message.

A. (i) Group communication

For group communications, the server distributes to each member a group key to be shared by all members of the group, distributing the group key securely to all members requires messages encrypted with individual keys (a computation cost proportional to group size). Each such message may be sent separately via unicast. Alternatively, the messages may be sent as a combined message to all group members via multicast. Either way, there is a communication cost proportional to group size (measured in terms of the number of messages or the size of the combined message). Observe that for a point-to-point session, the costs of session establishment and key distribution are incurred just once, at the beginning of the session. A group session, on the other hand, may persist for a relatively long time with members joining and leaving the session. Consequently, the group key should be changed frequently. To achieve a high level of security, the group key should be changed after every join and leave so that a former group member has no access to current communications and a new member has no access to previous communications.

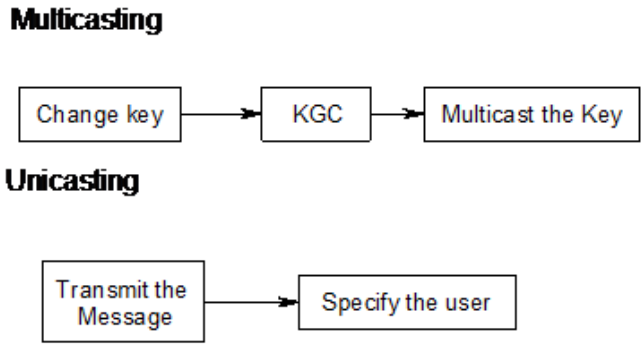

Fig.3: Multicasting and unicasting in group communication

B. Client

Whenever a new member is authorized to join the multicast group for the first time, the GC sends it (using a secure unicast) a session key. Once a session key is distributed to the group, any member can calculate the secret information that other members in the same group hold.The Login Module is used for the Newly joined users to send a request to the Group Controller and it is used for to retrieve the Private keys after the Group Controller assign keys to the new users. The user login the group to enter the user Id and Private Key. If the user Id and private key is correct means the user view the inbox and outbox message otherwise to display the message box "Enter the correct Password".

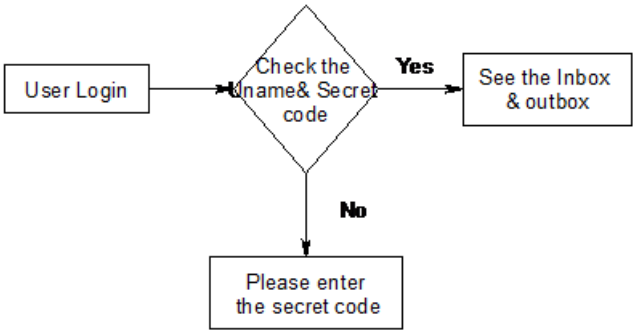

Fig.4: User login

C. Group Key Generation

In cryptography, a group key is a cryptographic key that is shared between groups of users. Typically, group keys are distributed by sending them to individual users, either physically, or encrypted individually for each user.

A common use of group keys is to allow a group of users to decrypt a broadcast message that is intended for that entire group of users, and no-one else. Various group users share their own session key. They can transfer the secret or private information within their groups.The group keys acts as session keys for each groups. It changes whenever a user joins a group or leaves a group.

C. (i) Private Key

The Private Key is generated using MDS code. The GC (Group Controller) sends his number of group members to the KGC (Key Generation Center). The keys are generated by the KGC and submitted to the GC.

C. (ii) Session Key

In session key generation, initially sixteen decimal digits are generated by using random number generation method .Then each decimal digit is splited and compared with pre determined binary format.In DES algorithm the 64 bits session key is considered as a message file and generated user's private key is considered as a key file. DES algorithm encrypts the session key by using user's private key and transmitted to the appropriate users.

C. (iii) Join operation

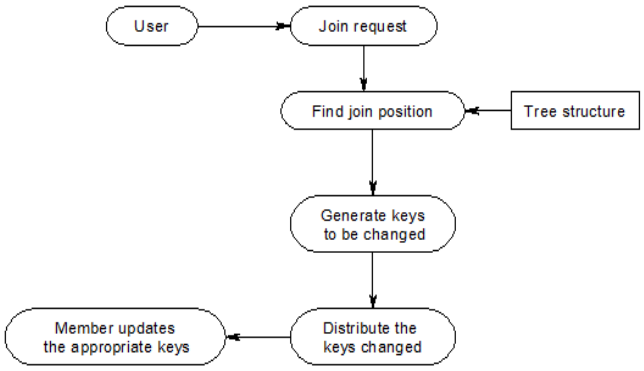

Fig.5: *Diagram for Join Operation*

C. (iv) Join Request

A Network node issues a request to GC to join the group. The GC check whether the request is from an authenticated member, if yes the GC accepts the request. The node then communicates its session key through some secure channel.

C. (v) Find join position

The group controller maintains a tree structure. The tree structure is the logical arrangement of members. The GC traverses the tree structure and finds a position for the new member. The GC(Group controller) inserts the member details in this new position, which is a leaf node.

C. (vi) Generate keys

From the new position onwards the GC generates the new key(s) along the path to root. The new keys are used to replace the old keys of the auxiliary nodes.

C. (vii) Update tree structure

Old keys are replaced by their corresponding new keys Henceforth newly generated keys are used for future communication. This operation provides backward secrecy, i.e. it prevents the newly joined member from accessing the previously communicated data.

C. (viii) Distribute keys

A packet is constructed, which consists of newly generated key(s) This packet is encrypted using the old key known by a member or sub-group of members.

C. (ix) User-oriented re-keying

In the user-oriented re keying, the group controller constructs each re keying message, rekey message contains the encrypted form of session key. So that they contain exactly all the messages that some user or a group of users need.

C. (x) Key-oriented re-keying

Key-oriented strategy emphasizes that each new key should be packed into a separate message and distributed to the holders

Leave operation

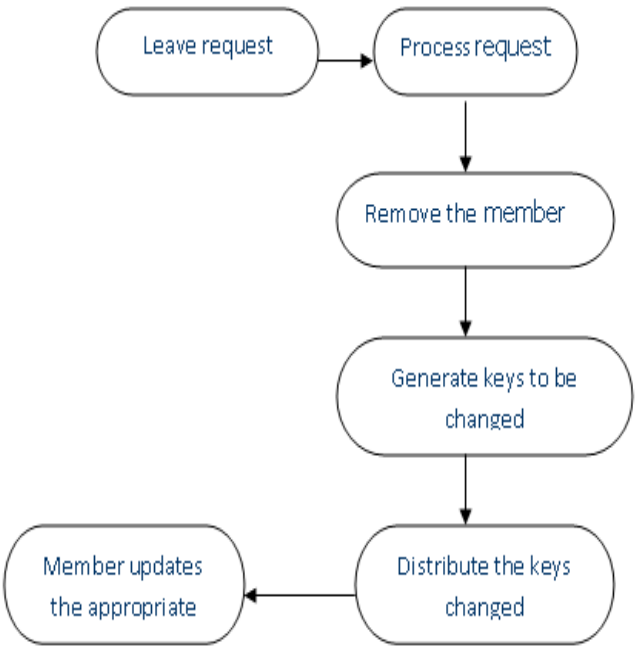

Fig.6: *Diagram for Leave Operation*

C. (xi) Leave Request

The member issues a request to leave the group.

C. (xii) Process Request

The GC checks whether the request is from an existing member, if so the GC accepts the request.

C. (xiii) Find leave position

The GC traverses the tree structure and finds the leaving position of the member. The GC then deletes the member details and removes the node from tree structure.

C. (xiv) Generate keys

From the leaving position onwards the GC generates the new key(s) along the path to root. Old keys are replaced by their corresponding new keys. Henceforth newly generated keys are used for future communication. This operation provides forward secrecy, i.e. it prevents the left member from accessing the data sent in future communication.

C. (xv) Distribute keys

A packet is constructed, which consists of newly generated key(s). This packet is encrypted using the old key known by a member or sub-group of members. These new keys help the members to decrypt the messages sent in future communication.

C. (xvi) Member updates keys

After receiving the message, the member updates the appropriate set of keys.

C. (xvii) User-oriented re-keying

In the user-oriented re keying, the group controller constructs each re keying message, re key message contains

the encrypted form of session key. So that they contain exactly all the messages that some user or a group of users need.

C. (xviii) Key-oriented re-keying

Key-oriented strategy emphasizes that each new key should be packed into a separate message and distributed to the holders.

D. Re-keying

Whenever some new members join or some old members leave a multicast group, the GC needs to distribute a new session key to all the current members. After an old member leaves, the GC needs to distribute a new key to n remaining members to achieve both forward and backward secrecy of the session key.

In the rekeying procedure, the GC needs to multicast a fresh random number r and (n-1) symbols of the new code word m2 ……mn. Each of the codeword symbols has mbits.The random number r is used to guarantee that the new session key is different from all the old keys used.

4. Design & analysis of the system

Design involves identification of classes, their relationships as well as their collaboration. In the Fusion method, some object-oriented approaches like Object Modeling Technique(OMT), Classes, Responsibilities, Collaborators(CRC),etc, are used. Objectory used the term "agents" to represent some of the hardware and software systems .

System Flow Diagram

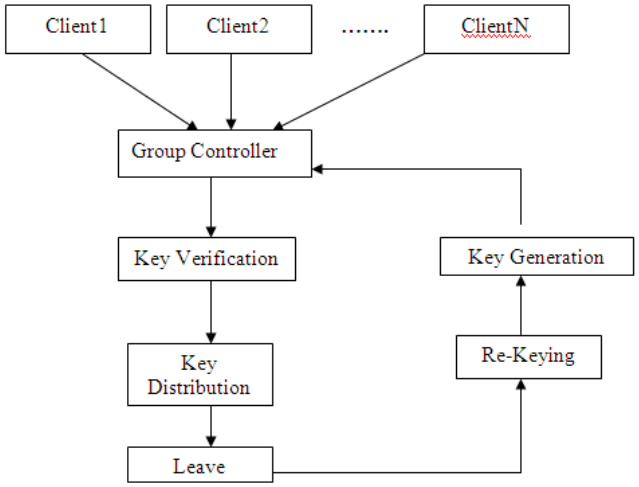

Fig.7: *System flow Diagram*

System Architecture

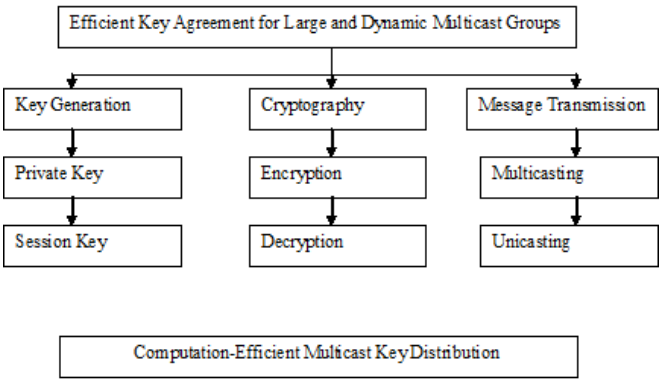

Fig.8: *System Architecture*

Usecase Diagram

Use case diagrams are behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

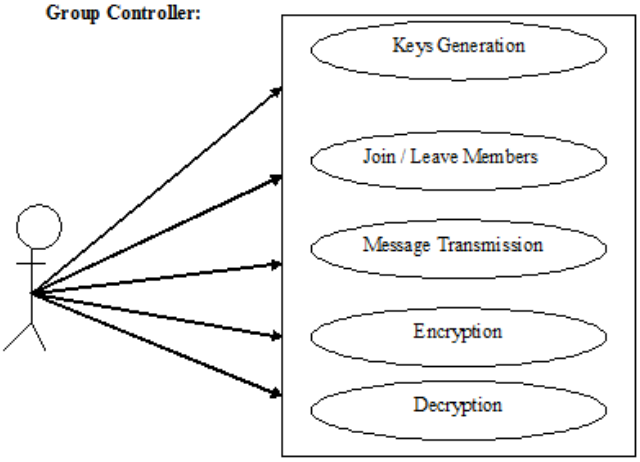

Fig.9: *Interoperability usecase diagram for Group controller*

Sample Code

```
public GrpCtrler(String
id,String sctkey)
{
super();
this.id = id;
g=id;
this.sctkey=sctkey;
initializeComponent();
this.setVisible(true);
setDefaultCloseOperation
(EXIT_ON_CLOSE);
}
```

Fig.10: *Sample code for group controller*

Fig.11: *Screenshot for sending group controller*

5. Results

The below are the obtained screen shots.

Fig.11: Screen shot for Login

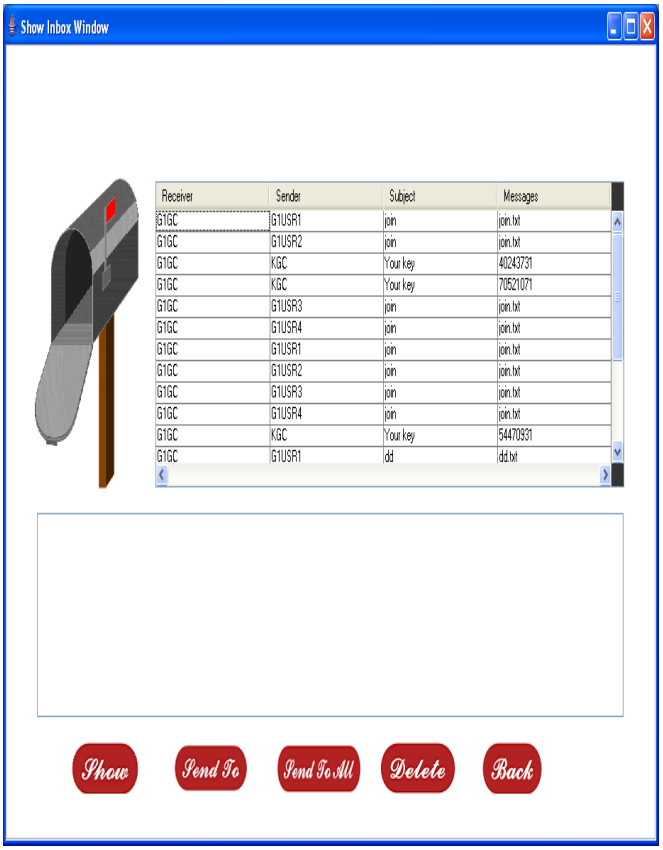

Fig.12 : *Screenshot for displaying inbox*

Fig.13: *Screenshot for acknowledging for message which was transmitted*

Conclusion

In this paper we can reduce the complexity of key distribution by using decoding of MDS codes as an alternative for costly encryption and decryption techniques. We concentrate on such a scheme which provides much lower computation complexity while maintaining low and balanced communication complexity and storage complexity for dynamic group key distribution.

Reference

[1] Peter S. Kruus and Joseph P. Macker, "Techniques and issues in multicast security," MILCOM98, 1998.

[2] Paul Judge and Mostafa Ammar**,** "Security Issues and Solutions in Multicast Content Distribution: A Survey", IEEE Network, February 2003, pp 30 – 36.

[3] M. Moyer, J. Rao and P. Rohatgi, "A Survey of Security Issues in Multicast Communications", IEEE Network Magazine, Vol. 13, No.6, March 1999, pp. 12-23.

[4] Yan Sun, and K.J. Ray Liu, "Securing Dynamic Membership Information in Multicast Communications," *IEEE INFO CONFERENCE 2004*.

[5] T.M. Cover and J.A. Thomas, Elements of Information Theory. John Wiley & Sons, 1991.

[6] S. Rafaeli and D. Hutchison, "A Survey of Key Management for Secure Group Communication," ACM Computing Surveys, vol. 35, no. 3, pp. 309-329, 2003

[7] F.J. MacWilliams and N.J.A. Sloane, The Theory of Error Correcting Codes. North- Holland Math. Library, 1977.

[8] H. Harney and E. Harder, Logical Key Hierarchy Protocol, IETF Internet draft, work in progress, Mar. 1999.

[9] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, Handbook of Applied Cryptography, fourth ed. CRC Press, 1999.

[10] M. Waldvogel, G. Caronni, D. Sun, N. Weiler, and B. Plattner,"The VersaKey Framework: Versatile Group Key Management,"IEEE J. Selected Areas in Comm., vol. 7, no. 8, pp. 1614-1631, Aug. 1999

Authors

Deepika Rani Kampally received Bachelor's degree in Computer science and Engineering from JNTUH, Pursuing M.Tech in Computer Science and Engineering from JNTUH. She is a research scholar in field of Information Security.

G. Praveen Babu is presently working as an Associate Professor of Computer Science & Engineering, at School of Information Technology, JNT University.He has a teaching experience of more than 9 years.Presently, pursuing Ph. D. In the area of Computer Networks.Areas of interest are computer Networks, Network Security, Analysis of Algorithms, Artificial Intelligence and Software Engineering Additionally, In-charge of the Placement and Training cell at School of Information Technology, JNTU, Hyderabad.