International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

# A Secure Group Sharing Framework Using TGDH Scheme

**Shahina K M**

[1]Department of Computer Science and Engineering, KMCT College of Engineering, Calicut

**Abstract:** ***Cloud computing is a kind of Internet computing mainly used for data storage and data sharing. Secure data sharing among the group members is also an important concept in cloud. But privacy and security of user's data are the major issues in cloud. And the semi trust nature of cloud service provider is also an important issue. Due to this semi-trusted nature of CSP traditional security mechanisms cannot be applied directly. . In this paper, we propose a secure group sharing framework using TGDH scheme for cloud computing, which can effectively take advantage of the Cloud Servers help without exposing any sensitive data to attackers and the cloud provider. This framework combines enhanced TGDH, proxy re-encryption, EIGamal signature generation and A-S algorithm together into a protocol. Enhanced TGDH scheme is used for the generation of dynamic group key pairs. Proxy re-encryption is adopted for ensuring forward and backward secrecy. Digital signatures are generated by the use of EIGamal algorithm with random key and group public key. A-S algorithm is used for data encryption. The security requirements for cloud based data sharing are fulfilled by our proposed scheme with high efficiency. And it can be proved by the extensive security and performance analysis.***

**Keywords:** A-S algorithm, EIGamal signature generation, Enhanced TGGDH scheme, Proxy re-encryption.

## 1. Introduction

Cloud computing can be simply defined as the practice of storing and accessing data and programs over the internet instead of using a local server's or a personal computer's hard drive. The word cloud means internet and so cloud computing can also be defined as a kind of internet computing that provides shared processing resources and data to computers and other devices on demand. Cloud is rising from recent advances in technologies such as hardware virtualization, web services, distributed computing, and utility computing. Cloud systems are used to enable data sharing capabilities and this can provide many benefits to the user. Cloud has a number of advantages such as minimal expenditure, access from innumerable options, On-demand self-service, broad network access, data centralization and rapid elasticity. Despite of above advantages, there still remain various challenging issues, among which the privacy and security of users data become two major issues.

In traditional system, the data owner stores his data in the trusted servers, which are generally controlled by a fully trusted administrator. Anyways the cloud is usually maintained and managed by a semi-trusted third party (Cloud provider), so he is responsible for selecting the key and encrypting the data before storing it into the cloud. So we cannot directly apply the traditional security storage mechanisms into cloud storage. In the second approach group leader is the privileged entity and all group members will send their data to the group leader. Group leader will select a random session key, encrypt the data and then store it into the cloud. Then group leader will distribute the encryption key to the N members by encrypting the key by using the public keys of all users. So this approach requires N+1 encryption and which leads to difficulty. For the past few years, many algorithms have been developed for data sharing in cloud. Most of them are based on one time password or a fixed key.

Proposed method is a complete group sharing frame work based on TGDH scheme. This framework combines Enhanced TGDH, Proxy re-encryption, A-S algorithm and EIGamal algorithm. In which a balanced binary tree having N leaf node is generated for the group. Where each leaf node represents a member and all nodes are associated with two keys, a public key and a private key. The keys associated with the root node are taken as the group key pair, i.e. group private and public key pair. These keys are calculated by using the Tree based Group Deffie-Hellman (TGDH) method.

The TGDH protocol in [9] uses a TGDH key tree based on Decisional Diffie Hellman problem. A TGDH key tree is an adaptation of balanced binary tree having N leaf nodes, were N is the number of members in the group. And this tree is used for the key calculation of the group. This group keys are dynamical in nature.

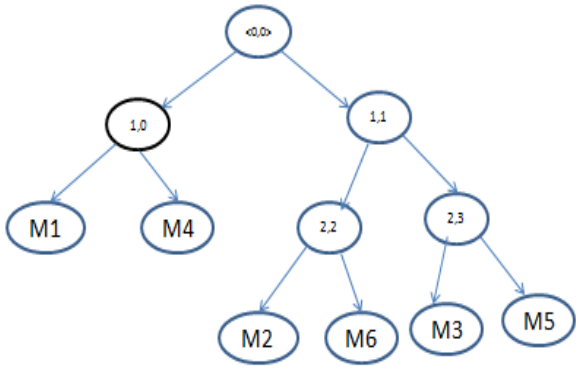


**Figure 1:** A TGDH key Tree with 6 nodes

Figure shows an example tree, in which each node ⟨l,v⟩ is associated with a secret key K⟨l,v⟩ and a blinded key $BK_{\langle l,v\rangle}$

$$BK_{\langle l,v\rangle} = g^{K\langle l,v\rangle} \bmod p.$$

$$K_{\langle l,v\rangle} = BK_{\langle l+1,2v+1\rangle}^{K\langle l+1,2v\rangle} \bmod p$$

$$= BK_{\langle l+1,2v\rangle}^{K\langle l+1,2v+1\rangle} \bmod p.$$

The key pair at the root node (K⟨0,0⟩ and BK⟨0,0⟩) is the assigned group key pair (group public key PuKG and group private key PrKG) shared by all group members: PuKG = K⟨0,0⟩ and PrKG = BK⟨0,0⟩. Each group member is associated with a leaf node, whose security key is randomly selected. Based on the TGDH protocol, Each group member Mi at the leaf node ⟨l,v⟩ knows all publicly shared blinded keys of sibling nodes of all nodes in the path from ⟨l,v⟩ to ⟨0,0⟩ and can compute all secret keys of nodes in the path. For example in above figure, M2 knows his secret key K⟨3,0⟩ and the blinded keys broadcasted by other group members: BK⟨3,0⟩, BK⟨3,1⟩, BK⟨2,3⟩. Therefore, M2 can compute the key pairs of nodes ⟨2,2⟩, ⟨1,1⟩ and ⟨0,0⟩.

Group members can upload the data in encrypted form into the cloud. Data is encrypted symmetrically by using the A-S algorithm. The random key used for encryption is asymmetrically encrypted by using EIGamal with algorithm the group public key. Then both encrypted data and digital signature is stored in the cloud. Any member of the group can decrypt the digital signature by using the group private key. Then he will get the random key and by using that random key, he can decrypt the data. And proxy re-encryption technique is used for ensuring forward secrecy.

## 2. Related Works

The conventional approach to address the problem of providing security to the data shared among the group in cloud is to use cryptographic encryption mechanisms, and store the encrypted data in the cloud. Authorized users can download the encrypted files and decrypt them with the given keys. But in this scenario, how to distribute and update session keys is one of the most important but hard problems. Digital Envelope[1] is used to address this task in [2], [3]: the data is encrypted with a randomly chosen session key by using symmetric encryption, and then the session key is encrypted with the public key of the specific user by using public-key encryption.

For example, we assume that the user A wants to securely send a file F to the user B. First, The user A chooses a random session key K, and uses a symmetric encryption algorithm (such as DES and AES) to encrypt the file FILE: {FILE}K. Then user A uses an asymmetric encryption algorithm (such as RSA) to encrypt the session key K: EPuKB(K) (PuKB is B's public key). Here, EPuKB(K) is named as a digital envelope, which can be transmitted in the open environment, and be decrypted only by the user B. However, in normal ways, if a file is shared to N specific authorized users, N digital envelopes are required to be generated. Therefore, the computing and communication overhead of generating digital envelopes is O(N) for one file. Meanwhile, the computational complexity and communication overhead of session key updating are both O(N). Moreover, we assume that one session key is required for each one sharing file. If the total number of shared files are M for N specific recipients, the overall overhead of digital envelope generation for all shared files is as large as O(MN).

There have been several other works[4], [5], [6], [7] on the privacy preserving data sharing issue in cloud based on various cryptographic tools, such as attribute based encryption (ABE)[8], proxy re-encryption[9] , etc. Among these existing schemes, in [4], Yu et al. have provided a fine-grained and scalable solution. The efficiency of Yu et al.'s scheme[4] relies on that there is high attribute variability between different files and high attribute variability between different users. The efficiency of the schemes in [5] and [6] depends on the assumption that Cloud Servers must be absolutely trusted. Otherwise, Cloud Servers can launch the collusion attack with some curious leaving group members. [7] has tried to realize an ABE and proxy re-encryption based data sharing scheme in mobile devices, which also has the problem mentioned in [5], [6].

Duc H. Tran [5] introduced a secure framework to efficiently share data among multi-users. This mechanism is based on proxy re-encryption scheme and which requires the encryption of data before sending it to the cloud. All the users encrypt their data by using the same public key and decrypt it by using different private keys. When a user makes a request for the data stored by another user, the proxy will pre-decrypt the data according to the requested user's private key before sending it to the requested one. A revoked user is prevented to access the data by simply avoiding the pre-decryption of data using his private key. Each request to data needs a re-encryption. If n members request for the same file, then the same file will be re-encrypted n times using private keys of each user. So this is difficult to share large number of files to different users.

Piotr K Tysowski and M. Anwarul Hasan[7] proposed a key management system for secure data outsourcing applications based on attribute-based re-encryption. Attribute-based encryption effectively permits authorized users to access secure content in the cloud based applications on the satisfaction of an attribute-based policy. The scheme has been modified in such a way that the data owner and a trusted authority cooperate in the key generation and encryption processes. Responsibility over key generation is divided between a mobile data owner and a trusted authority and the owner is relieved of the highest computational and messaging burdens, so mobile users can minimize their battery and wireless communication usage. Additional security is provided through a group keying mechanism were data owner controls access based on the distribution of an additional secret key, beyond possession of the required attributes.

A hybrid protocol is also used to allow message encryption based on a group key, allowing the user membership to be further refined for highly sensitive data. It also allows re-encryption to occur, and thus revocation become efficient without necessitating existing common remedies and their limitations. Thus this method is useful for securing mobile cloud computing with very large user populations.

Kan Yang[8] introduced a Privacy-Preserving Data Publish-Subscribe Service for Cloud-based Platforms. In public cloud, privacy issue becomes much more critical for data

publication and subscription service, as the cloud server cannot be fully trusted by both data publishers and data subscribers. Existing Attribute Based encryption allows the cloud server to evaluate whether user's attributes can satisfy the access policy. However none of the ABE schemes support the evaluation of both access policy and subscription policy. But the novel attribute based encryption known as Bi-Policy ABE supports both access and subscription policy.

## 3. Group Sharing Framework with TGDH Scheme

Public cloud provides an efficient platform for group data sharing. But ensuring privacy and security of group data stored in public cloud without exposing the private data into the semi trusted cloud provider or attacker is an important problem in cloud. Therefore a new framework is introduced, and this framework combines enhanced TGDH [10] [11], proxy re-encryption, AS algorithm [12] and EIGamal signature generation algorithm together. This frame work effectively take advantages of cloud server's help without exposing any sensitive data to attackers or the cloud provider.

This scheme supports the updation of group key pair during group members joining or leaving operation and which transfers most of the computational complexity and communication overhead to cloud server without exposing sensitive data. Enhanced TGDH scheme enables the group to negotiate and update the group key pairs when some of group members are online.

This frame work consists of mainly two types of users, a full privileged group leader and common group members. Both the leader and group member can upload and download data from the group by using group private key.

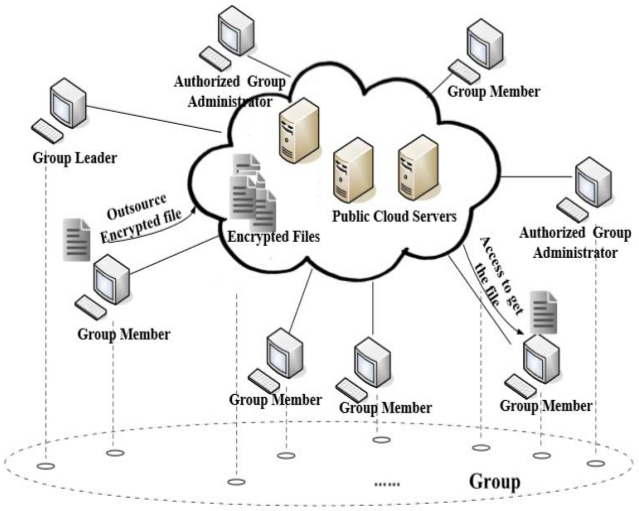


**Figure 2:** The Framework of secure group sharing

Figure 2 shows the framework of the proposed system. This frame work consists of mainly two types of users, a full privileged group leader and common group members. All the data are stored in the cloud. Both the group leader and group member can upload and download data to or from the group by using group key pairs.

This system consists of five modules. The first one is group initialization. In which the initial group is created with 2 members. The second module is new member joining, and this module is responsible for adding new members into the group. Third part is member leaving, and this is responsible for removing the requested member from the group. The last but important module is data sharing.

### 3.1 Group Initialization

In the initialization phase, group leader generates a shortest binary tree with 2 leaf nodes, with the assumption that initially there are only two members in the group. One node is assigned to the group leader and another one is for the other initial member .GL chooses a random security key for each node and computed the blind key by using Deffie-Hellman key exchange mechanism. Then GL uploads tree structure and related information to the cloud.

### 3.2 Data Sharing Management

This phase is responsible for uploading data to the cloud or downloading data from the cloud. Before uploading data, owner gives the semantic description of the file. Then symmetrically encrypts the file with randomly chosen session key by using A-S algorithm. Data owner also uploads a digital envelope, which is asymmetrically encrypted with group public key by using EIGamal signature generation algorithm.

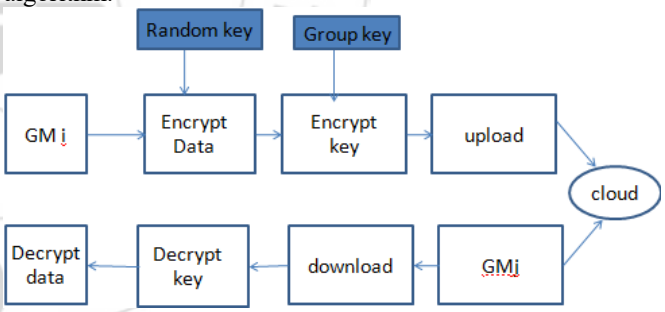


**Figure 3:** Block diagram for secure group sharing

Any member who wishes to download a file will send a request with corresponding file ID. Then he will get the encrypted data with digital signature. The member should decrypt the digital signature by using the group private key calculated from the TGDH tree structure. By this decryption he will get the random key used for the encryption of the corresponding file. And he can easily access the file through decryption.

A-S algorithm is used for data encryption. In which a random key value is selected first. ASCII value of each letter in the plain text will be XORed with the corresponding ASCII value of the key document. Corresponding character will be written in cipher document.

**Plain Text : GOOD MORNING**

| G | O | O | D | | M | O | R | N | I | N | G |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 71 | 79 | 79 | 68 | 32 | 77 | 79 | 82 | 78 | 73 | 78 | 71 |
| @ | s | [ | ; | | 6 | T | → Key | | | | |
| 64 | 115 | 91 | 59 | | 54 | 84 | | | | | |

**Cipher Text ← Plain Text ^ Key**

| 7 | 60 | 20 | 127 | 22 | 25 | 15 | 33 | 21 | 114 | 120 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| • | < | ¶ | ⌂ | — | ↓ | ☼ | ! | § | r | x | !! |

**Figure 4:** Plain Text Encryption

The decryption process is just reverse of the encryption. In which cipher text is EXORed with the corresponding values in key and the plain text is recovered from the corresponding ASCII values.

**Cipher Text : •<¶⌂—↓☼!§rx!!**

| • | < | ¶ | ⌂ | — | ↓ | ☼ | ! | § | r | x | !! |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 60 | 20 | 127 | 22 | 25 | 15 | 33 | 21 | 114 | 120 | 19 |
| @ | s | [ | ; | | 6 | T | → Key | | | | |
| 64 | 115 | 91 | 59 | | 54 | 84 | | | | | |

**Plain Text ← Cipher Text ^ Key**

| 71 | 79 | 79 | 68 | 32 | 77 | 79 | 82 | 78 | 73 | 78 | 71 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| G | O | O | D | | M | O | R | N | I | N | G |

**Figure 5:** Plain text decryption

### 3.3 Group Member Joining

When a group member joins, he will send a joining request to group leader (GL). Then GL tries to find a leaf node. At the same time new member randomly select a security key and get the BK of all sibling nodes from the cloud provider. Send all BKs from his node to the root node to GL. After receiving all BKs from the new member, GL uploads all this BKs to the Cloud Server. Then Cloud provider updates the tree structure and the corresponding BKs. Forward Secrecy should be guaranteed by using proxy re-encryption, when a group member joins, which ensures that the newly joined user can also access and decrypt the previously published data.

### 3.4 Group Member Leaving

When a group member leaves, GL should mandate leaving group member's position in the binary tree or delete the corresponding node and act as a sponsor to implement the group member leaving process. GL computes the proxy re-encryption key by combining old group private key with new group public key. Then cloud provider updates all existing digital envelopes and thus ensures backward secrecy.

## 4. Implementation and Analysis

The secure group sharing frame work using TGDH scheme is implemented using ASP.NET. Were Inbuilt function tree builder is used as part of tree generation. The key tree is generated for different no of users and key is calculated for different values of P.


**Figure 6:** Initial tree with two members

Here BK⟨1,0⟩=7 and BK⟨1,1⟩=10, then by using this values group key pairs are calculated as BK⟨0,0⟩=1 and K⟨0,0⟩=10.

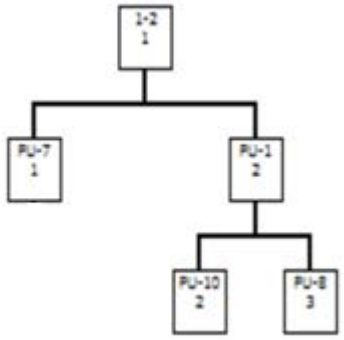


**Figure 7:** Tree after new member joining


**Figure 8:** File selection

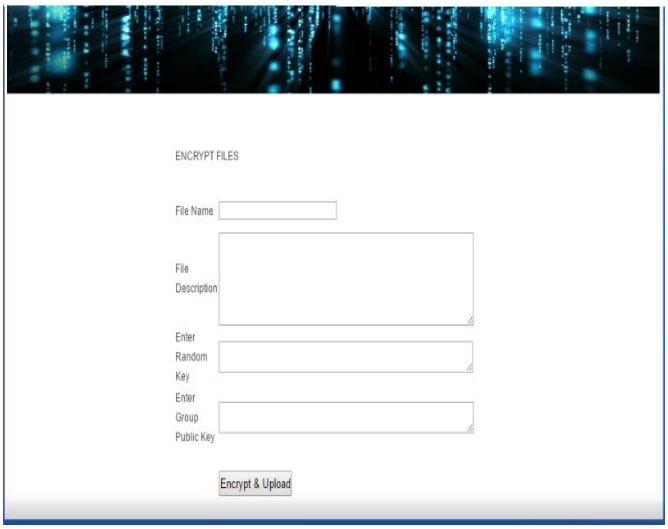


**Figure 9:** File upload


**Figure 10:** File download

By comparing with the existing similar methodologies, the proposed system performs more efficiently.

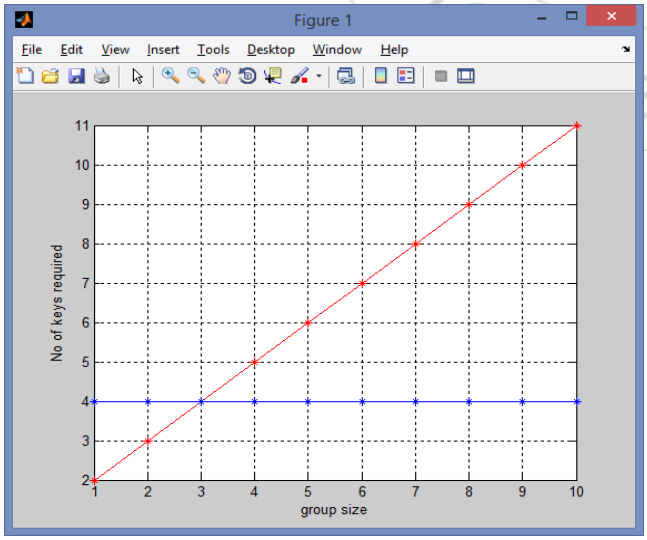


**Figure 11:** Comparison graph

A comparison graph is plotted with Group size (No of group members) against No of Keys required. The proposed system is compared with cryptographic encryption method. In which no of keys required for sharing a single file among the group members is directly proportional to the no of members in the group. In the proposed method requires only two keys for data sharing among all members. In the graph red coloured line indicates cryptographic method and blue coloured line indicate proposed method. This graph shows that the proposed method outperforms well.

## 5. Conclusion

A Secure Group Sharing Framework using TGDH Scheme performs well for sharing data among the group members securely. This framework is a combination of enhanced TGDH, proxy re-encryption, A-S algorithm and EIGamal signature generation algorithm. Tree based Group Diffie Hellman (TGDH) mechanism is used for group key generation and which is dynamic in nature. Forward secrecy is ensured during a new member joining into the group. Proxy re-encryption technique is used for ensuring forward secrecy. Backward secrecy is also ensured during the leave operation of a member from the group. TGDH scheme also provides a way to update the group keys when some of the group members are not online. Achuth Shankar (A-S) algorithm is used for the symmetric encryption of the file. The random key used for this symmetric encryption is asymmetrically encrypted with group private key by using EIGamal key generation algorithm. The proposed method provides better security and privacy for the user data and reduces the complexity of selecting large no of keys for the encryption of a single file. As a future work TGDH scheme can be extended to handle more than one group.

## References

[1] RFC2315, "PKCS #7: Cryptographic message syntax (version 1.5)," http://www.ietf.org/rfc/rfc2315.txt, Mar 2000.

[2] Y.Tang, P. Lee, J. Lui, and R. Perlman, "Secure overlay cloud storage with access control and assured deletion," IEEE Transactions on Dependable and Secure Computing, vol. 9, no. 6, pp. 903–916, 2012.

[3] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," Ieee Internet Computing, vol. 16, no. 1, pp. 69–73, 2012.

[4] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in IEEE INFOCOM 2010: Proc. The 29th Conference on Computer Communications. IEEE, 2010.

[5] L.Nguyen,W.Zha,andW.K.Ng,"Towardssecurity in sharing data on cloud-based social networks," in ICICS 2011: Proc. 8th International Conference on Information, Communications and Signal Processing. IEEE CS, 2011.K.Zuiderveld,"Contrast limited adaptive histograph equalization.," Graphic Gems IV, pp. 474–485, 1994

[6] W. Jia, H. Zhu, Z. Cao, L. Wei, and X. Lin, "SDSM: A secure data service mechanism in mobile cloud computing," in WKSHPS 2011: Proc. 2011 IEEE Conference on Computer Communications Workshops. IEEE CS, 2011, pp. 1060–1065.

[7] P. Tysowski and M. Hasan, "Hybrid attribute-and re-encryptionbased key management for secure and scalable mobile applications in clouds," IEEE Transactions on Cloud Computing, vol. 1, no. 2, pp. 172–186, 2013.
[8] Kan Yang, Xiaohua Jia "Privacy-Preserving Data Publish-Subscribe Service on Cloud-based Platforms", IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 1, pp. 131–143, Jan. 2013
[9] Y. Kim, A. Perrig, and G. Tsudik, "Tree-based group key agreement," ACM Transactions on Information and System Security (TISSEC), vol. 7, no. 1, pp. 60–96, 2004.
[10] Kaiping Xue ,Peilin Hong, "A Dynamic Secure Group Sharing Framework in Public Cloud Computing", IEEE Transactions on Cloud Computing (2014)
[11] Preeti Gulab Sonar, Vishakha Ashok Patil, "A Novel Approach for Secure Group Sharing in Public Cloud Computing", International Journal of Computer Applications (0975 – 8887) Volume 127 – No.11, October 2015
[12] 47 Aswin Achuthshankar , Aswathy Achuthshankar , "A Novel Symmetric Cryptography Algorithm for Fast and Secure Encryption " in IEEE Sponsored 9th International Conference on Intelligent Systems and Control (ISCO)2015

## Author Profile

**Shahina K M**, received her B.Tech Degree in Computer Science & Engineering, in year 2014, from UKF, College of Engineering, Kerala University. Currently she is pursuing her Masters of Technology in Computer Science & Engineering from KMCT College of Engineering, affiliated to Calicut University.