Journal of Strategic Security

Volume 15 | Number 1 Article 6

A Hierarchical Multi-Authority Access Control Scheme for Secure and Efficient Data Sharing in Cloud Storage

Smita Athanere
IET, DEVI AHILYA VISHWAVIDYALAYA INDORE MP INDIA, smita.athanere@gmail.com

Dr. Ramesh Thakur
IIPS, DEVI AHILYA VISHWAVIDYALAYA INDORE MP INDIA

pp. 126-147

Recommended Citation
Athanere, Smita and Thakur, Dr. Ramesh. "A Hierarchical Multi-Authority Access Control Scheme for Secure and Efficient Data Sharing in Cloud Storage." *Journal of Strategic Security* 15, no. 1 (2022) : 126-147.
DOI: https://doi.org/10.5038/1944-0472.15.1.1970
Available at: https://digitalcommons.usf.edu/jss/vol15/iss1/6

A Hierarchical Multi-Authority Access Control Scheme for Secure and Efficient Data Sharing in Cloud Storage

Abstract

Enterprises choose to keep their data on the cloud to allow for flexible and efficient data exchange among their authorized staff when dealing with huge data. However, during the sharing of sensitive data, data security and users privacy has become major challenges. Most of the existing studies have several limitations, including weak model security, single point of failure, and lack of efficiency during user revocation. This article proposes cloud storage based Hierarchical Multi-authority Access Control Scheme (HMA-ACS) for secure and efficient data sharing. Through theoretical analysis, this article proves that the proposed mechanism efficiently performs cryptographic key operations and secured plus adaptive in the standard model while supporting the access policies. Furthermore, the proposed approach evaluated and compared recent state-of-art schemes in terms of storage overhead, computation overhead, average encryption, and decryption performance. Experimental results analysis shows that the proposed solution is resistant to many types of security threats and ensures data privacy when sharing data in the cloud.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive suggestions and comments, which have been incorporated into the article.

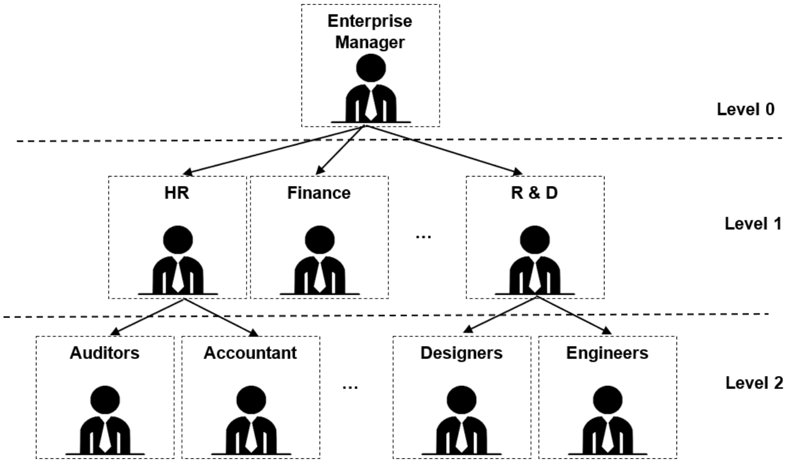
Introduction

Virtualization and containerization technologies make it possible for customers with resource constraints to share computational resources and services on demand.[1] In comparison with traditional data storage methods, the cloud computing storage services have increasingly become popular. Many cloud service providers, such as Google Drive, Amazon S3, Microsoft One Drive, and others, have developed in recent years to provide data storage and processing.[2] Users can access and easily recover data utilizing cloud technology at any time and from any location. There are multiple reasons due to which the organization prefers cloud storage services for storing their company data, such as:

1. Reduction in financial cost and overload due to renting of storage servers instead of buying;
2. Easy sharing of data to the remote users;
3. Internet of Things (IoT) devices generates a massive amount of data, which requires a back-end to store it.

Even though cloud offers numerous advantages, still there are a plenty of security challenges that require attention while sharing and storing data in cloud.[3] Because data owners' needs to access and share private data, cloud servers are no more reliable, this makes access control more difficult. The cloud data owners are responsible for determining and controlling access policy. For example, Figure 1 show that in a typical enterprise scenario, where the enterprise manager at level 0 can access all data of the whole organization. While the Research and Development (R&D) department managers at level 1 authorizes their groups and all their subordinate groups to access the data, as a contrast, the engineers of level departments can only access the data of their own groups.

Figure 1. An Example of Hierarchical Access Control



Source: Author.
Notes. This shows hierarchical organization of users bearing different responsibilities

There have been several recent publications that discuss attributes-based encryption schemes and access control records. To avoid congestion in the central point of the system and possible attacks, this article offers multi-authority access control systems. This framework distributes the computational tasks between different authorities to provide secure and efficient access control. This article mainly focused on three major security concerns particularly.

1. Grant data access to only authorized users using efficient key management mechanism;
2. Guarantee secure data sharing between the group members, whenever there is a request of upload or download;
3. Tasks decentralization between the pools of authorities to avoid the single point failure during data sharing.

Motivation

Existing attribute-based encryption schemes have presented different contributions for the security attacks. However, still the existing schemes are increasing the central authority role and do not capture the dynamic behavior of dynamic users and the level of network threat. As a result, our primary aim is to create a novel hierarchical multi- authority access control for cloud data sharing that is both secure and efficient.[4] The proposed design eliminates the role of the central authority while continuously monitoring the dynamic behavior of cloud users.

Contribution

This research article presents a novel hierarchical multi-authority access control mechanism for cloud-based storage data sharing that is much more secure and efficient than traditional access control mechanisms. This article presents a framework of several interoperable algorithms for meeting the basic authorization requirements of multi-authority cloud systems. Here are some of the most significant contributions:

1. To avoid the single point failure during data sharing, this article proposes that decentralization of the central authority's functions among the pool of authorities.
2. Each domain authority can oversee accepting or denying access requests for a certain group of users.
3. This research proposes a secure and robust hierarchical cryptography strategy for data sharing in cloud storage. Also, supports blacklisting a specific group member to ban that member for a predetermined amount of time based on current network threat level.
4. Through theoretical and experimental evaluations, this research demonstrates the robustness of proposed approach against different types of security attacks during sharing of data in the cloud environment.

Organization

The article organizes in following manner: Related work compares the proposed schemes with the other existing schemes. The proposed work shows hierarchical multi-authority access control framework and its various algorithms. The experimental setup, scheme implementation, and results analysis are discussed. Finally, in the conclusion and future works are there.

Related Work

Attribute based encryption provided fine-grained access control in a rapid and straightforward manner. The access policy is known as Ciphertext-Policy Attribute-Based Encryption (CP-ABE).[5] While using the ciphertext-policy attribute-based encryption to encrypt a message, only members of the group with access permissions and matching attributes can receive and decrypt the ciphertext. To detect the malware applications in mobile phone, the researchers proposed a technique named as Rough Dorid.[6] The authors proposed an idea of developing the attribute-based encryption approach in resource-constrained devices. Furthermore, Bhushan and Gupta presented a network flow analysis-based approach. In a multimedia cloud context, it identifies and mitigates fraud-related threats.[7] However, present access control techniques have several flaws, such as dealing with collusion attacks to resolve these issues; the researchers further propose some innovative solutions to prevent collusion-based attacks. Designs of several attribute authorities are available. For instance, for wireless body area networks, Shen et al., presented a cloud- integrated light weight certificate less authentication protocol with anonymity.[8] This method ensures that only the network manager has access to the user identity. Zkik et al., have devised a homomorphic encryption-based authentication and confidentiality strategy, as well as a recovery-based approach for enabling secure access for mobile users at the multi-cloud server remotely.[9] This research investigates various attribute authorities considering certain relevant work such as Lewko et al., Li et al., and Jouini et al., all of which mainly dealt with security issues in cloud environments.[10]

Following are three main classes of the existing protocols for group key distribution:

1. Centralized method: where single authority handles the entire group.
2. Decentralized method: where entire group divides into multiple sub-groups and controlled by their respective subgroup managers.
3. Distributed approach: where the group members are in charge for generating the key. The centralized methods are mostly based on the idea of Logical Key Hierarchy (LKH) protocol.

In this approach, a trusted server maintains a hierarchical tree structure. The decentralized approaches divide the group of members into multiple

https://digitalcommons.usf.edu/jss/vol15/iss1/6
DOI: https://doi.org/10.5038/1944-0472.15.1.1970

tiny groups; an intermediate key distribution server is in charge of each of them.[11] Mitrra et al., proposed an idea of Iolus framework, where the Group Security Agent (GSA) oversees the subgroup.[12]

The ABE technique uses attributes to link with data throughout the encryption process. In a key policy-based attribute-based encryption proposed by Goyal et al., private keys link data and attributes.[13] A mechanism for access control proposed by Nair et al., in this technique, file control and authentication uses public key cryptography for file control, and public key cryptography is for identification.[14] Niu et al., suggested an access control system for cloud environments that allows lightweight devices to securely access resources.[15] Qiu et al., introduced a new key-aggregate encryption-based hierarchical access control system that allows users to share data with any user group in cloud storage.[16] The size of each key in the proposed method is constant and unaffected by the hierarchical user structure's scale. The proposed technique makes key administration more convenient by eliminating the key derivation that is commonly used in existing hierarchical key assignment methods. He et al., presented a hierarchical ciphertext-policy attribute-based encryption algorithm whose access structure is based on linear secret sharing technique to accomplish fine-grained access control of many hierarchical files.[17] They also provided a hierarchical access control mechanism based on attributes (AHAC). When the qualities of a data visitor match a component of the access control structure decrypts the data associated with that part. Experiments reveal that attribute based hierarchical access control has a high level of security and performance. Furthermore, as the number of encrypted data files grows, AHAC's efficiency will become more prominent. Li et al., presented a novel multicast key distribution technique that enables multi-level controllers to oversee a specific group.[18] The suggested technique effectively balances controller activity, improves group key distribution reliability, and allows group members to create dynamic sessions without the usage of controllers.

Praveen et al., presented an efficient multi-authority access system using ABE scheme in cloud storage. The proposed system having an algorithm based on Role Hierarchy Algorithm (RHA) and Hierarchical Access Structure (HAS) to provide privacy and fine-grained access to stored data.[19] The role hierarchy algorithm classified the cloud users into different groups based on assigned attributes. The hierarchical access structure support to fine grained access through multi-authorities from cloud storage. Huang et al., proposed a secure and efficient data collaboration scheme using attribute-based Encryption and attribute-based signature (ABS). [20] This technique allows for fine-grained ciphertext access control and safe data writing operations. Here, our approach uses a full delegation mechanism based on Hierarchical Attribute-Based Encryption (HABE) to relieve the attribute authority of the burden of key management. They also developed a partial decryption and signing construction approach that offloads most of the computing overhead from the user to the cloud service provider. The proposed strategy is secure and efficient, according to the security and performance analyses. Wei et al., presented a cost effective and secure system based on attribute-based encryption scheme for multi-authority access from cloud storage.[21] In this article, they have constructed feature-based ciphertext-policy attribute-based encryption. Those features are:

1. System does not require the fully trusted central authority.
2. Each attribute can dynamically remove any user from its domain.
3. Cloud server can update the encrypted data by which users cannot access the previously available data.
4. Cipher text and secret keys updated publicly.

Proposed Work

System Model

The proposed hierarchical multi-authority access control scheme relies on securing cloud storage using multi- authority systems. The proposed scheme encrypts the secret key and distributes it among different domain authorities. The suggested encryption algorithm defines the set of privileged users. The permitted cloud users relate to a user's threshold level to ensure the system's consistency. Figure 2 depicts the overall system model. When it comes to secure data sharing, there are five parties involved: Cloud storage servers, data owners, group members, central authority, and domain authorities. Following subsection describes these entities:

- *Cloud storage servers:* This entity comprises a set of cloud

DOI: https://doi.org/10.5038/1944-0472.15.1.1970

storage servers which offer cloud services. These servers have a massive amount of storage capabilities to handle millions of access and storage requests. In order to guarantee security, data is stored in encrypted form on servers. Furthermore, these servers have no legal authority to violate the privacy of data owners.

- *Data owners (DOs):* The owners of the data define their own access policies for each and every file, which are recorded in an access control list (ACL). Before storing or uploading data to cloud storage systems, they encrypt it. To access the public key of authorities, data owners must first communicate with each domain authority in the multi-authority system.

- *Group Members (GMs):* The group members are the users of the organization, who can download the files for access. The group members register with cloud and send their access request using their <member id (GID), File id (FID)>.

- *Central authority (CA):* Central authority has the responsibilities for domain authorities, data owners, and group members. It created a unique identifier (OID) for data owner and a unique authority identifier (AID) for each domain authority. The central authority provides access grants to each requesting group member using a unique member identifier (UID). In addition, based on the network threat level, the central authority applies a blacklisting algorithm to block some of the group members based on their UID and blocked their access for varying predefined period. The length of time varies depending on the severity of the present network threat. Once the blacklisting period has passed, the group member can submit a request to the central authority for data access. The domain authorities then recalculate the requests.

- *Domain authorities (DAx):* It is a group of cloud storage servers with a lot of computing power that can handle a huge storage of data and related access requests. The servers for cloud storage have no authority over or access to the owner's data. The cloud is a non-privileged third-party system.

Proposed Scheme

This reduces overall responsibility of central authority. Also, in the proposed approach, each domain authority could play the role of central authority for a set of group members.

There are various steps in our recommended method that must be fulfilled, and they are as follows:

Setup: There are four sub-algorithms in this section:

- *CA-Setup:* This setup involves initialization of central authority and S_0 is shared between the domain authorities. Each service can be provided a weight attribute (w_a).

- *UID Setup:* This setup involves generation of unique user identifier for each group user (UID). Each group member must provide the UID to get the secret key.

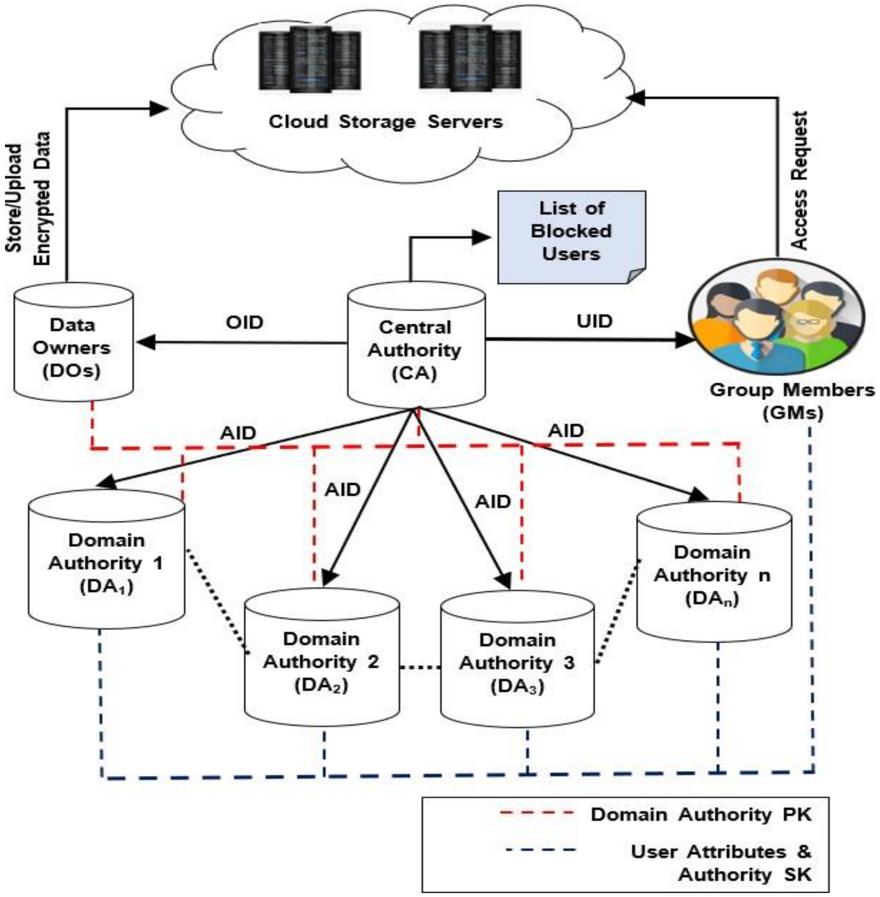
- *AID Setup:* The central authority assigned each group member a unique domain authority identity (AID). The AID's major job is to keep the security system safe from suspicious or unauthorized domain authorities that have not been verified by the central authority. In addition, each domain authority has a designated prime attribute. This prime property is solely responsible for generating the secret key for the domain authority. The following keys are created.

- *Attribute Secret Key:* A domain authority generates an attribute secret key for its own set of attributes. The members of the group use this secret key as a secret key.

- *Attribute Public Key:* Each domain authority also generates an attribute driven public key for the group members.

- *Key Generation:* Each domain authority will create a secret key along with a public key for each attribute. Domain authorities should be identified by a unique AID. The public key will be given to the data owners and will be used to encrypt their data. The public key, which will be used to encrypt the data, will be given to the data owners. Suggested method selects some public parameters, such as a random value (Z) and a huge prime integer (Q).

Figure 2.The Proposed Hierarchical Multi-Authority Cloud System



Source: Author.
Notes. This shows hierarchical organization of Central Authorities to different Domain Authorities to their Member Nodes and related flow of control

A secret value “s” is allocated to each group member, and a random value “m” is chosen by the members when they join the system. The authority server generates the group key by using the respective member id. The following equation 1 is used to denote the group key formula 'GK':

$$G_k = \prod_{i=1}^{n} \left\{ Q + \frac{Z^{s_i \oplus m_i} modQ}{s_i \oplus m_i} \right\} + s_k$$

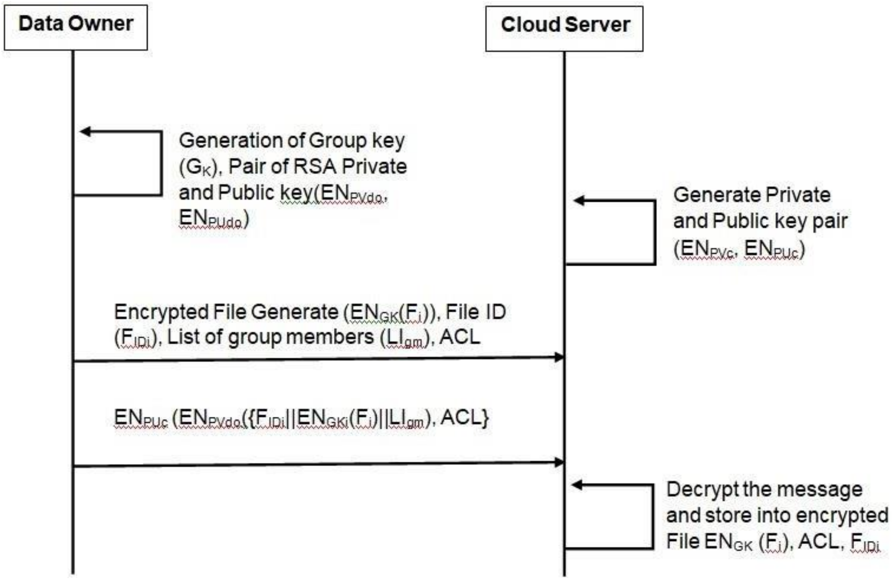
The secret values s_i and m_i are selected by group members such that $2< s_i< Q-2$ and $2< m_i <Q-2$. For all actions involving the key value, a random secret key is chosen.

- *Encryption:* The entire encryption algorithm converts plain text into ciphertext. The cipher text is the result of this phase (C_T is cipher text and P_T is plain text).
  Ciphertext $C_T= P_T \bmod n$
- *Decryption:* Only members of the authorized group are allowed to decipher cipher text. Decryption is a simple process that includes systematic analytics for calculation. (C_T is cipher text and P_T is plain text)
  Plaintext $P_T = C_T \bmod n$

.

Figure 3 shows the communication procedure between the cloud storage server and the data owner. Figure 4 depicts the message sharing between the group member and the cloud storage server. Figure 5 shows the message sharing between the data owner and the member of group. Here access control list represented as Access control list. The communication procedure between the Central Authority and member belong to group is depicted in Figure 6.

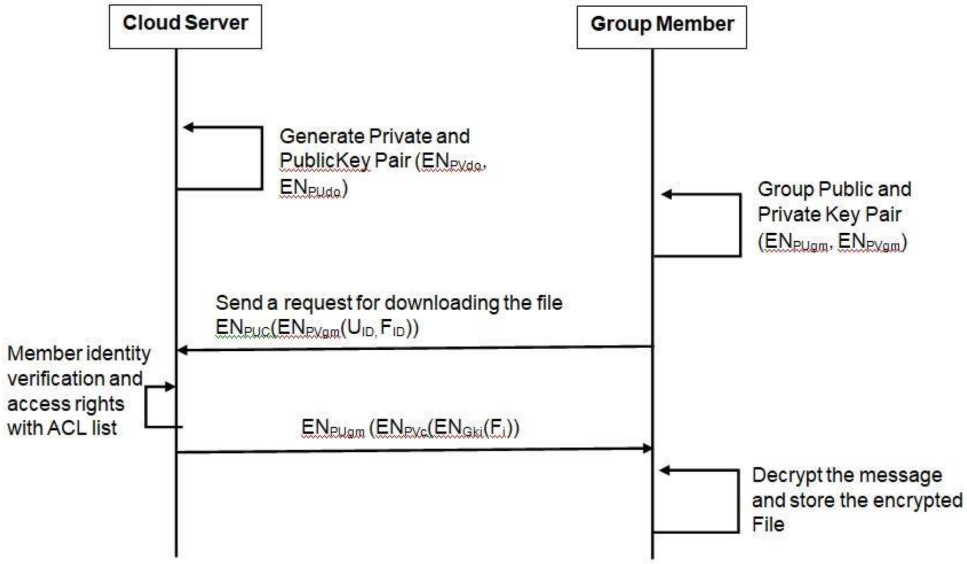
Figure 3. Message Sharing between the Data Owner and the Cloud Storage Server


Data Owner
Cloud Server
Generation of Group key (G_K), Pair of RSA Private and Public key(EN_{PVdo}, EN_{PUdo})
Generate Private and Public key pair (EN_{PVc}, EN_{PUc})
Encrypted File Generate ($EN_{GK}(F_i)$), File ID (F_{IDi}), List of group members (LI_{gm}), ACL
EN_{PUc} (EN_{PVdo}({F_{IDi}||$EN_{GKi}(F_i)$||LI_{gm}), ACL}
Decrypt the message and store into encrypted File EN_{GK} (F_i), ACL, F_{IDi}

Source: Author.

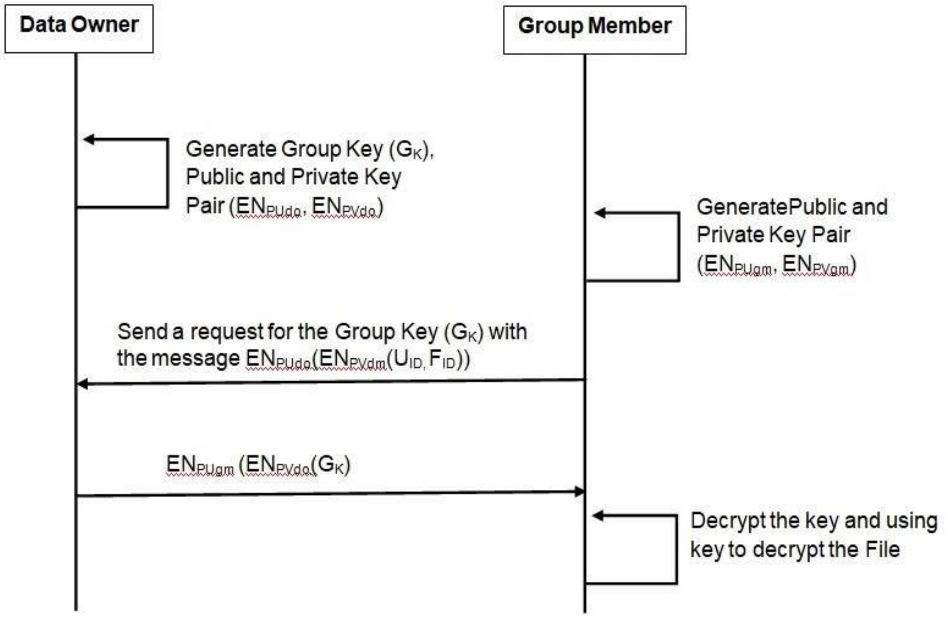
DOI: https://doi.org/10.5038/1944-0472.15.1.1970

Figure 4. Message Sharing between the Cloud Storage Servers and the Group Member


Cloud Server
Group Member
Generate Private and PublicKey Pair (EN_{PVdo}, EN_{PUdo})
Group Public and Private Key Pair (EN_{PUgm}, EN_{PVgm})
Send a request for downloading the file $EN_{PUC}(EN_{PVgm}(U_{ID}, F_{ID}))$
Member identity verification and access rights with ACL list
EN_{PUgm} ($EN_{PVc}(EN_{Gki}(F_i))$
Decrypt the message and store the encrypted File

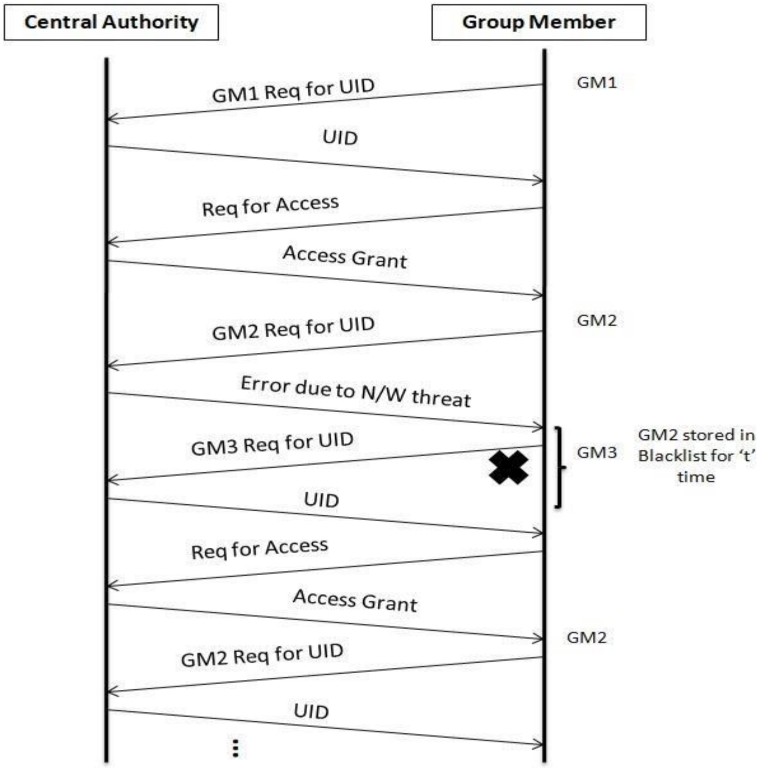
Source: Author.

Figure 5. Message Sharing between the Data Owner and the Group Member


Data Owner
Group Member
Generate Group Key (G_K), Public and Private Key Pair (EN_{PUdo}, EN_{PVdo})
GeneratePublic and Private Key Pair (EN_{PUgm}, EN_{PVgm})
Send a request for the Group Key (G_K) with the message $EN_{PUdo}(EN_{PVdm}(U_{ID}, F_{ID}))$
EN_{PUgm} ($EN_{PVdo}(G_K)$
Decrypt the key and using key to decrypt the File

Source: Author.

Figure 6. Interaction between the Central Authority and the Member of Group



Source: Author.

Experimental Results and Analysis

Performance evaluation of this method takes into account both the total number of members as well as the number of members who join or leave the group. Let us assume the size of total group members is denotes as 'l' Then, this computes the total count of members of any group as 'l'= 2^{ht}, where 'ht' denotes the height of member tree with $\log_2 l$. It further analyzes the performance in terms of encryption & decryption, key generation, and total storage overhead. Table 1 shows the process of encryption and decryption, key generation overhead, and the storage complexity at the central authority server. In case of key generation overhead for group members at the time of joining and leaving phase no message passing required in classical and logical key hierarchy method but in proposed method one message required for leaving phase because of multi-authority control. Similarly, due of authentication restrictions, authority server message overhead is necessary during the joining phase but not during the leaving phase. In encryption and decryption overhead, logical key hierarchy protocol required more messages as compare to proposed method because logical key hierarchy uses tree structure of group members but proposed method uses tree structure with the concept of domain authority server. The communication overhead between the group members and domain authorities is depends on the product of attributes set from all domain authorities represented by 'UAT_{UID}' and total bit length of elements denotes as 'δ' In Table 1 shows the communication overhead analysis for the proposed scheme, while the other schemes do not support the concept of domain authorities

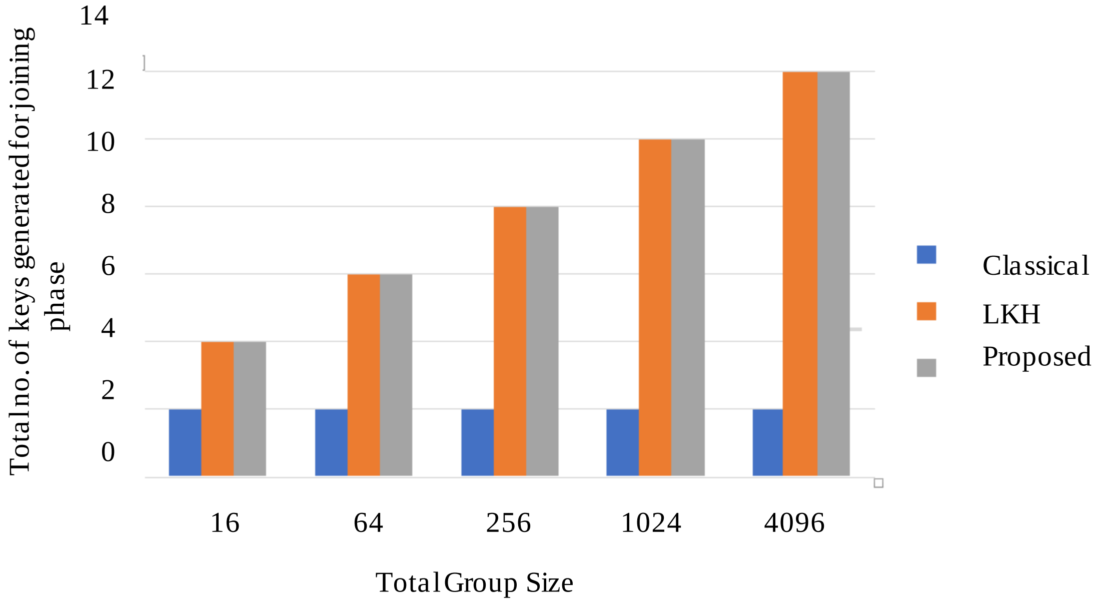
Table 1.Analysis of Key Generation for Joining and Leaving Phase in terms of Time and Space Complexity

Scheme	**Key Generation Overhead**				**Communication Overhead**	**Encryption/Decryption Overhead**				**Total Storage Complexity**	
	Member Node of Group		**Authority Server**		**Domain Authorities to Group Members**	**Authority Server**	**Member Node of Group**	**Authority Server**	**Member Node of Group**	**Authority Server**	**Member Node of Group**
	Joining Phase	**Leaving Phase**	**Joining Phase**	**Leaving Phase**		**Joining Phase**	**Leaving Phase**	**Joining Phase**	**Leaving Phase**		
Classical	0	0	2	1	Not Supported	2	l-1	1	1	l	2
LKH	0	0	log2l	log2l-1	Not Supported	3. log2l	2. log2l	log2l	log2l	2l	log2l+1
Proposed	0	1	log2l	0	|UATUID| δ	log2l+1	log2l-1	1	0	2l	log2l+1

Source: Author.

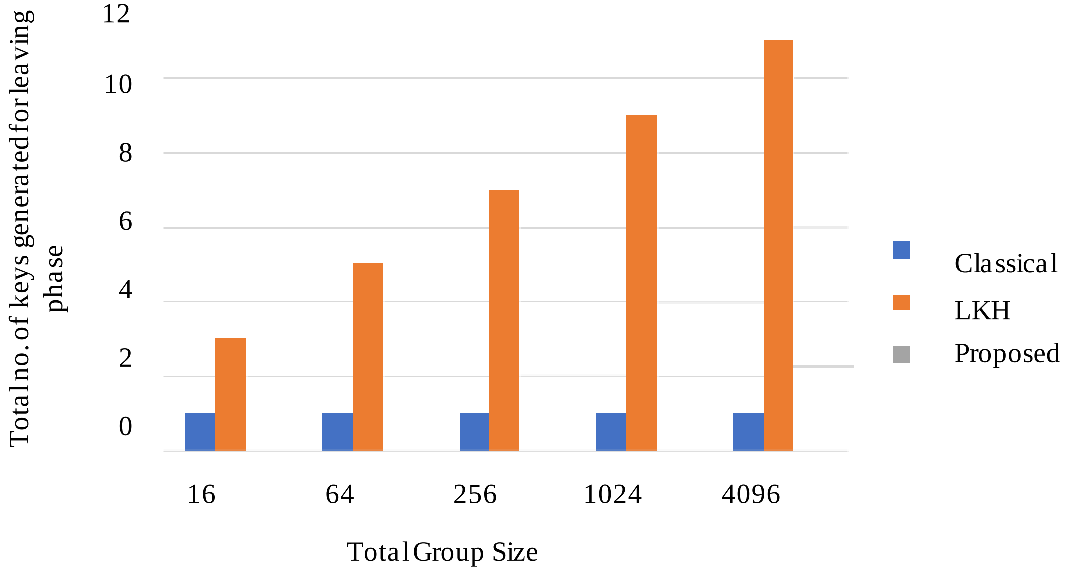
Notes. This table is comparatives summary of classical method, Logical key Hierarchy approach and proposed scheme on Quality-of-Service parameters like Key Generation Overhead, Communication overhead, Encryption Decryption Overhead and Storage Overhead

Figure 7. Server-side Key Generation during the Operation of new Joining



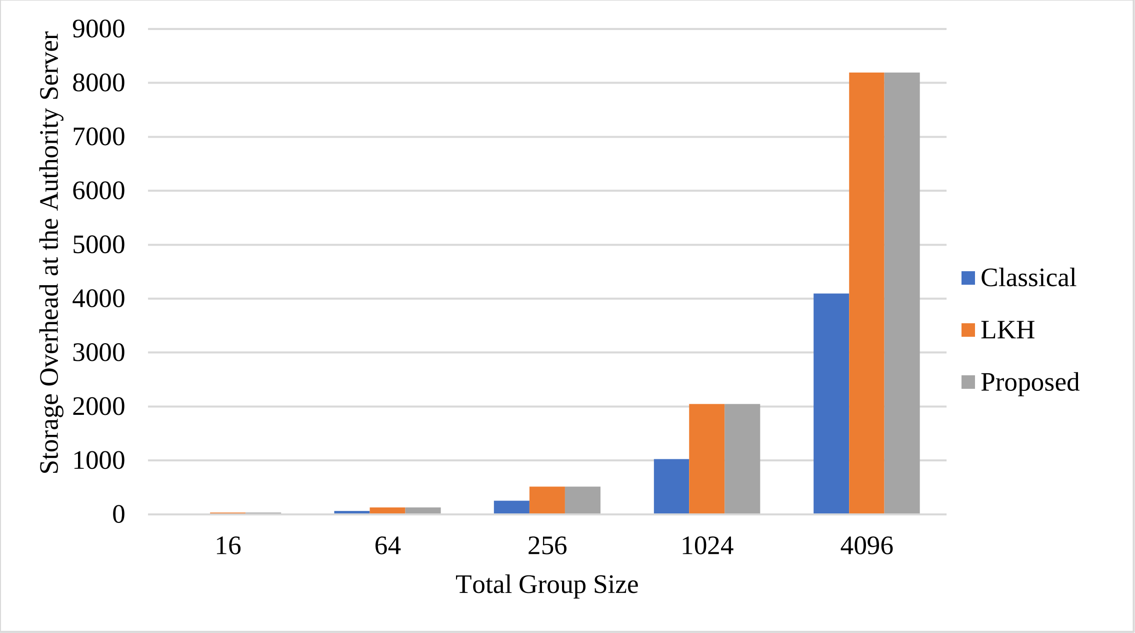
Source: Author.

Figure 8. Server-side Key Generation during the Operation of Leaving



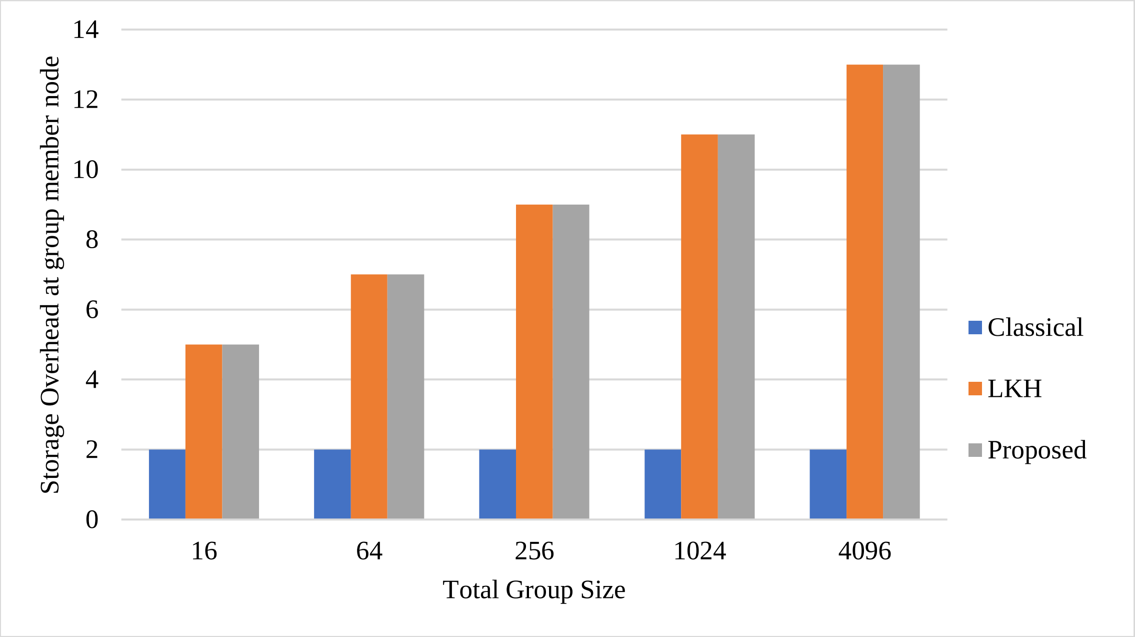
Source: Author.

Figure 9. Authority Server-side Storage Overhead during Joining Phase



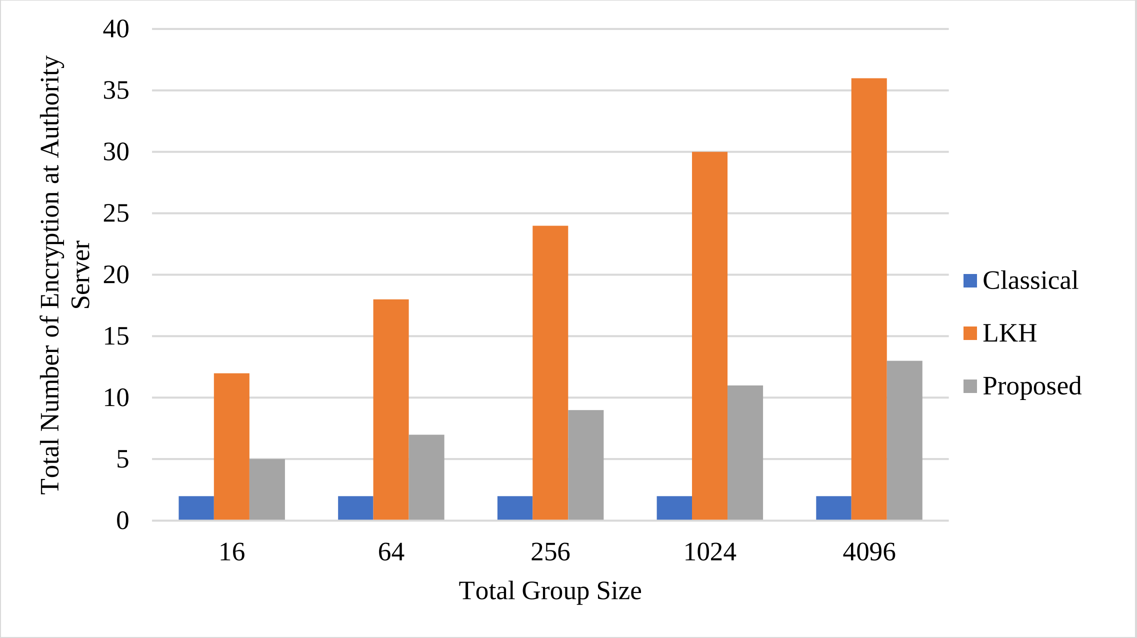
Source: Author.

Figure 10. Group Member Node-side Storage Overhead



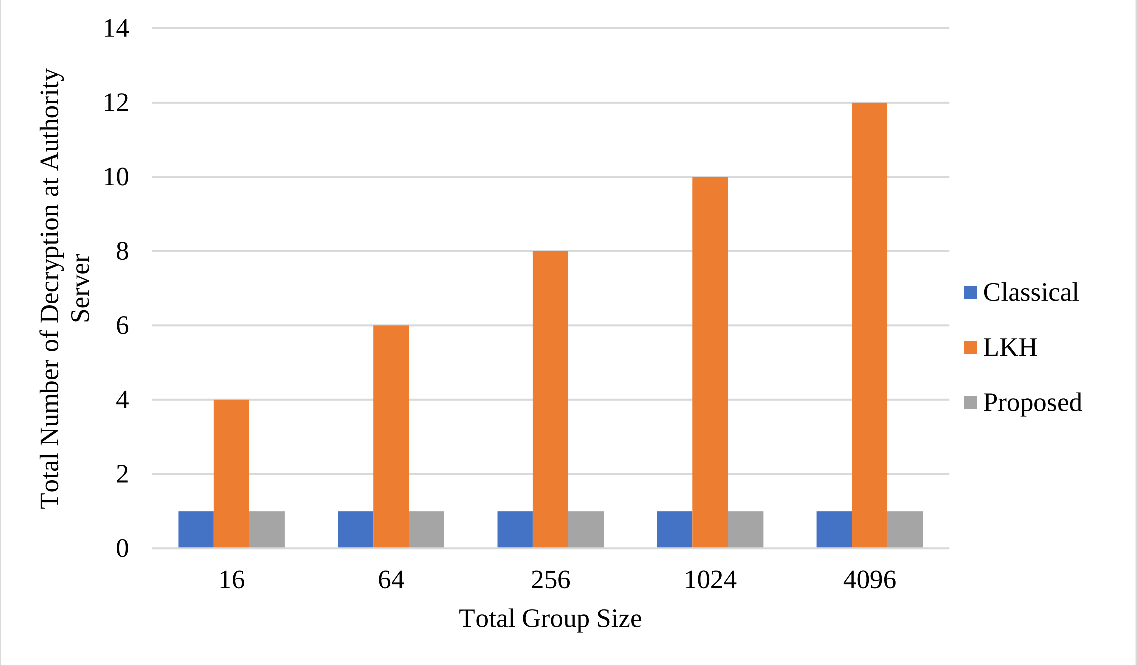
Source: Author.

Figure 11. Authority Server-side Encryption analysis during Joining Phase



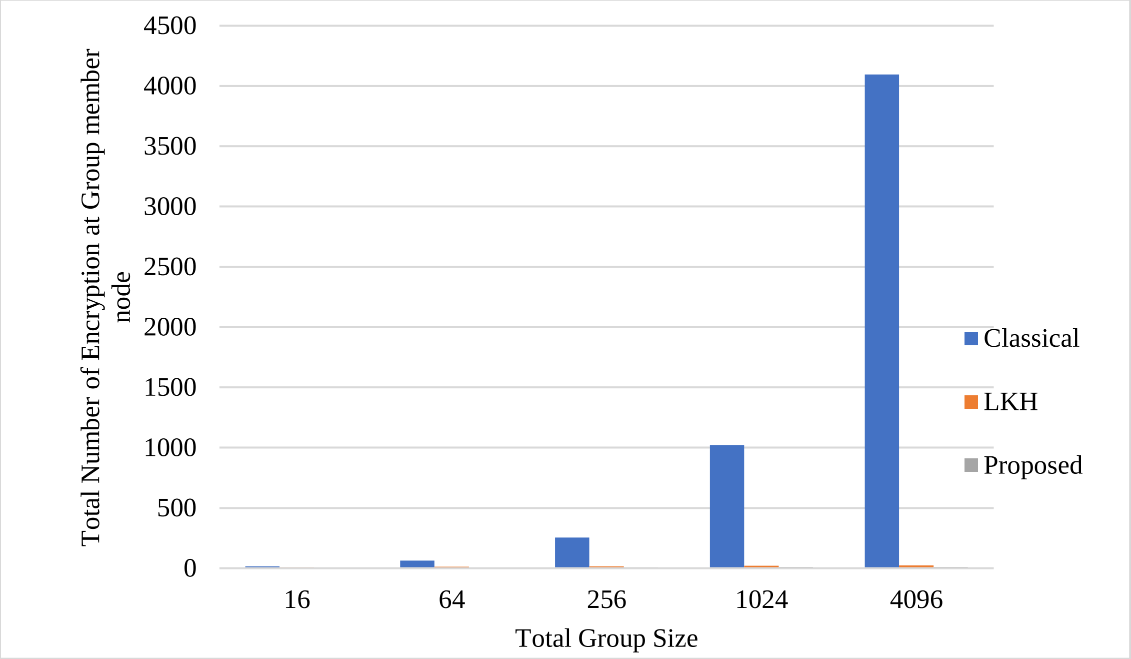
Source: Author.

Figure 12. Authority Server-side Decryption Analysis during Leaving Phase



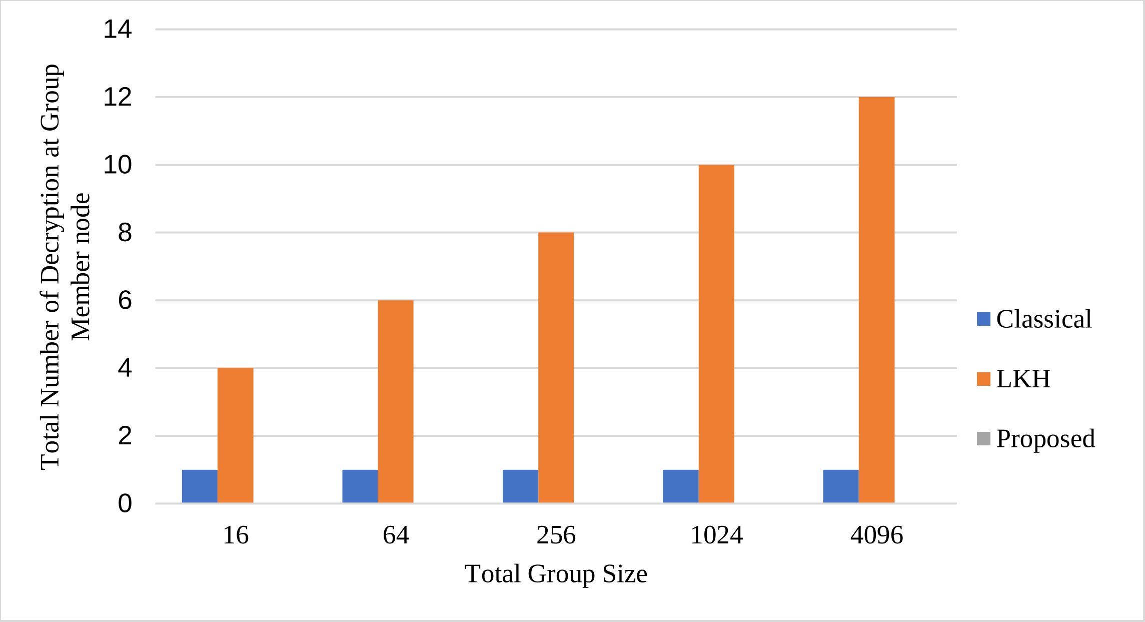
Source: Author.

Figure 13. Group Member Node-side Encryption Analysis during Joining Phase



Source: Author.

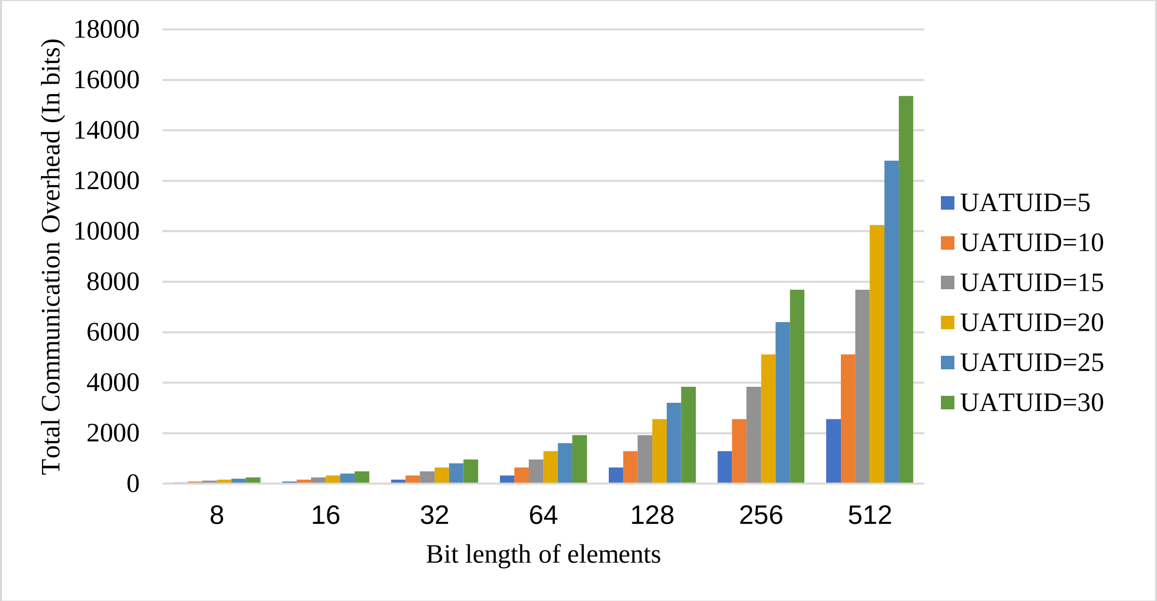
Figure 14. Group Member Node-side Decryption Analysis during Leaving Phase



Source: Author.

DOI: https://doi.org/10.5038/1944-0472.15.1.1970

Figure 15. Communication Overhead Analysis between Group Members to Domain Authority



Source: Author.

Conclusion & Future Work

This article presents a new secure and efficient hierarchical multi-authority access control scheme for data sharing in cloud storage. The intention of suggested framework is to address a variety of challenges that arise during data sharing in cloud storage. The proposed scheme achieved following key objectives.

1. To prevent becoming a central point of failure, tasks decentralization of central authority between the pools of authorities has done.
2. The concept of blacklisting a specific group member has-been proposed to block the member for a predetermined length of time based on current network threat level.
3. Testing findings show that the proposed technique efficiently evaluates cloud users access requests.

The future work of proposed approach is based on linking the permission of each group member to a threshold range. This threshold range is based on the current threshold level of the group member. Assigning for a specific period and re-evaluate on the time-period expiration. In future, it is possible to achieve a dynamic threshold-based vector to revoke the ticker for users at various stages of permission, depending on the amount of network hazard.

Endnotes

[1] Mustaque Ahamad, "Multicast Communication in Distributed Systems," (*IEEE Computer Society*, 1990).

[2] Heba K. Aslan, "A scalable and distributed multicast security protocol using a subgroup key hierarchy," *Computers & Security* 23 (2004): 320-329, doi:10.1016/j.cose.2003.11.003; Yacine Challal and Hamida Seba, "Group Key Management Protocols: A Novel Taxonomy," *International Journal of Computer and Information Engineering* 2 (2008): 3620 – 3633, doi.org/10.5281/zenodo.1077968; Lawrence Harte, "Introduction to Data Multicasting, IP Multicast Streaming for Audio and Video Media Distribution," (*Fuquay-Varina: Althos Publishing 2008*).

[3] Nicolas Bonmariage and Guy Leduc. "A survey of optimal network congestion control for unicast and multicast transmission," *Computer Networks* 50 (2006): 448-468,doi:10.1016/j.comnet.2005.04.015; Kin Ching Chan and S.H. Gary Chan, "Key management approaches to offer data confidentiality for secure multicast," *IEEE Network* (October 2003): 1730-9,https://doi.org/10.1109/MNET.2003.1233915

[4] Jiannong Cao, Lin Liao and Guojun Wang, "Scalable key management for secure multicast communication in the mobile environment," *Pervasive and Mobile Computing* 2 (2006): 187-203. doi:10.1016/j.pmcj.2005.11.003.

[5] Amit Sahai, Brent Waters. "Fuzzy Identity-Based Encryption," Advances *in Cryptology – EUROCRYPT Lecture Notes in Computer Science*, ed. R Cramer (Heidelberg: Springer Berlin 2005); John Bethencourt, Amit Sahai and Brent Waters, "Ciphertext-Policy Attribute-Based Encryption," in *IEEE Symposium on Security and Privacy* (2007): 321-334, doi: 10.1109/SP.2007.11; Amit Sahai, Brent Waters "Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization," in *Lecture Notes in Computer Science, Public Key Cryptography* – PKC 2011 ed. Catalano Dario Fazio Nelly, Gennaro Rosario, Nicolosi Antonio (Heidelberg :Springer, Berlin 2011), https://doi.org/10.1007/978-3-642-19379-8_4.

[6] Riad Khaled, and Lishan Ke. "RoughDroid: Operative Scheme for Functional Android Malware Detection," *Security and Communication Networks* (September 20, 2018): 1–10, doi:10.1155/2018/8087303.

[7] Kriti Bhushan and Brij Gupta, "Network flow analysis for detection and mitigation of Fraudulent Resource Consumption (FRC) attacks in multimedia cloud computing," *Multimedia Tools and Applications* 78(February 2019): 4267–4298, https://doi.org/10.1007/s11042-017-5522-z.

[8] Shen Jian, Gui Ziyuan, Ji Sai, Shen Jun, Tan Haowen, and Tang Yi, "Cloud-aided lightweight certificate less authentication protocol with anonymity for wireless body area networks," *Journal of Network and Computer Applications* 106 (March 2018): 117-123, https://doi.org/10.1016/j.jnca.2018.01.003.

[9] Karim Zkik, Ghizlane Orhanou and Said El Hajji, "Secure mobile multi cloud architecture for authentication and data storage," *International Journal of Cloud Applications and Computing (IJCAC)*, 7 (2017): 62–76, https://doi.org/10.4018/IJCAC.2017040105.

[10] Allison Lewko, Brent Waters, "Decentralizing Attribute-Based Encryption," Lecture

Notes in Computer Science ed. Paterson K.G. (Heidelberg: Springer Berlin 2011),https://doi.org/10.1007/978-3-642-20465-4_31; Li Jin, Xiaofeng Chen, Sherman S. M. Chow, Qiong Huang, Duncan S. Wong and Zheli Liu, "Multi-authority fine-grained access control with accountability and its application in cloud," *Journal of Network and Computer Applications* 112 (June 2018): 89-96, https://doi.org/10.1016/j.jnca.2018.03.006; Mouna Jouini and Latifa Ben Arfa Rabai, "A Security Framework for Secure Cloud Computing Environments," International *Journal of Cloud Applications and Computing* 6 (July 2016): 32-44, doi:10.4018/IJCAC.2016070103

[11] Zhang Jun, Zhou Yu, Ma Fanyuan, Gu Dawu and Bai Yingcai, "An extension of secure group communication using key graph", *Information Sciences*, 176(2006): 3060-3078, https://doi.org/10.1016/j.ins.2005.12.008.

[12] Suvo Mittra, "Iolus: a framework for scalable secure multicasting," *ACM SIGCOMM Computer Communication Review,* 27(1997) 277–288, https://doi.org/10.1145/263105.263179.

[13] Vipul Goyal, Omkant Pandey, Amit Sahai A, Brent Waters, "Attribute-based encryption for fine-grained access control of encrypted data," *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security* (October 2006) 89-98, https://doi.org/10.1145/1180405.1180418.

[14] Srijith K. Nair , Mohammad T. Dashti , Bruno Crispo , Andrew S. Tanenbaum," A hybrid PKI-IBC based ephemerizer system," *in New Approaches for Security, Privacy and Trust in Complex Environments: Proceedings of the IFIP TC-11 22nd International Information Security Conference*,(Boston, MA: Springer 2007):241-252. https://doi.org/10.1007/978-0-387-72367-9-21.

[15] Shaozhang Niu, Shanshan Tu and Yongfeng Huang, "An Effective and Secure Access Control System Scheme in the Cloud," *Chinese Journal of Electronics* 24 (July 2015): 524-528 https://doi.org/10.1049/cje.2015.07.015.

[16] Zhenyao Qiu, Zhiwei Zhang, Shichong Tan, Jianfeng Wang and Xiaoling Tao, "Hierarchical Access Control with Scalable Data Sharing in Cloud Storage*", Journal of Internet Technology* 20 (May 2019): 663-676 DOI: 10.3966/160792642019052003002; Rajasekaran Velumadhava, Selvamani Kadirvelu, Kanimozhi Sakthivel and Arputharaj Kannan, "Hierarchical group key management for secure data sharing in a cloud-based environment," *Concurrency and Computation: Practice and Experience* 31 (October 2018): e4866-e4881, https://doi.org/10.1002/cpe.4866.

[17] Heng He, Liang-han Zheng, Peng Li, Li Deng, Li Huang and Xiang Chen, "An efficient attribute-based hierarchical data access control scheme in cloud computing" *Human-centric Computing and Information Sciences*, 10 (December 2020): 1-19 https://doi.org/10.1186/s13673-020-00255-5

[18] Li Jie, Shaowen Yao, Jing Liu, and Yunyun Wu, "A Hierarchical Multicast Key Distribution Protocol," *Electronics* 10(2021): 995-1009 https://doi.org/10.3390/electronics10090995; Yejun Wu, Fansong Meng,"Categorizing Security for Security Management and Information Resource Management", *Journal of strategic security* 11(February 2019): 72-84 doi:10.5038/1944-0472.11.4.1694; Yejun Wu, "Developing a Taxonomic Framework of Security Methods for Security Management and Information Resource Management," *Journal of strategic security* 13(2020): 64-77, https://doi.org/10.5038/1944-0472.13.2.1787; Lanfranco Lopriore, "Key management in tree shaped hierarchies," *Information Security Journal: a Global Perspective*, 27 (2018): 205-213 https://doi.org/10.1080/19393555.2018.1516835.

[19] Praveen Challagidad, Mahantesh Birje, "Efficient multi-authority access control using attribute-based encryption in cloud storage," *Procedia Computer Science*, 167 (2020):840-849, https://doi.org/10.1016/j.procs.2020.03.423

[20] Qinlong Huang, Yixian Yang, Mansuo Shen, "Secure and efficient data collaboration with hierarchical attribute-based encryption in cloud computing," *Future Generation Computer Systems*, 72 (2017): 239-249, https://doi.org/10.1016/j.future.2016.09.021.

[21] Jianghong Wei, Wenfen Liu, Xuexian Hu, "Secure and efficient attribute-based access control for multiauthority cloud storage," *IEEE Systems Journal*, 12 (2016): 1731-1742, doi:10.1109/JSYST.2016.2633559.