International Journal on Applications in Engineering and Technology
Volume 10 : Issue 1 : March 2024 , pp 6 –12 www.aetsjournal.com ISSN (Online) : 2455 - 0523

BINARY TREE BASED KEY GENERATION IN CLOUD BASED SECURE DATA SHARING MECHANISM FOR MOBILE USERS GROUP

Mrs.P.PADMA SREE
Assistant Professor, Department of Computer Science and Engineering, PET Engineering College, Vallioor, India.
sreepadma78@gmail.com

Mrs.T.ANUJA
Department of Information Technology, Jeppiaar Engineering College, Semmencherry, Chennai, TamilNadu, India.
Email id: anuanuja2013@gmail.com

M.ELZA MELIF
Assistant professor , PET Engineering college , Vallioor, India.
Elzofficial92@gmail.com

***Abstract*—** In the cloud computing paradigm, allows the cloud to provide a scalable infrastructure for the storage of applications, data, and files. The amount of data that can be stored in mobile devices has significantly increased since the introduction of more sophisticated mobile devices. As a direct consequence of this, there has been a notable surge in recent years in the demand for cloud storage for the outsourcing of the private data of mobile users. It is important to protect users' privacy and safety by encrypting the data and preventing unauthorised users from accessing it. Because cloud service providers are generally thought of as a partially trusted third party, it is of the utmost importance to design a framework that improves the data protection when it is being shared in a group by mobile users. Using the proposed Binary Tree based Key Generation (BTKG), the group key pairs are updated with the assistance of cloud servers by using the Binary Tree based Key Generation (BTKG). However, no sensitive data is revealed to the cloud servers during this process. In addition, the confidentiality of the file that was downloaded is protected by a digital envelope, which is created with the keys that are generated by BTKG. The proposed system is both highly effective and complies with the security requirements for cloud-based public user groups that can be accessed by mobile devices.

Index Terms— BTKG, digital envelope, cloud storage, and group sharing

I. INTRODUCTION

Smart mobile devices have emerged as the primary computing platform in recent years. As a result of mobile users storing images, files, and generally using applications like cameras, the battery life of mobile devices has been shown to decrease in studies that have been conducted by a variety of different sources. The operation of many applications on a mobile device requires a significant amount of computational resources, and these applications consume a lot of power. In addition, there are a plethora of applications that enable users of mobile devices to collaborate on the sharing of data. Therefore, in order to carry out computations, a service-based environment is required if an end user on a mobile device wishes to have access to those applications.

People are now able to share and store large amounts of data thanks to the significant increase in storage capacity, computing power, and networking that has occurred over the course of the last few decades. As a consequence of the creation of new software applications, end users will eventually be in a position to make demands regarding the capabilities of the underlying computing infrastructure. In order to satisfy the requirements that mobile users place on computing infrastructure, system designers will need to select new algorithms, methods, and architectures for the processing of enormous amounts of data in a short amount of time. In today's world, a lot of academic and business organisations have built large systems by making use of computers, networks, and discs that are relatively inexpensive. Because of this, the hardware is now much easier to manage and to programme.

Many different organisations have proposed a diverse range of original solutions in an effort to address the issues that were discussed earlier. These companies sometimes use their hardware and software to provide pay-as-you-go storage, computation, and data management services to both their internal users and external customers. These services can be accessed by using a pay per use model. The hardware and software configuration that brings about this kind of service-based environment is what's known as a "Cloud-computing environment," and the term refers to that configuration.

The use of cloud computing makes the deployment of large-scale distributed systems much easier. Cloud computing also makes it feasible to outsource data backups to third-party cloud storage services, which in turn helps to reduce the costs associated with data management. However, it is absolutely necessary to ensure the data's security as well as its privacy while it is being stored in the cloud.

In addition to this, the sharing of data among groups has become an increasingly important practise in many different organisations. It is possible to construct a risk-free method of information exchange for the community of mobile users by making use of the appropriate key management techniques. The purpose of this paper is to propose a Binary Tree based Key Generation in order to make sharing information more securely (BTKG). After the file has been uploaded by a member of the group, the leader of the group will use BTKG to generate group key pairs and then distribute them to each and every participant in the group.

The file is encrypted using symmetric encryption with a session key that is chosen at random, and then the session key itself is encrypted using public-key encryption with the group's public key. Members of the group who have access to the group private key are the only ones who are able to decrypt or download the file. Because cloud storage only stores encrypted files and group key pairs, the cloud service provider is unable to mine the users' files. The fact that cloud servers are considered to be only partially trusted third parties is the primary driver behind the use of this mechanism. Because BTKG was used, the user is able to effectively store data and share it while maintaining a high level of security.

II. WORK THAT IS CONNECTED

Using symmetric encryption, data is encrypted using a session key that is generated at random, and then using public-key encryption, the session key is encrypted using the user's public key. Finally, the data is encrypted using asymmetric encryption.

The name for this strategy is "Digital Envelope," and it can be found in [1,2]. There are a lot of works that use cryptographic techniques, such as attribute-based encryption (ABE), to demonstrate the difficulty of sharing data in cloud-based systems while still protecting users' privacy. [7] Some examples of these kinds of works include: In [3,] Yu et al. present a solution that is both comprehensive and extensible. The efficiency of this system is directly proportional to the large amount of attribute variability that exists between the different users and files. The success of the strategies described in [4] and [5] is dependent on the assumption that cloud servers can be relied upon as honest third parties. In that case, cloud servers could initiate a collusion attack with members of the group who are naturally inquisitive and inclined to split off on their own. A data sharing scheme for mobile devices that is based on ABE and proxy re-encryption has been proposed in [6], but it also has the issue that is mentioned in [4] and [5] [4] and [5]. [6] There is a possibility that maintaining both forward and backward secrecy is necessary for security.

The first one ensures that the user whose access has been revoked is unable to decrypt any new cypher texts that are created. [3, 4], and [6] are the references that cover these mandatory security precautions. The latter one ensures that newly joined users are able to access and decrypt previously published data as soon as they become active users.

Before storing data, the cloud will authenticate the sequence even though it is unaware of the identities of the users.

[8] Also includes user revocation and access control, both of which work to ensure that only authorised users have the ability to decrypt data that has been stored in the cloud. Access policies can be expressed in any of the following formats, including Boolean functions of attributes, LSSS matrices, and monotone span programmes. It is possible to construct a Boolean function using any access structure. An individual is able to create a file and store it in the cloud in a secure manner. The scheme described in [8] makes use of the protocol ABE. Users can be broken down into three categories: writers, readers, and creators. Creator Alice is presented with a token by the trusted party, also known as the Trustee, which may be the United States government. When she demonstrates her identification by presenting a number such as her social security or health insurance card, the trustee gives her a token.

The scheme that is proposed in [9] supports a number of features that are not supported by the other schemes. One of these features is called 1-W-M-R, and it restricts the number of users who are able to write while allowing multiple users to read. According to the additional plan M-W-M-R that has been proposed, read and write operations can be managed by more than one user at a time. The vast majority of schemes do not support many different types of writes. In contrast to the majority of other authentication methods, this one is not only robust but also decentralised. It also protects users' privacy while providing authentication.

The primary advantage is that it prevents replay attacks and simplifies the reading, editing, and creation of data that will be saved in the cloud. Additionally, this makes it possible to store more data. The cloud does not make any attempt to determine the identity of the person storing information; it only verifies the credentials of the user. The distribution of keys is accomplished through decentralisation. The fact that the cloud is aware of the access policy for every record that is stored there is the most significant disadvantage.

Role-based access control, also known as RBAC, allows for flexible management and control of data stored in the cloud as a result of two mappings: one between users and roles, and another between roles and privileges on data objects. Combining RBAC with various cryptographic procedures is what the Role Based Encryption (RBE) scheme in [9] does.

The RBE scheme makes it possible for RBAC policies to be applied to encrypted data that is stored in public clouds.

[9] presents a new RBS scheme with efficient user revocation that is achieved by combining RBAC policies and encryption to improve secure large-scale data storage in public clouds. The main contribution of this paper is a secure RBAC-based hybrid cloud storage architecture that enables an organisation to store data securely in a public cloud while maintaining the sensitive information related to the organization's structure in a private cloud. A secure RBE-based cloud storage architecture is provided in the plan that is suggested in [9]. This architecture also includes hybrid cloud features, which enable an organisation to store data securely in a public cloud. The most important advantage is that it ensures efficient account cancellation for users. Encryption and decryption methods that are performed on the client side are effective, and the utilisation of multiple processors helps to speed up the decryption process. Controlling who can access the data stored in the cloud is the most effective way to guarantee and improve data security issues.

The inclusion of data outsourcing and only partially trusted cloud servers makes access control of the data a significant issue in cloud storage systems. These systems also present a challenge for maintaining data confidentiality. [10] This demonstrates that traditional methods of access control are not suitable for cloud storage systems because cloud servers may store multiple encrypted copies of the same data or are only considered to be partially trustworthy.

It's called cypher text-policy attribute-based encryption, and it's a promising method for restricting access to data that's been encrypted (CP-ABE). Due to the inefficiency of decryption and revocation, the existing CP-ABE scheme cannot be used directly to build a scheme for data access control in multi-authority cloud storage systems. These systems allow users to hold attributes from multiple authorities, which makes it impossible to use the scheme directly to build a scheme for data access control. data access control for multi-authority cloud storage (DAC-MACS), which is an efficient decryption and revocation scheme for a secure and effective data access control scheme, is suggested by [10]. DAC-MACS is an acronym for data access control for multi-authority cloud storage.

It not only designs a new multi-authority CP-ABE scheme that has efficient decryption, but it also designs an efficient attribute revocation method that can achieve both forward security and backward security. In the CP-ABE scheme that is being proposed, the majority of the decryption computation is performed by the server. In addition, this demonstrates that the fundamental security assumption that non-revoked users will not share their received key update keys with revoked users is false.

III. AN OVERVIEW OF THE ARCHITECTURE OF CLOUD STORAGE

The file that is going to be shared is uploaded by the mobile user of the group, who also generates an Advanced Encryption Standard key by using a value that is chosen at random (AES). Calculating the group public key is the next step for the user after encrypting the secret key with the Rivest Shamir Adelman algorithm using the group initialization procedure (RSA).

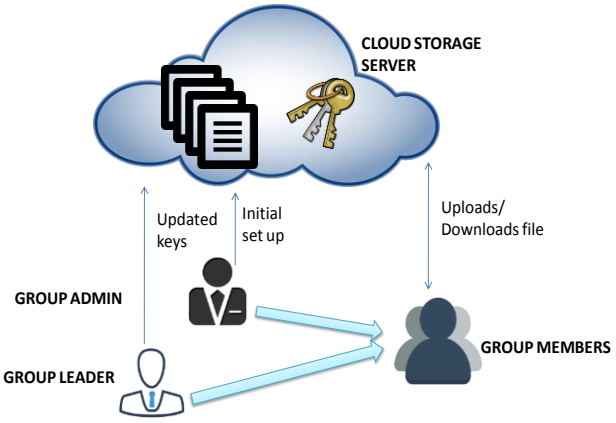

Figure 1: Cloud Storage Architecture

Cloud storage is used to keep group key pairs and encrypted versions of files, which allows for the problem of forward secrecy to be solved. The files can only be downloaded by the mobile user who has been granted permission by the group to do so; this is the user who is in possession of the group's private key. It is less necessary to rely on unreliable cloud providers because the keys are automatically updated whenever a user joins or leaves the group.

A. Individuals who take part in the system

Users of mobile devices take on one of the following roles: Administrator of the Group (GA), Members of the Group, and Leader of the Group (GL) (GM)

To quote the Group Leader (GL): There is only one person who serves as the leader of the group; this person was also a founding member of the organisation and is currently its chief administrator. They acquire computational and storage resources by either purchasing them from a cloud provider or receiving them themselves. GL has the ability to grant specific group members permission to manage the group; however, GL also has the ability to revoke this permission.

GL provides the initial group security parameters to each individual member of the group.

Administrator of the Group (GA): A group could have zero, one, or more authorised group administrators at its helm at any given time. They are responsible for serving as sponsors for the implementation of group key updating and possess the ability to manage group membership. The person in charge of the group always has the ability to take away their management responsibilities at any time.

Members of the Group (GM): Within the authenticated group, each GM has the ability to perform actions such as downloading and uploading files. Using specific pertinent public information obtained from cloud servers, each GM has the ability to generate their own one-of-a-kind set of security parameters. This can include a group key pair.

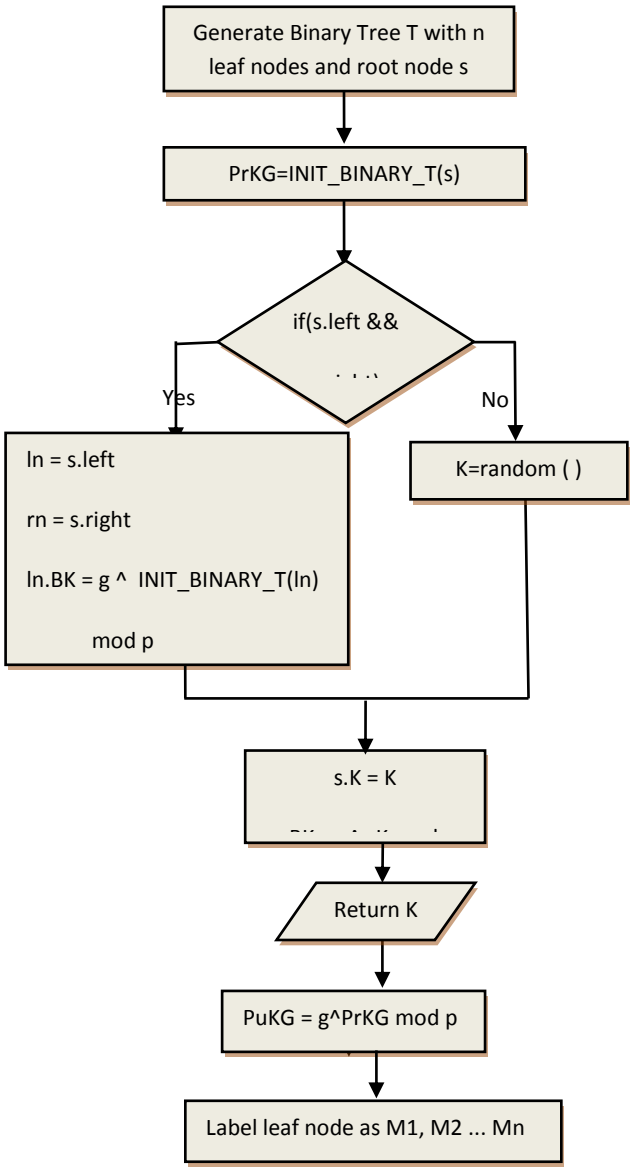

Figure 2: Group Initialization

B. Starting the group off for the first time

The group leader (GL) is the one who interacts with the cloud provider to acquire storage and computational resources. During the step of group initialization, which involves initialising a binary tree and certain security information associated with the group, the GL is also responsible for carrying out the initialization step.

GL has the ability to unicast the private key of each leaf node to the associated group member while maintaining the appearance of encryption and signing.

Using the storage provided by the cloud server, each member can independently compute the group private key.

In addition, we divide the phase of group members joining and leaving the group into three additional sub-phases, which are as follows: group member joining, group member leaving, and group administrator leaving. Figure 2 depicts the method of initialising groups in its entirety.

C. Electronic Envelope

Figure 3 depicts the technique. The data in the file is encrypted with symmetric encryption using a session key that was chosen at random ("S," for short), and then the session key itself is encrypted with public-key encryption ("EPuKG") using the group's public key ("S").

D. Joining the Group as an Associate

The group administrator and the newly added member of the group must communicate with one another in order to bring the group's security information, which must include the pair of group keys known as PrKG (Private key) and PuKG, up to date (Public key).

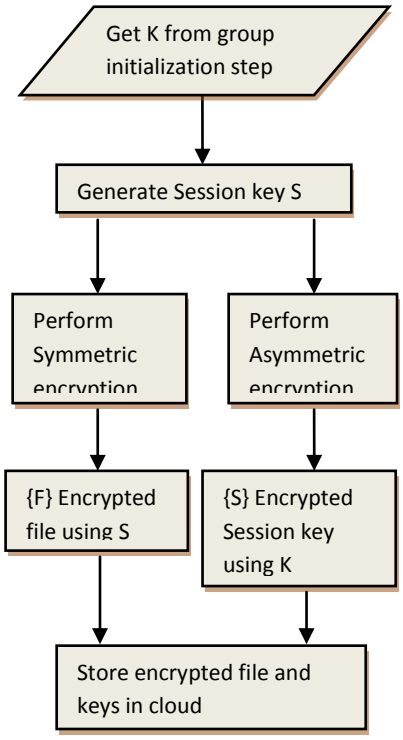

Figure 3: Digital Envelope Procedure

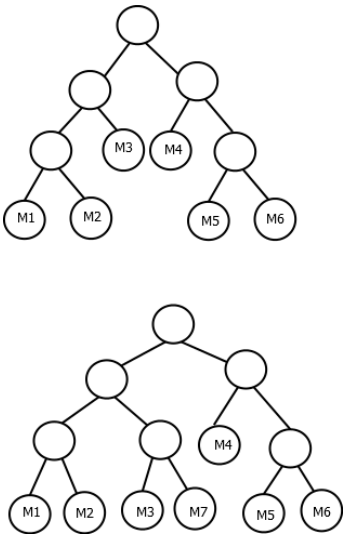

Figure 4: Group Member joining

E. One of the people in the group leaves.

The group administrator (GA) will launch a Binary Tree-based key generation process for the purpose of group key upgrading (BTKG).

The GA will then produce a re-encryption key by utilising the version of the group's public key that is currently being utilised in the digital envelopes. This step is necessary in order to obtain the new updated version. As shown in Figure 4, the group sharing structure is subject to change when mobile user M2 leaves the group.

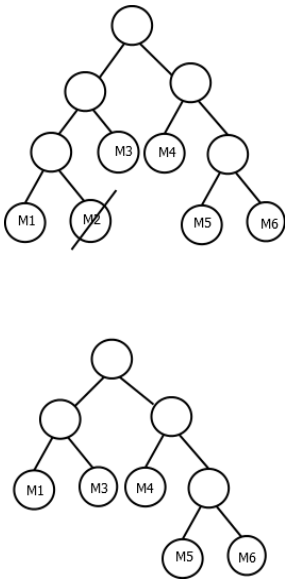

Figure 5: Group Member Leaving

F. The administrator of the group who is leaving

The administrator of the group will typically demand multiple leaf nodes, and he or she will be aware of the secret keys for all of the leaf nodes. As a result of the fact that these leaf nodes are required to be mandated, the security keys have to be updated, and the group's security details, including the group private key, have to be brought up to date whenever an administrator leaves the group.

IV. ANALYSIS OF PERFORMANCE RESULTS

When it comes to the actual deployment of a system, performance is invariably one of the most significant challenges. The methodology that was suggested provides security and privacy for data that is stored in the cloud. When it is implemented in a cloud environment, it provides a recognised level of security in addition to improved availability.

The amount of time required to upload a file varies according to its size, as shown in Figure 6. In addition, it demonstrates that the amount of time required to upload a file increases proportionally with the file's size.

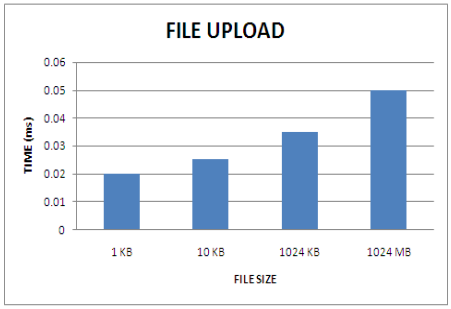

Figure 6: File Upload Time

The amount of time required to download the file is depicted in Figure 7 in relation to the file's size. It is abundantly clear that the amount of time required to download the file is significantly longer than the amount of time required to upload the file.

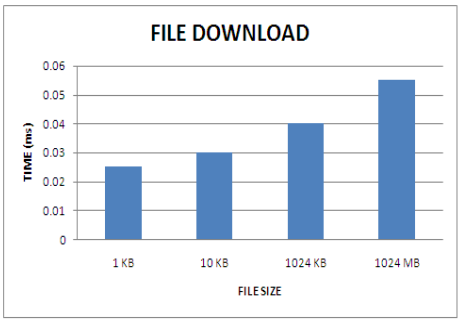

Figure 7: File Download Time

V. CONCLUSION

Under the strategy that has been proposed, all of the shared files are stored in a protected environment on cloud servers, and all of the session keys are encrypted and stored in digital envelopes. We use an upgraded BTKG scheme that is based on a cloud server to dynamically update the group key pair whenever members of the group leave or join the group. It is not necessary for every member of the group to be online at the same time for our plan to be successful. In order to provide forward secrecy and backward secrecy, digital envelopes should be upgraded so that they can transfer the majority of their computing overhead to cloud servers without revealing any security information. This should be done without revealing any sensitive information. According to the findings of the security and performance study, the proposed system might be able to satisfy the security requirements of public clouds while also maintaining a lower level of computational complexity and communication overhead on the parts of each member of the group.

REFERENCES

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

[2]. K. Ren, C. Wang, and Q. Wang, "Security problems for the public cloud," IEEE Internet Computing, vol. 16, no. 1, Jan.-Feb. 2012, pp. 69-73.

[3]. "Achieving safe, scalable, and fine-grained data access control in cloud computing", in Proc. IEEE 29th Conference on Computing Communication, 2010, pp. 534-542. S. Yu, C. Wang, K. Ren, and W. Lou.

[4]. "Towards security in sharing data on cloud-based social networks," in Proc. 8th International Conference on Information, Communication and Signal Processing, 2011, pp. 1–5. D. H. Tran, H. L. Nguyen, W. Zha, and W. K. Ng.

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

[6]. P. Tysowski and M. Hasan, "Hybrid attribute- and re-encryption based key management for safe and scalable mobile apps in clouds," IEEE Transactions on Cloud Computing, vol. 1, no. 2, July-December 2013, pp. 172-186.

[7]. "Attribute-based encryption for fine-grained access control of encrypted data," in Proc.13th ACM Conference on Computing Communication Security, 2006, pp. 89–98.

[8]. Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds, IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 2, pp. 384 to 394, February 2014. Sushmita Ruj, Milos Stojmenovic, and Amiya Nayak.

[9]. Michael Hitchens, Lan Zhou, and Vijay Varadharajan, "Achieving Secure Role-Based Access Control on Encrypted Data in Cloud Storage," IEEE Transactions on Information Forensics and Security, VOL. 8, NO. 12, pp. 1947–1960, DEC 2013.

[10]. "DAC-MACS: Effective Data Access Control for Multi-authority Cloud Storage Systems", IEEE Transactions on Information Forensics and Security, Vol. 8, No. 11, pp. 1790–1801, NOV 2013.