JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 8, ISSUE 05, 2021

Implementation Logical Key Hierarchy to a Nosql Database in Cloud Computing

[1]D Navya Associate Professor,

dubbaka.navya@gmail.com,

[2]Dr. M Bal Raju Professor,

drrajucse@gmail.com,

[3]M Arya Bhanu Associate Professor,

mabhanuu@gmail.com,

E Krishna Associate Professor,

krishna.cseit@gmail.com,

Department of CSE Engineering,

Pallavi Engineering College,

Kuntloor(V),Hayathnagar(M),Hyderabad,R.R.Dist.-501505.

Abstract:

Data, software, and services are stored in faraway data centres but may be accessed via the internet at any time and on any device that has internet connection. This is known as "cloud computing." Broadcast communication has become a pressing problem in many locations because to the fast rise of the Internet. Transmission of a message from a broadcast centre to all or part of the users who are linked to it is known as "broadcast communication." In order to convey data to several recipients from the same source, various systems have been created. This scheme's Logical Key Hierarchy is by far the most often used one (LKH). To demonstrate how LKH structure may be integrated into a Nosql database on cloud computing, two apps have been built as part of this research: one for broadcasting and the other for user usage.

INTRODUCTION:

Today, cloud computing is becoming more and more popular and important. Using cloud computing, consumers may access services from wherever they are, without the requirement for a device, infrastructure or software. Users may rent cloud computing services to utilise the system or applications. Alternatively, the user may be able to store and handle his or her own information. Organizations that offer cloud computing services are referred to as cloud computing providers. Providers of cloud computing services are responsible for guaranteeing the security of the infrastructure they supply to cloud consumers. In lieu of purchasing the hardware or software they need, users may just rent it from the cloud. Cloud computing provides a number of benefits, including device, time, and location independence, a robust hardware infrastructure, and lower costs. On the other hand, it comes with certain security issues in addition to these benefits. Systems, software, and data that benefit users are at the heart of security. Broadcast communication includes the sending of data to several consumers. In broadcast communication, encryption technologies are often used to send messages to a large number of people. At this stage, multicast transmission necessitates the optimization of encryption algorithms. Several major management strategies have been studied by Prathap and Vasudevan in one of these studies. For user add/removal procedures, they've developed a novel hybrid key tree structure that combines the benefits of both techniques [1]. Key Tree Reuse is a new key management strategy presented by Gu et al. in another research (KTR). As a result of KTR, users can register with the same key value for multiple broadcast programmes. Re-keying expenses are lower in this structure than in the LKH structure [2]. Using an asymmetric (public key) infrastructure, Song et al. have devised a novel method for managing group keys based on dynamic group membership. Even if the cloud server is attacked by malevolent users, the suggested scheme's use of public-key encryption ensures data security [3]. Diffie-Hellman key exchange implementation on the LKH structure was described in another paper by Alyani et al., who also sought to construct a key management system by altering the LKH structure. Increasing the number of users in the subgroups of the tree is the basis for this adjustment [4]. The users' path to the root node of the LKH tree may be shortened by Sakamoto et al. in another investigation. The suggested technique makes use of the Huffman algorithm [5]. Liu et al. have presented a novel tree structure that uses an intuitive search method to lower the cost of the key update when users are added or removed. The number of nodes at each level of the LKH tree is likewise variable in this

ISSN- 2394-5125 VOL 8, ISSUE 05, 2021

research [6]. In another research, Sakamoto claims that if the average number of users added or deleted from a key tree is known, the cost of the key update may be lowered [7]. When creating the LKH scheme's shared secret key, a Hellman scheme is used to ensure its safety. To enable secure broadcast communication from a source to cloud system users, the broadcast method should be implemented into the cloud. One of the broadcast systems now in use is the LKH structure, and it will be shown how to incorporate it into a Nosql database in the cloud. It has been designed two cloud-based mobile apps, one belonging to the broadcast centre and the other to the end-user.

LOGICAL KEY HIERARCHY (LKH):

It was Chung Kei Wong and his colleagues that devised the LKH system in 1997 [9]. An encryption key set and permitted users are part of this scheme's key tree structure. Figure 1 depicts a network in which users are located in the leaf nodes and the broadcast centre is located in the root node.

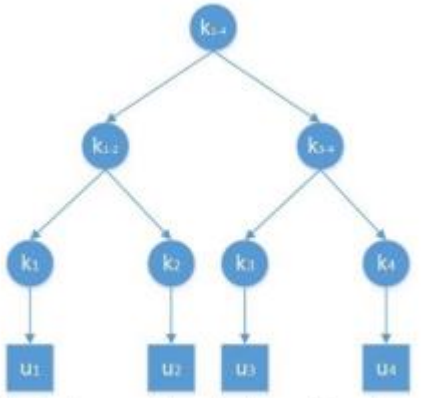

Fig. 1. An example broadcast encryption scheme.

Broadcast messages may be sent to all users simultaneously via the broadcast centre. Many different encryption techniques are used to protect the message. One-way Function Tree (OFT) [10], One-way Function Chain (OFC) [10] [11] and Tree-Based Group Hellman (TGDH) are other techniques that allow secure communication in addition to LKH. This structure is used in OFT, OFC and TGDH schemes. Users' broadcast centres calculate the tree's keys instead of the broadcast centre itself, like in LKH. A Key Server handles all user additions and deletions (KS). One-way broadcast messages may only be sent from the broadcast centre to the user's device. A symmetric key is provided to each user. The tree also contains a symmetric key based on the root node and its subordinates. A route from the user node to the root node for each user must be established to transfer secret key values. Keys are made and distributed by KS, which has a significant impact on the process. All users have 1 + logdn keys on a route from their node to the root node in the event that the tree is fully and evenly distributed. Users are represented by n and d, the number of degrees in the subset in which they are placed. KS produces and securely distributes keys according to the scheme. The LKH tree's structure is constantly changing. The LKH tree structure may be entered or exited at any moment by a user.

Secrecy must be maintained at all times when a user is added. Forward secrecy must be preserved when a user is removed from the tree. Using forward secrecy, an untrained user will be unable to decipher future broadcast signals. The purpose of back secrecy is to prevent a user who has been added to the broadcast environment from solving the historical messages that have been broadcast. All encryption keys in the route from the schematic location of the user to the broadcast centre must be changed to guarantee forward and backward secrecy when a user is added or removed from the system scheme. Users who need the new root node key and intermediate node key values may get them from KS.

APPLIED WORK:

The LKH method has been implemented in a cloud-based Nosql database using two mobile apps. Fig. 2 depicts the application that is used to carry out broadcast centre activities. Sending encrypted data to other users is possible with this app, as long as you have access to the current secret key. Text, picture, and video are all examples of data types. The broadcast centre programme encrypts the data with the current secret key before uploading it to the Nosql database. An email notice may also be sent out to users. Besides user keys, the database also includes the current keys of intermediate and root nodes, respectively. '

ISSN- 2394-5125 VOL 8, ISSUE 05, 2021

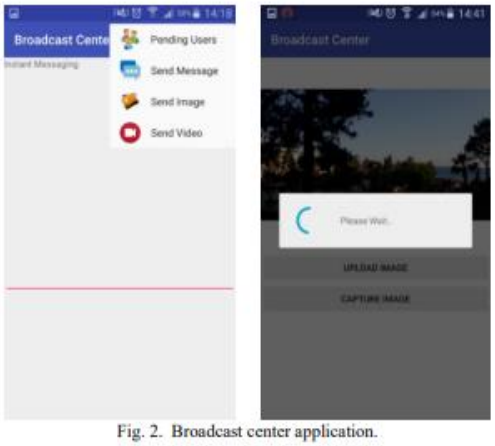

Fig. 2. Broadcast center application.

KS is part of the broadcast centre app. The broadcast centre is also in charge of acting as the company's principal administrator. This application allows users to accept or refuse membership in the tree. It is also possible to remove an individual from the tree at any point in time. In Fig. 3, you can see the programme that users are most likely to encounter. To begin, a user must first create an account and log in to the tree. Users who have been accepted by KS will have access to all future broadcasts. When the broadcast centre sends out a message, this app notifies its users. Secret key value may be used to decipher encrypted message.

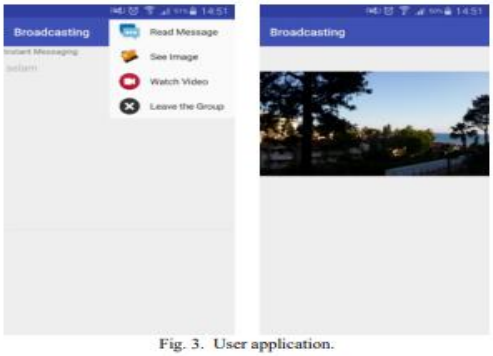

Fig. 3. User application.

The LKH scheme is used to add and delete users from the Nosql database. In the same way as LKH changes are executed on the Nosql database, so too are key updates. There are two binary positions for users when they're added to the tree. For example, in a tree with 8 users and 3 degrees, the position value of the first user is 111, and the position value of the eighth user is 111. Additions to the tree will raise its depth, which in turn will update all current user position values as well as assign new position values to newly added users. Firebase, a NoSQL database, is utilised in the research to construct mobile apps. The database has a number of tables. There are two tables, one for the user's location and the other for the current secret key and the depth information of the tree, as illustrated in Figs. 4 and 5 respectively. Fig. 6 shows a user database that includes the user's name, surname, phone number, token that is used for notification operations, username and password values.

Fig. 4. Positions table.

Fig. 5. Settings table.

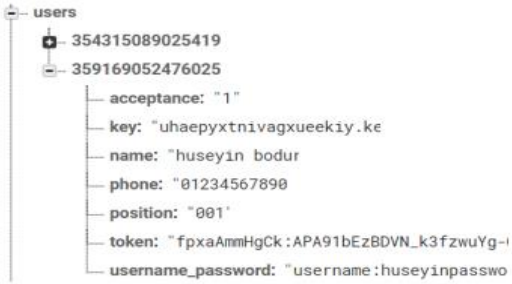

Fig. 6. Users table.

Many additional tables are added to the database as the depth of the tree or the number of messages sent from the root node rise. Images and videos are stored in tables that include the secret key used to encrypt and send them, as shown in Figures 7, 8, and 9, while text messages are stored in a database that contains information about text messages that are delivered to users from a root node, as shown in Figures 7 and 8.

Fig. 7. Intermediates nodes table.

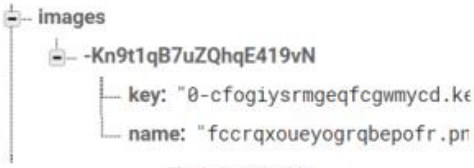

Fig. 8. Images table.

Fig. 9. Messages table.

ISSN- 2394-5125 VOL 8, ISSUE 05, 2021

There are certain files in the database that are also accessible. These files include the keys file, which contains the key values of all users and intermediate nodes. There are also photographs and videos, both of which include encrypted images and video.

ANALYSIS AND COMMENTS

No matter how many people utilise the cloud server, a broadcast centre may convey text, images, or video data for a cheap cost of encryption by employing powerful encryption methods. In addition, powerful cloud servers may be used to add, remove, and rekey users, and broadcast messages can be stored on cloud servers that are quick and dependable. After each user is added or deleted, all node keys are updated from the user node to the root node to maintain forward and backward secrecy. Confidentiality concerns are a major concern when LKH is applied to the cloud. These issues may be resolved in two ways. Using the old secret key value and encrypting the data with the new secret key value is the first option for decrypting the cloud-stored data. However, as a result of this, computational costs and resource use become quite high. There is also the option of storing secret keys on the cloud and keeping encrypted data's key information. Even if the cloud server's secret key is altered, the previous values must be preserved. The second approach, on the other hand, does not need additional computation expenses.

CONCLUSION

Two cloud-based Nosql apps based on the LKH scheme have been built as part of the research. Both apps are designed to be used on mobile devices. Briefly, the pros and downsides of using the LKH cloud method are discussed. Other frequently used broadcast techniques will be included into the cloud in future studies. According to the cost of computation, encryption, and user keys, the schemes will be compared to each other.

REFERENCES

[1] Prathap M Joe and Vasudevan V 2009 Analysis of the various key management algorithms and new proposal in the secure multicast communications arXiv preprint arXiv:0906.3956.

[2] Gu Q, Peng L and Wang-Chien L 2009 KTR: An efficient key management scheme for secure data access control in wireless broadcast services IEEE Transactions on Dependable and Secure Computing 6.3 p 188-201.

[3] Song W, Zou H, Liu H and Chen J 2016 A practical group key management algorithm for cloud data sharing with dynamic group China Communications 13.6 p 205-216.

[4] Alyani N, Seman K, Nawawi NM and Sayuti MNSM 2012 The Improvement of Key Management Based On Logical Key Hierarchy by Implementing Diffie Hellman Algorithm J. Emerging Trends in Computing and Information Sciences 3.3.

[5] Sakamoto T, Tsuji T and Kaji Y 2008 Group key rekeying using the LKH technique and the huffman algorithm Information Theory and Its Applications (ISITA) p 1-6.

[6] Liu H, Li J, Hao X and Zou G 2014 A novel LKH key tree structure based on heuristic search algorithm Communication Problem-Solving (ICCP) p 35-38.

[7] Sakamoto N 2014 An efficient structure for LKH key tree on secure multicast communications In Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD) p 1-7.

[8] Bodur H and Kara R 2017 Implementing Diffie-Hellman key exchange method on logical key hierarchy for secure broadcast transmission Computational Intelligence and Communication Networks (CICN), p 144-147.

[9] Wong CK, Gouda M and Lam SS 1998 Secure group communications using key graphs IEEE/ACM transactions on networking 8.1 28 p 16-30.

[10] Sherman AT and McGrew DA 2003 Key establishment in large dynamic groups using one-way function trees IEEE transactions on Software Engineering 29.5 p 444–458.

[11] Canetti R, Garay J, Itkis G, Micciancio D, Naor M, et al. 1999 Multicast security: A taxonomy and some efficient constructions in INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings. IEEE 2 p 708- 716.

[12] Kim Y, Perrig A and Tsudik G 2004 Tree-based group key agreement ACM Transactions on Information and System Security (TISSEC) 7.1 p 60– 96