Implementation Logical Key Hierarchy to a Nosql Database in Cloud Computing

Hüseyin Bodur[1,*], Resul Kara[1]
[1]*Department of Computer Engineering, Düzce University, 81620, Konuralp/Düzce, Turkey*

***Abstract*—Cloud computing is a system that keeps the system, software or data contained in remote data centers and enables them to access at a desired time and on a desired device over the internet. With the rapid growth of the Internet, broadcast communication has become an issue to be implemented in many areas. In shortly, broadcast communication is the transmission of a message from broadcast center to all or some of the users which connected to it. Various schemes have been developed to allow a single sender to transmit a data to multiple users. The most common use among these schemes is Logical Key Hierarchy (LKH). In this study, two applications have been developed to explain how to integrate LKH structure into a Nosql database on cloud computing, one of which is broadcasting center and the other is user application.**

***Keywords*— Cloud Computing, Logical Key Hierarchy, Nosql.**

I. INTRODUCTION

Today, the usage rate and importance of cloud computing is constantly increasing. Cloud computing is an information system that enables users to receive services from the place in which they are without needing any device, infrastructure or software. The user can use the system or software on the cloud by renting cloud computing services. The user may also have the possibility to store and process his own data.

Cloud service provider organizations are called cloud computing providers. Cloud computing providers are responsible for providing infrastructure to users on cloud and ensuring the security of this infrastructure. Users can rent services over the cloud and reduce their needs at a lower cost, instead of buying the systems or software they need. Cloud computing has advantages in many aspects such as device, time and space independence, strong hardware infrastructure and cost. However, in addition to these advantages, it brings some security problems.

The privacy of the systems, software and data that users benefit from are at the core of security.

The transmission of a data to multiple users is included in the broadcast communication area. Encryption methods are usually used to transmit messages to multiple users in broadcast communication. At this point, encryption methods must be optimized in accordance with multicast transmission.

In one of these studies, Prathap and Vasudevan have examined various key management schemes. They have proposed a new hybrid key tree structure by combining the advantages of these methods for user add / remove operations [1]. In another study, Gu et al. have proposed an effective key management scheme called Key Tree Reuse (KTR). KTR is a new key management approach that allows users to register with the same key value to multiple programs in the broadcast system. Although it is LKH based, it has lower re-keying costs than the LKH structure [2].

In another study, Song et al. have proposed a new group key management algorithm based on the public key (asymmetric) infrastructure to enable encrypted cloud data sharing to dynamic groups. Data security is provided through the proposed scheme, which takes advantage of public-key encryption, even if exposed to attacks by malicious users on the cloud server [3].

In another study, Alyani et al. have given information on how to implement the Diffie-Hellman key exchange on the LKH structure and attempted to develop the key management scheme by modifying the LKH structure. This modification is based on increasing performance by increasing the number of users in the subsets of the tree [4].

In another study, Sakamoto et al. have proposed a scheme to reduce the length of the LKH tree from the users to the root node. They used the Huffman algorithm in the proposed scheme [5].

In another study, Liu et al. have proposed a new tree structure based on an intuitive search algorithm to reduce the cost of the key update performed after adding / removing users. The study also has a different number of nodes at each level of the LKH tree [6].

In another study, Sakamoto has proposed a study that argues the cost of the key update can be reduced if the average number of users added to or removed from a key tree is known [7]. In another study, how to use the Diffie-Hellman scheme is explained for the securely creation of the common secret key in the LKH scheme [8].

The broadcast scheme to be used should be integrated into the cloud in order to ensure secure broadcast communication from a source to users in a cloud system. In this study, how to integrate LKH structure, which is one of the broadcast schemes used today, into a Nosql database on the cloud will be explained. Two mobile applications, which integrated into the cloud structure and one belonging to the broadcast center and the other to the users, have been developed.

II. LOGICAL KEY HIERARCHY (LKH)

The Logical Key Hierarchy (LKH) scheme was developed by Chung Kei Wong and his team in 1997 [9]. The purpose of this scheme is to create a key tree structure which contains an encryption key set and authorized users.

Manuscript received December 09, 2018; accepted December 27, 2019.
*Corresponding author: huseyinbodur@duzce.edu.tr

Users are in the leaves and the broadcast center in the root node in this scheme, as shown in Fig. 1.

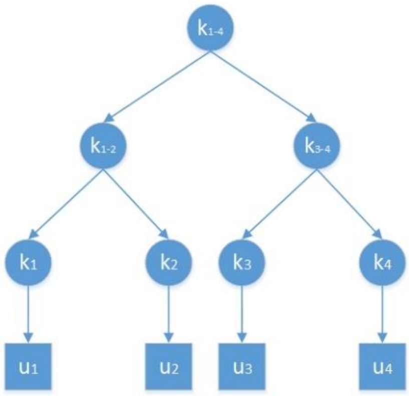


Fig. 1. An example broadcast encryption scheme.

The broadcast center can send broadcast messages to all users at once. Various encryption algorithms are used for the message security. In addition to the LKH scheme, there are also schemes that provide secure communication such as One-way Function Tree (OFT) [10], One-way Function Chain (OFC) [10] [11] and Tree-Based Group Diffie-Hellman (TGDH) [12]. OFT, OFC and TGDH schemes are based on a binary tree structure like LKH. Unlike LKH, the keys on the tree are computed from users to broadcast center in these schemes. User adding / removing operations are performed by a Key Server (KS).

Broadcast messages can only be forwarded one-way to the users from the broadcast center. Each user has a symmetric key. The tree also has a symmetric key of the root node and intermediate nodes. The secret key values must be transmitted on a path from the user node to the root node for each user.

KS has great importance in the creation and distribution of keys. If the tree is full and balanced, each user stores a total of $1 + log_d\, n$ keys on a path from his / her node to the root node.

d represents the degree of the subset in which the user is located and n represents the number of users in the tree. KS generates keys on the scheme and distributes them to users in a secure way.

The LKH tree has a dynamic structure. A user may want to join or leave the LKH tree structure at any time When a user is added to the tree, backward secrecy must be ensured. When a user is removed to the tree, forward secrecy must be ensured.

Forward secrecy is to prevent a user left to the broadcast environment from solving future broadcast messages. Back secrecy is to prevent a user added to the broadcast environment from solving history messages.

When a user is added, or deleted in scheme, all encryption keys in the path from the user's schematic position to the broadcast center must be updated to provide forward and backward secrecy. Then, KS distributes the updated root node key and the updated intermediate node keys to users who need these key values.

III. APPLIED WORK

Two mobile applications have been developed for the implementation of the LKH scheme in a Nosql database on cloud computing. The first of these is the application which broadcast center operations are performed, as shown in Fig. 2. This application allows the sending of encrypted data to users using the current secret key.

The data type can be of text, image, or video. Before the data is sent to the users, it is encrypted with the current secret key in the broadcast center application and then uploaded to the Nosql database. It is also sent to users via a notification service. In addition to the user keys, the database also contains the current keys of the intermediate node and the root node.

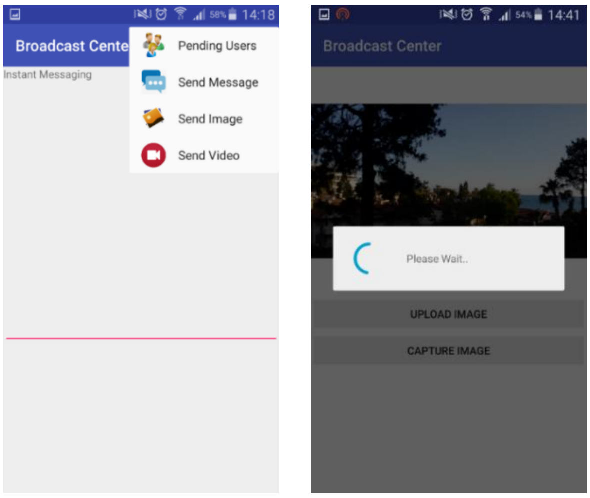


Fig. 2. Broadcast center application.

KS is included in the broadcast center application. The broadcast center also has the key manager role. A user who wants to join the tree can accept or reject through this application. A user can also be removed from the tree. Fig. 3 shows the application that users use. Through this application a user firstly registers to the tree. If KS accepts user, user can access future broadcast messages.

Users receive notifications via this application when a message is sent from the broadcast center. They can decrypt the encrypted message with the current secret key value.

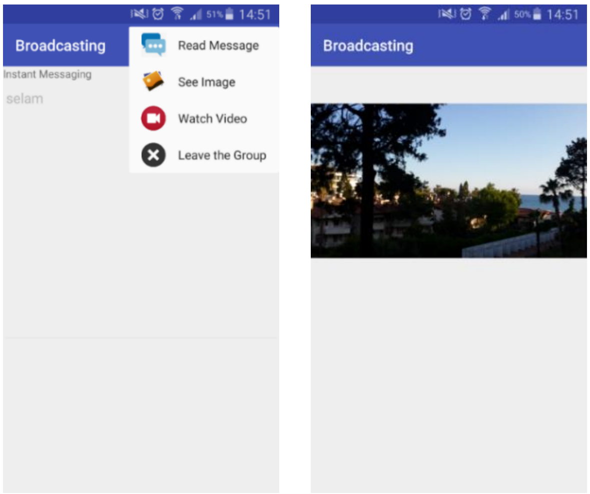
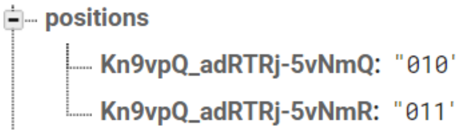


Fig. 3. User application.

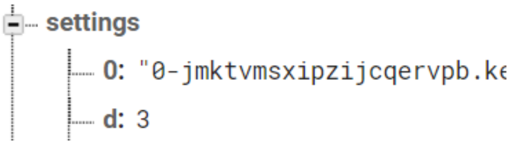
Users are added to or removed from the Nosql database as in the LKH scheme. Likewise, key updates on the Nosql database are performed as in the LKH scheme. When each user is added to the tree, s/he is given a binary position value. For example, the position value of the first user is 000 and the position value of the eighth user is 111 in a tree with 8 users and 3 degree. If the tree is completely full and a user is to be added, the depth of the tree is increased and the position values of the existing users are updated and new position values are obtained. In the study, mobile applications are developed using Android Studio and Firebase is used as a Nosql database.

There are some tables on the database. These tables; a positions table which contains the appropriate position value to be given to the user, as shown in Fig. 4, and a settings table which contains the current secret key and the depth information of the tree, as shown in Fig. 5. A users table which contains user's name, surname, phone number, token which used for notification processes, username and password values, as shown in Fig. 6.



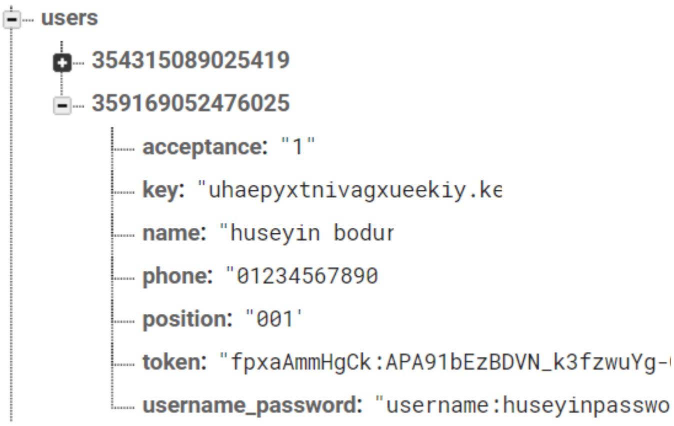
```
positions
    Kn9vpQ_adRTRj-5vNmQ: "010'
    Kn9vpQ_adRTRj-5vNmR: "011'
```

Fig. 4. Positions table.



```
settings
    0: "0-jmktvmsxipzijcqervpb.ke
    d: 3
```

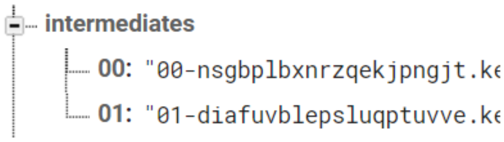
Fig. 5. Settings table.



```
users
    354315089025419
    359169052476025
        acceptance: "1"
        key: "uhaepyxtnivagxueekiy.ke
        name: "huseyin bodur
        phone: "01234567890
        position: "001'
        token: "fpxaAmmHgCk:APA91bEzBDVN_k3fzwuYg-
        username_password: "username:huseyinpasswo
```

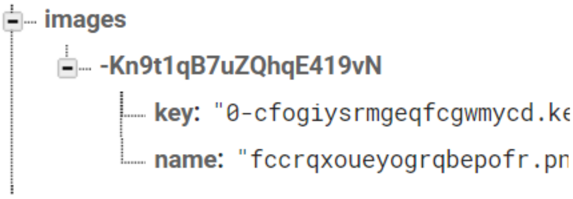
Fig. 6. Users table.

Several more tables are added to the database as the depth of the tree increases or various messages are sent from the root node to the users. These tables; the intermediate nodes table that contains the current key information as shown in Fig. 7, the videos and pictures table which contain the video and image data which are encrypted by the current secret key and sent as shown in Fig. 8, the messages table which contains text messages sent to users from the root node as shown in Fig. 9.



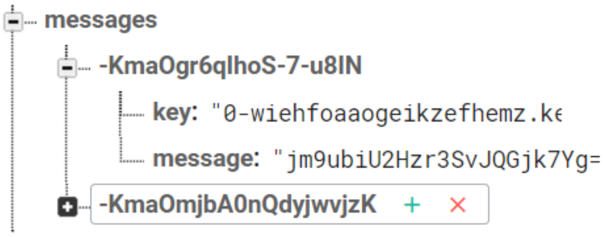
```
intermediates
    00: "00-nsgbplbxnrzqekjpngjt.ke
    01: "01-diafuvblepsluqptuvve.ke
```

Fig. 7. Intermediates nodes table.



```
images
    -Kn9t1qB7uZQhqE419vN
        key: "0-cfogiysrmgeqfcgwmycd.ke
        name: "fccrqxoueyogrqbepofr.pn
```

Fig. 8. Images table.



```
messages
    -KmaOgr6qlhoS-7-u8IN
        key: "0-wiehfoaaogeikzefhemz.ke
        message: "jm9ubiU2Hzr3SvJQGjk7Yg=
    -KmaOmjbA0nQdyjwvjzK   +  ×
```

Fig. 9. Messages table.

Some files are also kept in the database. One of these files is keys file that holds the key values of all users and intermediate nodes. The other files are a pictures file which contains encrypted pictures and a videos file which contains encrypted videos.

IV. EVALUATION

With this study, a broadcast center can send a text, image or video data over a cloud server at a low cost of encryption, using strong encryption algorithms, regardless of the number of users. In addition, performing user adding / removing and re-keying operations with powerful cloud servers and storing broadcast messages on the fast and reliable cloud servers are the advantages of study.

All node keys are updated from the respective user node to the root node after each user is added / removed operations in order to ensure both forward and backward confidentiality. One of the most important problems that arise with the implementation of the LKH scheme to cloud computing is how to ensure the confidentiality of data stored in the cloud. There are two options for these problems. The first option is to decrypt the data stored in the cloud with the old secret key value after the update of the broadcast center key and to encrypt it with the new secret key value. But this leads to very high computational costs and redundant resource usage. Another option is to store the secret keys in the cloud and to keep key information about encrypted data, as shown in Fig. 8 and Fig. 9. In this case, even if the secret key is updated on the cloud server, the old secret key values must be kept. But this does not lead to unnecessary calculation costs, such as the first option.

V. CONCLUSIONS

In the study, two applications have been developed on how to integrate the LKH scheme into a Nosql database on the cloud. Both applications are mobile based. The advantages and disadvantages of the LKH scheme on the cloud are briefly mentioned.

In the future study, other widely used broadcast schemes will also be integrated into the cloud. The schemes will be compared to each other in terms of calculation cost, encryption cost and number of keys in the user.

ACKNOWLEDGMENT

This study was presented orally as abstract paper at the ICONDATA 2018 conference.

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.