Journal of Theoretical and Applied Information Technology
20th February 2013. Vol. 48 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SECURE KEY SHARING FOR GROUP COMMUNICATION UNDER COMMUNITY CLOUD

[1]K. GOVINDA, [2]E.SATHIYAMOORTHY

[1]AP(SG), SCSE, VIT University, Vellore, India
[2]Assoc. Prof., SITE, VIT University, Vellore, India
E-mail: [1]kgovinda@vit.ac.in, [2]esathiamoorthy@vit.ac.in

ABSTRACT

The cloud is one of the most discussed topics among IT professionals today, and organizations are increasingly exploring the potential benefits of using cloud computing or solutions for their businesses. In our modern world people join hands together to form groups and that forms communities through which the ability of one can enhance by communicating with others. In order to satisfy these people over cloud, transformed itself as community cloud. As a cloud infrastructure that is shared by several organization's and supports a specific community, such as healthcare, telecommunication, education and businesses that has shared concerns around mission, policy and compliance considerations. The goal of a community cloud is to have participating organizations realize the benefits of a public cloud -- such as multi-tenancy and a pay-as-you-go billing structure but with the added level of privacy, security and policy compliance usually associated with a private cloud. The community cloud can be either on-premises or off-premises, and can be governed by the participating organizations or by a third-party managed service provider (MSP). In this community cloud large number of groups which were formed by huge number users. At this situation in order to have secure communication and data storage in cloud, consumers use keys. Though these keys were kept private in many cases they are in need to be shared at least once for the sake of security. So, in this paper we are going to discuss a secured protocol through which a community cloud users can share keys among themselves in a secured manner.

Keywords: *Key Graph, Diffie- Hellman, Community Cloud, Super Group, Sub Group.*

1. INTRODUCTION

Cloud is one of the most intelligent, elegant, powerful and useful platform. Since the cloud provides on-demand high quality data storage service. The data owners can be comforted from the burden of data storage and safeguarding of the data's. However, the fact that data owners and cloud server are not in the same trusted domain may put the outsourced data at risk, as the cloud server may no longer be fully trusted[1][2][3]. It follows that sensitive data usually should be encrypted prior to outsourcing for data privacy and struggle unwanted accesses. Since there are plenty of outsourced data files in the cloud data utilization become a challenging task. Moreover, the data owners may share their outsourced data with a large number of users in cloud. The individual users might want to only retrieve certain specific data files they are interested during a given session.

In most of the cases cloud uses secret keys or passwords to authenticate the user and verifies and retrieves the data based on the result [8][9]. Among these in all the cases there comes a session in which the key is in need to be shared. While the keys are getting shared there are chances to be get hacked by the hackers or others. Hence we take this chance to propose a protocol through which a community can share and also exchange any number of times as they wish in a secured manner [6][7]. The following paper will now show how the procedure flow brings the security.

The rest of the as follows, Chapter2 describes literature review, The proposed method described in chapter3, Chapter4 details the secure communication, Results are shown in Chapter5 followed by conclusion.

2. LITERATURE REVIEW

As described before the communication of keys in cloud is a most common thing to happen. There are chances for the keys which hold all our security of our digital assets we are in a situation to provide

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

data security by enabling secure key sharing in our cloud. The most widely-deployed authentication system for online services is also the least expensive: "something you know", typically a username and (likely weak) password [6]. Authentication credentials are revealed to the computer that they are typed on and the centralized service's provider, both of which have the ability to violate the user's security expectations and must thus be trusted absolutely.

Consider a communication in which a group is involved there as usual they are in need to share the group key among them they primarily uses some of the secure key sharing algorithm but when they compute the common key the server distributes the keys as it is[6]. This becomes advantage for the hacker which makes him easy to hack the keys by simply listening to the communication [8]. Threats are no longer the purview of isolated hackers looking for personal fame. More and more, organized [4].

There are systems that supports secure key Sharing like the Powerful Diffie-Hellman Algorithm [9][10]. There are so many encryption techniques that allow the user to communicate securely. These algorithms always help in providing Security over cloud model like private cloud. But when it comes for Community Cloud, we are in need to enhance the security for the model. In most cases the Community Cloud uses the Hybrid Cloud Model (HCM) [15]. Here we strengthen the protocol using both the practice of Secure Key Exchange as well as the method of Encryption [14][15]. In order to propose the procedures for the secure key sharing in our paper we are using the key graph data structure which helps in optimize the proposal. This led us to the secure key sharing in the community [7].

3. PROPOSED METHODOLOGY

In our proposed method we form the community cloud with the help of Hybrid Cloud, Where more than two cloud model which are private in basic join together to form a community under an another private cloud which holds them as a member. The community cloud that holds the other clouds and being the authority to enable secure communication among the users and across their cloud that comes under his community provides the key to share the message, data securely [4]. In order to provide the key the server in cloud generates the key instead of sending the key as it is for security purposes the steps for key sharing is given below. Our Proposed methodology uses the Key-Graph model to make the community more strong and static. A key graph is a directed acyclic graph with two types of nodes, In our proposed methodology each of the U nodes representing users and the K-nodes representing keys. Each U-node has one or more outgoing edges but no incoming edge. Each K-node has one or more incoming edges. If a K-node has incoming edges only and no outgoing edge, then this K-node is called a root. (A key graph can have multiple roots.) Given a key graph, it specifies a secure group as follows.

1. There is a one-to-one correspondence between and the set of U-nodes in the graph.

2. There is a one-to-one correspondence between and the set of K-nodes in the graph.

3. U and K lies in the graph If and only if has a directed path from the U-node that corresponds to the K-node that corresponds to k.

In the given figure the nodes describes:

U = {u1, u2, u3, u4… u12}

K = {k1, k2, k3, k4… k12}

As the first step the Community cloud now authenticates the other private cloud. After verification the requested cloud is allowed to join the Community. Now the community is maintained at two levels of servers. From this the community can be viewed in the form of a tree in which the key sharing occurs [4].

1. A server to hold each and every cloud that joins the community (Level 2)

2. A server that holds the entire level 2 server (Level 1).

From now the level 1 server will be known as the Super Community Server (SCS) and the sub servers that holds the groups will be known as Sub Group Server (SGS). In our protocol the SCS connects with the entire SGS [4] as shown in Fig1. The SGS connects the users of the group. Thus the tree structure controls the key sharing flow.

Journal of Theoretical and Applied Information Technology
20th February 2013. Vol. 48 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

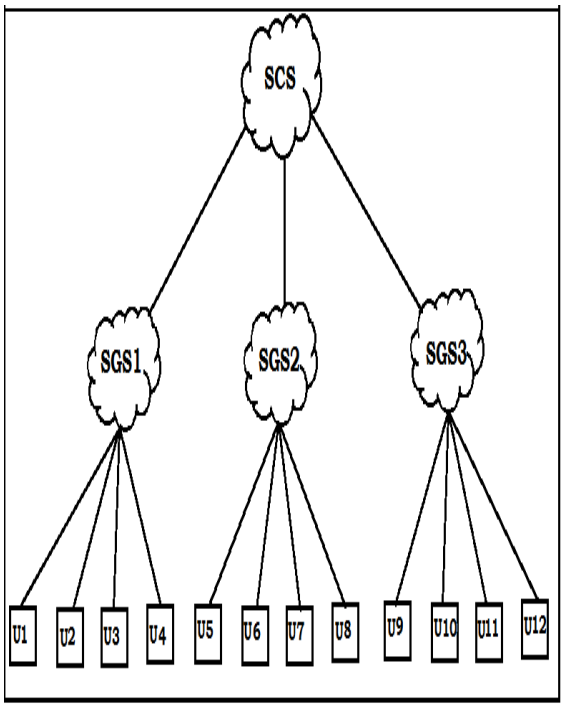

Figure 1. User's under Community Cloud

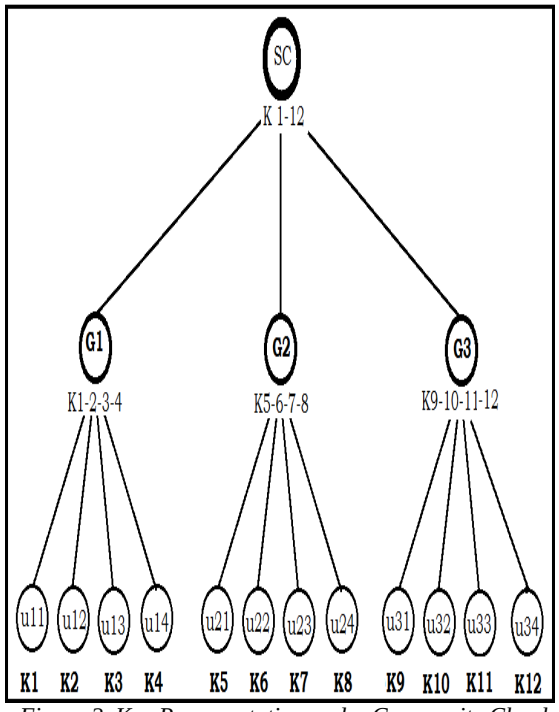

Figure 2. Key Representation under Community Cloud

3.1 Super Community Server(SCS)

The SCS holds all the SGS in order to provide the connectivity between the SGSs. The server is capable of communicating as Unicast or Multicast at the same time with the SGS. This is the server which generates the required keys for the users. This is the server responsible for the communication among the users.

3.2 Sub Group Server(SCS)

The SGS connects the primary connection that connects the users to the community .This server is also capable of communicating as a unicast or multicast connection with the users connect to it. This server is responsible for the key sharing and group key holding among the users. This server plays a major role in the protocol by identifying the user of its own group and making him an active participant of the community.

3.3 Keys In Community Cloud

The each and every single user in the community uses three keys with respect to the proposed protocol as shown in Fig2.

1. User's Private Key – [α]
2. Sub Group Key - [β]
3. Super Group Key or Community Key - [γ]

The user's private key that the user shares a key with the SCS. The Group Key is a key the key generated and distributed to the users of the respective group by the SCS. This key is maintained by the SGS. This Key helps in identifying the user behalf of his group since it is common inside the group. The Community key is the key which is common all over the community. The procedure that deals with the keying, re-keying, secure key sharing of the set of key discussed below.

4. SECURE KEY COMMUNICATION

As the first step the groups connect the community server and gets Authenticated by the SCS. Then the SCS allocate a SGS the groups, hence the number of SGS is equal to the number of groups gets connect to it. Thus each and every single group gets connected with the Community. Then each and every single user communicates with SCS and shares a private key using Diffie Hellman's Algorithm and this key is the user's private key. Since the communication between the user and the super server occurs group wise the sever can identify the user's group. This step loops for all the groups that are connected to the group. Now the server has the private keys of all the users in group wise as shown below.

Journal of Theoretical and Applied Information Technology
20th February 2013. Vol. 48 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

S → {u1.....u12} : {k1-12} ki ,
where i = 1....12
S → {u1.....u4} : {k1-4}ki
S → {u5.....u8} : {k5-8}ki
S → {u9.....u12} : {k9-12}ki

The group key for a group is given by the server by combining the keys of all the user of the group. In the same way the server generates the Super key or Community Key by combining the keys of the users all over the Community [4]. Now both the community key as well as the sub group key is taken and encrypted with the private keys individually and unicasted all over the community. The user will get the key set which is encrypted with his private key. Then he decrypts the set and recognizes the community key and sub group key. Thus the initial key sharing ends securely. When it comes for the topic of group or community join or leave becomes more common .In most of the cases leakage of message or data loss in group occurs while join or leave in order to provide security, We here produce a special procedure on our protocol which benefits both the user as well as the protocol in maintaining the security.

4.1 Joining the Community

When a user joins under the group which is connected to the community the user who belongs to that specific group must share new private keys with the SCS. The SCS computes the new sub group and encrypts the sub group key with the private keys of the user and unicast it to the respective users of the sub group through the SGS. Then the new group key is calculated with the new keys shared and the new group key is encrypted with the sub group keys and multicasted to the respective entire group through the SGS as shown in Fig3.

S → {u1.....u11} : {k1-12} k1-11

S → {u9.....u12} : {k9-12}k7-9

S → {u12}: {k9-12,k1-12}k12

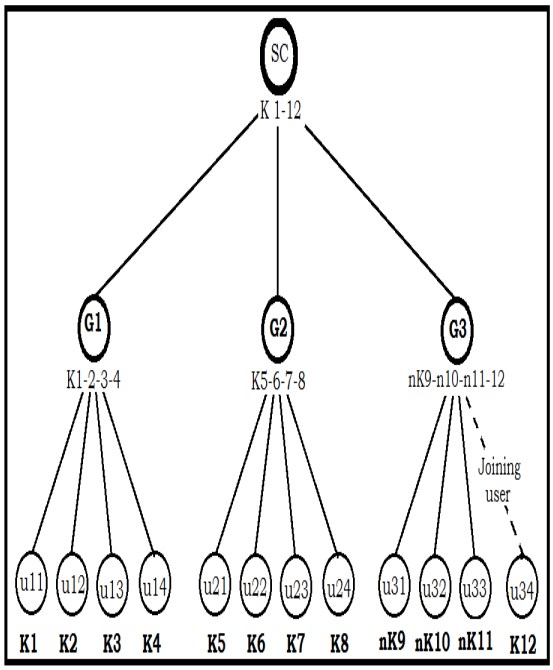

Figure 3. User Joining in Community Cloud

4.2 Leaving the Community

When the user leaves the group that is connected to the community, the remaining users in the group shares new private key with the server again. With this the server computes a new sub group key for that specific group and sends it by encrypting the sub group key with the newly shared private keys of the group to the respective users. Then the new group key is computed with the newly shared keys and multicasted over the groups in the community by encrypting the community key with the sub group keys of the users for the sake of security as shown in Fig4..

S → {u1.....u11} : {k1-11} k1-4

S → {u5.....u8} : {k1-11} k5-8

S → {u9.....u11} : {k9-11} ki

S → {u9.....u11} : {k1-11} k9-11

Journal of Theoretical and Applied Information Technology
20th February 2013. Vol. 48 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

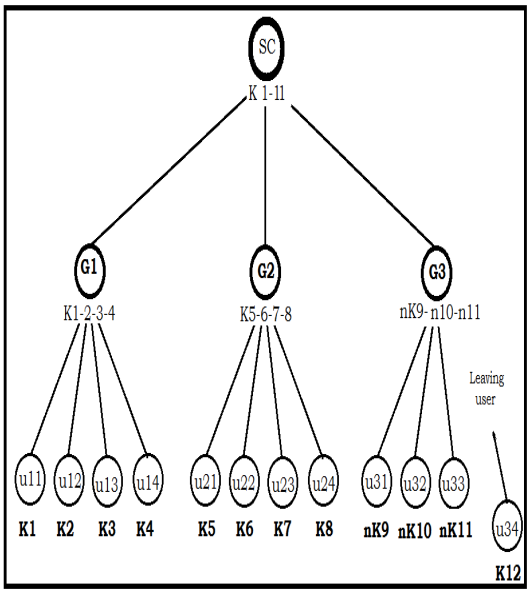

Figure 4. User leaves from Community Cloud

5. RESULTS

We built a Windows Azure cloud on server with the powerful and intelligent AMD Althon processor which has 6 layers of each layer containing 12 cores that can process 64 bit at a same time. The Fig5.shows clearly the number of keys generated with respect to the key size in the above mentioned hardware configuration. When the key size is increased to 512 bits the number of keys generated is 9. Similarly when it is twined the number of keys generated will be 4 per second.

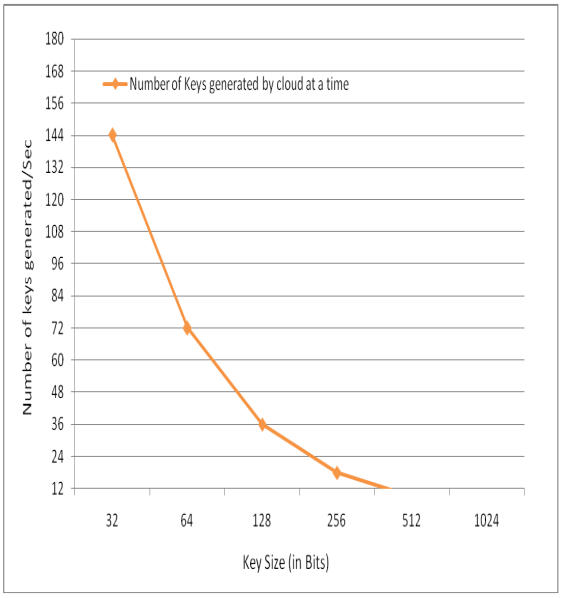

Figure 5. Shows the Key Generation

The Fig6. Shows CPU processing time Vs key size

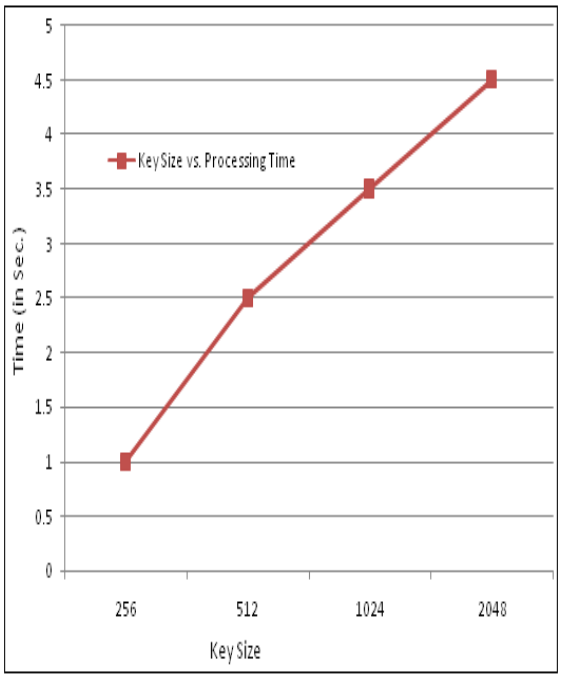

Figure 6. Shows Key size Vs Processing time

The Fig7. Shows time in sec for joining users in community cloud.

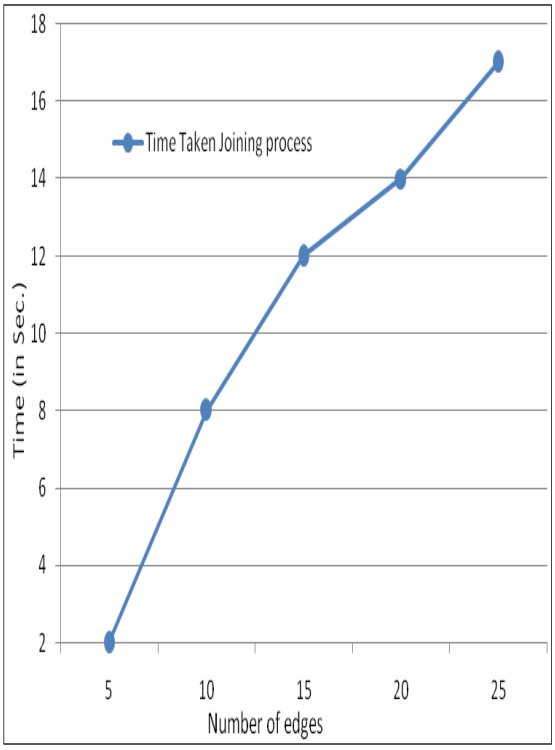

Figure 7.User Joining the Community

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The Fig8. Shows time in sec for leaving users in community cloud.

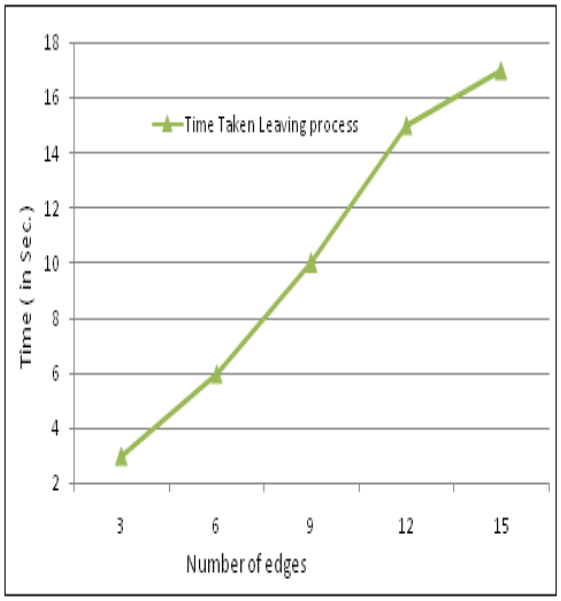

Figure 7. User Leaving the Community

6. CONCLUSION

In our work we proposed a protocol through which we can share keys in a secured manner under community cloud. Here our proposal is constructed favoring the security at the same time it also optimizes the community by ignoring unwanted key generation and sharing which leads time optimization. At the same time keys are securely shared by providing attention as a peer connection on the user and also with the help of the encryption technique and our proposal become stronger. In future this can also be done by using some Hash Values attached with the exchanging keys which will give more integrity to our work.

REFRENCES:

[1] M. Haynie, "Enterprise cloud services: Deriving business value from Cloud Computing," Micro Focus, Tech. Rep., 2009.

[2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz,A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley view of Cloud Computing," University of California, Berkeley, Tech. Rep.,2010.

[3] M. Papazoglou, "Service-oriented computing: concepts, characteristics and directions," in International Conference on Web Information Systems Engineering,2011.

[4] Planning Guide-Cloud Security Seven Steps for Building Security in the Cloud from the Ground Up. Intel Cloud.

[5] Chung Wong,Mohamed Gouda , Simon S. Lam, "Secure Group Communications Using Key Graphs",IEEE,2009.

[6] Jonathan Anderson, Frank Stajano," On Storing Private Keys - In the Cloud" .

[7] Eun-Jun Yoon, Kee Young Yo , "An Efficient Diffie-Hellman –MAC Key Exchange Scheme",2009.

[8] I.R.Jeong, J.O.Kwon, D.H.Lee," Strong Diffie-Hellman –DSA Key Exchange",IEEE Transactions Vol11, No5 May 2007.

[9] P. Bhattacharya, M. Debbabi and H. Otrok," Improving the Difffie-Hellman Secure Key Exchange",2005.

[10]David Carts, "A Review of the Diffie Hellman Algorithm and its use in secure internet protocol",2011.

[11]T. ElGamal – " A Public Key CryptoSystem and a Signature Scheme Based on Discrete Logarithms", IEEE Transactions on Information Theory Vol31, No4 July 1985.