© 2015 IJSRSET | Volume 1 | Issue 1 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Batch Rekeying Management for Subscription Based Mobile User

Aarthi S[1], Karthika[2], Meena M[3], Geetha Rani S[4]
Dhanalakshmi College of Engineering, Chennai, India

ABSTRACT

In mobile computing, to provide entry to batch members batch key will be issued and gets upgraded whenever the batch gets rescinded. This will cause regeneration of batch key high up and extend the conveyance value which will degrade the performance. We have given a concept of batch rekeying in which, the conveyance cost gets reduced accordingly by evolving two techniques, named initial tree development and idealisation of the path tree. The first technique involves the setting up of the path in the way to drop the batch key high up and in turn; the second technique will make the path tree to stay idealised.

Keywords: Batch Key Management, Batch Id, Multichannel, Dynamic Contributors

I. INTRODUCTION

In the future, Mobile Computing is going to be most important diligence in the actual world for providing services to the contributors. As the multichannel conveys the information to the batches the same time, the conveyance cost gets decreased. The only problem is that the identity needs to be changed whenever the batch gets reformed. Hence, it causes vulnerability and the performance gets degraded. This problem can be resolved by allowing the authorized persons and providing proper access mechanisms. As the mobile phones are getting increased, the applications are developed based on the batching or grouping.

To provide both protection and proficiency, a batch Id will be provided to each and every member in the particular batch for accessing and can be accessed only by the persons having that batch Id. If any information needs to be conveyed to the batch means, the information can be coded in certain form and gets broadcasted and decoded using the coding and decoding mechanisms. However, the batch Id gets changed whenever the batch gets reformed. This increases the operating cost significantly. Many people made many researches regarding this problem and made many suggestions. Hence our paper deals with the concept of providing effective batch rekeying management.

II. RELATED CONCEPTS

For Batch keying Management, one of the ultimate essential concepts is to decrease the conveyance high up when the common batch key is restored among the contributors. Tree-based GKM, one classy of GKM, has accepted considerable attention from many analysis since the measure of transmission cost high up for batch rekeying is equal to the logarithmic of the batch size. In addition many analysis have found different techniques namely rational key hierarchy, one way function tree (OFT) and one way key derivation (OKD).The most notable technique is rational key hierarchy which was suggested by wallner et al. To exactly analyse the conveyance cost high up, researchers have tried to find out the conveyance cost of the path tree. Zhu clearly examined that the 4 degree LKH seems to be idealised.

Eventhough many concepts have emerged to find the exact conveyance cost but that result in incompetence and out of synchronisation problem. Hence, to decrease the above stated difficulties Li et al proposed the concept of Batch Rekeying (BR) technique. In batch rekeying,

the key or Id gets changed at a particular time interval whereas, in logical key hierarchy, it gets changed whenever the batch gets rescinded. But the difficulty is that the batch Rekeying does not give much protection to the path tree.

III. PROBLEMATIC ANALYSIS

Although many analysis have made many decisions there are certain restrictions.

1. The contribution period members involved may be either short or long. They cannot have the similar relieving likelihood which seems to be not practical.
2. The framework of path tree already given seems to be idealised only for particular limitations of applicability. Though the tree can be made idealised but only for limited applicability. Hence, by doing this the conveyance cost high up can be dropped significantly.
3. The Id gets changed, whenever the batch gets rescinded. So the idealisation cannot be achieved in real world.
4. It cannot be applied for huge number of batches as it will degrade the performance and only applicable to limited contributors.

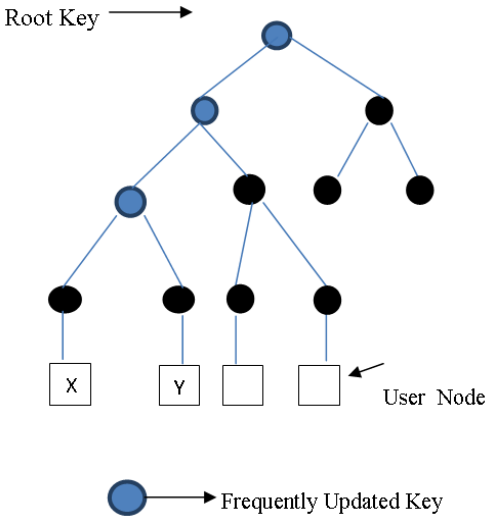


Figure 1: Example in previous works

Nevertheless, the expectation of a unique privilege has been approved as suited to common mobile environments. Since, no universal analysis methods for arbitrary relieving probabilities are available. Figure 1 is the illustration for validating the inaccuracy of this idea directly. The equal and whole design shown in Figure. l has been treated as the optimum design in the earlier works. However, it is no more perfect in most privilege where members have different relieving likelihood. Consider the privilege that two node X and Y is extra possible to relieve the batch, i.e., they have greater relieving likelihood than the remains. Three keys coloured with indigo in Figure. 1 will be updated more frequently.

Our Contributions

The important concept of this area is to generate a model for rescinding mobile contributors. Also involves less conveyance cost management and competency which will be useful to increase the performance. The contradiction between the proposed and existing will be provided as follows.

We introduced a general mathematical approach that provides the specific median value of the conveyance cost high up under certain restrictions. The conditions may be that it can have a huge count of customer, no need of same relieving probabilities or likelihood, and no need of complete or balanced tree. The median value of the rekeying information can be measured even for different periods of mobile contributors. In addition, we have also included the necessary conditions to form the tree to be idealised and also to decreases the conveyance cost high up.

We have also suggested two techniques namely, initial development of path tree and idealisation of path tree. The first technique involves the design of the framework and the next one is meant to keep the tree idealised even after the batch gets rescinded. It also generates a key tree that tends to keep the tree idealised and in turn drops the conveyance cost.

IV. MODEL OF FRAMEWORK

This structure contains a batch controller, server and batch of particular mobile. The batch controller tends to carry all the key types of the path and the server maintains the batch information along with the coded format involving the key or id.

The traffic encryption key (TEK) is used between the server and the contributors (users). It gets renewed whenever the batch gets rescinded meant for protection.
Consider M be the key structure, Um be the contributors node, Km be the key node .The traffic encryption key will be at the root of the structure. Individual key will be placed between particular contributors and the server. The KEK will be placed in the remaining nodes.

When the batch gets rescinded the traffic encryption key and KEK should get upgraded. The information that gets upgraded is called rekeying message and the period in which it gets upgraded called as the rekeying intervals.

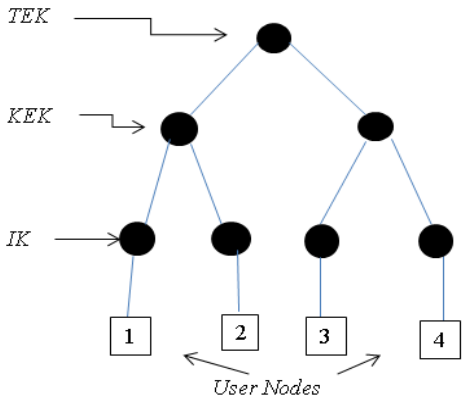


Figure 2: Example of LKH for four users

Where:
TEK-Traffic Encryption Key
KEK-Key Encryption Key
IK-Individual Key

Relieving probability

Consider U be the arbitrary contributor of the batch. Consider that the rekeying period is constant. Then U's relieving likelihood is P (U) is the likelihood that relieve the batch in the rekey timeslot.

$$P(U) = \frac{\text{The member of user relieving the batch}}{\text{The number of user in the batch where u contributes}}$$

There available two algorithms or techniques involved namely initial construction of path tree and idealisation of the path tree when the batch gets rescinded. Using these algorithms the conveyance cost high up will get minimised significantly.

V. PROPOSED ALGORITHMS

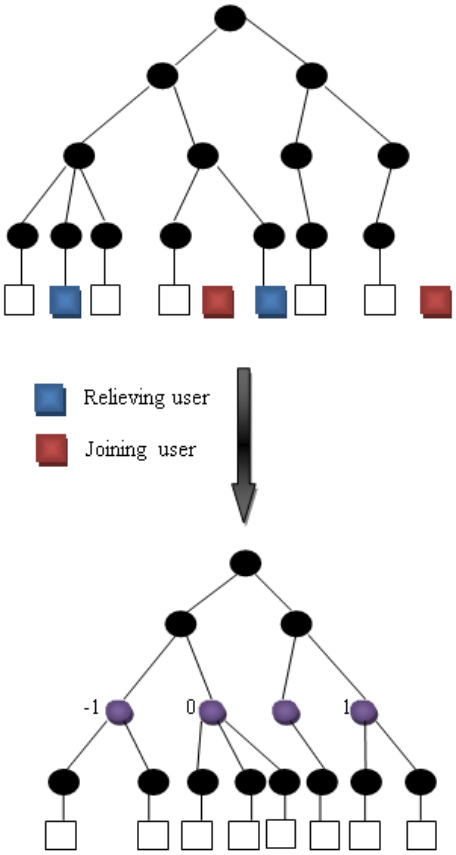
A. ALGORITHM FOR INITIAL TREE CONSTRUCTION

Inputs: User $u1, u2, \cdots, uN$, where $p(u1) \leq p(u2) \leq \cdots \leq p(uN)$

Output: Key-tree T
1: Allocate TEK in T;
2: $KT(TEK) \leftarrow \{k1, k2, \cdots, kN\}$;
3: if *Create Child* (T, TEK) = true then
4: for all $k \in KT(TEK)$ do
5: *makeTree* (*k*);
6: end for
7: end if
Function *makeTree*: key-node *k*
8: if $[cT(k) - 1] \cdot QT(k) + 1 > RT(k)$ then
9: if *Split Key*(T, *k*) = true then
10: for all $kchild \in KT(k)$ do
11: *makeTree* (*kchild*);
12: end for
13: end if
14: else
15: if *Create Child* (T, *k*) = true then
16: for all $kchild \in KT(k)$ do
17: *makeTree* (*kchild*);
18: end for
19: end if
20: end if

This section deals with the construction of path tree in which it could be formed by contributor's information. The input is the particular contributor information and the path tree will be formed from the particular information given and also placed based on the increased order of the probabilities.

The output will be the path tree (K (T)).This algorithm involves two recursive functions namely split key or create child. It will create the tree, whose height is 1, and the root node is traffic encryption key and the top most nodes are contributor's node or user nodes.



If there is no key encryption key, then there will the process of creating the child nodes will be processed. T is should have the lowest value of M (t) where M (t) is the median of rekeying information. If the repeated function returns true, then the path tree become competent. If the function returns false, then the repeated function stops.

B. ALGORITHM FOR IDEALISATION OF THE PATH TREE

Input: key-tree T, Joining Users $u1j, u2j, \cdots, u\lambda j$, Relieving Users $u1l, u2l, \cdots, u\mu l$

Output: Modified key-tree T_
1: Set the joining users in and remove the relieving users from T in order of the users' relieving probabilities;
2: for δ = depth of T − 1 to 1 do
3: for all KEK *k* in depth δ of T do

4: if *c*T (*k*) increases then
5: Mark *k* as 1;
6: if Splitting *k* is more idealized than creating *k*'s new children then7: *SplitKey*(T,*k*);
8: else
9: *CreateChild*(T,*k*);
10: end if
11: else if *c*T(*k*) decreases then
12: Mark *k* as −1;
13: if *c*T(*kp*) >2 then
14: *MergeKey*(T,*kp*);
15: else
16: *DeallChild*(T,*kp*);
17: end if
18: else
19: Mark *k* as 0;
20: end if
21: end for
22: end for
23: Update the relieving amount of the relieving users;

The idealisation of the path tree means that maintaining the tree even after the batch gets rescinded and hence by minimising the conveyance cost. The proposed technique involves two parts namely removing and joining contributors.
The first part involves that the algorithm tends to remove the users those are revocated and add the contributors those who want to join the batch.

The likelihood of relieving contributors will be denoted as -1, for added contributors it will be denoted as 1 and for both relieving and added contributors it will be denoted as 0. The node tree denoted as -1 indicates that the child nodes get decreased in the path tree and those denoted as 1 indicates that the child nodes gets increased in the node tree.

The next part involves again two parts namely processing of the user and marking the parent user. If the denoted node is 1 means that either the forming or separating of the nodes will be processed. If it returns true either one of the two function will gets executed otherwise the algorithm stops.

If the denoted node is -1 then either relocating or joining of the node tree will be processed. If the function returns true then either of the two functions gets executed otherwise the algorithm stops. If the denoted node is 0 then the either four basic functions will gets executed. Either separating or forming and either joining or relocating functions will get executed. If the separating or forming node is executed it will be denoted as 1 and if either the joining or relocating will get executed it will be denoted as -1.

VI. CONCLUSION

This paper reveals that the model we have proposed does not include any special restrictions such as same relieving probabilities, specified number of contributors. We are just calculating the conveyance cost or value based on two techniques named initial development of path tree and idealisation of that path tree once the batch gets rescinded. The first technique generates a path tree framework in the way the tree remains idealised and the next technique maintains the tree to be idealised. In this manner the competency gets improved. For multichannel usage of communication, it seems that the particular Batch Key Management seems to be ideal and also the competency gets improved.

VII. REFERENCES

[1]. M. Park, Y. Park, H. Jeong, and S. Seo, "Secure Multiple Multicast Services in Wireless Networks," IEEE Trans. Mobile Computing, vol. 12, no. 9, pp. 1712-1723, Sept. 2013.

[2]. M.-H. Park, Y.-H.Park, and S.-W. Seo, "A Cell-Based Decentralized Key Management Scheme for Secure Multicast in Mobile Cellular Networks," Proc. IEEE 71st Vehicular Technology Conf. (VTC '10-Spring), pp. 1-6, May 2010.

[3]. D.-H. Je, J.-S.Lee, Y. Park, and S.-W.Seo, "Computation-and-Storage-Efficient Key Tree Management Protocol for Secure Multicast Communications," Computer Comm., vol. 33, no. 2, pp. 136-148, Feb. 2010.

[4]. J.S. Lee, J.H. Son, Y.H. Park, and S.W. Seo, "Optimal Level-Homogeneous Tree Structure for Logical Key Hierarchy," Proc. Third Int'l Conf. Comm. Systems Software and Middleware and Workshops (COMSWARE '08), pp. 677-681, Jan. 2008.

[5]. W.H.D. Ng, M. Howarth, Z. Sun, and H. Cruickshank, "Dynamic Balanced Key Tree Management for Secure Multicast Communications," IEEE Trans. Computers, vol. 56, no. 5, pp. 590-605, May 2007.

[6]. V. Kondratieva and S.-W. Seo, "Optimized Hash Tree for Authentication in Sensor Networks," IEEEE Comm. Letters, vol. 11, no. 2, pp. 149-151, Feb. 2007

[7]. W.T. Zhu, "Optimizing the Tree Structure in Secure Multicast Key Management," IEEE Comm. Letters, vol. 9, no. 5, pp. 477-479, May 2005.

[8]. J. S., S. Suri, and G. Varghese, "A Lower Bound for Multicast Key Distribution," Computer Networks, vol. 47, no. 3, pp. 429-441, Feb. 2005.

[9]. Y. Sun, W. Trappe, and K.J.R. Liu, "A Scalable Multicast Key Management Scheme for Heterogeneous Wireless Networks," IEEE/ ACM Trans. Networking, vol. 12, no. 4, pp. 653-666, Aug. 2004.

[10]. A.T. Sherman and D.A. McGrew, "Key Establishment in Large Dynamic Groups Using One-Way Function Trees," IEEE Trans. Software Eng., vol. 29, no. 5, pp. 444-458, May 2003.

[11]. S. Rafaeli and D. Hutchison, "A Survey of Key Management for Secure Group Communication," ACM Computing Surveys, vol. 35, no. 3, pp. 309-329, Sept. 2003.

[12]. W. Trappe, J. Song, R. Poovendran, and K.J.R. Liu, "Key Distributionfor Secure Multimedia Multicasts via Data Embedding," Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP '01), pp. 1449-1452, 2001.