SECURED TREE BASED KEY MANAGEMENT IN WIRELESS BROADCAST SERVICES

K.Vijaya Babu

Department of IT, UCEV JNTUK ,
Vizianagaram, A.P, India
vjb059@gmail.com

O. Srinivasa Rao

Department of CSE, UCEV JNTUK ,
Vizianagaram, A.P, India
osr_phd@yahoo.com

Dr. MHM Krishna Prasad

Department of IT, UCEV JNTUK ,
Vizianagaram, A.P, India
krishnaprasad.mhm@gmail.com

Abstract :
Wireless broadcast is an effective approach for spreading data widely to a number of users. To provide secure transmission, symmetric-key-based encryption is one of the widely used method, that ensures only valid users can decrypt the data. With regard to various subscriptions, an efficient key management for distributing and changing keys is needed to control broadcast services. In this paper, we propose an efficient key management scheme, namely, Secured Tree Based Key Management (STBKM), to handle key distribution with regard to complex subscription options and user activities. STBKM needs to hold one set of keys for all subscription activities in wireless broadcast services. In this paper shows experimental results and concludes that, STBKM can reduce 45 percent of communication overhead in the broadcast channel and 50 percent of decryption cost for each user compared with logical-key-hierarchy-based approaches.

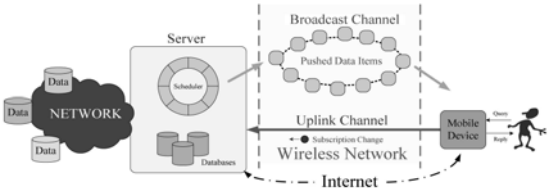
Keywords: Wireless networks; broadcasting; key management.

1. Introduction

Wireless broadcasting [1] is one of the active research area in network communication. Wireless broadcast is an effective approach for disseminating data to a number of users. By rapid advances in wireless technology, there has been an increasing interest in wireless data services among both industrial and academic communities in recent years. Different approaches are developed, allows a very efficient usage of the scarce wireless bandwidth for number of mobile clients.

Wireless data broadcasting mainly consists of 3 components as shown in Fig-1; 1) the broadcast server 2) the mobile user 3) the communication mechanism. The server schedules broadcasting all data into air. The mobile device listen broadcast channel retrieves filtered data according to the user input query. The communication mechanism includes wireless broadcast channels and uplink channels. Broadcast channel broadcast data periodically so that users can recover lost or missed data items. The uplink channels have limited bandwidth, are reserved to change subscriptions dynamically.

In broadcast services, the data items such as a piece of news or a stock price are grouped into programs. By sending specifications to server, user decides which programs he want to access. Typical programs could be weather, news, stock quotes, etc. A user may subscribe via the Internet or uplink to one or more programs that they are interested in receiving. The set of subscribed programs is called user's subscription. Wireless broadcast



services mainly focused on performance issues such as reducing data access latency and battery life time. In this paper, we focused on critical security requirements of the broadcast service by ensuring backward and forward secrecy with respect to dynamic membership.

In the wireless broadcast environment, any user can monitor the broadcast channel and record the broadcast data. If the data is not encrypted, the content is open to the public, and anyone can access the data. User can obtain data beyond his subscription privilege. Hence, access control should be enforced via encrypting data in a proper way, so that user can access subscribed programs only.

Symmetric-key-based encryption is a natural choice for transmit and access data securely through network. Broadcast data encrypted, so that user who have valid key can decrypt the data. Each program has one unique key to encrypt the data items. If a user subscribes to multiple programs, it needs an encryption key for each program. When a user subscribes/ unsubscribe to a program, the encryption key needs to be changed to ensure that the user can only access data in his subscription period. By the literature survey, two category of techniques used to effectively manage the keys when a user joins/leaves/changes the service without compromising security and interrupting the operations of other users.

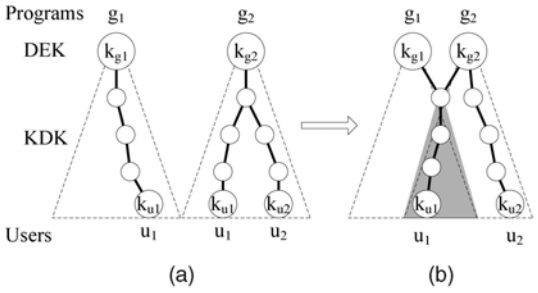
The two categories of key management schemes for broadcast service are; 1) logical key hierarchy (LKH)-based techniques [2][3][4][5] proposed for multicasting with respect to reliability and different solutions using Forward Error Correction (FEC) or retransmission techniques (ARQ) have been proposed aiming at reducing the probability of losses in a rekeying. However, in all proposed solutions, the LKH scheme still suffers from the "one affects all" scalability failure which occurs when the arrival or departure of any single member causes the update of at least one key with all members. Consequently, members who do not leave the group during an entire session can be strongly affected by frequent membership changes. This taxonomy is very general in nature and does not highlight the differences between applications with respect to security. 2) Broadcast encryption techniques [6][7][8][9], is an interesting application of cryptography which allows one to broadcast a

Fig. 1. wireless broadcast system.

secret to a changing group of intended recipients in such a way that no one outside this group can view the secret. Interest in using broadcast encryption techniques has grown considerably in recent years and such

techniques have been integrated in many applications and technologies such as virtual private networks, cable TV networks, mobile and wireless networks and many more.

2. Related work



2.1. *Shared Key Structure*

We use a shared key structure to address the key management. In the following, we describe how a shared key structure is applied and then raise the security and efficiency problems of this scheme..

2.1.1. *Key Forest*

To address scalability and flexibility in key management. An intuitive solution is to use a key tree for each program. When user u1 subscribes two programs simultaneously, he needs to manage two sets of keys in both trees, which is not very efficient (Fig-2a). SKT is proposed to overcome the problem, by sharing the same sub key tree (Fig-2b), as represented by the gray triangle. Shared keys are modeled as a key forest, in which all keys form a directed acyclic graph (DAG). The top keys in the forest are DEKs of the programs. All other keys (KDKs) form trees. Users are placed in trees according to their subscriptions. A tree represents not only a unique subscription but also a group of users having this subscription. Since a subscription is a set of programs, the root of the subscription's tree is connected to the DEKs of the programs belonging to the subscription. As keys in a tree are shared by the programs, a user only needs to handle the keys in the tree and the DEKs of the connected programs.

2.1.2. *Root Graph*

The root graph in Fig-2 depicts how programs share keys. Since m programs could generate 2m-1 different

Fig-2: SKT. (a) No share. (b) Share..

subscription's. This is a two-layer structure will brings two major problems in terms of rekey overheads when the number of programs is large.

First, a program may be included in many subscriptions, which means that the DEK of the program is connected with many trees. Assume that the DEK is connected with n trees. When a user stops subscribing the program, the DEK needs to be updated and distributed to users in n trees. If n is large, a leave event results in a huge rekey message. We can overcome this problem by using a multilayer structure to connect the DEK with the roots of the shared trees. As in Fig-2b, kg1 is connected (bold lines) with kr1, kr2, kr3 and kr5 via two intermediate key nodes (gray circles).
Second, a subscription is not a conventional plan that a broadcast service provides, because the subscribed programs of a plan normally cannot be changed by a user. In this paper we customize the subscription of user program.

2.2. *Identifying Critical Key's*

2.2.1. *Rekey Spots*

STBKM basically logs how a key was used in rekey messages; mainly have two operations: 1) A key's value is changed 2) a key is used to encrypt its parent key when the parent key's value is changed.

When a user changes his subscription, the server needs to change certain keys according to the algorithm presented in Algorithm 3 and to broadcast corresponding rekey messages. Then, refresh and renew spots are logged using Algorithm 1 based upon user event. The sequence of refresh and renew spots thus forms spot series in the time order.

Algorithm 1: update of refresh and renew spots.
Assume that ki is used in the rekey messages upon a user event.

```
    if ki is in a leave path then
    renew spots must be added to all ki's spot series;
  end if
  if ki is critical in an enroll path then
    renew spots must be added to all ki's spot series;
  end if
```

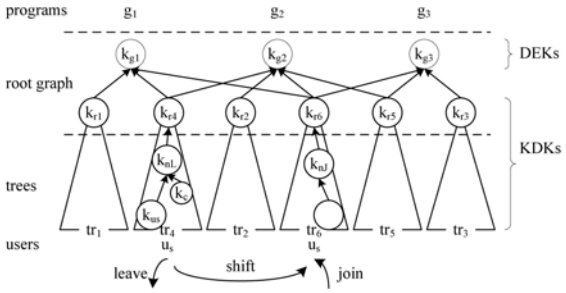


Fig- 3: Key forest.

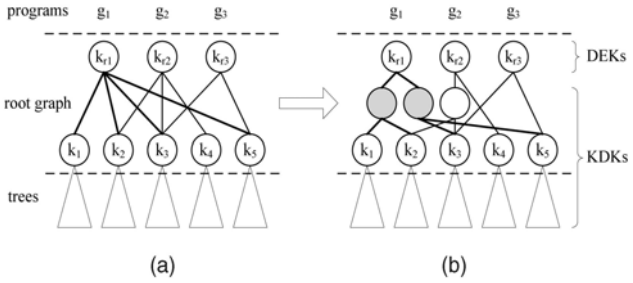


Fig-4: Multilayer root graph

if ki's parent key kj is in a leave path then
refresh spots must be added to ki's spot series that
are associated with the programs sharing kj;
end if

By logging key spots, the server can inspect a key's past usage. Using past information server calculates key age and subscription age to decide whether a key is a critical key. The server decides age of a KDK is 0 if and only if there is no revive spot between the current time and the latest renew spot. Otherwise, the age of the key is greater than 0. The subscription's age is 0 if and only if the user is not in the program. Otherwise, if the user is in the program, his subscription age is greater than 0. If a user stops subscribing a program, the subscription age associated with that program turns to 0.

2.3. *Identifying Critical Key's*

As per rekeying server maintain past confidentiality, age and subscription age of critical key. Hence, given a key forest, Algorithm 2 is applied to find the best enroll path to minimize the rekey cost. When a join or shift event happens to a tree, the algorithm uses the depth-first tree traversal approach to find all critical keys in the tree. If a path is found to have fewer critical keys than previously visited paths, the algorithm records it as the best enroll path.

Algorithm 2: algorithm of KTR in a broadcast server.
if a join or shift event happens then
according to TCK, find all critical keys in the tree that the user wants to join or shift to;
select the best enroll path that has the minimum number of critical keys;
change all critical keys in the best enroll path and broadcast corresponding rekey messages;
end if
if a leave or shift event happens then
change all keys in the leave path and broadcast corresponding rekey messages;
end if
update renew, refresh, and revive spots according to
the latest rekey messages;

3. Implementation

In this paper, we implemented Secured Tree Based Key Management (STBKM) algorithm and compared with logical-key-hierarchy-based approaches. Both algorithms are implemented and compiled with the jdk1.6; communications between server and client developed using Remote Method Invocation (RMI).

We analyze and evaluate the performance of algorithm by performing simulations on both server side and client side respectively.

Experiments are performed on 2-GHz CPU and a 2-GB RAM running on windows. Table 2 shows server side simulation results for two services with 5,000 users(n). The first row shows small service with 5 programs (m) and 15 valid trees (e), and each tree shares (d) 1.5 programs on the average. The second row shows large service with 50 programs and 300 valid trees, and each tree is shares 12 programs on the average.

Table 1. Server side computations.

Service Type	parameters n m e d	inspection	Update	computation(ms)
small	5K 5 15 1.5	320	20K	10 - 12
large	5K 50 100 4	50	80K	28 - 30

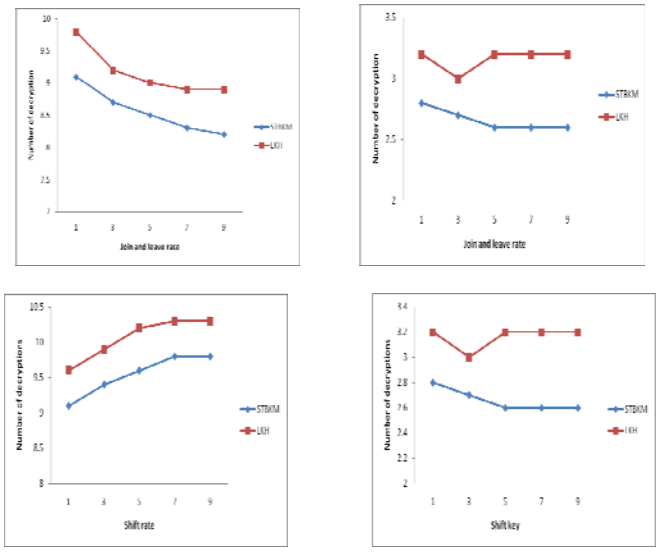


Table 1. Cases of Key Management.

Case	Major Subscriptions	Major events
Case 1	Multiple	Join and leave
Case 2	Single	Join and leave
Case 3	Multiple	Shift
Case 4	Single	Shift

In client side simulation 4 test cases are generated based on subscription and user events (i.e., join, shift, and leave) shown in Table 2. In cases 1 and 2, the major events are joins and leaves, while in Cases 3 and 4, the major events are shifts. The simulation results for average rekey message size & decryption per user event showed in below Fig: 5 & Fig: 6

If tables need to extend over to a second page, the continuation of the table should be preceded by a caption, e.g., "*Table 1* (*Continued*)". Notes to tables are placed below the final row of the table and should be flushleft. Footnotes in tables should be indicated by superscript lowercase letters and placed beneath the table.

From the above results, we observe that the Secured Tree Based Key Management (STBKM) is more flexible than logical-key-hierarchy (LKH) approach. And also observe that STBKM has light communication overhead, less computation and power & storage when compared with LKH.

4. Conclusion

Fig. 6. Average number of decryption per event per user (a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4

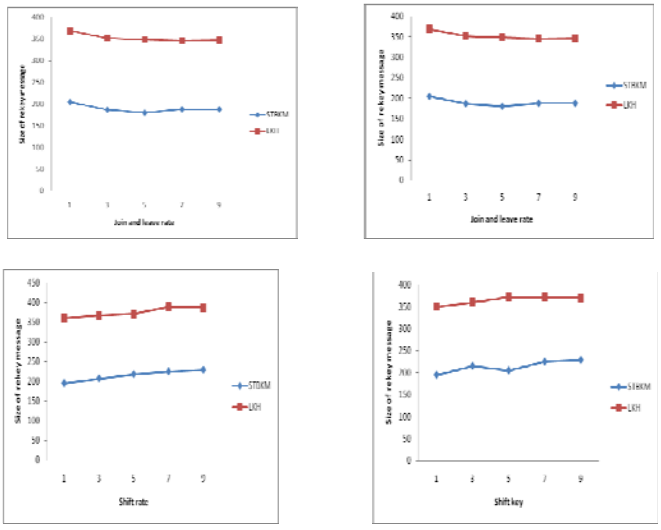


Fig. 5. Average rekey message size per event. (a) Case 1. (b) Case 2. (c) Case 3. (d) Case 4

In this paper, we propose an efficient key management scheme, namely, Secured Tree Based Key Management (STBKM), to handle key distribution with regard to complex subscription options and user activities. STBKM holds one set of keys for all subscription activities in wireless broadcast services. In this paper by the experimental results, we concludes that, STBKM is flexible and reduce 45 percent of communication overhead in the broadcast channel and 50 percent of decryption cost for each user compared with logical-key-hierarchy-based approaches.

References

[1] J. Xu, D. Lee, Q. Hu, and W.-C. Lee, "Data Broadcast," Handbook of Wireless Networks and Mobile Computing, I. Stojmenovic, ed., John Wiley & Sons,
[2] D. Wallner, E. Harder, and R. Agee, Key Management for Multicast: Issues and Architectures.
[3] J. Snoeyink, S. Suri, and G. Varghese, "A Lower Bound forMulticast Key Distribution," Proc. IEEE INFOCOM '01, vol. 1,.
[4] S. Mittra, "Iolus: A Framework for Scalable Secure Multicasting,"Proc. ACM SIGCOMM '97,
[5] S. Setia, S. Koussih, S. Jajodia, and E. Harder, "Kronos: A Scalable Group Rekeying Approach for Secure Multicast," Proc. IEEE Symp. Security and Privacy.
[6] B. Briscoe, "Marks: Zero Side Effect Multicast Key Management Using Arbitrarily Revealed Key Sequences," Proc. First Int'l Workshop Networked Group Comm., pp. 301-320, 1999.
[7] Wool, "Key Management for Encrypted Broadcast," ACMTrans. Information and System Security, vol. 3, no. 2, pp. 107-134, 2000.
[8] M. Just, E. Kranakis, D. Krizanc, and P.V. Oorschot, "On Key Distribution via True Broadcasting," Proc. Second ACM Conf.
[9] Computer and Comm.security.
[10] M. Luby and J. Staddon, "Combinatorial Bounds for Broadcast Encryption," Advances in Cryptology—Proc. Int'l Conf. Theory and Application of Cryptographic .