Research Paper

An Effective Key Management Approach in Multicast Networks

Vinay Sarwaiya Timothy¹, Mrs. Navdeep Kour Saluja² and Mr Sarvesh Rai³

¹ M.Tech, Scholar in Dept of Computer Science & Engineering, Infinity Management & Engineering College, Sagar MP, India

² Head of Department of Computer Science & Engineering, Infinity Management & Engineering College, Sagar MP, India

³ Asst Prof, Department of Computer Science & Engineering, Infinity Management & Engineering College, Sagar MP, India

ABSTRACT

The technique of multicasting is widely used for the purpose of communication from a number of people to communicate with a group of users. Some of the common application of this, are internet TV and on-line streaming. At present most of the applications are multicasting based. For example, package updates, multimedia system content distribution, interacting play and stock information distribution need multicast services. All the application being used need the authenticity and privacy of the participants. The problem that we face here in widespread or
worldwide use of safe and secure multicast communication is key management. Key management is the mechanism by which all members of a multicast group come to know of a secret key that is used in the encryption of data sent on the group. This paper presents an easy, efficient & secure-technique for key generation, key distribution/dissemination & a good storage for multicasting networks. The scope of this paper is to enhance the performance and functioning of hierarchical binary trees that is based on key distribution mechanism by distributing the work among a number of subgroup controller that dynamically share their load and balance the members such that every controller has as many members as it can safely accommodate and the whole tree is kept as much balanced as possible. This paper mainly concentrates on the performance analysis of total performance for better understanding of overhead and storage requirement which require for key management. The simulations for the scheme have been done by using MATLAB.

**Keywords:** Key Management, multicasting, Group controller, group key management, multicast, and session encryption key, Key encryption key.
INTRODUCTION

1.1 Motivation

The phenomenal growth of the Internet in the last few years and the increase of bandwidth in today’s networks have provided both inspiration and motivation for the development of new services, combining voice, video and text "over IP". Although unicast communications have been predominant up to now, the demand for multicast communications is increasing each from the net Service suppliers (ISPs) and from content or media suppliers and distributors. Indeed, multicasting is associated more) used as an economical communication mechanism for group-oriented applications within the net like video conferencing, interactive cluster games, video on demand (VoD), TV over net, e-learning, computer code updates, info replication and broadcasting stock quotes. yet, the dearth of security within the multicast communication model obstructs the effective and huge scale readying of such strategic business multi-party applications. This limitation impelled a bunch of analysis works that have addressed the various problems concerning securing the multicast, like confidentiality, authentication, watermarking and access management. These problems should be seen within the context of the protection policies that prevail at intervals the given circumstances [2]. For instance, in a public stock quotes broadcasting, while authentication is a fundamental requirement, confidentiality may not be. In the contrary case, each authentication and confidentiality area unit needed in video-conference applications. Multicast Security becomes
remarkably advanced and difficult to understand after we take into account different ingredients, such as mobility and fault tolerance. Mobility induces the problem of seamless and safe transfer of security context. Moreover, suitable mechanisms have to be developed to ensure access control for end-users whose access point may change during the multicast session. Additionally, since the failure of one security part might compromise safety and privacy of thousands of shoppers, hardiness and fault tolerance are basic and indispensable necessities. During this paper, we tend to specialize in 2 keystone components of any secure multicast architecture over wired networks: confidentiality and data origin authentication [3]. The distribution of data with commercial value or State top-secret content requires the use of appropriate mechanisms to prevent non-legitimate recipients from having access to the content. Besides, the recipient needs to ascertain the origin of the multicast data he receives and the content provider needs also to provide such assurance to protect himself from the dangerous consequences of being impersonated by a fraudulent third party. Even though a multitude of data origin authentication and confidentiality mechanisms currently exist, these security services stay a difficult drawback in terms of measurability, efficiency, and performance [4], [5].

1.2 Background

1.2.1 Introduction of Key Management

Key management for users within the communication networks relies upon the protection of the keys, it's typically applicable to plan a reasonably complicated mechanism to manage them. In cluster communication several people are concerned, with a demand for
distinctive keys to be sent to every for encryption/decryption of transmitted information. During this case, a variety of comprehensive and tried key management systems should be enforced [2],[3].

**1.2.1.1 Key Management Role**

Key management plays a vital role imposing access management on the cluster key(and consequently on the cluster communication). It supports the institution and maintenance of key relationships between valid parties in line with a security policy being implemented on the cluster [8]. It encompasses techniques and procedures that may carry out:

**1.2.1.2 Providing member identification and authentication.**

Authentication is very important so as to stop associate degree interloper from impersonating a legitimate cluster member. Additionally, it's vital to stop attackers from impersonating key managers. Thus, authentication mechanisms should be accustomed permit associate degree entity to verify whether or not another entity is admittedly what it claims to be.[8]

**1.2.1.3 Access management.**

After a celebration has been identified, its be a part of operation ought to be valid. Access management is performed so as to validate cluster members before giving them access to clustercommunication1 (the group key, in particular) [8].

**1.2.1.4 Generation, distribution and installation of key material**

it's necessary to vary the key at regular intervals to safeguard its secrecy [Schneider 1996].Additional care should be taken once selecting a replacement key to ensure key independence. Every key should be utterly freelance from any previous used and future
keys, different wise compromised keys could reveal other keys [8].

1.2.2 Distribution of Key Management
Several techniques are planned for the distribution of public keys. Virtually all these proposals is classified into the subsequent general schemes.

1.2.2.1 Public Announcement of Public Keys

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{uncontrolled_public_key_distribution}
\caption{Uncontrolled Public-Key Distribution}
\end{figure}

On the face of it, the purpose of public-key encoding is that the general public secret's public. Thus, if there's some loosely accepted public-key rule, like RSA, any participant will send his or her public key to the other participant or broadcast the key to the community at massive. As an example, attributable to the growing quality of PGP (pretty smart privacy, that makes use of RSA, several PGP users have adopted the follow of appending their public key to messages that they send to public forums, like USENET newsgroups and web mailing lists. Although this approach is convenient, it's a serious weakness. Anyone will forge such a public announcement. That is, some user might faux to be user A and send a public key to a different participant or broadcast such a public key. till such time as user A discovers the forgery and alerts alternative participants, the forger scanty read all encrypted messages supposed for A and may use the cast keys for authentication[9].

1.2.2.2 Publicly Available Directory

A larger degree of security are often achieved by maintaining a in public out there dynamic directory of public keys. Maintenance and distribution of the general public directory would
need to be the responsibility of some trustworthy entity or organization. Such a theme would come with the subsequent elements:

1. The authority maintains a directory with a entry for every participant.
2. Every participant registers a public key with the directory authority. Registration would need to be in the flesh or by some variety of secure genuine communication.
3. A participant could replace the present key with a replacement one at any time, either owing to the will to switch a public key that has already been used for an outsized quantity of information, or as a result of the corresponding personal key has been compromised in how.
4. Participants may additionally access the directory electronically. For this purpose, secure, genuine communication from the authority to the participant is necessary.

Figure 1.2 Public-Key Publication

This theme is clearly safer than individual public announcements however still has vulnerabilities. If AN antagonist succeeds in getting or computing the non-public key of the directory authority, the antagonist may magisterially pass out counterfeit public keys and after impersonate any participant and eaves-drop on messages sent to any participant. Differently to attain identical finish is for the antagonist to tamper with the records unbroken by the authority [9].

1.2.2.3 Public-Key Authority

Stronger security for public-key distribution is achieved by providing tighter management over
the distribution of public keys from the directory. A typical situation, that is predicated on [POPE79]. As before, the situation assumes that a central authority maintains a dynamic directory of public keys of all participants. Additionally, every participant dependably is aware of a public key for the authority, with solely the authority knowing the corresponding non-public key. The subsequent steps (matched by variety to occur);

1. A sends a time sealed message to the public-key authority containing a request for this public key of B.
2. The authority responds with a message that's encrypted victimization the authority’s non-public key. Thus, A is ready to decipher the message victimization the authority’s public key. Therefore is assured that the message originated with the authority. The message includes the following:

- B’s public key, that A will use to encode messages destined for B
- The first request wont to alter A to match this response with the corresponding earlier request and to verify that the first request wasn't altered before reception by the authority.
- The first timestamp given therefore A will verify that this is often not a recent message from the authority containing a key aside from B’s current public key
3. A stores B’s public key and conjointly uses it to encode a message to B containing an symbol of and a nowadays, that is employed to spot this dealing unambiguously.
4. 5. B retrieves A’s public key from the authority within the same manner as A retrieved B’s public key. At this time, public keys are firmly delivered to A and B, and that they could begin their protected exchange. However, 2 extra steps are desirable:
6. B sends a message to A encrypted with and containing A’s nowadays yet as a replacement nowadays generated by. As a result of solely B may have decrypted message (3), the presence of in message (6) assures A that the correspondent is B.

7. A returns, that is encrypted victimization B’s public key, to assure B that its correspondent may be a.

Figure 1.3 Public-Key Distribution Scenarios.

Thus, a complete of seven messages square measure needed. However, the initial four messages want be used solely sometimes as a result of each A and B will save the other’s public key for future use—a technique called caching. Sporadically, a user ought to request contemporary copies of the general public keys of its correspondents to make sure currency.

1.2.3.4 Public-Key Certificates

The situation of Figure is enticing, nevertheless it's some drawbacks. The public-key authority might be somewhat of a bottleneck within the system, for a user should attractiveness to the authority for a public key for each alternative user that it needs to contact. As before, the directory of names and public keys maintained by the authority is susceptible to meddling.

An alternative approach, 1st steered by Kohnfelder [Kohn78], is to use certificates which will be employed by participants to exchange keys while not contacting a public-key authority, during a method that's as reliable as if
the keys were obtained directly from a public-key authority. In essence, a certificate consists of a public key, Associate in Nursing symbol of the key owner, and also the whole block signed by a trustworthy third-party. Typically, the third party may be a certificate authority, like a authority or a institution, that's trustworthy by the user community. A user will gift his or her public key to the authority during a secure manner and acquire a certificate. The user will then publish the certificate. Anyone needing this user’s public key will acquire the certificate and verify that it's valid by method of the hooked up trustworthy signature. A participant may convey its key info to a different by transmission its certificate. Alternative participants will verify that the certificate was created by the authority. We will place the subsequent necessities on this scheme:

1. Any participant will scan a certificate to work out the name and public key of the certificate’s owner.
2. Any participant will verify that the certificate originated from the certificate authority and isn't counterfeit.
3. Solely the certificate authority will produce and update certificates. These necessities square measure glad by the initial proposal in [KOHN78]. Denning [DENN83] superimposed the subsequent further requirement:
4. Any participant will verify the currency of the certificate.
1.3 Overview of the Proposed Work

The proposed algorithm utilizes the concept of PN sequence generation in a linear feedback shift-register and exclusive OR-gate circuits. The concept is if we rotate the condition of switches the sequence will be change. For example let the connections are represented by a string of ones and zeros like “00100100” the sequence generated by this configuration will not match the sequence generated by the configuration “00010010” (1 bit rotated version). So instead of complete keys we can only send the rotations which reduce the data to be sent. From the above example it is clear that for 8 bit key only used to initially generate the 256 bits long keys at the nodes and then after only 3 bits required to update it. The keys are generated by know configuration shift register the manager need not to store all the keys instead only the configuration and the rotations are stored.

1.4 Research Objective

The main objective of our work is the proper key management for the multicasting supported networks is one of the most important considerations to maintain the security of the network while reducing the overhead and storage requirements. The multicasting is used for communicating with a group of users from large number of users the most common application of the multicasting internet TV and online streaming.

1.5 Proposed algorithm

The proposed algorithm utilizes the concept of PN sequence generation in a linear feedback shift-register and exclusive OR-gate circuits. The concept is if we rotate the condition of switches the sequence will be change. For example let the connections are represented by a string of ones
and zeros like “00100100” the sequence generated by this configuration will not match the sequence generated by the configuration “00010010” (1 bit rotated version). So instead of complete keys we can only send the rotations which reduce the data to be sent. From the above example it is clear that for 8 bit key only used to initially generate the 256 bits long keys at the nodes and then after only 3 bits required to update it. The keys are generated by know configuration shift register the manager need not to store all the keys instead only the configuration and the rotations are stored. The algorithm can be expressed in following steps:

1. Let the network needed a multicasting communication.
2. The root node calculates the number of users in group (let it be N).
3. The root node generates the configurations for N, PN sequence generator as KEK generator one for each users and store it.
4. Now the KEKs (PN sequence generator configuration) are transferred using private keys to group members.
5. Now a common SK for multicasting is encrypted through KEK and transmitted to each user.
6. If the change in group users is detected a random number is generated by root node (as rotation number).
7. This number is transferred to all the users in the updated group.
8. Now each user uses this rotation of update their PN sequence generator configuration for SK.

Experimental & Simulation Result

2.1 Performance Parameter

In this paper, a logical tree structure is constructed and the total number of users N is
divided into groups of size N. Each group is assigned to the leaf node of the logical tree. Logical key tree is used for inter-group key management and CRT scheme is used as intra-group key management. Since the group-controller alone cannot do the computation for ending the common solution X for each group, the work is given to an entity what is called as a GC [6].

Each cluster is assigned to individual SGC, so that the SGC will only compute the common solution X based on the session encryption key and the public key of the users within that cluster. After ending the common solution X for that cluster, the SGC multicast the solution to all the users within that cluster [7].

The key storage required for each user is very minimal because each user has to store its own public key and private key. Key storage required for each [6] GC is also very minimal because it has to store public keys of users within that cluster. In this scheme, the GC accepts the request from users for joining a multicast group. After accepting the request, the GC authenticates the user and then requests the user for its public key. The user in response sends the public key to the GC. The GC constructs the logical tree structure according to the maximum number of users that can be supported by the system [6].

Then the GC sends the initial KEK to the appropriate users according to the logical tree structure. After sending KEK, the GC sends the public key of the valid users.

Then the GC encrypts the SEK with the KEK and sends it to the SGC. The SGC after receiving the SEK from the GC computes the secure lock (i.e) the common solution X using the SEK and public key of each user.

Then it multicast the secure lock to all the users within the group. Each user within the group,
after receiving the secure lock, applies CRT and decrypts it using its own private key to get the SEK. The user uses the SEK to encrypt any outgoing message to the group members or to decrypt any incoming message from any of the group members. Whenever there is a change in the group membership, the GC generates new SEK and encrypts it and sends it to the user. Then the secure lock is recomputed by all GCs using the new SEK and public keys of the currently valid users in that group [6],[9].

2.1.1 Key Storage
The key storage requirement in different schemes has been compared. This is plotted in, the minimal key storage scheme has the lowest key storage complexity. The key storage in our scheme is only slightly higher than that of hybrid key tree structure which is the minimum overhead for the reduction in update message [6].

2.1.2 Key Storage in Group Controller
The relationship between the number of users and the number of keys stored in the GC. As the graph implies, there is no apparent increase in the number of keys stored as compared to the hybrid key tree based scheme [6].

2.1.3 Key Storage in User
The relationship between the number of users and the number of keys stored in each user. As the graph implies, there is there is a moderate increase in the number of keys stored in the users in the proposed scheme as compared to the group size.

2.1.4 SEK
Session encryption key (group key), used for communicating with all the users in the group (highest level node key)

2.1.5 KEK
Key encryption keys, also called as sub-group keys. They are used for encrypting the group key for its transfer (intermediate level node keys).

2.2 Simulation Result

Tables below shows the simulation and result obtained while simulation work is conducted on MATLAB.

In the simulation setup, the number of users is 50. SEK length is 128 bits, KEK is also 128 bits, similarly SEK and KEK configurations are 16 bits respectively. SEK rotations are 8 bits for simulation time of 100 seconds. We have taken the number of users in normal, lognormal and exponential pdf for mean 0 and variance as 1.

### Table 2.1 Scenario configuration

<table>
<thead>
<tr>
<th>Parameter used</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Number of Users</td>
<td>50</td>
</tr>
<tr>
<td>Total Simulation Time (sec.)</td>
<td>100</td>
</tr>
<tr>
<td>KEK length (bits)</td>
<td>128</td>
</tr>
<tr>
<td>SEK length (bits)</td>
<td>128</td>
</tr>
<tr>
<td>KEK Config Bits</td>
<td>64</td>
</tr>
<tr>
<td>SEK Config Bits</td>
<td>64</td>
</tr>
<tr>
<td>SEK Rotation Bits</td>
<td>8</td>
</tr>
<tr>
<td>Users Group conn./Disconn.</td>
<td>0.5</td>
</tr>
<tr>
<td>PDF Type</td>
<td>Gaussian</td>
</tr>
<tr>
<td>Mean</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter used</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variance</td>
<td>1</td>
</tr>
</tbody>
</table>

### Number of users Vs KEK overhead

Table 2.2 shows the relationship between the number of users and KEK overhead (bits). The Table implies, there is number of bits required by users in the proposed scheme as compared to the existing scheme size and overhead bits. The overhead produced by KEK bits is shown in the table below with proposed algorithm which shows minimum overhead instead of previous algorithm.

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Number Of User</th>
<th>Proposed Scheme Overhead (Bits)</th>
<th>Standard Scheme Overhead (Bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>700</td>
<td>6000</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>2.5*10^3</td>
<td>2.5*10^4</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>5*10^4</td>
<td>5*10^4</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>10*10^4</td>
<td>10*10^4</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>15*10^4</td>
<td>15*10^4</td>
</tr>
</tbody>
</table>

**Table 2.2 Number of users Vs KEK overhead**
It is observed that with the increasing number of users overhead bits are not increased rapidly as compared to the existing scheme. There is a reduction of approximately 7% in our scheme.

**Number of users Vs SEK overhead**

Table 2.3 shows the description of the number of users and SEK overhead (bits). During the simulation it is observed that overhead bits with the existing Standard scheme is much higher than the proposed scheme. The Table implies, the overhead produced by SEK bits is shown in the table below with proposed algorithm which shows minimum overhead instead of previous algorithm.

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Number Of User</th>
<th>Proposed Scheme Overhead(Bits)</th>
<th>Standard Scheme Overhead(Bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>700</td>
<td>6000</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>2.5*10^4</td>
<td>2.5*10^4</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>5*10^4</td>
<td>5*10^4</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>1.1*10^5</td>
<td>10*10^4</td>
</tr>
</tbody>
</table>

**Table 2.3 Number of users Vs SEK overhead**

It is observed that with the minimum number of users overhead bits size in the proposed scheme is almost eight times smaller than the existing standard scheme. It is also found that with the increasing number of users proposed scheme required much lower overhead bits as was too large in the existing standard scheme.

**GC Vs Number of User**

Table 2.4 shows the comparison of the number of users and the storage overhead bits produced by GC with the proposed scheme and existing standard scheme. During the simulation work it is observed that storage overhead bits with the existing Standard scheme is also much higher than the proposed scheme. The Table implies, the storage overhead produced by GC in both proposed and existing standard scheme which
shows minimum overhead storage of GC instead of previous algorithm.

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Number</th>
<th>Proposed Scheme</th>
<th>Standard Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>0</td>
<td>1400</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>0</td>
<td>2550</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>0</td>
<td>4000</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>0</td>
<td>5100</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>0</td>
<td>6400</td>
</tr>
</tbody>
</table>

Table 2.4 GC Vs Number of User

User Key Vs Number of user

Table 2.5 shows the comparison of the number of users and the storage overhead bits produced by user key with the proposed scheme and existing standard scheme. It is assume that the same key will not we sending again for a same user, it is also assume that all the user have already received the key there for the storage overhead(bits) size in proposed scheme is zero whereas it is increasing rapidly with the increasing number of users. This is the measure advantage with the proposed scheme which results in the form of smaller overhead storage size. The tables below shows the comparison of the number of user and user key with the above maintain assumption.

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Number</th>
<th>Proposed</th>
<th>Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>400</td>
<td>1400</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>800</td>
<td>2500</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>1200</td>
<td>4000</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>1600</td>
<td>5100</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>2000</td>
<td>6400</td>
</tr>
</tbody>
</table>

Table 2.5 User Key Vs Number of user

The proposed algorithm is simulated in MATLAB with different scenarios and the results are presented. The standard technique result is shown in the figure 2.1. Our results are plotted in the figure 2.1. pdf for two schemes are same.
In the figure 2.3, the overhead produced by KEK bits is shown in which proposed algorithm shows minimum overhead instead of previous algorithm. The number of users increases but overhead bits are not more. There is a reduction of about 7% in our scheme.

In the figure 2.4, the overhead produced by SEK bits is shown in which proposed algorithm shows minimum overhead instead of previous algorithm. The number of users increases but...
overhead bits are not more. there's a reduction of about 8% in our scheme.

**Figure 2.4:** Number of users Vs SEK overhead the graph shows that the proposed algorithm reduced the SEK overhead about 8 times.

In the figure 2.5, the storage overhead bits produced by GC is shown in which proposed algorithm shows minimum overhead instead of previous algorithm. the number of users increases but overhead bits are not more. there's a reduction of about 3% in our scheme.

**Figure 2.5:** Storage requirement for group key Vs number of users. The group key storage requirements reduces by 3 times.

In the figure 2.6, the user key storage produced by users is shown in which proposed algorithm shows constant instead of previous algorithm since the number of users increases but overhead bits are same for each as each user has same configuration bits.
CONCLUSION

In this Paper, we show the secure key management scheme for efficient multicasting in a group but limited to $2^8(256)$ nodes that has been used to achieve the optimum security of layer 2 control under the consideration of overhead metric. Among the three parameters tested, the over head found to be increases with increase in node size which is found to be the most significant followed by normal, lognormal, and exponential network size. It is important to note that, our results and conclusions are based upon the parameter levels used in this design and may vary, if different parameter levels are used.

REFERENCES


