Computer Systems Science & Engineering
DOI:10.32604/csse.2022.019561
Article

Tech Science Press

Decision Tree Based Key Management for Secure Group Communication

P. Parthasarathi[1,*] and S. Shankar[2]

[1]Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Sathyamangalam, 638401, India
[2]Department of Computer Science and Engineering, Hindusthan College of Engineering and Technology, Coimbatore, 641032, India
*Corresponding Author: P. Parthasarathi. Email: researchparthasarathi@gmail.com
Received: 17 April 2021; Accepted: 01 September 2021

Abstract: Group communication is widely used by most of the emerging network applications like telecommunication, video conferencing, simulation applications, distributed and other interactive systems. Secured group communication plays a vital role in case of providing the integrity, authenticity, confidentiality, and availability of the message delivered among the group members with respect to communicate securely between the inter group or else within the group. In secure group communications, the time cost associated with the key updating in the proceedings of the member join and departure is an important aspect of the quality of service, particularly in the large groups with highly active membership. Hence, the paper is aimed to achieve better cost and time efficiency through an improved DC multicast routing protocol which is used to expose the path between the nodes participating in the group communication. During this process, each node constructs an adaptive Ptolemy decision tree for the purpose of generating the contributory key. Each of the node is comprised of three keys which will be exchanged between the nodes for considering the group key for the purpose of secure and cost-efficient group communication. The rekeying process is performed when a member leaves or adds into the group. The performance metrics of novel approach is measured depending on the important factors such as computational and communicational cost, rekeying process and formation of the group. It is concluded from the study that the technique has reduced the computational and communicational cost of the secure group communication when compared to the other existing methods.

Keywords: Key generation; adaptive Ptolemy decision tree; cost reduction; secure group communication

1 Introduction

With the explosive growth of internet and the increasing power of computers, and communication networks, many new applications are emerging and many of them are based on group communications. Examples of group communications are interest groups, live and on-demand media, video conferences and online distributed games. Secure group communications which is provided by sheer confidentiality, authenticity and integrity of messages communicated between group members is a critical networking

issue. In group communications, the usage of efficient protocols such as multicast could reduce network congestion. The rapid progress in the technologies underlying multicast networking has led to the deployment of many multicast services, such as streaming stock quotes and multimedia services. When members of a multicast group need to receive the same information securely or allowed to join or leave the group dynamically, the security entails not only the distribution of secret among many, but may also be concerned with confidentiality of information as the membership changes. Group communication applications use Internet Protocol multicast for the transfer of data to all the group members using minimum resources. Efficiency is achieved because messages need to be transmitted once and they traverse any link between two nodes only once, hence saving bandwidth. This contrasts with unicast-based group communication where the sender has to transmit multiple copies of the same message from one point to another point. However, a scalable IP multicast does not provide mechanisms to limit the access control. The security challenge for a multicast is by providing an effective method for controlling access to the group and its message that is as efficient as the underlying multicast.

1.1 Secure Group Communication

Secure group communication is the method that provide privacy, genuineness and truthfulness of the messages communicated between the group members which is a critical network problem. In group communications, the usage of efficient protocols reduce the network congestion. The rapid development in the technologies of basic multicast networking has led to the exploitation of many multicast services, such as stream supply quotes and multimedia services. When the members of a group need to receive the same information securely or allowed to join or leave the group dynamically, the security involves not only the distribution of a secret among many, but may also be concerned with the privacy of information when the membership gets changed. Group communication applications use Internet Protocol to transmit data to all the group members using least amount of the resources. Efficiency is achieved only if the messages are transmitted once and they pass through any link between two nodes only once and saving the bandwidth. In contrast with the unicast-based group communication where the sender has to transmit multiple copies of the same message from one point to another point. The security challenge of the multicast is the process of providing an effective method for controlling access to the group and its key.

1.2 Group Key Management

In recent days, several applications or computers collectively communicate, providing shared access to the application, files etc. get misused in the internet. Secure multicasting is used in many of these applications. Privacy and authentication must be important in the process of the group key management. Access to the message by the unauthorized users can be restricted by the use of cryptographic encryption and selective distribution of the keys used to encrypt group information. Many secure group communications depend on the secret shared by the group members called as the group key. The encryption of the message using a group key which is known only to the intended recipients ensures its privacy. Although the encryption process will provide a privacy and information protection, a number of security risk associated with the integrity and secrecy of the encryption keys cannot be handled without the effective key management. Hence to ensure the right of entry control in dynamic multicast groups where members join and leave, the group keys need to be updated only then the key won't get misused by the unauthorized user. Otherwise, it poses a threat on forward and backward secrecy of the multicast communication.

Managing a set of secure group keys and group dynamics are the fundamental building blocks for the secure group communication systems. All the group members can make use of a shared group key to decrypt the communicated information. The Session Encryption Key (SEK) is established either by a server or a key that is more common. The schemes that involve a centralized key server are called as the centralized key

management schemes and the centralized key server is called as the Key Distribution Center (KDC). In such schemes, the KDC is the single entity that is employed for controlling the whole group. Hence a group key management protocol aims to minimize in terms of storage requirements, computational complexity on both the sides *i.e*., the client and the server and bandwidth utilization. The keys are created in a more secure and efficient way for a distribution of a group key.

The key management plays an important role and it chains the establishment and maintenance of key relationships between valid users as per the security policy enforced on the group. It includes methods and measures that can provide some provisions such as member identification and authentication, access control, generation, distribution and installation of key material. Key storage and key update are two important processes in the key management system. A key management scheme should tackle the overheads due to this process. A successful group-oriented multicast with right access manages the mechanism that can be attained by the appropriate update of the group key if there is a change in the membership. Rekeying is a mechanism that will change the affected keys. As the size of the group grows and/or the rate of membership change get increased, the frequency of rekeying becomes the primary bottleneck. Batch rekeying has been proposed to alleviate the problem of the frequent rekeying, where the key server waits for a period of time called rekeying period and then processes the rekeying procedure.

The use of periodic rekeying batch improves both the efficiency and the out-of-sync problem. Then the time efficiency of the causal key agreement is the important aspect. The time efficiency is calculated by the dispensation of time in group key establishment and update necessarily. In order to participate in the group communications, a joining new user has to wait in anticipation and the group key will need to be updated. Since computing cryptographic primitives and exchanging rekeying messages are still time-consuming processes, the waiting time is not minor in such cases. Similarly, the amount of time wanted to recompute a new group key reflects a tedious task on the other side. Thus, from a quality-of-service perspective, the rekeying time cost is directly related to the satisfaction of the users and the performance of the systems. Traditionally, the rekeying time difficulty is being analyzed only in terms of one join or leaving the event. Whereas, in this study, the perspective has been changed to look into the combination of multiple events and optimize the time cost over the membership. In order to improve the time efficiency, a new key tree topology with join and exit sub-trees are involved. With this key tree topology, an update is made on the member who joins and leaves. The rest of the session in the paper are organized as 2. Related Work, 3. Problem statement, 4. Proposed Methodology, 5. Member Join and member Leave 6. Results and 7. Conclusion.

2 Related Works

Appolini [1] presented about a recurrent neural network which meant to suggest the next move at the descent along with the branches of a decision tree. These values constituted the main for the chance of selecting the move left to the set of the values of the current node. It would result in a common way for driving the sharp discrete-state process running with the decision tree by means of the incremental methods on the parameters of continuous- values of the neural network. Boaron et al. [2] proposed about the Secure Quantum Key Distribution over 421 kms of Optical Fiber which revealed the method of secure key distribution. Bunander et al. [3] proposed about the quantum key distribution method for the secure key distribution. Zhao et al. [4] proposed about the problems of key distribution in ad-hoc networks which was presented as an important issue. Unfortunately, this problem has been largely ignored because most of the protocols for secure ad-hoc routing refers to the key distribution which has already been taken place. Yin et al. [5] proposed about the decoy-pulse method to overcome the photon-splitting attack quantum key distribution protocol in the presence of high loss of keys.

Sharma et al. [6] proposed a study on the Identity based secure authentication scheme pertinent to the quantum key distribution for cloud computing. Maa et al. [7] studied on the secure key distribution method

for the dynamic conferences. In the specified protocol, *any* member of a group of *the* users can compute a common key using only his/her private identification code initial piece of information and the *identities* of the other users in the common group. Mood et al. [8] proposed a study on the decoy-protocol quantum key distribution (QKD). The purposed key rates had achieved many more ideas to the use of self-differencing. Begam et al. [9] determined the adaptive and the current state of the art of research in quantum cryptography. In particular, it discussed the present security model together with its assumptions, strengths and weaknesses. After that, it briefly introduced the recent experimental process and its issues. The main significance was given to brief introduction of lkh and lkh+ trees. Acin et al. [10] proposed a new QKD protocol and proved its security against any of the individual attack by an adversary condition which was only limited by none of the signaling condition. Sherman et al. [11] studied about lkh and lkh+trees and their comparison on the basis of nodal mode. Thomas et al. [12] examined the rekeying method on lkh+ trees. Minder et al. [13] proposed a study on the secure quantum communications and its security basics. Alexander [14] researched on the serious issues of key distribution muddles.

Engle et al. [15] examined about Implementing the decoy state protocol in a practically oriented Quantum Key Distribution system-level model. Hodson et al. [16] proposed a research on the modeling of the quantum key method. Yin et al. [17] studied about the Measurement-Device-Independent Twin-Field Quantum Key Distribution. Zhu et al. [18] determined the decoy state of the quantum key distribution method. Mao et al. [19] proposed a study on the dynamic key distribution technique. Parthasarathi et al. [20] researched on Weighted Ternary Join Exit Tree approach for handling member join and leave in the secure Group Communication.

3 Problem Statement and Proposed Work

The first main drawback of the group communication is the costing of the key generation and the communication security. Keying relationships need to maintain the confidentiality throughout the process. When a member leaves or joins in a group, the rekeying process should be done. But the cost of the rekeying is also very high. So, there is a need of an efficient method to overcome the cost and the security in the group communication process.

3.1 Methodology

The main aim of this paper work is to design a robust cost-effective secure group communication by Ptolemy decision tree technique. In the study, the star topology was implemented and all the nodes were scattered around the central hub point. By the implementation of the star topology, the group communication could occur in a secure and an efficient way. Due to the employment of routing algorithm, each node could be assigned as a separate core active member. The active and the passive member in the group could be merged depending on the protocol. An efficient algorithm was proposed to construct a group key by using the adaptive Ptolemy decision tree and assigned an individual value for each node in the tree. In addition to that, the adaptive Ptolemy decision tree was constructed. The Fig. 1 represents the overall proposed methodology for the secure cost-effective group communication.

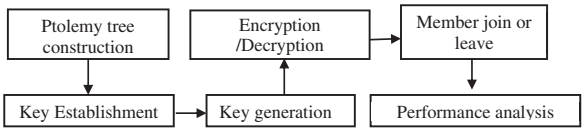

Figure 1: Schematic representation of the proposed methodology

3.2 Build an Adaptive Ptolemy Decision Tree

3.2.1 Nodes Creation

Star networks are one of the most common computer network topologies. In its simplest form, a star network is consisted of one central switch, hub or computer, which act as a medium for transmitting the messages. It is also consisted of a central node, to which all the other nodes are connected; this central node provides a common connection point for all nodes through a hub. In star topology, every node is connected to a central node called a hub or a switch. The switch is the server and the peripherals are the clients. Thus, the hub and leaf nodes, and the transmission lines between them, form a graph with the topology of a star Tab. 1.

Table 1: Decryptions for the creation of the nodes

Notations	Descriptions	Type
S	Identification of the node	Integer
ST	Type of the node	CAM,AM,PM
PATH	List of the path	Integer array
CK	Contributory key	Integer
a,b	Key pairs	Integer
GK	Group key	Integer
Pkey	Private key	Integer
Status	Node status	String
Max hop count	Maximum hop count	Integer
H count	Hop count	Integer

```
Pseudo code: 1 Assigning the nodes
//let n should be the number of the network and BL be the battery
Level inside
Percentage of the nodes in the network
//Let S is the array structure of the node
Type of the node
Start
    For j=1 to n
    Start
        If (S(j).BL<30) then
            S[j].ST=Path PM,
        Else if
            S[j].ST=AM
        Else
            S[j].ST=CAM;
    End
End
```

3.2.2 Establishment of the Group

In this paper, we assume that the nodes are arranged in the star topology and the nodes are classified into three different categories as Core Active Member, Active Member and a Passive Member using the Node

Type algorithm. The Fig. 2 represents the star topological grouping of the members. PM represents the passive member, ordinary circle is the active member, dotted circle is the core active member and NPM is the non- participating member, the steps for the process of building a Ptolemy decision tree using DCMP protocol.

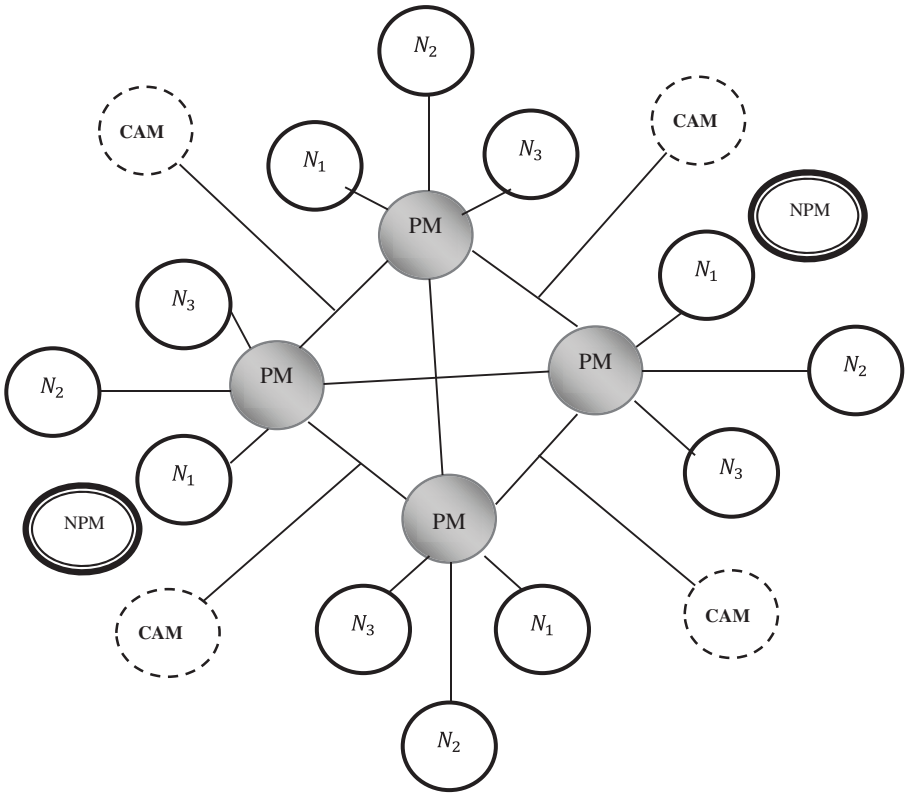

Figure 2: Star topological group alignment

By implementing the DCMP protocol, the joining of an average member will be represented as

$$T_{join} = \frac{R_{join}}{N_{join}} \tag{1}$$

Similarly, the leaving time of the user is represented by

$$T_{leave} = \frac{R_{leave}}{N_{leave}} \tag{2}$$

Let

$$\mathrm{N} = N_{join} + N_{leave} \tag{3}$$

$$\mathrm{R} = R_{join} + R_{leave} \tag{4}$$

The overall processing time is represented by,

$$\mathrm{T} = \frac{R}{N} \tag{5}$$

Then the weighted average of T_{join} *and* T_{leave} can be represented by,

$$\mathrm{T} = \frac{N_{join}}{N} T_{join} + \frac{N_{leave}}{N} T_{leave} \tag{6}$$

The join and the exit of the tree technique are of binary tree technique depending upon the DCMP protocols. whereas, in this study, the new user can be added to join the tree and if the tree reaches its maximum capability, all the users get shifted or relocated to the main tree, The fig. 3 represents the adaptive Ptolemy decision of join and exit tree.

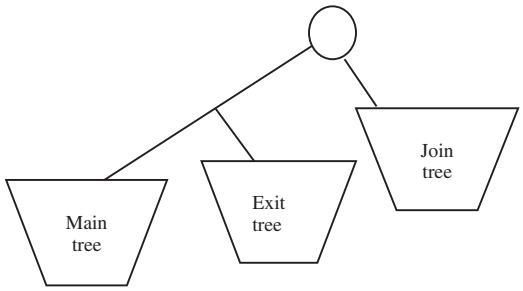

Figure 3: Adaptive Ptolemy decision join and exit tree

Moreover, the core active member and the active member will create a join request message and the message can be sent to all one hop neighbors in the network. While receiving the join request message, the core-active members forward the message to the hop neighbors After that, checking process will occur whether the member (node identifier Nid) is already present in the path list or not. The receiving nodes will set the status as ACK or NACK and send the reply message to the source node through the reverse path. If each node append the node id, the R path will start generating the message for the source Sid. The R-path will generate the group key using the Ptolemy decision tree algorithm and it can construct a Ptolemy decision tree.

3.2.3 Key Generation Process

1. Each of the individual node can generate the key.
2. Triplets of the nodes are passed to the Ptolemy tree to build a contributory key. In this Adaptive Ptolemy decision tree, the constants a, b, c, and d are considered for the study of Ptolemy theorem (AC.BD=AB.CD+BC.AD).
3. Values are assigned *S[j].pkey=a, S[j].pkey=b, S[j].pkey=c, and then, the value of D is returned to the key generation algorithm*.
4. Computation of each node is calculated as *S[j].pkey (ac. bd) = S[j].pkey(ab.cd+bc.ad)*
5. Finally, a triplet of key will get exchanged among the nodes.

```
Pseudo code: 2 Group formation
// For each node in the star topology, formation of group is done,
For j =1 to n
Start
    If S[j].ST=CAM or S[j].ST==AM then
        Join request creation (S Nid, PATH, H count, Max hop count ) ;
        All the individual hop neighbors,
        Forward join request (S Nid, PATH, H count, Max hop count) to Start
        H count= H count +1;
          If S[i] Nid not in the path then,
                Append path with S[I].Nid;
    End
    Else stop forwarding the
        Join req (S Nid, PATH, H count, Max hop count) mess;
    End
    Reply creation ACK (Nid, PATH,Status) message
     Forward in the reverse path list;
     For I =Sto 1
      Start
        If I is the last node in the path list,
        If S[i].status==ACK; then
            Store the S[i] in the reply path,
            Forward reply ACK to the S[i];
        Else
            Forward reply ACK to the S[i];
        End
  End
Group key generated
```

3.2.4 Encryption and Decryption

Blowfish is a symmetric block cipher which is used as a drop-in replacement for DES or IDEA. It takes a variable-length key from 32 bits to 448 bits by making it ideal for encryption and decryption purpose which is distinctly shown in Fig. 3.

Steps to Encrypt the Message:

1. Assign the value of alphabets as A = −1, B = −2, …, M = −13 and N = 13, O = 12, …, Z = 1.
2. Get the message for Encryption. Let the message be $W_1, W_2, \ldots, W_n$ where, n is the number of words in the message.
3. Use point 1, assign each character in $W_1, W_2, \ldots, W_n$ to digit, separated by spaces between the characters and the words.
4. Draw Cyclic Square Matrix with characters in W_i for each i = 1, 2, … n
5. Calculate the number of characters in a word, $\eta(W_i)$ for each i = 1, 2, …, n
6. Construct diagonal matrix, $D(A_m)$, m = 1, 2, …, i with A_m values along diagonals and find $D(A_m) - \eta(W_i)I_{\eta(Wi)}$ for all i = 1, 2, …, n and m = 1, 2, …, i
7. The key is $D(A_m) - \eta(W_i)\ I_{\eta(Wi))}$ for all i = 1, 2, …, n and m = 1, 2, …, i separated by commas.

```
Pseudo code: Group key generation
key generation (RPATH list)
Start
//For each node in the RPATH
//An array R can be stored (intermediate to the contributory key), I=1 and j=1
When the I value is greater than n
Start
        R[j]= Adaptive Ptolemy decision tree ( S(i), S[i+1]);
I=i+2; j=j+1;
End
//Find out the highest value contributory key
Start
If ( R(i), Ri+1]); then
CR=R[i+1]
Else
CR=R[i]
End
Assign the contributory key to each and every node in the group
For i=1 to n
Begin/Start
S[i].a=CR/S[i] .p.key;
S[i].b=CR/S[i] .p.key;
S[i].c=CR/S[i] .p.key;
S[i].d=CR mod S[i] .p.key;
End
    If node I wants to communicate the data to j, a triplet of keys as ac[i],bd[i],ab[i],cd[i],bc[i],ad[i] and
    ac[j],bd[j],ab[j],cd[j],bc[j],ad[j] are exchanged between I and j respectively.
Start
Compute the group key
End
End
```

Steps to Decrypt the message:

1. Get the decryption key $D(A_m) -\eta(W_i)\ I_{\eta(Wi)\ )}$ for all i =1, 2, …, n and m = 1, 2, …, i separated by commas.
2. Assign b and E. Compute the value k_i implying the first digit of B_i which is the k_i^{th} character of the i^{th} word of the decryption key.
3. Align C_i, i = 1, 2, … , n in the cyclic order along with the value k_i^{th} order and rephrase to the order from 1^{st} to i^{th} digits.
4. Use point 1 in the Encryption algorithm and assign digits to each character.
5. The decrypted value is obtained. The preliminaries that are chosen are associated in the order of each number to each alphabet as mentioned.

A	B	C	D	E	F	G	H	I	J	K	L	M
−1,	−2.	−3	−4*	−5−	−6+	−7@	−8’	−9&	−10.	−11..	−12…	−13−
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	12	11#	10	9	8	7	6	5	4	3	2	1

For the purpose of encryption and decryption, the coding of an apple is −1,11#11#−12…−5−

Pseudo code: 4 Blowfish

```
Blowfish_Init(&ctx, key, keylen);
printf(“Plaintext message string is: %s\n”, plaintext_string);
/* encrypt the plaintext message string */
printf(“Encrypted message string is:”);
while (plaintext_len)
{
message_left = message_right = 0UL;
/* crack the message string into a 64-bit block (ok, really two 32-bit blocks); pad with zeros if necessary */
or (block_len = 0; block_len < 4; block_len++)
{
message_left = message_left << 8;
if (plaintext_len)
{
message_left += *plaintext_string++; plaintext_len−;
}
else message_left += 0;
}
for (block_len = 0; block_len < 4; block_len++)
{
message_right = message_right << 8;
if (plaintext_len)
{
message_right += *plaintext_string++;
plaintext_len−;
}
else message_right += 0;
}
/* encrypt and print the results */
```

(continued)

Pseudo code: (continued)

```
Blowfish Encrypt(&ctx, &message_left, &message_right);
printf("%lx%lx", message_left, message_right);
/* save the results for decryption below */
*ciphertext_string++ = (uint8_t)(message_left >> 24);
*ciphertext_string++ = (uint8_t)(message_left >> 16);
*ciphertext_string++ = (uint8_t)(message_left >> 8);
*ciphertext_string++ = (uint8_t)message_left;
*ciphertext_string++ = (uint8_t)(message_right >> 24);
*ciphertext_string++ = (uint8_t)(message_right >> 16);
*ciphertext_string++ = (uint8_t)(message_right >> 8);
*ciphertext_string++ = (uint8_t)message_right;
ciphertext_len += 8;
printf("\n");
/* reverse the process */
printf("Decrypted message string is:");
  ciphertext_string = &ciphertext_buffer[0];
  while(ciphertext_len)
  {
  message_left = message_right = 0UL;
  for (block_len = 0; block_len < 4; block_len++)
  {
   message_left = message_left << 8;
   message_left += *ciphertext_string++;
   if (ciphertext_len)
   ciphertext_len−;
  }
  for (block_len = 0; block_len < 4; block_len++)
{
   message_right = message_right << 8;
  message_right += *ciphertext_string++;
   if (ciphertext_len)
   ciphertext_len–;
}
Blowfish Decrypt(&ctx, &message_left, &message_right);
   /* if plaintext message string padded, extra zeros here */
```

(continued)

Pseudo code: (continued)

```
  printf("%c%c%c%c%c%c%c%c",
    (int)(message_left >> 24), (int)(message_left >> 16),
    (int)(message_left >> 8), (int)(message_left),
    (int)(message_right >> 24), (int)(message_right >> 16),
    (int)(message_right >> 8), (int)(message_right));
}
printf("\n");
return 0;
}
```

4 Joining and Leaving of a Member

4.1 Join Tree Algorithm

- New node will join the group with a new id
- If newST==CAM or the AM,the new member can perform the group activity
- New node search for a hop neighbor or the core active member in the network,
- After joining, the new node can be communicated with any other node by exchanging the triplet of the keys
- The encryption or the decryption can be performed using the group key securely among the nodes.

4.2 Exit Tree Algorithm

It is mainly estimated that the duration of the staying time helps in the reduction of the cost in the method of rekeying operation during the process of leaving. Also, it consists of four parts; 1. Batch Movement, 2. User Insertion in the Exit Tree 3. Optimal Exit Tree Capacity and 4. Activation of Exit Tree

In the batch movement, the users will be generally moved from the main tree to the exit tree. This method does not affect the group key communication and after that, the insertion node should be selected in the exit tree for the maintenance of balance in the exit tree. Then, the parent node and the user node of the leaving user should be deleted. The average leaving time of the user has to be calculated using simple key tree. By comparing the results of the reduction in the average leave of the member, the time should be calculated.

- If the node wants to leave the group, they can send a leave request message to all the nodes in the group to get the acceptance from all the other nodal members in the group.
- Leave req (Nid,Path,Type,n)
- After sending a leave request, the other nodes in the group will send a reply (ACK) to the source node *i.e*., to the leaver node.
- Reply ACK(Nid,RPATH.Status)
- During the reply process, the reply message should be stored in the Nid, so that the information of the leaver should be removed easily.

5 Results

In the paper, the Adaptive Ptolemy decision tree technique was represented in a general way for the purpose of secure and cost-efficient key distribution. The performance of the Ptolemy decision tree was evaluated for improving the efficiency of the group communication. In order to prove the results, the comparison was made between the other existing methods.

5.1 Communication Cost

Whenever a member leaves or joins the group, there is a change in the database which is notified once for each change. Hence, only one message is appropriate for any change in the network *i.e.*, the log 1.5 = 0.176 is the communication cost for APDT.

Tab. 2 is the comparison of communication costs between the proposed APDT and the other existing techniques.

Table 2: Communication costs of the other existing techniques with the proposed APDT

Techniques	Join	Leave
OFT	log_2 n + 1	Log n + 1
LKH	$2log_2\ n - 1$	2 log n
CEGKMS	1	1
APDT(proposed)	Log 1.5	Log 1.5

The Fig. 4 shows the communication cost for the process of joining. Whenever a member leaves or joins, a single key is produced. When n number of members leave or join, the n number of keys get generated. Similarly, Fig. 5 shows the communication cost for the process of leaving.

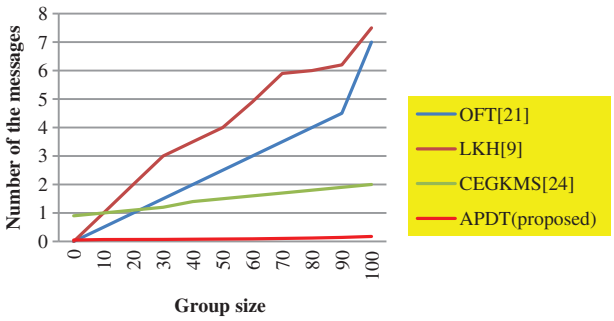

Figure 4: Communication cost at joining process

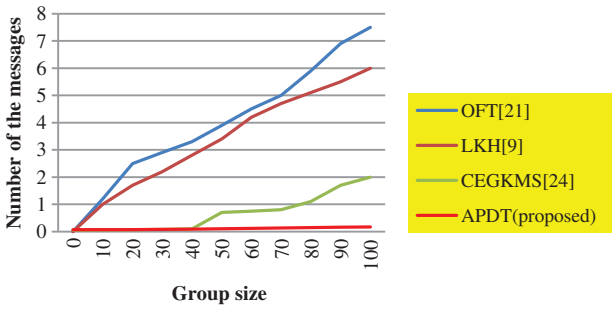

Figure 5: Communication cost at leaving process

5.2 Computation Cost

There is a rekeying process in the APDT where the group key remains the same for any change in the number of members ensuring the securities. If there is any change in the group, the rekeying process will take place.

The Fig. 6 represents the computation cost for the process of joining. Whenever a member leaves or joins, a single key is produced. When n number of members can leave or join, the n number of keys get generated Fig. 7. Tab. 3 shows computation cost in group key generations.

Table 3: Brief comparison of computation costs in group key generation

Techniques	Join	Leave
OFT	log_2 n + 1	Log n + 1
LKH	$2log_2\ n - 1$	2 log n
CEGKMS	1	1
APDT(proposed)	Log 1.5	Log 1.5

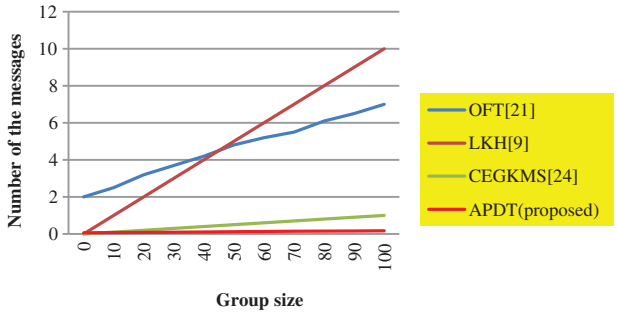

Figure 6: Computation cost at the joining process

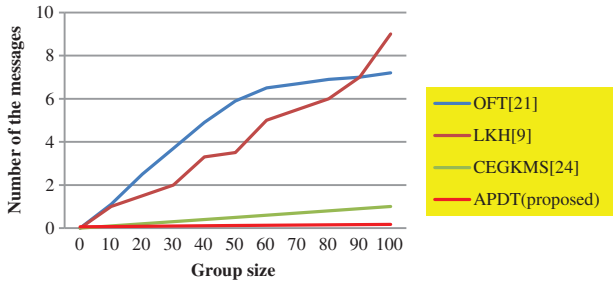

Figure 7: Computation cost at the leaving process

6 Conclusion

The APDT technique is a suitable choice in reducing the key costing problem whenever a user joins or leaves the group. From the interpretations of the research, it was very clear that the results were proved to be better when compared to the other existing methods. In addition to that, the cost-efficiency of the technique was the main advantage which was observed from the group communication processes. Thus, from the analysis and the comparisons of the study, APDT is proved to be the efficient key management scheme and provided well organized security for the data communications.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] B. Apolloni, G. Zamponi and A. M. Zanaboni, "Learning fuzzy decision trees," *Neural Networks*, vol. 11, no. 5, pp. 885–895, 1998.

[2] A. Boaron, G. Boso, D. Rusca, C. Vulliez, C. Autebert *et al.,* "Secure quantum key distribution over 421 km of Optical Fiber," *Physical Review Letters*, vol. 121, no. 19, pp. 1–5, 2018.

[3] D. Bunandar, A. Lentine, C. Lee, H. Cai, C. Long *et al.,* "Metropolitan quantum key distribution with silicon photonics," *Physical Review*, vol. 8, no. 2, pp. 0210091–021009-12, 2018.

[4] Y. Zhao, Y. Cao, W. Wang, H. Wang, X. Yu *et al.,* "Resource allocation in optical networks secured by quantum key distribution," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 130–137, 2018.

[5] Z. Q. Yin, S. Wang, W. Chen, Y. G. Han, R. Wang *et al.,* "Improved security bound for the round-robin-differential-phase-shift quantum key distribution," *Nature Communications*, vol. 9, no. 1, pp. 661, 2018.

[6] G. Sharma and S. Kalra, "Identity based secure authentication scheme based on quantum key distribution for cloud computing," *Peer-to-Peer Networking and Applications*, vol. 11, no. 2, pp. 220–234, 2018.

[7] X. Ma, P. Zeng and H. Zhou, "Phase-matching quantum key distribution," *Physical Review*, vol. 8, no. 3, pp. 031043-1–031043-26, 2018.

[8] D. Abbasinezhad-Mood and M. Nikooghadam, "An anonymous ECC-based self-certified key distribution scheme for the smart grid," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 7996–8004, 2018.

[9] S. J. Begum and T. Purusothaman, "Hierarchical tree structure-based clustering schemes for secure group communication," *Mobile Networks and Applications*, vol. 21, no. 3, pp. 550–560, 2016.

[10] A. Acín, N. Gisin and L. Masanes, "From bell's theorem to secure quantum key distribution," *Physical Review Letters*, vol. 97, no. 12, pp. 1–5, 2019.

[11] A. T. Sherman and D. A. McGrew, "Key establishment in large dynamic groups using one-way function trees," *IEEE Transactions on Software Engineering*, vol. 29, no. 5, pp. 444–458, 2003.

[12] L. R. Dondeti and T. Hardjono, *Multicast and group security.* Norwood, MA: Artech House, 2003.

[13] M. Minder, M. Pittaluga, G. L. Roberts, M. Lucamarini, J. F. Dynes *et al.,* "Experimental quantum key distribution beyond the repeaterless secret key capacity," *Nature Photonics*, vol. 13, no. 5, pp. 334–338, 2019.

[14] Q. Li, X. Wen, H. Mao and X. Wen, "An improved multidimensional reconciliation algorithm for continuous-variable quantum key distribution," *Quantum Information Processing*, vol. 18, no. 1, pp. 3121, 2019.

[15] R. D. Engle, L. O. Mailloux, M. R. Grimaila, D. D. Hodson, C. V. McLaughlin *et al.,* "Implementing the decoy state protocol in a practically oriented Quantum Key distribution system-level model," *Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 16, no. 1, pp. 27–44, 2019.

[16] D. D. Hodson, M. R. Grimaila, L. O. Mailloux, C. V. McLaughlin and G. Baumgartner, "Modeling quantum optics for quantum key distribution system simulation," *Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 16, no. 1, pp. 15–26, 2019.

[17] H. L. Yin and Y. Fu, "Measurement device independent twin tield quantum key distribution," *Scientific Reports*, vol. 9, no. 1, pp. 1–13, 2019.

[18] J. R. Zhu, C. M. Zhang and Q. Wang, "Efficient scheme for passive decoy-state reference-frame-independent quantum key distribution," *Physics Letters A*, vol. 383, no. 4, pp. 311–315, 2019.

[19] Y. Mao and Y. Sun, "JET: Dynamic join-exit-tree amortization and scheduling for contributory key management," *IEEE Transactions on Networking*, vol. 14, no. 5, pp. 1128–1140, 2006.

[20] P. Parthasarathi and S. Shankar, "Weighted ternary tree approach for secure group communication among mobile applications," *Wireless Personal Communications*, vol. 117, no. 4, pp. 2809–2829, 2021.