ISSN:0975-9646

Mitul K. Patel et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 3777-3783

An Efficient and Secure Multicast Key Management Scheme based on Star Topology

Neha Mittal[1], Vinod Kumar[2]

[1]Student, CDAC Noida, GGSIPU, Delhi,India
[2]Assistant Professor, CDAC Noida, GGSIPU, Delhi,India

Abstract-The today's world relies heavily over the applications, such as e-commerce, which are using the multicast communications, carried out over the insecure channel like internet, henceforth, the need for secure group communication is realized. The secure group communication helps in transmitting data in such a way that only the members of the group are able to receive the message. In 2012, K. Saravanan and T. Purusothaman proposed a scheme for multicast key management in which the group members generated the individual private keys and henceforth, reduced the burden upon the server. However, in some cases, the forward secrecy and backward secrecy may be violated upon member's leaving and joining the group respectively. In this paper, we proposed an improvement on the Saravanan-Purusothaman scheme to solve the above problem. The proposed scheme preserves the forward secrecy and backward secrecy in multicast group key management and therefore, is more secure. In addition, it also eliminates the rekeying process whenever a member joins/ leaves a group.

Keywords:- ***Group key management, Rekeying, Multicast***

1. INTRODUCTION

With the growth of the Internet, the usage of group communication becomes more popular. These applications include the pay TV channels, secure videoconferencing, multi-partner military action, wireless sensor, and ad hoc networks. In today's era, information security is the prime concern as with the technological advancements, the attackers are provided with more powerful and sophisticated tools. Today, the Internet is not totally secure for privacy. The usage of multicast applications increases day by day so it needs secure multicast services.

Multicasting is a simple way to send one stream of data to multiple users simultaneously. It helps in reducing the required bandwidth significantly, as it enables splitting of a single transmission between multiple users [9]. Multicasting not only optimizes the performance, but also enhances the efficiency of the network. For these reasons, multicasting has become the preferred transmission method for most group communication.

Group key management plays an important role in group communication. A common group key is required for individual users in the group for secure multicast communication. Group key has to be updated frequently whenever a member joins and leaves in order to provide forward and backward secrecy. Forward secrecy ensures that an expelled member cannot gather information about future multicast communication and backward secrecy ensures that a joining member cannot gather information about past multicast communication [11]. For this reason, group key needs to be updated with each membership change and given away to the authenticated users. This process is known as group re-keying.

For group communication , Wong et al. and Waller et al. has proposed a scheme 'logical key hierarchy (LKH) tree approach' [3, 4] which provides an efficient and secure mechanism to maintain the keys. In addition, communication and computation cost increases logarithmically with the group size for a join or depart request. Communication cost in LKH is reduced from $O(n)$ to $O(logn)$ in the rekeying method, where n is the number of group members. One-way function (OFT) scheme was proposed by Sherman and McGrew [5] to reduce the communication cost from $2logn-1$ to $logn$. These schemes need to rekeying message whenever member joins/leaves the group [3, 4, 5, 6, 7] . To overcome the above problem, Lin et al. [2] proposed the SBMK scheme using the star based architecture, in which there is no need for rekeying when a member joins and leaves the group. In SBMK, secret key is calculated by the key server using RSA algorithm [10] and then is unicast to every group member separately. Therefore, it increases the burden on the server. Efficient star topology based multicast key management algorithm was proposed by Saravanan, K. and T. Purusothaman [1] which is based on the RSA algorithm [10] in which secret keys are calculated by the group members. This eliminates the need to unicast the secret keys to every member separately, henceforth, reducing the load on the server to great extent. In that scheme when a member A leaves the group, the forward secrecy is thwarted if p_A equals p_B or q_B and q_A equals p_C or q_C, where p_i, q_i for i $\in \{A, B, C\}$ represents the prime numbers of members A, B, C. Also, when a new joining member D selects the prime numbers such that p_D equals p_E or q_E and q_D equals p_F or q_F for some anonymous members E and F, then the backward secrecy is violated. In this paper, we proposed an improvement on the Saravanan-Purusothaman scheme to solve the above problem. The proposed scheme preserves the forward secrecy and backward secrecy in multicast group key management and therefore, is more secure.

The proposed scheme differs from the previous work as it is always maintaining the forward and backward secrecy. In addition, our scheme does not need to maintain the key tree topology [3, 4, 5, 6, 7] and eliminates the rekeying process whenever member joins/leaves the group. In the aspect of security, our proposed scheme guarantees the group secrecy, forward secrecy, backward secrecy, The rest of the paper is organized as follows. In section 2, review work is discussed in brief. In section 3, we present our proposed scheme for multicast key management. Section 4 gives an example to illustrate our scheme. Section 5 provides security analysis .A comparison of our scheme with other schemes proposed is discussed in section 6. Finally the conclusions are made in section7.

2. BRIEF REVIEW OF SARAVANAN–PURUSOTHAMAN SCHEME

In this scheme, whenever a member wants to join the group, he/she is allowed to select two prime numbers and compute the secret information X as the product of their prime numbers. The private key is then computed by the member using Extended Euclidean algorithm and is unicast to the key server. The scheme eliminates the rekeying process when a member joins/leaves the group [1]. The Saravanan-Purusothaman scheme consists of three phases: the key assignment phase, the message encryption phase, the message decryption phase. We briefly describe the three phases as follows:-

Key assignment phase

The following steps have been performed for key assignment in the scheme.

1. The server declares the public parameter e to the users and also authenticates the member who wants to join the group
2. The individual member U_i chooses two prime number p_i and q_i randomly and also compute the product X_i and where, $X_i = p_i \times q_i$ and $\Phi(X_i)=(p_i-1) \times (q_i-1)$
3. The member calculates the private key by applying the extended Euclidean algorithm, computing a unique integer $d_i > \Phi(X_i)$, such that
   $$e \times d_i \equiv 1 \bmod (\Phi(X_i)) \qquad \text{eq (1)}$$
4. The authenticated individual members deliver their X value to the key server.
5. The server checks and accepts that X_i value only if it is unique among the other group members' values and stores the X_i value.

Message encryption phase

The steps to be performed by the key server for message encryption are described as follows:-

1. After determining the set of registered users, the server uses public key e as well as secret values of message Xi to encrypt a message M by using the following formulae [5]:-
   $$C = (M)^e \bmod \Pi X_i;\ i=1 \text{ to } n \qquad \text{eq (2)}$$
   Where n is equals to no of selected members to whom secret message is to be sent
2. After computing the cipher text C by the key server, the server sends a broadcast message to all the registered member of the group..

Message decryption phase

The process of message decryption by the members is as follows:-

Upon receiving the cipher text C, the individual member U_i can use its private key d_i and his/her public parameter X_i to decrypt the cipher text C sent from the key server and obtain the confidential message M by using the following formulae [5]:-

$$U_i \rightarrow M = (C \bmod X_i)^{di} \bmod X_i \qquad \text{eq (3)}$$

When a new member U_{i+1} want to join the group, the key assignment procedure is repeated. Also, when a member U_i leaves the group, the key server deletes the secret value X_i, thereby, removing the modulus operation w.r.t. $X_i(p_i \times q_i)$ in the cipher text computation.

In that scheme when a member A leaves the group, the forward secrecy is thwarted if p_A equals p_B or q_B and q_A equals p_C or q_C, where A, B, C are group members and p, q represents their respective prime numbers. Also, when a new joining member D selects the prime numbers such that p_D equals p_E or q_E and q_D equals p_F or q_F for some anonymous members E and F, then the backward secrecy becomes questionable. For example , Assume that there are three members U_1, U_2, U_3 in the group .

*U_1 selects p_1=11, q_1=17 computes X_1=11*17= 187*

*U_2 selects p2=31, q_2=47 computes X_2=31*47= 1457*

*U_3 selects p_3=17, q_3=47 computes X_3=17*47= 799*

$X_1 \neq X_2 \neq X_3$, so all the three members are registered in the multicast group. Let us say, U_3 leaves the group. Therefore, the modulus operation in cipher text computation is performed with respect to X_1 and X_2. Although member U_3 has left the group, but both his factors p_3 and q_3 of X_3 are already present in X_1 and X_2 respectively. Therefore, he can decrypt the future cipher text to obtain the original confidential message M using secret information X_3 and private key d_3. Hence, forward secrecy is not preserved in this case.

Similarly, when a U_4 wants to join the group, he can decrypt the past cipher text to obtain original confidential message M using secret information X_4 and private key d_4.Thus, backward secrecy is violated

3. PROPOSED SCHEME

In this section, we describe the proposed scheme which is based on RSA algorithm [10]. Our scheme differs from the Saravanan-Purusothaman scheme [1] as it is based on the uniqueness of the Xi values, but our scheme states that even the prime factors contributing to the Xi values must be unique. This characteristic always maintains the forward and backward secrecy in the proposed algorithm. In addition, our scheme does not need to maintain the key tree topology and eliminates the rekeying process whenever member joins/leaves from the group.

3.1 Efficient and secure GKM scheme

Our improvement is divided into three phases: the key assignment phase, the message encryption phase, the message decryption phase. Brief description of three phases is given as:

3.1.1 Key assignment phase

The following steps have been performed for key assignment in our proposed scheme.

1. The server declares the parameter e to the users and also authenticates the member who wants to join the group.
2. The individual member U_i chooses two prime number p_i and q_i randomly and also compute the product X_i and $\Phi(X_i)$
   Where, $X_i = p_i \times q_i$ and $\Phi(X_i)=(p_i-1) \times (q_i-1)$

3. The member calculates the private key by applying the extended Euclidean algorithm [10], computing a unique integer d_i, such that
   $e \times d_i \equiv 1 \bmod (\Phi(X_i))$ eq (4)
4. The authenticated members deliver their X_i value to the key server.
5. The server registers the member if and only if $gcd\ (Z, X_i) = 1$ and keeps X_i as secret. Then, the server stores the product of Xi values in Z.

3.1.2 Message encryption phase

The steps to be performed by server for message encryption are described as follows:-

1. After determining the set of registered user, the server uses e as well as secret values of message xi to encrypt message M by using the following formulae [1,2]:-
   $C = (M)^e \bmod \Pi X_i;\ i=1 \text{ to } n$ eq (5)
   where, n is the no. of selected members to whom secret message is to be sent.
2. After computing the cipher text C by the key server , the server sends a broadcast message to all the registered member of the group..

3.1.3 Message decryption phase

The process of message decryption by the members is as follows:-

After the cipher text C is received, the individual member U_i can decrypt C, using his private key d_i and secret information X_i, to obtain the confidential message M as follows [1,2]:-

$U_i \rightarrow M = (C \bmod X_i)^{di} \bmod X_i$ eq (6)

The proposed scheme ensures that only the members whose X_i values are used to encrypt the message are able to decrypt it.

3.2 Member joining and leaving the group

3.2.1 Member joining the group

Whenever a new member U_{i+1} wants to join the group , the server repeats the same procedure as that of key assignment. The rekeying process for existing members is eliminated because their secret information Xi and private key d_i remains unaffected.

3.2.2 Member leaving off the group

Whenever a member U_i want to leave the group, the remaining members do not need to modify the secret information X and private key d. The key server deletes the secret value X_i from the active list.

In the cipher text computation, the modulus operation is performed with the subset of X_k values of the remaining members U_k. For example, if the key server sends a message to the remaining group members, the encryption is performed as:

$C=M^e \bmod(X_1 \times X_2 \times \ldots \times X_{i-1} \times X_{i+1} \times \ldots \times X_n)$

Thus, U_i cannot decrypt the message.

4. An Example

In this section, a simple example is discussed to illustrate our scheme for group communication.

Key assignment phase

Suppose that there are six members U_1, U_2, U_3, U_4, U_5 and U_6 who want to join the group. Then the following steps are performed for key assignment:-

1. The server declares the parameter $e=101$ to the members and also authenticates the member who wants to join the group.
2. The individual member U_i chooses two prime number p_i and q_i randomly and also compute the product X_i and $\Phi(X_i)$
   Where, $X_i = p_i \times q_i$ and $\Phi(X_i) = (p_i-1) \times (q_i-1)$
   U_1 chooses $p_1 = 283$ and $q_1 = 367$ computes
   $X_1 = 103861$ and $\Phi(X_1) = 103212$
   U_2 chooses $p_2 = 163$ and $q_2 = 181$ computes
   $X_2 = 29503$ and $\Phi(X_2) = 29160$
   U_3 chooses $p_3 = 137$ and $q_3 = 191$ computes
   $X_3 = 26167$and $\Phi(X_3) = 25840$
   U_4 chooses $p_4 = 109$ and $q_4 = 179$ computes
   $X_4 = 19511$ and $\Phi(X_4) = 19224$
   U_5 chooses $p_5 = 193$ and $q_5 = 139$ computes
   $X_5 = 26827$ and $\Phi(X_5) = 26496$
   U_6 chooses $p_6 = 163$ and $q_6 = 191$ computes
   $X_6 = 31133$ and $\Phi(X_6) = 30780$
3. The private key is calculated by members using extended Euclidean algorithm as mentioned in eq(4).
   $101 \times d_1 \equiv 1 \bmod (\Phi(X_1)) \equiv 1 \bmod 103212$ and
   $d_1 = 92993$
   $101 \times d_2 \equiv 1 \bmod (\Phi(X_2)) \equiv 1 \bmod 29160$ and
   $d_2 = 2021$
   $101 \times d_3 \equiv 1 \bmod (\Phi(X_3)) \equiv 1 \bmod 25840$ and
   $d_3 = 4861$
   $101 \times d_4 \equiv 1 \bmod (\Phi(X_4)) \equiv 1 \bmod 19224$ and
   $d_4 = 18653$
   $101 \times d_5 \equiv 1 \bmod (\Phi(X_5)) \equiv 1 \bmod 26496$ and
   $d_5 = 25709$
   $101 \times d_6 \equiv 1 \bmod (\Phi(X_6)) \equiv 1 \bmod 30780$ and
   $d_6 = 29561$
4. The authenticated individual member send their $X_1, X_2, X_3, X_4, X_5, X_6$ values to the server.
5. In this step, server checks that p and q should be unique for all members using gcd function
   Server computes, $gcd(Z,X_1) =$
   $gcd\ (1,103861)=1$, U_1 is registered in the group.
   Then, $Z = Z \times X_1 = 1 \times 103861 = 103861$
   Server computes, $gcd(Z,X_2)$
   $= gcd(103861,29503)=1$, U_2 is registered in the group
   Then, $Z = Z \times X_2 = 103861 \times 29503$
   $= 3064211083$
   Server computes, $gcd(Z,X_3)$
   $= gcd(3064211083,26167)=1$, U_3 is registered in the group
   Then, $Z = Z \times X_3 = 3064211083 \times 26167$
   $= 80181211408861$
   Server computes, $gcd(Z,X_4)$
   $= gcd(80181211408861,19511)= 1$, U_4 is registered in the group
   Then, $Z = Z \times X_4 = 80181211408861 \times 19511$
   $= 1564415615798286971$
   Server computes,$gcd(Z,X_5)$

= gcd(1564415615798286971,26827)=1, U_5 registered in the group
Then, $Z = Z \times X_5 = 1564415615798286971 \times 26827 = 41968577725020644571017$
Server computes, $gcd(Z,X_6)$
$= gcd(41968577725020644571017,31133) \neq 1$, U_6 is not registered in the group
Then, $Z = 41968577725020644571017$

In Saravanan-Purusothaman scheme [1], the member U_6 gets successfully registered in the group. Whenever a member U_6 leaves, he / she also obtain confidential message M because factors p_6 and q_6 are already present in U_2 and U_3 respectively. Our proposed scheme does not add the member U_6 and hence, the forward secrecy is maintained.

Message encryption phase

1. When the server wants to send a secret message $M = 7$ to selected group members U_1, U_2, U_4 among a group of five *members* $(U_1, U_2, U_3, U_4, U_5)$ then the key server uses public key $e = 101$ as well as the values X_1, X_2, X_4, to encrypt the message .The cipher text is generated by the key server as follows
   $C = 7^{103} \bmod (X_1 \times X_2 \times X_4)$
   $C = 7^{103} \bmod (103861 * 29503 * 19511)$
   $C= 376363539506$
2. The key server then broadcasts cipher text C to all the members of the group.

Message decryption phase

After receiving the cipher text C, the members U_1, U_2, U_4 in the group can decrypt the encrypted message using their respective private keys d_1, d_2, d_4 and secret information X_1, X_2, X_4. The message M is decrypted by the members as follows:-

$U_1 \to M = (C \bmod X_1)^{d1} \bmod X_1$
$= (376363539506 \bmod 103861)^{92993} \bmod 103861 = 7$
$U_2 \to M = (C \bmod X_2)^{d2} \bmod X_2$
$= (376363539506 \bmod 29503)^{2021} \bmod 29503 = 7$
$U_3 \to M = (C \bmod X_3)^{d3} \bmod X_3$
$= (376363539506 \bmod 26167)^{4861} \bmod 26167 = 451$
$U_4 \to M = (C \bmod X_4)^{d4} \bmod X_4$
$= (376363539506 \bmod 19511)^{18653} \bmod 19511 = 7$
$U_5 \to M = (C \bmod X_5)^{d5} \bmod X_5$
$= (376363539506 \bmod 26827)^{25709} \bmod 26827$
$= 21592$

As shown, only members U_1, U_2, U_4 can obtain the original message.

Member joining

When a new member U_6 wants to join the group, the server performs the following steps :-

1. The server informs his public key $e=101$ to the member U_6 and also authenticates the member U_6.
2. The Member U_6 chooses two prime numbers $p_6=149$ and $q_6=113$ randomly and also compute the product X_6 and $\Phi(X_6)$
   Where, $X_6 = p_6 \times q_6$ and $\Phi(X_6) = (p_6-1) \times (q_6-1)$
   $X_6 = 16837$ and $\Phi(X_6) = 16576$
3. Private key is calculated by member U_6 using the extended Euclidean algorithm
   $101*d_6 \equiv 1 \bmod (\Phi(X_6)) \equiv 1 \bmod 26496$ and $d_6 = 6897$
4. The Member U_6 sends his X_6 value to the server
5. The server registers the member if and only if $gcd(Z, X_6) = 1$.
   Server computes, $gcd(Z,X_6) = gcd(41968577725020644571017,16837)=1$ U_6 is registered in the group
   $Z=Z \times X_6 = 41968577725020644571017 * 16837$
   $Z= 706624943156172592642213229$

When the key server wants to send a new confidential message $M= 5$ to all the members in the group. The server will compute new cipher text using the following formulae:-

$C = 5^{103} \bmod (X_1 \times X_2 \times X_3 \times X_4 \times X_5 \times X_6)$
$C=5^{103} \bmod (103861 \times 29503 \times 26167 \times 19511 \times 26827 \times 16837)$
$C= 698424121352690019387071398$

Then, the key server broadcast cipher text C to all the members of the group.

After receiving the cipher text C, all the pre-existing members U_1,U_2,U_3,U_4,U_5 and the new member U_6 in the group can decrypt the encrypted message using their private keys d_i and his/her secret information X_i, where, $i=1$ to 5 as shown below:-

$U_1 \to M = (C \bmod X_1)^{d1} \bmod X_1$
$(698424121352690019387071398 \bmod 103861)^{92993} \bmod 103861 = 5$
$U_2 \to M = (C \bmod X_2)^{d2} \bmod X_2$
$(698424121352690019387071398 \bmod 29503)^{2021} \bmod 29503 = 5$
$U_3 \to M = (C \bmod X_3)^{d3} \bmod X_3$
$(698424121352690019387071398 \bmod 26167)^{4861} \bmod 26167 = 5$
$U_4 \to M = (C \bmod X_4)^{d4} \bmod X_4$
$(698424121352690019387071398 \bmod 19511)^{18653} \bmod 19511 = 5$
$U_5 \to M = (C \bmod X_5)^{d5} \bmod X_5$
$(698424121352690019387071398 \bmod 26827)^{25709} \bmod 26827 = 5$
$U_6 \to M = (C \bmod X_6)^{d6} \bmod X_6$
$(698424121352690019387071398 \bmod 16837)^{6897} \bmod 16837 = 5$

Member leaving

Suppose the member U_1, U_4 and U_6 want to leave the group, the remaining members U_2, U_3, U_5 do not need to change their private keys d_2, d_3, d_5 and secret information X_2, X_3, X_5. The key server deletes the secret values X_1, X_4, X_6, corresponding to U_1, U_4, U_6, from the active list. Thus, the key server does not include X_1, X_4, X_6 in the cipher text computation.

Now, if the key server wants to send a confidential message $M= 15$ to all the members in the group. .The cipher text is generated by the key server as follows

$C = 15^{103} \bmod (X_2 \times X_3 \times X_5)$
$= 15^{103} \bmod (29503 * 26167 * 26827)$
$= 16499470714897$

The key server broadcast cipher text C to all the group member. After receiving the cipher text C, the confidential message M is decrypted by members as given below:-

$U_1 \rightarrow M = (C mod X_1)^{d1} \ mod \ X_1$
$=(16499470714897 mod 103861)^{92993} mod 103861 = 29942$
$U_4 \rightarrow M = (C mod X_4)^{d4} \ mod \ X_4$
$=(16499470714897 mod 19511)^{18653} mod 19511$
$= 10413$
$U_6 \rightarrow M = (C mod X_6)^{d6} \ mod \ X_6$
$=(16499470714897 mod 16837)^{6897} mod 16837$
$= 6204$

Members U_1, U_4, U_6 are not able to decrypt the message correctly. Thus, forward secrecy is maintained.

5. SECURITY ANALYSIS

We discuss the security analysis of the proposed scheme in this section.

5.1 Group key Secrecy

It ensures that for any adversary A, it is computationally difficult to obtain the private key d from publicly available parameter, similar to the difficulty of factoring the large integers as in RSA algorithm. Therefore, the security of the proposed scheme depends on RSA cryptosystem [10]. According to our scheme, when any member wants to join the group, then the two large prime numbers p, q are selected by the member. Server checks that p and q should be unique for all members and stores the value X_i as secret information. This secret value is known only to the corresponding member and the key server, then, it is extremely difficult for the adversary A to compute private key. Hence group secrecy is preserved.

5.2 Forward secrecy

It ensures that when any adversary A leaves the group, the key server deletes the secret information X_A from the active list .Therefore, in the cipher text computation, the modulus operation with respect to X_A is removed. Although, A has his private key d_A and secret information X_A but, he cannot access original future confidential message M. Hence, forward secrecy is always preserved.

5.3 Backward secrecy

Suppose that if adversary A wants to join the group, the server checks that p_A and q_A should be unique for all the members. Thus, the value $X_A = p_A \times q_A$ was not used to calculate the past cipher texts. Although, A has his own private key d_A and secret information X_A but, he cannot obtain the original past confidential message M. Hence backward secrecy is not violated.

5.4 Preventing the unauthorized access:-

Proposed scheme ensures that any unauthorized member cannot access the original message. The original message is decrypted by only those members whose X value is used to compute cipher text C. Suppose if unauthorized member U_k tries to use decryption method to obtain confidential message M then he fails to access the original message M. The decryption process is described below:

$$U_k \rightarrow M = (C \ mod \ X_k)^{dk} \ mod \ X_k$$
$$= (M^e \ mod \ \Pi \ X_i)^{dk} \ mod \ X_k$$
$$=(M^{edk} \ mod \ \Pi \ X_i) \ mod \ X_k$$
$$\neq M^{edj} mod \ X_j \neq M$$

Therefore, the unauthorized member cannot obtain the original plaintext.

5.5 Collaborative attacks

Suppose some collutive members $U_1, U_2....U_k$ having their private keys $d_1, d_2....d_k$ and secret information $X_1, X_2....X_k$, want to obtain private keys of some other members. Then, these members will reveal their private keys to each other in ordet to achieve the goal. The Private keys are generated using extended Euclidean algorithm [10] as follows:

$e \times d_1 \equiv 1 \ mod \ (\Phi(X_1))$
$e \times d_2 \equiv 1 \ mod \ (\Phi(X_2))$
$e \times d_k \equiv 1 \ mod \ (\Phi(X_k))$

The parameter e is already known to all the members, then even after revealing their private keys colluted members cannot obtain the private keys and secret information of the other members. Hence, collaborative attack is not possible in our scheme.

6. COMPARATIVE ANALYSIS

In this section, we compare the security parameters, computation overhead, communication overhead and storage overhead of the proposed scheme with the LKH scheme [3,4], OFT scheme [5], SBMK scheme [2] and Efficient SBMK scheme [1].

In Table 1, we compare the security parameter of the proposed scheme with the previous schemes [1, 2, 3, 4, 5]. As compare to the Efficient SBMK scheme [1], our scheme is more secure as it preserves the forward and backward secrecy in multicast group key management.

Schemes	Group secrecy	Forward secrecy	Backward Secrecy
LKH	Yes	Yes	Yes
OFT	Yes	Yes	Yes
SBMK	Yes	Yes	Yes
Efficient SBMK	Yes	No	No
Proposed scheme	Yes	Yes	Yes

Table 1. A comparison of security parameter

Schemes	Communication overhead		Computation overhead	
	Join	Leave	Join	Leave
LKH	2logn-1	logn	2log n-1	2log n
OFT	log n +1	log n+1	log n+1	logn+1
SBMK	1	0	1	0
Efficient SBMK	1	0	1	0
Proposed scheme	1	0	1	0

Table 2. A comparison of communication and computation cost

Table 2 illustrates the comparison of communication and computation overhead, between our scheme and the other schemes, for a join and leave request. In our scheme, communication and computation overhead are $O(1)$ when a member joins the group and is nil when a member leaves the group. Therefore, the cost is less compared to LKH [3, 4] and OFT [5]. Figure 1-4 shows the numerical results for communication and computation overhead during join/ leave operations.

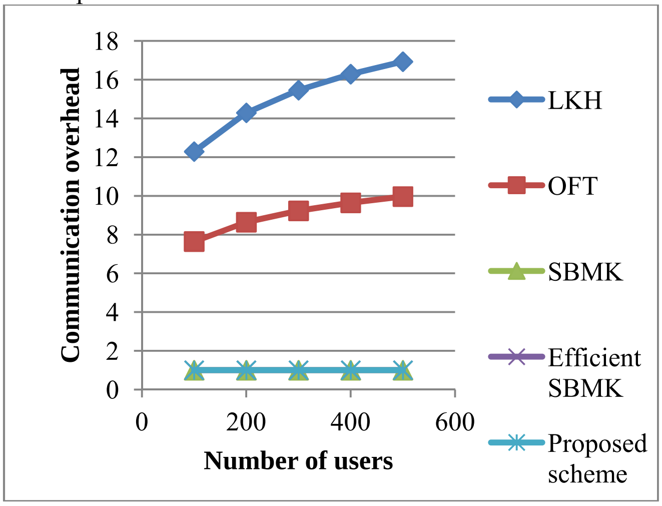

Figure 1. Communication overhead for join operation

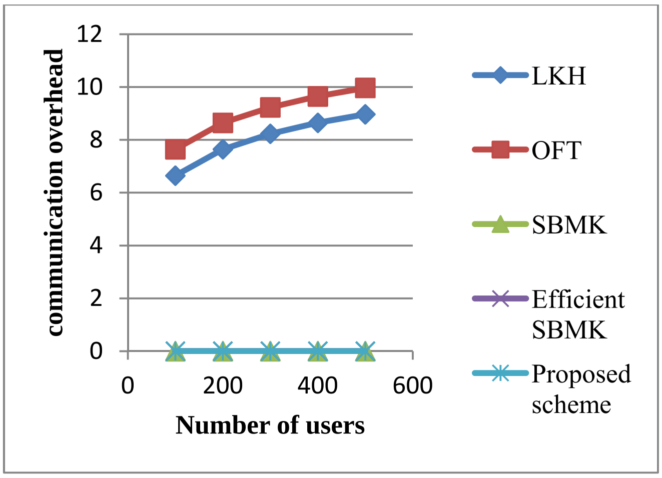

Figure 2. Communication overhead for leave operation

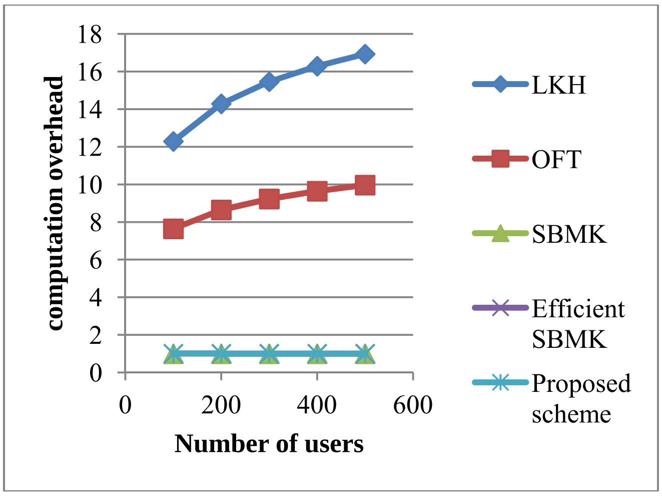

Figure 3. Computation overhead for join operation

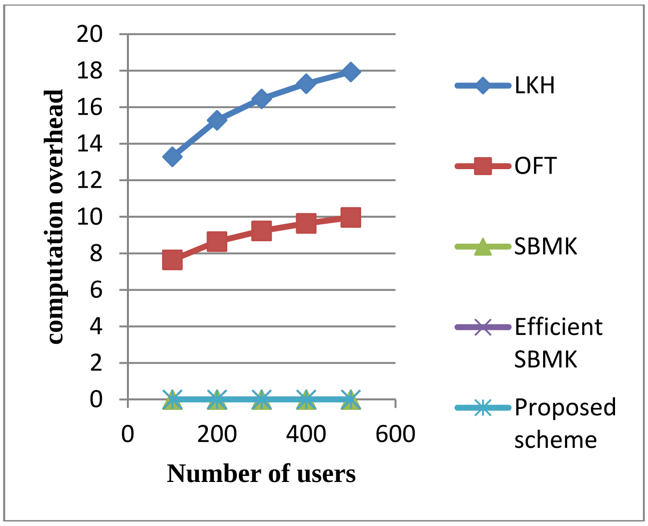

Figure 4. Computation overhead for leave operation

Schemes	Server	User
LKH	2n-1	log n+1
OFT	2n-1	2log n+1
SBMK	2n+2	3
Efficient SBMK	n+2	3
Proposed scheme	n+3	3

Table 3. A comparison of storage overhead during join and leave operation

The comparison of storage cost with the related schemes is shown in Table 4. Storage cost of our scheme on the server side is one more than the efficient SBMK scheme [1]. Our scheme enhances the security in terms of forward and backward secrecy. Figure 5 and 6 illustrates the numerical results for storage overhead at server and member site.

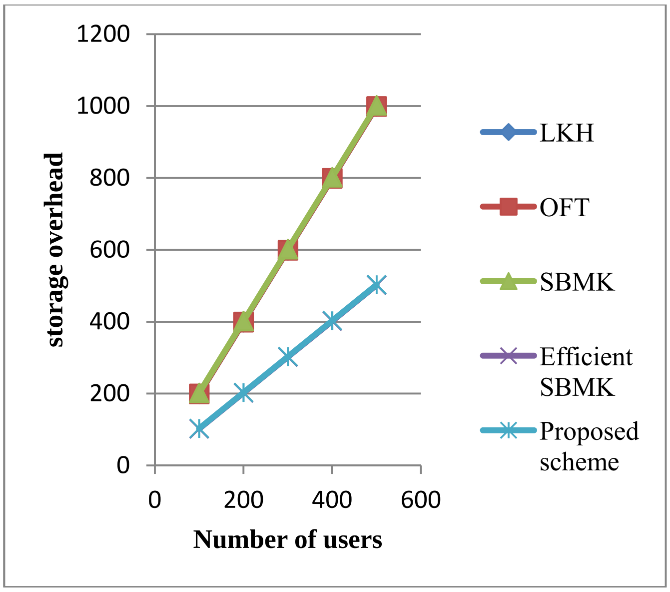

Figure 5. Storage overhead at the server site

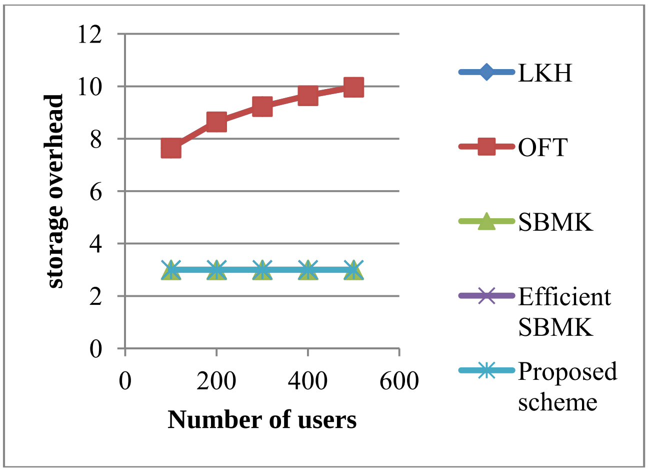

Figure 6. Storage overhead at member site

7. CONCLUSION

In this paper, we proposed an efficient and secure GKM scheme using RSA algorithm. Our proposed scheme maintains the uniqueness of the prime factors contributing to the secret value X. This characteristic helps in maintaining the forward and backward secrecy and thus overcoming the limitation of the Saravanan–Purusotham scheme. The proposed scheme does not affect the efficiency of the previous scheme, keeping the computation and communication cost as O(1). In addition, our scheme does not need to maintain the key tree topology and eliminates the rekeying process when a member joins/leaves the group. . In future, it can be extended to reduce the cost using ECC.

REFERENCES

[1]. Saravanan, K. and T. Purusothaman (2012) Efficient Star Topology based Multicast Key Management Algorithm. Journal of Computer Science 8 (6): pp. 951-956

[2]. Lin, I.C., S.S. Tang and C.M. Wang (2010) Multicast key management without rekeying processes. Comput. J., 53: 940-950.

[3]. Wallner, D., Harder, E. and Agee, R. (1999) Key Management for Multicast: Issues and Architectures. Technical Report RFC 2627, Internet Engineering Task Force.

[4]. Wong, C.K., Gouda, M. and Lam, S.S. (1998) Secure Group Communications Using Key Graphs. Proc. ACM SIGCOMM'98, Vancouver, Canada, September, pp. 68–79.

[5]. Sherman, A.T. and McGrew, D.A. (2003) Key establishment in large dynamic groups using one-way function trees. IEEE Trans. Softw. Eng. 29, 444–458

[6]. Heeyoul Kim, Seong-min Hong,H.Yoon and J.W.Cho(2005), Secure Group Communication with Multiplicative One-way Functions, Proc IEEE International Conference on Information Technology:-Coding and Computing, pp. 1-6.

[7]. Yi-Ruei Chen and Wen-Guey Tzeng(2012), Efficient and Provably-Secure Group Key Management Scheme Using Key Derivation, IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 295-300

[8]. Sridhar J K , Senthil Kumar R , Arun Kumar S(2012), Design An Optimal And Cost Effective Key Management Scheme For Secure Multicast Communication, Journal of Theoretical and Applied Information Technology, pp. 202-207.

[9]. Miller, C.K. (1999) Multicast Networking and Applications. Addison-Wesley, Reading, MA

[10]. Rivest, R.L., Shamir, A. and Adleman, L.M. (1978) A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 21, 120–126

[11]. V.Vasudevan and Joe Prathap P.M , (2009) .Analysis of the various key management algorithms and new proposal in the secure multicast communications research paper. International Journal of Computer Science and Information Security Vol.2, No.1.