Secure Multiple Multicast in Wireless Networks

Aiswarya.V
M Tech ISCF
Department Of It
Srm University

Arokiaraj Jovith. A
Assistant Professor
Department Of It
Srm University

***Abstract*— With the help of multiple multicast groups, Users can subscribe to multiple groups concurrently. Existing Group key Management Schemes mainly focusing on secure communication within single group are not applicable for multiple multicast environments because of irrelevant use of keys and larger keying overheads. This paper presents a multi-group management scheme which achieves hierarchical group access control .This scheme utilize asymmetric keys i.e. a master key and many slave keys which are wrought from Master key Management algorithm which is used for proficient distribution of group key. It makes less severe of being rekeying overhead by using asymmetry of the master key and multiple slave keys. if one of the slave keys is modified, the remaining keys can be still untouched by mitigating only the master key.**

Index Terms—Group key Management Hierarchical group access control, Multicast, Rekeying

I. INTRODUCTION

Multicast is communication between a single sender and multiple receivers on a network. In other words, Multicast is a delivery of a message or information to a group of destination computers simultaneously in a single transmission from the source. Typical uses include the updating of mobile personnel from a home office and the periodic issuance of online newsletters. Therefore, Multicast in wireless networks is awaited to cover the way for capable group communication by which group based communication such as video conferencing can be popularized.

The broadcasting channel, still, makes the wireless network exposed in danger to various security attacks then anyone can easily snoop on messages communicated in the air. To contrivance the multicast,i.e The distribution of data only to the members of the group in wireless networks, we require to have an access control mechanism for the broadcasted messages, which guarantees confidentiality, bulwarks digital contents, and facilitates precise accounting. Therefore, it is one of the key requisites for prosperous commercialization of these multicast accommodations in wireless networks.

The conventional way to provide an access control mechanism for the secure group communication is to employ asymmetric key, kenned as a group key, shared only by group members. Messages, encrypted by a member having a group key, can be decrypted by other group member sharing the same group key, which can ensure secure group communication. Albeit this mechanism, utilizing this hared group key, is an proficient way to ensure security, it causes some difficulties in maintaining an proficient key management system since the group key must be updated according to membership changes such as the utilizer leaving or joining, which is referred to as rekeying.

However, the subsisting Group Key Management schemes still face the constraint of rekeying performance as the number of multicast accommodations increases. However, in the prognostic able future, multiple multicast groups will coexist in a single network due to the emergence of many group-predicated applications. In such a situation, it is likely that the accommodation provider may suffer from considerable key management overhead for fortifying multiple multicast groups.

II. RELATED WORK

. In the rekeying procedure, the Key Distribution Centre distributes an incipient group key to each member to revoke an old group key so that a leaving (joining) utilizer is not sanctioned to access future (prior) messages, which are referred to as the forward (rearward) secrecy. Forward secrecy implicatively insinuates that a compromise of the current key should not compromise any future key. Rearward secrecy betokens that a compromise should not compromise any earlier key.

Let us consider a situation where a user tends to leave a multicast group. Before the user leaves, all the members have shared a group key to encrypt/decrypt messages among themselves. After the member leaves, the old group key should be revoked and updated with an incipient group key. This rekeying process may cause an abundance of key management overhead. Since the subsisting members do not have any shared secret keys except for the old group key, the KDC should distribute the incipient group key to these members in a unicast manner. As a result, it is conspicuous that the more the number of users in a multicast accommodation, the more astronomically immense the rekeying overhead would be.

To resolve this quandary, the authors in [1], [2] proposed a incipient data structure called the logical key hierarchy (LKH). In the LKH scheme, each group member shares a fraction of the key encryption keys (KEKs), the traffic encryption key (TEK),

and the individual keys (IKs) with the Key Distribution Center. More concretely, all these keys comprise a logical key tree, where the TEK is the root node, an IK is a leaf node, and the KEKs are the rest of the nodes in the key tree; each utilizer has the KEKs along the path from its IK (a leaf node) to the TEK (the root node). It can significantly reduce the amount of rekeying overhead which is a logarithmic function of a group size. In addition, there have been a bunch of variations of the tree-predicated approach such as one-way function tree [4] and one-way key derivation [5].

To resolve the above quandary, the hierarchical access control (HAC) scheme for Multiple Group Key Management has been proposed by Sun and Liu [6]. The HAC scheme can be visually perceived as an extension of the subsisting GKM scheme. While all users in a group have the same access right to the same data stream in the subsisting GKM scheme, the users in the HAC scheme have sundry access privileges for the different data streams

Motivated by the HAC scheme, Zhang and Wang [7] proposed an enhanced HAC (E-HAC) scheme. Homogeneous to the HAC scheme, the E-HAC scheme constructs a logical key graph. However, this scheme proposes the utilization of a resource group consisting of opportune data streams. Then, the E-HAC scheme encrypts all data streams in a resource group with a single TEK, which results in fewer TEKs than the HAC scheme. Notice that the rekeying performance of this scheme depends on the way it engenders its resource group. Since it is very arduous to make an efficient resource group for the users with perplexed cognations, its rekeying performance may withal decrease. In fact, the more perplexed cognations the users have, the worse the rekeying performance of both HAC schemes becomes.

III. METHODOLOGY

The method followed in this paper deals with client server relationship. Each client should register their details and authenticate utilizing their respective username and password. Clients have to cull the accommodations which they like. Clients can update and leave the accommodations at any time. Server can authenticate utilizing username and password. Server will maintain all client details which include Accommodation culls; Key cull etc. Server can access any details at any time, If the client leaves the group, Server will transmute the respective key in the accommodation and transmute the master key simultaneously.

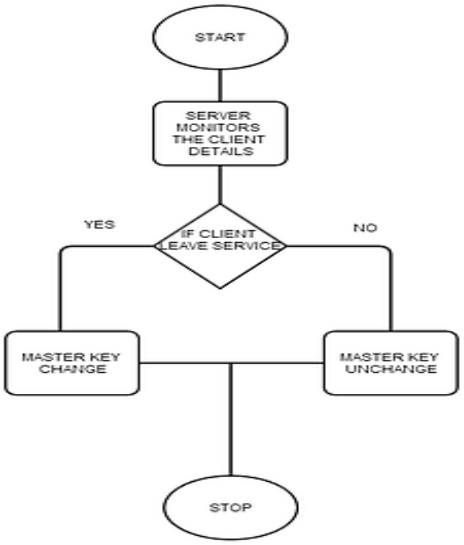

Fig 1 Methodology

IV. MASTER KEY ENCRYPTION SCHEME

The Master Key Encryption scheme is a RSA-predicated public-key cryptosystem[9] proposed by Koyama [10] where every utilizer in the RSA system has a key pair that consists of a public key and a private key, each of which is utilized for encryption and decryption in an asymmetric pair wise manner .The master key can be acclimated to encrypt messages, which can be decrypted by several different private keys or to decrypt messages encrypted with several different public keys. The most paramount feature of the MKE for MGKM is that one of the key pairs can be facilely transmuted by modifying only the master key, without any transmutations to other users' key pairs.

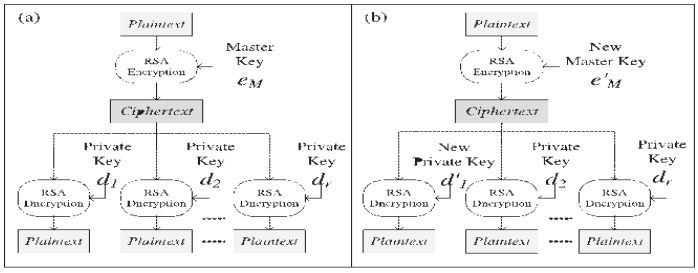

Fig 2 (a).The ideological diagram of Master Key Encryption (b).When d' is updated to d_1', the other private keys are still valid without any changes.

A. DETAILS OF MASTER KEY ENCRYPTION SCHEME

Consider r public-private key pairs. Let s_j and t_j be the j^{th} public and private key pairs respectively. Also let u_j and v_j be their prime numbers. (i.e.) $s_j t_j \equiv 1 \bmod \phi(u_j v_j)$ where $\phi(n)$ is Euler's totient function. If s_M and t_M are the master keys used for the encryption and decryption. The following Congruence equations are established for any plaintext P and Cipher text C.

$$P^{sM} \equiv P^{sj} \bmod (u_j v_j)$$
$$C^{tM} \equiv C^{tj} \bmod (u_j v_j)$$

The sufficient conditions of the above congruence equations are

$s_M \equiv s_j \pmod{\phi(u_j v_j)}$
$t_M \equiv t_j \pmod{\phi(u_j v_j)}$
$\forall 1 \le j \le r$

Obtaining s_M and t_M satisfying the above condition is not different from finding a solution to a system of congruence's through Chinese Remainder Theorem.However,unless the modulus $\phi(u_j v_j)$ is mutually prime to each other, the existence of solution cannot be guaranteed. Therefore, to make two master keys s_M and t_M from the public and the private keys, we employ the Generalized Chinese Remainder Theorem (GCRT)[12] where the system of congruence's

$x \equiv a_1 \bmod n_1$
$x \equiv a_2 \bmod n_2$
.
.
$x \equiv a_r \bmod n_r$ has an integrated solution that uniquely determines modulo $\text{lcm}(n_1,n_2,....n_r)$
if and only if $a_i \equiv a_j \bmod \gcd(n_i,n_j)$ where i and j are different integers in the range [1,r].

B. ALGORITHM

u_j and v_j are chosen from safe primes. A safe prime is a prime number of the form 2u +1 where u is also a prime. If prime number such as u_j and v_j are used, then s_j can be determined much easier. Let $u_1v_1,u_2v_2,.....u_rv_r$ be safe prime numbers and let $u_j=2c_j+1, v_j=2d_j+1$ for $1 \le j \le r$. From the definition of safe prime,c_j and d_j are also prime numbers
.

1.Determine $u_1,.....u_r,v_1,.....v_r$ of safe prime numbers
2.for j=1 for
3.$\phi_j=(u_j=1)x(v_j-1)$
4.$c_i=(u_i-1)/2$
5.$d_i=(v_i-1)/2$
6.s_i=4x Random +1
7.$t_i=s_i^{2(cj-1)(dj-1)-1} \bmod 4c_id_i$
8.End for
9.n=1
10.for j=1 to r
11.$n=nx(c_jd_j)$
12 End for
13 for j=1 to r;
14 $M[j]=n/(c_jd_j)$;
15$N[j]=M[j]^{(cj-1)(dj-1)-1} \bmod (c_jd_j)$
16 End for
17 s_M=0
18 for j=1 to r
19 $s_M=(s_M+(s_i \text{ x } M[j] \text{x } N[j])) \bmod n$
20 End for
21 while($s_M \bmod 4 != 1$)$s_M=s_M+n$
22 sleep
23 Interrupt (when k^{th} key pair is updated)
24.s_k=4xRandom+1;
25 $t_i=s_i^{2(cj-1)(dj-1)-1} \bmod 4c_id_i$
26 goto 17

Theorem 1: Let $u_1u_2....u_r$ and $v_1v_2.....v_r$ be safe prime numbers and all public keys must satisfy the following condition.
$.s_1 \equiv s_2 \equiv .....s_r \pmod 4$

. Then there exists a unique master key,s_M modulo $4c_1d_1 c_2d_2.....c_rd_r$ where $c_j=(u_j-1)/2$ and $d_j=(v_j-1)/2$ for j=1, 2.....n.

Proof: To apply the GCRT, the condition must be satisfied, where. $\forall j,k \in N$ $e_j \equiv e_k \pmod{\phi(u_jv_j),\phi(u_kv_k)}$_____(1)
.From the facts that $\phi(u_jv_j)=(u_j-1)v_j-1)=4c_jd_j$
and both c_j and $d_j$are prime numbers, we can know that
$\gcd(\phi(u_jv_j),\phi(u_kv_k))=\gcd(4c_jd_j,4c_kd_k)=4$
if $s1 \equiv s2 \equiv .....\equiv s_r \pmod 4$ condition(1) must be satisfied. also $\text{lcm}(\phi(u_1v_1),\phi(u_2v_2).....\phi(u_rv_r))$ is $4u_1v_1u_2v_2.....u_rv_r$.
Master key can be calculated by $s_M=\in_{i=1 \text{ to } r} s_jM_jk_j \pmod{4c_1d_1c_2d_2.....c_rd_r}$, where $M_j=\frac{c1d1c2d2\ldots crdr}{cjdj}$ and N_j is an integer such that $M_jN_j=1 \pmod{4c_jd_j}$

C. EXAMPLE

Assume a communication model and whereby, a data source sends encrypted message to two user groups. Also assume there is a Key Distribution Center which takes charge of the key management and each group key is different from others The data source wants to encrypt and transmit the message only once by using a master key.First of all, the KDC generates two public-private key pairs for the two groups. The KDC randomly choose the value of u_1,v_1,u_2 and v_2 from safe prime numbers.
$u_1=7$, $v_1=47$, $u_2=11$ and $v_{2=23}$
Then the KDC determines the exponents of the public and private key pairs , $s_1=53$, $t_1=125$, $s_2=9$and $t_2=49$ from lines 3 to 7 of the algorithm. The public-private key pairs of two groups become $K_1^{pub}=(s_1,u_1v_1)=(53,329)$, $K_1^{pri}=(t_1,u_1v_1)=(125,329)$, $K_2^{pub}=(s_2,u_2v_2)=(9,523)$ and $K_2^{pri}=(t_2,u_2v_2)=(49,253)$.

From lines 9 t0 22,KDC obtains the master key s_M=3,089.It is assumed that KDC distributes The public private key pairs to the corresponding groups and send the master key to the data source in a safe manner. Then if the data source wants to send "13" to two groups securely, it encrypts "13" with s_M. The cipher text will be $13^{3.089} \bmod u_1v_1u_2v_2$=13,481.Since all the users in group 1 have the private key k_1^{pri}=(125,329) they can obtain the plaintext as $13,841^{t1} \bmod u_1v_1$=13.likewise all the users in the group can obtain the plain text in the same manner.

Then consider the case that the key pair of group 1 should be updated when a user leaves group 1.From line 24 to 25 of algorithm, the KDC sets the new value of s_1 and t_1 to 13 and 85. Therefore, k_1^{pub} and k_1^{pri} become (13,329) and (85,329) respectively. After that s_M is recalculated as 14,089 from lines 17 to 21.The KDC renews the user's key of group 1 except for the leaving user, and the master key of the KDC. .At the data source, the plain text 13 is encrypted as $13^{14,089} \bmod u_1v_1u_2v_2$=54,214.

After receiving the ciphertext 54,214, each user of the two groups can decrypt it with its individual private key that as users of group 1 ->$54,214^{49} \bmod 253$=13 .Although the user group 1 know the old keys (s_1,u_1v_1)=(125,329) be cannot obtain the correct plain text from the cipher text through the old keys. It is noticed that, even if the key pair changes. ,the remaining key pairs can still be valid through by modifying the master key.

V. CONCLUSION

If client leaves/joins any service .it won't affect any operation in the process. Users can access to multiple groups at same time. Thus, Rekeying overhead is alleviated by means of asymmetry of the master key and slave keys. It makes use of efficient distribution of keys. The process can be untouched if any clients update any service which results in efficient distribution of keys in multicast group communication. By using a set comprising a master key and slave keys, a TEK can be efficiently distributed to multiple Service Groups (SGs). . It is expected that this scheme can be a practical solution for various group applications, especially for those requiring many SGs, such as TV streaming services charged on a channel by channel basis

REFERENCES

[1] C.K. Wong, M.G. Gouda, and S.S. Lam, "Secure Group Communications Using key Graphs," ACM SIGCOMM Computer Comm. Rev., vol. 28, pp. 68-79, 1998.
[2] D.M. Wallner, E.J. Harder, and R.C. Agee, "Key Management for Multicast: Issues and Architectures," IETF RFC 2627, http:// www.ietf.org/rfc/rfc2627.txt, June 1999.
[3] Y. Challal and H. Seba, "Group Key Management Protocols: A Novel Taxonomy," Int'l J. Information Technology, vol. 2, no. 1, pp. 105-118, 2005.
[4] S. McGrew, "Key Establishment in Large Dynamic Groups Using One-Way Function Trees," IEEE Trans. Software Eng., vol. 29, no. 5, pp. 444-458, May 2003.
[5] J.-C. Lin, F. Lai, and H.-C. Lee, "Efficient Group Key Management Protocol with One-Way Key Derivation," Proc. IEEE Conf. Local Computer Networks, pp. 336-343, http://doi.ieeecomputersociety. org/10.1109/LCN.2005.61, 2005.
[6] Y. Sun and K.J.R. Liu, "Hierarchical Group Access Control for Secure Multicast Communications," IEEE/ACM Trans. Networking, vol. 15, no. 6, pp. 1514-1526, Dec. 2007.
[7] Q Zhang and Y. Wang, "A Centralized Key Management Scheme for Hierarchical Access Control," Proc. IEEE GLOBECOM, pp. 2067-2071, 2004.
[8] D. Wallner, E. Harder, and R. Agee, "Key Management for Multicast: Issues and Architectures," IETF RFC 2627, 1999.
[9] R.L. Rivest, A. Shamir, and L. Adelman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Comm. ACM, vol. 21, no. 2, pp. 120-126, 1978.
[10] K. Koyama, "A Master Key for the RSA Public-Key Cryptosystem-tem," IEICE Trans. Information and Systems, pp. 163-170, 1982.
[11] A. Kondracki, "The Chinese Remainder Theorem," Formalized Math., vol. 6, no. 4, pp. 573-577, 1997.
[12] L.R. Yu and L.B. Luo, "The Generalization of the Chinese Remainder Theorem," Springer Acta Mathematica Sinica, vol. 18, no. 3, pp. 531-538, 2002.
[13] X. Zou, B. Ramamurthy, and S.S. Magliveras, "Chinese Remainder Theorem Based Hierarchical Access Control for Secure Group Communication," Proc. Third Int'l Conf. Information and Comm. Security, pp. 381-385, 2001.
[14] X. Zheng, C.-T. Huang, and M.M. Matthews, "Chinese Remainder Theorem Based Group Key Management," Proc. ACM Southeast Regional Conf., D. John and S.N. Kerr, eds., pp. 266-271, 2007.
[15] G. Caronni, M. Waldvogel, D. Sun, N. Weiler, and B. Plattner, "The VersaKey Framework: Versatile Group Key Management," IEEE J. Selected Areas in Comm., vol. 17, no. 9, pp. 1614-1631, Sept. 1999.
[16] Y. Kim, A. Perrig, and G. Tsudik, "Simple and Fault-Tolerant Key Agreement for Dynamic Collaborative Groups," Proc. Seventh ACM Conf. Computer and Comm. Security (SIGSAC '00), 2000.
[17] H. Lu, "A Novel High-Order Tree for Secure Multicast Key Management," IEEE Trans. Computers, vol. 54, no. 2, pp. 214-224, Feb. 2005.