A Novel Dual-Key Management Protocol Based on a Hierarchical Multicast Infrastructure in Mobile Internet

Jiannong Cao[1], Lin Liao [1,2], Guojun Wang [1,2], and Bin Xiao[1]

[1] Department of Computing, Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong
[2] School of Information Science and Engineering, Central South University,
Changsha, P.R. China
{csjcao, cslliao, csgjwang, csbxiao}@comp.polyu.edu.hk

Abstract. This paper describes a secure multicast infrastructure for large-scale group communications in Mobile Internet and proposes a key management protocol based on the infrastructure. The multicast communication domain is logically divided into several administrative areas with a key server associated with each area. All the key servers participate in a Public Key Infrastructure (PKI) as trusted entities known by the subgroup members. Therefore, it's efficient to minimize the re-key overhead implemented in the mobile host tier. The simulation results show that the proposed protocol has better performance compared to the centralized protocols without PKI support. The numbers of the real re-key messages and the re-key events are reduced to approximately 30% and 65%, respectively.

1 Introduction

The proliferation of the Internet technology and mobile computing devices gives rise to the growth of applications emerging in mobile Internet. Its popularity is fuelled by the growing importance of group-oriented and collaborative applications. One of the major challenges of group communications is secure and efficient group key management, where the basic step to secure the traffic data is to provide a cryptographic group key shared by all the members within a group.

However, the group key should be updated when the members change their status during the group communication session. Furthermore, the delivery of the valid key to all the members of a group is a challenging task due to the fact that the group key and group members can dynamically change. Since the communication among the group members may be inconsistent while data encryption keys are being updated, the challenge for any key management schemes is how to generate and distribute new group keys to authorized group members such that the communication remains secure while the overall impact on the system performance is minimized.

In mobile Internet, the frequent mobility of mobile hosts and limited bandwidth add complexity to the security problem in multicast group communications.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 560 – 569, 2005.

Especially when the number of group members becomes larger and the covering area becomes wider, the key distribution and re-keying process can impose a huge overhead. Researchers have proposed many key management approaches to minimize such an overhead in a scalable and secure manner.

In this paper, we firstly investigate the issues of designing key management protocols for multicast communications in mobile Internet. We propose a secure multicast infrastructure for large-scale group communications, and propose a key management protocol based on the infrastructure. Compared to the centralized key management algorithms under the mobile and dynamic environment, the proposed distributed key management protocol shows better performance in terms of the re-key events and the real re-key messages.

2 Related Work

The most important tasks involved in secure group communications include how to reduce the overhead of key distribution, how to minimize the number of encryptions and decryptions, how to reduce the number of re-key messages, and how to share a secure group key in a large-scale group [1,2,3]. Re-keying efficiency is evaluated based on the following aspects: the communication complexity, the time complexity and the storage requirements [4]. In a small-scale group, the tree structure is widely adopted to cope with key management, such as Tree Key Graph (TKG) [5] and Logical Key Hierarchy (LKH) [6]. In such schemes, there is a trusted third party, known as Key Distribution Centre (KDC), which maintains a tree of keys where the change of one sub-tree will inevitably trigger the re-key operation involving other sub-trees. Extending to large-scale groups, such a centralized KDC turns out to be somewhat burdensome and the single server turns out to be the point of attack for intruders.

One established way for enhancing the fault tolerance of centralized components is to distribute the components to a set of servers and use replication algorithms to mask faulty servers. Consequently, hierarchical approaches have recently been proposed to manage the distribution of the Traffic Encryption Key (TEK) in a scalable manner. The main idea of such a mechanism is that the whole group is divided into many disjoint subgroups, each of which is controlled by an Area Key Distributor (AKD), assisting the group key distribution with KDC. It is obvious that the overhead of the KDC will be diminished by means of distributed AKDs. Iolus [7], a hierarchical framework for secure multicast is proposed with this philosophy in a scalable manner. The divided subgroups sketch out a tree hierarchy with individual address and individual subgroup key for every subgroup respectively. In [8], an inter-domain key management protocol is proposed and each “leaf” region in this architecture is connected together through “trunk” region (backbone). There exists an Initiator Key Distributor (IKD) that holds a copy of the multicast-key and a copy of all the subgroup-keys. Thereafter, the IKD is actually the organizer in all the Autonomous Systems (ASs) as well as the initiator of the whole multicast instance. Due to the existence of global multicast-key, the re-keying of any one AS raised by some members’ dynamic change will give rise to the update of the global multicast-key, as well as the delivery of a new multicast-key to other ASs.

All the schemes summarized above focus on the wired environment. Since wireless devices are gaining in popularity with feasible network connections and powerful computing capabilities, the research of extending them to secure multicast group is worthy of being explored.

The impact of mobility on secure multicast is firstly considered in[9]. Besides the common issues in traditional networks, some other issues, such as transparency of the security features and self-efficiency of mobile users who is willing to take part in a secure group are identified as the specific features for mobile multicast. In mobile multicast scenario, we are facing the difficulty that we have to minimize the participation and computation of mobile hosts because of the intrinsic limitations, while the cost of the Group Manager (GM) tends to be minimized. In order to solve the problem, researchers come up with many solutions, such as matching the key management tree to the network topology called the TMKM tree[10], and the enhanced LKH protocol called LKH++[11].

Nevertheless, there are not many schemes, which solve key management in a large-scale group, and even less in a mobile and wireless environment. Based on the PKI [12] and the clustering techniques, in this paper, we propose a dual-key management protocol to combine the two into a hierarchical multicast infrastructure for secure mobile group communications.

3 The Hierarchical Multicast Infrastructure

3.1 The Proposed Infrastructure

The basic infrastructure for multicast communications in mobile Internet is depicted in Fig. 1. The proposed infrastructure consists of three tiers and four classes of network entities. The three tiers are: the Wired Station (WS) tier, the Access Proxy (AP) tier, and the Mobile Host (MH) tier. WS is the top tier of the infrastructure, consisting of some server stations with high computational capability and high stability. The multicast source disseminates data from this top tier. This tier is implemented through network entities typically found on the wired Internet today, such as routers, switches and servers, together with their corresponding network protocols. AP is the middle tier through which the mobile hosts access and connect to

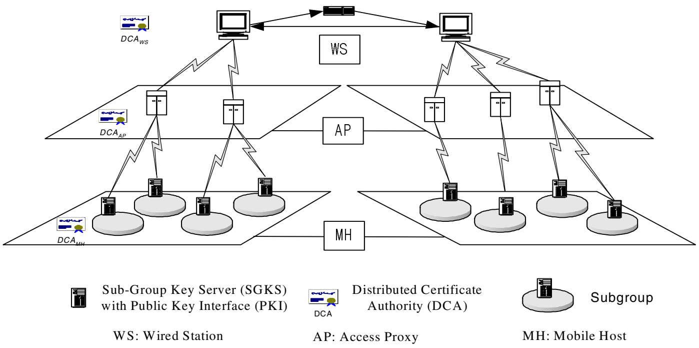
DCA_WS
WS
DCA_AP
AP
DCA_MH
MH
Sub-Group Key Server (SGKS) with Public Key Interface (PKI)
DCA
Distributed Certificate Authority (DCA)
Subgroup
WS: Wired Station
AP: Access Proxy
MH: Mobile Host

Fig. 1. Hierarchical multicast infrastructure

the WSs to receive multicast data. Suppose it is a cellular network connected to the wired backbone network, the AP tier can act as the Mobile Support Stations (MSS) role to provide the interfaces for mobile hosts. MH is the bottom tier of the infrastructure, consisting of a set of MHs. The MH is a host whose location relative to the rest of the network changes with time, as it is capable of moving between different locations.

Besides the entities of the three tiers mentioned above, another important entity called Sub-Group Key Server (SGKS) in the MH tier involving the dual-key is supposed (see Fig. 2). SGKS acts like the AKD but it differs from AKD in that it works in the PKI infrastructure. All the Sub-Group Key Servers (SGKS) are assumed to be trusted parties known by all the MHs and applied in a PKI infrastructure. To the upper-tier entities, the SGKS is the representation of one subgroup of the MH tier and the direct communication object of the MH tier; while to the MHs, the SGKS acts as the group manager of one subgroup of the MH tier. The MHs that have identified themselves with a particular SGKS are considered local to the SGKS.

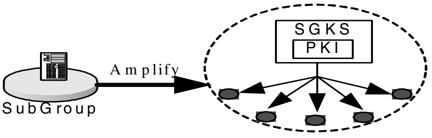

Fig. 2. Subgroup model

3.2 How the Infrastructure Works

As the WS tier is fixed and the AP tier is of lower mobility compared to the MH tier, we divide the MH tier into smaller administrative subgroups, with each subgroup associated with one SGKS as the group manager. When implementing a secure multicast instance in a mobile environment, the SGKS is not the member of the multicast group, but just kind of a known server. In general, the WS tier, the AP tier and the SGKS of the MH tier, are connected and they constitute the backbone of the network infrastructure. As mentioned above, the SGKS is different from the normal GM because it integrates the PKI interface into the unit as shown in Fig. 2. The responsibility of SGKS can be summarized as key distribution of the inner subgroup and the communications with the AP tier. Within each subgroup, the key distribution can be implemented by existing symmetric key management protocols such as Key Graph [5] or LKH [6]. On the other hand, the backbone entities exchange secret data encrypted by asymmetric keys, i.e. Public Key (PK) and Secret Key (SK), due to their low mobility and reliability.

3.3 Assumptions

- All the SGKSs of the MH tier are distributed into the multicast network as service centers, which initially need to register with their DCA for a pair of keys.

- We use SGKS$_i$ to denote the subgroup key server of SG$_i$ (subgroup with sub-index i). Two types of keys held by $SGKS_i$ are a pair of asymmetric keys, i.e. $PK(SGKS_i)$ and $SK(SGKS_i)$, and a symmetric subgroup key $K_{S,i}$. Both of them are integrated together in the entity SGKS. Specifically, $K_{S,i}$ stands for the subgroup key shared by the subgroup members, while $PK(SGKS_i)$ stands for the public key held by the $SGKS_i$ in representation of SG_i, and $SK(SGKS_i)$ stands for the other half of the asymmetric keys.
- It is assumed that a cross-tier authentication mechanism exists. Under such circumstances, certificates issued by one certain DCA can be authenticated by DCAs of other tiers. Consequently, DCAs of all tiers are authentic between each other.
- Once the upper tier obtains the PK of the entry it needs to communicate with, the PK is buffered into the buffer box identified by the SGKS's identity number of the certificate.
- We assume that a Distributed Certificate Authority (DCA) is associated with each tier of the infrastructure. Each DCA is responsible for the generation, authentication, expiration and regeneration of the PKs owned by the tier.
- An important requirement is that the available (trusted) SGKSs should be known in advance in order to reduce the possibility of masquerading.

4 The Key Management Protocol

4.1 Adoption of PKI

Many collaborative group settings require distributed key agreement techniques. In the PKI system, all the PKs are public and visible for enquiry and the owner of every PK is a unique one who can decrypt a message by using its secret SK. Unless the SK is expired or disclosed or a fake PK is detected by the DCA, all the PKs are convincing and firm during the valid period. Because of the advantage of the PKI that the security property is high and re-key cost is low, the PKI is widely used in current commercial and educational intranets. Nevertheless, entities in wireless network are not capable of offering the PK computation cost. Therefore, in our assumption, only the backbone hierarchy, which consists of the WSs, APs and all the SGKSs in the MH tier, is applied in the PKI mechanism.

During the initiation, the WSs, APs and all the SGKSs are required to register with the corresponding Distributed Certificate Authority (DCA) they belong to, to announce their identities and parent-children relationships. After the information validation, each DCA issues a pair of keys for all registered members, with a certificate for authentication. The data exchange between different tiers in the infrastructure relies on the PKI mechanism to transmit packets. For example, WS_i has to query DCA_{AP} (DCA of the AP tier) for the PK of its descendant AP by putting the checking information. Notice that WS_i only needs to query the direct downward DCA for efficient and convenient check. As to the SGKS of the MH tier, on receiving data packets encrypted by its PK from its parent entity, it starts to decrypt it and disseminates it to its subgroup members encrypted by its subgroup key. Fig. 3 illustrates of how it works.

Data transmission from AP_i of the AP tier to $SGKS_j$ of the MH tier.

Let P denotes the data packets; $+K(P)$ means using K to encrypt P ; and $-K(P)$ means using K to decrypt P .

$AP_i :$ $P' = +PK(SGKS_j)(P)$

$SGKS_j :$ $P = -SK(SGKS_j)(P')$ $P'' = +K_{S,j}(P)$

The MHs of $SGKS_j$ with subgroup key $K_{S,j} :$ $P = -K_{S,j}(P'')$

Fig. 3. Data transmission process

However, determining the owner of a public key or, conversely, determining the public key for a user, appears to be a basic functionality for executing transactions securely in any large-scale open system. For such authentication issues, many schemes such as DSSA and SPX are found to tackle them [13,14]. It's assumed that the PK searched from the trusted DCA is simply regarded as authentic in our proposed protocol. Once the DCA detects that any PK expires, it will inform the entity of the upper tier to renew its buffer.

4.2 Handling of Changes of Group Members

In this subsection, we describe how to handle the events of members join, members leave and members transfer. Among the three scenarios, the first two belong to *group dynamics* and the last one belongs to *population dynamics* [4].

4.2.1 Join Event

Let's consider the situation that a member needs to join SG_i. Upon approval, it sends to $SGKS_i$ a signal message to notify $SGKS_i$ of its arrival. Then, a new $K_{S,i}$ must be generated by the $SGKS_i$ and multicast to the previous members encrypted by the old $K_{S,i}$ as in most other schemes. $SGKS_i$ is responsible for the re-keying of the SG_i to ensure the backward confidentiality. Approaches for inner subgroup re-keying include logical tree-based algorithms such as key graph [5]. Because the $PK(SGKS_i)$ of $SGKS_i$ is not altering as the change of $K_{S,i}$, and the $K_{S,i}$ is only generated by $SGKS_i$, it's apparent that other subgroups needn't to carry out the re-key operations. All the re-key operations are accomplished by the SGKS within the subgroup, and the whole overhead is only concerned within the changing subgroup, which is apparently reduced compared to the centralized protocols without PKI [8,15,16].

4.2.2 Leave Event

When a member of SG_i tends to leave from the group session, actually it firstly needs to send a request signal of departure to the $SGKS_i$. Upon receiving the signal, the $SGKS_i$ starts the re-keying process to ensure forward confidentiality. Similar to a join event, the re-keying process only happens within the subgroup by multicasting the new $K_{S,i}$ to the remaining group members encrypted by members' individual keys. While in centralized key management schemes without PKI [8,15,16], the group manager still needs to update the global group key and subgroup key database and deliver the new group key to all the subgroups. From the above analysis, join and leave events merely trigger the re-key processes within that subgroup, and other subgroups remain unaffected.

4.2.3 Transfer Event

Mobility complicates the key management by allowing members to not only leave or join but also transfer between subgroups while remaining in the session (see Fig. 4). Mobility impacts performance only when members cross between subgroups, where re-keying messages must cross the boundaries resulting in performance degradation.

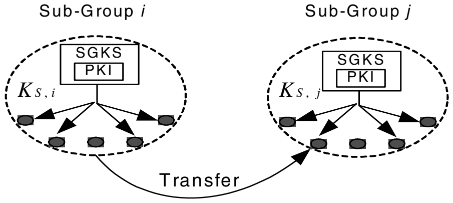

Fig. 4. Mobile nodes are transferring

The algorithms describing a member transferring from one subgroup to another subgroup are outlined as three approaches. It's analyzed that First Entry Delayed Re-key + Periodic (FEDRP) has a low re-key rate and message rate [17]. We adopt the scheme to do our comparison with the centralized schemes without the PKI. In FEDRP, when a member transfers from SG_i to SG_j, SG_i doesn't perform re-keying process right now. Thus, a member may accumulate $K_{S,i}$ as it visits different subgroups. If the entering member has previously visited SG_j, no rekey occurs for SG_j. If there is no visiting record, $SGKS_j$ will send the current $K_{S,j}$ to it by a secure unicast channel as needed. If the member is entering into SG_j for the first time, a new $K_{S,j}$ is generated and distributed through one multicast transmission (to current SG_j members using previous $K_{S,j}$) and one unicast transmission (to the newly entered member using a secure channel). To bound the maximum time that $K_{S,i}$ can be held by a member outside SG_i, each SGKS maintains a timer to bound it. Once the timer reaches the value, the subgroup re-keys itself and the timer is reset to zero. To trace member's movement history, $SGKS_i$ maintains a table of group members that hold a valid $K_{S,i}$ residing outside the subgroup. The table is reset once the member leaves the group or the timer expires. A member is added to the table when it transfers out of it, and a member is removed from the table when it transfers back.

In such a situation, FEDRP behaves with lower re-key rate than merely treated as firstly leave and then join [17]. Since the dual-role of the SGKS and the absence of the global group key, the transfer process is only handled by the two involved SGKSs, which still gets the benefit from the PKI system.

5 Simulation Studies

Because of the introduction of the public key infrastructure, the backbone of the proposed infrastructure relies on the PKI mechanism and re-keying processes occur within the subgroups. The complexity of our protocol, e.g. the join event, the leave event, is $O(\log n')$, here n' is the number of members of each subgroup; while in the centralized protocols without PKI [8,15,16] the complexity is $O(\log n)$ and n denotes the number of overall group members. In addition, in the centralized protocols without PKI support [8,15,16], the subgroups are not absolutely independent because of the existence of the global key manager managing all the subgroups. It is inevitable that the variation of one subgroup key will give rise to the global key update, and that the GM will still send many re-key messages to the other subgroups.

Three performance metrics are used:

- *Delay time (D_t)* measures the time difference between the time the member sends its willing to join or leave and the time the member is really granted for join or leave after the re-key process completes.
- *Re-key events (NumEvents)* measures the total number of control events to notify a new key when doing the re-key operations. The corresponding re-key events fall into three categories: signal events, unicast events, and multicast events.
- *Re-key messages (NumMsgs)* measures the total number of real re-key messages transmitted to all the mobile members for re-key operations. If a re-key message is a multicast re-key message, then there will be more than one user who receive it and use their correct keys to decrypt the required segments of the message respectively. As to the signal events and unicast events, such a re-key events is equivalent to one real re-key message. Therefore, *NumMsgs* is used to evaluate the real-transmitted number of re-key message packets in terms of the number of receivers in the multicast group.

We conducted the simulation to evaluate the performance with a comparison to the centralized key management protocol without PKI support [8,15,16]. We define the simulation time to be 600s and the whole area to be 600m*400m.

As to the delay time, it is obvious that our proposal performs better. In our proposed protocol, the join and leave procedures complete just after the subgroup re-creates a new subgroup key and distributes it to the valid members. In contrast, the centralized protocols without PKI support need two further steps to update the global multicast key and distribute it to the remaining subgroup controllers.

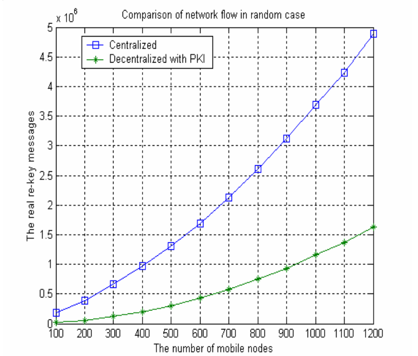

Fig. 5. The real re-key messages in random case

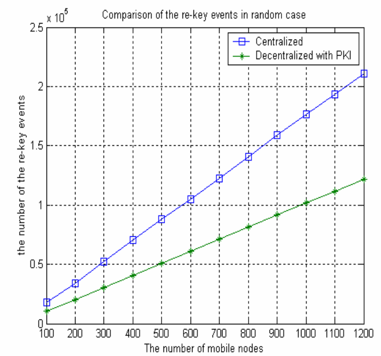

Fig. 6. The re-key events in random case

In Fig. 5 and Fig. 6, we plot the real re-key messages (*NumMsgs*) and the re-key events (*NumEvents*) for the two protocols with the change of network size in the random case, where all the members move at random speed and directions. Each data of the curve is the average result of ten rounds of independent running. We compare the data of the two protocols to get the ratios of improvements for each X-axis value. It's concluded that in average our protocol has 30% and 65% of the real re-key messages and the re-key events respectively in contrast to centralized protocols. The reason is that in our protocol re-keying is almost occurred within the subgroups, while the centralized one needs to have the global key update since the existence of a global group key all subgroups share.

We also carried out simulations for the situation where each member moves back and forth between two subgroups. In the regular case, we can find the differences become more evident in Fig. 7 and Fig. 8 than in the random case. Due to the reason that the movement between two subgroups in the centralized protocols give rise to two subgroup re-keying and the global key update, which is much bigger than our protocol in PKI. In such regular case, our protocol just gets 25% and 55% of the real re-key messages and the re-key events compared to centralized protocols without PKI in average respectively [8,15,16].

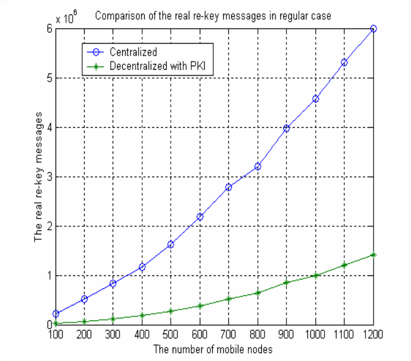

Fig. 7. The real re-key messages in regular case

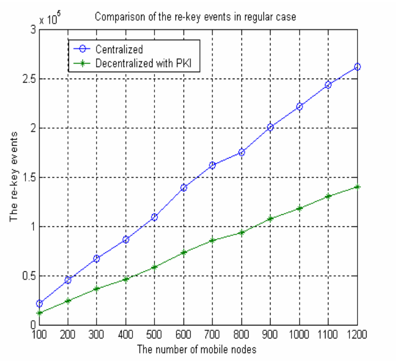

Fig. 8. The re-key events in regular case

6 Conclusions

The proposed dual-key management protocol with PKI support has better performance than the centralized key management protocol without using PKI. Such a conclusion is drawn on the basis of the stability of the PKI system and the trustiness of the SGKSs. However, the proposed protocol requires that the system heavily relies on the PKI infrastructure. Once the authentication of the PKI fails, the consequence will be serious. The delay due to the decryption of SK and encryption of subgroup key may also affect the performance. Nonetheless, the computation power of the SGKS counterbalances the delay. Although some delay cannot actually be avoided, in large-scale multicast communications, such a drawback will not affect the whole performance much and our proposed protocol outperforms centralized protocols without PKI support.

Acknowledgment

This work is supported in part by the Hong Kong Polytechnic University Central Research Grant *G-YY41*, and in part by the University Grant Council of Hong Kong under the CERG Grant PolyU *5170/03E*.

References

1. B. DeCleene, L. Dondeti, S. Griffin. Secure Group Communications for Wireless Networks. *Military Communications Conference (MILCOM 2001).* Vol. 1, pp. 113-117, 2001.
2. T. Hardjono, B. Cain, N. Doraswamy. A Framework for Group Key Management for Multicast Security. *Internet Draft*, draft-ietf-ipsec-gkmframework-03.txt, 2000.
3. S. Rafaeli, D. Hutchison. A Survey of Key Management for Secure Group Communication. *ACM Computing Surveys*, Vol. 35, No. 3, pp. 309-329, 2003.
4. D. Bruschi, E. Rosti. Secure Multicast in Wireless Networks of Mobile Hosts: Protocols and Issues. *ACM/Kluwer Mobile Networks and Applications*, Kluwer Academic Publishers, Vol. 7, pp. 503-511, 2002.
5. C. K. Wong, M. Gouda, S. S. Lam. Secure Group Communications Using Key Graphs. *IEEE/ACM Transactions on Networking*, Vol. 8, No.1, pp. 16-30, 2000.
6. H. Harney, E. Harder. Logical Key Hierarchy Protocol. *Internet draft*, draft-harney-sparta-lkhp-sec-00.txt, 1999.
7. S. Mittra. Iolus: A Framework for Scalable Secure Multicasting. *Proceedings of ACM SIGCOMM'97*, pp. 277-288, 1997.
8. T. Hardjono, B. Cain. Secure and Scalable Inter-domain Group Key Management for N-to-N Multicast. *Proceedings of International Conference on Parallel and Distributed Systems*, pp. 478-485, 1998.
9. L. Gong, N. Sacham. Multicast Security and its Extension to a Mobile Environment. *ACM/Kluwer Wireless Networks*, Vol.1, No. 3, pp. 281-295, 1995.
10. Y. Sun, W. Trappe, K. J. Ray Liu. A Scalable Multicast Key Management Scheme for Heterogeneous Wireless Networks. *IEEE/ACM Transactions on Networking*, Vol.12, No. 4, pp. 653-666, 2004.
11. R. Di Pietro, L. V. Mancini, S. Jajodia. Efficient and Secure Key Management for Wireless Mobile Communications. *Proceedings of the Second ACM International Workshop on Principles of Mobile Computing*, pp. 66-73, 2002.
12. P. R. Zimmermann. The Official PGP User's Guide. MIT Press, 1995.
13. M. Gasser, A. Goldstein, C. Kaufman, B. Lampson. The Digital Distributed System Security Architecture. *Proceedings of the 12th NIST/NCSC National Conference on Computer Security*, pp. 305-319, 1989.
14. J. J. Tardo, K. Alagappan. PX: Global Authentication Using Public Key Certificates. *Proceedings of IEEE Symposium on Research in Security and Privacy*, pp. 232-244, 1991.
15. G. Wang, L. Liao, J. Cao, K. Chan. Key Management for Secure Multicast Using the RingNet Hierarchy. *Proceedings of International Symposium on Computational and Information Sciences (CIS 2004),* LNCS (Spring-Verlag), Vol. 3314, pp.77-84, 2004.
16. T. Kostas, D. Kiwior, G. Rajappan, M. Dalal. Key Management for Secure Multicast Group in Mobile Networks. *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX'03)*, pp. 1-3, 2003.
17. C. Zhang, B. Decleene, J. Kurose, D. Towsley. Comparison of Inter-area Re-keying Algorithms for Secure Wireless Group Communications. *Performance Evaluation*, Elsevier Science, Vol. 49, pp. 1-20, 2002.