A Secure Group Key Management Scheme for Wireless Cellular Systems[*]

Hwayoung Um and Edward J. Delp
(Corresponding author: Hwayoung Um)

School of Electrical and Computer Engineering, Purdue University
465 Northwestern Avenue, West Lafayette, Indiana 47907-2035, USA (Email: ace@ecn.purdue.edu)

(Selected paper from ITNG 2006)

Abstract

In wireless networks, secure multicast protocols are more difficult to implement efficiently due to the dynamic nature of the multicast group and the scarcity of bandwidth at the receiving and transmitting ends. Mobility is one of the most distinct features to be considered in wireless networks. Moving users onto the key tree causes extra key management resources even though they are still in service. To take care of frequent handoff between wireless access networks, it is necessary to reduce the number of rekeying messages and the size of the messages. The multicast protocol used in wired networks does not perform well in wireless networks because multicast structures are fragile as the mobile node moves and connectivity changes. When we choose a key management scheme, the structure of the wireless network should be considered very carefully. In this paper, we design a key management tree such that neighbors on the key tree are also physical neighbors on the cellular network. By tracking the user location, we localize the delivery of rekeying messages to the users who need them. This lessens the amount of traffic in wireless and wired intervals of the network. The group key management scheme uses the pre-positioned secret sharing scheme.

Keywords: Cellular system, group key management, mobility, pre-positioned secret sharing

1 Introduction

As the technology and popularity of cellular networks like 3G and CDMA2000 [5] grows, there has been considerable progress in the area of multimedia streaming over wireless networks in the last few years. A lot of applications, such as video conferencing, video-on-demand, stock-quote distribution, and software update, have been developed for streaming digital multimedia contents to a set of clients. In such applications, the multicast protocol plays an important role because it can efficiently deliver data from a source to multiple receivers. It reduces the bandwidth of the wireless networks and the computational overhead of mobile devices. This makes multicast an ideal technology for communication among a large group of users because wireless channels are very limited and precious resources [26, 27, 28]. An important issue is how to provide security to these applications. Security could involve a number of issues, like authentication of clients, secure data transmission and copyright protection. For each of these security needs, a number of security protocols (especially for multicast) have been developed and a great deal of research continues in this area. The problem then is how to flexibly integrate security protocols into multimedia streaming applications even though these applications are usually developed without security. As part of the new issues involved with multicast communications, multicast security and scalability have received particular attention due to the various vulnerabilities found in their application [13, 31, 14, 15, 10, 22, 23].

Multicast protocols require an access control mechanism such that only authorized members can access group communications. Access control is usually achieved by encrypting the content with an encryption key. This key is known as the session key (SK) that is shared by all valid group members. Access control typically employs a tree of encryption keys to update and maintain the SK. Tree-based schemes [13, 31] have advantages that include computation, communication, and storage resources for the user and the group manager. In such schemes, the group key should be changed periodically or after a user leaves or joins the service to prevent the leaving/joining user from accessing future/prior communication. This is known as "forward message secrecy" and "backward message secrecy," respectively. Key management schemes in multicasting should also be "scalable." By scalable we mean that the overhead involved in key exchange, updates, data transmission, and encryption must not be dependent on the size of the multicast group. Moreover, addition or removal of a host from the group should not affect the other

[*] A preliminary version of this work appeared in proceedings of international conference on Information Technology: New Generations (ITNG 2006).

members. This is known as the “1 affects n” scalability rule.

1.1 Basic Multicast Steps

The process of secure multicast is composed of two basic steps: key distribution and transmission of encrypted data. Once a group key has been securely established among the members of the multicast group, it can be used with any fast symmetric encryption algorithm to encrypt the data to be transmitted. Therefore, the challenge in developing secure multicast protocols is primarily in designing efficient schemes for the key distribution. Let us indicate the conditions that require establishment of a new group key during a secure multicast session:

- When the multicast group is formed - This is the first time that the group key is established at the beginning of the session.

- When a member of the group leaves or is expelled - Rekeying is required to prevent the member from using its key to decrypt future communication. This is called Forward Rekeying.

- When a new member joins the group - Rekeying is required if the new member is to be prevented from decrypting earlier communication (which it could have stored). This is called Backward Rekeying.

- When a timeout occurs - Keys are usually associated with a timeout after which they become potentially insecure. The length of this timeout depends on many factors like key length, encryption algorithm used, wired or wireless network etc. If such a timeout exists and occurs, rekeying is required. This is referred to as Periodic Rekeying.

We thus see that a key distribution can take place quite often during a multicast session. It is very important to optimize this process. Most of the secure multicast protocols, therefore, differ from each other only in the key distribution scheme.

Many secure multicasting protocols have been proposed in the past few years. Existing key distribution schemes can be classified into non-scalable and scalable protocols. Some of this discussion is drawn from [18, 6, 4].

1.2 Key Management Role

Key management plays an important role enforcing access control on the group key and consequently on the group communication. It supports the establishment and maintenance of key relationships between valid groups according to a security policy being enforced on the group. It encompasses techniques and procedures that can carry out [18]:

- Providing member identification and authentication. Authentication is important in order to prevent an intruder from impersonating a legitimate group member. In addition, it is important to prevent attackers from impersonating key managers. Thus, authentication mechanisms must be used to allow an entity to verify whether another entity is really what it claims to be.

- Access control. After a group has been identified, its join operation should be validated. Access control is performed in order to validate group members before giving them access to group communication, in particular the group key.

- Generation, distribution and installation of key material. It is necessary to change the key at regular intervals to safeguard its secrecy. Additional care must be taken when choosing a new key to guarantee key independence. Each key must be completely independent from any previous used and future keys, otherwise compromised keys may reveal other keys.

1.3 Review of Group Key Management Schemes

We can classify the scalable protocols into two main classes: centralized group key management protocols and distributed key management protocols [6].

The distributed key management approach is characterized by having no group controller [3, 1, 12, 9]. The group key can be either generated in a contributory fashion, where all members contribute their own share to computation of the group key, or generated by one member. In the latter case, although it is fault-tolerant, it may not be safe to leave any member to generate new keys since key generation requires secure mechanisms, such as random number generators, that may not be available to all members. Moreover, in most contributory protocols (apart from tree-based approaches), processing time and communication requirements increase linearly in term of the number of members. Additionally, contributory protocols require each user to be aware of the group membership list to make sure that the protocols are robust [18]. The basic idea here is that every member can compute a group key when all blinded keys on the key tree are known. After any group membership event, every member unambiguously adds or removes some nodes related with the event, and invalidates all keys and blinded keys related with the affected nodes. A special group member, the *sponsor*, then takes on a role to compute keys and blinded keys and to broadcast the key tree to the group. If a sponsor could not compute the group key, then the next sponsor will compute comes into play. Eventually, some sponsor will compute the group key and all blinded keys, and broadcast the entire key tree to facilitate the computation of the group key by the other members of the group.

In centralized schemes, a single entity is employed for controlling the whole group, hence a group key management protocol seeks to minimize storage requirements, computational power on both client and server sides, and bandwidth utilization. Many schemes use the logical key hierarchy to give the scalability. Here we will explain the logical key hierarchy schemes in detail.

Several contributions propose the use of a Logical Key Hierarchy (LKH) [13, 31, 18, 12, 29, 17, 2]. In this approach, a Key Distribution Center (KDC) maintains a tree of keys. The nodes of the tree hold Key Encryption Keys (KEK). The leaves of the logical key tree correspond to group members and each leaf holds a KEK associated with that one member. Each member receives and maintains a copy of the KEK associated with its leaf and the KEKs corresponding to each node in the path from its parent leaf to the root. The key held by the root of the tree is the group key. For a balanced tree, each member stores at most $(log_2 N) + 1$ keys, where $(log_2 N)$ is the height of the tree and N is the group size.

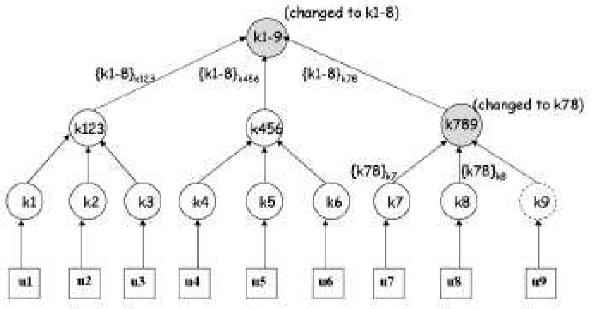

Figure 1: An example of logical key hierarchy

For example, as shown in Figure 1, suppose u_9 is granted to join the upper key graph in the figure. The joining point is k-node k_{78} in the key graph, and the key of this k-node is changed to k_{78} in the new key graph below. Moreover, the group key at the root is changed from k_{1-8} to k_{1-9}. Users $u_1, \cdots, u_6$ only need the new group key k_{1-9}, while users u_7, u_8, and u_9 need new group key k_{1-9} as well as the new key k_{789} to be shared by them.

After granting a leave request from user u, server s updates the key graph by deleting the u-node for user u and the k-node for its individual key from the key graph. The parent of the k-node for its individual key is called the leaving point. To prevent the leaving user from accessing future communications, all keys along the path from the leaving point to the root node need to be changed. After generating new keys for these k-nodes, server s needs to securely distribute them to the remaining users. For example, as shown in Figure 1, suppose u_9 is granted to leave the lower key graph in the figure. The leaving point is the k-node for k_{789} in the key graph, and the key of this k-node is changed to k_{78} in the new key graph above. Moreover, the group key is also changed from k_{1-9} to k_{1-8}. Users $u_1, \cdots, u_6$ only need the new group key k_{1-8}, while users u_7, u_8, and u_9 need new group key k_{1-8} as well as the new key k_{78} to be shared by them.

The algorithm proposed by Waldvogel [30] is different for joining operations. Instead of generating fresh keys and sending them to members already in the group, all keys affected by the membership change are passed through a one-way function. Every member that already knew the old key can calculate the new one. Hence, the new keys do not need to be sent and every member can calculate them locally [18].

The efficient large-group key (ELK) protocol is proposed by Perrig [17]. The ELK protocol uses a hierarchical tree and is similar to one-way function tree (OFM) [12] in the sense that a parent node key is generated from its children keys. ELK uses pseudo-random functions (PRFs) to build and manipulate the keys in the hierarchical tree. A PRF uses a key K on the input M of length m to generate output of length n. Using the PRF on a key, it is possible to drive four different keys to be used in the different contexts. ELK employs a timely rekey, which means that the key tree completely updated in each time of interval. ELK also introduce the idea of *hints*. A hint is a piece of information, which is smaller than a key update message, that can be used to recover possible lost rekey message updates. It is provided to improve the reliability of the rekey operation and it is conveyed in data messages [18].

LKH schemes are very efficient and hence scalable protocol for group rekeying when compared to a unicast-based näive approach. Let N be the group size, d be the degree of the key tree, then the communication cost for rekeying is $O(log_d N)$, whereas the näive approach requires a communication cost of $O(N)$. However for a large group with very dynamic memberships, LKH may not perform well because it performs a group rekeying for every membership change.

1.4 The Problem - User Mobility

In wireless networks, secure multicast protocols are more difficult to implement efficiently due to the dynamic nature of the multicast group and the scarcity of bandwidth at the receiving and transmitting ends. Mobility is one of the most distinct features to be considered in a wireless network. Moving users onto the tree causes extra key management resources even though they are still in service. To take care of frequent handoff between access points, it is necessary to reduce the number of re-keying messages and the size of the messages. The multicast protocol used in wired networks does not perform well in wireless networks because multicast structures are fragile as the mobile node moves and connectivity changes. When we choose a key management scheme, the structure of the wireless network should be considered very carefully. For example, the wireless cellular network has a unique hierarchy structure such that a key management scheme should be easy to deploy. Some papers already address the access control schemes in wireless networks. In [24], they propose the topology matching key management trees (TKMK) and test in respect to the communication cost. By matching the key tree to the network topology, the communication traffic is reduced by 33%

- 45% compared to the conventional key trees that are independent of the network. In [2], they propose baseline, immediate, delayed and periodic re-keying schemes and test them in the wireless LAN network. To our best knowledge, this is the first paper that computes the handoff impact of centralized key management scheme in the real wireless cellular networks.

1.5 Our Solutions

Several protocols exist for the efficient key distribution during secure multicast. The main aims of these protocols include network architecture independence, robustness and scalability. In this paper, we design a key management tree such that the neighbors on the key tree are also physical neighbors on the cellular network. By tracking the user location, we localize the delivery of re-keying messages to the users who need them. This lessens the amount of traffic in wireless and wired intervals of network. The group key management scheme uses the prepositioned secret sharing scheme [26, 27].

1.6 Paper Outline

The remainder of this paper is composed of as follows: In Section 2, a location-based handoff scheme is explained. In Section 3, a location tracking scheme is presented. In Section 4, the basic concept of the pre-positioned secret sharing are shown. In Section 5, group key management is explained in detail. In Section 6, simulation results are explained. In Section 7, the conclusion is presented.

2 Handoff Schemes

There are 2 types of handoffs: a hard handoff and a soft handoff, as shown in Figure 2. In the hard handoff, the connection to the current cell is broken, and the connection to the new cell is made. This is known as a "break-before-make" handoff. The soft handoff refers to the overlapping of Base Station (BS) coverage zones, so that every cell phone is always well within range of at least one base station. In some cases, mobile sets transmit signals to, and receive signals from, more than one BS at a time. This is known as a "make-before-break" handoff.

We describe a soft handoff scheme and a hard handoff scheme based on the location of a user instead of the use of the strength of a pilot signal from the user to the BS, as shown in Figure 3. There are two important parameters, L_ADD and L_DROP. L_ADD and L_DROP indicate the beginning of handoff and the termination of handoff based on the location of the user.

2.1 Soft Handoff

In general, the system administrator decides the values of two parameters. In our simulation, 30% of soft handoff area is used. That is, the L_ADD is the boundary of overlapping area of two BSs and the L_DROP is the middle of two BSs as shown in Figure 3. In this example, a MS moves from A of BS1 to B of BS2. The Mobile Station (MS) requests a handoff to the neighboring BS when the location of neighboring BS exceeds the handoff threshold L_ADD. If the handoff request is accepted in the neighboring BS, BS2, the MS maintains two traffic channels assigned by the serving BS, BS1 and the neighboring BS. As the MS moves away from the serving BS and approaches the neighboring BS, the location of MS falls below the handoff drop threshold L_DROP for the servicing BS. If the location of the MS is close to the neighboring BS during the specific time interval, the traffic channel assigned by the serving BS is released, and the handoff is terminated.

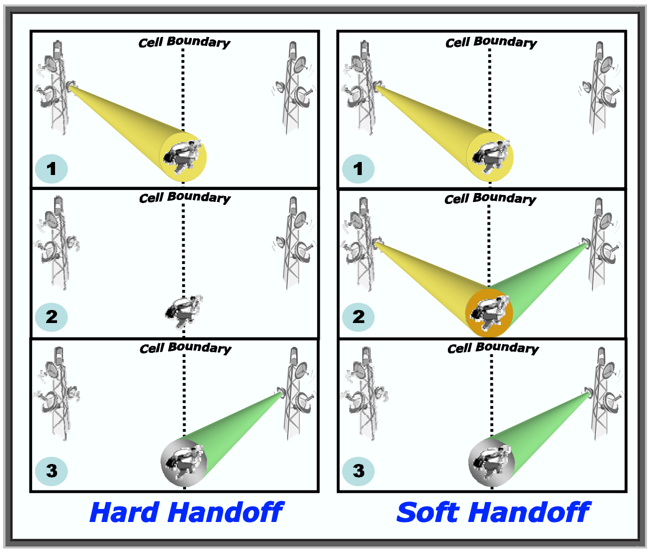

Figure 2: Handoff methods

2.2 Hard Handoff

In the case of hard handoff, MS requests a handoff to the neighboring BS immediately after exceeding the handoff threshold L_DROP. The moving MS does not maintain 2 traffic links in the handoff region. The handoff-add threshold can be thought of as the "largest" distance between a MS and a BS such that the MS can reliably transmit information through the given BS. The handoff-drop threshold is the distance where the MS cannot communicate with the servicing BS any more. In general, the system administrator determines L_ADD and L_DROP to optimize wireless channel utilization. Each serving BS broadcasts this information.

We propose a new handoff scheme to reduce the traffic of key updating during a handoff call. In the revised handoff scheme, two links are maintained during the handoff for the data transmission while the key update is only performed after completing the handoff. That is, the key updating does not occur when a call enters the handoff region. The connection to the new BS is just established without a key rekeying to prepare for the new connection.

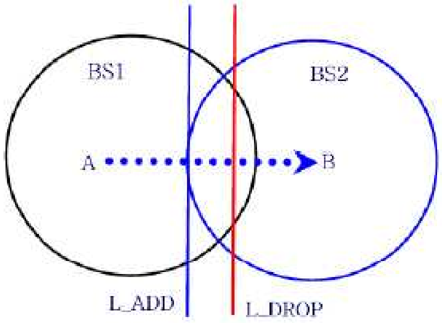

Figure 3: An example of L_DROP and L_ADD

We can reduce the traffic of key update in handoff region. This is a variation of the soft handoff scheme.

3 Location Tracking

We explain briefly about measuring the location of user in Code Division Multiple Access (CDMA) cellular system [25]. CDMA2000 [5] is synchronized with the Universal Coordinated Time (UCT). The forward link transmission timing of all CDMA2000 base stations worldwide is synchronized within a few microseconds. Base station synchronization can be achieved through several techniques including self-synchronization, radio beep, or through satellite-based systems such as GPS, Galileo, or GLONASS. Reverse link timing is based on the received timing derived from the first multipath component used by the terminal.

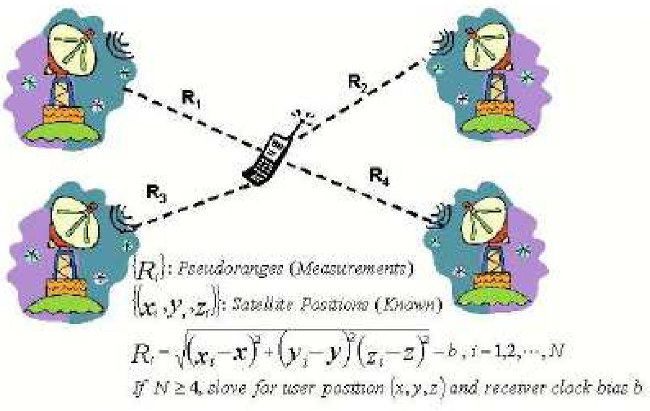
R1
R2
R3
R4
{Ri}: Pseudoranges (Measurements)
{(xi, yi, zi)}: Satellite Positions (Known)
$R_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 (z_i - z)^2} - b, i = 1,2,\cdots,N$
If N ≥ 4, slove for user position (x, y, z) and receiver clock bias b

Figure 4: The principle of location tracking

The most widely known position location system is the Global Positioning System (GPS). The GPS is a satellite-based psuedo-ranging position location system that provides geolocation of user's with GPS receivers. The idea behind GPS is that one's position (x,y,z) can be determined with the distance values from three different known positions by the triangulation method. The distance is measured in terms of delay, where an accurate clock at the receiver measures the time delay between the signal leaving the satellite and arriving at the receiver. Four simultaneous delay measurements from four satellites are required to solve three unknowns and the user's clock offset as shown in Figure 4. Some proposals for positioning, using one or two satellites, were presented in [11, 16] based on recently proposed mobile satellite systems.

4 Pre-positioned Secret Sharing (PSS)

We propose to use secret sharing techniques for the construction of the key trees. Secret sharing methods have been used for various security applications requiring users to share keys. We use the Pre-positioned Secret Sharing (PSS) scheme described in [19, 20]. We already show in the previous works [7, 8] that PSS based scheme is comparable to the Tree-based schemes [13, 31] in the respect of communications cost, rekeying time cost, and memory cost in the the wired network.

Shamir's secret sharing scheme [21] is a threshold scheme based on polynomial interpolation. It allows a dealer D to distribute a secret value s to n players, such that at least players are required to reconstruct the secret. The protocol is information theoretically secure, i.e., any fewer than t players cannot gain any information about the secret by themselves.

Let's see how we can design an (n, t) secret sharing scheme. To make the presentation easy to understand, let's start with the design of an $(n, 2)$ scheme.

Let's say we want to share a secret s among n parties. We use some basic geometry as shown in Figure 5. Select the point $(0, s)$ on the Y axis that corresponds to the secret. Now, randomly draw a line that goes through this point. Pick n points on that line: $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$. Each point that is picked represents a share. We claim that these n shares constitute an $(n, 2)$ sharing of s. Now we need to show that this scheme satisfies both the availability and confidentiality properties.

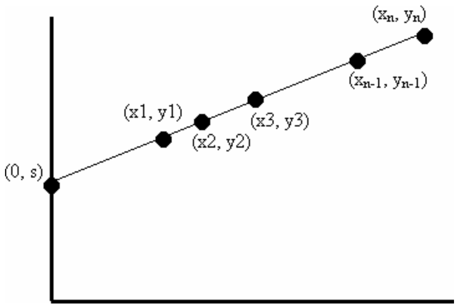

Figure 5: (n,2) secret sharing scheme

To show availability, we need to prove that two parties can recover the secret. Two parties have two shares; that is two points. Given these two points, how can we recover the secret? Well, we know that two points determine a line, so we can figure out the line that goes through both points. Once we know the line, we know the intersection of the line with the y axis. Then, we get the secret. So, it only takes us two points (shares) to make the secret available.

Now we will move on confidentiality. We need to show that one share does not disclose any information about the secret. There are infinite possible lines that go through this point, and these lines intersect with the y-axis at different points, all of which yield different "secrets". In fact, given any possible secret, we can draw a line that goes through the secret and the given share. This means that with one point, no information about the secret is exposed.

Using the same idea, can we design an $(n, 3)$ secret sharing scheme? Note that the key point in the $(n, 2)$ scheme is that a line is determined by two points, but not by 1. Now we need a curve that is determined by three points, but not 2. This curve happens to correspond to a quadratic function $y = a_2 * x_2 + a_1 * x + a_0$. Again, we find the point on the y-axis that corresponds to the secret, then we randomly select a curve corresponding to a quadratic function that goes through the point. Finally, we select n points on that curve as n shares to n parties as shown in Figure 6. Using a similar proof as in the $(n, 2)$ case, we can show that this is actually an $(n, 3)$ scheme that satisfies both availability and confidentiality [32].

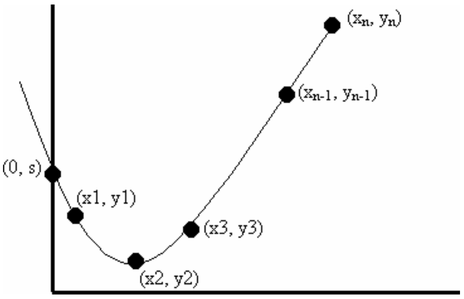

Figure 6: (n,3) secret sharing scheme

To generalize the scheme even further, we have a construction of an (n, t) secret sharing scheme. Now we use the curve that corresponds to a $(t-1)$ degree polynomial:

$$f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1} \mod (q). \tag{1}$$

To reconstruct the secret from each subset of t shares out of n shares, we use interpolation property and Lagrange interpolation. Given distinct t pairs of $(i, f(i))$, there is a unique polynomial $f(x)$ of degree $t-1$, passing through all the points. This polynomial can be effectively computed from the pairs $(i, f(i))$. Without loss of generality we will mark this subset:$f(1), \cdots, f(t)$. We use Lagrange interpolation to find the unique polynomial $f(x)$ such that $degree f(x) < t$ and $f(j) = share_j(s)$ for $j = 1, 2, \cdots, t$, where $share_j(s) = (x_i, f(x_i))$, $i = 1, 2, \cdots, n$.

$$f(x) = \sum_{j=1}^{t} f(x_j) \times L_j(x), L_j(x) = \prod_{i \neq j, 1 \leq i \leq t} \frac{(x - x_i)}{(x_j - x_i)}, \tag{2}$$

where, $L_i(x)$ is the Lagrange polynomial which has value 1 at x_i, and 0 at every other x_j. Then we can reconstruct the secret to be $f(0)$.

PSS uses a polynomial of order $(m-1)$ to generate shares. The shares will be used to generate the keys for the key tree. PSS is an interpolating scheme based on polynomial interpolation like Shamir's secret sharing scheme [21]. An $(m-1)$-degree polynomial over the finite field $GF(q)$:

$$F(x) = a_0 + a_1 x + \cdots + a_{m-1} x^{m-1} \mod (q), \tag{3}$$

is constructed such that the coefficient a_0 is the secret and all other coefficients are random elements in the field. Each of the n shares is a point (x_i, y_i) on the curve defined by the polynomial, where x_i is not equal to 0. Given any m shares, the polynomial is determined uniquely and hence the secret a_0 can be computed. However, given $m-1$ or fewer shares, the secret can be any element in the field. Therefore, PSS is a perfect secret sharing scheme. PSS uses a tree structure, which is composed of user nodes, subgroup-manager nodes, and the group-manager node in a bottom-up order. In the PSS, $(m-1)$ shares are assigned to each node while the m^{th} share is broadcasted as publication information. The $(m-1)$ shares of a node, which are secret, are referred to as the prepositioned shares, while the broadcast share, is referred to as the activation share (AS). In PSS, the AS helps determine the symmetric keys for each node. Once a node obtains the AS, the original polynomial of order m can be reconstructed and hence the keys can be recovered, using the AS along with the private $(m-1)$ shares owned by the node.

5 Group Key Management

We design a key management tree such that the key tree matches the network topology. We localize the delivery of rekeying messages to small regions of network by transmitting the key update messages only to the users who need them. This lessens the amount of traffic in wireless and wired intervals.

We explain the group key management operations, join, leave and handoff, through the example as shown in Figure 7 and Figure 8. In our scheme, each node has (n-1) shares if the secret is generated by n^{th} order polynomials. The shares are used to generate the keys for the key tree when each node receives a share, AS.

For each join, leave, and handoff, the shares will be changed to prevent the joining user from accessing past/future communications. After each join or leave, a new secure group is formed. The key server has to update the group's key graph by replacing the keys of some existing k-nodes, deleting some k-nodes and adding some k-nodes. Only one activating share is multicast by the key server, and it is used together with the pre-positioned information to generate three simultaneous keys.

In this example, 1 Group Manager (GM), 2 Subgroup Managers (SGM) and 6 users are considered. In Handoff operations, a 2 inter-BS handoff scheme is used for simplicity even though there are many handoff schemes [25].

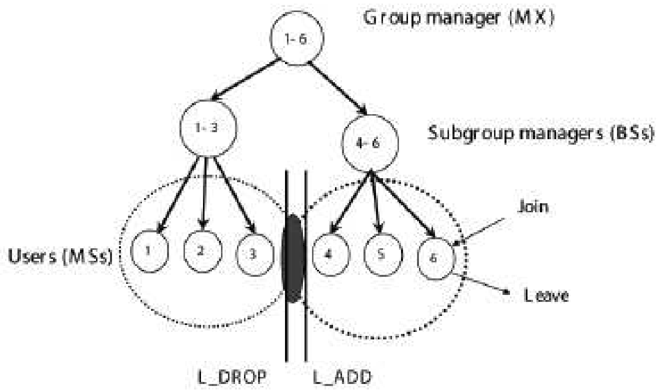

Figure 7: Hierarchical tree for join/leave

5.1 Joining a Group via BS1

For example, as shown in Figure 7, suppose user 6 wants to join the secure group. To prevent the joining user from accessing past communications, all keys along the path from the joining point to the root node need to be changed.

User 6 sends a join request message to the key server. After granting the new user, the key server associates s_6 with the new member and creates a new node and a new set node. The key server attaches the set node to the existing joining point. After changing s_{1-5} to s_{1-6} and s_{4-5} to s_{4-6}, the key server constructs the following two messages:

1) AS,$\{s_{1-6}\}_{k1-5}$, $\{s_{4-6}\}_{k4-5}$;

2) AS,$\{s_{1-6}, s_{4-6}\}_{k1-6}$.

Where AS is the activating share, the fresh keys k_{1-5}, k_{4-5} and k_6 are obtained by AS and the sets s_{1-5}, s_{4-5}, and s_6, respectively. The key server multicast the first message to the existing members, through $1-5$, while it unicast the second to the new member, 6. The members construct the new set of group keys, k'_{1-6}, when the new AS is multicast with the encrypted content.

5.2 Leaving a Group via BS1

Now suppose user 6 wants to leave the secure group, as shown in Figure 7. To keep the leaving user from accessing future communications, all keys along the path from the leaving point to the root node need to be changed.

User 6 sends a leaving request message to the key server. After granting the leaving user, the key server deletes the member node and the set node from the key tree. The key server replaces s_{4-6} by s_{4-5} and s_{1-6} by s_{1-5}. Then it constructs the following messages and multicast to the remaining members:

1) $\{s_{1-5}\}_{k1-3}, \{s_{1-5}\}_{k4-5}$;

2) $\{s_{4-5}\}_{k4}$, $\{s_{4-5}\}_{k5}$;

3) AS.

5.3 Handoff

As shown in Figure 8, user 4 is moving from BS2 to BS1 while the user is in the group service. The serving subgroup manager, BS2, requests a new connection to the neighboring BS, BS1, when the moving user exceeds the handoff add threshold, *L_ADD*. The key server associates s_4 with the new member of BS1, and creates a temporary node and a new set node. These sets are used within the handoff area. The key server attaches the set node to the existing joining point. After changing s_{1-3} to s_{1-4}, it constructs the following two messages:

1) AS, $\{s_{1-4}\}_{k1-3}$;

2) AS, $\{s_{1-4}\}_{k1-6}$.

The key server multicasts the first message to the existing member of BS1 while it unicasts the second message to the handoff member. Thus the handoff user keeps two links until it exceeds the handoff drop threshold, *L_DROP*. Immediately after the handoff user exceeds the *L_DROP*, the key server performs the leave procedure for BS2 and the add one for BS1.

The key server deletes the member node, here 4, and the set node from the key tree. The key server replaces s_{4-6} by s_{5-6}. Then it constructs the following messages and multicasts to the remaining members:

1) $\{s_{1-6}\}_{k1-4}$, $\{s_{1-6}\}_{k4-5}$;

2) $\{s_{4-5}\}_{k4}$, $\{s_{4-5}\}_{k5}$;

3) AS.

In the case of hard handoff, the leave and join operations are taken immediately after the moving user exceeds the boundary of the serving BS. That is, we can consider the hard handoff user as a leaving and a joining user to the group service. In this case, the handoff user does not keep two links in the handoff region. This is the main difference between the soft handoff and the hard handoff operations.

Neither handoff schemes are practical for cellular networks with frequent handoffs because the extra communication cost is too high if the system does not limit the number of group members. Thus the system manager uses a resource management scheme, CAC function, in real system. We describe a simple CAC function in section 6.

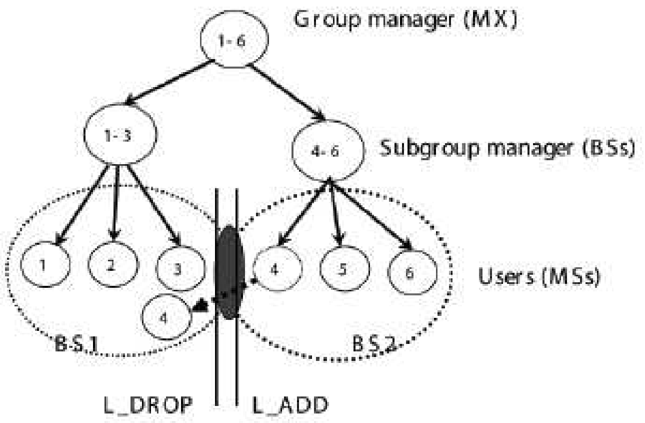

Figure 8: Hierarchical tree for handoff

6 Simulations and Results

6.1 Comparison of PSS and LKH

First, three measures are used to compare Tree-based schemes (LKH) [31] and PSS [7]: Storage cost, communication cost and computational cost where both schemes use the logical key heirarchy. The observations are summarized in the following tables. The group key tree is assumed full and balanced. The height h of the tree is the length of the longest directed path in the tree, and the degree d of the tree is the maximum number of incoming edges of a node in the tree.

Table 1: Comparison of LKH and PSS schemes: Storage cost

	LKH	PSS
# of keys held by server	$dn/(n-1)$	-
# of keys held by each member	h	-
# of share sets held by server	-	$dn/(n-1)$
# of share sets held by each member	-	h

The number of encryptions and decryptions required by join/leave operations are the same in both schemes. In the PSS scheme, however, neither the server nor the members need to store the node keys generated after each rekeying. They can be deleted as soon as they are used in the decryption process. The sets (both the group set and the auxiliary ones), however, need to be kept until they are replaced. There is a 1-1 correspondence between the number of keys generated for each member and the number of sets held by each member.

The size of the messages sent on join/leave operations are the same in both schemes. An additional communication cost in the PSS scheme for join/leave operations is the delivery of the activating share. The two schemes have different requirements in periodic rekeying. The communication cost for the PSS scheme is the delivery of the activating share and the communication cost for the LKH scheme is the delivery of d encrypted messages.

Table 2: Comparison of LKH and PSS schemes: Communication cost

	LKH	PSS
Join	$O(log_d(n))$	$O(log_d(n))$ and $O(1)$
Leave	$O(dlog_d(n))$	$O(dlog_d(n))$ and $O(1)$
Periodic rekeying	$O(d)$	$O(1)$

Table 3: LKH computation cost

	Server	Requesting member	Non-requesting member
Join	$2(h-1)$	$h-1$	$d/(d-1)$
Leave	0	$d/(d-1)$	$d(h-1)$
Periodic	d	1	

Table 4: PSS computation cost

	Server	Requesting member	Non-requesting member
Join	$2(h-1)$	$h-1$	$d/(d-1)$
Leave	$d(h-1)$	0	$d/(d-1)$
Periodic	0	0	

An additional computational cost in the PSS scheme for join/leave operations is the processing needed for the construction of the polynomials. There is a 1-1 correspondence between the number of polynomials constructed by the server and the number of encryptions performed by the server. There is also a 1-1 correspondence between the number of polynomials constructed by each member and the number of decryptions performed by each member. The two schemes have different computational requirements to recover the group key in periodic rekeying. The PSS scheme needs one polynomial construction for the server and one polynomial construction for each member whereas the LKH scheme needs d encryptions for the server and one decryption for each member.

6.2 Simulation Parameters

Now we test the group key management scheme based on the pre-positioned secret sharing in the wireless cellular network. We employ a wireless cellular network that consists of 16 concatenated cells with 1 Mobile switching

Table 5: Polynomial construction cost

	Server	Requesting member	Non-requesting member
Join	$2(h-1)$	$h-1$	$d/(d-1)$
Leave	$d(h-1)$	0	$d/(d-1)$
Periodic	1	1	

eXchanger (MX). We use 4 mobility models: 0 ∼ 1 km/hr for walking, 2 ∼ 5 km/hr for running, 6 ∼ 25 km/hr for low speed vehicle, and 26 ∼ 100 km/hr for high speed vehicle. The Poisson distribution with rate λ is used to model the number of calls occurring within a given time interval where λ is the shape parameter which indicates the average number of events in the given time interval. Exponential distribution with mean $1/\mu$ is used for the call duration. The close connection between the Poisson arrival process and the exponential interarrival time can be exploited immediately in properties of the exponential service time distribution. Table VI shows the range of values and the constants for the parameters.

In a cellular system, a call originated in a cell gets a channel and holds it until that call is completed in the cell or the MS moves out of the cell. The channel holding time is either the call duration time or the time for which MS resides in the cell. This is a function of parameters such as the cell radius R(km), the MS speed V(km/hr), the direction of MS, etc.

Table 6: Simulation parameters

Parameter	Value
# of MX	1
# of BS	16
# of MS	Up to 100 per BS
Call generation	Poisson with λ (calls/sec)
Call duration	Exponential with $1/\mu$ (1/sec)
User mobility	0-1 km/h (walking) 2-5 km/h (running) 6-25 km/h (low speed vehicle) 26-100 km/h (high speed vehicle)
Cell radius	1Km
Service	Voice, Data, Video
L_ADD	30% of BS coverage area
L_DROP	Boundary of BS

Including the handoff users and the new users, each BS can accommodate up to 100 group service users. Users are uniformly distributed in each BS. The CAC function, which is located in BS, counts the number of users to decide whether to accept new users or handoff users. We reserve some channels, here 30%, to give a priority to handoff users.

6.3 Key Update Costs in Wireless and Wireline Intervals

We set parameters to measure the number of transactions in wireless and wireline intervals such that μ=1/60 (/sec), λ=100 (calls/sec), V=50 (km/h), R=2(km), and simulation time=5 minute. The cost represents the key updates transactions. That is, a new call arrival and a call termination mean 1 key update respectively. The wireless cost and the wireline cost of the location matching trees (our scheme) and the Logical tree are shown for different quantities of participating BSs. We observed that the location matching trees have both smaller wireless cost and smaller wireline costs than the logical trees when the number of BSs is equal or greater than 2, and the advantages of the matching trees are more significant when the system contains more BSs. In this system, the communication cost of the matching trees can save as low as 20% of the communication cost of the independent trees as shown in Figures 9 and 10.

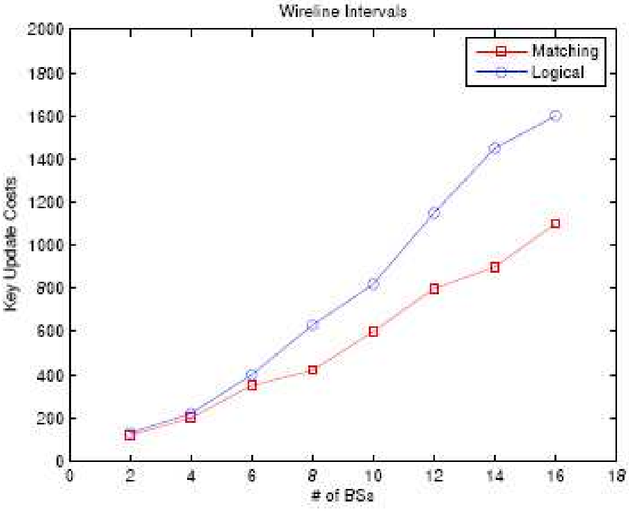

Figure 9: Key update costs in wireline intervals

6.4 Handoff Cost

We set parameters to measure the number of handoff attempts for each user group such that μ=1/180 (/sec), λ=20 (calls/sec), R=2(km), and simulation time=10 minutes. We observe that each user group undergoes 3-8 handoffs during the call duration. Moving users onto the key tree causes some extra key management resources even though they are still in service because of handoff. To take care of frequent handoff between wireless access networks, we proposed a new revised handoff schemes. This new handoff scheme can reduce some key update costs in wireless and wireline intervals because it only updates the keys after completion handoff.

A call can have 3 key transactions during the call duration: call generation, handoffs, and call termination. A handoff call requires 2 key update transactions: (1)

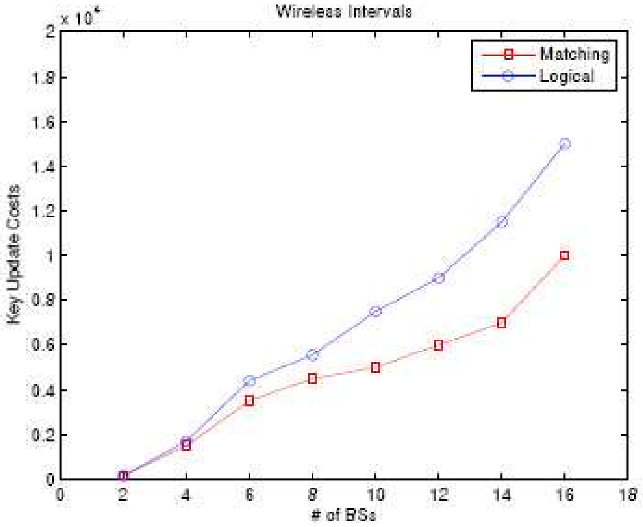

Figure 10: Key update costs in wireless intervals

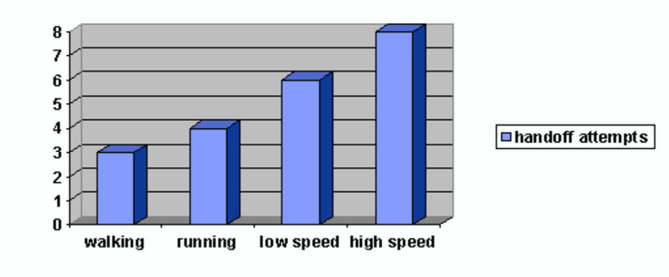

Figure 11: Handoff attempts for each user type

adding a new channel when a call enters handoff region and (2) deleting a serving channel after completing handoff.

Thus the number of key transactions during call duration, N, equals to

$$\begin{aligned} N = 1 \times (callgeneration) + 1 \times (calltermination) \\ + 2 \times (\#ofHandoff). \end{aligned} \tag{4}$$

We run our simulation 10 times and calculate the average handoff attempts per user according to the mobility models. Each call has 3 $\sim$ 8 handoffs during the call service time. We show the result in Figure 11.

In Figures 12 and 13, we plot the number of handoff attempts as a function of the number of new calls. The number of handoff attempts increases linearly as the number of new calls increases. With the the hard handoff schemes, the number of handoffs per call is reduced by about 20% comparing to the results of Figure 12. It's very expected result because the hard handoff scheme requires less key update transactions in handoff region.

Now we find that the handoff part can be the largest inefficiency in wireless cellular networks. To reduce the number of handoffs, we can increase the radius of cell. However, as the radius of cell increases, the system capacity decreases. That is, the total number of users in a system will be decreased if the radius of cell is increased. So we need an alternative method.

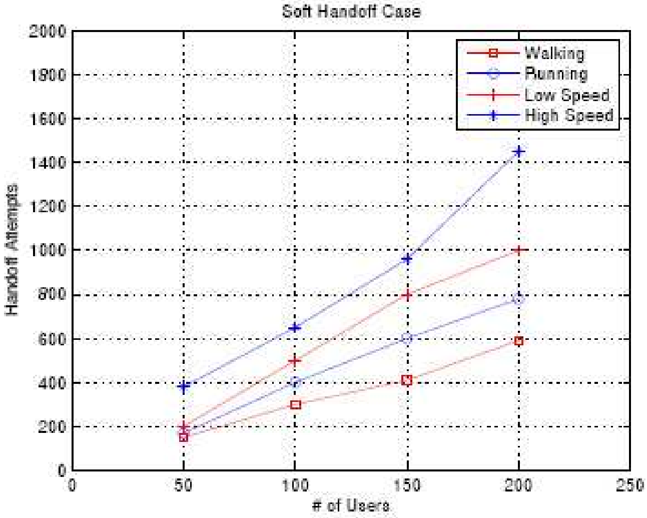

Figure 12: The number of handoff attempts in soft handoff case

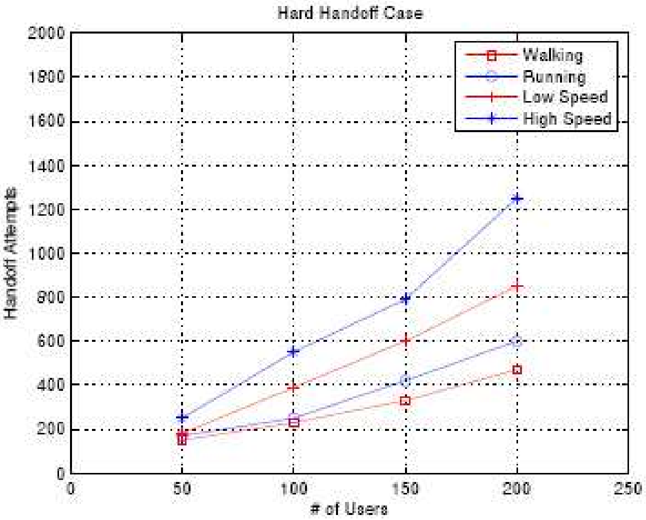

Figure 13: The number of handoff attempts in hard handoff case

We don't take into account a call admission control (CAC) so far. That is, we don't restrict the number of users for each BS. The CAC function determines whether to accept a new call and a handoff call. With the CAC and the revised handoff schemes, the number of handoffs per call is reduced by almost 20% comparing to the results of Figure 12 until the threshold of CAC, here 100 users per BS. After the threshold, the handoff attempts stay to a certain level since the CAC limited the number of new calls. In Figure 14, we plot the number of the handoff attempts as a function of the number of new calls with a CAC and a revised handoff scheme. We find that the number of handoff attempts don't increase after 100 users.

Because of the CAC, only 100 users are accepted in each BS.

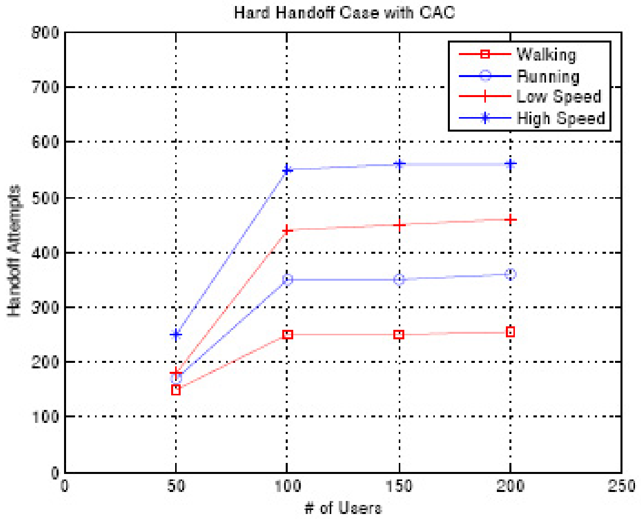

Figure 14: The number of handoff attempts vs. the number of new calls with a CAC

7 Conclusion

In order to provide access control to the multicast communication, the data is typically encrypted using a key that is shared by all legitimate group members. In secure and reliable wireless group communication, managing the conference key is critical problem. We focus on the group key management suitable for wireless environment. We design a key management tree such that the neighbors on the key tree are also physical neighbors on the network. The group key management scheme uses the pre-positioned secret sharing scheme. By tracking the user location, we localize the delivery of rekeying messages to the right nodes that need them. This lessens the amount of traffic in the cellular network. We find that each call undergoes an average of 3 $\sim$ 8 handoffs during a call duration according to the user mobility models. Thus the largest burden of the key updating comes from the handoffs. We propose a new handoff scheme to minimize the key updating transactions. This new handoff scheme reduces one of the two key update transactions in the handoff region - adding a new channel when a call enters handoff region. In the handoff area, only a new traffic channel is added to minimize the interruption time of the data transmission. With the revised handoff scheme, the number of handoff per call is reduced by almost 20% compared to that of the soft handoff. A simple CAC function is also used to maintain key updating transactions to a certain level. By restricting the number of users according to the available bandwidth, the CAC function makes the cellular network reliable.

References

[1] C. Becker and U. Wille, “Communication complexity of group key distribution,” in *Proceedings of the 5th ACM conference on computer and communications security*, Nov. 1998.

[2] D. BT *et al.*, “Secure group communications for wireless networks,” in *Proceedings of the IEEE MILCOM 2001*, Oct. 2001.

[3] M. Burmester and Y. Desmedt, “secure and efficient conference key distribution system,” in *Proceedings of the EUROCRYPT’94*, pp. 275-286, 1994.

[4] R. Canetti, J. Garayt, G. Itkid, D. Micciancios, M. Naore, and B. Pinkasll, “Multicast security: A taxonomy and some efficient constructions,” in *Proceedings of the IEEE INFOCOM ’99*, vol. 2, pp. 708-716, Mar. 1999.

[5] CDMA Technology Group, 2001. (http://www.cdg.org/technology/3g/)

[6] L. Dondeti, S. Mukherjee, and A. Samal, “Comparison of hierarchical key distribution schemes,” in *Proceedings of the IEEE Globecom Global Internet Symposium*, Dec. 1999.

[7] A. M. Eskicioglu and M. R. Eskicioglu, “Multicast security using key graphs and secret sharing,” in *Proceedings of the Joint International Conference on Wireless LANs and Home Networks ICWLHN 2002 and Networking ICN 2002*, pp. 228-241, Aug. 2002.

[8] A. M. Eskicioglu, S. Dexter, and E. J. Delp, “Protection of multicast scalable video by secret sharing: Simulation results,” in *Proceedings of the IEEE MILCOM 2003*, Jan. 2003.

[9] Y. Kim, A. Perrig, and G. Tsudik, “Communication-efficient group key agreement,” in *Proceedings of the 17th International Information Security Conference IFIP SEC’01*, 2001.

[10] S. Kumar, and P. Radoslavov, “The MASC/BGMP architecture for inter-domain multicast routing,” in *Proceedings of the ACM SIGCOMM’98*, pp. 93-104, Aug. 1998.

[11] N. Levanon, “Quick positioning determination using 1 or 2 LEO satellites,” *IEEE Transactions On Aerospace and Electronic systems*, vol. 34, July 1998.

[12] D. A. McGrew and A. T. Sherman, *Key Establishment in Large Dynamic Groups Using One-Way Function Trees*, May 1998.

[13] S. Mitta, “Iolus: A framework for the scalable secure multicasting,” in *Proceedings of the ACM SIGCOMM’97*, pp. 277-288, Sep. 1997.

[14] R. Molva, and A. Pannetrat, “Scalable multicast security in dynamic groups,” in *Proceedings of the 6th ACM conference on Computer and communications security*, pp. 101-112, 1999.

[15] R. Molva, and A. Pannetrat, “Scalable multicast security with dynamic recipient groups,” *ACM Transactions on Information and System Security*, vol. 3, pp. 136-160, Aug. 2000.

[16] K. Narenthiran, R. Tafazolli, and B. G. Evans, "Simple positioning method for location tracking in mobile satellite communications," in *Proceedings of the 18th AIAA International Communication Satellite Systems Conference*, Apr. 2000.

[17] A. Perrig, D. Song, and J. D. Tygar, "Elk: A new protocol for efficient large-group key distribution," in *Proceedings of the IEEE Symposium on Security and Privacy*, May 2001.

[18] S. Rafaeli, and D. Hutchison, "A survey of key management for secure group communication," *ACM Computing Survey*, vol. 35, pp. 309-329, Sep. 2003.

[19] G. J. Simmons, "How to (really) share a secret," in *Proceedings of the Advances in Cryptology - CRYPTO'88*, pp. 390-448, 1990.

[20] G. J. Simmons, "Prepositioned shared secret and/or shared control schemes," in *Proceedings of the Advances in Cryptology - EUROCRYPT'89, Springer-Verlag*, pp. 436-467, 1990.

[21] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 11, pp. 612-613, Nov. 1979.

[22] C. Shields, and J. J. Garcia-Luna-Aceves, "KHIP a scalable protocol for secure multicast routing," in *Proceedings of the ACM SIGCOMM'99*, pp. 53-63, Aug, 1999.

[23] R. Song and L. Korba, "Cryptanalysis of scalable multicast security protocol," *IEEE Communications Letters*, vol. 7, pp. 561-563, Nov. 2003.

[24] Y. Sun, W. Trappe, and K. J. R. Liu, "An efficient key management scheme for secure wireless multicast," in *Proceedings of the IEEE International Conference on Communication (ICC'02)*, pp. 1236-1240, 2002.

[25] *TIA/EIA Interim Standard (IS-95), Mobile Station - Base Station Compatibility Standards For Dual Mode Wideband Spread Spectrum Cellular System*, TIA/EIA Standard, July 1993.

[26] H. Um and E. J. Delp, "A secure group key management scheme for wireless cellular networks," in *Proceedings of the International Conference on Information Technology: New Generations (ITNG'06)*, Apr. 2006.

[27] H. Urn and E. J. Delp, "A new secure group key management scheme for multicast over wireless cellular networks," in *Proceedings of the International Performance, Computing, and Communication Conference (IPCCC'06)*, Apr. 2006.

[28] H. Urn and E. J. Delp, "Selective video encryption of a distributed coded bitstream using LDPC codes," in *Proceedings of IS-T/SPIE Symposium on Electronic Imaging on Security, Steganography, and Watermarking of Multimedia Contents VIII (EI'06)*, Jan. 2006.

[29] D. Wallner, E. Harder, and R. Agee, *Key management for multicast: Issues and Architecture*, RFC2627, 1999.

[30] M. Waldvogel, G. Caronni, D. Sun, N. Weiler, and B. Plattner, "The versakey framework: Versatile group key management," *IEEE Journal on Selected Areas in Communications (Special Issue on Middleware)*, vol. 17, pp. 1614-1631, Aug. 1999.

[31] C. Wong, M.Gouda, and S. Lam, "Secure group communication using key graphs," *IEEE/ACM Transaction on Networking*, vol. 8, pp. 16-30, Feb. 2000.

[32] L. Zhou and M. Frei, 2000. (http://www.cs.cornell.edu/courses/cs513/2000sp/secretsharing.html)

Hwayoung Um received the B.S. and M.S. degrees in Department of Electrical and Computer Engineering at Hanyang University in 1991 and 1993, respectively. He received Ph.D. degree from the School of Electrical and Computer Engineering at Purdue University, West Lafayette, Indiana, USA, in August 2006. Before he came to Purdue, he worked at Electronics and Telecommunication Research Institute (ETRI), Korea from 1993 to 2000. After joining to Purdue, he was a graduate research assistant at Video and Image Processing Laboratory (VIPER Lab.) working with Professor Edward J. Delp. Since August 2006, he has been with SAMSUNG NETWORKS Inc., Seoul, South Korea as a general manager. His research interests include image and video compression, multimedia security, wireless networks, multimedia systems, communication and information theory.

Edward J. Delp was born born in Cincinnati, OH. He received the B.S.E.E. (cum laude) and M.S. degrees from the University of Cincinnati and the Ph.D. degree from Purdue University, West Lafayette, IN. In May 2002, he received an Honorary Doctor of Technology from Tampere University of Technology, Tampere, Finland. From 1980 to 1984, he was with the Department of Electrical and Computer Engineering, The University of Michigan, Ann Arbor. Since August 1984, he has been with the School of Electrical and Computer Engineering and the School of Biomedical Engineering, Purdue University. In 2002, he received a Chaired Professorship and currently is The Silicon Valley Professor of Electrical and Computer Engineering and Professor of Biomedical Engineering. His research interests include image and video compression, multimedia security, medical imaging, multimedia systems, communication, and information theory.

Dr. Delp is a Fellow of the IEEE, a Fellow of the SPIE, the Society for Imaging Science and Technology (IS&T), and the American Institute of Medical and Biological Engineering. He is Co-Chair of the SPIE/IS&T Conference on Security, Steganography, and Watermarking of Multimedia Contents that has been held since January 1999. He was the Program Co-Chair of the IEEE

International Conference on Image Processing that was held in Barcelona in 2003. In 2000, he was selected a Distinguished Lecturer of the IEEE Signal Processing Society.

He received the Honeywell Award in 1990, the D. D. Ewing Award in 1992 and the Wilfred Hesselberth Award in 2004 all for excellence in teaching. In 2001 he received the Raymond C. Bowman Award for fostering education in imaging science from the Society for Imaging Science and Technology (IS&T). In 2004 he received the Technical Achievement Award from the IEEE Signal Processing Society. In 2002 and 2006, he was awarded Nokia Fellowships for his work in video processing and multimedia security.