Topology-aware Key Management Schemes for Wireless Multicast

Yan Sun*, Wade Trappe†, and K. J. Ray Liu*
*Department of Electrical and Computer Engineering, University of Maryland, College Park
Email: ysun, kjrliu@glue.umd.edu
†Wireless Information Network Laboratory (WINLAB), Rutgers University
Email: trappe@winlab.rutgers.edu.

***Abstract*— In secure multicast applications, key management must be employed to provide access control to the multicast content. In wireless networks, where the error rate is high and the bandwidth is limited, the design of key management schemes should place emphasis on reducing the communication burden associated with key updating and improving the reliability of key distribution. The topology-matching key management (TMKM) scheme has been proposed to reduce the communication burden associated with rekeying by matching the key tree to the network topology and localizing the transmission of rekeying messages. This scheme, however, is only suitable for homogeneous networks where mobiles users are uniformly distributed in the service area and experience similar delay and transmission error rates. In this paper, we present an improved topology-aware key management scheme that is suitable for a large-scale cellular wireless network where the heterogeneity of the network is taken into consideration. The proposed scheme not only reduces the communication overhead, but also improves the reliability of the key distribution.**

I. Introduction

The rapid progress in multicast networking has led to the deployment of many multicast services, such as streaming stock quotes, video conferencing and communal gaming [1]. At the same time, there has been significant advancements in building a global wireless infrastructure that will free users from the confines of static communication networks. When wireless connections finally become ubiquitous, consumers will desire to have multicast applications running on their mobile devices. Before the wireless multicast market can be successful, *access control* mechanisms must be deployed in order to guarantee that only authorized users can access the multicast content.

Access control is achieved by encrypting the content using an encryption key, known as the session key (SK), which is shared by all legitimate group members. A trusted third party, known as the key distribution center (KDC), is responsible for generating and securely communicating key material to the group members. Since users may join and leave at any time, *rekeying messages* need to be sent to update keys in order to prevent the leaving user from accessing future communication and prevent the joining user from accessing previous communication. Rekeying messages must be delivered reliably because the loss of rekeying messages results in severe performance degradation [2]. Further, in real-time multicast applications the rekeying messages should be delivered in a timely manner so that users receive the rekeying messages before the new key takes effect. In wireless multicast scenarios, where the bandwidth is limited and the data typically experience a higher transmission error rate than in conventional environment, the key management design should place emphasis on improving the reliability of key distribution as well as reducing the communication burden associated with key updating, especially when there are many users and frequent additions to or deletions from the group membership.

In [3], a communication-efficient key management scheme, called topology-matching key management (TMKM), was introduced for secure wireless multicast. By matching the key tree to the network topology and localizing the transmission of rekeying messages, the TMKM scheme significantly reduces the communication cost of rekeying messages compared with the traditional schemes [2], [4], [5] that are independent of the network topology. This scheme, however, is suitable only for homogeneous networks where mobile users are uniformly distributed in the service area and experience similar delay and transmission error rates.

In this paper, we present an improved topology-aware key management scheme that is suitable for a large-scale cellular wireless network, where users do not have the same join/departure/mobility behavior and the network conditions may vary. Although focusing on the communication overhead reduction, we prove that the proposed method can also be used to improve the reliability of key distribution with minor modifications. In addition, we present a *unicast-assisted topology-matching key management* (uTMKM) scheme, which employs both unicast and multicast communication to deliver rekeying messages. The uTMKM scheme further improves the performance of the proposed method.

The rest of the paper is organized as follows. Section II introduces the concept of the TMKM and the uTMKM schemes. Section III defines the performance measure and formulates the optimization problem. Section IV describes a tree design procedure considering the heterogeneity of the network. The simulation results are presented in Section V, followed by the conclusion in Section VI.

II. Topology-Matching Key Management Tree

In this section, we introduce the idea of matching the key tree to the network topology, and outline a procedure to design topology-aware key management schemes.

A common class of multicast key management schemes employ a tree hierarchy for the maintenance of keying material [2] [4] [5], as depicted in Figure 1. Each user stores his private key u_i, the session key K_s, and a set of key encrypting

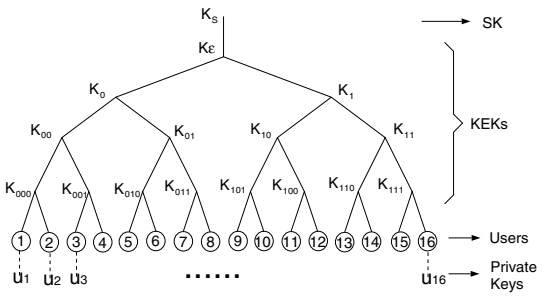

Fig. 1. A typical key management tree

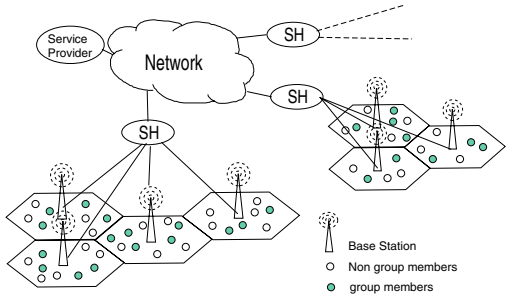

Fig. 2. A cellular wireless network model

keys (KEKs) on the path from himself to the root of the key tree. The session key (SK) is used to encrypt the multicast content. KEKs are a set of auxiliary keys, which are used solely for the purpose of updating the SK and other KEKs. A user's private key is only known by that user and the KDC. Since the member join operation can be achieved without transmitting rekeying messages [5], we shall only focus on member departure. It is observed that most rekeying messages are only useful to a subset of users. For example, when user 16 leaves the multicast service, all of the keys he possesses, $\{K_s, K_\epsilon, K_1, K_{11}, K_{111}\}$, should be updated. The message used to update K_{111} is only useful to user 15, the message used to update K_{11} is only useful to user 13,14,15, the message used to update K_1 is only useful to user $9, 10, \cdots, 15$, and the message used to update K_ϵ and K_s are useful to all users. Therefore, rekeying messages do not have to be sent to every user in the multicast group.

This observation motivates us to design a key management tree that matches the network topology in such a way that the neighbors on the key tree are also physical neighbors on the network. Particularly, the key tree is designed to match the cellular network topology depicted in Figure 2. This cellular network model, as proposed in [6], consists of mobile users, base stations (BS) and supervisor hosts (SH). SHs administer the BSs and handle most of the routing and protocol details for mobile users. The service provider, the SHs, and the BSs are connected through a wireline backbone, while the BSs and the mobile users are connected through wireless channels. The key management tree is designed in three steps:

- Step 1: Design a subtree for the users under each BS. These subtrees are called *user subtrees*.
- Step 2: Design subtrees which govern the key hierarchy between the BSs and the SHs. These subtrees shall be called *BS subtrees*.
- Step 3: Design a subtree which governs the key hierarchy between the SHs and the KDC. This subtree shall be called the *SH subtree*.

Figure 3 illustrates an example of a key tree that matches the network topology shown in Figure 2.

By delivering the rekeying messages only to the users who need them, we can take advantage of the fact that the key tree matches the network topology, and localize the delivery of rekeying messages to small regions of the network. This lessens the amount of traffic load as well as enhances the reliability of key distribution.

Similar to [3], we assume that the SHs and the BSs have the knowledge of whether the rekeying messages are useful for the users under them. Using this knowledge, which can be conveyed in the rekeying message header, it is possible to make only a subset of SHs or BSs transmit messages. That is, a SH will multicast a message to their BSs if the message is useful to at least one of its BSs, and a BS will multicast if the message is useful to at least one of its users.

When unicast connections between BSs and SHs are available and the number of BSs that need the messages, denoted by T_{bs}, is small, it is possible to reduce the amount of communication by using unicast channels as opposed to multicast channels. Multicast routing protocols do not have the ability to target message delivery to specific subsets of users. Therefore, when multicasting is employed, a rekeying message is sent to the entire group of BSs, regardless of whether or not all BSs need that message. Unicast, however, achieves an advantage since it allows for the rekeying message to be sent only to those BSs that need that message. Therefore, we propose a transmission strategy between SHs and BSs as:

- When $T_{bs} \leq T_{th}$, the SH sends rekeying messages to the BSs who need the messages using unicast.
- When $T_{bs} > T_{th}$, the SH multicast rekeying messages to all of its BSs.

The threshold T_{th} should be determined from the relative cost of the unicast communication compared to the multicast communication.

In the reminder of the paper, TMKM will represent the topology-matching key management scheme with the first transmission strategy and unicast-assisted TMKM (uTMKM) will represent the topology-matching key management scheme using both unicast and multicast communications. Traditional key management trees, such as those in [4], [5], are independent of the network topology, and we call them Topology Independent Key Management (TIKM) trees. When using a TIKM tree, the users are scattered all over the network, and therefore, it is not possible to localize the delivery of rekeying messages. The comparison amongst TIKM, TMKM and uTMKM schemes will be further discussed through simulations in Section V.

In the mobile environment, the user will subscribe to a multicast service under an initial host agent, and through the course of his service move to different cells and undergo *handoff* to different base stations. Since the TMKM and uTMKM trees depend on the network structure, the physical

location of a user affects the user's position on the key management tree. In this paper, we assume that the handoffs only occur between BSs under the same SH, and the efficient handoff scheme proposed in [3] is employed to handle user relocation on the key tree.

III. Performance Measure

A. Communication Overhead

As discussed in the previous section, rekeying messages are first multicast/unicast to BSs, then broadcast to mobile users. We first define several variables as follows.

- The *multicast-message-size* is defined as the size of the rekeying messages multicast from the l^{th} SH to its BSs, denoted by S_m^l.
- The *unicast-message-size* is defined as the size of the rekeying messages unicast from the l^{th} SH to its BSs, denoted by S_u^l.
- The *wireless-message-size* is defined as the size of the rekeying messages broadcast by the BSs to mobile users under the l^{th} SH, denoted by S_w^l.
- n_{sh} denotes the number of SHs in the system. Under the l^{th} SH, there are n_{bs}^l BSs, each of which has n_{user}^l users. n_{sh} and n_{bs}^l are assumed to be fixed during the service.

Then, the communication cost of the key management schemes is described by wireline cost C_{wire}, wireless cost $C_{wireless}$, and total cost C_T, as:

$$C_{wire} = \sum_{l=1}^{n_{sh}} \alpha_m^l E[S_m^l] + \alpha_u^l E[S_u^l], \tag{1}$$

$$C_{wireless} = \sum_{l=1}^{n_{sh}} \alpha_w^l E[S_w^l], \tag{2}$$

$$C_T = \gamma \cdot C_{wireless} + (1-\gamma) \cdot C_{wire}, \tag{3}$$

where $E[.]$ is the expectation over the statistics governing the user joining and leaving behavior. Here, $0 \le \gamma \le 1$ is the *wireless weight*, which represents the importance of the wireless cost. $\{\alpha_m^l\}$, $\{\alpha_u^l\}$ and $\{\alpha_m^l\}$ are sets of weight factors that describe the relative importance of the multicast-message-size, unicast-message-size, and wireless-message-size under the l^{th} SH, respectively. The threshold T_{th} in the uTMKM scheme shall be chosen as $\lfloor \frac{\alpha_m^l}{\alpha_u^l} \rfloor$. When the SHs administer areas with similar physical network structure and conditions, we approximate $\{\alpha_m^l\}$, $\{\alpha_u^l\}$ and $\{\alpha_m^l\}$ by 1.

B. Reliability of Key Distribution

The reliability of key distribution is critical for key management schemes in wireless networks where the transmission error rate is usually higher than that in conventional wireline environment. The loss of rekeying messages results in severe performance degradation [2]. If a user loses one key, he not only will be unable to access multicast content encrypted by this key, but also may not be able to acquire future keys from future rekeying messages.

We introduce three error probabilities for each SH: (1) the probability that one user cannot receive a rekeying message correctly from the BS, denoted by p_w^l; (2) the probability that one BS cannot receive a rekeying message correctly from the SH through multicast channel, denoted by p_m^l; and (3) the probability that one BS cannot receive a rekeying message correctly from the SH through the unicast channel, denoted by p_u^l, where $l = 1, 2, \cdots, n_{sh}$. Users' packet loss is assumed to be independent. We define the reliability measure P_s as the probability that all users receive rekeying messages correctly with the average multicast-message-size, unicast-message-size and wireless-messages-size, i.e.

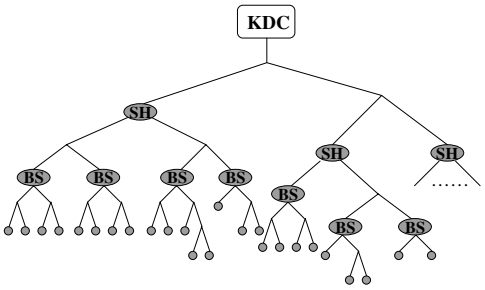

Fig. 3. A Topology Matching Key Management Tree

$$P_s = \prod_l (1-p_w^l)^{n_{user}^l E[S_w^l]} \cdot (1-p_m^l)^{n_{bs}^l E[S_m^l]} \cdot (1-p_u^l)^{E[S_u^l]},$$

and

$$\log P_s = \sum_l \left(\hat{\alpha}_w^l E[S_w^l] + \hat{\alpha}_m^l E[S_m^l] + \hat{\alpha}_u^l E[S_u^l] \right), \tag{4}$$

where $\hat{\alpha}_w^l = n_{user}^l \log(1-p_w^l)$, $\hat{\alpha}_m^l = n_{bs}^l \log(1-p_m^l)$, and $\hat{\alpha}_u^l = \log(1-p_u^l)$.

Comparing (4) with (1)-(3), it is seen that P_s is closely related with C_T. The techniques that reduce the communication cost can be easily extended to improve the reliability of the key distribution. Particularly, when choosing $\alpha_w^l = \hat{\alpha}_w^l/\gamma$, $\alpha_m^l = \hat{\alpha}_m^l/(1-\gamma)$, and $\alpha_u^l = \hat{\alpha}_u^l/(1-\gamma)$, the solution that minimizes the total communication cost in (3) would maximize P_s in (4). Thus, we only focus on reducing the communication overhead in the remainder of the paper.

IV. Optimization and Subtree Design

In this section, we formulate the optimization problem and describe the key tree design procedures.

When a user leaves the service, the keys that need to be updated are divided into three *categories*: (1) the keys on the user subtree, (2) the keys on the BS subtree, and (3) the keys on the SH subtree. We can prove that

$$\begin{aligned} C_T \;=\; & \sum_{l=1}^{n_{sh}} \sum_k p^l(k) G^l(k) A_1^l(k) + \sum_{l=1}^{n_{sh}} \sum_k p^l(k) G^l(k) A_2^l(k) \\ & + \sum_{l=1}^{n_{sh}} A_3^l \cdot \left(\sum_k p^l(k) G^l(k) \right), \end{aligned} \tag{5}$$

where $p^l(k)$ is the probability mass function of the number of users under the l^{th} SH, and $G^l(k)$ is the probability that one of the k users leaves from the l^{th} SH. Here, $A_1^l(k)$, $A_2^l(k)$ and A_3^l describe the communication cost due to updating keys in category 1, 2, and 3 respectively, when there are k users

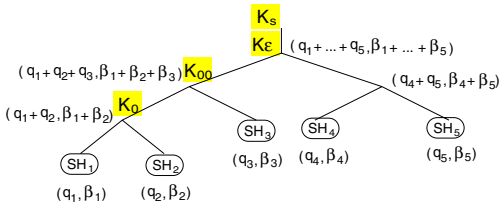

Fig. 4. An example of the SH subtree

under the l^{th} SH and one of them leaves. We can prove that the structure of the user-subtrees only affects $A_1^l(k)$, the structure of the BS-subtrees only affects $A_2^l(k)$, and the structure of the SH-subtrees only affects A_3^l.

Equation (5) indicates that the user subtrees, BS subtrees and SH subtree can be designed and optimized separately. Particularly, the user subtrees under the l^{th} SH should be designed to minimize $\sum_k p^l(k)G^l(k)A_1^l(k)$, the BS subtree under the l^{th} SH should be designed to minimize $\sum_k p^l(k)G^l(k)A_2^l(k)$, and the SH subtree should be designed to minimize $\sum_{l=1}^{n_{sh}} A_3^l \cdot \left(\sum_k p^l(k)G^l(k)\right)$. This is a desired property because optimizing the subtrees separately reduces the dimension of the search space for optimal tree parameters.

We use the ALX tree structure proposed in [3] to design user subtrees and BS subtrees. Since the design method is very similar to what has been presented in [3], we only discuss the design of the SH subtree in this paper.

In a typical cellular network, each SH administers a large area where both the user dynamics and the network conditions may differ significantly from the areas administered by other SHs. The ALX tree structure, which treats every leaf equally, is not suitable for building the SH subtree. Instead, the SH heterogeneity may be addressed by building a tree where the SHs have varying path lengths from the root to their leaf node.

The root of the SH subtree is the KDC, and the leaves are the SHs. The design goal is to minimize the third term in equation (5), which we denote by C_{sh} and is given by

$$C_{sh} = \sum_{l=1}^{n_{sh}} q_l \cdot A_3^l, \tag{6}$$

where $q_l = \sum_k p^l(k)G^l(k)$. Let β_l be the communication cost of transmitting one rekeying message to all the users under the l^{th} SH. Based on the definition of α_1^l and α_2^l in Section II, it is easy to show that $\beta_l = (1-\gamma)\alpha_m^l + \gamma n_{bs}^l \alpha_w^l$.

The value of A_3^l can be calculated directly from $\{\beta_l\}$. For example, if the SH subtree has the structure illustrated in Figure 4 and a user leaves the service from SH_1, then,

$$A_3^1 = 2(\beta_1+\beta_2)+2(\beta_1+\beta_2+\beta_3)+2(\beta_1+\beta_2+\beta_3+\beta_4+\beta_5).$$

The goal of SH subtree design is to find a tree structure that minimizes C_{sh} given β_l and q_l. However, it is very difficult to do so based on (6). Thus, we compute C_{sh} in a different way.

We shall assign a pair of positive numbers, called a *cost pair*, to each node on the tree as follows: the cost pair of the leaf node that represents the l^{th} SH is (q_l, β_l); the cost pairs of the intermediate nodes are the element-wise summation of their children nodes' cost pairs, as illustrated in Figure 4. The cost pairs of all intermediate nodes are represented by (x_m, y_m), where $m = 1, 2, \cdots, M$, and M is the total number of intermediate nodes on the tree. Then, we can prove that C_{sh} can also be calculated as $C_{sh} = n\sum_{m=1}^{M} x_m \cdot y_m$, where n is the degree of the SH-subtree. This new formulation leads to a tree construction method for $n = 2$ as:

1) Label all the leaf nodes together with their cost pairs, and mark them to be active nodes.
2) Choose two active nodes, (x_i, y_i) and (x_j, y_j), such that $(x_i+x_j)\cdot(y_i+y_j)$ is minimized among all possible pairs of active nodes. Mark those two nodes to be inactive and merge them to generate a new active node with the cost pair $(x_i + x_j, y_i + y_j)$.
3) Repeat step 2 until there is only one active node left.

This method, which we call greedy-SH subtree-design (GSHD) algorithm, can be easily extended to $n > 2$ cases. We can prove that the GSHD algorithm produces the optimal solution when $\beta_1 = \beta_2 = \cdots \beta_{n_{sh}}$, but it is not optimal in general. The performance of the GSHD algorithm will be demonstrated through simulations in Section V.

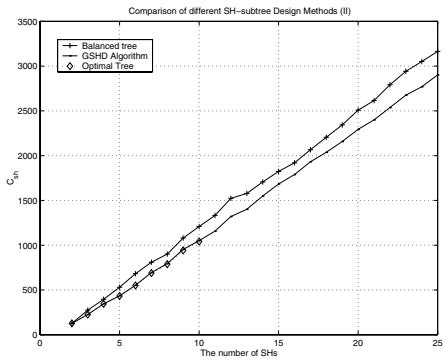

Fig. 5. Comparison of several SH subtree design methods

V. Simulation Results

We first compare the SH subtree generated using the proposed GSHD algorithm with the optimal tree obtained by exhaustive search, and with a balanced tree that treats each SH equally. In the simulation, half of the $\{\beta_l\}$ are randomly selected from $[1, 20]$, which represent rural areas, and the other half of $\{\beta_l\}$ are randomly selected from $[101, 120]$, which represent metropolitan areas. In addition, q_l is chosen to be proportional to β_l, where $l = 1, 2, \cdots, n_{sh}$, and $\{q_l\}$ are normalized so that $\sum q_l = 1$. In Figure 5, the communication cost due to updating keys on SH-subtrees, C_{sh}, is shown for different SH subtree design methods. Since exhaustive search is computationally very expensive, it is only done for 10 and fewer SHs. It is observed that the performance of the GSHD algorithm is very close to the optimum. Compared with the balanced tree, the GSHD algorithm reduces C_{sh} by up to 18%.

Figure 6 demonstrates the performance of TIKM, TMKM and uTMKM schemes in systems with one SH. Similar to [7], we employed a homogeneous cellular network that consists of 12 concatenated cells, and wrap the cell pattern to avoid edge effects. The mobility model proposed in [8] is adopted.

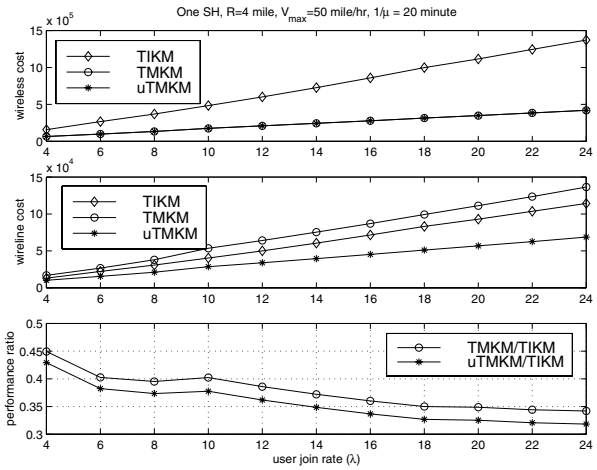

Fig. 6. Performance Comparison for different user join rate

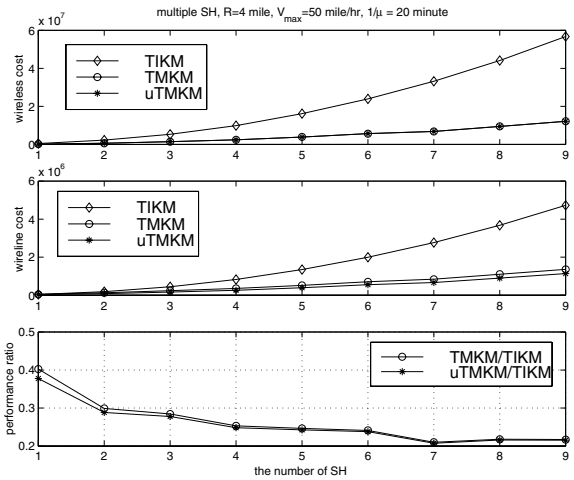

Fig. 7. Performance comparison for different number of SHs

R denotes the radius of the cells, and V_{max} denotes the maximum speed of the mobile users. The $M/M/\infty$ queueing model is used to describe the membership dynamics for a multicast service. We define the *performance ratio* η as the total communication cost of the TMKM tree divided by the total communication cost of the TIKM tree. Figure 6 shows the simulation results of wireline cost, wireless cost and the performance ratio for different user join rates, when $R = 4$ miles, $V_{max} = 50$ miles per hour, $\gamma = 2/3$, $\alpha_m^l/\alpha_u^l = n_{bs}^l/4$, and the average service time is 20 minutes. Compared with the TIKM scheme, TMKM scheme reduces the wireless cost but increases the wireline cost, while uTMKM scheme reduces both wireless cost and wireline cost. For this particular simulation setup, the wireline cost of the uTMKM scheme is only about 50% of the wireline cost of the TMKM scheme. The simulation results of the performance ratio indicate that the TMKM scheme reduces the total communication cost to 35-45% and the uTMKM scheme reduces the total communication cost to 32-43% of the total communication cost of the TIKM scheme.

In Figure 7, TIKM, TMKM and uTMKM schemes are compared when the system contains multiple SHs, under which users have the same joining/leaving/mobility behavior and experience the same network conditions. When $n_{sh} \geq 2$, the TMKM and uTMKM scheme reduce both the wireless and wireline cost. The advantages of the TMKM and uTMKM trees are more significant when the system contains more SHs. Although the uTMKM tree always has smaller wireline cost, the difference between the TMKM and uTMKM scheme becomes less significant when the number of SHs is large. In this case, the communication cost of the TMKM and uTMKM schemes can be as low as 20% of the communication cost of the TIKM trees. This indicates an 80% reduction in the communication cost.

A more complicated system containing 5 SHs with user joining rate 5, 10, 15, 20 and 25 per second respectively was also simulated. When $R = 4$ miles, $V_{max} = 50$ miles/hr, and average service time is 20 minutes, the wireless cost of the TMKM and uTMKM scheme is 21.8% of that of the TIKM scheme, the wireline cost of the TMKM scheme is 35.9% of that of the TIKM scheme, and the wireline cost of the uTMKM scheme is 32.0% of that of the TIKM scheme.

VI. Conclusion

In this paper, we presented a topology-aware key management scheme that is suitable for a large-scale cellular wireless network, where users do not have the same join/departure/mobility behavior and the network conditions may vary under different SHs. Simulations were performed for different user join rates and varying number of SHs. Compared with the traditional key management schemes that are independent of the network topology, the proposed TMKM scheme can significantly reduce the communication cost by up to 80%. In addition, we presented the uTMKM scheme, which employs both unicast and multicast communication in the delivery of rekeying messages and can further reduce the wireline communication of rekeying messages by about 50%. We also showed that the proposed methods can be used to enhance the reliability of the key distribution.

References

[1] S. Paul, *Multicast on the Internet and its applications*, Kluwer Academic Publishers, 1998.

[2] M.J. Moyer, J.R. Rao, and P. Rohatgi, "A survey of security issues in multicast communications," *IEEE Network*, vol. 13, no. 6, pp. 12–23, Nov.-Dec. 1999.

[3] Y. Sun; W. Trappe; K.J.Ray Liu, "An efficient key management scheme for secure wireless multicast," *Proc. of IEEE International Conference on Communication*, pp. 1236 –1240, May 2002.

[4] C. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs," *IEEE/ACM Trans. on Networking*, vol. 8, pp. 16–30, Feb. 2000.

[5] 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*, vol. 17, no. 9, pp. 1614–1631, Sep. 1999.

[6] K. Brown and S. Singh, "RelM: Reliable multicast for mobile networks," *Computer Communication*, vol. 2.1, no. 16, pp. 1379–1400, June 1996.

[7] M. Rajaratnam and F. Takawira, "Nonclassical traffic modeling and performance analysis of cellular mobile networks with and without channel reservation," *IEEE Trans. on Vehicular Technology*, vol. 49, no. 3, pp. 817–834, May 2000.

[8] M. M. Zonoozi and P. Dassanayake, "User mobility modeling and characterization of mobility patterns," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1239–1252, Sep. 1997.