This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2008 proceedings.

A New Dynamic Group Key Management Scheme with Low Rekeying Cost

Rongxing Lu†, Xiaodong Lin†, Haojin Zhu†, Pin-Han Ho†, Xuemin (Sherman) Shen† and Zhenfu Cao‡
†Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1
‡Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, P.R. China
Email: {rxlu, xdlin, h9zhu, pinhan, xshen}@bbcr.uwaterloo.ca, zfcao@cs.sjtu.edu.cn

***Abstract*—To achieve secure group communications, it is critical to develop a secure group key management strategy to guarantee security of the group keys. In this paper, based on the forward security and secret sharing techniques, we propose a new dynamic group key management scheme to minimize the rekeying cost. The forward security technique reduces the rekeying operations in joining event, while the secret sharing technique ensures the scalability in leaving event. In addition, the proposed scheme can provide anonymous authentication as well as forward and backward confidentiality. Theoretical analysis also confirms the efficiency of the proposed scheme.**

***Keywords*: Dynamic group key management, low rekeying cost, anonymous authentication, confidentiality**

I. Introduction

Recent advances in wireless communication and mobile computation technologies have paved the way for the proliferation of many group-oriented applications, such as video conferences, network games, and Mobile TV services [1]. In these group-oriented applications, the access control is one of the important security issues, which can ensure that the protected group resources are only accessed by authorized users. Usually, the group access control can be achieved by encrypting the group resources using a secret key, called group key, only known to all authorized group members [2], [4], [6]. However, since the group members may come and depart dynamically, how to manage the *group key* becomes challenging. For example, when a user joins, the group key should be updated to prevent the joining user from accessing the past communication; and when a user departs, the group key also needs to be updated to prevent the leaving user from accessing the future communications. These key updating operations are referred to as *group rekeying* operation. Many group key management schemes have been proposed, and have been well surveyed in [1], [5]. Key tree-based group key management schemes [6]–[8] are scalable and efficient. However, the *group rekeying* operations in other schemes are not very efficient, as they are usually dependent on the group size. When the group is large, the *group rekeying* operation becomes a big burden.

In this paper, we propose a new efficient dynamic centralized group key management scheme. Based on the forward security [9] and secret sharing [10], [11] techniques, the proposed scheme can achieve low rekeying cost. In addition, the proposed scheme provides the anonymous authentication for group members, as well as excellent forward and backward confidentiality.

The remainder of this paper is organized as follows. In Section II, we present the system model, security requirements and design goals. In Section III, we review basic tools – pairing technique [12] and forward security technique. We propose our scheme in Section IV, followed by the security and performance analysis in Section V and Section VI, respectively. In Section VII, we review the related work. Finally, we draw our conclusions in Section VIII.

II. Model, Requirements and Objectives

A. System Model

We address a class of wireless network consisting of a dynamically changed set of user $\mathcal{U} = \{U_1, U_2, \cdots, U_n\}$ and a group manager GM who sets up and manages a communication group $\mathcal{G}$, as shown in Fig. 1. The user set $\mathcal{U}$ is further divided into two subsets: the group member subset $\mathcal{U}_{gm}$ and the non-group member subset $\mathcal{U}_{nm}$. To guarantee the secure group communication in $\mathcal{G}$, at the beginning of each session, the group manager GM sets a group key $\mathbf{gk}$ and distributes it to all group members. When a non-group member $U_i \in \mathcal{U}_{nm}$ wants to join group $\mathcal{G}$, it must first authenticate itself to GM. Then, only being authenticated and authorized, could it get the group key $\mathbf{gk}$ and become a valid group member $U_i \in \mathcal{U}_{gm}$.

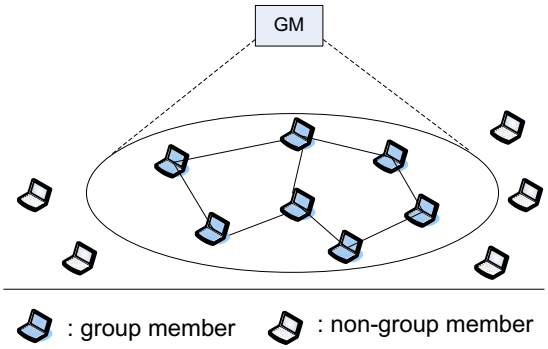

Fig. 1. System model under consideration

B. Security Requirements

There are a number of important design requirements in the effort of developing a dynamic group key management

scheme. Some of the general security properties are *group access control*, *anonymous authentication*, and *confidentiality*, which will be focused in this paper.

1) Requirements on group access control: Group access control is critical because it ensures that only authorized users can join the group.

2) Requirements on anonymous authentication: Authentication among the group members is important, and a group member should be able to prove its member legality to its neighboring group members. At the same time, location privacy gradually becomes an important requirement for personal communications [13]–[17], where a group member is obviously not willing to leak its real identity information to others. Thus, the anonymity should be ensured when the group members authenticate each other.

3) Requirements on confidentiality: An important and challenging issue in dynamic group key management is providing the confidentiality [1], which mainly includes two aspects: *forward-confidentiality* and *backward-confidentiality*.

- Forward-confidentiality: when a new user joins the group, the group manager should prevent it from accessing the past group communications.
- Backward-confidentiality: when a user leaves the group, the group manager should also prevent the leaving user from accessing the future communications.

C. Design Objectives

The goal we address here is to design an efficient dynamic group key management scheme that not only satisfies the above three security requirements, but also achieves the following two efficiency requirements:

1) Rekeying with low-cost computation and communication overheads: In order to achieve both forward confidentiality and backward confidentiality, when the group membership U_{gm} changes, it requires *group rekeying* to distribute a new group key $\mathbf{gk}$ to all the authorized members in a secure and reliable fashion. Therefore, from the efficiency perspective, the *group rekeying* should be low-cost in terms of both computation and communication overheads.

2) Anonymous authentication without extra storage overhead: When a user leaves the group $\mathcal{G}$, it should no longer be anonymously authenticated by the other group members. A straightforward way to achieve this is to maintain a *Left User List* (LUL) at each group member. However, with the increase of the number of the left users, the LUL also increases quickly. Therefore, the goal in our scheme is that each group member does not consume extra storage for anonymous authentication.

III. Preliminaries

A. Pairing Technique

Bilinear pairing [12] serves as the basis of the proposed group key management scheme, we thus firstly review the definition of the pairing technique in this subsection. Let $\mathbb{G}$ be an additive cyclic group of prime order q, and $\mathbb{G}_T$ be a multiplicative cyclic group of the same order. Assume that the discrete logarithm (DL) problem is hard in both $\mathbb{G}$ and $\mathbb{G}_T$. An admissible bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ between these two groups satisfies the following properties: i) Bilinear: We say that a map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is bilinear if $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}$ and all $a, b \in \mathbb{Z}_q^*$. ii) Non-degenerate: The map does not send all pairs in $\mathbb{G} \times \mathbb{G}$ to the identity in $\mathbb{G}_T$. That is, $\exists P, Q \in \mathbb{G}$ such that $e(P, Q) \neq 1$. iii) Computable: There is an efficient algorithm to compute $e(P, Q)$ for any $P, Q \in \mathbb{G}$. Typically, the modified Weil pairing and Tate pairing are examples of cryptographic bilinear [12]. We refer to [12], [18] for a more comprehensive description of the pairing technique.

Definition 1: A bilinear parameter generator $\mathcal{G}en$ is a probabilistic algorithm that takes a security parameter k as input and outputs a 5-tuple $(q, P, \mathbb{G}, \mathbb{G}_T, e)$ where q is a k-bit prime number, $(\mathbb{G}, +)$ and $(\mathbb{G}_T, \times)$ are two groups with order q, $P \in \mathbb{G}$ is a generator, and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is an admissible bilinear map.

B. Forward Security Technique

The forward security technique is widely believed as an efficient way to provide forward security requirement for cryptographic schemes. In 1999, Bellare and Miner [9] proposed the first forward secure signature (FSS), and then other forward security public key cryptographic schemes were appeared [19], [20]. In private key cryptography, the forward security technique is usually used for shared key updating, and can be easily implemented by a secure hash function $h()$ [21]. Fig. 2 illustrates such a paradigm, wherein the key $\mathbf{k}_i$ of the current period i is computed from the key $\mathbf{k}_{i-1}$ of the previous period by way of the hash function $h()$, e.g., $\mathbf{k}_i = h(\mathbf{k}_{i-1})$. Due to the one-wayness of $h()$, the leakage of current key $\mathbf{k}_i$ does not lead to exposure of previous keys, and thus the forward security is achieved.

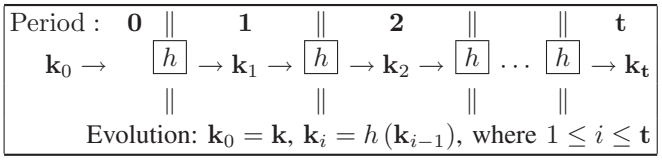

Fig. 2. Forward security key update mechanism

IV. Dynamic Group Key Management

In this section, we propose an efficient dynamic group key management scheme specifically designed for wireless ad hoc networks. The proposed scheme consists of three parts: *group session setting* including *anonymous authentication protocol* for group member, *rekeying algorithm when users joining*, and *rekeying algorithm when users leaving*. Before proceeding the scheme, some reasonable assumptions taken in the study are listed as follows.

- *Assumption-1:* The group manager GM is always trusted and serves a global observer, which can obtain the knowledge that i) when a new prospective user joins the group and ii) when an existing group member leaves the group.

- *Assumption-2:* The group manager GM can estimate the total number of joining users n and the possible total number of leaving users t in a group session. This assumption follows by the fact that the GM can stipulate the duration and capacity of a group session.
- *Assumption-3:* Any group member $U_i \in \mathcal{U}_{gm}$ is always trusted and will never collude with other non-group members before it leaves the group. However, once U_i leaves the group, it may become compromised when U_i returns. In other words, it could collude with other non-group members.
- *Assumption-4:* Both GM and each user $U_i \in \mathcal{U}$ are equipped with certain computation capability, and can execute the necessary cryptographic operations.

A. Group Session Setting

Given two security parameters k, l, the group manager GM first generates the bilinear parameters $(q, P, \mathbb{G}, \mathbb{G}_T, e)$ by running $\mathcal{G}en(k)$. Then, GM picks a random number $s \in \mathbb{Z}_q^*$ as master key, computes the system public key $P_{GM} = sP$, and chooses three hash functions H_1, H_2, h, where $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*$, $H_2 : \{0,1\}^* \times \mathbb{G}_T \to \mathbb{Z}_q^*$ and $h : \{0,1\}^* \to \{0,1\}^l$. The group parameters are

$$params := \langle q, P, \mathbb{G}, \mathbb{G}_T, e, P_{GM}, H_1, H_2, h \rangle.$$

Assume that the GM wants to establish a group session gs which lasts a time interval T and has capacity of n. Then, the GM performs the following steps.

Step 1: The GM chooses a group session key $\mathbf{gk} \in \{0,1\}^l$.

Step 2: The GM divides the time interval T into several periods $0, 1, 2, \cdots, \mathbf{t}$, and sets group key $\mathbf{gk}_i$ in different period i as

$$\mathbf{gk}_i = \begin{cases} \mathbf{gk}, & i = 0; \\ h(\mathbf{gk}_{i-1}), & 1 \le i \le \mathbf{t}. \end{cases} \quad (1)$$

Step 3: Based on Assumption-2, the GM estimates the number of joining users n and the number of leaving users t in time interval T.

Step 4: According to Shamir threshold secret sharing [10], the GM selects a t degree polynomial $F(x)$ with constant term secret $x_0 \in \mathbb{Z}_q^*$ and random coefficients $a_1, a_2, \cdots, a_t \in \mathbb{Z}_q^*$, i.e.,

$$F(x) = x_0 + a_1 \cdot x + a_2 \cdot x^2 + \cdots + a_t \cdot x^t \bmod q \quad (2)$$

Step 5: Whenever a user $U_i \in \mathcal{U}_{nm}$ wants to join the group $\mathcal{G}$ with session gs in period j, where $1 \le j \le \mathbf{t}$, the GM first authenticates it and then authorizes the group key $\mathbf{gk}_j$ and an anonymous secret key $\mathbf{sk}_i = (pid_i, A_i, b_i)$ to U_i, where

$$\begin{cases} pid_i \in \mathbb{Z}_q^* \text{ is the pseudonym of } U_i; \\ A_i = \dfrac{1}{H_1(pid_i) + s} P, \quad B_i = F(pid_i) \end{cases} \quad (3)$$

After being authorized, U_i becomes a valid group member, i.e., $U_i \in \mathcal{U}_{gm}$. Then, U_i can use the group key $\mathbf{gk}_j$ to participate in secure group communications in period j, and use $\mathbf{sk}_i$ to anonymously authenticate itself to other neighboring group members.

Anonymous authentication protocol The anonymous authentication protocol (AAP) for a group member, as shown in Fig. 3, is based on ID-based signature [18], which is described as follows.

U_i with pid_i		Neighboring member with pid_o
$params := \langle q, P, \mathbb{G}, \mathbb{G}_T, e, P_{GM}, H_1, H_2, h \rangle$, period j		
1) $r \in \mathbb{Z}_q^*, R = e(P,P)^r$		
gain current timestamp T_1		
$c_1 = H_2(T_1 \| pid_o, R) \in \mathbb{Z}_q^*$		
$C_2 = (r + c_1)A_i$	$\xrightarrow{pid_i, T_1, c_1, C_2}$	
2)		gain current timestamp T_2
		check $\lvert T_2 - T_1 \rvert > \Delta T$, if hold, terminate;
		$c_1 \stackrel{?}{=} H_2(T_1 \| pid_o, e(C_2, H_1(pid_i)P + P_{GM})e(P,P)^{-c_1})$
		if hold, U_i is anonymously authenticated

Fig. 3. Anonymous authentication protocol (AAP) for group members

Step 1: When U_i with pseudonym pid_i tries to authenticate itself to its neighboring group member with pseudonym pid_o, it first selects a random number $r \in \mathbb{Z}_q^*$ and computes

$$R = e(P,P)^r. \quad (4)$$

It then picks up the current timestamp T_1 and computes

$$c_1 = H_2(T_1 \| pid_o, R) \quad (5)$$

$$C_2 = (r + c_1)A_i. \quad (6)$$

In the end, U_i sends (pid_i, T_1, c_1, C_2) to its neighboring group member.

Step 2: Suppose that the neighbor group member with pid_o receives (pid_i, T_1, c_1, C_2) at T_2. It first checks the time interval between T_1 and T_2, if $|T_2 - T_1| > \Delta T$, where ΔT is the expected legal time interval for transmission delay, the authentication request is rejected. Otherwise, it continues checking the following equation:

$$c_1 = H_2(T_1 \| pid_o, e(C_2, H_1(pid_i)P + P_{GM})e(P,P)^{-c_1}) \quad (7)$$

If it holds, U_i is anonymously authenticated; otherwise, U_i does not pass the authentication. The correction is as follows.

$$\begin{aligned} & H_2(T_1 \| pid_o, e(C_2, H_1(pid_i)P + P_{GM})e(P,P)^{-c_1}) \\ &= H_2(T_1 \| pid_o, e(C_2, H_1(pid_i)P + P_{GM})e(P,P)^{-c_1}) \\ &= H_2(T_1 \| pid_o, e((r + c_1)A_i, (H_1(pid_i) + s)P)e(P,P)^{-c_1}) \\ &= H_2(T_1 \| pid_o, e(\frac{r + c_1}{H_1(pid_i) + s} P, (H_1(pid_i) + s)P)e(P,P)^{-c_1}) \\ &= H_2(T_1 \| pid_o, e(P,P)^{r + c_1} e(P,P)^{-c_1}) \\ &= H_2(T_1 \| pid_o, e(P,P)^r) = c_1 \end{aligned}$$

B. Rekeying Algorithm When Users Joining

In order to achieve the forward confidentiality, when a prospective user tries to join group $\mathcal{G}$, the group manager GM should execute the rekeying operation to distribute the new group key to all group members. For example, a naive solution for GM is encrypting the new group key with the old one and broadcasting to all group members.

Algorithm 1: **[A1]** Rekey($\mathbf{gk}_j$)

Data: Group key $\mathbf{gk}_j$ in period j
Result: New group key $\mathbf{gk}_{j+1}$ in period $j+1$
begin
  compute $\mathbf{gk}_{j+1} = h(\mathbf{gk}_j)$
  return $\mathbf{gk}_{j+1}$
end

In our scheme, the rekeying operation is based on the forward security technique as discussed in section III-B. When a new user joins, the group manager GM does not need to execute the rekeying operation through broadcasting. Actually, each group member can run Algorithm A1 to update its group key itself. Therefore, in terms of computation and communication overheads, our scheme is very efficient.

C. Rekeying Algorithm When Users Leaving

When a group member leaves, the rekeying operation is also required to achieve the backward confidentiality. In this subsection, we present our efficient rekeying strategy. For the presentation simplicity, we only consider the rekeying operation upon the departure of a single group member.

Let the GM observe the departure of a group member U_a. It executes the following rekeying operation:

Step 1: The GM first chooses a new group key $\mathbf{gk} \in \{0,1\}^l$ and a random number $r \in \mathbb{Z}_q^*$, and computes (α, β) as

$$\begin{cases} \alpha = rP \\ \beta = \mathbf{gk} \oplus h(x_0 rP) \end{cases} \tag{8}$$

Step 2: Let $\mathbf{sk}_a = (pid_a, A_a, B_a)$ be U_a's secret key. GM uses s and pid_a to compute

$$P' = \frac{1}{H_1(pid_a) + s} P \tag{9}$$

as the new system parameter and broadcasts $(\alpha, \beta, \mathbf{sk}_a)$ to all group members.

After receiving $(\alpha, \beta, \mathbf{sk}_a)$, each group member $U_i \in \mathcal{U}_{gm}$ runs Algorithm A2 to update its private key $\mathbf{sk}_i$.

Note that, according to the implementation of Algorithm A2, the left group member U_a could not update its private key any more. Therefore, only the valid group member $U_i \in \mathcal{U}_{gm}$ can update its private key and anonymously authenticate itself with respect to the new system parameter P'.

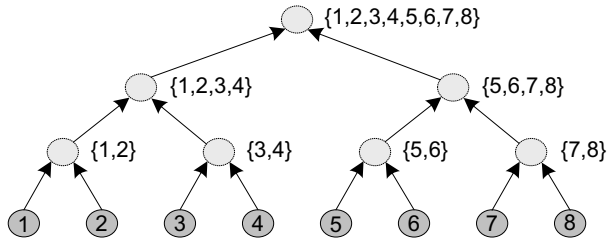

Fig. 4. Authentication tree (AT) establishment with $n_c = 8$

Algorithm 2: **[A2]** UpdatePrivateKey($\mathbf{sk}_i$, $\mathbf{sk}_a$)

Data: U_i's secret key $\mathbf{sk}_i = (pid_i, A_i, B_i)$ and $\mathbf{sk}_a = (pid_a, A_a, B_a)$
Result: U_i's new secret key $\mathbf{sk}_i = (pid_i, A_i', B_i)$
begin
  since $A_i = \frac{1}{H_1(pid_i)+s}P$, $A_a = \frac{1}{H_1(pid_a)+s}P$, pid_i, pid_a are known by U_i, it computes

$$\begin{aligned} A_i' &= \frac{1}{H_1(pid_a) - H_1(pid_i)} \cdot (A_i - A_a) \\ &= \frac{1}{(H_1(pid_i)+s)(H_1(pid_a)+s)} \cdot P \\ &= \frac{1}{(H_1(pid_i)+s)} \cdot P' \end{aligned}$$

  return new secret key $\mathbf{sk}_i = (pid_i, A_i', B_i)$
end

Let n_c be the current number of group members. To achieve all group members' mutual authentication, n_c group members can establish an authentication tree (AT) in a bottom-up fashion as the group key establishment in [1]. Firstly, n_c group members are grouped into pairs, and each pair performs anonymously mutual authentication by running the AAP in Fig. 3 to form a sub-group. Then, these sub-groups will again pair up with each other and run the AAP to form larger sub-groups. Continuing in this way, after total $\lceil \log_2 n_c \rceil$ steps, the AT is established. Fig. 4 shows an illustration example with $n_c = 8$.

With the established authentication tree, the group rekeying can be easily implemented, and the detailed steps are described in Algorithm A3.

Rekeying with extra storage. The number of the required steps in Algorithm A3 is $\lceil \log_2(t+1) \rceil + 2$ when $n_c \geq t+1$, but it does not require each group member to store any extra overhead. However, if each group member keeps all left group members' information, then the group rekeying operation becomes more flexible. Assume that total t_c group members have left the group and $n_c + t_c \geq t+1$, then only $t+1-t_c$ group members cooperate, can they recover the group key $\mathbf{gk}$. Then, the number of the required steps is only $\lceil \log_2(t+1-t_c) \rceil + 2$.

V. Security Analysis

In this section, we discuss the security of the proposed dynamic group key management scheme.

- **Group access control.** When a user U_i tries to join the group $\mathcal{G}$, it must be first authenticated and authorized by the group manager GM and then obtain the group key. Thus, it is impossible for an unauthorized user to get the group key $\mathbf{gk}$ to access the group $\mathcal{G}$.
- **Anonymous authentication.** The authentication request message (pid_i, T_1, c_1, C_2) in the AAP, as shown in Fig. 3, is unforgeable and resistant to the replay attack, since i) it is based on the provable secure ID-based signature

TABLE I
REKEYING COST ON COMPUTATION, BROADCAST COMMUNICATION, AND EXTRA STORAGE

	Computation Costs	GM Broadcast Comm.	Extra Storage
Rekeying for users joining	negligible	0	0
Rekeying for a user leaving (no extra storage)	$\begin{cases} \mathsf{Pmul} + (\lceil \log_2 n_c \rceil + 1)(2\mathsf{Pmul} + \mathsf{Pair}), & n_c < t+1; \\ 2\mathsf{Pmul} + (\lceil \log_2 (t+1) \rceil + 1)(2\mathsf{Pmul} + \mathsf{Pair}), & n_c \geq t+1. \end{cases}$	1	0
Rekeying for a user leaving (extra storage)	$\begin{cases} \mathsf{Pmul} + (\lceil \log_2 n_c \rceil + 1)(2\mathsf{Pmul} + \mathsf{Pair}), & n_c + t_c < t+1; \\ 2\mathsf{Pmul} + (\lceil \log_2 (t+1-t_c) \rceil + 1)(2\mathsf{Pmul} + \mathsf{Pair}), & n_c + t_c \geq t+1. \end{cases}$	1	t_c

Algorithm 3: **[A3]** UpdateGroupKey(n_c group members)

Data: n_c group members
Result: Each valid group member gets the new group key **gk**

begin
  switch n_c **do**
    case $n_c < t+1$
      n_c group members establish an AT in $\lceil \log_2 n_c \rceil$ steps, then authenticate AT to the GM and gain the new group key **gk** in an addition step;
    case $n_c \geq t+1$
      since n_c can be expressed as $n_c = (t+1) \cdot u + v$, where $0 \leq v < t+1$, n_c members are divided into $u+1$ sub-groups, the size of the first u sub-groups are $t+1$, and the size of the last one is v;
      Each sub-group $\mathcal{SG}$ with size $t+1$ establishes an AT in $\lceil \log_2(t+1) \rceil$ steps, and recovers the new group key **gk** with each group member U_i's secret key (pid_i, B_i) by Lagrange interpolation as follows:

$$\mathbf{gk} = \beta \oplus h(\alpha'), \text{ where } \alpha' = \sum_{pid_i \in \mathcal{SG}} (b_i \cdot B_i)\alpha$$

$$b_i = \prod_{pid_j \in \mathcal{SG}/\{pid_i\}} \frac{pid_j}{pid_j - pid_i}$$

      The rest v group members in the last sub-group authenticate themselves to other members and gain the group key **gk** in last step.
    end
  end
end

scheme in [18]; ii) the replay of the old request message (pid_i, T_1, c_1, C_2) sent by the user will fail because the validity of these messages can be checked through the timestamp T_1. In addition, the pseudonym pid_i also won't reveal the real identity of user U_i. Therefore, based on these properties, each group member can use the AAP to achieve anonymously authentication.

- **Confidentiality.** Since the forward security technique in Fig. 2 is applied, the *forward-confidentiality* obviously follows. When a user leaves the group $\mathcal{G}$, it could not update its private key and be anonymously authenticated by the other group members. At the same time, we have assumed that the group member is trusted and will not collude with any non-group member, thus, the number of left group members is at most t, which is less than the threshold value $t+1$. Therefore, the *backward-confidentiality* can be guaranteed.

VI. PERFORMANCE ANALYSIS

To evaluate the performance of the proposed scheme, we focus on the group rekeying cost when users join and leave. Table I lists the group rekeying cost in terms of computation, GM broadcast communication, and extra storage. The computation cost is mainly measured by the most cost-consuming pairing (Pair) operation and the point multiplication (Pmul) operation, while the pre-computation and parallel processing techniques are also considered.

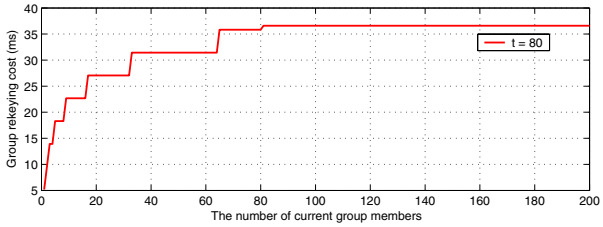

Fig. 5. Group rekeying cost (without extra storage) varies with n_c, where $1 \leq n_c \leq 200$, $t = 80$

From Table I, we can observe that the group rekeying for user joining is particularly efficient. By adopting the pairing technique in [18], we can obtain the rough time costs for Pair and Pmul, which is 2.82 ms and 0.78 ms, respectively. Fig. 5 illustrates that the group rekeying cost for a user departure (without extra storage) varies with the current number of group members n_c, where $1 \leq n_c \leq 200$ and t is fixed as 80. It can be seen that the group rekeying cost first increases with n_c, and keeps a constant 36.6 ms independent with n_c after $n_c \geq 81$. Fig. 6 shows the group rekeying cost for a user departure (with extra storage) varies with the number of current group members n_c and the current number of left group members t_c, where $1 \leq n_c \leq 200$, $1 \leq t_c \leq 80$ and t is fixed as 80, the maximal value of t_c. We can observe from Fig. 6 that for a large fixed n_c, for example $n_c = 200$, the group rekeying

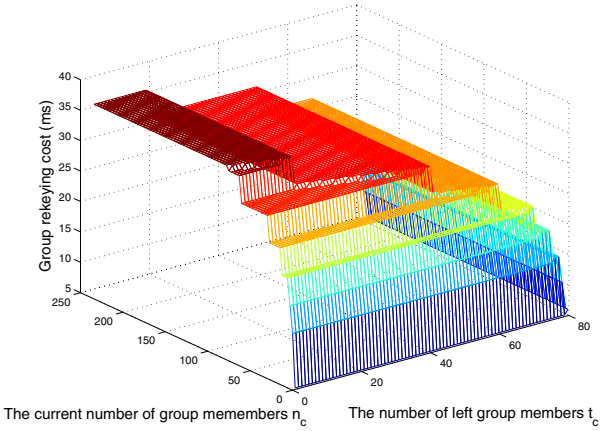

Fig. 6. Group rekeying cost (with extra storage) varies with n_c, where $1 \leq n_c \leq 200$, $t = 80$

cost will decrease with the increase of t_c. Therefore, the group rekeying for a user departure is also efficient.

VII. Related Work

Group key management schemes have two different flavors: centralized group key management [5]–[8] and distributed group key management including subgroup control and member control. Since the focus of this work is to provide a dynamic group key management with a centralized group manager, we only review the related work on centralized group key management schemes.

All centralized group key management schemes require a group manager for rekeying the group. A trivial rekeying solution is the group manager chooses a new group key $\mathbf{gk}$, encrypts the new group key with the old group and broadcasts to group members in the case of a joining; encrypts the new group key $\mathbf{gk}$ with each member's individual key and sends it to the remaining members one by one in the case of a leaving. The number of rekeying messages in joining is $O(1)$, and the number of rekeying messages in leaving is $O(n)$. Therefore, there is no scalability when group member leaves [5]. Wong et al. [6] proposed an efficient key tree approach, LKH, that associates keys in a hierarchical tree and rekeys at each joining or leaving event. In LKH, each group member keeps $O(\log n)$ personal keys, and the number of rekeying messages in joining and leaving events are both $O(\log n)$. Therefore, LKH is scalable. Later, [7] and [8] apply the periodic rekeying concept from [22] to the key tree setting, which save the group manager cost substantially. In the proposed dynamic group key management scheme, the forward security technique ensures the rekeying operations more efficient in joining event, and the anonymous authentication between the group members also makes the rekeying operations scalable in leaving event.

VIII. Conclusions

In this paper, we have proposed an efficient dynamic group key management scheme. The proposed scheme can achieve not only anonymous authentication, well forward, and backward confidentiality, but also low rekeying cost. More comprehensive simulations are underway to further evaluate the rekeying cost of the proposed scheme.

References

[1] Y. Mao, Y. Sun, M. Wu, and K. Liu, "JET: dynamic join-exit-tree amortization and scheduling for contributory key management", *IEEE/ACM Transactions on Networking*, Vol. 14, No. 5, pp. 1128-1140, 2006.

[2] M. Moyer, J. Rao, and P. Rohatgi, "A survey of security issues in multicast communications", *IEEE Network*, Vol. 13, No. 6, pp. 12-23, 1999.

[3] C. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs", *IEEE/ACM Transactions on Networking*, Vol. 8, No. 1, pp. 16-30, 2000.

[4] Y. Jiang, C. Lin, M. Shi, and X. Shen, "Self-healing group key distribution with time-limited node revocation for wireless sensor networks", *Ad Hoc Networks*, Vol. 5, No. 1, pp. 14-23, 2007.

[5] B. Pinkas, "Efficient state updates for key management", *Proceedings Of The IEEE*, Vol. 92, No. 6, pp. 910-917, 2004.

[6] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs", *IEEE/ACM Transaction on Networking*, Vol. 8, No. 1, pp. 16-30, 2000.

[7] X. S. Li, Y. R. Yang, M. G. Gouda, and S. S. Lam, "Batch rekeying for secure group communications", in *Proc. 10th Int. World Wide Web Conf. (WWW10)*, pp. 525-534, Orlando, FL, May 2001.

[8] Y. R. Yang, X. S. Li, X. B. Zhang, and S. S. Lam, "Reliable group rekeying: a performance analysis", in *Proc. ACM SIGCOMM*, pp. 27-38, Orlando, San Diego, CA, Aug. 2001.

[9] M. Bellare and S. Miner, "A forward-secure digital signature scheme", in *Advances in Cryptology - CRYPTO'99*, LNCS 1666, pp. 431-448, Springer-Verlag, 1999.

[10] A. Shamir, "How to share a secret", *Communications of the ACM*, Vol. 22, No. 11, pp. 612-613, 1979.

[11] X. Lin, R. Lu, P.-H. Ho, X. Shen, and Z. Cao, "TUA: a novel compromise-resilient authentication architecture for wireless mesh networks", *IEEE Transaction on Wireless Communications*, to appear.

[12] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing", in *Advances in Cryptology - CRYPTO 2001*, LNCS 2139, pp. 213-229, Springer-Verlag, 2001.

[13] Y.-C. Hu and H. J. Wang, "A framework for location privacy in wireless networks", in *Proc. of ACM SIGCOMM Aisa Workshop 2005*, Beijing, China, April 2005.

[14] K. Ren, W. Lou, R. H. Deng, and K. Kim, "A novel privacy preserving authentication and access control scheme in pervasive computing environments", *IEEE Transaction on Vehicular Technology*, Vol. 55, No. 4, pp.1373-1384, July 2006.

[15] R. Lu, X. Lin, H. Zhu, P.H. Ho and X. Shen, "ECPP: Efficient Conditional Privacy Preservation Protocol for Secure Vehicular Communications", Proc. IEEE INFOCOM'08, Phoenix, AZ, USA, April 14-18, 2008.

[16] C. Zhang, R. Lu, X. Lin, P.H. Ho and X. Shen, "An Efficient Identity-based Batch Verification Scheme for Vehicular Sensor Networks", Proc. IEEE INFOCOM'08, Phoenix, AZ, USA, April 14-18, 2008.

[17] X. Lin, X. Sun, P.H. Ho, and X. Shen, "GSIS: A Secure and Privacy Preserving Protocol for Vehicular Communications", IEEE Trans. on Vehicular Technology, Vol. 56, No. 6, pp. 3442-3456, 2007.

[18] P. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps", in *Advances in Cryptology - AISACRYPT 2005*, LNCS 3788 , pp. 515-532, Springer-Verlag, 2005.

[19] M. Abdalla and L. Reyzin, "A new forward-secure digital signature scheme", in *Advances in Cryptology - AISACRYPT 2000*, LNCS 1976, pp. 116-129, Springer-Verlag, 2000.

[20] R. Lu, Z. Cao and X. Dong, "Authenticated encryption protocol with perfect forward secrecy for mobile communication", *Wireless Communications and Mobile Computing*, Vol. 6, No. 3, pp. 273-280, 2006.

[21] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "A novel anonymous mutual authentication protocol with provable link-layer location privacy", *IEEE Transaction on Vehiculare Technology*, submitted.

[22] S. Setia, S. Koussih, and S. Jajodia, "Kronos: a scalable group rekeying approach for secure multicast," in *Proc. IEEE Symp. Security and Privacy*, pp. 215-228, May 2000.