Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 51 (2007) 4303–4321
Computer Networks
www.elsevier.com/locate/comnet

Stateless key distribution for secure intra and inter-group multicast in mobile wireless network ☆

Weichao Wang [a,*], Tylor Stransky [b]

[a] Department of Software and Information Systems, University of North Carolina at Charlotte, Charlotte, NC, United States
[b] Department of EECS, University of Kansas, Lawrence, KS, United States

Received 26 November 2006; received in revised form 5 April 2007; accepted 8 June 2007
Available online 16 June 2007

Responsible Editor: J. Misic

Abstract

Group communication has become an important component in wireless networks. In this paper, we focus on the environments in which multiple groups coexist in the system, and both intra and inter-group multicast traffic must be protected by secret keys. We propose a mechanism that integrates polynomials with stateless secret updates to achieve personal key share distribution and efficient key refreshment during group changes. The proposed mechanism distributes keys via true broadcast. Compared to previous approaches, the proposed mechanism has the following advantages: (1) The adoption of symmetric encryption/decryption for multicast traffic matches the limited processing capability of wireless nodes. (2) The stateless feature of key distribution matches the properties of mobile wireless networks including frequent topology changes and temporary connection disruptions. (3) Special mechanisms are designed to reduce the communication overhead during key updates and provide protection against both intra and inter-group impersonation. The storage, computation, and communication overhead of the proposed mechanism is investigated. Analysis and simulation are conducted to demonstrate the improvements over previous approaches.

Keywords: Stateless key distribution; Secure inter-group multicast; Overhead reduction; Impersonation prevention

1. Introduction

Group communication has become an important component of many applications in mobile wireless networks. It takes advantage of the broadcast characteristic of wireless communication to accelerate information propagation and improve energy efficiency at the mobile nodes when they are equipped with omni-directional antenna. For example, traditional multicast, stateless multicast, and overlay multicast protocols have been developed for wireless networks and a good review can be found in [1]. To prevent attackers from paralyzing the network and services by manipulating and abusing multicast communication, secret keys must be distributed

☆ This research is supported in part by NSF DUE Award #0633143, smart work zone deployment initiative (SWZDI), and KUCR NFGRF. A preliminary version of this work appeared in the proceedings of ACM SASN 2005.
* Corresponding author. Tel.: +1 704 724 0238.
E-mail addresses: weichaow@gmail.com (W. Wang), tstransk@ittc.ku.edu (T. Stransky).

1389-1286/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2007.06.001

and properly maintained throughout the lifetime of the network. Therefore, key establishment and refreshment becomes a critical problem for the applications and must be paid special attention.

In this paper, we focus on the problem of key distribution and update for secure inter-group communication. There are various applications in which the mobile nodes are divided into multiple groups and multicast traffic exists both within the same group and among different groups. Below we describe two examples that can adopt secure inter-group multicast to improve the robustness and efficiency of application level services in wireless networks.

In a United Nations Peacekeeping Operation, three groups of soldiers coming from countries A, B, and C respectively work together to secure an area. Soldiers from the same country or different countries can communicate through multihop wireless connections. Driven by the differences in responsibilities and security clearance levels, when an event is observed by a soldier of country A, descriptions with different contents or different levels of details will be provided to different soldier groups. To support such requirements, a wireless node needs to encrypt its messages with different keys. Secure inter-group multicast is expected in the scenario: only members of the target group could recover the information, and all other nodes should not get access.

Inter-group communication can also be used by soldiers from the same country. We may divide the soldiers into different groups based on their ranks. Each group has its own security level and access right to the information. For example, a soldier may report an event that can be read only by the generals, but not the captains. Secret keys must be deployed to restrict the nodes that can recover the information and participate in the operations.

Enforcing security in these environments puts new challenges to researchers. First, it is different from secure multicast because it involves both intra-group and inter-group communication and multiple keys are required. It is also different from the pair-wise key establishment or pre-distribution methods. Second, membership changes among groups will bring new difficulties to key management. For example, a node may join another group temporarily and switch back later. Therefore, the changes are not necessarily monotonic. Finally, some of the mobile nodes may become temporarily disconnected from the rest of the network because of various reasons such as unreliable communication medium, node movements, and device malfunction. When they are connected again, they should be able to recover the latest keys by passively listening to the broadcast key distribution messages. Therefore, a new approach that supports stateless and efficient key distribution is required to protect multicast traffic in these applications.

A straightforward solution is to deploy a public–private key pair for every group. Every node knows all the public keys and only the private key of the group that it belongs to. For example, for the application described above, a soldier will know Pub_{soldier}, Pub_{captain}, Pub_{general}, and Pri_{soldier}. When he wants to send a message that can be read only by the generals, he can use the Pub_{general} to encrypt the information. To support key updates during group changes, existing approaches such as Logical Key Hierarchy (LKH) [2,3] can be adopted.

This approach is simple, yet with three major disadvantages: (1) Asymmetric encryption, which usually involves exponential computation, must be adopted to protect multicast traffic. It is not efficient for a wireless node when its limited energy and computation capability is considered. (2) When the security level of a mobile node changes or a compromised node is detected and expelled from the current group, secret keys must be updated. It will introduce an overwhelming amount of computation overhead for generating secure public–private key pairs when such changes happen frequently [4]. (3) Since the public keys are known to every node, we cannot determine the identity of the sender based on the encrypted message unless additional authentication methods are adopted. An attacker can easily impersonate another node. This threat is especially severe in inter-group communication since the mobile nodes belonging to different groups usually have weaker trust among each other.

In this paper, we propose a new mechanism that integrates polynomial-based personal key determination with stateless secret update to overcome these difficulties. First, symmetric keys are used to protect the multicast traffic in the same group. At the same time, polynomials are adopted to determine the keys to protect inter-group communication. We calculate the personal key share of a node by applying its unique identity *ID* to the polynomial. When a node changes its group, we adopt the stateless key distribution approaches [5–7] to update secrets via true broadcast. To reduce key update overhead, improve the scalability of the

approach when the number of groups increases, and improve its safety, special methods are designed to reduce the broadcast traffic and detect inter-group impersonation.

While the research is rooted from stateless key management [5–7], we make the following contributions through the proposed mechanisms: (1) We design mechanisms to protect both intra and inter-group multicast traffic among wireless nodes. The proposed mechanism avoids heavy computation and improves information processing efficiency. Since the stateless property is integrated, recovering the latest keys does not depend on the knowledge of previous keys when the separated nodes become reconnected to the network. This feature is especially valuable to improving tolerance to network disruptions. (2) We design mechanisms to reduce the broadcast traffic for key updates during group member changes and defend against inter-group impersonation attacks. (3) Analysis and simulation are conducted to demonstrate the improvements over previous approaches.

The remainder of the paper is organized as follows: In Section 2, we review the previous research that contributes to our approach. Section 3 presents the assumptions and models of the system. Section 4 describes how secure intra-group and inter-group communication is achieved. In Section 5, we describe the basic approach to stateless key distribution when a node joins or leaves a group. Forward and backward secrecy are enforced. Section 6 describes the overhead reduction and impersonation prevention mechanisms in detail. Section 7 presents the simulation results to demonstrate the improvements and investigates the robustness of the proposed mechanism. Finally, Section 8 concludes the paper and discusses future extensions.

2. Related work

Key management for secure group communication has attracted a lot of research efforts, and very encouraging results have been collected. Below we summarize some of the previous approaches.

In the early solutions such as Group Key Management Protocol (GKMP) [8], the centralized controller will distribute a key encryption key (KEK) and a traffic encryption key (TEK) to a node when it joins the group. These one-to-one distribution mechanisms do not scale to large networks.

To address the scalability problem, members of a multicast group have been organized into a hierarchy. Every node is treated as a leaf and it holds all the keys from the leaf to the root. This Logical Key Hierarchy [2,3] also reduces the size of rekeying messages. Various approaches have been proposed to improve the method by reducing the number of keys stored at group members, reducing the broadcast traffic during key refreshment, and supporting forward and backward secrecy. The adopted methods include using one way functions to lessen the key distribution overhead when a node joins the group [9–11], using a-ary to reduce tree size [12], using flat tables to reduce keys held by KDC [11], and using pseudo-random functions to build and manipulate the keys in the hierarchical tree [13].

To avoid single point of failure and to restrict the impacts of a group member change, several mechanisms have been developed to divide nodes into multiple subgroups. In Iolus [14] each subgroup uses an independent key and the agents of the subgroups form a top-level management team. The separation of encryption keys in different subgroups enables the membership changes to be handled locally. The disadvantage is that inter-subgroup traffic must be translated by the agents. Dual encryption protocol [15] has been proposed to deal with the trust of third parties. Cipher sequences [16] have been integrated into the subgroups to improve the efficiency of key distribution and update. A synchronized group key distribution protocol is adopted by Hydra [17] to achieve key refreshment when a membership change in a subgroup happens.

In several mechanisms the keys are updated as a function of time. For example, in [18], short slices of time are organized as a tree and every slice uses a different key. Every node will receive decryption keys corresponding to the time duration in which it is a legal group member so that access to traffic is granted. The approaches such as Kronos [19] will periodically rekey the group and they provide an efficient solution for the environments in which membership changes happen very frequently.

Various approaches have been proposed to improve the efficiency and security of group communication in wireless networks. They target at special features such as node mobility and frequent link changes. The limited resources on computation capability, energy, and available bandwidth are also considered. LKHW [20] extends the application of Logical Key Hierarchy to sensor networks and it enforces both backward and forward secrecy. In [21], a node will join a multicast group by attaching to the closest member so that a physical security tree

structure is constructed. The joining and leaving operations are managed by the upstream node in the tree structure. The research efforts in [22,23] consider the location information and different models of signal attenuation when constructing the multicast hierarchy so that a better energy efficiency can be achieved. To reduce the maintenance overhead of the forwarding state in wireless nodes, stateless multicast protocols [24,25] and overlay multicast protocols [26,27] have been developed.

Since the schemes such as Diffie-Hellman and the public key infrastructure involve exponential computation, the mechanisms that adopt them [28,29] will put a severe challenge to the computation capability of the mobile nodes. Researchers have integrated trust with secure group communication and proposed several approaches [30,31]. The mobile nodes are divided into different areas or clusters, and the key distribution and revocation methods under various trust models are studied.

Both CKDS [32] and GKMPAN [33] avoid the adoption of LKH. CKDS uses a matrix-like key distribution structure in which the unknown secrets to the revoked nodes can be used to distribute new keys. GKMPAN depends on TESLA for the authentication of multicast packets and group key updates. It assumes high node mobility and provides the desirable *stateless property*, which allows the mobile nodes that miss the rekeying procedure due to network partition to recover the current group keys.

In a mobile wireless network, the nodes may become disconnected from the network because of various reasons (node mobility, unreliable transmission medium, etc.). Therefore, the stateless property of group key distribution, which enables a reconnected group member to recover new session keys by passively listening to the broadcast packets, is especially important to reduce key distribution overhead and improve its tolerance to network disruptions. In [6], a subset-cover framework is proposed to achieve the goal. The mobile nodes are organized as a full binary tree and each node is equipped with multiple secret keys corresponding to different subsets. The approaches in [13,34] take a tree-based structure to distribute keys and achieve resistance to packet loss by appending additional information to subsequent messages.

Polynomial interpolation was first used to implement threshold secret sharing [35]. It allows a dealer to distribute a secret s to n players and at least $t \leqslant n$ players are required to recover the information. Researchers have extended the stateless property and developed self-healing approaches based on these techniques. Staddon et al. [7] proposed a self-healing key distribution mechanism with revocation capability. The users are capable of recovering lost group keys without interacting with the *group manager*. The *manager* uses a bivariate polynomial as a masking function to privately transmit information to group members. Liu et al. [5] proposed a more efficient self-healing group key distribution scheme with revocation capability based on the result. They assume that the set of revoked nodes changes monotonically. A novel personal key distribution scheme is developed and the storage and communication overhead is reduced. More et al. [36] have improved their previous result by applying sliding window to the self-healing procedure so that more consistent robustness and less overhead can be achieved.

Using polynomials to distribute personal key shares for secure inter-group multicast is also adopted in [37]. However, the approach uses flat tables [11] to support key updates and the secrets have to be recovered sequentially, which is not robust against network disruptions.

3. Our models

3.1. Network and communication model

We assume that the links among wireless nodes are bidirectional and two neighboring nodes can always send packets to each other. This assumption will hold under most conditions when the power of the nodes has not been exhausted.

We adopt a simplified model to describe the intra and inter-group communication. We assume that the nodes are divided into multiple groups and secret keys are deployed to control the access to multicast packets, whose target could be the members in the same group or in a different group. A node may change its group as time passes by and new members can join the network dynamically. The nodes that are compromised by attackers will be expelled from the network when they are detected. We assume that the multicast data packets have a much higher frequency than group member changes and they explain a majority of the computation and communication overhead caused by multicast operations. This model is powerful enough to describe the applications in Section 1, and a lenient space has been left for future extensions.

Secret keys must be deployed to protect the multicast traffic so that only the group members with valid keys can send out the messages and get access to the encrypted information. For the simplicity of the presentation, we assume that a centralized *group manager GM* is in charge of key distribution and update for all different groups. In real applications, a distributed approach can be adopted and the role of *GM* can be jointly played by multiple special nodes (e.g. group leaders) in the network. The generation of group managers, the communication among them, and their relationships to node mobility will be discussed in Section 7. We must emphasize that the responsibilities and workload of *GM* are totally different from a centralized data transmission gateway that "translates" all multicast packets among different groups. We also assume that a multicast packet can be forwarded by both the members in the target group and the nodes in other groups.

3.2. Threat model

Security threats to wireless networks may come from all layers. The malicious nodes can jam the physical layer. There have been approaches using spread spectrum [38] to provide resistance to such attacks. There are also Denial-of-Service (DoS) attacks on the medium access control layer [39]. For example, if a malicious node keeps sending noises and causes collisions, the communication within the neighborhood will be paralyzed. Fairness control mechanisms such as time division multiple access [40] can avoid one attacker consuming all available bandwidth. This paper will not discuss solutions to these attacks.

We assume that malicious nodes can eavesdrop on and record the packets that are transmitted over the wireless medium. They can also conduct active attacks by inserting, modifying, or discarding packets. We assume that malicious nodes do not have the computation resources to directly break encryption keys.

When a node changes its group, new keys must be generated to replace the old secrets held by it. During these updates, two features need to be enforced by the key management scheme as described in [13,41]: *forward* and *backward secrecy*. *Forward secrecy* guarantees that when a node is expelled from a group, it cannot derive subsequent keys based on the knowledge of the old ones. *Backward secrecy* guarantees that when a node joins a group, it cannot discover the old keys based on its current knowledge and get access to previous traffic. Two features together will prevent information leakage in highly dynamic environments.

3.3. Notations

We assume that every node is uniquely identified by a node ID u, where $u \in \{1 \cdots n\}$ and n is the total number of nodes. The nodes are divided into d different groups, which are represented by G_1 to G_d, respectively. All operations described in the protocol will take place in a finite field F_q, where q is a prime number with a large enough value.

We assume that in a group G_i, at most t mobile nodes will collude together and attempt to compromise the key management mechanism. Since a mobile node can switch its group dynamically and rejoin the current group later, the group membership changes are not monotonic. We assume that at any moment during the network lifetime, at most l nodes who were members of G_i do not belong to G_i any more, and their key shares need to be revoked. Based on this definition, if a node rejoins G_i, it will not be counted in the set of revoked nodes.

We use $E_k(msg)$ and $D_k(msg)$ to represent the encryption and decryption of the message *msg* with a symmetric key k, respectively. We use $h(x)$ to represent a t-degree polynomial in $F_q[x]$, and $h(u)$ is the value of the function at point u. Similarly, we use $f(x)$ and $F(x)$ to represent l degree and $l + t$ degree polynomials, the functionality of which will be described in detail in Section 5. We use $S_{GM}(msg)$ to represent the digital signature of the *group manager* on the message, and every node in the network can verify this signature. Similar to [5,7], we assume that the network lifetime can be divided into m sessions and key refreshment will be conducted at the beginning of every session. A session can be determined by a time duration or a certain number of group changes. Different groups can reside in different sessions. Although the proposed key distribution mechanism may be restricted by the number of sessions m, we note that the lifetime extension schemes in [7] can be applied to our approach, which will be discussed in Section 7.

We assume that a packet has the format (*sender*, *receiver*, *objective*, *data contents*, *integrity protection*). The *group manager* is represented by *GM* in a packet. If a packet has a group name as the *receiver*, it is a multicast message that targets at all current members of the group.

4. Secure group communication

During the network initiation procedure, every node will receive a set of secret keys from the *group manager* through a secure channel such as the physical contact before deployment. These keys can be divided into two groups: traffic encryption keys (TEK) to protect multicast packets, and key encryption keys (KEK) to support secret refreshment. Without losing generality, we assume that the nodes are divided into three groups G_1, G_2, and G_3. Below we use a node u in group G_2 as an example to illustrate the secret keys that it holds.

We assume that node u can communicate with the *group manager* securely. This can be achieved through a pair-wise key $K_{u,GM}$ shared between the two entities. As a member of G_2, u will get a copy of the symmetric group key $K_{2,j}$ which is used to encrypt and decrypt the multicast traffic within the group in session j. Here the first index '2' represents the group number, and the second index 'j' represents the session number.

We use t-degree polynomials $h(x)$ to determine the personal key shares and protect inter-group multicast traffic. As a member of G_2, u must be able to recover multicast packets sent by the nodes in G_1 and G_3. Therefore, it will be aware of two such functions, $h_{2,1,j}(x)$ and $h_{2,3,j}(x)$. Here the first and second indexes represent the destination and source groups of the multicast packets, respectively. The third index represents the session number in the destination group. For example, $h_{2,1,j}(x)$ is the polynomial to determine the personal key shares of the members in G_1 to send multicast packets to G_2 in session j. A node v in G_1 will get its personal key share $h_{2,1,j}(v)$ from the *group manager*. When it wants to send a multicast packet *msg* to the members in G_2, it will send out $(v, G_2, E_{h_{2,1,j}(v)}(msg, H(msg)))$. Since every node in G_2 knows $h_{2,1,j}(x)$, it can calculate the personal key share $h_{2,1,j}(v)$ by applying v to the polynomial and recover the information. Similarly, u is aware of the polynomial $h_{2,3,j}(x)$ so that it can decrypt multicast messages from the members in G_3. To enable node u to send multicast packets to the members in G_1 and G_3, it will get two personal key shares $h_{1,2,j'}(u)$ and $h_{3,2,j''}(u)$ from the *group manager*. Here we assume that the sessions in different groups are not synchronized, and G_1 and G_3 are in sessions j' and j'' respectively.

Two advantages have been brought by the personal key shares determined by polynomials. First, for two different nodes v and w in G_1, they will have different personal keys $h_{2,1,j}(v)$ and $h_{2,1,j}(w)$ to encrypt multicast packets to G_2. Therefore, information isolation has been achieved, and only the sender and members in the target group can recover the packet. Second, it becomes more difficult for an attacker to impersonate another node in the same group unless it can collect $t+1$ personal keys and reconstruct the polynomial $h(x)$. Secret separation among the nodes in the same group is especially valuable to wireless networks that consist of mobile nodes coming from different organizations. Under these conditions, the members in one group usually have weaker trust on the members in another group. Therefore, the mobile nodes want to confirm the identity of

Table 1
Secrets held by node u and their usage

Secret keys	Domain	Usage
Traffic encryption keys		
$K_{u,GM}$	F_q	Pair-wise key shared between u and the *group manager*
$K_{2,j}$	F_q	Group key shared by members of G_2
$h_{2,1,j}(x)$	t-degree polynomial in $F_q[x]$	Polynomial to determine the keys for decrypting the multicast traffic from a node in G_1
$h_{2,3,j}(x)$	t-degree polynomial in $F_q[x]$	Polynomial to determine the keys for decrypting the multicast traffic from a node in G_3
$h_{1,2,j'}(u)$	F_q	Personal key share to encrypt multicast traffic sent to the members of G_1
$h_{3,2,j''}(u)$	F_q	Personal key share to encrypt multicast traffic sent to the members of G_3
Pre-distributed key encryption keys		
$f_{2,i}(u)$ $(i = 1 \cdots m)$	F_q	Values used for group key refreshment in G_2 for session i
$F_{1,2,i}(u)$ $(i = 1 \cdots m)$	F_q	Value used for personal key share update for G_1 in session i
$F_{3,2,i}(u)$ $(i = 1 \cdots m)$	F_q	Value used for personal key share update for G_3 in session i

This table illustrates the traffic encryption keys (TEK) and key encryption keys (KEK) for node u and their usage. We assume that u is a member of G_2 and G_2 is in session j. G_1 and G_3 are in session j' and j'' respectively.

the source when an inter-group packet is received. Mechanisms to prevent inter-group impersonation will be discussed in detail in Section 6.

Table 1 summarizes the traffic encryption keys held by node u and their usage. We assume that u is a member of G_2 and G_2 is in session j. We also assume that group G_1 and G_3 are in session j' and j'' respectively. The key encryption keys and the refreshment operations will be discussed in detail in Section 5.

5. Basic approach to stateless key distribution

When a group change happens, the corresponding keys must be updated to enforce forward and backward secrecy. In this section, we present the basic approach to stateless key updates for secure communication among multiple groups. We first introduce the pre-distributed information that is used in secret recovery. Section 5.2 introduces key update operations for the nodes that do not change groups. Section 5.3 investigates key refreshment for the newly added group members or expelled nodes. In Section 5.4, we discuss the generation of secrets and polynomials. Section 5.5 proves the safety of the approach. Mechanisms to extend lifetime of the approach are discussed in Section 5.6. In Section 5.7, we investigate the overhead of the basic approach and demonstrate the urgency to reduce it.

5.1. Pre-distribution

To support stateless key refreshment, the *group manager* will distribute some information that is essential to the secret recovery operations to a mobile node during the system initiation procedure. We assume that the mobile nodes are divided into d groups, and the network lifetime is divided into m sessions. Within the same group, at most t nodes will collude together to impersonate another member. At the same time, at most l nodes who were members of the group G_i do not belong to the group in the current session j. We do not distinguish the nodes that leave a group voluntarily from the expelled members, and all of their node *ID*s will be put in the revocation set $R_{i,j}$. Here the first index represents the group number, and the second index denotes the session number. The numbers t and l will jointly determine the degrees of the polynomials that are used during key refreshment.

The *group manager* will randomly select m l-degree polynomials from $F_q[x]$ for each group, which can be denoted as $f_{i,j}(x)$, $i = 1 \cdots d$, $j = 1 \cdots m$. The first index of the functions represents the group number, and the second index denotes the session number. These functions serve as the 'masking functions' during the group key update operations. At very beginning, every node u belongs to a group G_w, and it will get its personal values $f_{w,j}(u)$, $j = 1 \cdots m$, from the *group manager* through the secure communication channel. The *group manager* will also generate m session keys for each group, which can be represented as $K_{i,j}$, $i = 1 \cdots d$, $j = 1 \cdots m$, from F_q. When group changes happen, the key management mechanism must distribute new group keys to present members of the groups.

The mobile nodes can switch their groups dynamically. In session j, we assume that $w_{i,j}$ nodes who were members of G_i are no longer in the group. Based on the assumption in Section 3, we know $w_{i,j} \leqslant l$. The set of nodes can be represented as $R_{i,j} = \{r_{i,j,1}, r_{i,j,2}, \ldots, r_{i,j,w_{i,j}}\}$, where $r_{i,j,1}$ to $r_{i,j,w_{i,j}}$ are *ID*s of the revoked nodes. Here we do not require the changes of $R_{i,j}$ to be monotonic. If a node $u \in R_{i,(j-1)}$ but it rejoins G_i in session j, it will be removed from $R_{i,j}$.

The *group manager* will select $(d-1) \times m$ polynomials with the degree $(l+t)$ from $F_q[x]$ for each group to serve as the 'masking functions' for personal key update operations. Every function is denoted as $F_{i,i',j}(x)$, where $i = 1 \cdots d$, $i' = 1 \cdots d$, $i \neq i'$, and $j = 1 \cdots m$. Here i and i' represent the destination and source groups of an inter-group personal key share, and j represents the session number in group G_i. Every node u in group G_w will receive the values $F_{i,w,j}(u)$, $i = 1 \cdots d$, $i \neq w$, $j = 1 \cdots m$. The pre-distributed key encryption information is also summarized in Table 1.

5.2. Stateless key update for unchanged group members

Without losing generality, we assume that there are three groups, G_1, G_2, and G_3, in the network. A group change (could be a joining or leaving event) happens in session j of G_1. Below we first describe the stateless key update operations for the nodes that do not change groups.

Step 1. Updating the group key and polynomials

1. The current members of G_1 have been using $K_{1,(j-1)}$ to encrypt the multicast traffic within the group. To enforce backward and forward

secrecy, the new group key $K_{1,j}$ must be established.

Given the set of revoked nodes $R_{1,j}$, the *group manager* will broadcast:

$$(GM,\ G_1,\ \textit{group key update for } G_1 \textit{ in session } j,\ R_{1,j},\ \overline{P}_{1,j}(x) = g_{1,j}(x)\cdot K_{1,j} + f_{1,j}(x),\ E_{K_{1,j}}(h_{1,2,j}(x), h_{1,3,j}(x)),\ \textit{digital signature}),$$

where $g_{1,j}(x)$ is determined by *ID*s of the revoked nodes as $g_{1,j}(x) = (x - r_{1,j,1})(x - r_{1,j,2})\cdots(x - r_{1,j,w_{1,j}})$.

2. Every node u in G_1 that does not belong to $R_{1,j}$ will try to recover the new group key $K_{1,j}$ from the received packet. It can calculate $\overline{P}_{1,j}(u)$ and $g_{1,j}(u)$ by applying its node ID to the polynomials. Since u has received $f_{1,j}(u)$ during the system initiation procedure, it can calculate $K_{1,j} = \frac{\overline{P}_{1,j}(u) - f_{1,j}(u)}{g_{1,j}(u)}$. After recovering the session key, node u can decrypt the message and acquire the new polynomials $h_{1,2,j}(x)$ and $h_{1,3,j}(x)$. For any node $y \in R_{1,j}$, since $g_{1,j}(y) = 0$, it cannot recover the new group key, thus failing to identify the new polynomials.

Step 2. Updating personal key shares for nodes in other groups

Since the polynomials $h_{1,2,j}(x)$ and $h_{1,3,j}(x)$ have been updated, the personal key shares of the nodes in G_2 and G_3 need to be refreshed as well. We adopt the personal key share distribution method proposed in [5] to accomplish the task. Since the sessions in different groups are not synchronized, we assume that G_2 is in session j' and G_3 is in session j''. The sets of revoked nodes can be represented as $R_{2,j'}$ and $R_{3,j''}$, respectively. The *group manager* will broadcast

$$(GM,\ G_2,\ G_3,\ \textit{personal key share update for } G_1 \textit{ in session } j,\ R_{2,j'},\ \overline{Q}_{1,2,j}(x) = g_{2,j'}(x)\cdot h_{1,2,j}(x) + F_{1,2,j}(x),\ R_{3,j''},\ \overline{Q}_{1,3,j}(x) = g_{3,j''}(x)\cdot h_{1,3,j}(x) + F_{1,3,j}(x),\ R_{1,j},\ \textit{digital signature}),$$

where $g_{2,j'}(x) = (x - r_{2,j',1})(x - r_{2,j',2})\cdots(x - r_{2,j',w_{2,j'}})$ and $g_{3,j''}(x) = (x - r_{3,j'',1})(x - r_{3,j'',2})\cdots(x - r_{3,j'',w_{3,j''}})$.

Let us consider a node u in G_2 that has not been revoked. It can calculate $\overline{Q}_{1,2,j}(u)$ and $g_{2,j'}(u)$. Using the pre-distributed value $F_{1,2,j}(u)$, it can calculate $h_{1,2,j}(u)$, which is its new personal key share. For any revoked node y in $R_{2,j'}$, since $g_{2,j'}(y) = 0$, it cannot recover the new personal key share. Similar condition will happen to the nodes in G_3.

5.3. Key updates for newly added or revoked nodes

In this section, we discuss the key update operations for the nodes that are newly introduced into a group or expelled from a group.

Key updates for newly added nodes

We assume that node v joins group G_1 in session j. To enforce backward secrecy, the *group manager* can distribute keys to v through the secure communication channel between the two entities. The secrets will include the group key $K_{1,j}$, the polynomials $h_{1,2,j}(x)$ and $h_{1,3,j}(x)$, the personal key shares $h_{2,1,j'}(v)$ and $h_{3,1,j''}(v)$, and the values of masking functions $f_{1,i}(v)$ $(i = j \cdots m)$, $F_{2,1,i}(v)$ $(i = j' \cdots m)$, and $F_{3,1,i}(v)$ $(i = j'' \cdots m)$.

The polynomials $h_{2,1,j'}(x)$ and $h_{3,1,j''}(x)$ determine the personal key shares of a node in G_1 that are used to encrypt multicast traffic to the members in G_2 and G_3. We argue that these two functions do not have to change. Since node v will only get its personal key shares $h_{2,1,j'}(v)$ and $h_{3,1,j''}(v)$ from the *group manager*, it will not be able to reconstruct the t-degree polynomials, and it cannot calculate the key shares of the other nodes in G_1. Therefore, previous multicast traffic from G_1 to G_2 and G_3 is still safe.

Key updates for newly revoked nodes

We assume that node v is expelled from G_1 in session j. As described in Section 5.2, v cannot recover the new group key or polynomials. v still has the personal key shares $h_{2,1,j'}(v)$ and $h_{3,1,j''}(v)$, and it can use these keys to send false information to the members of G_2 and G_3. To prevent such scenarios from happening, the nodes in G_2 and G_3 will maintain a list of the expelled nodes of G_1 based on $R_{1,j}$ until the new polynomials $h_{2,1,(j'+1)}(x)$ and $h_{3,1,(j''+1)}(x)$ are established. Since v will not get the updated personal key shares, it will not be able to generate false information to mislead the wireless nodes in the network.

The joining and leaving events are the building blocks to describe various member changes in the system. For example, a group switch can be viewed as a leaving action followed by a joining action. The

investigation above focuses on the situations when only one group change happens. The scenarios in which multiple changes happen simultaneously can be handled by adjusting the revocation sets and the polynomials.

From the description in Sections 5.2 and 5.3, we find that when a group change happens, new keys are distributed through true broadcast in a stateless manner, which matches the characteristics of mobile wireless networks such as temporary network disruptions. Although in the paper we put several broadcast messages in separate steps, in the real implementation of the proposed mechanism the information can be merged into one packet to further reduce the communication overhead.

5.4. Generation of keys and polynomials

Since the safety of the proposed mechanism heavily depends on the quality of the secrets and coefficients of polynomials that are generated by pseudo-random number generators, below we discuss the generation of these parameters.

If the network size, group number, and network lifetime can be pre-determined, the secret keys and polynomials can be generated off-line and copied to the group manager during the system initiation procedure. Under this condition, the generation procedure is not restricted by the computation capabilities of wireless nodes, and those strong yet complicated generators can be adopted [42–47]. If the secret keys and polynomials need to be generated by wireless nodes, we can adopt the approaches that are specially designed for resource restricted environments such as sensor networks [48,49] to accomplish this task. To prevent flaws in one pseudo-random number generator from compromising the safety of the whole mechanism, responsibilities of the group manager can be jointly played by multiple nodes, which will be discussed in Section 7.

5.5. Proof of safety

We adopt the definitions of "personal key distribution" and "session key distribution" that are proposed in [5,7] and illustrate the safety of the approach as follows:

Theorem 1. *The proposed approach achieves unconditionally secure personal key distribution with l-revocation capability.*

Proof. Please refer to Appendix A. □

Theorem 2. *The proposed approach achieves unconditionally secure session key distribution with l-revocation capability and zero bit privacy.*

Proof. Please refer to Appendix A. □

5.6. Extending the lifetime

In Section 3 we assume that the network lifetime can be divided into m sessions. Therefore, rekeying operations for the mobile nodes must be conducted when m sessions have expired. A straightforward approach is to let the group manager generate new personal keys and distribute them to individual nodes through unicast with the protection of pair-wise keys. This method will introduce a large amount of communication overhead. In the following section, we will introduce a method based on Shamir's secret sharing in the exponent of a generator r in a cyclic group O. This approach originated in [50,51].

The basic idea of the approach is that group manager will generate and broadcast a group of random numbers at the end of a set of m sessions. The mobile nodes will be able to calculate their new group keys and personal secrets. The approach is based on the assumption that Decision Diffie-Hellman (DDH) [52] is hard. In the following part, we introduce the evolvement of the group keys. The distribution of personal secrets can be derived in a similar way. As a notation, when we have a function $f(x) = a_0 + a_1 x + \cdots + a_t x^t$, the power of generator $r^{f(x)} = (r^{a_0}, r^{a_1}, \ldots, r^{a_t})$.

Distributing group secrets

1. The group manager will generate m l-degree masking functions for each group that can be represented as $f_{i,j}(x)$, $i = 1 \cdots d$, $j = 1 \cdots m$, where the first index represents the group number, and the second index represents the session number. Every node u in group G_w will receive its personal values $f_{w,j}(u)$, $j = 1 \cdots m$.
2. When the group manager needs to distribute the group key of session j for group G_w in the αth set of m sessions, it will generate a random number $v^{\alpha}_{w,j} \in F_q$, and broadcast $r^{v^{\alpha}_{w,j}}$. Every member u in group G_w can use its personal value to calculate $r^{v^{\alpha}_{w,j} \cdot f_{w,j}(u)}$.

3. The group manager generates a group secret $K^{\alpha}_{w,j} \in F_q$, and constructs the function $g^{\alpha}_{w,j}(x)$ based on the revoked node set $R^{\alpha}_{w,j}$. Now the group manager will broadcast:

$$(GM, G_w, \textit{group key update for } G_w \textit{ in session } j \textit{ of round } \alpha, R^{\alpha}_{w,j}, r^{g^{\alpha}_{w,j}(x)\cdot K^{\alpha}_{w,j} + f_{w,j}(x)\cdot v^{\alpha}_{w,j}}, \textit{digital signature}).$$

4. Every node u in G_w can use $r^{v^{\alpha}_{w,j}\cdot f_{w,j}(u)}$ and $r^{g^{\alpha}_{w,j}(u)}$ to calculate $r^{K^{\alpha}_{w,j}}$. $r^{K^{\alpha}_{w,j}}$ will be the group key of session j for group G_w in round α. A revoked node y cannot recover the new key since $y \in R^{\alpha}_{w,j}$ and $g^{\alpha}_{w,j}(y) = 0$.

Using this method, the approach will not be restricted by number of sessions of the network lifetime. Costs to this long-lived key management method are several exponential computations during the secret distribution procedures. Users can choose to adopt the basic approach or approach with extended lifetime based on the special properties of their applications.

5.7. Overhead of the basic approach

In this section, we investigate the storage, computation, and communication overhead of the proposed mechanism at each mobile node.

Although the *group manager* generates many polynomials in the proposed mechanism, the information that every node needs to store will take only a small space. We assume that there are d groups in the network, and the network lifetime is divided into m sessions. Therefore, every node needs to store one group key, $(d-1)$ t-degree polynomials to determine personal key shares for other groups, and $(d-1)$ personal key shares. To enable stateless key recovery, the node will also store $d \times m$ values of the masking functions. Therefore, the total storage overhead of the basic approach is $d \times m + (d-1) \times (t+2) + 1$ numbers in F_q. If $d = 10$, $t = 40$, $m = 200$, and q is 64 bits, the mobile node will consume only 19K Byte storage space to support stateless key update.

The proposed mechanism will cause a limited amount of computation overhead at the mobile nodes. To recover the new keys, a mobile node needs to evaluate a few polynomials to get the keys, conduct a symmetric decryption to recover the new polynomials, and verify a digital signature of the *group manager* to prevent malicious nodes from generating fake key update packets. Most of the operations, except for digital signature verification, can be accomplished efficiently on mobile devices [53]. Verifying digital signatures will not cause a large amount of computation overhead when elliptic curve based approaches are adopted [54–56] and the low frequency of group changes is considered.

The communication overhead of the basic approach is relatively heavy. When a session ends, several broadcast messages for key update will be generated and transmitted throughout the network. Since the messages usually contain multiple polynomials, their size makes it very difficult to piggyback them with data packets. Since packet transmission is more power consuming than computation [57], below we propose a mechanism that will partially sacrifice forward secrecy of keys to drastically reduce the amount of broadcast traffic when new nodes are added into a group.

6. Improving efficiency and safety of the basic approach

In this section, we introduce two mechanisms that can improve the efficiency and safety of the basic approach: the first method will drastically reduce the amount of broadcast traffic during the processing of joining events, and the second method can be adopted to prevent inter-group impersonation.

6.1. Reducing broadcast traffic in joining events

As we have illustrated in Section 5, a majority of the broadcast traffic during key update comes from the distribution of polynomials. If we can avoid these polynomials, we can remove a large part of the communication overhead. When new nodes are added into a group, we need to enforce backward secrecy and guarantee that the new nodes cannot derive old keys based on the current knowledge. Since one way functions satisfy this requirement, we propose to develop a key determination method based on hash functions when processing the joining events.

We use a network with three node groups as an example to illustrate the proposed method. We assume that node u joins group G_1 in session j. The current members of G_1 can determine the new group key by applying a hash function to the old one, as $K_{1,j} = \mathrm{Hash}(K_{1,(j-1)})$. Similar operations will be used to determine the new personal key shares. For example, the new personal key of a node v in G_2 will be $\mathrm{Hash}(h_{1,2,(j-1)}(v))$. v can derive its new

key by applying the hash function to the current key. This method will introduce some extra computation overhead when the current members of G_1 need to calculate the new personal key shares of the nodes in G_2 and G_3. They have to evaluate the polynomial to get $h_{1,2,(j-1)}(v)$ before the hash result can be calculated. Because of the one way property, multiple times of hash calculation can be adopted to derive new keys when the nodes are added into G_1 in multiple sessions.

The key distribution operations to node u will be impacted by this method. To enforce backward secrecy, the *group manager* cannot provide the polynomials $h_{1,2,(j-1)}(x)$ and $h_{1,3,(j-1)}(x)$ to node u. Therefore, u will not be able to derive the personal key shares of the nodes in G_2 and G_3. To enable node u to get a copy of the multicast messages from G_2 and G_3, we adopt the following scheme. After receiving the new group key $K_{1,j}$, u will initiate a localized broadcast to locate another node in G_1 that can calculate the new personal key shares. Whenever that node receives a multicast packet from G_2 or G_3, it will recover the plaintext, encrypt the message with $K_{1,j}$, and forward it to u.

This method can be implemented as follows. When new nodes are added into G_1, the *group manager* will broadcast the message:

$$(GM,\ all\ groups,\ key\ update\ for\ G_1\ in\ session\ (j-1+s)\ based\ on\ session\ (j-1),\ number\ of\ hash\ times\ s,\ digital\ signature).$$

The number of hash times s can be larger than 1 when the nodes are added into G_1 in multiple sessions. We remove polynomials from this broadcast message, and it can be piggyback with data packets. This method reduces the broadcast traffic when it is compared to the basic approach in Section 5. The costs to this improvement are several hash calculations at the mobile nodes.

6.1.1. Discussion and analysis

Deriving new secret keys through one way functions cannot be used to handle leaving events since the revoked nodes will be able to get the new keys. Therefore, in a group switch operation, only the joining group could benefit from the improvement. The key update method can be summarized as

$$\begin{cases} use\ one\ way\ function \\ \quad \text{if the session contains only joining events;} \\ use\ method\ in\ Section\ 5.2 \quad \text{otherwise.} \end{cases}$$

Using one way functions to derive new group keys partially compromises forward secrecy: if a malicious node gets a group key, it has a certain probability to derive subsequent keys. If we divide sessions of network lifetime based on the number of group changes and assume that every change is independent, we may have the following analysis. When a session contains c group changes and each change has 50% probability to be a leaving or joining event, the probability that the session contains only joining events is $1/2^c$. This value determines the chance that we can use one way functions to derive new keys and the probability that a malicious node can get subsequent secrets. Users can choose an appropriate value of c based on the special properties of their applications to achieve a tradeoff between security and efficiency.

6.2. Preventing inter-group impersonation

The distribution of personal key shares enables secret isolation among mobile nodes. Therefore, it is more difficult for a malicious node to impersonate another member in the same group. However, inter-group impersonation is still a threat since the polynomials determining these personal key shares are known by other nodes. As an example, let us consider a network containing three groups G_1, G_2, and G_3, which reside in sessions j, j', and j'', respectively. Both node u in G_2 and all members of G_1 have $h_{1,2,j}(u)$, which is the personal key share of node u. Therefore, a malicious node y in G_1 can impersonate node u by encrypting multicast packets with u's personal key share. Some methods must be adopted to defend against such attacks.

We can distinguish u from the impersonator y based on their knowledge about polynomials. For example, if u belongs to G_2 and y is in G_1, only u knows the polynomial $h_{2,1,\ j'}(x)$. The impersonator will hold only its personal key share $h_{2,1,j'}(y)$. To prevent inter-group impersonation, we propose to design a method that can prove to the receivers in G_1 that the source of the multicast packet knows the polynomial $h_{2,1,j'}(x)$.

The method works as follows. When node u in G_2 wants to send an inter-group multicast message *msg* to G_1, it will first calculate the hash result Hash(*msg*). u will then randomly select a group of nodes from G_1 and encrypt the hash result with each of their personal key shares. These encryption results will be attached to the packet and sent to G_1. When nodes in G_1 receive this packet, they can

use $h_{1,2,j}(u)$ to recover the data. The selected nodes in G_1 will then verify the encrypted hash code. If the hash code is correct, it will remain silent. Otherwise, it will broadcast an alarm packet identifying this error. To prevent a malicious node from sending false alarms, the alarm packet will be protected by the sender's digital signature. All members in G_1 will verify the alarm. If there are more than a threshold number of nodes sending alarms, the packet will be discarded. Otherwise, the packet will be accepted. In this way, if the packet is really from node u, it will pass the check and no additional traffic will be introduced. The impersonator y will not be able to generate the encryption results since it does not know the polynomial $h_{2,1,j'}(x)$.

The nodes in G_1 whose personal keys are selected by u must be randomly determined to prevent a malicious node from manipulating the result. Here we propose one approach to the problem. The *group manager* will calculate a bloom filter [58] for every group based on the identities of the current members in it. The bloom filters will be distributed to all nodes in the network. When u wants to send a message *msg* to G_1, it will use the hash result of the message Hash(*msg*) as seed to generate a sequence of node identities. The first b identities that satisfy the bloom filter of G_1 will be selected, and their personal key shares will be used. Since a bloom filter has false positive errors, we can adjust the length of the filter and the value of b to control the expected number of nodes in G_1 that can verify the encryption.

Using this method, the contents of an inter-group multicast packet can be illustrated as follows:

$$(u,\ G_1,\ \textit{this is a data packet},\ E_{h_{1,2,j}(u)}(msg),\\ r_1,\ r_2,\ldots,\ r_b\ E_{h_{2,1,j'(r_1)}}(hash(msg)),\\ E_{h_{2,1,j'(r_2)}}(hash(msg)),\ldots,E_{h_{2,1,j'(r_b)}}(hash(msg))),$$

where r_1 to r_b are the first b identities that are generated based on Hash(*msg*) and satisfy the bloom filter of G_1. Every node in G_1 can calculate the personal key share $h_{1,2,j}(u)$ of node u to recover the message. The nodes can verify the selected nodes r_1 to r_b by regenerating the identity sequence. The nodes r_1 to r_b (if they are in G_1) can verify the encryption and prove that u knows the polynomial $h_{2,1,j'}(x)$. Using this method, we can mitigate the threats of inter-group impersonation.

6.2.1. Discussion and analysis

In this section, we study the extra communication overhead that is introduced by the impersonation prevention method. In Appendix B, we investigate the false positive and false negative alarm rate of the method.

The extra communication overhead includes node *ID*s and hash codes. The analysis in Appendix B will show that when $b \approx 10$, we have a very low false alarm rate. Therefore, if we use 8 Byte encrypted hash code and 2 Byte node *ID*, we will introduce 100 Byte overhead into the packet. If the data packets are 1500 Byte long, it will consume about 7% of the bandwidth. The mobile nodes can further reduce overhead by sending only a part of the encrypted hash codes (e.g. only the first 4 Bytes) to achieve a tradeoff between security and efficiency.

Please refer to Appendix B for the analysis of false positive and false negative alarm rate of the method.

7. Discussions

7.1. Advantages of stateless key distribution

The key refreshment approach described in previous sections has the stateless property: recovering the latest group keys, polynomials, and personal key shares does not depend on the knowledge of keys for previous sessions. This feature is especially important for mobile wireless networks since the mobile nodes may miss some of the key update packets because of various reasons. If the keys have to be recovered sequentially, more and more nodes will not be able to decrypt the multicast traffic unless they initiate individual requests to get the latest secrets from the *group manager*.

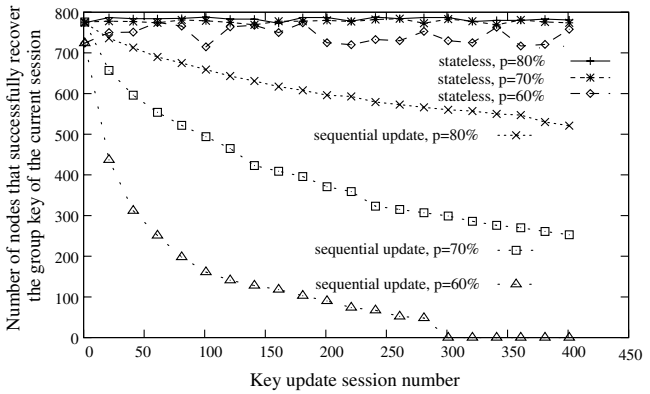
Simulation has been conducted to demonstrate the improvements. In the simulation environment, we assume that 900 nodes are randomly and roughly uniformly deployed in a 2000 m × 2000 m square area. A unit disk communication model [59] is adopted, and different communication ranges from 85 m to 100 m are considered. We assume that the *group manager* is located at the center of the area, and it will periodically broadcast key update packets. Two approaches are compared: the method proposed in SASN'05 [37] in which the group keys have to be recovered sequentially, and the stateless approach presented in Section 5. To demonstrate the impacts of the loss of key update packets, the mobile nodes that do not receive the current session keys will not send individual requests to the *manager*. However, they will continue to rebroadcast key update packets in subsequent sessions.

We assume that a mobile node has probability p to successfully receive a broadcast packet sent by a neighbor. Every node will rebroadcast the received key update packets exactly once. Since the results in [60] have shown that most links have non-bursty loss patterns, we assume that the receiving events are independent. We examine different values of p from 60% to 80%. The simulation focuses on the nodes that are five or more hops away from the *manager*. Two factors, the probability p and the average node connectivity c, and their relationship to the fraction of wireless nodes that can successfully recover the group key of current session, are of special interest.

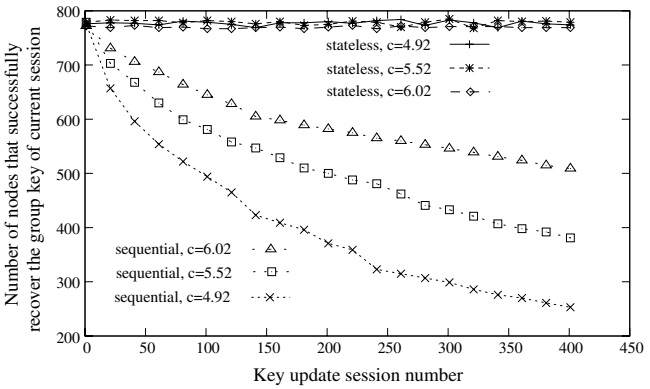
The simulation results are illustrated in Fig. 1. In Fig. 1a, the average connectivity of nodes is 4.92. We compare the two approaches under different values of p. We find that the proposed mechanism allows a majority of wireless nodes to recover the latest group keys. On the contrary, packet loss puts a severe impact on the stateful method. Similar results are shown in Fig. 1b when p is 70% and the average connectivity c changes.

7.2. Comparison to existing approaches

In this section, we examine the special properties of five approaches to secure group communication and key management: (1) approach using public/private key pairs [61]; (2) SASN'05 approach [37]; (3) approach proposed in this paper; (4) CKDS [32]; and (5) GKMPAN [33]. These approaches can be divided into two groups. Group one contains the first three approaches. They are all specially designed (or can be directly applied) to support secure communication for both intra and inter-group multicast. Comparison among them will demonstrate the advantages of the proposed approach. The other two methods, CKDS and GKMPAN, are designed to support secure communication and



(a) Improvements in key distribution when the receiving probability p changes.



(b) Improvements in key distribution when the average node connectivity c changes.

Fig. 1. Improvements in key distribution that are brought by the stateless property.

Table 2
Comparison of properties among five approaches

	Pub/Pri approach	CKDS	GKMPAN	SASN'05 approach	Proposed mechanism
Encryption of multicast traffic	Asymmetric encryption	Symmetric encryption			
Need loose clock synchronization	No	No	Yes	No	No
Support inter-group multicast	Yes	Need updates	Need updates	Yes	Yes
Support stateless key updates	No	No	Partial	No	Yes
Support multiple node revocation	Restricted by union of KEKs held by revoked nodes				Restricted by degree of polynomials
Support personal key distribution	No	No	No	Yes	Yes
Need a group manager	Yes	Yes	Yes	Yes	Yes

key management in a single group. Special updates must be developed to tailor them for multi-group applications. Since these two approaches are highly efficient during key update operations, we plan to explore the potential of integrating them with the proposed approach to build a more efficient and secure key management mechanism for multi-group environments. Below we first provide a short introduction of CKDS and GKMPAN, then examine the approaches from different perspectives.

In CKDS, all group members share the same traffic encryption key (TEK). A total number of $(k+m)$ key encryption keys (KEK) are generated for secret update. Using exclusion-basis systems (EBS), every node will be assigned to a unique set of k KEKs. When a node is revoked, the *group manager* uses the m keys that are unknown to the revoked member to encrypt the new TEK and KEKs. To reduce key update overhead, members sharing the same KEKs are organized into an m-dimensional space and special multicast schemes are developed for key refreshment.

In GKMPAN, a total number of l KEKs are generated. Every group member will receive m KEKs, the identities of which are determined by the node *ID*. Authenticity of the revocation message is protected through delayed secret disclosure using TESLA. When a node is revoked, a KEK that is known to the largest number of remaining nodes will be used to distribute the new TEK. Localized distribution of TEK through shared KEKs among neighbors is then adopted to reduce communication overhead. KEKs are updated based on their old values and an intermediate secret.

As we have illustrated in Section 5.2, a majority of communication overhead during key refreshment comes from the coefficients of polynomials. Therefore, the efficiency of the proposed approach can be further improved if the following techniques of CKDS and GKMPAN can be integrated without compromising the properties such as stateless key refreshment and personal key distribution: (1) grouping nodes based on their shared secrets to provide guidance for key update through unicast or directed propagation; (2) deriving new personal key shares based on a limited number of intermediate secrets that are much fewer than the coefficients of polynomials.

Properties of the five approaches are listed in Table 2. If the KEKs that a node holds are chosen from a pre-determined set, the approach's capability to support revocation of multiple members simultaneously will be restricted by the union of KEKs held by these nodes. Since CKDS and GKMPAN are designed for single group communication, special updates must be developed to support inter-group multicast.

7.3. Security and robustness

We have investigated the following security and robustness problems of the proposed mechanism.

Generating group managers

The *group managers* play an important role in the proposed mechanism: they are in charge of generating and distributing polynomials and group keys. In addition to the capability of generating secure secrets, other features of the mobile nodes (such as the trustworthiness value, power level, and compu-

tation capability) should also be considered during the generation procedures.

If a pre-distributed infrastructure exists in the wireless network, the manager generation procedure can take advantage of those special nodes. For example, in the military operation described in Section 1, the officers with the highest rank in the soldier groups of each country can serve as the *managers*. As another example, in a cellular–ad hoc integrated system, the base stations can manage the membership and generate new keys for every group.

In a self-organized environment such as pure ad hoc networks, a more complicated manager election or generation procedure must be adopted. One possible solution is to adopt a variation of the secure leader election algorithms for ad hoc networks [62]. The mobile nodes use a preference function that integrates multiple decision factors to represent the desirability of a candidate. The node that receives the most "votes" will become the manager.

Distributing key management overhead

For the simplicity of presentation, we have assumed a single *group manager* in the paper. To improve robustness of the proposed mechanism and avoid single point of failure, distributed key management can be adopted. Multiple *managers* may perform equally or form a hierarchy to control the key distribution and update procedures for a group. When a joining or a leaving event happens, they can generate the new keys in a collaborative manner. Another advantage is that a wireless node has a higher probability to receive the key update packets from a *manager* locally, which will reduce the communication overhead caused by control traffic. The organization of the *managers* can benefit from previous research in distributed systems [31,17]. For example, the *group managers* can form an overlay structure [27]. To improve the robustness of the method to node mobility, location based routing through Distributed Hash Table (DHT) can be adopted.

Defending against collusive attacks

Wireless nodes in the network may collude to get illegal access to multicast traffic. The proposed mechanism is robust against collusive attacks from the malicious nodes in the same group. Mechanisms to defend against inter-group collusive attacks will be investigated in future work.

Malicious nodes in the same group can benefit from collusion by reconstructing polynomials of other groups. They can recover the personal key shares of innocent members and get illegal access to the multicast traffic that is not destined to them. Since a t-degree polynomial is robust against the collusion of up to t compromised members, we can adjust the choice of this parameter based on the security requirements to balance the safety of the mechanism and the storage, computation, and communication overhead. Similar analysis can be applied to the parameter that defines the largest number of revoked nodes from a group. Different values can be adopted by different groups based on their security requirements and model of group changes.

8. Conclusions and future extensions

Secure multicast has become an important component of many applications in mobile wireless networks. In this paper, we focus on key distribution and update for secure inter-group communication. The proposed mechanism adopts polynomials to support the distribution of personal key shares and employs stateless secret update to achieve efficient key refreshment. Special methods are designed to reduce the communication overhead during group changes. Compared to previous approaches, it reduces the processing overhead for the multicast traffic by switching from asymmetric encryption to symmetric encryption. It becomes more difficult for an attacker to impersonate another entity in the network. The proposed mechanism introduces only a small amount of storage and computation overhead to the mobile nodes.

While the applications in wireless networks are investigated in detail, the ideas presented here can be extended to any other inter-group communication environments in which frequent group changes are expected and continuous network connections cannot be guaranteed. Furthermore, the revocation capability can be adopted to temporarily isolate some entities during distribution of sensitive information.

In the future we plan to investigate the relationships among the system parameters l, t, and m and their impacts on the scalability, lifetime, and security of the proposed mechanism. The lifetime extension schemes [7] will be integrated with the proposed mechanism. Motion prediction methods can be adopted to assist the formation of subgroups and

further reduce the key management overhead. The results will lead to a more robust and efficient key distribution protocol for secure intra and inter-group communication in various networking environments.

Appendix A. Proof of safety of the approach

Theorem 1. *The proposed approach achieves unconditionally secure personal key distribution with l-revocation capability.*

Proof. Based on the definition in [5,7], we prove that the personal key distribution method has the following properties:

(a) For any non-revoked node u, its personal key can be jointly determined by pre-distributed information and the broadcast packet. As illustrated in Section 5.2 step 2, a node u in group G_i can use $\overline{Q}_{i\prime,i,j\prime}(u)$, $g_{i,j\prime}(u)$, and $F_{i\prime,i,j\prime}(u)$ to determine its new personal key share $h_{i\prime,i,j\prime}(u)$.

(b) For any set of B nodes in one group, $|B| \leqslant t$, and any node $u \notin B$ in the same group, the members in B are not able to learn the pre-distributed information of node u. Although the members in B are aware of the polynomials $\overline{Q}(x)$ and $g(x)$, they have at most t points on the new personal key distribution function $h(x)$, which is a t-degree polynomial. Therefore, they cannot use Lagrange interpolation to recover the polynomial and figure out the pre-distributed secrets of node u.

(c) The new personal key share cannot be recovered by either the pre-distributed information or the broadcast packet alone. Since the coefficients of polynomials $h(x)$ and $F(x)$ are randomly picked, the new personal secret cannot be determined only by broadcast packets or pre-distributed information.

(d) The proposed approach has l-revocation capability. As we have illustrated in Section 5.2 step 2, any node that has not been revoked from the current group will be able to recover its new personal key. On the contrary, for any set of R revoked nodes from one group, $|R| \leqslant l$, they can have at most l points on $F(x)$. Since the function $g(x)$ is generated based on their IDs, any t-degree function can be the function $h(x)$, and the revoked nodes cannot derive their new personal secrets. □

Theorem 2. *The proposed approach achieves unconditionally secure session key distribution with l-revocation capability and zero bit privacy.*

Proof. Based on the definition in [5,7], we prove that the session key distribution method has the following properties:

(a) For any non-revoked node u in group G_i, the new group key $K_{i,j}$ can be jointly determined by pre-distributed information and the broadcast packet. As shown in Section 5.2 step 1, node u can use $\overline{P}_{i,j}(u)$, $g_{i,j\prime}(u)$, and $f_{i,j\prime}(u)$ to calculate the new group key.

(b) For a non-revoked node u in group G_i, the proposed approach does not protect the privacy of its pre-distributed information $f_{i,j\prime}(u)$ from the other non-revoked nodes in the same group. For another non-revoked node v in the same group, it can use $\overline{P}_{i,j}(x)$, $g_{i,j\prime}(x)$, and $K_{i,j}$ to figure out $f_{i,j\prime}(x)$, thus calculating the pre-distributed information at every node. However, the revoked nodes will not get this information. To achieve a better privacy of the pre-distributed information, we can decompose the group key $K_{i,j}$ as the sum of two t-degree polynomials $K_{i,j} = p_{i,j\prime}(x) + q_{i,j\prime}(x)$, as suggested in [5].

(c) The new group key cannot be determined from the broadcast packet or pre-distributed information alone. Since the group keys $K_{i,j}$ and coefficients of polynomials $f_{i,j\prime}(x)$ are all randomly generated, the group secrets cannot be determined only by broadcast packets or the pre-distributed information.

(d) The proposed approach has l-revocation capability. As we have illustrated in Section 5.2 step 1, any node that has not been revoked will be able to recover the new group key. On the contrary, for any set of R revoked nodes from one group, $|R| \leqslant l$, they can have at most l points on $f(x)$. Since the function $g(x)$ is generated based on their IDs, any value can be $K_{i,j}$, and the revoked nodes cannot derive the new group key. □

Appendix B. False alarm rate of impersonation prevention method

To analyze the false alarm rate of the impersonation prevention method, we make the following assumptions. We assume that the hit rate of bloom filters is p. Since b identities are attached to a packet,

on average there will be about $c = b \times p$ nodes in the target group that can verify the encrypted hash codes. We assume that there are $a + i$ nodes in the target group, in which a nodes are malicious, and i nodes are innocent and will follow the protocol. We also assume that the threshold is t: if there are at least t nodes claiming incorrectly encrypted hash codes, the packet will be rejected. While we do not consider the probability that a node fails to receive the data packet, it can be easily integrated into the parameter p.

False positive alarm rate. To persuade all group members to discard a good packet, there must be at least t malicious nodes in the c chosen identities when they send the alarm packets. If every node in the target group has the same probability to be chosen, we have (C_c^{a+i}) combinations to choose the nodes. The probability that at least t malicious nodes are chosen will be: $\frac{\sum_{j=t}^{j=\min(a,c)}(C_j^a)\cdot(C_{c-j}^i)}{(C_c^{a+i})}$.

False negative alarm rate. To persuade all group members to accept a bad packet, we can have at most $t-1$ innocent nodes in the chosen identities sending alarms. If every node in the target group has the same probability to be chosen, the probability that at most $t-1$ innocent nodes are chosen is: $\frac{\sum_{j=\max(0,c-a)}^{j=t-1}(C_{c-j}^a)\cdot(C_j^i)}{(C_c^{a+i})}$.

Some numerical results. We assume that the target group has $a = 10$ malicious nodes and $i = 90$ innocent nodes. When $b = 10$ and the hit rate of a bloom filter is $p = 70\%$, the expected number of nodes that can verify the hash code is $c = 7$. When $t = 3$, the false positive alarm rate is 2% and the false negative alarm rate is 0.006%. When $t = 4$, the false positive alarm rate is 0.16% and the false negative alarm rate is 0.16%. From the results, we find that a small number of t would provide a decent detection accuracy when a majority of the nodes in the group are innocent.

References

[1] X. Chen, J. Wu, Multicasting techniques in mobile ad hoc networks, in: The Handbook of Ad Hoc Wireless Networks, CRC Press Inc., 2003, pp. 25–40.

[2] D. Wallner, E. Harder, R. Agee, Key management for multicast: issues and architectures, RFC 2627, 1999.

[3] C. Wong, M. Gouda, S. Lam, Secure group communications using key graphs, IEEE/ACM Transactions on Networking 8 (1) (2000) 16–30.

[4] C. Wolf, Efficient public key generation for hfe and variations, in: Cryptographic Algorithms and Their Uses, 2004, pp. 78–93.

[5] D. Liu, P. Ning, K. Sun, Efficient self-healing group key distribution with revocation capability, in: Proc. of ACM Conference on Computer and Communications Security, 2003, pp. 231–240.

[6] D. Naor, M. Naor, J. Lotspiech, Revocation and tracing schemes for stateless receivers, in: CRYPTO'01, LNCS 2139, 2001, pp. 41–62.

[7] J. Staddon, S. Miner, M. Franklin, D. Balfanz, M. Malkin, D. Dean, Self-healing key distribution with revocation, in: Proc. of IEEE Symposium on Security and Privacy, 2002.

[8] H. Harney, C. Muckenhirn, Group key management protocol (GKMP) architecture, RFC 2094, 1999.

[9] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, B. Pinkas, Multicast security: a taxonomy and some efficient constructions, in: Proceedings of IEEE INFOCOM, 1999, pp. 708–716.

[10] D. McGrew, A. Sherman, Key establishment in large dynamic groups using oneway function trees, Tech. rep. no. 0755, Network Associates, Inc., 1998.

[11] M. Waldvogel, G. Caronni, D. Sun, N. Weiler, B. Plattner, The Versakey framework: versatile group key management, IEEE JSAC, Special Issue on Middleware 17 (9) (1999) 1614–1631.

[12] R. Canetti, T. Malkin, K. Nissim, Efficient communication-storage tradeoffs for multicast encryption, in: Advances in Cryptology – EUROCRYPT, 1999, pp. 459–474.

[13] A. Perrig, D. Song, J. Tygar, Elk, a new protocol for efficient large-group key distribution, in: Proc. of IEEE Symposium on Security and Privacy, 2001.

[14] S. Mittra, Iolus: a framework for scalable secure multicasting, in: ACM SIGCOMM, 1997, pp. 277–288.

[15] L. Dondeti, S. Mukherjee, A. Samal, Scalable secure one-to-many group communication using dual encryption, Computer Communications 23 (17) (1999) 1681–1701.

[16] R. Molva, A. Pannetrat, Scalable multicast security in dynamic groups, in: Proc. of ACM CCS, 1999, pp. 101–112.

[17] S. Rafaeli, D. Hutchison, Hydra: A decentralized group key management, in: Proc. of IEEE International Enterprise Security Workshop, 2002.

[18] B. Briscoe, Marks: multicast key management using arbitrarily revealed key sequences, in: Proc. of International Workshop on Networked Group Communication, 1999.

[19] S. Setia, S. Koussih, S. Jajodia, Kronos: a scalable group re-keying approach for secure multicast, in: Proc. of IEEE Symposium on Security and Privacy, 2000.

[20] R. Pietro, L. Mancini, Y. Law, S. Etalle, P. Havinga, LKHW: a directed diffusion-based secure multicast scheme for wireless sensor networks, in: Proc. of IEEE International Conference on Parallel Processing Workshops, 2003.

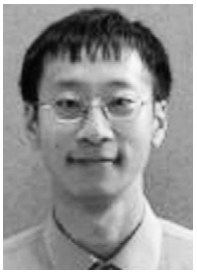
[21] T. Kaya, G. Lin, G. Noubir, A. Yilmaz, Secure multicast groups on ad hoc networks, in: Proc. of ACM workshop on security of ad hoc and sensor networks, 2003, pp. 94–102.

[22] L. Lazos, R. Poovendran, Energy-aware secure multicast communication in ad-hoc networks using geographic location information, in: Proc. of IEEE International Conference on Acoustics Speech and Signal Processing, 2003.

[23] L. Lazos, R. Poovendran, Location-aware secure wireless multicast in ad-hoc networks under heterogeneous pathloss, Technical report uweetr-2003-0012, University of Washington, 2003.

[24] L. Ji, M. Corson, Differential destination multicast – a MANET multicast routing protocol for small groups, in: Proc. of IEEE INFOCOM, 2001.
[25] L. Ji, M. Corson, Explicit multicasting for mobile ad hoc networks, Mobile Networks and Applications 8 (5) (2003) 535–549.
[26] K. Chen, K. Nahrstedt, Effective location-guided tree construction algorithms for small group multicast in MANET, in: Proc. of IEEE INFOCOM, 2002, pp. 1180–1189.
[27] C. Gui, P. Mohapatra, Efficient overlay multicast for mobile ad hoc networks, in: Proc. of IEEE Wireless Communications and Networking Conference (WCNC), 2003.
[28] S. Mäki, T. Aura, M. Hietalahti, Robust membership management for ad-hoc groups, in: Proc. of Nordic Workshop on Secure IT Systems, 2000.
[29] A. Yasinsac, V. Thakur, S. Carter, I. Cubukcu, A family of protocols for group key generation in ad hoc networks, in: Proc. of IASTED International Conference on Communications and Computer Networks, 2002, pp. 183–187.
[30] D. Bruschi, E. Rosti, Secure multicast in wireless networks of mobile hosts: protocols and issues, Mobile Networks and Applications 7 (6) (2002) 503–511.
[31] B. DeCleene, L. Dondeti, S. Griffin, T. Hardjono, D. Kiwior, J. Kurose, D. Towsley, S. Vasudevan, C. Zhang, Secure group communications for wireless networks, in: IEEE MILCOM, 2001.
[32] M. Moharrum, R. Mukkamala, M. Eltoweissy, CKDS: an efficient combinatorial key distribution scheme for wireless ad-hoc networks, in: Proc. of IEEE International Conference on Performance, Computing, and Communications, 2004, pp. 631–636.
[33] S. Zhu, S. Setia, S. Xu, S. Jajodia, GKMPAN: an efficient group rekeying scheme for secure multicast in ad-hoc networks, in: Proc. of International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004, pp. 42–51.
[34] C. Wong, S. Lam, Keystone: a group key management service, in: Proceedings of International Conference on Telecommunication (ICT), 2000.
[35] A. Shamir, How to share a secret, Communications of the ACM 22 (1979) 612–613.
[36] S. More, M. Malkin, J. Staddon, D. Balfanz, Sliding-window self-healing key distribution, in: Proc. of ACM workshop on Survivable and self-regenerative systems, 2003, pp. 82–90.
[37] W. Wang, B. Bhargava, Key distribution and update for secure inter-group multicast communication, in: Proceedings of ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN), in conjunction with ACM CCS, 2005, pp. 43–52.
[38] R. Pickholtz, D. Schilling, L. Milstein, Theory of spread spectrum communications – a tutorial, IEEE Transactions on Communication COM-30 (5) (1982) 855–884.
[39] V. Gupta, S. Krishnamurthy, M. Faloutsos, Denial of service attacks at the MAC layer in wireless ad hoc networks, in: Proc. of Milcom, 2002.
[40] P. Bjorklund, P. Varbrand, D. Yuan, Resource optimization of spatial TDMA in ad hoc radio networks: a column generation approach, in: Proceedings of IEEE INFOCOM, 2003.
[41] M. Steiner, G. Tsudik, M. Waidner, Key agreement in dynamic peer groups, IEEE Transactions on Parallel and Distributed Systems 11 (8) (2000) 769–780.
[42] L. Deng, H. Xu, A system of high-dimensional, efficient long-cycle and portable uniform random number generators, ACM Transactions on Modeling and Computer Simulation (TOMACS) 13 (4) (2003) 299–309.
[43] M. Goresky, A. Klapper, Efficient multiply-with-carry random number generators with maximal period, ACM Transactions on Modeling and Computer Simulation (TOMACS) 13 (4) (2003) 310–321.
[44] M. Mascagni, A. Srinivasan, Algorithm 806: Sprng: a scalable library for pseudorandom number generation, ACM Transactions on Mathematical Software (TOMS) 26 (3) (2000) 436–461.
[45] F. Panneton, P. L'ecuyer, On the xorshift random number generators, ACM Transactions on Modeling and Computer Simulation (TOMACS) 15 (4) (2005) 346–361.
[46] F. Panneton, P. L'ecuyer, M. Matsumoto, Improved long-period generators based on linear recurrences modulo 2, ACM Transactions on Mathematical Software 32 (1) (2006) 1–16.
[47] A. Seznec, N. Sendrier, Havege: a user-level software heuristic for generating empirically strong random numbers, ACM Transactions on Modeling and Computer Simulation (TOMACS) 13 (4) (2003) 334–346.
[48] W. Che, H. Deng, X. Tan, J. Wang, Scheme of truly random number generator application in rfid tag, in: P. Cole, D. Ranasinghe (Eds.), Networked RFID Systems and Lightweight Cryptography Raising Barriers to Product Counterfeiting, Springer, 2007.
[49] D. Seetharam, S. Rhee, An efficient pseudorandom number generator for low-power sensor networks [wireless networks], in: Proc. of IEEE International Conference on Local Computer Networks, 2004, pp. 560–562.
[50] P. Feldman, A practical scheme for non-interactive secret sharing, in: Proc. of IEEE Symposium on Foundations of Computer Science, 1987, pp. 427–437.
[51] M. Naor, B. Pinkas, Efficient trace and revoke schemes, in: Proceedings of Financial Cryptography, 2000, pp. 1–20.
[52] D. Boneh, The decision Diffie-Hellman problem, in: Proceedings of the Third Algorithmic Number Theory Symposium, 1998, pp. 48–63.
[53] P. Ni, Z. Li, Energy cost analysis of IPSec on handheld devices, Microprocessors and Microsystems, special issue on Secure Computing Platform 28 (10) (2004) 585–594.
[54] A. Liu, P. Ning, Tinyecc: Elliptic curve cryptography for sensor networks (version 0.2), Released September 29, 2006 at <http://discovery.csc.ncsu.edu/ software/TinyECC/>.
[55] D. Malan, M. Welsh, M. Smith, A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography, in: First IEEE International Conference on Sensor and Ad Hoc Communications and Network, 2004.
[56] H. Wang, B. Sheng, Q. Li, Elliptic curve cryptography based access control in sensor networks, International Journal of Sensor Networks 1 (3/4) (2006) 127–137.
[57] L. Ferrigno, S. Marano, V. Paciello, A. Pietrosanto, Balancing computational and transmission power consumption in wireless image sensor networks, in: Proc. of IEEE VECIMS, 2005.

[58] B. Bloom, Space/time trade-offs in hash coding with allowable errors, Communications of the ACM 13 (7) (1970) 422–426.
[59] B. Clark, C. Colbourn, D. Johnson, Unit disk graphs, Discrete Mathematics 86 (1–3) (1990) 165–177.
[60] D. Aguayo, J. Bicket, S. Biswas, G. Judd, R. Morris, Link-level measurements from an 802.11b mesh network, in: Proc. of SIGCOMM, 2004.
[61] S. Yi, P. Naldurg, R. Kravets, Security-aware ad hoc routing for wireless networks, in: Proc. of the Second ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2001.
[62] S. Vasudevan, B. DeCleene, N. Immerman, J. Kurose, D. Towsley, Secure leader election algorithms for wireless ad hoc networks, in: Proc. of IEEE DARPA Information Survivability Conference and Exposition (DISCEX), 2003.



Weichao Wang is an Assistant Professor in the Department of Software and Information Systems at the University of North Carolina, Charlotte. He received a bachelor degree in Computer Science from Tsinghua University, China in 1998, and a PhD in Computer Science from Purdue University in 2005. His research interests include designing protocols and mechanisms to secure pervasive systems, especially the resource-restraint wireless networks. He is a member of IEEE, ACM, and ASEE.

Tylor Stransky is a graduate student in the Department of Electrical Engineering and Computer Science at the University of Kansas. His research efforts focus on security protocols in mobile wireless networks and vehicular ad hoc networks.