2007

Enabling Confidentiality for Group Communication in Wireless Mesh Networks

Jing Dong

Cristina Nita-Rotaru
Purdue University, crisn@cs.purdue.edu

Report Number:
07-010

ENABLING CONFIDENTIALITY FOR GROUP COMUNNICATION IN WIRELESS MESH NETWORKS

Jing Dong
Cristina Nita-Rotaru

Department of Computer Science
Purdue University
West Lafayette, IN 47907

CSD TR #07-010
April 2007

Enabling Confidentiality for Group Communication in Wireless Mesh Networks

Jing Dong Cristina Nita-Rotaru
Department of Computer Science, Purdue University
305 N. University St., West Lafayette, IN 47907 USA
{dongj,crisn}@cs.purdue.edu

Abstract— **Wireless mesh networks (WMNs) have emerged as a promising technology for providing low-cost community wireless services. Despite recent advancement in securing wireless networks, the problem of secure group communication on wireless networks has received relatively little attention. Characteristics specific to WMNs, such as limited communication range and high link error rate, raise unique challenges in designing such protocols.**

In this paper we focus on providing data confidentiality for group communications on WMNs. First, we propose W-LKH, a protocol that combines centralized key management and reliable key delivery, to address the less robust communication present in wireless networks. Next, we introduce WSOM, a new protocol framework designed specifically for the WMNs to overcome the performance and security limitations of W-LKH. Simulation results show that all of the proposed protocols can provide good performance to the upper layer applications, while the WSOM protocols incur smaller overhead and are more responsive than W-LKH. Finally, we suggest the applicability of each of the proposed protocols under different application requirements.

I. Introduction

Wireless mesh networks (WMNs) consist of a set of fixed wireless routers that form a multi-hop wireless backbone and a set of wireless clients. In recent years, WMNs have become a promising key technology for providing low-cost high-bandwidth community wireless services. Given the community oriented nature of the WMNs, group applications such as real time conferencing, multimedia content broadcasting, and file sharing are an important class of applications in the WMN environment. As with other types of wireless applications, the openness of the wireless environment makes security a critical concern in deploying such group applications.

The problem of securing group communication in traditional network environments has received significant attention, such as the IP multicast [1], [2], [3] and overlay multicast networks [4], [5], [6]. However, the constraints and peculiarities of the wireless medium are not considered by the protocols designed for wired networks, preventing them from being directly applied in the wireless environment. For example, the limited range of the wireless signal mandates multi-hop delivery of both unicast and multicast data, and hence precludes the possibility of direct communication between nodes that some of the protocols for wired networks rely on. The limited communication range also necessitates the participation of non-group members in the data forwarding protocol which is absent in the wired networks. Furthermore, unreliable wireless links make loss recovery an essential component of the protocol, while scarce network bandwidth resource demands keeping the overhead low to be a top priority in the protocol design.

The secure group communication problem and the related key management problem have also been studied for wireless sensor networks (WSNs) [7], [8], [9] and mobile ad hoc networks (MANETs) [10], [11], [12]. However, services for WSNs or MANETs were designed to sustain severe computation power, storage, mobility and energy constraints, and as a result, they have limited scalability and robustness. As WMNs have less restrictive constraints, they create opportunities for designing more scalable and robust protocols.

In this paper, we focus on the problem of ensuring data confidentiality for group communications on WMNs. We consider single-source group applications where a single source disseminates data to a dynamically changing set of receivers. The main contributions of this paper are:

- We study the design space for secure group communication protocols on WMNs. We propose W-LKH, a centralized membership protocol that combines the well-known protocol LKH [1] with reliable key delivery mechanisms, and a new protocol framework WSOM with decentralized membership management that overcomes the limitations inherent in W-LKH.
- We compare all the proposed protocols analytically by examining the overhead and their responsiveness to the upper layer applications.
- We validate our design experimentally with extensive simulations based on the *ns* simulator [13]. Simulation results show that all of the proposed protocols can provide good performance to the upper layer applications, and with proper optimization, the WSOM based protocols incur less overhead and are more responsive than W-LKH. We also demonstrate that reliable key delivery is critical on WMNs.
- We discuss the applicability of each of the proposed protocols under different application requirements.

The rest of the paper is organized as follows. We first present related work in Section II. We then describe the network and security model we consider in this work in Section III. We discuss the design goals and main challenges in Section IV. Sections V and VI describe the W-LKH and the WSOM protocols. We present the analytical and experimental comparison results for the proposed protocols in Section VII and VIII, respectively. We conclude our paper in Section IX.

II. Related Work

In this section, we review the existing work on the secure group communication problem for both wired networks and wireless networks.

The problem of secure group communication has received significant attention for wired networks. In the context of the IP multicast environment, the main focus was primarily to reduce the computation overhead of key updates at the source. The most well-known protocols are LKH [1] and its variants [14], [15], [16], [17], [18]. The problem of key transportation was studied both in the context of IP multicast [15], [18] and more recently in overlay networks [4], [19], [20], [21], [6]. In the latter case, overlays were used as a more realistic structure to deliver keys due to lack of deployment of IP multicast. However, none of these protocols considered the wireless specific constraints and challenges, such as limited bandwidth, multi-hop communication through possible non-member nodes and higher link error rates. Thus these protocols are not directly applicable to WMNs.

In the wireless environment, work related to secure group communication focused on securing the multicast protocols and key management. The problem of securing the multicast protocol [22] is complementary to providing data confidentiality, as it focuses only on the control message and not the data traffic. Several researchers [7], [8], [9] proposed schemes for establishing pair-wise symmetric keys for sensor networks and wireless ad hoc networks. These schemes focus on secure pair-wise communications instead of group communications. Zhu et al [10] proposed GKMPAN, a secure group communication that uses symmetric keys to distribute the common group key for data encryption among group members. The main focus of GKMPAN is on handling member revocations, instead of the potentially much more frequent member join and leave events. Moreover, GKMPAN requires key pre-distribution, which is not always available, and a broadcast authentication scheme, such as TESLA [23], which has the additional requirement of time synchronization. Balachandran et al proposed CRTDH [11] for secure group communication which relies on the Chinese Remainder Theorem and the Diffie-Hellman group key agreement for establishing group keys. The shortcoming of the CRTDH is that every group join and leave event requires the number of messages being delivered be proportional to the group size, hence it is not scalable in a wireless environment where the bandwidth resource is scarce. Kaya et al [12] present a secure multicast scheme for mobile ad hoc networks. Instead, our protocols focus on the WMNs which allow for further optimizations by exploiting the static network topology.

III. Network and Security Model

A. Network Model

Our target network environment is WMNs, where nodes are assumed to be static and communicate through multi-hop wireless links. Possible link and node failures are allowed in the network. We focus on the group communication scenario where one data source broadcasts data to a set of receivers (group members) that can dynamically change throughout the broadcast session. We assume a tree-based on-demand multicast protocol is used to deliver the group data. For concreteness, we consider the well-known MAODV [24] protocol in presenting our protocols.

Due to the multi-hop communication of WMNs, it is necessary that non-group members participate in the multicast tree construction. Hence, the multicast tree contains two types of nodes: *member nodes* and *non-member nodes*. Member nodes are nodes on the tree that are also members of the multicast group. The non-member nodes are not part of the multicast group but rather act as routers that help to connect the member nodes. We refer to the nodes in the multicast tree (both member and non-member) as *tree nodes*. Nodes that are not part of the multicast tree are called *non-tree* nodes.

B. Security and Adversarial Model

Our focus is on providing confidentiality of the data from outside adversaries, where an outsider is any non-member node, including non-member nodes that are on the multicast tree. Nodes that have left the group are also considered outsiders. We assume that the current group members do not leak data or keys to non-authorized nodes.

We assume there is a group manager that manages that group membership. The group manager acts as a certificate authority (CA) for the group, responsible for issuing member certificates that bind a member's public key to the group IP address and for revoking group memberships. We also assume all group members know the public key of the group manager, so that all member certificates can be verified by any group member.

We do not consider attacks against the multicast protocol itself. For example, we do not consider denial of service (DoS) attacks against data forwarding and assume both group members and non-member nodes forward application and control data according to the protocol specification. Protecting the multicast protocol is complementary to our work.

IV. Design Space

The security goal of our protocol is to ensure data confidentiality. However, this goal should not be achieved at the price of sacrificing performance and robustness. More specifically, properties we want to achieve are:

- Group secrecy: this property makes it computationally infeasible for a non-member node to discover the group data; this also includes properties like forward or backward secrecy which guarantee that it is computationally infeasible for a member node to get access to group data before joining the group, or after leaving (or being revoked from) the group, respectively.
- Efficiency: the wireless environment requires that the protocol be efficient in terms of both communication cost and computation cost.

• Robustness: the protocol should be resilient to unreliable links and possible link and node failures.
• Performance: the secure protocol should maintain similar data throughput to the upper layer application as the unsecured protocol.

Efficient confidentiality and integrity of data delivery for group communication can be achieved by using symmetric-key based cryptographic algorithms. We consider two main approaches: one relies on using a common key to encrypt and decrypt the data, while the other uses per-hop keys to achieve the same goals.

Common-key based approach. In this approach, the critical component is the protocol that defines how the common data encrypting key (also referred as *group key*) is computed and disseminated. Such protocols are also referred to as *group key management protocols*. Although the group key management protocols are already extensively studied for the wired networks, the unique characteristics of wireless communication introduces new challenges that require new solutions tailored for the wireless environment. For example, many previously proposed protocols were designed under the assumption that there exist mechanisms for reliable key delivery. However, in the wireless environment, links are inherently much less reliable. In addition, the multi-hop nature of wireless communication exacerbates the problem of unreliable links, since missing one key packet at one node affects all downstream nodes that rely on this node. Therefore, achieving efficient reliable key delivery is a critical component for group key management protocols in wireless networks. Compared to the wired networks, the key delivery structure is also less straightforward in the wireless environment. On one hand, the existing group data delivery structure may not be optimized for delivering keys, since keys have much more stringent reliability requirement than data. On the other hand, building a customized delivery structure for keys requires additional protocols for handling of possible link and node failures. Careful selection of the key delivery structure is necessary for wireless networks.

Per-hop key based approach. In the per-hop key based approach, the group data is encrypted hop-by-hop by relying on the secure channels established between group members. One of the main challenges for such protocols in the wireless environment is that group members do not directly communicate with each other. Therefore, non-member nodes are required in the establishment of the secure channels, which introduces additional security concerns. Secondly, the straightforward way of using hop-by-hop encryption disallows the use of broadcast for data dissemination, instead hop-by-hop unicast must be used. Additional mechanisms are required for the per-hop key approach to take advantage of the broadcast nature of wireless communication for data dissemination.

Given the above described design space and challenges, in the rest of the paper, we first present a protocol that adopts the common key based approach and several other protocols that adopt the per-hop key based approach. We will discuss in detail how these protocols address the challenges in the wireless networks, and describe their advantages and limitations.

V. A Centralized Key Management Protocol

In this section, we present W-LKH, a secure group communication for WMNs that uses the common key based approach. We first provide an overview of W-LKH, then describe its reliable key delivery mechanisms, and finally discuss its limitations.

A. Overview of W-LKH

W-LKH is based on a well-known centralized scheme, LKH. We selected LKH because it was intensively studied and it was shown to work well in wired networks. We chose its batching variant [14], which we refer to as B-LKH, given the benefits of batching in reducing the computation and communication overhead.

In W-LKH, data is encrypted using a group key and delivered on the multicast tree. In order to ensure forward and backward secrecy of the group data, at every join and leave event, the source is notified and a new key is generated and distributed to the current group members. As in B-LKH, the source maintains a logical key tree to ensure a logarithmic bound for the size of the message rekey. The main difference between W-LKH and B-LKH is the message delivery process.

Delivery of join and leave messages: In order to maintain the consistency of the logical key tree maintained at the source, the join and leave requests have to be delivered via reliable channels. The TCP protocol, which is normally used in wired networks, does not work well for the delivery of the join and leave requests on WMNs, as building a TCP session requires several round trip time and the delivery of several control packets, and consequently results in large latency and bandwidth overhead. Instead, we use a simple reliable transport protocol which involves only an ACK from the receiver to ensure reliable delivery. Therefore, for most cases, only one round trip time and one additional control message are required to complete a join or leave request.

B. Rekey Message Transportation

The responsibility of the rekey message transportation process is to deliver the rekey messages generated by the data source reliably to each group member. The approach we use for the rekey message transportation is to enhance the existing MAODV tree built for the data delivery with hop-by-hop reliability for delivering the rekey messages, such that each node retransmits the rekey message until all of its downstream members receives the message.

1) Hop-by-Hop Reliable Key Delivery: The most common approach to the hop-by-hop reliable delivery is the ACK mechanism, where the receiver sends an ACK to the sender after receiving a message, as in the 802.11 unicast protocol. However, since in the multicast environment, there are usually multiple downstream receivers for each rekey message, the ACK mechanism can cause the well-known ACK implosion problem. Instead, we choose to use the NACK mechanism, where a node sends a NACK to the sender when it detects packet misses. The missing of rekey messages can be detected when a node receives a data packet encrypted with an unknown key. Since receiving data packets is a frequent event, the detection of missing keys happens quickly. Compared to the ACK mechanism, the NACK mechanism also has the benefit of smaller overhead, as it is expected that the probability of a node receiving a rekey message is greater than the probability of missing the message.

To further reduce the protocol overhead, we exploit the broadcast nature of wireless signal by applying the NACK suppression technique [25]. With the NACK suppression technique, when a node detects that it misses a rekey message, instead of firing the NACK immediately, it sets a NACK timer with a random timeout up to some maximum value. If it receives a NACK from another node requesting the same rekey message before its NACK timer expires, it resets its NACK timeout value. The NACK timer is cancelled once the node receives the missing rekey message it requested. Since most downstream nodes are close to each other, for most cases only one NACK message is necessary even though multiple downstream nodes miss the same rekey message. Furthermore, if the NACK timer is set small enough, the missing rekey message can be recovered before the next data packet is broadcasted by the parent. This allows for time sensitive applications to resume the decryption of data as soon as possible while keeping the overhead low.

2) Rekey Message Recovery: Even with hop-by-hop reliability, a number of rekey messages can still be lost for a large duration of time for a particular node due to link or node failures and network partitions. In such cases, the key recovery procedure is invoked to recover the missing keys. Instead of requesting the missing packets directly from the data source, as in the wired network, we adopt a local recovery procedure in order to minimize the bandwidth overhead while not affecting the application performance.

The local recovery procedure is only invoked at a node when the node can receive a continuous stream of data packets from its upstream node, as the continuous stream of data packets indicates the path between the node and the data source is functional. To initiate the local recovery process, the node transmits a NACK packet containing all the sequence numbers of the missing rekey messages to its tree parent. Once the tree parent receives the NACK packet, it sends to the requesting node the requested rekey messages for which it has already received. For the other rekey messages, a local recovery procedure is recursively invoked on the tree parent. The process repeats until all the requested rekey messages are delivered to the original requesting node.

Note that in the above described local recovery process, it is necessary for each node to buffer the rekey messages that it receives for some period of time so that the request for missing rekey messages from downstream nodes can be satisfied as locally as possible. Since rekey messages are of small size and are issued infrequently with the batch rekeying technique, it is feasible for each nodes to buffer recent rekey messages for the purpose of rekey message recovery.

3) Data and Key Message Ordering: Since in the above rekey transportation and recovery scheme the missing of rekey messages is detected by receiving data packets that are encrypted with unknown keys, we require that each node forwards only data packets for which it has the decryption key and buffers undecryptable data packets until the corresponding decryption key has been received and forwarded to the downstream node. This requirement minimizes the out-of-order problem of key and data message, thus reducing the number of NACKs for missing key messages. Under most cases, it also ensures that when a node receives a NACK for a rekey message, it has already received the requested rekey message, thus recursive propagation of NACK message is eliminated. Note that delaying forwarding undecryptable packets does not affect the data throughput for the application, as a packet undecryptable in a node is necessarily undecryptable in its downstream nodes.

C. Limitations of W-LKH

Although W-LKH has been optimized for the WMNs, it still has several limitations. First, the join and leave requests require message exchanges between the joining or leaving node to the data source. Depending on the distance from the joining or leaving node to the data source, this operation can incur significant latency and bandwidth overhead. Second, the rekey message which includes key encryptions required by all the group members needs to be transmitted throughout the multicast tree, even though typically only a subset of the encryptions are required by a particular branch of the tree. Finally, the use of batching for reducing the bandwidth overhead also causes partial loss of the forward and backward data secrecy. These limitations are the consequence of the fundamental design choice made by the LKH scheme, centralized group membership management, where the data source is the central point that handles all group join and leave events.

VI. Secure Overlay Based Secure Multicast in WMNs

In this section, we present a new secure multicast protocol framework, WSOM, that uses the decentralized membership management principle to address the limitations in W-LKH. We first provide an overview of the framework, then present three different protocols and a member revocation mechanism.

A. Overview of WSOM

The WSOM framework is based on an overlay tree maintained on top of the data delivery multicast tree. The overlay consists of only member nodes and two member nodes are connected on the overlay if they are adjacent in the underlying multicast tree disregarding non-member nodes. Figure 1 shows an example of the overlay structure for a sample multicast scenario. Neighboring nodes on the overlay maintain a symmetric key, referred to as *link key*, between them, which establishes a secure channel between these two nodes. We refer to this overlay network as a *secure overlay*. Since we only consider tree based multicast structure, the overlay structure we just described is necessarily a tree. For convenience, we use the term overlay parent, overlay children and overlay neighbor to refer to the parent, children, and neighbor of a node in the overlay, respectively.

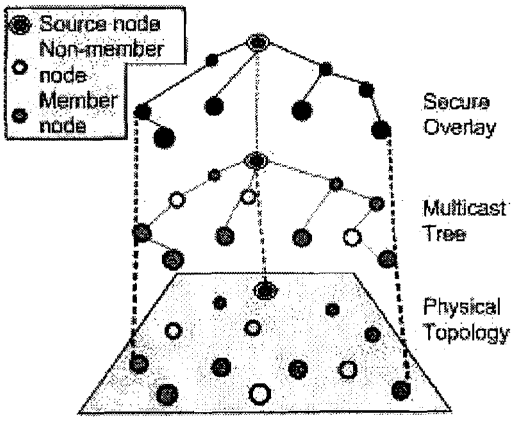


Fig. 1. Example secure overlay

Maintenance of the Secure Overlay: The key for maintaining the secure overlay is for each node to maintain an updated link key with its overlay neighbors as the underlying multicast tree changes, which can be caused by group join, group leave, and link and node failures. We now present the operations required for handling each such event in detail.

For group joins, after being part of the multicast tree, the joining node communicates with its overlay parent along the multicast tree path to establish a link key using the standard public key infrastructure (PKI) techniques. If the joining node is already a tree node before joining, it also needs to build a link key with each of its overlay children. To accomplish this, the joining node broadcasts a parent change packet including its member certificate downward along the tree. Each of its overlay children, upon receiving the parent change packet, generates a random link key and sends it to the joining node after proper signing and encrypting using the standard PKI techniques. For graceful group leaves, the overlay parent of the leaving node is notified of the event and establishes link keys with the overlay children of the leaving node in a way that is similar to when the joining node builds link keys with its overlay children. For ungraceful leaves and link and node failures, the link keys are re-established once the downstream nodes get reconnected back to the tree much like the joining case.

Note that in handling all of the above events, only local message exchanges are required. Moreover, in the WMN environment, where all nodes are static, most changes on the underlying multicast tree are due to group join and leave events. Therefore, for stable groups and network environment, the overhead for maintaining the secure overlay is very small.

B. WSOM Protocols

In this section, we present three different secure multicast protocols that use the secure overlay structure as described in the previous section: WSOM-GK, WSOM-LK, and WSOM-HK.

1) WSOM-GK: WSOM with Group Key for Data Encryption: In this protocol, a group key is maintained among all group members. The group data is encrypted with the group key at the source, then disseminated on the multicast tree. The source periodically refreshes the group key by generating a new group key. The new group key is disseminated to all group members using the secure overlay.

For group joins, besides updating the secure overlay, the overlay parent of the joining node piggy-backs the current group key on the messages required for updating the secure overlay, so that the joining node can start decrypting group data immediately. For group leaves, only the update of the secure overlay is required. Key loss due to node or link failures can be handled in a way similar to the local key recovery strategy in W-LKH.

The main limitation of this protocol is that it suffers from partial loss of the forward and backward data secrecy. However, the application can adjust the key refreshment period to balance the bandwidth overhead and the loss of the forward and backward secrecy.

2) WSOM-LK: WSOM with Link Key for Data Encryption: In this protocol, the group data is delivered directly on the secure overlay. To forward a data packet, the node encrypts the data packet with the link key of each of its overlay children and then forwards the encrypted packet to the corresponding children. This basic scheme suffers from two drawbacks. First, re-encrypting the data packet for each of the overlay children requires computation cost linear to the number of overlay children for each node, which can be significant for nodes with many children. Second, it is impossible to exploit the broadcast nature of wireless transmission, as each of the encrypted data packet is only useful for one downstream child. To overcome these drawbacks, instead of using link keys to encrypt the data packets directly, the source encrypts the data packet with a randomly generated data encryption key (k_d). To disseminate the data packet, the source encrypts k_d with the link key of each of its overlay children, piggy-backs all the encryptions of k_d to the data packet, and then broadcasts the packet on the multicast tree. When a member node receives an encrypted packet, it can decrypt the packet by first decrypting k_d with its corresponding link key and then using k_d to decrypt the data packet. For forwarding the received packet to its downstream nodes, the member node re-encrypts k_d with the link keys of its overlay children, and replaces the k_d encryptions on the received packet with the new set of encryptions of k_d. Although the number of encryptions required for each node is also linear to the number of overlay children of the node, this scheme is still computation-wise efficient, as the size of k_d is typically only 128 bits.

Since no additional control data is maintained in WSOM-LK, the handling of join and leave events only requires updating the secure overlay. Unlike WSOM-GK, this protocol does not suffer from key loss problem.

3) WSOM-HK: WSOM with Hop Key for Data Encryption: In WSOM-LK, even with the optimization of using data encryption keys, there is still per data packet computation and bandwidth overhead on each member node for encrypting and delivering the key for each of its overlay children. The aim of WSOM-HK is to reduce both the computation and bandwidth overhead by maintaining a hop key among every member node and its overlay children. With the help of the hop key, the data encryption key only needs to be encrypted once with the hop key and only one encryption of the key needs to be appended to the data packet for forwarding to the downstream nodes, instead of one for each overlay child as in the WSOM-LK protocol.

Each hop key can be regarded as a mini group key with the member node as the source and its overlay children as the group members. Due to the small scale, a straightforward approach, such as encrypting and delivering the new hop key to each of the overlay children whenever the overlay children set changes, can be employed to maintain the hop key.

The cost of maintaining a hop key is amortized over all the data packets delivered using that key. Unlike WSOM-LK scheme, this scheme has lower per packet overhead.

C. Revocation in the WSOM Based Protocols

Unlike in the centralized membership management schemes where member revocations can be easily performed by the central point, in decentralized membership management schemes, a separate membership revocation mechanism has to be provided. Instead of using the straightforward certificate revocation list (CRL) approach, which requires the reliable delivery of the CRLs to all group members, we design a new more efficient revocation mechanisms for the WSOM based protocols. The main observation we exploit is that under the static topology of WMNs, it is possible to restrict a node to join the secure overlay only through a few nearby member nodes, which we will refer to as the *join points* of the node. Then to revoke a member node, it is sufficient to delivery the revocation notice to only the small number of join points of the node, instead of to the whole group, thus saving the network bandwidth. In the following, we describe the details of the revocation protocol together with the required changes on the WSOM protocol. For convenience, we refer to our revocation protocol as *WSOM-revoke* and the entity responsible for issuing and revoking the member certificates as the group manager.

1) Overview of WSOM-revoke: With WSOM-revoke, prior to obtaining the member certificate, the node attempting to join the group selects a set of its nearby member nodes as its join points. Then during the process obtaining the member certificate, the node provides the pre-selected join points to the group manager, which then saves the join points and also includes them in the member certificate for the node. To join the secure overlay, the node only activates the multicast tree branch that leads to one of its pre-selected join points, which we will refer to its *actual join point*. The actual join point verifies that itself is in the set of pre-selected join points of the joining node by checking the node's member certificate and that the joining node is not revoked before admitting the node as its overlay child. Now, to revoke a member node, the group manager only needs to delivery the revocation notice to the pre-selected join points of the node[1]. Once all the join points of the node receive the revocation notice, the node can no longer join the secure overlay, thus loses its ability to decrypt the group data.

2) Details of WSOM-revoke: Now we discuss some more subtle details of the WSOM-revoke protocol.

Pre-selecting the join points To obtain a suitable set of join points, the joining node broadcasts in the local scope a member request message. The member nodes that receive the member request message reply a member reply message including its identity and its distance to the data source. The joining node then selects the best join points among all the member nodes who replied by considering the distance of the replying member node to itself and to the data source.

The size of the join point set In order to prevent arbitrary large join point set, which can potentially be used to mount DoS attack during the revocation process and to delay the revocation of the node, the group manager can impose an upper bound on the number of join points each node can use. Due to the static nature of the network topology, an upper bound as few as three can be sufficient.

Handling group leaves Since group leave is a common event, it is possible for a node that its actual join point decides to leave the group, or all the pre-selected join points of the node leave the group. In both cases, it is desirable that the ability of the node to join the secure overlay is not affected. To achieve this, we introduce a join point delegation mechanism. With the delegation mechanism, when a node decides to leave the group, it delegates the join point responsibility for its overlay children to its overlay parent by sending a signed delegation message to its overlay parent. Similarly, when a node that has left the group receives join request, it delegates the join point responsibility to its overlay parent with a signed delegation message. Therefore, in both cases, the joining node can continue to join the secure overlay via the join point that has left; its ability to join the secure overlay is oblivious of the leave status of its selected join points.

Updating join points It is possible that all of the pre-selected join points of a member node are revoked or fail. In such cases, it is necessary for the member node to obtain new join points in order to continue to participate in the secure overlay. A member node may also desire to change its join point set if it finds a better set of join points. In both cases, a join point update procedure is called for. With WSOM-revoke, updating join points is achieved by obtaining a new member certificate with the new join point set from the group manager. Since it is expected that the member revocation and failure events are infrequent and the static network environment limits the opportunity of finding better join points, we expect the join

[1]Delivering the revocation notice to the join points of the node is sufficient for denying the access to the group data for the node. The revocation notice may also need to be delivered to the member nodes which have the revoked node as one of its join points, so that those nodes will not select the revoked node as their overlay parent. However, under the assumption of no DoS attack, the revoked node cannot pretend to be member node to prevent member nodes' access to data.

point update procedure is only invoked infrequently, hence the centralized design for handling the procedure is acceptable.

VII. Analytical Comparison

In this section, we analyze and compare the overhead of the proposed protocols. We focus on the communication cost since bandwidth is the main limitation.

In order to have a clear comparison between the protocols, we make the following assumptions. We assume there is no interference, thus the bandwidth cost for sending a message depends only on the path length (in hop count) and the size of the message. We use b to denote the bandwidth cost of transmitting one byte to the group via the multicast. Thus, the bandwidth cost of multicasting a packet of size D to the group is bD. We assume that the latency of a message depends only on the number of hops travelled by the message and both join and leave require only one round trip of message exchange. Table I shows all the parameters we use in the comparison.

Table II shows the results of different metrics for different operations in the proposed protocols. Based on these comparison results, we now highlight a few differences between the protocols.

- For join and leave operations, there is potentially a large bandwidth and latency cost for W-LKH (depending on the distance between the data source and the joining or leaving node), whereas, WSOM based schemes only incur constant costs.
- W-LKH and WSOM-GK, which use the common group key to encrypt group data, require rekey operations, whereas, no rekey operations are necessary for WSOM-LK and WSOM-HK. The rekey operations consume network bandwidth resource, while batching introduces a vulnerability window.
- WSOM based protocols require explicit revocation messages, which is not necessary for W-LKH. In applications with only infrequent revocations, the bandwidth cost for revocation is insignificant. For applications that require frequent revocations, we can batch process the revocations in the same way as batching the rekey operations for the group key and use the life time of membership certificates to reduce the revocation bandwidth overhead to an acceptable range.
- For common group key based protocols (W-LKH and WSOM-GK), there is no per data packet overhead, whereas, WSOM-LK and WSOM-HK incur per data packet overhead.

VIII. Experimental Evaluations

In this section, we present the results of our experiments with the *ns* [13] network simulator for evaluating the proposed protocols. We first demonstrate the importance of reliable key delivery for group key management protocols in WMNs, then we evaluate and compare the performance and overhead of the proposed protocols.

A. Simulation Setup

We implemented our experiments based on the *ns* network simulator [13] (version 2.26) with CMU Monarch extensions. The MAODV implementation we used is provided by Zhu et al [26].

Nodes are configured to use the IEEE 802.11 radios with 2Mbps physical bandwidth and 250-meter nominal range. In each simulation, 100 nodes are randomly placed within a 1500 meters by 1500 meters area and the multicast data source is placed at the center of the area at the coordinates (750, 750).

The duration of a simulation is 900 seconds. In the beginning of each simulation, a set of nodes are randomly selected to be the initial group members and join the group sequentially at the rate of one join per three seconds. For experiments with no group dynamics, the initial group size is the fixed group size for the experiment. For experiments with group dynamics, the initial group size is the stable group size for the experiment. After the initial joins are completed, the source starts to multicast data packets of size 256 bytes to the group at a rate specific to each experiment until the end of the simulation. For the experiments that examine the effect of group dynamics, the data rate is fixed at 5 packets/second.

Based on previously observed group dynamics for multicast applications [27], [28], [29], we use Poisson process to model the member join and leave events with different rates to reflect different levels of group dynamics. We set the join and leave rates to be equal, so the group size remains stable. For each join event, a random non-member node is selected to join the group; similarly for each leave event, a random member node is selected to leave the group.

For protocols that require periodic rekeying (which includes B-LKH, W-LKH and WSOM-GK), the rekey period is set to be 30 seconds. The maximum NACK timeout value used for the reliable key delivery is set to be 100ms. We also assume in all the protocols the size of symmetric keys is 128 bits, the size of public/private keys is 1024 bits, and the computation delay for PKI signatures is 4ms.[2]

We experimented with different group sizes, however, since the comparison results of different protocols are similar for different group sizes, we only present the results for the group size of 50. In all the figures, each data point is the average of 10 different runs with different random topologies and different random group join and leave events.

B. Metrics

We measure the performance of the secure multicast protocols with two metrics, the *delivery ratio* and the *decryption ratio*. For each member node, the delivery ratio is defined as the fraction of data packets that are received by the node out of all the data packets that are broadcasted by the data source during the time when the node is a group member. The decryption ratio for a member node is defined as the fraction of data packets that can be decrypted by the member out of all the data packets received by the member. Thus, the delivery ratio measures the impact of the secure multicast protocol on the data delivery ability of the underlying multicast

[2]This value is based on the 1024 bits RSA implementation of `openssl` on 3GHz Intel Pentium IV computer.

number of members	n
multicast tree height	h
average tree degree	d
data packet size	D
symmetric key length	k
CRL length	r
Total message size exchanged for join or leave	s
distance between the joining or leaving node and the source	δ
bandwidth cost for group multicast	b

TABLE I

PARAMETERS FOR PROTOCOL OVERHEAD COMPARISONS

		W-LKH	WSOM-GK	WSOM-LK	WSOM-HK
Join/Leave	bandwidth	δs	s	s	s+dk
	latency	2δ	2	2	3[a]
Rekey	bandwidth	$bk \log n$	bdk	–	–
	latency	h	h	–	–
Revoke	bandwidth	–	br	bdr	$br + bk$
	latency	–	h	h	h
Data	bandwidth	0	0	bdk	bk
	latency	h	h	h	h
vulnerability window?		Yes	Yes	No	No

TABLE II

OVERHEAD COMPARISON RESULTS

[a]One additional hop time for the new link key

protocol, whereas, the decryption ratio measures the impact of the secure multicast protocol on the actual data goodput received by the upper layer application. In order to get a lower bound on the delivery ratio and the decryption ratio, we assume the upper layer application requires time sensitive delivery, that is, a member node cannot buffer undecryptable packets for decryption and forwarding upon the receiving of proper keys, instead such packets are dropped by the member.

The overhead of the secure multicast protocols are measured in terms of the bandwidth overhead of the protocol and the latency for group join and leave events. Due to the scarcity of bandwidth resources on WMNs, it is essential to compare the bandwidth overhead incurred by different protocols. The latency of join and leave events reflects the responsiveness of the protocol to the upper layer applications.

C. Reliable Key Transport

We now demonstrate the importance of reliable key transport for secure multicast protocols on WMNs, which motivates our design for W-LKH. Figure 2(a) and 2(b) show the delivery and decryption ratio of LKH and B-LKH for different levels of group dynamics. As we can see from these figures, while both LKH and B-LKH maintain a similar delivery ratio as the case without any security mechanism, these two protocols have very poor decryption ratios. The poor decryption ratio for LKH is due to two reasons: the high probability of key loss on the wireless network and the frequent rekeying operations which exacerbate the key loss problem. B-LKH improves the decryption ratio over LKH by reducing the frequency of the rekey operations, however, since the key loss problem is not solved, the end result is still not satisfactory. If we can further solve the key loss problem, as we will see in the performance results of W-LKH below, the decryption ratio turns out to be dramatically improved. Therefore, based on these observations, we conclude that reliable key delivery is essential for secure multicast protocols on WMNs.

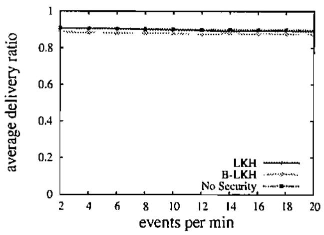
D. Protocol Performance and Robustness

Figure 2(c), 2(d) and Figure 2(e), 2(f) show the delivery and decryption ratio for all the proposed protocols for different levels of data rate and group dynamics, respectively. We observe that for all the data rates and group dynamics examined, all the proposed secure multicast protocols can maintain a similar high delivery ratio as in the case where no security mechanisms are being used. The decryption ratios for all the protocols are also almost 1. Therefore, we conclude that all the proposed protocols can provide good transparency in terms of data throughput to the upper layer applications. We also experimented with random node and link failures to examine the robustness of the protocols in the case of failures. The resulting performance is similar to the performance results shown for the case with no artificial failures. We omit these graphs for the lack of space.

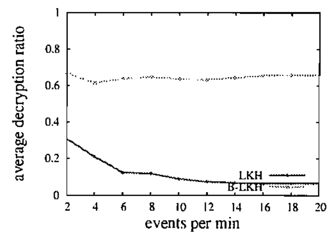
E. Protocol Overhead

1) Computation overhead: Figure 3(a) and3(b) show the computation overhead due to symmetric encryptions and asymmetric encryptions at the source node and a randomly selected member node for different protocols for experiments with the data rate of 5 packets/second (10kbps) and the group dynamics of 5 joins and 5 leaves per minute. For the symmetric encryption overhead, we observe that WSOM-LK has much higher overhead than the other protocols, especially at the source node. This is because WSOM-LK requires per data packet computation overhead that is linear to the number of children of the node. For the asymmetric encryption overhead, we observe that W-LKH has a significantly higher number of asymmetric encryptions performed at the source node than the other protocols. The reason is that with W-LKH the source node handles all the join and leave requests, each of which requires asymmetric encryption operations, whereas for the WSOM based protocols, the join and leave requests are handled in a distributed fashion, hence the required asymmetric encryptions are shared by all member nodes. Since asymmetric encryptions are computationally intensive operations, the high number of asymmetric encryptions at the source node in W-LKH can potentially introduce a performance bottleneck at the source, especially at high group dynamics. It also allows for potential DoS attacks that aim at exhausting the computation resource at the source node.

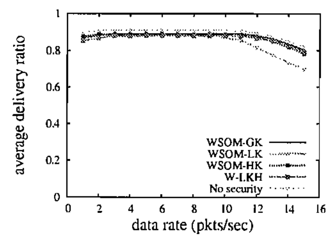
2) The bandwidth overhead and latency for join and leave operations: Figure 3(c), 3(d) and Figure 3(e), 3(f) show the bandwidth overhead and latency for the join and leave events, respectively, for different levels of group dynamics. From these graphs, we can make the following observations. First, for all proposed protocols both the bandwidth overhead and



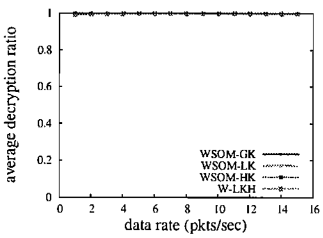
(a) Delivery ratio of LKH and B-LKH



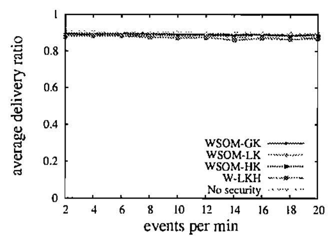
(b) Decryption ratio of LKH and B-LKH



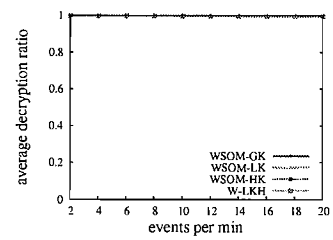
(c) Delivery ratio v.s. data rate



(d) Decryption ratio v.s. data rate

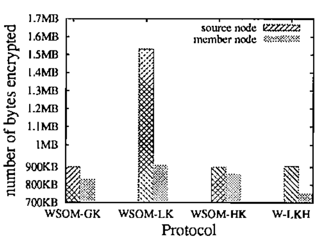


(e) Delivery ratio v.s. group dynamics

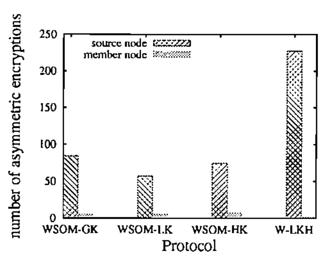


(f) Decryption ratio v.s. group dynamics

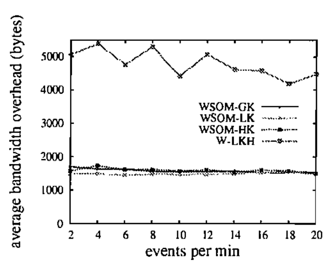
Fig. 2. The delivery and decryption ratio comparison



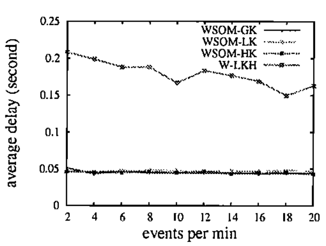
(a) The symmetric encryption overhead



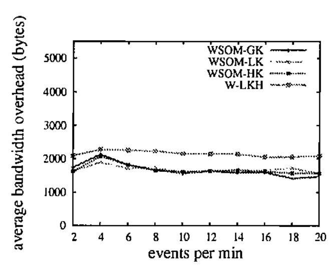
(b) The asymmetric encryption overhead



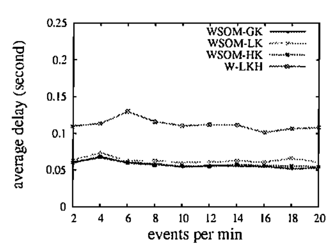
(c) Join bandwidth overhead



(d) Join latency



(e) Leave bandwidth overhead

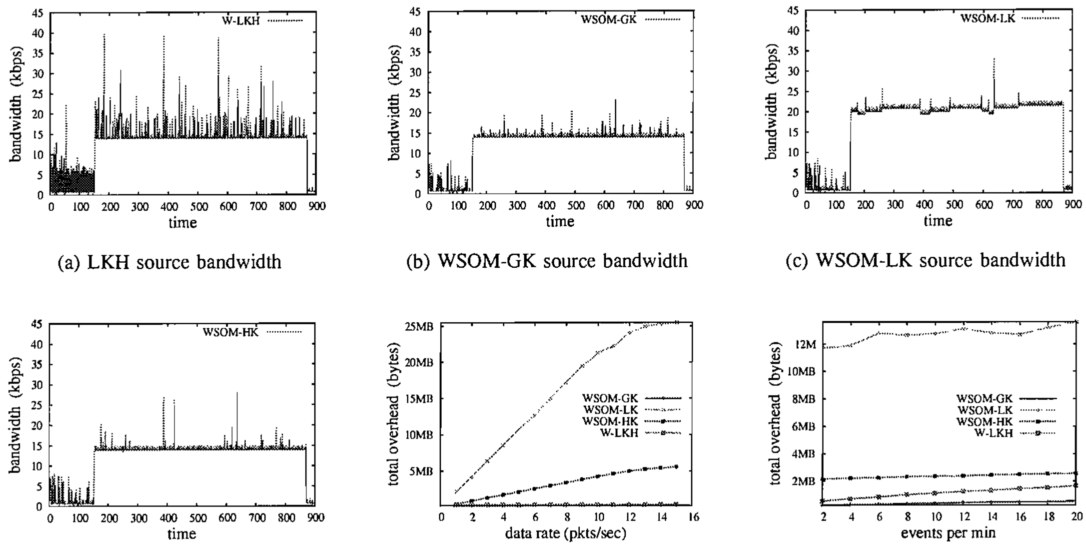


(f) Leave latency

Fig. 3. The computation overhead and join/leave overhead comparison

latency remain stable for different levels of group dynamics. Second, the WSOM based schemes have much less bandwidth overhead and latency than the W-LKH protocol for both join and leave events. This is the manifestation of the difference between the centralized and decentralized membership management principles. With decentralized membership management, as in the case of WSOM, only local messages are required for joins and leaves. On the other hand, centralized membership management schemes, as W-LKH, require global messages between the joining or leaving node to the data source.

3) Peak bandwidth: Figure 4(a), 4(b), 4(c), and 4(d) show the bandwidth consumed at the source node over time for all the different protocols for a simulation run with the data rate of 5 packets/second (10kbps) and the group dynamics of 5 join and leave events per minute. From these graphs, we can see that WSOM based protocols consume relatively stable bandwidth at the source over time, while W-LKH exhibits high variability of bandwidth consumption. The reason for



(a) LKH source bandwidth
(b) WSOM-GK source bandwidth
(c) WSOM-LK source bandwidth
(d) WSOM-HK source bandwidth
(e) Total bandwidth overhead v.s. data rates
(f) Total bandwidth overhead v.s. group dynamics

Fig. 4. The peak and total bandwidth overhead comparison

the high peak bandwidth requirement of W-LKH is two-folds. First, the size of the rekey packets in W-LKH is relatively large, since potentially many keys on the key tree needs to be updated for a rekey event. Second, all the join and leave requests require communication with the source in W-LKH. Since high bandwidth peaks can cause packet loss and possible congestions on the network, W-LKH is less favorable than the WSOM based protocols in this respect.

4) Total bandwidth overhead: In order to get an overview of all the bandwidth overhead introduced by the secure multicast protocol, Figure 4(e) and 4(f) show the average total bandwidth overhead due to the secure multicast protocol for an entire simulation session for different data rates and group dynamics, respectively.

We first observe that the bandwidth overhead for both WSOM-LK and WSOM-HK increase linearly with the data rate. However, the increase rate for WSOM-HK is significantly smaller than WSOM-LK, which makes the bandwidth overhead of WSOM-HK comparable to other protocols while the bandwidth overhead of WSOM-LK are significantly higher. This difference shows the effectiveness of the hop key in WSOM-HK for reducing the bandwidth overhead. From Figure 4(f), we can also observe that for all the protocols, the total bandwidth overhead remains quite stable for different levels of group dynamics.

F. Applicability of the Protocols

Based on the above experiment results, we now suggest the best protocol under different application requirements. For applications with low data rate, WSOM-HK is the best choice, since WSOM-HK has small join and leave latency and overall bandwidth overhead and it does not have a vulnerability window for forward and backward data secrecy. For applications with higher data rate, if the application can tolerate partial loss for forward and backward data secrecy, then WSOM-GK is the best choice. Otherwise, neither W-LKH and WSOM-GK can be used; the best choice is still WSOM-HK.

IX. Conclusion

In this paper, we explored different design choices for solving the problem of secure multicast service for WMNs. We proposed several secure multicast protocols, and compared them both analytically and experimentally. We discussed the trade-offs among different design choices and suggested the best design choices for different application scenarios. Future work includes extending the proposed protocols to multi-source group communications, and experimenting with the protocols in a wireless mesh testbed.

References

[1] C. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs," *IEEE/ACM Trans. Netw.*, vol. 8, no. 1, 2000.

[2] A. Perrig, D. Song, and J. D. Tygar, "Elk, a new protocol for efficient large-group key distribution," in *Proc. of S&P*, 2001.

[3] S. Mittra, "Iolus: a framework for scalable secure multicasting," in *Proc. of ACM SIGCOMM '97*, 1997.

[4] X. Zhang, S. Lam, and H. Liu, "Efficient Group Rekeying Using Application Layer Multicast," in *Proc. of ICDCS '05*, 2005.

[5] C. Abad and I. Gupta, ""Adding confidentiality to application-level multicast by leveraging the muticast overlay"," in *Proc. of ADSN '05*, 2005.

[6] R. Torres, X. Sun, A. Walters, C. Nita-Rotaru, and S. Rao, "Enabling confidentiality of data delivery in an overlay broadcasting system," in *to appear in INFOCOM 2007*, 2007.

[7] W. Du, J. Deng, Y. Han, and P. Varshney, "A key predistribution scheme for sensor networks using deployment knowledge," *IEEE TDSC*, vol. 3, no. 1, 2006.

[8] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proc. of S&P*, 2003.

[9] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. of CCS '02*, 2002.

[10] S. Zhu, S. Setia, S. Xu, and S. Jajodia, "Gkmpan: An efficient group rekeying scheme for secure multicast in ad-hoc networks," *Mobiquitous*, vol. 00, 2004.

[11] R. Balachandran, B. Ramamurthy, X. Zou, and N. Vinodchandran, "Crtdh: an efficient key agreement scheme for secure group communications in wireless ad hoc networks," in *Proc. of IEEE ICC '05*, 2005.

[12] T. Kaya, G. Lin, G. Noubir, and A. Yilmaz, "Secure multicast groups on ad hoc networks," in *Proc. of SASN '03*, 2003.

[13] "The network simulator - ns2." http://www.isi.edu/nsnam/ns/.

[14] X. Li, Y. Yang, M. Gouda, and S. Lam, "Batch rekeying for secure group communications," in *Proc. of WWW '01*, 2001.

[15] C. Wong and S. Lam, "Keystone: A group key management service," in *Proc. of ICT '00*, 2000.

[16] X. Zhang, S. Lam, D.-Y. Lee, and Y. Yang, "Protocol design for scalable and reliable group rekeying," *IEEE/ACM Trans. Netw.*, vol. 11, no. 6, 2003.

[17] X. Zhang, S. Lam, and D.-Y. Lee, "Group rekeying with limited unicast recovery," *Comput. Networks*, vol. 44, no. 6, 2004.

[18] Y. Yang, X. Li, X. Zhang, and S. Lam, "Reliable group rekeying: a performance analysis," in *Proc. of SIGCOMM '01*, 2001.

[19] W.-P. Yiu and S.-H. Chan, "Sot: secure overlay tree for application layer multicast," in *Proc. of ICC '04*, 2004.

[20] C. Abad, I. Gupta, and W. Yurcik, "Adding confidentiality to application-level multicast by leveraging the multicast overlay," in *Proc. of ADSN in ICDCSW '05*, 2005.

[21] S. Zhu, C. Yao, D. Liu, S. Setia, and S. Jajodia, "Efficient security mechanisms for overlay multicast-based content distribution.," in *ACNS*, vol. 3531 of *Lecture Notes in CS*, pp. 40–55, 2005.

[22] S. Roy, V. Addada, S. Setia, and S. Jajodia, "Securing maodv: Attacks & countermeasures," in *Proc. of SECON '05*, 2005.

[23] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The tesla broadcast authentication protocol," *RSA CryptoBytes*, 2002.

[24] E. Royer and C. Perkins, "Multicast operation of the ad-hoc on-demand distance vector routing protocol," in *Mobile Computing and Networking*, 1999.

[25] S. Pingali, D. Towsley, and J. Kurose, "A comparison of sender-initiated and receiver-initiated reliable multicast protocols," in *Proc. of SIGMETRICS '94*, 1994.

[26] Y. Zhu and T. Kunz, "MAODV implementation for NS-2.26," Technical Report SCE-04-01, Carleton University.

[27] K. Sripanidkulchai, A. Ganjam, B. Maggs, and H. Zhang, "The feasibility of supporting large-scale live streaming applications with dynamic application end-points," in *SIGCOMM*, 2004.

[28] Y.-H. Chu, A. Ganjam, E. Ng, S. Rao, K. Sripanidkulchai, J. Zhan, and H. Zhang, "Early experience with an internet broadcast system based on overlay multicast," in *USENIX Annual Technical Conference*, 2004.

[29] K. Almeroth and M. Ammar, "Collecting and modeling the join/leave behavior of multicast group members in the mbone," in *Proc. of (HPDC) '96*, 1996.