Efficient and Secure Multicast in WirelessMAN: A Cross-layer Design

Chin-Tser Huang, Manton Matthews, Matthew Ginley, Xinliang Zheng, Chuming Chen, and J. Morris Chang

Abstract: **Effectively adding security measures to a multicast service is an intriguing problem, especially when the service is deployed in a wireless setting. Next generation IEEE 802.16 standard WirelessMAN networks are a perfect example of this problem, and the latest draft specification of the standard includes a secure protocol solution called Multicast and Broadcast Rekeying Algorithm (MBRA). In this paper, we expose the security problems of MBRA, including non-scalability and omission of backward and forward secrecy, and propose new approaches, ELAPSE and ELAPSE+, to address these problems. In particular, ELAPSE+ makes use of membership and mobility information gathered in the application layer to augment the adaptive group management in the MAC layer. We analyze the security property of ELAPSE and ELAPSE+, and compare their performances with MBRA by simulating group rekeying scenarios.**

Index terms: **802.16 WirelessMAN, Multicast, Privacy and Key Management (PKM) Protocol, Multicast and Broadcast Rekeying Algorithm (MBRA), Efficient sub-Linear rekeying Algorithm with Perfect Secrecy (ELAPSE), ELAPSE+**

I. Introduction

There are many emerging applications that depend on secure group communications, which require the privacy of participants and access control at the multicast server. On the other hand, scalability is another critical concern for the multicast service underlying these applications due to the possible large number of group members. In the domain of wired networks, efficient and secure multicast is a widely studied problem and several popular protocols have been proposed. This is not necessarily true for the domain of wireless networks, where attention has been less significant.

Wireless networks have become more and more pervasive due to their many advantages. The IEEE 802.16 standard [1] aims to provide broadband wireless access for Metropolitan Area Networks (MAN) and the recently released IEEE 802.16e [2] adds mobility features and some other functions including multicast. Multicast in Wireless Metropolitan Area Networks (WirelessMAN) is a promising service, suitable for many applications, such as stock option bidding, pay per view TV broadcasting, video conferencing, etc., for both fixed and mobile subscriber stations (SS).

The challenge of a secure multicast service, such as the one in IEEE 802.16, is to provide an efficient method for controlling access to the group and its communications. Encryption of group messages and selective distribution of the keys used for encryption is the primary method for ensuring the security. For a dynamic group in which membership changes frequently, the rekeying algorithm employed by the service is a critical ingredient of the overall service efficiency. This algorithm should guarantee forward secrecy, which prevents a leaving member from accessing future communications; and backward secrecy, which prevents a joining member from accessing former communications. On the other hand, a rekeying algorithm should be efficient as well. That means it should be scalable to a large group and exhibit good performance during key distribution, in which the performance is measured in terms of communication complexity, center (server) space complexity, and user (member) space complexity.

This paper reviews the Privacy and Key Management (PKM) protocol (with respect to the multicast setting) and the Multicast and Broadcast Rekeying Algorithm (MBRA) in IEEE 802.16e. The weaknesses of these protocols are detailed. The Efficient sub-Linear rekeying Algorithm with Perfect Secrecy (ELAPSE) protocol, a derivative of the Logical Key Hierarchy, and the ELAPSE+ protocol, an improved version of ELAPSE, are proposed. An overview of ELAPSE can be found in our preliminary work [3]. ELAPSE overcomes the lack of backward and forward secrecy of the 802.16 MBRA and operates more efficiently overall. The extended version, ELAPSE+, makes use of membership and mobility information gathered in the application layer to augment the adaptive group management in the MAC layer.

The rest of the paper is organized as follows. In Section II, we review the IEEE 802.16e solution to secure multicast rekeying, with its weaknesses emphasized. In Section III, related works on other approaches to secure multicast are described. A complete description of our approaches ELAPSE and ELAPSE+ are presented in Section IV and V,

Manuscript received May 3, 2007 and revised September 15, 2007. This research was supported in part by National Science Foundation (Award CNS-0551650), U.S.A., in 2007.

Chin-Tser Huang, Manton Matthews, Matthew Ginley, and Chuming Chen are with University of South Carolina, U.S.A. (e-mail: huangct@cse.sc.edu, matthews@cse.sc.edu, mginley@gmail.com, chen7@cse.sc.edu)

Xinliang Zheng is with Frostburg State University, U.S.A. (email: xzheng@frostburg.edu)

J. Morris Chang is with Iowa State University, U.S.A. (e-mail: morris@iastate.edu)

respectively. In Section VI, the performance of MBRA, ELAPSE, and ELAPSE+ are analyzed based on our simulation of different group rekeying scenarios. Our conclusions are made in Section VII.

II. Current 802.16e Standard

The Multicast and Broadcast Service in IEEE 802.16 is an efficient and power saving mechanism, which also provides subscribers with strong protection from theft of service by encrypting broadcast connections between an SS and BS. The Multicast and Broadcast Rekeying Algorithm (MBRA) is used to refresh traffic keying material for the multicast service of IEEE 802.16. Prior to receiving multicast service, an SS must register and authenticate with a base station (BS), during which the BS decides the level of service to be authorized. By use of the ranging procedure on the Initial Ranging or Basic Connection, an SS establishes a Primary Management Connection with a BS that is used to exchange MAC management messages. If the SS is to be managed, a Secondary Management Connection is established between the SS and BS. A Secondary Management Connection is used to transfer delay-tolerant, standards-based messages within IP datagrams such as DHCP, TFTP, and SNMP.

The Privacy Key Management messages are exchanged through the Primary Management Connection, with the exception that PKMv2 Group-Key-Update-Command is transferred over the Broadcast Connection. The Privacy and Key Management (PKM) protocol is applied in the IEEE 802.16 security sublayer within the 802.16 MAC layer and performs two functions. First, the PKM protocol provides secure distribution of keying material from a BS to SS, and second, the protocol enables a BS to enforce access control over network services. A brief summary of a PKM protocol run between an SS and BS is as follows. The SS initiates the protocol and first authenticates with a BS (PKMv2 also provides mutual authentication), establishing a shared secret — an Authentication Key (AK). The BS will also send a Secure Association Identifier (SAID) list, which indicates the services explicitly authorized to the SS. Then by a Key-REQ message from SS to BS and Key-RSP message from BS to SS, the SS receives the keying material that is appropriate for a specified SAID. Before proceeding with the details of the current standard, let us briefly discuss a trivial solution for securing multicast traffic.

A. A Trivial Solution

In this trivial solution, multicast traffic is sent from the BS to all SS encrypted using a single group wide session key, or Group Traffic Encryption Key (GTEK). It is assumed that all SS have the current key ready to decrypt the multicast data. When a new SS wishes to join the group, an individual request is sent to the BS for the GTEK. The BS responds to the new SS with a new GTEK, and then also sends the updated GTEK to all existing SS individually (all individual exchanges are encrypted with keys established through a previous authentication mechanism). When a member wishes to leave the group, the BS must again send a new GTEK to all other SS individually. Although offering strong backward and forward secrecy, this trivial solution has many problems, most importantly not being scalable due to the many unicast key exchanges.

B. 802.16 Standard

The IEEE 802.16 standard offers some improvement to this trivial solution. A lifetime is specified for the GTEK and thus the GTEK will expire after a certain amount of time. To ensure timely delivery of new GTEKs before expiration of the current one, the use of a Group Key Encryption Key (GKEK) is specified. The GKEK has a lifetime that parallels the lifetime of the corresponding GTEK. By using this GKEK to encrypt the GTEK, new GTEKs can be broadcast to all SS.

An SS may get the initial GTEK, which is used to encrypt the multicast traffic, by Key Request and Key Reply messages over the Primary Management Connection. A BS updates and distributes the traffic keying material periodically by sending two Group Key Update Command messages: for the GKEK update mode and for the GTEK update mode. The Group Key Encryption Key (GKEK) is used to encrypt the GTEK in GTEK update mode. Intermittently, a BS transmits the (1) Key Update Command message for GKEK update mode to each SS through its Primary Management Connection. This message contains the new GKEK encrypted with the Key Encryption Key (KEK), which is derived from the Authorization Key (AK) established during authentication. Then, the BS transmits the (2) Key Update Command message for GTEK update mode through the Broadcast Connection, which contains the new GTEK encrypted with the corresponding GKEK. The protocol can be specified as follows.

$$BS \rightarrow SS : \{GKEK\}_{KEK} \quad (1)$$

$$BS \Rightarrow \text{all } SS : \{GTEK\}_{GKEK} \quad (2)$$

where → stands for a unicast message and ⇒ stands for a broadcast message.

There are still two problems with this protocol. Firstly, this protocol is not scalable as it still needs to unicast to each SS. It can be generalized, especially in a potentially large network such as a WirelessMAN, that any rekeying scheme depending on unicast methods is not scalable. Secondly, this protocol does not address the issue of backward and forward secrecy. In the case of member joining, when a new member receives the current GTEK, it can decrypt all previous messages that were multicast during the lifetime of the same GTEK. In the case of member leaving, there is nothing in this protocol that prevents a leaving SS from receiving the next GKEK and decrypting the next GTEK.

Note that the lifetimes of GTEKs as specified by the IEEE 802.16 standard are an important security consideration. Currently, the range is specified to be 0.5 hours minimum, 12 hours by default, and 7 days maximum [3]. This lifetime has great leverage on the relationship between scalability and forward/backward secrecy provided by the standard. A long enough lifetime needs to be maintained to allow a BS enough time to individually update the GKEK so that the new GTEK can be broadcast. However, longer GTEK lifetimes imply much greater lapses in backward/forward secrecy on member join/leave events, respectively, as there will be more messages encrypted using the given GTEK.

III. Related Works

Since the first version of the IEEE 802.16 standard [4] was released in 2002, a few articles and books have been published. In [5], the chair of the standard gives a technical overview of IEEE 802.16. Some 802.16 group members also published a book [6] in 2006, which provides a detailed overview of the standard and explains the rationale behind development decisions. The authors of [7] review the standard, analyze the security provided by the standard, and discuss the requirement of mutual authentication between SS and BS. In [8] the PKM protocol is discussed in detail, more attacks on the versions of the PKM protocols listed in [4] and [6] are discovered, and revisions of PKM protocols are proposed. In [9], another attack on PKM version 2 in [2] is detailed. However, none of these publications cover the MBRA version released in earlier 2006 [2].

There is a report [10] which analyzes the IEEE 802.16 MBRA, which especially focuses on replay attacks against the MBRA, similar to the attacks listed in [7], [8] and [9]. However, it does not cover the backward and forward secrecy afforded to communications before/after rekeying, or the efficiency of the MBRA, both of which are paramount to a desirable, secure rekeying algorithm.

More generally, secure multicast has been a popular topic in the past ten years, and many protocols have been proposed. [11] and [12] are the first few works dealing with secure multicast, in which straightforward, yet not scalable methods, are described. The Iolus approach detailed in [13] is a distributed method in which a hierarchy of agents are used as subgroup controllers. Using Iolus, scalability is ensured because member changes in one subgroup do not affect other subgroups. It also provides other promising features such as fault-tolerance. However, Iolus may not be directly applicable to the 802.16 environment in which there is only one server (BS) and a number of clients (SS), and may not make the best use of the property of 802.16 that every SS within the radio range of BS can receive the multicast messages in one hop. Kronos [18], takes a unique periodical rekeying approach that rekeys the group only at specified time intervals. Customary rekeying upon member changes are delayed until the next rekeying interval, therefore the number of rekeying is reduced.

Logical Key Hierarchy (LKH) tree algorithms are proposed in [14] and [15], which provide O(log n) communication complexity, where n is the number of group members. There are three schemes in the Versa-key framework [16], one of which is a centralized tree-based management scheme. It applies a one-way function to update a key tree upon members joining, and thus is also referred to as LKH+. In [17] a hybrid system is proposed that integrates LKH with a simple flat scheme, providing a family of key management algorithms according to the degree of tree internal nodes and the number of members in each subgroup. Each subgroup is then organized as a leaf in the LKH tree. By dividing the group into subgroups with O(log n) members, the algorithm exhibits only O(n/log n) server space complexity. The authors of [17] claim it is the first rekeying algorithm to require only sub-linear space at the server. Besides, as a variation to all the above schemes, Rodeh et al [18] proposed a completely distributed group key management scheme which lets the members "agree" on the associated subgroup keys and eventually the overall group key in a distributed manner, thus waiving the need of a centralized server. However, this scheme is not suitable for 802.16 WirelessMAN due to the asymmetry in communication; that is, unlike the BS, an SS may not have the transmission capability to reach other SS in the group.

In this paper, ELAPSE, an alternative to the IEEE802.16 MBRA is proposed. ELAPSE is a more efficient alternative that provides complete backward and forward secrecy to communications, and integrates the advantages of the approaches presented in [17] and [19] to achieve better efficiency.

IV. ELAPSE

We have established that MBRA published in the latest 802.16 standard is insufficient. As we have shown, the MBRA offers only modest improvements over a trivial solution. A proper solution should maintain backward secrecy and forward secrecy. From these goals, an improved MBRA must re-key on member joins, on member leaves, and periodically if there is no member join or member leave. Also, an improved MBRA must be scalable so that its complexity is less than O(n) with respect to the size of the group.

The focus of the approach presented here will be sub-grouping SS so that the GKEK will not be maintained via unicasting to individual SS, but via broadcasting to sub-groups. For every cell of a BS and many SS in a multicast application, the SS will be sub-grouped into $N = 2^k$ sub-groups, with each sub-group maintaining k keys. The exact value of N is to be determined by the implementer to offer the best performance for a given application. For example, an application that averages 600 SS may pick a value of N = 8 sub-groups, with each sub-group averaging 75 members and maintaining k = 3 keys. When a new SS requests keying material, it will be grouped into the sub-group with the lowest member count. This is done to keep the sub-groups balanced in size. Otherwise, one sub-group may become very large with respect to the others and the efficiency of re-keying drops significantly. Meanwhile, this sub-grouping scheme has the advantage of avoiding the tree rebalancing cost as indicated in [20].

Each sub-group maintains a hierarchy of sub-group KEKs (SGKEK) instead of a single GKEK. According to a binary tree hierarchy, each SS within a sub-group will store k SGKEKs. Fig.1 shows the case for N = 4. In the figure, note that sub-group 1 stores $SGKEK_1$, $SGKEK_{12}$, and $SGKEK_{1234}$, and that $SGKEK_{1234}$ will function as the traditional GKEK did. Also, all future examples will be made with reference to Fig. 1.

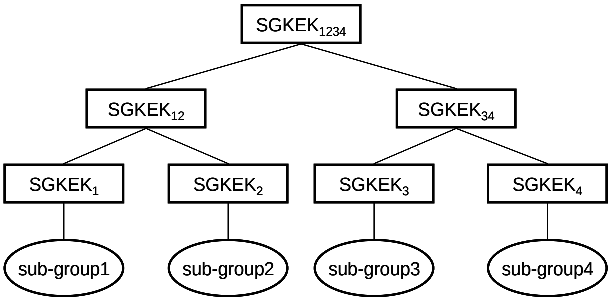


Fig. 1 Sample key hierarchy with 4 sub-groups

In the simplest case of re-keying, there are no member joins or leaves. For reference, every GTEK lifetime shall define a multicast session. In this case the GTEK, or session, expires due to time with no membership changes. The lifetime of the GTEK remains the same as it is in the 802.16 standard. In this case only one message needs to be sent.

$$BS \Rightarrow \text{all } SS : \{GTEK\}_{SGKEK1234} \quad (3)$$

The next case shall be re-keying due to a member join. The member join starts off as it does in the original specification with a key request sent from SS to BS, and a key reply sent from BS to SS. However, the key reply is modified to include a new hierarchy of SGKEKs. So for example when a new SS joins and sub-group 2 is currently the sub-group with the lowest number of members, the key reply is like message (4), with all keys being not current, but updated versions.

$$BS \rightarrow \text{new } SS : \{SGKEK_{1234}, SGKEK_{12}, SGKEK_{2}\}_{KEK} \quad (4)$$

Message (4) is delivered to all existing SS inside sub-group 2 via unicast as well. While (4) is being delivered, the BS re-keys all existing SS with new versions of appropriate keys in parallel. Continuing with the same situation of a SS joining sub-group 2, (5) and (6) would be delivered to re-key all SS not in sub-group 2.

$$BS \Rightarrow SS_{SG3}, SS_{SG4} : \{SGKEK_{1234}\}_{SGKEK34} \quad (5)$$
$$BS \Rightarrow SS_{SG1} : \{SGKEK_{1234}, SGKEK_{12}\}_{SGKEK1} \quad (6)$$

where SS_{SGi} means the collection of SS within sub-group i.

The updated GTEK is not included in these messages for a performance reason. If during the updates, more SS attempt to join, the situation has not changed. We will refer to this situation as a “multi-join”. To maintain efficiency, in a multi-join event the BS waits until all joining SS arrive and then places them into the same sub-group, which was the sub-group with lowest number of members at the start of the event, regardless if adding all the joining SS results in the sub-group not being the smallest anymore. The only addition in the case of a multi-join instead of a single join would be another message (4) to each additional SS joining the service. At the conclusion of all SGKEK updates during a join or multi-join, the new GTEK is broadcast to all SS with message (7).

$$BS \Rightarrow \text{all } SS : \{GTEK\}_{SGKEK1234} \quad (7)$$

On a member leaving the multicast service, re-keying proceeds almost exactly as a complete re-keying does for a join situation. If a member from group 2 were to leave, (4b) would be unicast to all remaining SS in sub-group 2. Next, (5b) and (6b) would be broadcast to the respective members not in sub-group 2. The difference between join and leaves is that with a leave there is no benefit of delaying the new GTEK broadcast until the end of the entire re-keying process. Once a SS receives updated SGKEK material, it will definitely be able to decrypt the next GTEK. Therefore, if an SS that has already received new SGKEK material in the middle of another leave process decides to leave as well, no re-keying can be combined and another re-keying process must commence. In this event messages (4b), (5b), and (6b) are sent by the BS; they are identical to their counterparts except for the inclusion of the newest GTEK.

$$BS \rightarrow SS: \{SGKEK_{1234}, SGKEK_{12}, SGKEK_{2}, GTEK\}_{KEK} \quad (4b)$$
$$BS \Rightarrow SS_{SG3}, SS_{SG4}: \{SGKEK_{1234}, GTEK\}_{SGKEK34} \quad (5b)$$
$$BS \Rightarrow SS_{SG1}: \{SGKEK_{1234}, SGKEK_{12}, GTEK\}_{SGKEK1} \quad (6b)$$

V. ELAPSE+

The mobility of nodes is an essential feature of 802.16e. However, the joining/leaving event due to the movement of one mobile SS node will force the entire multicast group to renew the group key in order to maintain forward secrecy and backward secrecy. It can be perceived that if the multicast group consists of a large number of fast moving nodes, then the group rekeying will be performed frequently. If we can make use of membership and mobility information, which can be gathered in the application layer, then the performance of ELAPSE protocol can be improved by reducing the amount of group rekeying messages. We name this improved version as ELAPSE+. Assuming we have additional mobility information from the application layer for each SS node available when it joins the BS’s multicast group, we can pass this information to the MAC layer. A relatively more mobile SS node means that it will leave the multicast group sooner than a predefined duration. According to the IEEE 802.16e standard [2], the support for node mobility is up to 150km/h, and the access range of a BS can be up to 15km for mobile nodes. Therefore, 12 minutes seems to be an appropriate threshold for defining fast moving nodes as it is the time needed for the fastest node to cross a BS cell via the diameter. However, the average mobility pattern and membership duration of nodes can be quite different in various circumstances. For example, the average mobility pattern observed by a BS deployed around a highway must differ sharply from the average mobility pattern observed by a BS deployed in a rural area. In fact, the access range of a BS is also influenced by the average mobility pattern it supports. For another example, in the pay per view application most membership duration may be aligned with the length of the program, but in an online gaming scenario there is no such boundary. In order to reduce the amount of group rekeying messages, an implementer may want to find out the approximate distribution of node speed and membership duration in the application scenario and set up the duration threshold accordingly.

With this additional cross-layer information available to the BS, the modification is very simple to carry out based on the ELAPSE protocol. First, we differentiate the sub-groups in ELAPSE. Some specific sub-groups are designated for fast moving SS nodes and by selecting an appropriate duration we make the size of those specific sub-groups smaller than other sub-groups. Then, every time a node tagged by the application layer as fast moving requests to join the multicast group, it will be put in one of the designated sub-groups instead of the sub-group with the lowest member count. For the same example used in the introduction of ELAPSE with 600 SS nodes and 8 sub-groups, if we know there are 60 fast moving SS nodes, then those 60 SS nodes can be put into 2 sub-groups with averagely 30 SS nodes each. The rest of relatively slow moving SS nodes can be put into 6 sub-groups with averagely 90 SS nodes each. This way, when member join/leave occurs, the number of message (4) in ELAPSE (the unicast message) will be decreased for those fast moving sub-groups. Even though the average number of message (4) will be increased for those slow moving sub-groups, the net tradeoff should be fewer number of messages (4) in total, since fast moving sub-groups will constitute most of the joins and leaves. In Section VI our simulation results confirm that with one simple modification of dividing groups into two types, we are able to improve the performance by 4~7% when compared to ELAPSE. We believe it is reasonable to assert that if we can optimize sub-group differentiation, which means more levels of sub-groups according to node mobility, we can improve the performance further.

VI. Evaluation

In the previous sections, we have overviewed the MBRA in 802.16e and its problems in security and efficiency, and introduced our ELAPSE and ELAPSE+ schemes that aim to address these problems. Next, we use theoretical analysis and empirical simulations to evaluate the security properties and efficiency performances of ELAPSE and ELAPSE+.

A. Security Analysis

As mentioned previously, we show that ELAPSE and ELAPSE+ provide backward and forward secrecy because at a join or multi-join event, the group key is updated, and at a leave event, the group key and all the sub-group key known to the leaving node are updated. Moreover, it is easy to verify that ELAPSE and ELAPSE+ is secure against collusion attacks by nodes that have left the group, because they cannot learn anything about the generation and distribution of the new key(s), as analyzed in [21]. Therefore, we can conclude that ELAPSE and ELAPSE+ possess the required security properties and are suitable for providing secure multicast services.

B. Efficiency Analysis

To evaluate the efficiency of ELAPSE, its communication and space complexity will be compared to other mulitcast approaches. In the simple flat scheme, such as the MBRA in 802.16, the server (group manager) should send rekeying messages to each group member respectively, with the new group key (GTEK in 802.16) encrypted with its secrete key (AK) shared with server (BS). Thus the communication complexity is O(n), server space complexity is O(1) (disregarding the individual AKs, which are created during authentication), and member space complexity is O(1). In the LKH scheme the communication complexity is O(log n) for the rekeying procedure; the server space complexity is O(n) and member space complexity is O(log n).

For ELAPSE, the exact complexity is determined by the number of subgroups, and the ranges of these complexities illustrate the tradeoffs associated with the choice of the number of subgroups. When the number of subgroups increases (from 1 to N, $N = 2^k$, $1 \leq N \leq n$), it can be generalized that the communication complexity decreases from O(n) to log N + m, where m is the number of current members in the subgroup to be updated. It is easy to see that when N increases, m will decrease, and when N = n, it becomes equivalent to the LKH scheme. As for the space complexity, the server space complexity increases from O(1) to 2N – 1 (the group key plus all the intermediate key in the hierarchy), and the member space complexity also increases from O(1) to log N + 1 (every key on the path from the subgroup to the root in the key hierarchy). The tradeoff between the communication complexity and the space complexity is up to the implementor. The authors in [17] find a (perhaps) optimal balance among these tradeoffs by dividing the group into subgroups with O(log n) members each. With this many sub-groups the communication complexity is still O(log n), the same degree as in LKH scheme, while the server space complexity is down to O(n/log n), and the member space is O(log n). However, due to the dynamic nature of membership and different node mobility levels it is difficult to always maintain equal-sized subgroups, let alone handling the case of multi-join.

The exact efficiency of ELAPSE+ is difficult to determine since it depends on the size of different types of sub-groups according to node mobilities. However, it is straightforward to see that the performance of ELAPSE+ should be close to ELAPSE in big O notation, while ELAPSE+ decreases the total number of unicasts to certain degree based on different mobility nature of sub-groups.

C. Simulation Results

To compare the performance, we simulate MBRA, ELAPSE, and ELASPE+ by randomly generating join and leave events. Two big sets of simulations have been performed. One allows up to 512 SS nodes and the other allows 1024 SS nodes. Within each set of simulations, tests have been done with different number of sub-groups, 8, 16, and 32, respectively. All tests are done using the same sequence of random events which include about 6000 joins and leaves, respectively. For the case of ELAPSE+, we differentiate about one thirds of the SS nodes to be fast moving and the rest to be slow moving, and different sub-groups are set up accordingly. For all the simulation runs we use the BS as point of reference for collecting statistics. The total number of messages and total amount of communications sent from the BS are used to evaluate the efficiency. The simulation results are shown in Fig. 2 through Fig. 5.

In Figs. 2 and 3, messages were tallied as unicast or multicast. Broadcast messages such as broadcast GTEK update mode messages were counted as multicast. Counting messages with the 802.16 was straightforward, as key response messages sent on join and leave, and GKEK update mode messages were counted as unicast. The broadcast GTEK update mode message was counted as multicast. For ELAPSE and ELAPSE+, all key response messages within the sub-group of the joining/leaving node were counted as unicast. The other messages, SGKEK and GTEK updates, were counted as multicast.

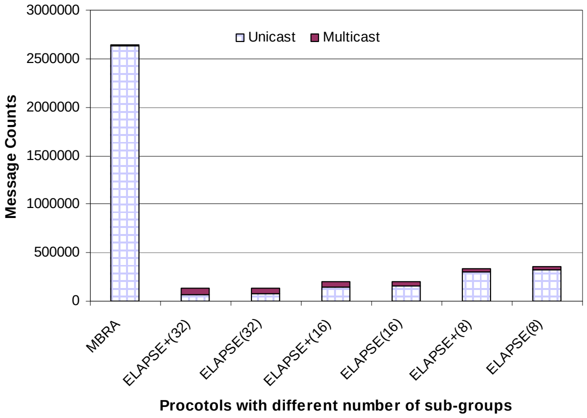


Fig. 2 Message counts for different protocols (512 nodes)

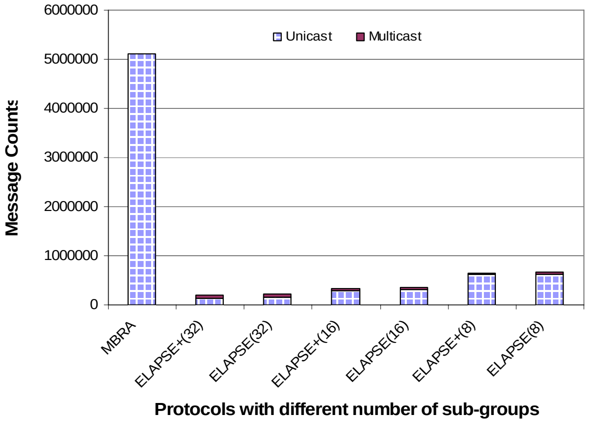


Fig. 3 Message counts for different protocols (1024 nodes)

In Figs. 4 and 5, the total units of communications are calculated as the sum of the number of keys in each message. For example in ELAPSE each unicast message (4) will add log(N) + 1 units to the total communication counts, where N is the total number of sub-groups.

A point about the implementation of the 802.16 MBRA must be made with respect to SS join and leave events. In the current standard, there is no explicit behavior defined, so we will assume the BS rekeys the entire group at every join and leave event. If it is to be assumed that no rekeying is performed on member joins and leaves and only on GTEK expiration, the number of messages sent would be drastically lower (equal to the number of join events that occurred during simulation). However, there would be lapses in secrecy on every join (and leave) equivalent to the amount of data sent before (and after). For these reasons, rekeying on SS joining and leaving was included with the 802.16 MBRA simulations so that all algorithms could be compared strictly in terms of efficiency, with the requirement that each algorithm ensures perfect backward and forward secrecy.

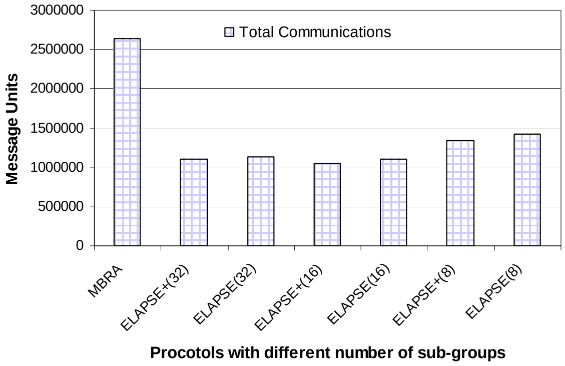


Fig. 4 Total communications by message units (512 nodes)

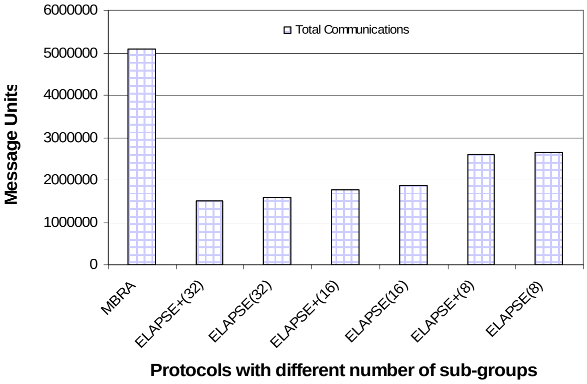


Fig. 5 Total communications by message units (1024 nodes)

From Figs. 2 and 3 it is obvious that to reach the same level of security ELAPSE and ELAPSE+ need much less number of messages to be transmitted by the BS, and Figs. 4 and 5 tell us that even considering message size, which is measured by the number of keys in each message, the total amount of communications needed to be done by the BS is less than half of the MBRA case. For both message counts and total communications our ELAPSE+ is about 4~7% less than our ELAPSE in each simulation scenario. Message counts (Figs. 2 and 3) will increase for both ELAPSE and ELAPSE+ when number of sub-groups increases, i.e. average sub-group size decreases, while total communications (Figs. 4 and 5) will be optimized when number of sub-groups is at certain level (when the number of sub-groups is 16 for 512 nodes case and 32 for 1024 nodes case). Both results agree with our theoretical analysis above.

VII. Conclusions

In this paper we have reviewed the challenges of secure multicast, and analyzed the MBRA of IEEE 802.16, as it is a noteworthy example of these challenges emerging in next generation networks. In terms of security, MBRA is an incomplete solution by not guaranteeing secrecy of messages before and after member joins and leaves, respectively. As for distributing keying material, it is inefficient, and does not take advantage of the recent research demonstrating the effectiveness of hierarchical approaches. The approach presented in this paper, ELAPSE and ELAPSE+, provide backward and forward secrecy and outperform the 802.16 MBRA in simulation. This does come at a cost of increased server and member space requirement, but this tradeoff is a matter of heightened requirements on the hardware that is to actually implement the 802.16 standard. Given the rapidly decreasing cost of client side hardware and the substantial requirements already in place on the server hardware, we believe the increased space requirement is reasonable and acceptable.

In the future work, we will continue to optimize ELAPSE+ sub-group differentiation technique, with the attempt to minimize message counts and total communications, by refining the additional information that could be passed from upper layers to the MAC layer.

References

[1] IEEE Std 802.16-2004: Air Interface for Fixed Broadband Wireless Access Systems, 2004.

[2] IEEE Std 802.16e: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, 2005.

[3] M. Ginley, S. Xu, C.-T. Huang, M. M. Matthews, “Efficient and Secure Multicast in WirelessMAN”, Proceedings of International Symposium on Wireless Pervasive Computing 2007 (ISWPC 2007), February 2007.

[4] IEEE Std 802.16-2001: Air Interface for Fixed Broadband Wireless Access Systems, 2002.

[5] Roger Marks: A technical Overview of the WirelessMAN Air Interface for Broadband Wireless Access, IEEE C802.16-02/05, 2002.

[6] C. Eklund, R. B. Marks, S. Ponnuswamy, K. L. Stanwood, N. J. M. V. Waes, “WirelessMAN: inside the IEEE 802.16 Standard for Wireless Metropolitan Networks”, Standards Informatin Network, IEEE Press, 2006.

[7] D. Johnston, and J. Walker, “Overview of IEEE 802.16 Security”, IEEE Security & Privacy, 2004.

[8] S. Xu, M. Matthews, and C. T. Huang, “Security Issues in Privacy and Key Management Protocols of IEEE 802.16”, Proceedings of the 44th ACM Southeast Conference (ACMSE 2006), March 2006.

[9] S. Xu, and C. T. Huang, “Attacks on PKM protocols in IEEE 802.16 and its later versions”, Proceedings of International Symposium on Wireless Communication Systems (ISWCS 2006), September 2006.

[10] J. Y. Kuo, “Analysis of 802.16e Multicast/Broadcast group privacy rekeying protocol”, CS 259 Final Project Report, Stanford University.

[11] A. Ballardie, “Scalable Multicst Key disribution”, RFC 1949, 1996.

[12] H. Harney, and C. Muckenhirn, “Group Key Management Protocol Specification”, RFC 2093, 1997.

[13] S. Mittra, “Iolus: A Framework for Scalable Secure Multicasting”, in Proc. ACM SIGCOMM’97, 1997.

[14] D. M. Wallner, E. J. Harder, and R. C. Agee, “Key Management for Multicast: Issues and Architectures”, RFC 2627, June 1999.

[15] C. K. Wong, M. Gouda, and S. S. Lam, “Secure group Communications Using Key Graphs”, IEEE/ACM Transaction on Networking, Vol. 8, No. 1, Feb 2000.

[16] M. Waldvogel, G. Caronni, D. Sun, N. Weiler, and B. Plattner, “The VersaKey framework: Versatile Group Key Management”, IEEE Journal on Selected Areas in Communications Vol. 17, No. 9, 1999.

[17] R. Canetti, T. Malkin, and K. Nissim, “Efficient communication-storage tradeoffs for multicast encryption,” in Advances in Cryptology-EUROCRYPT’99, 1999.

[18] O. Rodeh, K. Birman, D. Dolev, “Optimized Group Rekey for Group Communication Systems,” Proceedings of Network and Distributed System Security, February 2000, San Diego, California.

[19] S. Setia, S. Koussih, S. Jajodia, and E. Harder, “Kronos: A Scalable Group Re-Keying Approach for Secure Multicast”, Proc. of IEEE Symposium on Security and Privacy, 2000.

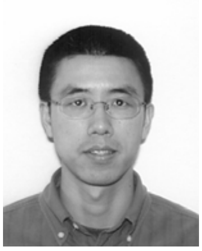
[20] H. Lu, “A Novel High-Order Tree for Secure Multicast Key Management,” IEEE Transactions on Computers, Vol. 54, No. 2, February 2005.

[21] J. Pegueroles, J. Hernandez-Serrano, F. Rico-Novella, M. Soriano, “Adapting GDOI for balanced batch-LKH,” draft-irtf-gsec-gdoi-batch-lkh-00.txt, June 2003, Work in Progress.

Chin-Tser Huang received the B.S. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1993, and the M.S. and Ph.D. degrees in computer sciences from the University of Texas at Austin in 1998 and 2003, respectively. He joined the faculty at the University of South Carolina at Columbia in 2003 and is now an Assistant Professor in the Department of Computer Science and Engineering. His research interests include network security, network protocol design and verification, and distributed systems. He is the director of the Secure Protocol Implementation and Development (SPID) Laboratory at the University of South Carolina. He is the author (along with Mohamed Gouda) of the book “Hop Integrity in the Internet,” published by Springer in 2005. He is a member of Sigma Xi, Upsilon Pi Epsilon, IEEE, and ACM.

Manton Matthews received the B.S. in mathematics in 1972, and the M.S. in computer science and Ph.D. in mathematics from the University of South Carolina in 1980. He joined the faculty at the University of South Carolina at Columbia in 1979 and is now the Associate Chair of the Department of Computer Science and Engineering. His current research interests include network security, network protocol design, natural language processing, and the logical basis for the evolution of ontologies. Dr. Matthews is a member of the IEEE Computer Society, and ACM.

Matthew Ginley was a undergraduate and then graduate student at the University of South Carolina from 2002 to 2006, graduating with his BS in Computer Science in 2006. After his undergraduate work in Computer Science, he immediately enrolled in graduate studies at the University, again in Computer Science. His graduate work was in network security, with an emphasis in secure re-keying in wireless networks. After receiving his Masters, he has found consulting work with various investment firms in New York City implementing a broad range of portfolio accounting systems.



Xinliang Zheng was a Ph.D. student at University of South Carolina at Columbia from 2002 to 2007. His graduate study is focusing on network security, especially in the area of key management for secure group communication. Since Auguest 2007 he became an assistant professor of Computer Science Department at Frostburg State University. He currently continues his research in the area of wired and wireless network security, which includes security protocol design, protocol verification, and proofs of security. He is a member of IEEE.

Chuming Chen was with Department of Biostatistics, Bioinformatics and Epidemiology at the Medical University of South Carolina from 2004 to 2006, working on NHLBI Proteomics Center funded projects in developing two-dimensional gel electrophoresis and Mass Spectrometry data management and analysis tools. He is now a PhD candidate of Department of Computer Science and Engineering at the University of South Carolina, working on logic based reasoning about ontology evolution on the Semantic Web. His research interests include network security, bioinformatics, ontology engineering and high performance computing. He is a student member of ACM.

J. Morris Chang is currently an associate professor with the Department of Electrical and Computer Engineering, Iowa State University. He received a Ph.D. degree in Computer Engineering from North Caroline State University. His industrial experience includes positions at Texas Instruments, Microelectronic Center of North Carolina and AT&T Bell Laboratories. He received the University Excellence in Teaching Award at Illinois Institute of Technology in 1999. Dr. Chang’s research interests include Managed Run-time Environment and Wireless Networks. Currently, he is on the editorial board of Journal of Microprocessors and Microsystems. He is the Middleware and Wireless Networks area editor of IEEE IT Professional Magazine. Dr. Chang is a senior member of IEEE.