Computers and Mathematics with Applications 60 (2010) 217–223

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Revisiting WiMAX MBS security

Georgios Kambourakis *, Elisavet Konstantinou, Stefanos Gritzalis

Laboratory of Information and Communication Systems Security, Department of Information and Communication Systems Engineering, University of the Aegean, 83200 Karlovassi, Samos, Greece

ARTICLE INFO

Keywords:
Asymmetric group key agreement
MBS
MBRA
WiMAX
Security

ABSTRACT

IEEE 802.16 technology also well known as WiMax is poised to deliver the next step in the wireless evolution. This is further fostered by the 802.16e specification which, amongst other things, introduces support for mobility. The Multicast/Broadcast Service (MBS) is also an integral part of 802.16e destined to deliver next generation services to subscribers. In this paper we concentrate on the Multicast and Broadcast Rekeying Algorithm (MBRA) of 802.16e. This algorithm has been recently criticized for various vulnerabilities and security inefficiencies, as its designers are trying to balance wisely between performance and security. After surveying related work, we extensively discuss MBRA security issues and propose the use of a novel asymmetric group key agreement protocol based on the work in Wu et al. (2009) [3]. Our scheme guarantees secure delivery of keys to all the members of a given group and mandates rekeying upon join and leave events. It can prevent insider attacks since only the Base Station possesses a secret encryption key while all other members in the network acquire the transmitted data by using their secret decryption keys. We compare our scheme with related work and demonstrate that although heavier in terms of computing costs, it compensates when scalability and security come to the foreground.

1. Introduction

Up until now the IEEE 802.16 technology, also well known as WiMAX, may not have the adoption rate of 802.11, but it will likely be the predominant technology for Metropolitan Area Networks (MAN) deployments for the next decade. The reason is that WiMAX can support all-IP core network architecture, low latency, advanced Quality of Service (QoS) and sophisticated security [1]. The IEEE 802.16 working group on broadband wireless access standards, actually a unit of the IEEE 802 LAN/MAN standards committee (http://wirelessman.org/), is preparing and revising formal specifications for the global deployment of broadband Wireless MANs.

A major new feature that emerges in broadband wireless standards is the support of Multicast and Broadcast Services (MBS). As with Multimedia Broadcast Multicast Service (MBMS) specified for 3GPP networks, MBS is an integral part of 802.16e also referred to as Mobile Wimax [2]. MBS is yet incipient but is anticipated to deliver true next generation services to the subscribers of such networks. Note that the first commercial network in Europe to use a mobile version of the Wimax standard was launched in Amsterdam in June 2008. Actually, MBS allows 802.16 providers to deliver advanced multicast and/or broadcast services (e.g., show video multicast service in a cell) to their subscribers. In fact, MBS is a mechanism for distribution of data content across multiple Base Stations (BSs) from a centralized media server.

MBS is by nature unidirectional which means that the BS can send messages to all Mobile Stations (MSs) registered in the same multicast group. However, for securing such services group keys are required. It is implied that before receiving any MBS, a Mobile Station (MS) must register and authenticate with a Base Station (BS) via the Privacy Key Management

* Corresponding author.
E-mail addresses: gkamb@aegean.gr (G. Kambourakis), ekonstantinou@aegean.gr (E. Konstantinou), sgritz@aegean.gr (S. Gritzalis).

0898-1221/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2009.12.032

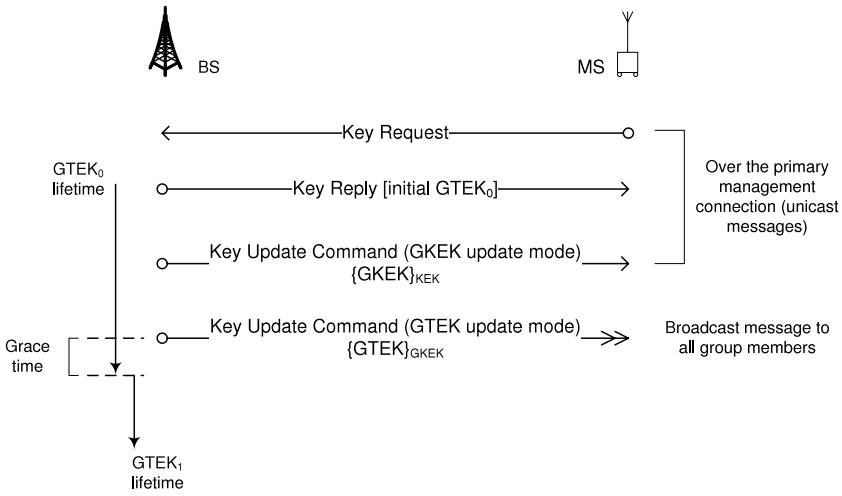

Fig. 1. MBRA message flow.

(PKM) protocol [2]. After that, the Group Traffic Encryption Key (GTEK) is used to encrypt multicast data packets and it is shared among all MSs that belong to the same multicast group. This key is randomly generated by the BS or by certain network nodes and should be refreshed frequently. Thus, it must be transmitted (after encryption) in short time intervals to all the MSs that belong to the same group employing a multicast message. Also, a key called Group Key Encryption Key (GKEK) is used for GTEK encapsulation in a PKMv2 Group Key Update Command message (GTEK update mode). GKEK is randomly generated at the BS, encrypted with the Key Encryption Key (KEK), and unicasted to each MS through the primary management connection. The KEK is derived from Authorization Key (AK) after authentication and is unique for each MS. Also, there is one GKEK per Group SA (GSA).

Our contribution: This paper specifically focuses on the Multicast and Broadcast Rekeying Algorithm (MBRA) of 802.16e. MBRA is responsible for delivering and refreshing GKEK and GTEK to each MS in a group before or during MBS acquisition. Although this algorithm has been designed with efficiency and power saving in mind, it has several security problems that have been already identified in the literature. We thoroughly discuss MBRA security inefficiencies and extensively survey existing solutions to the same problem. Focusing on the security vulnerabilities of these solutions, we propose the use of a novel public key oriented group key agreement protocol based on the very recent work in [3].

The proposed scheme achieves secure delivery of decryption key to all members of a given group and guarantees fresh key upon join and leave events. This is realized by using a single key for encryption but different key – unique for each group member – for decryption. In fact, the encryption key is only accessible to the BS; nevertheless, after the execution of the protocol, every MS in the group has a different decryption key and only legitimate nodes can decrypt the incoming encrypted messages. This warrants protection against malicious insiders who may attempt to forge the service. To the best of our knowledge, the proposed scheme is the only one which protects the network from such attacks. Moreover, the scheme provides backward/forward secrecy, efficient solutions for join/leave events while the use of asymmetric cryptography considerably increases the lifetime of the keys. Finally, we theoretically evaluate our work in terms of network and computing costs and provide comparisons to similar mechanisms proposed in the literature so far.

The rest of the paper is organized as follows: The next section gives background information, presents MBRA protocol and elaborates on the problem statement. Our proposal is presented and evaluated in Section 3. Section 4 extensively surveys previous work and offers comparisons among possible alternatives. Finally, Section 5 concludes the paper.

2. MBRA and security vulnerabilities

The MBS of 802.16e enables the distribution of data to multiple MS with one single message thus saving cost and bandwidth. In order for the BS to distribute the data in a secure way, all nodes must possess a common, secret group key. This key is called GTEK and is used by the BS to encrypt the multicast/broadcast traffic. For the creation, maintenance and renewal of GTEKs, the MBS of 802.16e uses an algorithm called MBRA (Multicast and Broadcast Rekeying Algorithm). According to this algorithm, a MS acquires the initial GTEK by employing the Key Request and Key Reply messages. These messages are transmitted over the primary management connection. After that, for refreshing a GTEK, a BS may transmit a PKMv2 Group Key Update Command message including (pushing) an encrypted – using the GKEK – fresh GTEK to all the MS group members. Actually, the 802.16e standard defines two types of the PKMv2 Group Key Update Command message: the GKEK Update Mode and GTEK Update Mode. The first one is used for refreshing the GKEK, while the second one for refreshing the GTEK for MBS. These two messages contain a counter, namely Key Push Counter, for protection against replay attacks.

More specifically, the MBRA is executed as follows (see Fig. 1): the BS through its primary management connection sporadically sends a Key Update Command for the GKEK update mode to each MS. This message contains the new GKEK

encrypted with the KEK. Every MS has a secret key KEK which is derived from the Authorization Key (AK) established during authentication [2]. Then, the BS transmits a Key Update Command for the GTEK update mode through the broadcast connection. The latter contains the new GTEK encrypted with the corresponding GKEK.

In this context, broadcast messages are encrypted symmetrically with a shared key (GTEK) known to each member of the same group. Also, every member can decrypt the traffic using the same key. Message authentication is based on the same shared key as well. However by doing so, every group member is able to encrypt and authenticate messages as if they originate from the legitimate BS. The distribution of the GTEK when the MBRA is used is another important issue. Specifically, GTEK is encrypted with the GKEK and broadcasted to all group members. Note that the GKEK is also a shared key known to every group member. Therefore, using the GKEK, a malevolent insider is able to create fake encrypted and authenticated GTEK Key Update Command messages and attempt to distribute another GTEK. If successful, group members are no longer possible to decrypt MBS traffic coming from the legitimate BS.

An insider can force MSs to accept the forged key in a number of ways as described in [4]. If the system is not properly implemented, the key contained in the last one of subsequently transmitted GTEK update command messages may replace the original one. Consequently, all the adversary has to do is send its GTEK update command message after the BS broadcasted a Key Update one. If the implementation is by the standard, the keys of both messages are accepted. So, the insider could falsify certain parts of the BS's GTEK update command message making the receiving MS to discard it. After that, the attacker can transmit its own GTEK update command message to the MSs. Even worse, considering the fact that MBS is unidirectional, the BS is not able to detect that the MS has different GTEKs.

Moreover, the MBRA has another two major shortcomings. The first one has to do with scalability: whenever the BS has to refresh GKEK, he has to unicast the new GKEK to each MS using their corresponding secret keys KEKs. Second, it is not able to cope with the issue of backward and forward secrecy. When a newcomer joins a given group and receives the current GTEK, it can decrypt all previous multicast messages transmitted during the same GTEKs lifetime. Also, a MS which leaves the group is able to receive the next GKEK and decrypt the next GTEK until the current KEK expires. Furthermore, GTEK lifetime affects scalability and forward/backward secrecy as well. The standard does not contain any directions on GTEK lifetime. Though we can assume that GTEK lifetime is the same as that of Traffic Encryption Keys (TEKs, more details can be found in [2]), given the fact that GTEK is a special kind of TEK. So, based on the specifications we can infer that the lifetime of GTEK is 12 h by default, 30 min minimum and 7 days maximum. Considering join and leave events, increased GTEK lifetime leads to much greater gaps in backward and forward secrecy, because a greater number of messages are encrypted via the given GTEK [5]. Actually, MBRA is similar to the Group Key Management Protocol (GKMP) [6], which does not provide a solution for keeping the forward secrecy except by creating an entirely new group without the leaving member. Therefore, we can easily infer that this scheme is not scalable to large dynamic groups.

3. Proposed security solution

In this section we analyze a new solution for WiMAX MBS security based on asymmetric principles. The asymmetric operation requirements put on BS and MSs may be a computational burden but our proposal is the first (to the best of our knowledge) that deals with insider attacks, requires only few broadcasts from the BS and provides backward and forward secrecy at the same time. On top of that, it is common belief that modern and especially future WiMax mobile devices will be equipped with advanced hardware components and battery reserves so public key operations will not be stressful for them.

3.1. Overview of the protocol

Very recently, the authors of [3] proposed an asymmetric group key agreement protocol. Opposed to conventional group key agreement protocols where a group of nodes creates a common secret key, the idea of the proposed protocol was the construction of *only* a shared encryption key. This encryption key can be accessible to all group members and to attackers. However, every group node (after the execution of the protocol) has a different decryption key and only legitimate nodes can decrypt the encrypted messages.

The clever idea of asymmetric group key agreement can be applied in WiMAX and lead to a very efficient solution for some of its major security problems. Building on the protocol in [3], we will present a variant of it especially suited for WiMAX MBS. In our variant, only the BS can construct the group encryption key and thus is the only member in the group which can encrypt messages. Moreover, the broadcasts in [3] will be replaced by unicast messages from the MSs' side, and the number of rounds will be now two instead of one in the original protocol. After the execution of the protocol, the BS will have a secret encryption key while all MSs will possess a different decryption key.

Our approach can entirely replace MBRA and give solutions to its security vulnerabilities. In particular, it satisfies the following properties:

1. Only the BS can encrypt data and thus insider attacks can be prevented.
2. It guarantees backward and forward secrecy.
3. After a join/leave event the keys are refreshed by two and one broadcast respectively, which means that scalability problems can be dealt efficiently.
4. Due to the use of public key cryptography, the keys can remain unchanged for a long time.

All the keys involved in MBRA (e.g. GTEKs, GKEKs, KEKs) are replaced by one group encryption key available to the BS only, while the other members of the network possess a decryption key. The broadcast encryption is based on a protocol quite similar to ElGamal encryption and thus public key cryptography replaces the symmetric algorithms used in MBRA. This clearly increases the computational burden put on every node. However, this solution deals with most of the MBRA's security vulnerabilities, provides an efficient treatment for join/leave events and the use of public key cryptography considerably increases the lifetime of the keys. This means that if there is no join or leave event, the keys can remain unchanged for a long time (even months or years).

3.2. Construction of the asymmetric keys

The proposed protocol is based on elliptic curve cryptography and bilinear maps. Recall that a bilinear map is a mapping $e : G_1 \times G_1 \to G_2$ which satisfies the following properties:

- Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in Z_p^*$.
- Non-degeneracy: If P is a generator of G_1, then $e(P, P) \neq 1$ is a generator of G_2.
- Computability: There exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

G_1 is an additive group, G_2 is a multiplicative group and both groups have prime order p. In order to use bilinear maps for cryptographic purposes we assume that the discrete logarithm problem (DLP) is hard in both G_1 and G_2. Examples of cryptographic bilinear maps are Weil pairing [7] and Tate pairing [8].

Before the execution of the key agreement protocol, all nodes should agree upon the use of the same elliptic curve parameters, the same groups G_1 and G_2, and the same base point $P \in G_1$. Every new member who wish to join the group can acquire these parameters from the BS. We will denote the BS by M_1 and the other nodes in the network by M_i for $i \geq 2$. The protocol completes in the following stages:

Bootstrapping stage: Every MS M_i chooses a random integer $h_i \in G_1$ which can be connected in some way to him. For example, the value h_i can be the result of a hash function with input the identity ID_i of each MS. These values can be alternatively generated by BS and sent to all MSs.

First stage: Every member M_i randomly generates two values $X_i \in G_1, r_i \in Z_p^*$ and computes $R_i = -r_iP, A_i = e(X_i, P)$ and $\sigma_{i,j} = X_i + r_ih_j$ for all $j \neq i$. Then, he sends $\{\sigma_{i,j}, R_i, A_i\}$ to BS. The BS M_1 keeps his values secret.

Second stage: The BS constructs his encryption key pair (R, A) where $R = R_1 + \cdots + R_n$ and $A = A_1A_2 \cdots A_n$. Clearly, the encryption key (R, A) cannot be found by any other member or attacker. Then, he computes the sums $\Sigma_i = \sum_{j=1}^{n} \sigma_{j,i}$ for all $i \geq 2$ and broadcasts them. Every MS should have the ability to recognize his corresponding value Σ_i.

Third stage: Each member M_i with $i \geq 2$ computes his decryption key $\sigma_i = X_i + r_ih_i + \Sigma_i$.

If the BS wishes to send a message $m \in G_2$ to all nodes in the network, he generates a random integer $t \in Z_p$, computes the values $c_1 = tP$, $c_2 = tR$ and $c_3 = mA^t$ and broadcasts the triple (c_1, c_2, c_3). Then, each MS M_i can find the message by the relation $m = \frac{c_3}{e(\sigma_i, c_1)e(h_i, c_2)}$. It can be easily seen that $e(\sigma_i, P)e(h_i, R) = A$ for every value i and thus $e(\sigma_i, c_1)e(h_i, c_2) = e(\sigma_i, P)^t e(h_i, R)^t = A^t$.

Since our proposed protocol is a variation of the scheme in [3], the security properties of the latter are inherited by the first. An elaborate proof was presented in [3] showing that the proposed ASGKA scheme is secure against passive attacks in the standard model under the decision Bilinear Diffie–Hellman Exponentiation (BDHE) assumption. The same proof is also valid in our case. Additionally, the secrecy of the group encryption key is guaranteed in our scheme because the BS does not reveal his values R_1 and A_1. Finally, we note that the encryption and decryption keys can remain unchanged for a long time and this is a very important advantage of our protocol against the previous proposals (see Section 4) that are solely based on symmetric keys. However, when a new node enters the network or one leaves, the keys should be changed in order to guarantee backward and forward secrecy.

3.3. Managing join/leave events

A *Join Event* occurs when a single MS wants to join the existing group. The encryption group key and the decryption keys of the other nodes should be updated to include the new member. A *Leave Event* occurs when a MS wishes to leave the group, or is forced to leave it. The keys must be properly modified so that the departing participant can no longer use the old group key in order to decrypt the group's communications.

In the case of a join event, our protocol can be modified as follows. Suppose that M_{n+1} is the new MS. First, the BS should send to M_{n+1} all values h_i for $i = 2$ to $n + 1$. At the same time BS notifies the other MSs about the join event and sends to all the value h_{n+1}. Upon the receipt of the value h_{n+1}, all nodes M_i compute the value $\sigma_{i,n+1} = X_i + r_ih_{n+1}$ and send it to the BS. The new node M_{n+1} randomly generates two values $X_{n+1} \in G_1, r_{n+1} \in Z_p^*$ and computes $R_{n+1} = -r_{n+1}P, A_{n+1} = e(X_{n+1}, P)$ and $\sigma_{n+1,j} = X_{n+1} + r_{n+1}h_j$ for all $j < n+1$. Then, it sends $\{\sigma_{n+1,j}, R_{n+1}, A_{n+1}\}$ to the BS. The BS constructs his new encryption key pair (R, A) where $R = R_1 + \cdots + R_{n+1}$ and $A = A_1A_2 \cdots A_{n+1}$. The final step is the construction of the new decryption keys: the BS computes the sums $\Sigma_i = \sum_{j=1}^{n+1} \sigma_{j,i}$ for all $i \geq 2$ and broadcasts the values $\{\Sigma_2, \ldots, \Sigma_n, \Sigma_{n+1}\}$. Finally, each member M_i computes its new decryption key $\sigma_i = X_i + r_ih_i + \Sigma_i$.

Table 1
Efficiency of the protocol.

Participants	First execution	Join	Leave
Base Station	$(n+1)(n-1)$ point additions, 1 pairing, n scalar multiplications, 2 broadcasts	$2n$ point additions, 2 broadcasts	1 point addition, 1 broadcast
Member M_i	$n+1$ scalar multiplications, 1 pairing, $n+1$ point additions, 1 unicast to BS	1 scalar multiplication, 2 point additions, 1 unicast to BS	1 point addition
Joining member M_{n+1}	–	$n+2$ scalar multiplications, 1 pairing, $n+2$ point additions, 1 unicast to BS	–

The case of a leave event is treated in a much simpler way. Suppose that the MS M_l leaves the group. Then, the new encryption key of the BS is equal to $(R^*, A^*) = (R-R_l, A/A_l)$. The decryption keys of the other nodes are equal to $\sigma_i^* = \sigma_i - \sigma_{l,i}$, where the values $\sigma_{l,i}$ can be broadcasted from the BS.

3.4. Performance evaluation

Since our proposed scheme uses asymmetric keys for encryption/ decryption operations, it is clear that the lifetime of the keys can be quite long. This means that if the join/leave operations are often, then the protocol presented in Section 3.2 can be executed only one time in the setup phase of the network and then the keys will be refreshed after join or leave events.

In the bootstrapping stage of the proposed protocol in Section 3.2, the BS broadcasts the values h_i. In the first stage, all MSs and the BS perform n scalar multiplications, one pairing operation and $n-1$ point additions.[1] Then, all MSs send their values to the BS. In the second stage, the BS performs $n-1$ point additions for the construction of R (the construction of A is negligible), $(n-1)(n-1)$ point additions for the computation of the values Σ_i and one broadcast. Finally, in the third stage all MSs compute 1 scalar multiplication and two point additions.

In the case that a new MS enters the network, the BS broadcasts the value h_i, while the joining node computes one pairing, $n+1$ scalar multiplications and n point additions and sends the computed values with a unicast to BS. All other MSs calculate one scalar multiplication, one addition and unicast their data to BS. Next, the BS computes its encryption key with n point additions, calculates n values Σ_i after n point additions (it has to add only the contributions from the joining node) and broadcasts them. In the last step of the protocol, all MSs (except the joining MS) compute their decryption keys with only one point addition because the value $X_i + r_i h_i$ has been calculated in the first execution of the protocol. The newcomer computes his decryption keys with two point additions and one scalar multiplication.

Finally, a leave event requires only one broadcast by the BS and one point addition for all group members. The efficiency of our proposed scheme is summarized in Table 1.

4. Related work and comparison

As we mentioned in Section 2, MBRA has several security vulnerabilities. One solution towards the MBRA problem is to forbid broadcasted key updates [4]. The GTEK update command message could be unicastively transmitted to every MS, in a similar way as the GKEK update command message (bear in mind that the PKMv2 Group Key Update Command message for the GKEK update mode is carried over the primary management connection). The key should then be encapsulated using the (unique) KEK of each MS. Such a solution revokes GKEK distribution and makes the protocol simpler (just one message). However, as with the original MBRA, is not scalable, does not protect from message forging originating from insiders, does not cope with join and leave events and introduces several other problems related to the standard (i.e., management of key lifetimes).

Public key cryptography is another option [4]. According to this approach the GTEK update command message can be encrypted with the GKEK and broadcasted, but is additionally signed using the private key of the BS. Any MS that receives a GTEK update command message can verify the signature using the public key of the BS and obtain the GTEK. Of course, the public key approach has the obvious disadvantage of performance; a symmetric solution can be processed very fast and protects from outsiders. Nevertheless, a basic problem still remains; every group member is able to forge messages and masquerade as the legitimate BS and all MSs share a common symmetric key (GTEK).

A third option is to generate GTEKs as part of a hash chain as described in [4]. According to the authors initially the BS produces a random number that corresponds to the initial key $GTEK_0$. All subsequent GTEKs are generated by applying a one way hash function to the previous GTEKs respectively. This is repeated n times. Using the hash chain one can validate each GTEK by applying the same one way function to the previous one. This scheme can work properly, only if the last key

[1] Notice that according to state of the art algorithms for pairing computations [8], a pairing computation is approximately equal to three scalar multiplications and one scalar multiplication requires $O(\log_2 p)$ point additions.

Table 2
Communication cost for BS.

Protocol	Normal	Join	Leave
MBRA [2]	1 Broadcast, n Unicasts	Not considered	Not considered
GKDA [9]	1 Broadcast, k Unicasts	1 Broadcast, k Unicasts	1 Broadcast, k Unicasts
Deininger et al. [4]	1 Broadcast, n Unicasts	Not considered	Not considered
ELAPSE [5]	1 Broadcast	N Broadcasts, k Unicasts	$N-1$ Broadcasts, $k-1$ Unicasts
Xu et al. [10]	1 Broadcast, n Unicasts	1 Broadcast, 1 Unicast	1 Broadcast, n Unicasts
Our proposal	2 Broadcasts	2 Broadcasts	1 Broadcast

($GTEK_n$) is securely transmitted to each MS since it is the only key in the chain which cannot be authenticated by another one. This can be achieved by using the GKEK update command message. While this scheme presents a good performance, it does not considers join and leave events and thus is not able to cope with forward and backward secrecy. Also, it cannot counterfight masquerading attacks when launched by insiders.

The authors in [9] propose another symmetric scheme which can be used for secure distribution of keys in groups. In Group-Based Key Distribution Algorithm (GKDA), the MBS group is divided into N subgroups, and instead of one common GKEK, N GKEKs are used by the subgroups. So, upon a leave event, only the GKEK used in the specific subgroup needs to be updated. As normal, the GKEK is encapsulated using the user's KEK in the subgroup, and sent to them over the primary management connection. GKDA deals with forward and backward secrecy, but increases overhead in BS due to subgroup management (i.e., subgroups should be kept balanced). GTEK update mode message is also considerably lengthier and demanding to construct as it contains N GTEK ciphertexts encapsulated by different GKEKs. Moreover, as with other proposals, every member in a given subgroup is able to spoof the service using the shared symmetric key.

Work in [5] introduces a rekeying algorithm, called Elapse, supporting perfect secrecy to address the problems of MBRA. As GKDA, Elapse is based on subgrouping the MSs so that the GKEK is not maintained by unicasting to individual MS but by broadcasting to subgroups. Elapse divides the MSs into $N = 2^k$ subgroups and each subgroup manages k keys. The N factor is configured by the administrator. In this context, instead of using a single KEK each subgroup manages a hierarchy of subgroup KEKs. Elapse copes with join and leave incidents but as with GKDA must keep several active GKEK at the same time and manage the subgroups wisely. Additionally, it needs to use several unicast and broadcast messages upon each join/leave event (depending on the number of subgroups) with considerably bigger messages in order to update the GKEK hierarchy. Insider attacks are also not considered by Elapse.

Work in [10] also provides extensive analysis on MBS in 802.16 and proposes a scheme comprising of two messages that addresses this issue. That is,

$$Msg.1(unicast)BS \rightarrow MS : (GTEK)_{KEK}$$

and

$$Msg.2(broadcast)BS \rightarrow MS : \{update\_notice\}.$$

By sending the GTEK to MS periodically, the proposed scheme also reduces the BS's workload needed for key refreshment. However, it is needed to send an update notice in plaintext, thus saving both BS and MS encryption/decryption as well as key storage. More specifically, if GTEK is first distributed by Key Update Command messages (including leave events), both Msg. 1 and Msg. 2 need to be sent, and the key index in Msg. 2 is set to 0. Upon a join event only Msg. 2 needs to be transmitted using a greater index than previous one. Each group member will update the GTEK according to the index using agreed one way hash function. In the meantime the BS unicasts the updated GTEK, encrypted by KEK, to the new member. Upon key expiration, only Msg. 2 needs to be broadcasted, carrying a greater index in order to inform group members to update the GTEK. The main drawback of this protocol is Msg. 2. This message is transmitted in cleartext and if no integrity is protected, it may be easily exploited by attackers in order to cause Denial of Service (DoS) to the entire group. That is, the attacker may fool the MS to update its GTEK (using the pre-agreed one-way hash function) by sending it a Msg. 2 with a greater index. As with all schemes discussed the current proposal is still vulnerable to an insider attack.

In Tables 2 and 3 we summarize the communication cost of all protocols from the side of BS and MSs respectively. We have assumed that the number of MSs in the network is n and in the case that subgroups are formed, N is the number of subgroups and $k = n/N$ is the number of MSs in each subgroup. By 'Normal' we mean the procedure followed when the lifetime of the keys expires and new keys should be distributed in the network. When looking at the BS side, we easily notice that all protocols except Elapse and ours require to unicast in normal mode. However, for join and leave modes our proposal outperforms all the others by using only broadcasts. On the other hand, the proposed scheme needs from every MS to unicast in normal and join modes, and of course, is by nature heavier. This however is not a big issue nowadays because modern mobile devices afford advanced hardware components able to effectively cope with public key operations. In a nutshell, our scheme is the only that does not require unicasts from BS (which is very important for scalability issues) while it puts a slight burden in MSs requiring one unicast in normal mode and in join events.

Finally, in Table 4 we summarize some of the most important characteristics of all protocols. We can infer that when efficiency is the prime focus then MBRA is perhaps the best solution. But when increased security is needed one should choose the solution that achieves the best balance between performance and security services. In this context, only

Table 3
Communication cost for each MS.

Protocol	Normal	Join	Leave
MBRA [2]	–	Not considered	Not considered
GKDA [9]	–	1 Unicast for joining MS	–
Deininger et al. [4]	–	Not considered	Not considered
ELAPSE [5]	–	1 Unicast for joining MS	–
Xu et al. [10]	–	1 Unicast for joining MS	–
Our proposal	1 Unicast	1 Unicast	–

Table 4
General characteristics of the protocols.

Protocol	Admin. cost in BS	Supports B/F secrecy	Treat insiders	Subgrouping	Lifetime of keys
MBRA [2]	Low	No	No	No	Short
GKDA [9]	High	Yes	No	Yes, subgroups must be balanced	Short
Deininger et al. [4]	Low	No	No	No	Short
ELAPSE [5]	High	Yes	No	Yes, subgroups must be balanced	Short
Xu et al. [10]	High	Yes	No	No	Short
Our proposal	Low	Yes	Yes	Possible	Long

our protocol deals with insider attacks and provides long lifetime for the keys. This fact together with the absence of subgroups and unicasts, considerably lessens the administration cost in BS. Also, our scheme can be further improved when subgrouping is utilized. This will downsize the volume of the broadcast messages in the BS side and enhance the overall performance.

5. Conclusions

Due to the natural attributes of wireless communication, anyone in range can intercept or inject frames, making communication much more defenseless to attacks than their wired equivalents. In this paper we focused on MBRA of 802.16e networks. All current vulnerabilities of this mechanism were exposed and all alternative solutions as proposed in the literature so far were surveyed. Analysis showed that while certain mechanisms are able to overhaul MBRA security issues to a great degree, there is still room for additional refinements. In all cases the main problem that remains unresolved is how to protect the service from malicious insiders. In this context, we proposed an asymmetric group key agreement protocol giving a variation of [3]. Our all-entity contributional scheme constructs a secret encryption key to be used by the BS but different – per MS – decryption keys. The use of a unique encryption key known only to BS, guarantees security against forgery insider attacks. Moreover, the renewal of the keys upon join and leave events, satisfies the properties of backward and forward secrecy. In terms of network costs, the proposed scheme shows good performance when compared to similar but symmetric solutions. Also, we can argue that it scales better not requiring unicasts from the BS. This actually can compensate the additional overhead produced mainly to BS due to asymmetric cryptographic operations.

References

[1] K. Lu, Y. Quian, H.H. Chen, S. Fu, WiMAX networks: From access to service platform, in: IEEE Network, IEEE Press, 2008, pp. 38–45.
[2] IEEE std 802.16e, IEEE Standard for Local and Metropolitan Area Networks, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Amendment and Corrigendum to IEEE Std 802.16-2004, published Feb. 2006, IEEE Press.
[3] Q. Wu, Y. Mu, W. Susilo, B. Qin, J. Domingo-Ferrer, Asymmetric group key agreement, in: Eurocrypt 2009, in: LNCS, vol. 5479, Springer Verlag, 2009, pp. 153–170.
[4] A. Deininger, S. Kiyomoto, J. Kurihara, T. Tanaka, Security vulnerabilities and solutions in mobile WiMAX, IJCSNS International Journal of Computer Science and Network Security 7 (11) (2007) 7–15.
[5] C.T. Huang, J.M. Chang, Responding to security issues in WiMAX networks, IEEE IT Professional 10 (5) (2008) 15–21.
[6] H. Harney, C. Muckenhirn, Group key management protocol (GKMP) specification, IETF RFC 2093, July 1997.
[7] D. Boneh, M. Franklin, Identity-based encryption from the Weil pairing, in: Advances in Cryptology—CRYPTO 2001, in: LNCS, vol. 2139, Springer-Verlag, 2001, pp. 213–229.
[8] P.S.L.M. Barreto, H.Y. Kim, M. Scott, Efficient algorithms for pairing-based cryptosystems, in: Advances in Cryptology—CRYPTO 2002, in: LNCS, vol. 2442, Springer-Verlag, 2002, pp. 354–368.
[9] H. Li, G. Fan, J. Qiu, X. Lin, GKDA: A group-based key distribution algorithm for WiMAX MBS security, in: PCM 2006, in: LNCS, vol. 4261, Springer Verlag, 2006, p. 310318.
[10] S. Xu, C.T. Huang, M.M. Matthews, Secure multicast in WiMAX, Journal of Networks 3 (2) (2008) 48–57.