Shared Authentication Information for Preventing DDoS attacks in Mobile WiMAX Networks

Youngwook Kim∗
School of EE & INMC
Seoul National University
Seoul, Korea
Email: kiyewo@netlab.snu.ac.kr

Hyyoung-Kyu Lim†
Samsung Electronics
Suwon, Korea
Email: hk03.lim@samsung.com

Saewoong Bahk∗
School of EE & INMC
Seoul National University
Seoul, Korea
Email: sbahk@snu.ac.kr

Abstract—Recent broadband wireless technologies such as HSDPA and Mobile WiMAX achieve high data rate transmission, making wireless networking environments more similar to wired environments. As a result, wireless networks are also being exposed to DDoS attack. In this paper, we consider a possible DDoS attack in Mobile WiMAX networks and solve this problem by using our proposed Shared Authentication Information (SAI). SAI exploits unused upper 64 bits of the 128-bit Cipher-based Message Authentication Code (CMAC) which has been designed to provide the integrity of management message. In the Mobile WiMAX network, the lower significant 64 bits of CMAC are truncated and used. Therefore we are able to use the upper 64 bits for our SAI while assuring the same level of security guaranteed by CMAC. Since SAI can be obtained from CMAC calculation, no additional calculations or message exchanges are required for sharing SAI and only the entity having the CMAC key can know SAI. Owing to these properties, using SAI can be a simple defense mechanism against DDoS attack without incurring overhead at access service network gateway (ASN GW) and base station (BS).

I. INTRODUCTION

Conventional wireless networks such as 2.5G cellular networks were not exposed to Distributed Denial of Service (DDoS) attack since they are mainly based on circuit switching accompanied by call admission control. However, with emergence of broadband wireless services based on packet switching like Mobile WiMAX system, cellular-like networks are no more safe areas from DDoS attack. Mobile WiMAX networks support high-speed data transmission, thereby offering various data services such as multimedia streaming, web browsing, and P2P as well as voice service. Accordingly environments in Mobile WiMAX networks are becoming more or like those in wired networks and exposed to DDoS attacks.

This motivates us to design a defense mechanism against DDoS attack in Mobile WiMAX networks. In the paper, we first identify vulnerabilities to DDoS attack in Mobile WiMAX networks and propose a simple solution to resolve this problem. DDoS attack in wireless networks so far has been focused little because of the low possibility of attack. In [12], Bellardo, et al. spotted vulnerabilities in 802.11 networks that are exposed to DoS attack using Deauthentication message or virtual carrier sensing mechanism. They mitigated intensity of attacks by delaying Deauthentication or restricting the Network Allocation Vector (NAV) range. Gupta, et al. studied DoS attack in 802.11 ad-hoc networks. They recognized vulnerabilities in the MAC layer that incur congestion-based DoS attacks and resolved the problem by focusing on MAC layer fairness [13]. For security in 3G cellular networks, P. C. Lee, et al. identified a signalling DoS attack and suggested a solution based on the statistical CUSUM method [5]. In [11], Zhang and Fang suggested the possibility of redirection attack through a false BS in a 3G network and modified the 3GPP-AKA algorithm to defend the network against it. In [10], Liang and Wang analyzed the effect of authentication on QoS by applying queueing theory. They mainly focused on the aspects of authentication cost and delay. Most works supposed that not much data traffic has been used for DDoS attack and the target network is not centralized. In contrast, the Mobile WiMAX network is centralized and expected to handle much data traffic. So DDoS attack needs to be revisited. The rest of the paper is organized as follows: Section 2 describes the background about the Mobile WiMAX network. Section 3 describes a possible DDoS attack and Section 4 proposes a defense mechanism. In Section 5 we evaluate the performance of our defense algorithm through simple analysis and simulation. Finally, we conclude our paper in Section 6.

II. BACKGROUND

A. Mobile WiMAX Network Reference Model

The Mobile WiMAX network is a broadband wireless network based on the IEEE 802.16e standard. The 802.16e standard provides wireless last mile access to fixed, pedestrian, or mobile users in Metropolitan area to replace the wired last mile access such as xDSL, ISDN, CATV, etc. The Mobile WiMAX Network Reference Model (NRM) consists of Access Service Network (ASN) and Connectivity Service Network (CSN) as shown in Fig. 1. ASN formed by Base Station (BS) and ASN Gateway (ASN GW) offers radio access to a Mobile Subscriber (MS) and CSN provides IP connectivity service to it. A BS is located at the center of a cell and provides direct radio connection to each MS. ASN GW is placed at the boundary of ASN and connects BSs to CSN. The Mobile WiMAX defines ASN GW functionalities in profiles A, B and C. In this paper, for simple explanation, we refer to profile A or
C according to which ASN GW has Authenticator and Paging Controller (PC) functionalities. As Authenticator, ASN GW stores Primary Master Key (PMK) for each MS and provides AK context generated from PMK for BS. As PC, ASN GW stores the list of MSs in idle mode and directs underlying BSs to page those MSs.

B. Idle Mode and Location Update

An MS enters idle mode after De-Registration. There are two alternatives for performing De-Registration: MS-initiated or BS-initiated. In this paper, we explain the procedures for using SAI with MS-initiated case in which the MS first sends De-Registration Request (DREG-REQ) message to its serving BS. Then the BS transfers this request to the PC by using MS-info Request (MS-info REQ) message that contains information about the MS entering the idle mode. The PC responds to the BS by using MS-info Response (MS-info RSP) message. If the request is successful, the BS sends De-Registration Command (DREG-CMD) message to the MS, and then the MS changes its state into idle mode. The ‘De-Registration’ part in Figs. 2 depicts the procedures.

An MS in idle mode should update its location prior to the expiration of idle mode timer or other location-update conditions are met. There are two location-update procedures: Secure Location Update (Secure LU) and Unsecure Location Update (Unsecure LU). In this paper, we explain our idea mainly with Secure LU. If an MS has some pending traffic or its security context expires, it should perform Re-entry to the network from Idle Mode (IM Re-entry) which follows Unsecure LU procedure. IM Re-entry (or Unsecure LU) is almost the same as the initial network entry except that the BS can omit some procedures depending on MS’s information [3]. The Details of Secure LU is given in Section 3.

C. CMAC and Shared Authentication Information

Mobile WiMAX adopted CMAC to prevent management messages from being modified and forged. Fig. 3 shows the CMAC calculation result in Mobile WiMAX.

The least significant 64 bits are provided as CMAC value and the most significant 64 bits are of no use. We name these unused 64 bits Shared Authentication Information (SAI) and will use SAI to defend the Mobile WiMAX network against DDoS attack.

III. DDoS Attack on BS and ASN GW

A. Bandwidth Allocation for Ranging (DDoS Vulnerability)

An MS maintains the quality of wireless link between MS and BS through ranging which uses Ranging Request (RNG-REQ) and Ranging Response (RNG-RSP) messages. The BS always allocates bandwidth for ranging interval which is indicated in Uplink-MAP (UL-MAP). To get bandwidth for ranging, an MS chooses an appropriate ranging code and transmits it during the ranging interval. On successful reception of the code, the BS assigns bandwidth for ranging to the MS. There is no authentication or authorization for granting bandwidth for ranging. So any MS can request bandwidth allocation for ranging. This vulnerability gives a possibility of DDoS attack to BS and ASN GW.

B. Secure Location Update

In Secure LU, MS first sends a RNG-REQ message including CMAC tuple to BS. To verify the CMAC value, the BS informs PC of LU attempt by using Location Update Request (LU REQ) message. The PC generates the AK context for the MS and replies to the BS with LU RSP message. Using the AK context, the BS verifies the CMAC value by checking CMAC Digest.
context, the BS calculates the CMAC value and compares it with the one in the RNG-REQ. If the two values are matched, the BS sends RNG-RSP and LU Confirm messages to the MS and the PC, respectively. Otherwise, the BS ignores the RNG-REQ. The Secure LU procedures are depicted in the ‘Secure LU’ part of Fig. 2.

In the above explanation, both Authenticator and PC are assumed to be in the same entity. If the two functionalities are in different entities, secure LU procedures require more signaling overhead. Context Request (Context REQ) and Context Response (Context RSP) messages are exchanged between PC and remote Authenticator in this case. The ‘separate PC and Authenticator case’ part in Fig. 2 shows the procedures.

C. DDoS attack on BS and ASN GW

Since any MS can request bandwidth for LU and transmit RNG-REQ message without authentication, malicious MSs can generate as many spurious requests as they intend to. For each RNG-REQ message, BS and ASN GW have to bear some burden. BS has to verify CMAC value and ASN GW has to generate AK context. As a BS serves tens of MSs, a PC and an Authenticator in ASN GW serve hundreds of MSs. So if MSs which are spread over several cells send LU requests almost at the same time, the generated signalling overhead and burden can cause PC and Authenticator to go down. As BS also can experience the burden in calculating CMAC value, DDoS attack on BS, PC and Authenticator can occur. If the functionalities of PC and Authenticator are integrated at the ASN GW, the overhead will be much heavier.

IV. SAI: SHARED AUTHENTICATION INFORMATION

In this section, we use SAI as an authentication token to protect Mobile WiMAX network from the possible DDoS attack. The advantages of using SAI are

1) No additional calculations are required for obtaining SAI at BS and MS because SAI is a part of the result obtained from CMAC calculation.
2) No explicit message exchanges are required between BS and MS for sharing SAI because both entities acquire SAI during the CMAC calculation & verification, respectively.
3) No other entities in the network can possibly know SAI that is never being explicitly swapped over the air.
4) The same security assurance level as of CMAC value is guaranteed because the length of SAI is the same as that of CMAC value.

A. Outline of Applying SAI

When malicious MSs are performing DDoS attack, BS and PC and Authenticator are respectively stressed by the load of verifying CMAC values, checking whether the requesting MSs are in idle mode, and generating AK contexts for MSs. By using SAI, we can avoid these unnecessary procedures under DDoS attack. Fig. 4 shows the process of using SAI for Secure LU. At the beginning, MS and PC share SAI when the MS enters idle mode. Then when the MS updates its location or re-enters normal operation, it has to submit SAI to the PC. The PC compares this value with the original shared value. If they are matched, the remaining procedures follow. Otherwise, the other procedures are aborted. SAI update should be performed for next Secure LU when Secure LU is successful.

We explain the detailed procedures in the next subsection.

B. Procedures of Applying SAI

1) Set-up of SAI: Both DREG-REQ and DREG-CMD messages can be used to extract SAI. In this paper, we use DREG-REQ for illustration. When an MS enters idle mode, the MS calculates the CMAC value for DREG-REQ, extracts SAI from it, stores the SAI, and sends DREG-REQ to the BS. The BS receives DREG-REQ, verifies the CMAC value, extracts and stores SAI if the CMAC is valid. It sends the MS’s identifier and SAI to the PC through MS-info Request message. The PC then stores this information and replies with MS-info Response message which confirms the SAI processing result. The ‘SAI sharing’ part in Fig. 4 shows the procedures.

2) Transmission and Verification of SAI: When an MS does Secure LU or IM Re-entry, it is allocated the bandwidth for ranging and sends RNG-REQ including Type, Length, Value(TLV) field for SAI. Since the SAI is included as TLV, there is no change required in the Mobile WiMAX standard or 802.16e. The specific TLV for SAI can be given as follows.

SAI TLV = {
    • Type = TBD (To Be Determined)
    • Length = 1 ~ 64 bits (depending on the required security assurance)
    • Value = high order 64 bits of CMAC value
}

Upon receiving RNG-REQ, BS passes SAI TLV to PC using LU REQ message if Ranging is for LU or IM Re-entry.
The PC verifies the SAI. If the two values are equal, the PC requests the AK context of the MS to Authenticator through Context REQ message. The Authenticator generates the AK context and sends it to the PC by Context RSP message. If the PC and the Authenticator are in the same entity, there is no exchange of Context REQ and Context RSP messages. The PC returns the AK context to the BS by LU RSP message. The BS then calculates the CMAC value of the RNG-REQ with the CMAC key of the MS, validates it and sends RNG-RSP to the MS if the CMAC value is valid. If the two SAIs are not matched, the PC notifies the failure of LU to the BS which then ignores the RNG-REQ message. So the effectiveness of DDoS attack is diminished by skipping Context REQ and Context RSP message exchange, CMAC verification, and AK context generation.

3) Update of SAI: Once SAI is used, an MS has to update the SAI since the SAI is submitted to the PC in clear text. When the MS does IM Re-entry, it goes back to normal state. So the MS can update SAI when it re-enters idle mode by de-registration. Therefore no explicit update procedures are needed for the IM Re-entry case. In the case of Secure LU, an MS returns to idle mode after secure LU. The MS and BS exchange RNG-REQ and RNG-RSP messages during Secure LU. The detailed procedures of updating SAI depend on whether we use RNG-REQ or RNG-RSP. Here, we use RNG-RSP message to illustrate the update procedures. PC validates the MS's SAI through exchanging LU REQ and LU RSP messages with the BS. Then the BS creates RNG-RSP message, updates SAI using this RNG-RSP value of this RNG-RSP message and sends the RNG-RSP to the MS. The BS also transmits LU confirm message to the PC to inform the updated SAI. As a result, the PC and MS again are with the same new SAI. 4 shows the process of storing, verifying, and updating SAI in the case of Secure LU.

V. SIMULATION & ANALYSIS

In this section, we show how much load-reduction we can achieve by using SAI under DDoS attack and how much overhead SAI imposes on the system in normal situation. We first measure or calculate the average required time for finishing each workload and then add up to obtain the total reduced overload under DDoS attack. We represent the total reduced overload as the decreased delay or response time by comparing the cases of with and without SAI.

A. Simulation Outline

We modified the standard C-implementation of AES128-CMAC algorithm and dot16KDF function to apply SAI [14][15][17]. We measured the number of CPU cycles required for each workload using the rdtsnc instruction [16]. The required number of CPU cycles is averaged over 1000 simulation runs. The measured workloads are of calculating CMAC value, generating AK context, and transmitting AK context. Table I summarizes the simulation environments.

<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Pentium IV 2.4 GHz</td>
</tr>
<tr>
<td>Memory</td>
<td>512 MB</td>
</tr>
<tr>
<td>OS</td>
<td>Windows XP</td>
</tr>
<tr>
<td>RNG-REQ Message Size</td>
<td>64 Bytes</td>
</tr>
<tr>
<td>block cipher</td>
<td>AES-128</td>
</tr>
<tr>
<td>C compiler</td>
<td>lcc v4.0</td>
</tr>
</tbody>
</table>

B. CMAC value Calculation

The C-implementations of CMAC algorithm in RFC 4493 [15] and AES128 in Linux WPA/WPA2 supplicant [14] are used to measure the required CPU cycles in calculating CMAC value. We randomly generated 64-byte RNG-REQ messages and 128-bit CMAC keys as inputs to AES128-CMAC algorithm and gauged the number of CPU cycles using the rdtsnc instruction. The 64-byte length of RNG-REQ is assumed because RNG-REQ is expected to have 44 ~ 51-byte size depending on whether we use Secure LU or IM Re-entry and AES128 algorithm accepts only the 128-bit block as input, so the length of message is rounded up to 64 bytes. Then the CMAC calculation overhead is given as

20116 [cycles/CMAC message] / 2.4 GHz = 8.4 µs.

C. AK context Transmission

The overload in transmitting AK context consists of message transfer from application bufer to kernel buffer, message transmission, and message propagation through the wire to the destination. We add each delay to obtain the total transmission overload. To measure CPU cycles, we transmit randomly generated RNG-REQ through UDP-IP-Ethernet protocol stack. We use UDP between BS and ASN GW because it is the transport protocol in WiMAX network. Then the transmission overhead in time is given as

72009 [cycles/UDP message] / 2.4 GHz = 30 µs.

To measure the transmission delay, we assume a 100Mbps ethernet connection between BS and ASN GW. The minimum length of LU REQ without SAI is 44 bytes and with SAI 56 bytes. As LU RSP message contains MSID, BSID, AK Context, LU_Status, and other TLVs, its minimum length is 232 bytes normally, but with SAI it is 52 bytes under DDoS attack since AK context is omitted. So the total transmission times of LU REQ without SAI, LU REQ with SAI, LU RSP without SAI, and LU RSP with SAI under DDoS attack are given as follows, respectively.

\[ T_{\text{req}} = (42+44) \times 8 [\text{bits/byte}] / (100 \times 10^6 \text{bps}) = 6.88\mu s \]
\[ T_{\text{SAI}}^{\text{req}} = (42+56) \times 8 [\text{bits/byte}] / (100 \times 10^6 \text{bps}) = 7.84\mu s \]
\[ T_{\text{rsp}} = (42+232) \times 8 [\text{bits/byte}] / (100 \times 10^6 \text{bps}) = 21.92\mu s \]
\[ T_{\text{SAI}}^{\text{rsp}} = (42+52) \times 8 [\text{bits/byte}] / (100 \times 10^6 \text{bps}) = 7.52\mu s \]
In calculation, we counted 14-byte of Ethernet header, 20-byte of IP header, and 8-byte of UDP header, i.e, total of 42 bytes. Assuming the distance between BS and ASN GW to be 12Km (40 \( \mu s \)), the total transmission delay for LU REQ without SAI and with SAI are 76.9 \( \mu s \) (= 30 + 6.88 + 40) and 77.8 \( \mu s \) respectively. For LU RSP under DDoS attack, they are 91.9 \( \mu s \) and 77.5 \( \mu s \) respectively.

**D. AK context Generation**

AK context consists of AK, AKID, CMAC_KEY_U, CMAC_KEY_D, KEK, EIK, and others [3][1]. Among these, AK, AKID, CMAC_KEY_U, CMAC_KEY_D, and KEK have to be regenerated whenever the BS requests AK context to the ASN GW for Secure LU, which becomes a burden under DDoS attack. We estimate the computational load for AK context generation. Table II summarizes the results.

In normal situation, the computational load without SAI is 106423 cycles and with SAI 106530 cycles so the ratio is 1.001, which means 0.1% additional load is incurred by SAI. With SAI, if there is no attack, ASN GW has one additional step of SAI comparison before generating AK context. Under DDoS attack, the required number of CPU cycles without SAI is 106423 cycles while with SAI it is 8 cycles, which ratio is 0.001. This is because with SAI, most requests end up with a single comparison without incurring AK context generation. This results in a remarkable performance improvement under DDoS attack. Therefore using SAI is an efficient defense mechanism against DDoS attack.

**E. Total Overload**

We sum up all the previous overloads to compute the total overload according to attack intensity and compare the results for the cases of DDoS attack and normal operation. For representing attack intensity, we use the attack ratio which is the ratio of attack messages to total messages at the BS. The overload sum at BS and ASN GW forms the total overhead in the WiMAX system. We assume that PC and Authenticator are in the same entity following the WiMAX ASN GW profile A and C. If PC and Authenticator are in different entities, we can reduce the overhead more under DDoS attack. Tables III summarizes the result. The total overhead under DDoS with SAI is 155.4 \( \mu s \) while without SAI it is 221.6 \( \mu s \), which is 30% of improvement. In normal situation, the total overloads with SAI and without SAI are almost the same.

Using Tables III, we can represent the total overloads for the cases with SAI and without SAI as 222.5\((1 - x) + 155.4x\) \( \mu s \) and 221.6 \( \mu s \), respectively, where \( x \) is the attack ratio. We can decrease the total overloads by up to 30% depending on the attack ratio. For instance, when the attack ratio is 0.9, the overhead has been reduced by about 27%.

**VI. Conclusion**

In this paper, we considered a possible DDoS attack in Mobile WiMAX network when idle mode is involved. The validation process for CMAC value can cause DDoS attack if malicious MSs forge RNG-REQ messages. Against this type of DDoS attack, our proposal uses SAI mechanism that efficiently reduces the overhead by up to 30% depending on the attack ratio.

**REFERENCES**


[17] https://www.deadhat.com/wmancrypto/