Key Management Protocol in WIMAX Revisited

Noudjoud Kahya, Nacira Ghoualmi, Pascal Lafourcade

Key Management Protocol in WIMAX Revisited

Noudjoud Kahya[1], Nacira Ghoualmi[1], and Pascal Lafourcade[2]

[1] Networks and Systems Laboratory (LRS); Badji Mokhtar University, Annaba, Algeria
[2] VERIMAG Laboratory; Joseph Fourier University, Grenoble, France

Abstract. Without physical boundaries, a wireless network faces many more vulnerabilities than a wired network does. Compared to Wi-Fi, security has been included in the design of WiMAX systems at the very start. IEEE802.16 standard (WiMAX) provides a security sublayer in the MAC layer to address the privacy issues across the fixed BWA (Broadband Wireless Access). After the launch of this new standard, a number of security issues were reported in several articles. Ever since the beginning, work has been in progress for the neutralization of these identified threats.

In this paper, we first overview the IEEE802.16 standard, especially the security sublayer, and then authorization protocol PKM in WiMAX has been analyzed. We found that PKM (Privacy and Key Management) is vulnerable to replay, DoS, Man-in-the middle attacks and we propose a new methodology to prevent the authorization protocol from such attacks.

We also give a formal analysis of authentication protocol (PKMv2) and for the proposed protocol; we conclude that our proposition prevent the attacks like Denial of service (DOS), Man-in-the-middle and replay. The formal analysis has been conducted using a specialized model checker Scyther, which provides formal proofs of the security protocol.

1 Introduction

Scyther tool were developed by Cas Cremers in 2007 [1]. Scyther, is a formal protocol analysis tool, for the symbolic automatic analysis of the security properties of cryptographic protocols (typically confidentiality or variants of authenticity). It assumes perfect cryptography, meaning that an attacker gains no information from an encrypted message unless he knows the decryption key. Scyther takes as input a role-based description of a protocol in which the intended security properties are specified using claims. Claims are of the form claim (Principal, Claim, Parameter), where Principal is the user's name, Claim is a security property (such as 'secret'), and Parameter is the term for which the security property is checked.

The aim of this paper is using Scyther tool to verify the security properties and discover the vulnerabilities in Wimax (Worldwide Interoperability for Microwave Access).

Wimax is a broadband wireless system which offers packet switched services for fixed, nomadic, and mobile accesses. Wimax utilizes many advanced technologies and mechanism in the physical and medium access control (MAC) layers to provide high spectrum efficiency and protect the traffic confidentiality and integrity and to prevent different network security attacks. The 802.16 standard (Wimax) specifies a security sublayer at the bottom of the MAC layer. This security sublayer provides SS

D.C. Wyld et al. (Eds.): Advances in Computer Science, Eng. & Appl., AISC 167, pp. 853–862.
springerlink.com

(Subscriber Station) with privacy and protects BS (Base Station) from service hijacking. There are two component protocols in the security sublayer: an encapsulation protocol for encrypting packet data, and a PKM (Privacy and Key Management) protocol for providing the secure distribution of keying data from BS to SS as well as enabling BS to enforce conditional access to network services.

The contribution of this work is twofold: first, we formally and analyze PKMv2 protocol with scyther tool to extract holes or threat that might exist. Second, we propose a new protocol and we also use the formal method to verified if our proposed revision resolute the security problems of the PKMv2 protocol.

This paper is organized as follows. In Section 2, we provide background and detailed information about Wimax architecture and Privacy and Key Management (PKM) protocol. Section 3, we describe the designs of scyther tool and we performing an evaluation the security objectives. In Section 4, we model and analyze PKMv2 with Scyther tool. Section 5, covers the proposed solution and modified authentication model. Finally, we conclude in section 6.

2 Security Requirements

IEEE 802.16 is the standard to specify the air interface of fixed BWA. IEEE standard 802.16-2001 [2] was first designed to provide the last mile for Wireless Metropolitan Area Network (WMAN) with line-of-sight (LOS) working at 10-66GHz bands. The latest version, IEEE standard 802.16-2004 [3], which consolidates previous standards, also supports non-line-of-sight (NLOS) within 2-11 GHz bands and mesh nodes. The recently released amendment, IEEE 802.16e [4], aims to provide mobility in WMAN.

The protocol architecture of Wimax is structured into two main layers of OSI model: the Medium Access Control (MAC) layer and physical layer. The MAC layer consists of three sublayers: the service-specific convergence sub-layer (CS), MAC common part sub-layer (MAC CPS), and security sub-layer [5]. Security sub-layer has two goals, one is to provide privacy across the wireless network and the other is to provide access control to the network. By encrypting connections between the SS and the BS, privacy is accomplished by enforcing encryption of service flows across the network, the BS protects against unauthorized access. The base station uses a Privacy and Key Management (PKM) protocol to control the distribution of secret data to subscriber stations. We will focus on PKM because it is the main part of security.

PKM provides the authorization process and secure distribution of keying data from the BS (base station) to SS/MS (mobile station). BS uses the protocol to enforce conditional access to network services.

The IEEE 802.16 PKM protocol uses X.509 digital certificates, RSA public-key algorithm, and strong encryption algorithm to perform key exchanges between SS and BS, at client/server model. IEEE 802.16 PKM employs two-tier key systems. The Authentication Protocol first authenticates SS to BS, establishing a shared secret (Authorization Key, AK) via public-key cryptography; then via Key Management Protocol, SS registers to the network, during which AK is used to secure the exchange of Transport Encryption Keys (TEK) [6].

3 Security Property

3.1 Scyther Tool

Scyther is an automatic tool for the verification and falsification of security protocols. Scyther provides a graphic user interface which incorporates the scyther command line tool and python scripting interface. Scyther tool tales protocol description and optimal parameters as intput, and output a summary report and display a graph for each attack. The description of a protocol is written in SPDL language [7]. Security properties are modeless as claim events. Claim (Principal, Claim, and Parameter), where Principal is the user's name, Claim is a security property, and Parameter is the term for which the security property is checked.

For the protocol verification, Scyther can be used in three ways [7]:

- *Verification claim:* Scyther verifies or falsifiers security properties.
- *Automatic claims:* if user does not specify security properties as claim event the scyther automatically generates claims and verifies them.
- *Characterization:* each protocol role can be characterized. Scyther analyses the protocol and provides a finite representation of all traces that contain an execution of the protocol role.

Scyther generates attack graph for counter example, and represents individual attack graph for each claim.

3.2 Security Propriety

All security solution for WiMAX network should satisfy the requirements as follows.

Property 1- Confidentiality: This claim is fulfilled if the MS/SS has the guarantee that all exchanged user data is secret. The exchanged user data messages between the MS and the BS is called Msg. Every information (α) in Msg should remain secret. The formalization of information confidentiality is given below.

$$\forall \alpha \in \mathrm{Msg}(\mathrm{claim}(\mathrm{SS}, \mathrm{Secret}, \alpha)) \tag{1}$$

Property 2- Authenticity: This claim is fulfilled if an outsider, who keeps track of the communication, cannot relate the traffic to a specific MS. In order to fulfill authenticity the MAC address of the MS which identifies it must remain secret. The MAC address is included in the MS's certificate (MsCert). The formal definition of pseudonymity is given below.

$$\mathrm{claim}(\mathrm{SS}, \mathrm{Secret}, \mathrm{SSCert}) \tag{2}$$

Property 3- **Integrity:** This claim is fulfilled if the BS and the SS have the guarantee that all exchanged keys (described as key) are secret and unique. We have included an additional restriction that only claims concerning sessions between trusted agents are evaluated. Its formal definition is shown as follows:

$$\forall \mathrm{key}(\mathrm{claim}(\mathrm{BS/SS}, \mathrm{Secret}, \mathrm{key})) \tag{3}$$

***Property 4-* Access control:** A WiMAX network should have a correct mechanism to verify that a given user is authorized to use a particular service[8]. Furthermore, access control can guarantee that only authorized users are allowed to connect to a given network and get access to the offered services. A service should always be bound to an authenticated user. Its formal definition is given as follows:

$$\forall \alpha \in \mathrm{Msg}(\mathrm{claim}(\mathrm{BS}, \mathrm{Secret}, \alpha)) \tag{4}$$

Property 5- Freshly of messages: An important part of security protocols is the generation of fresh values which are used for challenge-response mechanisms (often called nonces), or as session keys. This claim is fulfilled if the BS and MS/SS have the guarantee that the session key is fresh.

$$(\mathrm{claim}(\mathrm{BS/SS}, \mathrm{Fresh}, \mathrm{key})) \tag{5}$$

4 Modeling and Analyzing Pkmv2 Protocol

In this section, we model PKMv2 protocol in Scyther tool and we verify if the five properties (claim events) are respected.

4.1 PKMv2 Protocol

The latest standard, IEEE 802.16e-2005, includes a new version (PKMv2). The major security problems were solved in PKMv2. It makes authorization procedure secure enough to prevent attacks. After initial authorization, PKMv2 also checks for reauthorization periodically. Complete authorization procedure has been defined by David and Jesse in [9].

PKMv2 supports two different mechanisms for authentication: the SS/MS and the BS may use RSA-based authentication or Extensible Authentication Protocol (EAP) - based authentication. We will focus in this paper on RSA based authentication for PKMv2 authentication protocol. The flow of messages exchange in RSA-based authentication is shown as follows:

msg1. SS→BS: *Mancert (SS)*
msg2. SS→BS: *{ N_{SS}, SSCert, Capabilities, BCID }sk(SS)*
msg3.BS→SS:*{N_{SS},N_{BS},{prePAK,SSID}pk(SS),SAIDlist,prePAKSeq,prePAKliftime,BSCert}sk(BS);*
msg4. SS→BS: *N_{BS}, SSaddr,{N_{BS}, SSaddr}AK;*

SS/MS sends its MCerSS (manufacturer's certificate) and then sends its own CerSS which is X.509 certificate along with a nonce; a 64 bit random number generated by the SS/MS, BC-Identity and cryptographic Capb (capabilities). BC-Identity is assigned to SS/MS when it enters in a network and requests for ranging. After receiving the authorization request message from SS/MS, BS responds by sending some information and a nonce; one generated by the BS and one that SS/MS sends in its request's message. BS also attaches its certificate (CerBS) in response to SS/MS for mutual authentication. BS also includes its signatures for validity in response message to SS/MS. A 256 bit key (Pre-Au-K) with the SS's identifier (SSID) is encrypted by the BS with the public key of SS/MS. A 4 bit sequence number for the authorization key (Seq_No) and its life time with the SAID's List (SAIDL) are sent by the BS.

After validating the message from BS, the SS/MS sends the acknowledgement message with nonce created by BS and MAC address (MAC_{SS}) of the subscriber station.

Authorization Key (AK) transmitted by BS to SS/MS in previous message is used to encrypt the NonceBS (BS generated random number) and MAC_{SS} [10].

4.2 Modeling PKMv2

In scyther, a protocol is described in SPDL language in which agent defined a role. PKMv2 can be modeled as follows:

```
// The protocol description
protocol pkmv2(SS,BS,CA)
{
role SS
{
const Mancert,cap,SAID: Data;
var CerSS,CerBS:Data;
const Ns:Nonce;
var Nb:Nonce;
var SAIDlist,AKSeq,AKlifetime:Data;

send_1(SS,BS,Mancert (SS));
send_2(SS,CA,SS);
read_3(CA,SS,{SS,{CerSS,pk(SS)}sk(CA)}sk(CA));
send_4(SS,BS,{CerSS,pk(SS)}sk(CA));
send_5(SS,BS,{cap,SAID,Ns,SS});
read_8(BS,SS,{CerBS,pk(BS)}sk(CA));
read_9(BS,SS,{{preAK}pk(SS), AKSeq,AKlifetime, SAIDlist,Ns,Nb}sk(BS));
send_10(SS,BS,{Nb,SS}AK);

}
role BS
{
var CerBS,CerSS,Mancert,cap,SAID: Data;
const Nb:Nonce;
var Ns:Nonce;
const SAIDlist,AKSeq,AKlifetime:Data;

read_1(SS,BS,Mancert (SS));
read_4(SS,BS,{CerSS,pk(SS)}sk(CA));
read_5(SS,BS,{cap,SAID,Ns,SS});
send_6(BS,CA,BS);
read_7(CA,BS,{BS,{CerBS,pk(BS)}sk(CA)}sk(CA));
send_8(BS,SS,{CerBS,pk(BS)}sk(CA));
send_9(BS,SS,{{preAK}pk(SS), AKSeq,AKlifetime, SAIDlist,Ns,Nb}sk(BS));
read_10(SS,BS,{Nb,SS}AK);
}
role CA
{
const CerSS,CerBS: Data;
read_2(SS,CA,SS);
send_3(CA,SS,{SS,{CerSS,pk(SS)}sk(CA)}sk(CA));
read_6(BS,CA,BS);
send_7(CA,BS,{BS,{CerBS,pk(BS)}sk(CA)}sk(CA));
}
}
```

4.3 Analysis of PKMv2

This model is going to be challenged with the following requirements using the Scyther tool.

1. Property 1: Scyther identified problems in the confidentiality protocol. It is a passive attack on confidentiality. An intruding entity eavesdrops the second message (Auth-REQ) and he is able to read the information that is sent from the SS/MS to the BS, gathering information about the trusted SS/MS (cryptographic capabilities and security association identifier (SAID)).

2. *Property 2:* Scyther detected a possible Authenticity attack. Message2 is sent in plaintext so an intruder eavesdrops this message and obtains the SS's certificate (MsCert). BS may face a replay attack from a malicious SS who intercepts and saves or modified the authentication messages sent by a legal MS/SS previously.

Property 3: it is proved that the authorization key exchanged in the authentication protocol is secret.

Property 4 and 5: It is proved that an adversary cannot obtain the pre-PAK, which will be used to extract the AK and the session key is fresh, as it is encrypted with the public key of the SS.

As seen in the formal analysis, the *secrecy of the keying* material distributed claim is valid in PKMv2. However, *Authenticity, integrity* and information *confidentiality* are broken, PKMv2 still vulnerable to replay, DoS and Man-in-the-middle attacks.

5 The Proposed Revised Authentication Protocol

As discussed in the previous section, the PKMv2 protocol does not fulfill the claims pseudonymity and information confidentiality because it still vulnerable to replay, DoS and Man-in-the-middle. In related works the nonce is used to prevent replay and man-in-the middle attacks, Nonce indicate that the requests were not used before, but he will not give any information about the time that was sent. Nonce is also not sufficient to tell the BS that it is the current message received from the SS/MS. In our revised protocol to assure synchronization between SS/MS and BS both nonce and timestamp are use. So the revised protocol has the timestamp attached with the SS/MS message to the BS along with the nonce. The protocol will be described as follows.

msg1. SS→CA: SS
msg2. CA→SS:{SS,{CertSS,pk(SS)}sk(CA)}sk(CA)
msg3. SS→BS:{{CertSS,N_s}pk(CA)}sk(SS)
msg4. BS→CA: BS
msg5. CA→BS:{BS, {CertBS, pk(BS)}sk(CA)}sk(CA)
msg6. BS→CA: {{{CerSS, N_s }pk(CA)}sk(SS), CertBS,N_b}sk(BS)
msg7. CA→BS:{{CerSS, N_s, N_b }pk(BS), {CerBS, N_s, N_b }pk(SS)}sk(CA))
msg8. BS→SS:{{CerBS, N_s, N_b }pk(SS)}sk(CA)
msg9. SS→BS:{{Ts, N_b,cap,SAID}pk(BS));
msg10. BS→SS:{{prePAK(BS)}sk(BS),SAIDlist,Ts,Tb, N_s, preSeq,prePAKlifetime}pk(SS)
msg11. SS→BS:{Tb, N_b }sk(SS)

The new protocol can be divided into four main stapes:

1-Certificates Register: SS/MS and BS send a message to find an X.509 certificate and it own public key information onto the server CA. This first step contained 1), 2) and 4), 5) messages: CA is only as a certification center which does not participant in the session key exchange.

2-Certificates Exchange: SS/MS and BS exchange their certificates through the trusted server CA in order to decide if etch particular is a trusted device or not. This step contained 3), 6), 7), 8) messages

3-Authorization request message: SS/MS sends a message contains the SS/MS certificate (SsCert) and a nonces (N_s) used for registration and exchange certificates, it also contains the timestamp of SS/MS along with SAID and its security capabilities. Authorization request message is encrypted with the public key of the BS pk(Bs), the timestamp addition could bring an extra layer of security since the BS could identify the message as current one. The timestamp could avoid the intruders who are trying to synchronize time with either BS or SS/MS.

4- Authorization reply message: If BS determines that the MS/SS is authorized it replies with a message authorization reply message. BS sends nonce (N_s) which was sent by the SS. That could ensure SS/MS that message 10 is the reply of the request send by SS/MS itself. BS Nonce ensures the SS about the authentication of BS. This mutual authentication gives extra layer of security. BS sends a pre-AK encrypted with the private key of BS sk(BS). From pre-PAK, the SS generates AK. If AK is used correctly, then SS gains the authorization to access the WIMAX channel. The message contained also Lifetime of Pre-AK a Sequence number of pre-AK. BS sends his Timestamp (Tb) to grant that is not copied by adversaries, the timestamp and the nonce of BS previously received to confirm authorization access. BS encrypted the message with his public key.

***5- Verification the information integrity*:** The last message ensures that the message is from the actual BS. Two layers of assurance are provided in this message: the nonce (N_b), and time stamp sent by BS (Tb). SS use it signature to ensure that message is from an actual SS and to assure the information integrity.

5.1 Modeling New Protocol in SPDL language

The new version of the PKM protocol can be modeled in SPDL as follows:

```
// The protocol description
protocol new version(SS,BS,CA)
{
role SS
{
const cap,SAID:Data;
var prePAKSeq,prePAKlifetime,CerSS,CerBS,  SAIDlist: Data;
var Ns:Nonce;
const Nb:Nonce;
var Ts:TimeStamp;
const Tb:TimeStamp;
var prePAK:SessionKey;
```

```
send_1(SS,CA,SS);
read_2(CA,SS, {SS,{CerSS,pk(SS)}sk(CA)}sk(CA));
send_3(SS,BS,{{CerSS,Ns}pk(CA)}sk(SS));
read_8(BS,SS, {{CerBS,pk(BS),Ns,Nb}pk(SS)}sk(CA));
send_9(SS,BS,{Ts,Nb,cap,SAID}pk(BS));
read_10(BS,SS,{{prePAK}sk(BS), SAIDlist,Ts,Tb,prePAKSeq,prePAKlifetime}pk(SS));
send_11(SS,BS,{Tb,SS}pk(BS));

claim_ss1(SS, Secret,CerSS);
claim_ss2(SS, Nisynch);
claim_ss3(SS, Niagree);
claim_ss4(SS, Secret,Data);
claim_ss5(SS,Secret,prePAK);
claim_ss8(SS,Secret,Ns);
claim_ss11(SS,Empty,(Fresh,prePAK));
}
role BS
{
const  prePAKSeq,prePAKlifetime, SAIDlist: Data;
var  Ns:Nonce;
const Nb:Nonce;
const Ts:TimeStamp;
var Tb:TimeStamp;
var cap,SAID,CerSS,CerBS:Data;
const prePAK:SessionKey;

read_3(SS,BS, {{CerSS,Ns}pk(TS)}sk(SS));
send_4(BS,CA, BS);
read_5(CA,BS, {BS,{CerBS,pk(BS)}sk(CA)}sk(CA));
send_6(BS,CA,{{{{CerSS,Ns}pk(CA)}sk(SS),CerBS,Nb}pk(CA)}sk(BS));
read_7(CA,BS,{{CerSS,pk(SS),Ns,Nb}pk(BS)}sk(CA),{{CerBS,pk(BS),Nb,Ns}pk(SS)}sk(CA));
send_8(BS,SS, {{CerBS,pk(BS),Ns,Nb}pk(SS)}sk(CA));
read_9(SS,BS,{Ts,Nb,cap,SAID}pk(BS));
send_10(BS,SS,{{prePAK}sk(BS),SAIDlist,Ts,Tb,  prePAKSeq,prePAKlifetime}pk(SS));
read_11(SS,BS,{Tb,SS}pk(BS));

claim_bs1(BS, Secret,CerBS);
claim_bs2(BS, Nisynch);
claim_bs3(BS, Niagree);
claim_bs4(BS, Secret,Nb);
claim_bs8(BS,Secret,prePAK);
claim_bs11(BS,Empty,(Fresh,prePAK));
}
role CA
{
const  Nb,Ns:Nonce;
const CerBS: Data;
const CerSS: Data;
      read_1(SS,CA, SS);
      send_2(CA,SS, {SS,{CerSS,pk(SS)}sk(CA)}sk(CA));
      read_4(BS,CA, BS);
      send_5(CA,BS,{BS,{CerBS,pk(BS)}sk(CA)}sk(CA));
      read_6(BS,CA,{{{{CerSS,Ns}pk(CA)}sk(SS),CerBS,Nb}pk(CA)}sk(BS));

send_7(CA,BS,{{CerSS,pk(SS),Ns,Nb}pk(BS)}sk(CA),{{CerBS,pk(BS),Nb,Ns}pk(SS)}sk(CA));
}
}
```

5.2 Analysis the New Version

This model is going to be challenged with the following requirements using the Scyther tool.

1. *Property 1 and 2:* In the formal analysis it is proved that an intruder cannot obtain the SS/MS certificate (MsCert) and data exchange between SS and BS.

2. Property 3: In the formal analysis it is proved that the authorization key exchanged in the authentication protocol is secret and not broken.

3. Property 4: It is proved that unauthenticated user cannot access the services provided, and cannot impersonate another user. Also, it is not possible to modify the data by an unauthorized individual.

4. *Property 5:* It is proved that an adversary cannot obtain the unique pre-PAK. Timestamp and nonce are used in the revised protocol to prevent replay and man-in-the-middle attack. The SS/MS appends the time stamp and nonce. This helps the BS to identify the request as a newer one. The nonce will wipe out the possibility of replay attack.

The nonce helps the BS to identify successive requests and it enhances the BS capacity to reject those requests which was sent by the intruders or adversaries so to prevent DOS attack. BS, thus, can identify the latest requests and it is able to filter out samples of replay attacks. In stapes authorization reply message, the BS sends the timestamp and nonce of SS/MS. That helps in preventing an adversary from forging a BS. This protocol also provides mutual authentication. The nonce value sent by the BS helps in preventing the man-in-the middle attack. The revised protocol helps SS/MS and BS exchange their certificates through the trusted server CA in order to decide if etch particular is a trusted device or not; hence it avoids the possibility of the DoS attack.

Second, the timestamp helps the BS in identifying the latest requests, which prevents reply attacks. It also helps the SS/MS to identify the recent messages, and hence it can identify the AK used by the SS/MS as new or not. The addition of nonce from the BS helps the SS/MS to identify whether the message which he received with pre-AK is a newer one or not. It is better to add more buffers to carry the used nonce values in the previous sessions. This gives more security to the BS and user SS/MS.

6 Conclusion

The paper analyzes the vulnerabilities in the basic authentication protocol PKMv2. As seen in the formal analysis, we formally verified the key management protocol of PKMv2 in terms of the secure session key establishment and distribution, confidentiality, authenticity, integrity, access control.

The *secrecy of the keying* material distributed claim is valid. However, *Authenticity, integrity* and information *confidentiality* are broken in PKMv2.

A revised authentication protocol is proposed by using nonce and timestamp together. The new solution is efficient to tackling the various security threats such as replay, man in the middle and DOS attacks. The revised authentication protocol is expected to provide better secure platform for IEEE 802.16(e).

References

[1] Cremers, C.: Scyther-Semantics and verification of security protocols. PhD dissertation; Eindhoven University of technology (2006)
[2] IEEE Std. 802.16-2001, IEEE Standard for Local and Metropolitan Area Networks Part16: Air Interface for Fixed Broadband Wireless Access Systems, IEEE 2002 (2002)
[3] IEEE Std. 802.16-2004, IEEE Standard for Local and Metropolitan Area Networks Part16: Air Interface for Fixed Broadband Wireless Access Systems, IEEE 2004 (2004)
[4] IEEE Std. 802.16e-2005, IEEE Standard for Local and Metropolitan Area Networks Part16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, IEEE 2006 (2006)
[5] Abbaci-kahya, N., Ghoualmi, N.: Security in Wimax. In: International Conference on Information Technology and e-Services, Tunisia (2011) ISBN 978-9938-9511-03
[6] Xu, S., Huang, C.T.: Attacks on PKM protocols of IEEE 802.16 and its later versions. In: Proceedings of 3rd International Symposium on Wireless Communication Systems (ISWCS 2006), Valencia, Spain (2006)
[7] Cremers, C.J.F.: The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols. In: Gupta, A., Malik, S. (eds.) CAV 2008. LNCS, vol. 5123, pp. 414–418. Springer, Heidelberg (2008)
[8] Lang, W.-M., Wu, R.-S., Wang, J.-Q.: A Simple Key Management Scheme based on WiMAX. In: International Symposium on Computer Science and Computational Technology, IEEE 2008 (2008)
[9] Johnston, D., Walker, J.: Overview of IEEE 802.16 security. IEEE Security and Privacy Magazine 2(3), 40–48 (2004)
[10] Altaf, A., Younus Javed, M., Ahmed, A.: Security Enhancements for Privacy and Key Management Protocol in IEEE 802.16e-2005. College of Signals, NUST. In: Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, IEEE 2008 (2008)