SECURE KEY MANAGEMENT PROTOCOL IN WIMAX

Noudjoud Kahya[1], Nacira Ghoualmi[2], Pascal Lafourcade[3]

[1, 2] LRS Laboratory; Badji Mokhtar University, Annaba, Algeria
[1] Kahya.noudjoud@gmail.com, [2] Ghoualmi@yahoo.fr
[3] VERIMAG Laboratory; Joseph Fourier University, Grenoble, France.
[3]pascal.lafourcade@imag.fr

ABSTRACT

The Worldwide Interoperability for Microwave Access (WIMAX /IEEE 802.16), is new technology based on wireless metropolitan area network. Security of connections access in WIMAX /IEEE 802.16 is complete with respect to the Privacy Key Management (PKM) protocol. The protocol is responsible for providing the secure distribution of keying data from Base Station (BS) to Subscriber Station (SS). In this paper we provide the formal analysis of PKMv2 using Scyther tool to verify the security properties. We found that PKMv2 is vulnerable to replay, DoS, Man-in-the middle attacks. At last we have proposed a secure protocol (SPKM) to prevent the authorization protocol from such attacks.

Keywords

WIMAX , PKM, Nonce, Timestamps, Analyze formal.

1. INTRODUCTION

WIMAX defines Privacy Key Management (PKM) protocol in security sub-layer, which assures the security of connections access in WIMAX channel. PKM protocol has two goals, one is to provide the authorization process and the other is to secure distribution of keying data from the BS (Base Station) to SS/MS (Mobile Station).

The PKM protocol is comparable to a conventional a client/server model, where the SS proceeds as a client to request keying material and the BS responds to these requests, making sure that the client is authorized to get the key material associated with the services that he is authorized to access. PKM uses X.509 certificates and symmetric cryptography to secure key exchange between an SS and a BS. Currently, there are two versions of PKM. The original design PKMv1, it is defined for use in the IEEE 802.16-2004 standard. This version was later extended PKMv2 to cope with mobility in the IEEE 802.16e standard.

The contribution of this work is twofold: first, we formally and analyze PKMv2 protocol with scyther tool [1] 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.

Overview

We give background and detailed information about WIMAX architecture, securities specifications and Privacy and Key Management (PKM) protocol in the Section 2. Section 3, we describe the designs of scyther tool and we performing an evaluation the security objectives.

DOI : 10.5121/ijnsa.2012.4609

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. BACKGROUND ON WIMAX

The IEEE 802.16 standard (mobile broadband wireless access system), which is also known as worldwide interoperability for microwave access (WIMAX), is a telecommunications technology that provides for the wireless transmission of data in a variety of ways, ranging from point-to-point links to full mobile cellular-type access [2]. The challenge of WIMAX is to ensure the quality of service, security in wireless communication.

IEEE standard 802.16-2001 [3] published on 2002, was first designed to provide the last mile for Wireless Metropolitan Area Network with line-of-sight (LOS) within 10-66GHz bands. The previous standards are consolidated in IEEE standard 802.16-2004 [4] (named 802.16d), which is stationary WIMAX, supports non-line-of-sight (NLOS), its licensed bands are between 2-11 GHz. The new IEEE standard 802.16e [5] published on February 2006, provides mobility in WIMAX. Table 1 review for WIMAX technology standards and versions.

Standard	802.16	802.16a/802.16-2004	802.16e-2005
Date Completed	December 2001	June 2004	December 2005
Spectrum	10-66 GHz	< 11 GHz	< 6 GHz
Modulation	QPSK 16-QAM 64-QAM	OFDM 256 subcarrier QPSK 16-QAM 64-QAM	Same as 802.16a
Channel condition	LOS	NLOS	NLOS
Bit Rate	32-134 Mbps	> 75Mbps	> 15 Mbps
Cell Radius	1-3 miles	3-5 miles	1-3 miles

Table 1. WIMAX standards and versions.

2.1 Architecture of WIMAX

WIMAX is structured into two layers: the physical layer and the Medium Access Control (MAC) layer.

The first layers in the protocol architecture of IEEE 802.16, named physical layer;

The Physical layer supports four physical specifications which are Wireless-MAN-SC (single carrier), OFDM (orthogonal frequency division multiplexing), and OFDMA (orthogonal frequency division multiple access) for ***the licensed bands***. In addition, IEEE 802.16 also supports wireless high-speed unlicensed MAN (Wireless HUMAN) specifications for the ***unlicensed bands***. Most physics operation in frequency bands a lower 11 Ghz and designed for non-line-of-sight (NLOS), except Wireless-MAN-SC, which is for operation in the 10-66 Ghz frequency band.

The second layers in the protocol architecture of IEEE 802.16: the MAC layer which is structured into three sub-layers: the service-specific Convergence Sub-layer, MAC Common Part Sub-layer, and Security Sub-layer [6].

The service specific Convergence Sub-layer (CS) maps higher level data services to MAC layer service flows and connections. There are two type of Convergence Sub-layer: ATM CS which is designed for ATM network and service, and Packet CS which supports Ethernet, point to-point protocol (PPP), both IPv4 and IPv6 internet protocols, and virtual local area network (VLAN) [6].

The MAC Common Part Sub-layer (MAC CPS) defines the rules and mechanisms for system access, bandwidth allocation and connection management [7].

The Security Sub-layer is responsible for encryption and decryption of data traveling to and from the PHY layer, and it is also used for authentication and secure key exchange [6] [7].

2.2 Privacy Key Management (PKM)

In both IEEE 802.16-2004 and IEEE 802.16e-2005 standards, MAC layer contains a security sub-layer. To protect network services from attacks and to guarantee secure distribution of susceptible data from the base station to his subscriber station, WIMAX applies strong support for authentication, key management, encryption and decryption, control and management of plain text protection and security protocol optimization [6]. The most of security issues as described in the following figure:

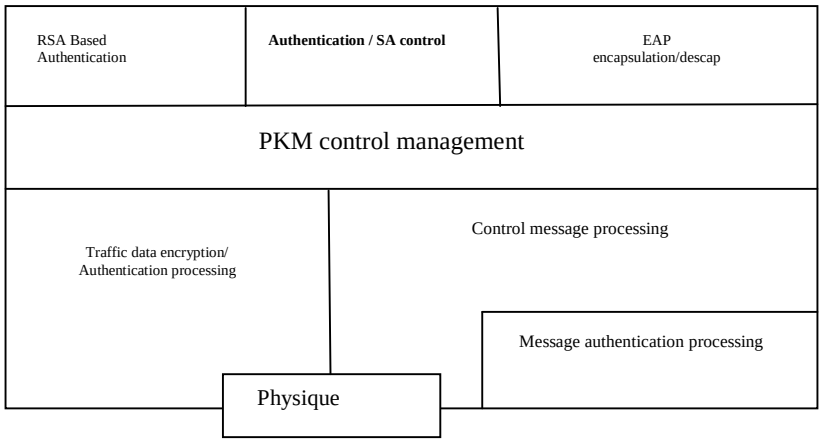

Figure 1: MAC Security Sub-layer

The Base Station and Subscriber Station are protected by the following WIMAX security features:

Security Association (SA): SA is a set of security information parameters that a BS and one or more of its client SSs share [8]. Each SA has its own identifier (SAID), cryptographic identifier, Traffic Encryption Keys (TEKs) and initialization vectors.

Public Key Infrastructure: To authenticate a mobile station to a base station and to secure key management, transfer and exchange between them, WIMAX uses PKM (Privacy and Key Management Protocol). The PKM uses X.509 certificates, RSA public key algorithm and a strong encryption algorithm [8] [9]. It is a three-phase based protocol, as shown in Figure 2. The remaining part of this section describes each of these phases.

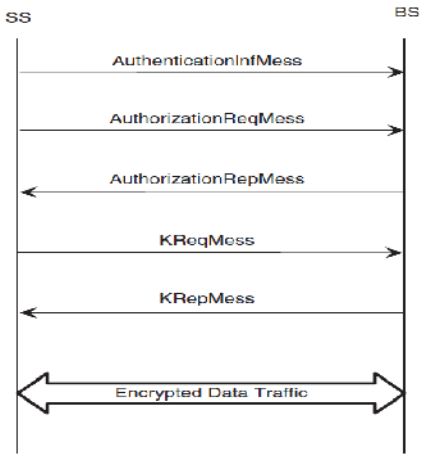

Figure 2: PKM Protocol phases.

2.2.1 Authentication

Authentication is achieved using a public key interchange protocol that ensures not only authentication but also the establishment of encryption keys [11]. WIMAX defines Privacy Key Management (PKM) protocol in security sub-layer, which allows three types of authentication [10]:

The first type is RSA based authentication: RSA based authentication applies X.509 digital certificates together with RSA encryption. In this authentication mode, a BS authenticates the SS through its unique X.509 digital certificate that has been issued by the SS manufacturer. The X.509 certificate contains the SS's Public Key (PK) and its MAC address. When requesting an Authorization Key (AK), the SS sends its digital certificate to the BS, and then BS validates the certificate, uses the verified Public Key (PK) to encrypt an AK and sends back to the SS. All SSs that use RSA authentication have factory installed private/public key pairs together with factory installed X.509 certificates [6] [11].

The second type is EAP (Extensible Authentication Protocol) based authentication: In the case of EAP based authentication, the SS is authenticated either by an X.509 certificate or by a unique operator-issued credential such as a SIM or by user-name/password. There are three types of EAP: the first type is EAP-AKA (Authentication and Key Agreement) for SIM based authentication; the second type is EAP-TLS (Transport Layer Security) for X.509 based authentication; the third type is EAP-TTLS (Tunneled Transport Layer Security) for SS-CHAPv2 (Microsoft-Challenge Handshake Authentication Protocol) [11].

The third type is RSA based authentication followed by EAP authentication [11].

2.2.2 Authorization

This process follows the authentication process. At first SS send an Authorization Request to BS which contains the SS X.509 certificate, encryption algorithms and cryptographic ID. In this message SS requests for an Authorization Key (AK) and a SAID (Security Association ID) from BS. After, the interaction between BS and an AAA server (Authentication, Authorization and Accounting) to validate the request from the SS. BS sends back an Authorization Reply message, in this response BS send an Authorization Key (AK) encrypted with the SS's public key and a lifetime key and a Security Association ID [8].

2.2.3 Encryption

The previous authentication and authorization process results in the assignment of an Authorization Key (AK), which is 160 bits long [6]. The Key Encryption Key (KEK) is derived directly from the AK and it is 128 bits long. The KEK is not used for encrypting traffic data; so SS require the Traffic Encryption Key (TEK) from BS. TEK is generated as a random number generating in the BS using the TEK encryption algorithm where KEK is used as the encryption key. TEK is then used for encrypting the data traffic [7].

3. SECURITY PROPERTY

3.1 Scyther Tool

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 description of a protocol is written in SPDL language. For the protocol verification, Scyther can be used in three ways [12]:

- *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 properties

Ensuring WIMAX protection means that we should satisfy these requirements to protect this network against different attacks.

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 [13]. The formalization of information confidentiality is given below

$$\forall \alpha \in \text{Msg}(\text{claim}(\text{SS}, \text{Secret}, \alpha)) \qquad (1)$$

Property 2- Authenticity

This claim is fulfilled if an outsider, who keeps track of the communication, cannot relate the traffic to a specific MS [2]. 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) [13]. The formal definition of pseudonymity is given below.

$$\text{claim}(SS, Secret, SSCert) \quad (2)$$

***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 [13]:

$$\forall \text{key}(\text{claim}(BS/SS, Secret, \text{key})) \quad (3)$$

***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 [14]. A service should always be bound to an authenticated user. Its formal definition is given as follows:

$$\forall \alpha \in \text{Msg}(\text{claim}(BS, Secret, \alpha)) \quad (4)$$

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.

$$(\text{claim}(BS/SS, Fresh, \text{key})) \quad (5)$$

4. MODELLING AND ANALYZING PKMV2 PROTOCOL

In [6] [11] [10] [15], Authors have described an overview of the various kinds of threats present in WIMAX. In this paper we will focus on PKM vulnerabilities because it is the main part of security, 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 [16].

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. In this paper we model and analyze PKMv2 with RSA based authentication.

The flow of messages exchange in RSA-based authentication is shown as follows:

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

SS/MS sends its MCerMS (manufacturer's certificate) and then sends its own CerMS 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 (MSID) 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_{MS}) of the subscriber station [17].
Authorization Key (AK) transmitted by BS to SS/MS in previous message is used to encrypt the NonceBS (BS generated random number) and MAC_{MS} [17].

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(MS,BS,CA)
{
role MS
{
const Mancert,cap,SAID: Data;
var CerMS,CerBS:Data;
const Ns:Nonce;
var Nb:Nonce;
var SAIDlist,AKSeq,AKlifetime:Data;

send_1(MS,BS,Mancert (MS));
send_2(MS,CA,MS);
read_3(CA,MS,{MS,{CerMS,pk(MS)}sk(CA)}sk(CA));
send_4(MS,BS,{CerMS,pk(MS)}sk(CA));
send_5(MS,BS,{cap,SAID,Ns,MS});
read_8(BS,MS,{CerBS,pk(BS)}sk(CA));
read_9(BS,MS,{{preAK}pk(MS), AKSeq,AKlifetime, SAIDlist,Ns,Nb}sk(BS));
send_10(MS,BS,{Nb,MS}AK);
}
role BS
{
var CerBS,CerMS,Mancert,cap,SAID: Data;
const Nb:Nonce;
var Ns:Nonce;
const SAIDlist,AKSeq,AKlifetime:Data;

read_1(MS,BS,Mancert (MS));
read_4(MS,BS,{CerMS,pk(MS)}sk(CA));
read_5(MS,BS,{cap,SAID,Ns,MS});
send_6(BS,CA,BS);
```

```
read_7(CA,BS,{BS,{CerBS,pk(BS)}sk(CA)}sk(CA));
send_8(BS,MS,{CerBS,pk(BS)}sk(CA));
send_9(BS,MS,{{preAK}pk(MS), AKSeq,AKlifetime, SAIDlist,Ns,Nb}sk(BS));
read_10(MS,BS,{Nb,MS}AK);
}
role CA
{
const CerMS,CerBS: Data;
read_2(MS,CA,MS);
send_3(CA,MS,{MS,{CerMS,pk(MS)}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 MS'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 MS.

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. SECURE KEY MANAGEMENT PROTOCOL (SPKM)

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 our secure 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. MS → CA: MS
msg2. CA → MS:{MS,{CertMS,pk(MS)}sk(CA)}sk(CA)

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

Our protocol (SPKM) can be divided into four main stapes (figure 3):

a. The 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.

b. Certificates Exchange

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

c. 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.

d. 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 MS. That could ensure SS/MS that message 10 is the reply of the request send by SS/MS itself. BS Nonce ensures the MS 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 MS generates AK. After generation of AK correctly, the MS is authorized 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.

e. 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 timestamp sent by BS (Tb). MS use it signature to ensure that message is from an actual MS and to assure the information integrity.

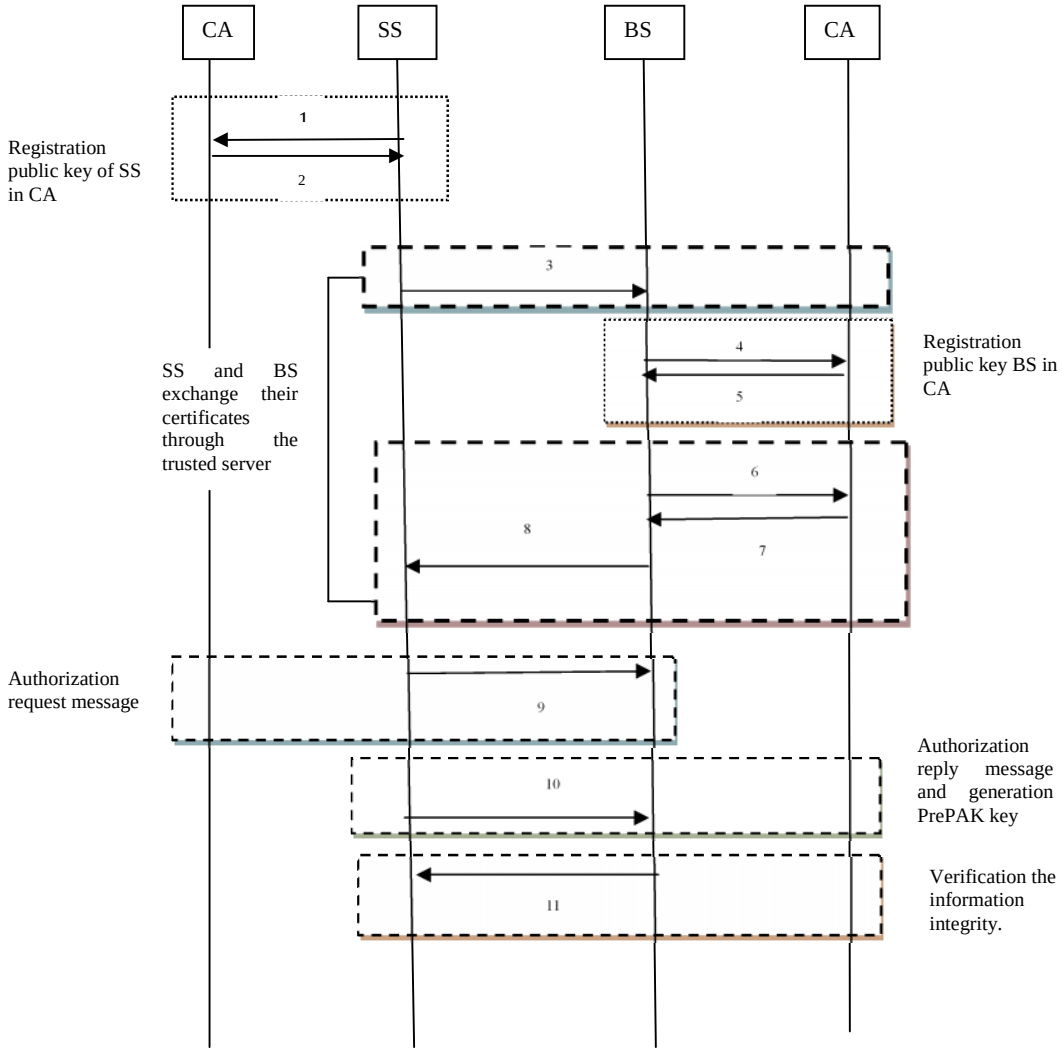

Figure 3: Secure Key Management Protocol (SPKM)

5.1 Modeling New Protocol (SPKM) in SPDL language

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

// The protocol description

protocol new version(MS,BS,CA)
{
role MS

{
const cap,SAID:Data;
var prePAKSeq,prePAKlifetime,CerMS,CerBS, SAIDlist: Data;
var Ns:Nonce;
const Nb:Nonce;
var Ts:TimeStamp;

```
const Tb:TimeStamp;
var prePAK:SessionKey;

send_1(MS,CA,MS);
read_2(CA,MS, {MS,{CerMS,pk(MS)}sk(CA)}sk(CA));
send_3(MS,BS,{{CerMS,Ns}pk(CA)}sk(MS));
read_8(BS,MS, {{CerBS,pk(BS),Ns,Nb}pk(MS)}sk(CA));
send_9(MS,BS,{Ts,Nb,cap,SAID}pk(BS));
read_10(BS,MS,{{prePAK}sk(BS), SAIDlist,Ts,Tb,prePAKSeq,prePAKlifetime}pk(MS));
send_11(MS,BS,{Tb,MS}pk(BS));
claim_MS1(MS, Secret,CerMS);
claim_MS2(MS, Nisynch);
claim_MS3(MS, Niagree);
claim_MS4(MS, Secret,Data);
claim_MS5(MS,Secret,prePAK);
claim_MS8(MS,Secret,Ns);
claim_MS11(MS,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,CerMS,CerBS:Data;
const prePAK:SessionKey;

read_3(MS,BS, {{CerMS,Ns}pk(TS)}sk(MS));
send_4(BS,CA, BS);
read_5(CA,BS, {BS,{CerBS,pk(BS)}sk(CA)}sk(CA));
send_6(BS,CA,{{{{CerMS,Ns}pk(CA)}sk(MS),CerBS,Nb}pk(CA)}sk(BS));
read_7(CA,BS,{{CerMS,pk(MS),Ns,Nb}pk(BS)}sk(CA),{{CerBS,pk(BS),Nb,Ns}pk(MS)}sk(CA));
send_8(BS,MS, {{CerBS,pk(BS),Ns,Nb}pk(MS)}sk(CA));
read_9(MS,BS,{Ts,Nb,cap,SAID}pk(BS));
send_10(BS,MS,{{prePAK}sk(BS),SAIDlist,Ts,Tb,  prePAKSeq,prePAKlifetime}pk(MS));
read_11(MS,BS,{Tb,MS}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 CerMS: Data;
      read_1(MS,CA, MS);
```

send_2(CA,MS, {MS,{CerMS,pk(MS)}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,{{{{CerMS,Ns}pk(CA)}sk(MS),CerBS,Nb}pk(CA)}sk(BS));

send_7(CA,BS,{{CerMS,pk(MS),Ns,Nb}pk(BS)}sk(CA),{{CerBS,pk(BS),Nb,Ns}pk(MS)}sk(CA));
}
}

5.2 Analysis the new version (SPKM)

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 MS 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.

In (SPKM) 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.

IN (SPKM), 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

Security of connections access in WIMAX is complete with respect to the Privacy Key Management (PKM) protocol which provides the authorization process and secure distribution of keying data from the base station to mobile station. In this paper we formally verified the key management protocol of version 2 in terms of the secure session key establishment and distribution, confidentiality, authenticity, integrity, access control.

As discussed in this paper, authentication protocol vulnerable to replay, DoS and Man-in-the-middle attacks. Some solutions are introduced to solve those problems in our secure protocol (SPKM) by using nonce and timestamp together.

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. The timestamp helps the BS in identifying the latest requests, which prevents reply attacks. The nonce value sent by the BS helps in preventing the man-in-the middle attack.

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. Our protocol (SPKM) also provides mutual authentication. 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.

Scyther has been successfully used for the analysis and design of protocols, and has also been used for theoretical research and teaching. Using this tool we prove that our solution is efficient to tackling the various security threats such as replay, man in the middle and DOS attacks.

REFERENCES

[1] Cremers. C (2006). "Scyther-Semantics and verification of security protocols"; PhD dissertation; Eindhoven University of technology.

[2] Ahmed M. Taha, Amr T. Abdel-Hamid, Sofiène Tahar (2009). "Formal Verification of IEEE 802.16 Security Sublayer Using Scyther Tool", ESRGroups France http://hvg.ece.concordia.ca/Publications/Conferences/N2S09.pdf.

[3] IEEE Std. 802.16-2001 (2002). "IEEE Standard for Local and Metropolitan Area Networks Part16: Air Interface for Fixed Broadband Wireless Access Systems", IEEE 2002.

[4] IEEE Std. 802.16-2004 (2004). "IEEE Standard for Local and Metropolitan Area Networks Part16: Air Interface for Fixed Broadband Wireless Access Systems", IEEE 2004.

[5] IEEE Std. 802.16e-2005 (2006). "IEEE Standard for Local and Metropolitan Area Networks Part16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems", IEEE 2006.

[6] Noudjoud Kahya-Abbaci, Nacira Ghoualmi (2012). "A New Classification Based on IEEE 802.16 for Wireless Access"; Journal of Data Processing Volume 2 Number 1 March 2012, pp 32-39. Print ISSN: 2278 – 6481.

[7] Rajesh Srivastava, Deepak Kumar Mehto (2011). "Prevention of Security Threats in IEEE 802.16 Standards", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-4, September 2011.

[8] Rakesh Kumar.J, Upena. D (2010). "A Journey on WIMAX and its Security Issues". International Journal of Computer and Information Technologies. Volume 1 Number 4, pp 256-263. Print ISSN: 0975-9646.

[9] Noudjoud Kahya-Abbaci, Nacira Ghoualmi (2012). "Analysis of Security Weaknesses in IEEE 802.16". Signals and Telecommunication Journal Volume 1 Number 1 March 2012, pp 31-40. Print ISSN: 2278 – 6449, Online ISSN: 2278 – 6457.

[10] Tao Han, Ning Zhang, Kaiming Liu, Bihua Tang, Yuan'an Liu (2008). "Analysis of Mobile WIMAX Security: Vulnerabilities and Solutions". 5th IEEE International Conference on Mobile Ad Hoc and Sensor Networks. Atlanta, USA.

[11] Reena Dadhich, GeetikaNarang, D.M.Yadav (2012). "Analysis and Literature Review of IEEE 802.16e (Mobile WIMAX) Security". International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1, Issue-3, February 2012.

[12] Cremers.C (2008). "The Scyther Tool: Verication, Falsication, and Analysis of Security Protocols?", Department of Computer Science, ETH Zurich, Switzerland Proceedings of the 20th International Conference on Computer Aided Verification (CAV 2008), Princeton, USA.

[13] E. Kaasenbrood (2006). "WIMAX Security - A Formal and Informal Analysis", Master's thesis, Eindhoven University of Technology, Department of Mathematics and Computer Science, Groningen, Netherlands, August 2006.

[14] Lang Wei-min, Wu Runsheng, Wang jian-qiu (2008). "A Simple Key Management Scheme based on WIMAX" . International Symposium on Computer Science and Computational Technology, IEEE 2008.

[15] Deepti, Deepika Khokhar, Satinder Pal Ahuja (2012). "A SURVEY OF ROGUE BASE STATION ATTACKS IN WIMAX /IEEE802.16", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X , Volume 2, Issue 1, January 2012.

[16] D. Johnston and j. Walker (2004). "Overview of IEEE 802.16 security. IEEE Security and Privacy Magazine", vol. 2, no. 3, pp. 40-48, May-June 2004.

[17] A. Altaf, M.Younus Javed, Attiq Ahmed (2008). "Security Enhancements for Privacy and Key Management Protocol in IEEE 802.16e-2005". College of Signals, NUST. Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, IEEE 2008.