

Formal Analysis of Key Management in 802.16e

Noudjoud Kahya¹, Nacira Ghoualmi², Pascal Lafourcade³

^{1,2}Networks and Systems Laboratory (LRS) Badji Mokhtar University, Annaba, Algeria

³LIMOS Laboratory Blaise Pascal University, Aubière, France.

ABSTRACT:

Security is always important in data networks, but it is particularly critical in wireless networks such as WiMAX. 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 and the generation of traffic encryption keys (TEKs) has been analyzed. Possible attacks are also considered including interleaving, replay, DoS, Man-in-the middle attack and a new methodology is presented to prevent these attacks. We also give a formal analysis of our new PKM protocol (authorization phase and exchange of TEKs phase); we conclude that is rigid against 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.

Keywords

IEEE802.16, security, PKM, TEK, analyze formal, scyther tools.

1. Introduction

Wireless networks are less secure due to the lack of physical infrastructure. The 802.16 standard specifies a security sublayer at the bottom of the MAC layer. This security sublayer provides MS (Mobile 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 MS as well as enabling BS to enforce conditional access to network services. Because through the open air interface, MS has no other way to differentiate legitimate BS from malicious adversaries, MS needs to authenticate BS to keep away from malicious ones. Also a certificate sent by MS allows BS to authenticate a legitimate MS. Previous works have addressed the necessity of mutual authentication as well as mechanisms to counter attacks on 802.16. However, there are still some flaws in their protocols. Our paper analyzes those possible attacks to both BS and MS, and proposes a revised authentication protocol to solve those problems. This paper is organized as follows. In Section 2, we provide background and detailed information about Wimax architecture and securities specifications in the

security sub-layer. Section 3, authorization protocol for both versions of PKM has been detailed. In Section 4, we start by performing an evaluation of the security objectives and build a clear attack model for various attacks to PKM. Afterwards, we analyze the protocols against such security objectives informally to check if there are any inconsistencies in our definitions and extract the main holes that exist in both protocols. Then, we apply formal methods on the authentication protocols using the Scyther tool to extract extra holes or threat that might exist. Section 5, covers the proposed solution and modified authentication model; and we conducted a simulation using scyther tools to prove that: authenticity, confidentiality, control access, secrecy and uniqueness of the session keys, and freshly of messages; are protected in our mechanism in both phases of PKM. Finally, Section 6 concludes the paper and describes some future work.

2. WIMAX OVERVIEW

In order to understand Wimax security issues, we first need to understand his architecture and how securities specifications are addressed in this technology.

A. Wimax Architecture

The protocol architecture of Wimax/802.16 is structured into two main layers: 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.

The service specific Convergence Sub-layer (CS) maps higher level data services to MAC layer service flows and connections. There are two type of CS: 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).

The MAC Common Part Sub-layer (MAC CPS) is the core of the standard. This layer defines the rules and mechanisms for system access, bandwidth allocation and connection management. The MAC protocol data units are constructed in this sub-layer.

The Security Sub-layer lies between MAC CPS and PHY layer. This 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.

B. Security Scheme

In WiMAX, security has been included in the design of systems at the very start. To provide secure distribution of sensitive data from the BS to the MS and protect network services from attacks, WiMAX applies strong support for authentication, key management, encryption and decryption, control and management of plain text protection and security protocol optimization. The most of security issues as described in the following figure:

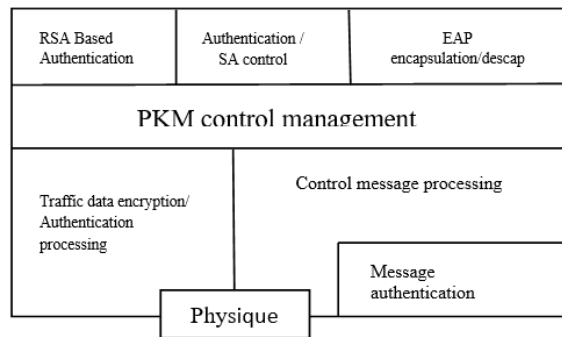


Figure 1: MAC Security Sub-layer

This sub layer basically performs three functions:

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

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 MS through its unique X.509 digital certificate that has been issued by the MS manufacturer. The X.509 certificate contains the MS's Public Key (PK) and its MAC address. When requesting an Authorization Key (AK), the MS 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 MS. All MSs that use RSA authentication have factory installed private/public key pairs together with factory installed X.509 certificates [1].

The second type is EAP (Extensible Authentication Protocol) based authentication: In the case of EAP based authentication, the MS 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) [1].

The third type is RSA based authentication followed by EAP authentication.

2- Authorization: This process follows the authentication process. MS requests for an AK and a SAID (Security Association ID) from BS by sending an Authorization Request message. This message includes the MS X.509 certificate, encryption algorithms and cryptographic ID. In response, the BS interacts with an AAA (Authentication, Authorization and Accounting) server to validate the request from the MS, and sends back an Authorization Reply which includes the AK encrypted with the MS's public key and a lifetime key and an SAID [1] [2].

3- Encryption: The previous authentication and authorization process results in the assignment of an Authorization Key (AK), which is 160 bits long. 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 MS requires 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.

Many attacks are identified on authentication protocols PKM during mutual authentication. The potential attacks that can be carried out are man-in-the-middle, replay, interleaving and DoS attacks.

3. PKM protocol

In WiMAX, Security of connections access in WiMAX is assured with respect to the Privacy Key Management (PKM) protocol. PKM is responsible for the normal and periodical authorization of MSs and distribution of key material to them, as well as reauthorization and key refresh. In PKM MS 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 is a three-phase based protocol, as shown in Figure 2. The remaining part of this section describes each of these phases[3][4].

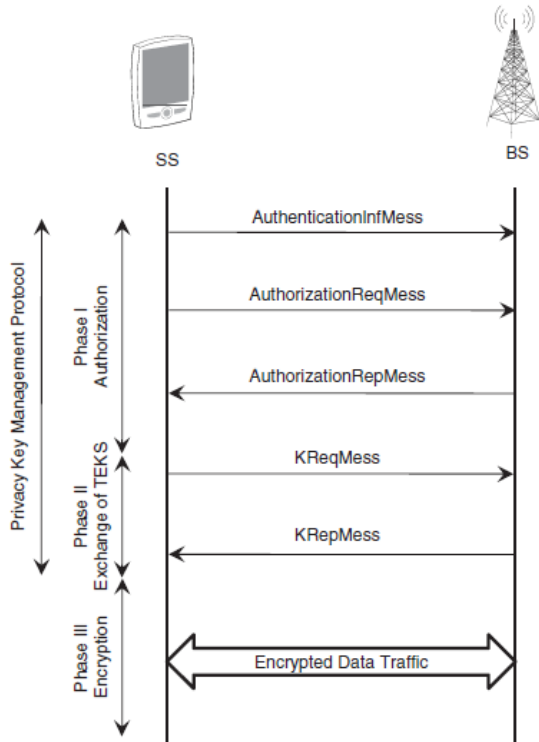


Figure 2 : PKM protocol phases [3]

A. PKM Authorization [3][4]

The first phase of the PKM is the process of authorizing the MS by the BS.

To connect with the BS, the MS sends an authentication message (AuthenticationInfMess) containing the certificate of MS vendor.

Immediately after that, the MS sends an authorization Request Message (AuthorizationReqMess) to the attached BS, requesting an Authorization Key (AK).

This information will be used as a shared secret. The message contains the following information:

- The MS certificate.
- A description of the cryptographic capabilities supported by the MS.
- The security association identifier (SAID) of the MS's primary SA. This value is equal to the primary 16-bit Connection Identifier (CID) that the MS receives from the BS during the network entry and the initialization phase.

The MS will be authorized based on the verification of its certificate. The public key contained in the certificate will be used for constructing the third message. The BS verifies also whether it supports one or more of the cryptographic capabilities of the MS. The response of the BS to the MS is described by message 3 (AuthorizationRepMess). It contains:

- The Authorization Key (AK) generated by the BS and encrypted using the MS public key contained in its

certificate. A proper use of this AK shows an authorization regarding the access of the WiMAX channel.

- A 4-bit AK sequence number to differentiate between the consecutive Authorization Keys.

- The AK life time value.

- The SAIDs descriptor(s) as the identity and properties of the primary SA and zero or more existing static SAs for which the MS may be authorized to get the keying information.

Last message is from BS in reply to MS containing the Authorization Key (AK) encrypted with MS's public key along with sequence number, life time of AK and Security Association Identity List (SAIDlist).

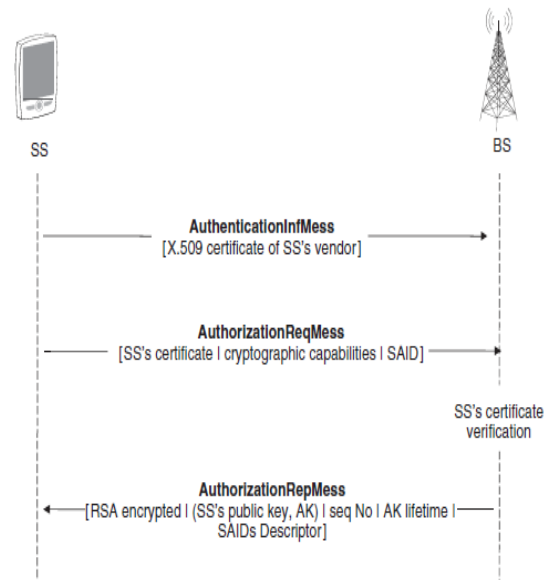


Figure 3 : PKM Authorization [3]

B .Exchange of TEKS [3][4]

The aim of the second phase of the PKM protocol, is to initiate the exchange of TEKSs, and establish a data SA. The TEKS will be later used for encryption. As stated previously, the authorizationRepMess message contains, in addition to the SAID and properties of the SA, from zero to several static SAs for which the MS is authorized to obtain the key material. Therefore, the MS starts, in this phase, a separate state machine for each of the SAID identified in the authorizationRepMess message.

Every state machine is responsible for managing the keying material associated with the related SAIDs.

Every MS sends periodically a Key Request Message (KReqMess) to the BS, asking it for the renewal of the TEK. This message is composed of:

- the AK sequence number which allows the BS to determine the Uplink HMAC Key used by the SS to generate the HMAC digest of this message;

- the SAID related to the SA whose keying material is requested. This SAID is related to the started TEK state machine;
- the HMAC digest produced by the application of the HMAC function on the message payload using the Uplink HMAC Key.

After making sure that the received SAID matches the SA at the MS and verifying the authenticity and the integrity of the KReqMess message by checking the HMAC digest, the BS responds to that message. It sends a key Reply Message (KRepMess) containing the new key material needed by the TEK state machine. At any time, the BS maintains two active key materials per SAID, which are denoted by TEK-Parameters in the KRepMess. A keying material includes:

- TEK encrypted with the KEKs using either the 3DES in EDE mode with 128 bits, RSA PKCS#1, or AES in ECB mode with 128 bits;
- the remaining lifetime of the TEK;
- the TEK sequence number;
- a 64-bit initialization vector.

The KRepMess message contains an AK sequence number, the SAID, the parameters related to the old TEK and the new TEK and an HMAC digest to ensure the MS that the message is sent by the BS without being tampered with. Note that the validity durations of the two TEKs overlap. In fact, the new TEK is being activated before the old TEK expires and the old TEK is destroyed after the activation of the new TEK. The lifetime of a TEK is also used by the MS to estimate when the BS will invalidate a previous TEK or request a new TEK.

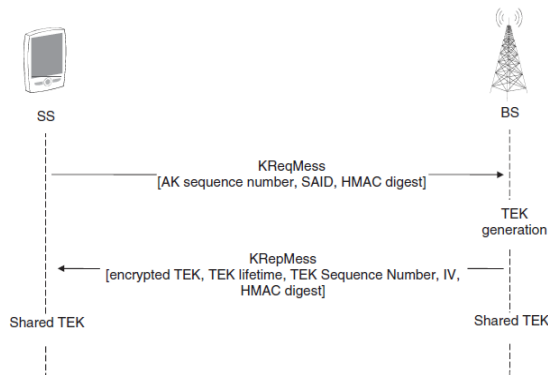


Figure 4 : Privacy and Key Management phase [3]

C. Data Encryption [3][4]

After achieving the SA authorization and the TEK exchange, transmitted data between the MS and BS starts to be encrypted using the TEK. An encryption algorithm is used to encipher the MAC PDU. Note that, neither the CRC nor the MAC header is involved in encryption in order to guarantee the forwarding of the MAC PDU and

support diverse services. In the MAC header, an Encryption Control (EC) field is set to 1 as an indication regarding the availability of an encrypted MPDS. In addition, the 2-bits Encryption Key Sequence (EKS) field indicates the used TEK. Encryption can be done by means of the Data Encryption Standard (DES) using Cipher Block Chaining (CBC) mode with 56 bits.

4. Formal Analysis of Pkm Using Scyther Tool

There are numerous robust tools available for formal security protocol analysis such as OFMC [5], Scyther [6], and ProVerif [7].

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 she 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.

This section describes the main security weakness related to the PKM standard, showing potential attacks in authorization phase and exchange of TEKs phase.

Similar to the approach taken by our analysis of PKM v1/v2[8][9], we contains the reserche and we formally verify our analysis on different phases of PKM protocols using scyther. In the end of this section we describe the proposed protocol and we discuss the obtained results.

A. Properties Specifications

Authenticity, confidentiality, access control, secrecy and uniqueness of the keys and freshly of message are selected for formal verification.

1) Authenticity: The principals (MS/BS) verify the authenticity of received messages (by verifying signatures or MACs). 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 (SSCert). The formal definition of authenticity is given below.

Property 1: *claim(SS, Secret, SSCert)*

2) Confidentiality: Expresses that certain information is not revealed to an intruder. The security satisfied if the MS has the guarantee that all exchanged user data to BS is secret. The formalization of information confidentiality is given below.

Property 2: $\forall \alpha \in \text{Msg}(\text{claim}(\text{SS}, \text{Secret}, \alpha))$

3) *Access control*: This claim is fulfilled if the BS has the guarantee that, neither an unauthenticated user should gain access to the services provided, nor should an unauthenticated user be able to impersonate another user. A service should always be bound to an authenticated user. Its formal definition is given as follows:

Property 3: $\forall \alpha \in \text{Msg}(\text{claim}(\text{BS}, \text{Secret}, \alpha))$

4) *Secrecy and uniqueness of the session keys*: This claim is fulfilled if the BS and the MS have the guarantee that all exchanged keys (AK and TEK) are secret and unique.

Property 4: $\forall \text{key}(\text{claim}(\text{BS}|\text{SS}, \text{Secret}, \text{key}))$

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 have the guarantee that the session key is fresh.

Property 5: $(\text{claim}(\text{BS}|\text{SS}, \text{Fresh}, \text{key}))$

B. Formal Verification

Pseudonymity, information confidentiality, no theft of service and secrecy and uniqueness of the session keys are selected for formal verification, we apply Scyther tool to verify if this properties are proved or not in PKM protocol. Our analysis reveals that the phases of Key Management Protocol PKM are vulnerable into many attacks; these attacks fall into the following categories: replay, DoS, Man-in-the middle attacks. In table 1 we list the violated properties.

Phases	Violated property	Respected property
Phase 1 ; PKM authorization	MS Agreement BS Agreement MS Synchronization BS Synchronization MS Secret Data MS Secret CertMS	MS/BS Secret AK
Phase2 : génération of TEK	MS Agreement BS Agreement MS Synchronization BS Synchronization MS Secret Data MS Secret CertMS	MS/BS Secret TEK

Formal analysis of the revised authentication protocol

- Property 1:** Scyther detected a possible attack, as an intruder eavesdrops the second message and obtains the MS's certificate (MsCert).
- 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 previously.

3. **Property 3:** It is proved that unauthenticated user cannot access the services provided, and cannot impersonate another user. The BS uses the certificate of the MS to determine if the MS is authorized, then sends the AK encrypted with the public key of the MS. This guarantees that only the specific MS can decrypt the AK.

4. **Property 4 and 5:** It is proved that an adversary cannot obtain the unique AK as it is encrypted with the public key of the MS. AK is proved to be unique using synchronization claim and the fact that AK is a constant in one of the roles appearing only in one send event.

After the MS authentication procedure has been done, the AK is used to derive KEK and HMACkey. TEK is then generated by BS randomly. The TEK is the key actually used to encrypt data traffic exchanged between the BS and MS. A key exchange message is authenticated by HMAC-SHA1 to provide message integrity and AK confirmation. It is proved that an adversary cannot obtain the unique TEK.

Similar to the authorization protocol, the exchange of TEKs phase of the PKM is vulnerable to the replay attack. If an attacker replays the first message, the BS will assign and send new keying material using a KRepMess message. The legitimate MS, which is not aware of the attack, will think that it is the BS which requested the rekeying and sent the first

optional message. As a consequence, this attack causes both the MS and BS to exchange keying material without intending to.

As seen in the formal analysis, the secrecy and uniqueness of the keying material distributed and the no theft of service possible claims are valid in both phases of PKM. However, pseudonymity and information confidentiality are broken.

Scyther results : verify

Claim	Status	Comments	Patterns
pkmv2n SS pkmv2n,rsa3 Niagree	Fail	Falsified At least 1 attack.	1 attack
pkmv2n,rsa4 Nisynch	Fail	Falsified At least 1 attack.	1 attack
pkmv2n,rsa5 Secret prepak	Ok	No attacks within bounds.	
pkmv2n,rsa6 Secret certSS	Fail	Falsified At least 1 attack.	1 attack
pkmv2n,rsa7 Secret Data	Ok	No attacks within bounds.	
pkmv2n,rsa8 SKR tek	Ok	No attacks within bounds.	
BS pkmv2n,rsa3 Niagree	Fail	Falsified At least 1 attack.	1 attack
pkmv2n,rsa4 Nisynch	Fail	Falsified At least 1 attack.	1 attack
pkmv2n,rsa5 Secret prepak	Ok	No attacks within bounds.	
pkmv2n,rsa6 Secret certBS	Fail	Falsified At least 1 attack.	1 attack
pkmv2n,rsa7 Secret Data	Ok	No attacks within bounds.	
pkmv2n,rsa8 SKR tek	Ok	No attacks within bounds.	

Done.

5. The Proposed Revised Authentication Protocol

As discussed in the previous section, the existing protocol does not fulfill the claims pseudonymity and information confidentiality because it still vulnerable to replay, DoS and Man-in-the-middle. Some solutions are introduced to solve those problems in our new revised protocol. To prevent replay and man-in-the-middle attacks we add timestamp. The problem with timestamp is that it requires time synchronization between MS and BS. In the wireless scenario, time synchronization is considered to be difficult (particularly under mobility). But In IEEE 802.16(e), it is assumed that time synchronization is done between MS and BS.

Nonce is a possible alternative to timestamps for use in the authentication protocols. Nonce shows that the request queued were not used before. Timestamp identifies which request are the newer one and also the time sent by the MS and BS. Nonce 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 MS. There are two problems with the protocol that has timestamps only. An adversary can easily capture the timestamp of MS by listening to message 2. The time adjustment can be done by the adversary accordingly. Hence the scope of man in middle attack is persists with timestamp added protocol. To prevent security threats like replay attacks, DoS attack and Man-in-the-middle attack, both nonce and time stamp are needed. So the revised protocol has the timestamp attached with the MS message to the BS along with the nonce.

The protocol is shown as follows:

MS sends a message to BS, which contains an X.509 certificate identifying MS's manufacturer. BS is using this message in order to decide if the particular MS is a trusted device or not. MS sends a second message without waiting for an answer from the BS. This second message contains the MS certificate (MsCert) and a nonce (Ns1) used for identification, both are encrypted with the public key of the BS $pk(Bs)$, it also contains the timestamp of MS and generated nonce of MS along with SAID and its security capabilities. MS signs the message ensuring the BS that he/she is not an adversary with her private key $sk(MS)$, the time stamp addition could bring an extra layer of security since the BS could identify the message as current one. The time stamp could avoid the intruders who are trying to synchronize time with either BS or MS. If BS determines that the MS is authorized it replies with a message. BS sends a generated nonce along with nonce which was sent by the MS. That could ensure MS that message3 is the reply of the request send by 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 secret key

of BS $sk(Bs)$, The AK is derived from Pre-AK. Use of Pre-AK gives the opportunity to avoid AK sending in raw format (though encrypted with the public key). From pre-PAK, the MS generates AK. If AK is used correctly, then MS gains the authorization to access the WMAN channel. The Lifetime of Pre-AK and Sequence no of pre-AK are sent in message3. This protocol using the public key of MS in message3 ensures MS that the message received is from a legitimate BS. As this message sends the BS certificate, the MS is now sure that the message is not copied by the adversaries. MS sends its Timestamp and the nonce of BS previously received to confirm authorization access. MS signs the message with its private key. Similar to the authorization phase, we used the timestamp attached with the MS message to the BS along with the nonce in all messages of key management phase of the PKM.

Formal analysis of the revised authentication protocol

The formal definition of the revised PKM is shown as follows:

1- Authorization phase :

$MS \rightarrow BS: Mancert(MS);$
 $MS \rightarrow BS: \{\{MMSCert, Ns1\}pk(Bs), Capabilities, SAID, Tms, Ns\}sk(MS);$
 $BS \rightarrow MS: \{\{prePAK\}sk(Bs), SAIDlist, Tms, Tbs, Ns, Nb, prePAKSeq, prePAKlifetime, BsCert\}pk(MS);$
 $MS \rightarrow BS: \{Nb, Tms\}sk(MS);$

2- Exchange of TEKs phase:

$RkeyMess (BS \rightarrow MS): TBS | NBS | SeqNo | SAID | HMAC(RkeyMess).$

$KReqMess (MS \rightarrow BS): TBS | TMS | NBS | NMS | SeqNo | SAID | HMAC(KReqMess)$

$KRepMess (BS \rightarrow MS): TMS | NBS | SeqNo | SAID | OldTEK | NewTEK | HMAC(KRepMess)$

Formal analysis of the revised authentication protocol

In this section, we formally verify our analysis on all phases of PKM protocols, and the correctness of our reversion. The revised authentication protocol is going to be challenged with the following requirements using the Scyther tool.

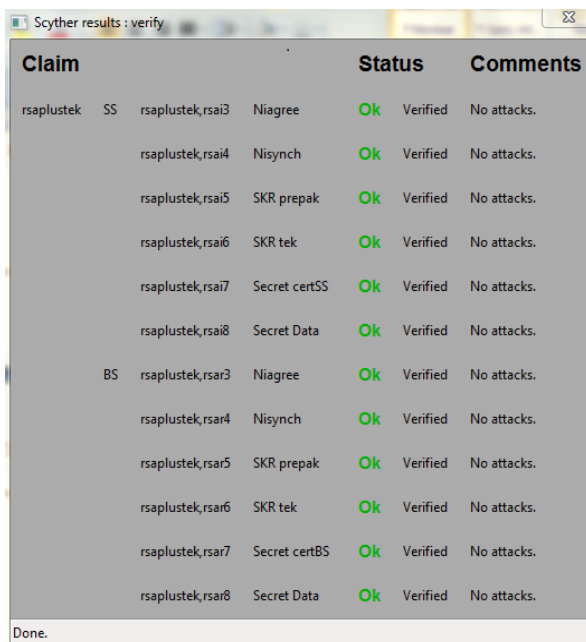
1. *Property 1:* In the formal analysis it is proved that an intruder cannot obtain the MS certificate (MsCert).
2. *Property 2:* In the formal analysis it is proved that the authorization key exchanged in the authentication protocol is secret and not broken.
3. *Property 3:* It is proved that unauthenticated 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 4 and 5:* It is proved that an adversary cannot obtain the unique pre-PAK and the TEK is secured Timestamp and nonce are used in the revised protocol to

prevent replay and man-in-the-middle attack. The 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. 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 time stamp and nonce of 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 timestamp helps the BS in identifying the latest requests, which prevents replay attacks. It also helps the MS to identify the recent messages, and hence it can identify the AK used by the MS as new or not. The addition of nonce from the BS helps the 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 MS.

Similar to the authorization phase, the nonce and timestamp helps the MS and BS to prevents replay attacks in the exchange of TEKs phase.



Claim	Status	Comments
rsaplustek SS rsaplustek,rsai3 Niagree	Ok Verified	No attacks.
rsaplustek,rsai4 Nisynch	Ok Verified	No attacks.
rsaplustek,rsai5 SKR prepak	Ok Verified	No attacks.
rsaplustek,rsai6 SKR tek	Ok Verified	No attacks.
rsaplustek,rsai7 Secret certSS	Ok Verified	No attacks.
rsaplustek,rsai8 Secret Data	Ok Verified	No attacks.
BS rsaplustek,rsar3 Niagree	Ok Verified	No attacks.
rsaplustek,rsar4 Nisynch	Ok Verified	No attacks.
rsaplustek,rsar5 SKR prepak	Ok Verified	No attacks.
rsaplustek,rsar6 SKR tek	Ok Verified	No attacks.
rsaplustek,rsar7 Secret certBS	Ok Verified	No attacks.
rsaplustek,rsar8 Secret Data	Ok Verified	No attacks.

Done.

6. Conclusion

The paper analyzes the vulnerabilities in the basic phases of PKM. As seen in the formal analysis, we formally verified the PKM 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 PKM.

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 in authorization phase and replay attack in generation of TEKs phase..

The revised authentication protocol is expected to provide better secure platform for all process of PKM.

REFERENCES

- [1] Sen Xu, Chin-Tser. "Attacks on PKM Protocols of IEEE 802.16 and Its Later Versions" 200 IEEE Huang Computer Science and Engineering Department University of South Carolina Columbia, SC 29208, USA
- [2] Ayesha Altaf, M.Younus Javed, Attiq Ahmed. "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.
- [3] Seok-Yee Tang, Peter Muller, Hamid R. Sharif "WiMAX SECURITY AND QUALITY OF SERVICE AN END-TO-END PERSPECTIVE" ISBN 978-0-470-72197-1 (H/B) .A John Wiley and Sons, Ltd., Publication, 2010.
- [4] Ramjee Prasad I Fernando J. Velez "WiMAX Networks, Techno-Economic Vision and Challenges" ISBN 978-90-481-8751-5 e-ISBN 978-90-481-8752-2 DOI 10.1007/978-90-481-8752-2 Springer Dordrecht Heidelberg London New York, 2010.
- [5] D. B. Sebastian M'odersheim Luca Vigano. Ofmc: A symbolic model checker for security protocols. International Journal of Information Security, 4(3):181–208, June 2005. Published online December 2004.
- [6] C. Cremers. The Scyther Tool: Verification, falsification, and analysis of security protocols. In Proc. CAV, volume 5123 of LNCS, pages 414–418. Springer, 2008.
- [7] B. Blanchet. An efficient cryptographic protocol verifier based on Prolog rules. In Proc. 14th IEEE Computer Security Foundations Workshop (CSFW), pages 82–96. IEEE, 2001.
- [8] N.Kahya, N. Ghoulmi, P.Lafourcade « Secure key management protocol in wimax. » International journal of network security and its application volume 4, number6, November 2012. ISSN 0974-9330
- [9] N.Kahya, N. Ghoulmi, P.Lafourcade « Key management protocol in wimax revisited » Advances in computer science, engineering and application, Springer 2012 ISSN1867-5662.