Attacks on PKM Protocols of IEEE 802.16 and Its Later Versions

Sen Xu, Chin-Tser Huang
Computer Science and Engineering Department
University of South Carolina
Columbia, SC 29208, USA
{xu4, huangct}@cse.sc.edu

Abstract—Without physical boundaries, a wireless network faces many more vulnerabilities than a wired network does. IEEE802.16 provides a security sublayer in the MAC layer to address the privacy issues across the fixed BWA (Broadband Wireless Access). Several articles have been published to address the flaws in IEEE802.16 security after IEEE802.16-2001 was released. However, even the enhanced version IEEE802.16-2004 does not settle all the problems and additional flaws emerge. In addition, we found that PKM (Privacy and Key Management) protocols version 2 (PKMv2), proposed by recently released IEEE802.16e, is also vulnerable to new attacks. In this paper, we first overview the IEEE802.16 standard, especially the security sublayer, and then investigate possible attacks on the basic PKM protocol in IEEE802.16 as well as in its other versions from related works and the newest PKMv2. We also give possible solutions to counter those attacks and verify our analysis using formal (BAN) logic.

Keywords-PKM; authentication protocol; IEEE802.16

I. INTRODUCTION

As a member of IEEE 802 group, IEEE 802.16 is the standard to specify the air interface of fixed BWA. IEEE standard 802.16 [1] 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 [2], which consolidates previous standards, also supports non-line-of-sight (NLOS) within 2-11GHz bands and mesh nodes. The recently released amendment, IEEE 802.16e, aims to provide mobility in WMAN.

In a WMAN, both the Base Station (BS) and Subscriber Station (SS) face almost all those attacks as their wired counterparts do. Moreover, wireless networks are inherently 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 SS with privacy and protects BS from service hijacking. There are two component protocols in the security sublayer: an encapsulation protocol for encrypting packet data across the fixed BWA (Broadband Wireless Access), 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 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).

A certificate sent by SS allows BS to authenticate a legitimate SS. On the other hand, SS also needs to authenticate BS to keep away from malicious ones. That is because through the open air interface, SS has no other way to differentiate legitimate BS from malicious adversaries. 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. This paper analyzes those possible attacks to both BS and SS, and proposes a revised authentication protocol to solve those problems.

This paper is organized as follows. In Section 2, we introduce related works. Section 3 brings up various attacks on the authentication protocols of basic PKM in 802.16 as well as of its later versions; our modified protocol is also proposed. In Section 4, we apply BAN logic to analyze those protocols and to verify our modified protocol. Finally, Section 5 concludes the paper and describes some future work.

II. RELATED WORKS

Since the first version of IEEE 802.16, a few papers have been published to introduce this new standard. In [3], Roger Marks gives a technical overview of 802.16. There are also some other papers and books that review this standard, such as [4] and [5]. However, few of them tackle the security issues. It is clear that so far WMAN has been less investigated than WLAN. With its great potential in the future's wireless service, WMAN deserves more attention than what it gets now.

The authors of [6] review the 802.16 standard, and analyze its security in many aspects, such as vulnerability in authentication and key management protocols, failure in data encryption, and lack of explicit definition. Mutual authentication is the major contribution to PKM protocols proposed by [6], which enables SS to authenticate BS as well.

In fact, the need for mutual authentication in wireless network is not a novel topic. It has been widely studied in the scope of WLAN. In WLAN, WS needs to authenticate AP while AP authenticates WS. However, the authentication and key management protocols in 802.11 and 802.16 are based on different methods. IEEE 802.11 applies the shared-key authentication method, while IEEE 802.16 is based on public-key authentication algorithm, specifically, X.509 certificate. Therefore, the authentication and key management in IEEE 802.16 needs separate study.

In our previous paper [7], we have analyzed security issues on the basic PKM protocols and proposed some solutions. Recently, PKM version 2 (PKMv2) is proposed in 802.16e standard, which will be publicly available in August 2006. References [8] and [9] are comments for this new amendment. We do not find any other analysis about this protocol till now. However, there are many works on protocols based on X.509, such as [10], [11], [12] and [13]. Some discussions about the Authentication Protocols in earlier versions of PKM in [7] are also included in this paper, both because they are related to our later discussions in this paper and because we want to use BAN logic to analyze them formally. The Key Management Protocol is not included though, because it has not changed since our last discussion.

III. Analysis and Modification of PKM Authentication Protocols

A. Authentication Protocol in IEEE 802.16

An SS begins authorization by sending an Authentication Information message which contains the SS manufacturer's X.509 certificate. This message is largely informative and the BS may choose to ignore it. Afterwards the SS sends an Authorization Request message (Auth-REQ) to its BS. In response to Auth-REQ, the BS validates the requesting SS's identity, determines the encryption algorithms and protocols to be shared with the SS, generates an Authentication Key (AK), and sends the AK to SS. The authentication protocol is illustrated in Fig. 1.

Message 1. SS → BS : Cert (SS. Manufacturer)
Message 2. SS → BS : Cert (SS) | Capabilities | BCID
Message 3. BS → SS : KU_{ss} (AK) | SeqNo | Lifetime | SAIDList

Figure 1. Authentication Protocol Scenario in 802.16

In Fig. 1, Cert (SS.Manufacturer) is the X.509 certificate of SS's manufacturer, and Cert (SS) is SS's X.509 certificate. BCID is the Basic CID of SS, which equals to its primary SAID. KU_{SS} (AK) is the Authentication Key encrypted by SS' public key. SeqNo is a 4-bit sequence number for AK. And lifetime gives the number of seconds before AK expires. SAIDList contains the identities and the properties of the SAs for which SS is authorized to obtain keying information.

B. Attacks on Basic PKM Authenctication Protocol

BS will face replay attack from malicious SS, who intercepts and saves the messages sent by a legal SS previously. We name this attack Simple Replay Attack. We call it simple because, unlike other replay attacks which usually require more tricks to succeed, Simple Replay Attack only involves falsifying instances of the request message. In [11], when analyzing Kerberos Protocol, the authors claim it is common for designers not to focus on such kind of attacks. They regard it as vulnerability but not serious flaw. However, we find it is not the same situation for PKM Protocols in IEEE802.16, in which it may lead to a severe result. The reason is that, if BS set a timeout value which makes itself to reject Auth-REQ from the same SS in a certain period, the legal request from the victim SS will also be ignored. Therefore, the Deny of Service occurs to the victim SS. Otherwise, if BS accept the request, it will have to generate new AK for SS, which usually involved nonce information. This will exhaust BS' capabilities. To avoid these replay attacks, we suggest adding timestamps in message 2, together with a signature by SS.

Similarly, message 3 also endangers SS in replay attacks. Even worse, malicious BS can make its own Auth-Reply message with the AK generated by itself, thus gaining the control of the communication of the victim SS. This is a typical Man-in-the-Middle attack, which brings forward the need of mutual authentication, i.e., SS needs to authenticate BS as well. This can be done by adding BS' certificate in message 3. The revised protocol with the proposed modifications is shown in Fig. 2.

Message 1. SS → BS : Cert (SS. Manufacturer)
Message 2. SS → BS : T_S | Cert (SS) | Capabilities | SAID | SIG_{SS} (2)
Message 3. BS → SS : T_S | T_B | KU_{SS} (AK) | Lifetime | SeqNo | SAIDList | Cert (BS) | SIG_{BS} (3)

Figure 2. Revised Authentication Protocol

In Fig. 2, T_S and T_B are timestamps generated by SS and BS respectively; SIG_{SS} (2) is the signature of SS over message 2; SIG_{BS} (3) is the signature of BS over message 3.

C. Attacks on Intel Nonce Version PKM

Nonce is a possible alternative to timestamp in the authentication protocol. In [6], the authors use nonce instead of timestamp. Their protocol is shown in Fig. 3.

Message 1. SS → BS : Cert (SS. Manufacturer)
Message 2. SS → BS : N_S | Cert (SS) | Capabilities | SAID
Message 3. BS → SS : N_S | N_B | KU_{SS} (pre-AK) | Lifetime | SeqNo | SAIDList | Cert (BS) | SIG_{BS} (3)

Figure 3. Authentication Protocol with nonce in [6]

However, the exchange of nonces only assures SS that message 3 is a reply corresponding to its request. The BS still faces the replay attack because BS cannot tell whether message 2 is sent recently or it is just a replayed message.

D. Attacks on PKMv2

IEEE 802.16e proposes PKMv2, in which one additional message is added at the end of the original protocol, shown as Fig. 4. SSID is SS's identifier from Cert (SS); AAID is the ID of Authorized Association (AA); SSAddr is the MAC address of SS.

Message 1. SS → BS : Cert (SS. Manufacturer)

Message 2. SS → BS : N_S | Cert (SS) | Capabilities | BCID

Message 3. BS → SS : N_S | N_B | KU_{SS} (pre-AK, SSID) | Lifetime | SeqNo | SAIDList | AAID | Cert (BS) | SIG_{BS} (3)

Message 4. SS → BS : N_B | SSAddr | AK (N_B | SSAddr)

Figure 4. Authentication Protocol for PKMv2

In fact, there are three optional protocols for X.509 certificate: one-way, two-way, and three-way authentication. Although the original IEEE 802.16 authentication protocol involves two messages[1], it is still a one-way authentication, because it only provides SS' certificate to BS. Our modified version and Intel Nonce version can be regarded as two-way authentication, which provide mutual authentication between communication parties. The PKMv2 belongs to the three-way authentication, with a confirmation message from SS to BS.

In X.509 certification, both timestamp and nonce are used. That is because the timestamp in X.509 is not used as a kind of nonce but as a lifetime, which includes start time (optional) and end time to prevent delayed sending. Therefore a nonce, which is unique during the lifetime, is also used to prevent replay attack. We have already shown that timestamp is critical for the one-way and two-way authentication protocols in previous subsections. In order to prevent falsifying and replay, the signature by SS is also necessary, which is included in the X.509 as well as in our modified 802.16 authentication protocol.

In three-way authentication, the Nonce N_B is included in the last message from SS to BS. It seems not necessary to check the timestamp any more, because the nonces from both parties are sent back to each other, thus both parties can check the replied nonce to prevent replay attack. This method is usually applied when there is no synchronized clock. Details discussion can be found in [14] and related IETF drafts.

Due to the reason above, the original description of the protocol [15] claimed that BS does not need to check the timestamp T_S. However, several defects have been found by many researchers' work, such as [10], [12], and [13], in which it is shown that an intruder can replay the request message to BS and use corresponding SS as an oracle to answer a nonce challenge from BS. Thus the protocol can still be in danger of attacks if it is not designed properly. For PKMv2, several possible attacks are illustrated as follows.

First, without signature by SS, the request message is easy to be modified or impersonated. This is similar to what we discussed before and we refer to it Simple Replay Attack.

Second, even with the signature from SS served as message authentication, attack still exists. This attack is the similar as the one proposed in [10], which is classified as Interleaving Attack in [16]. We elaborate it here for PKMv2. We assume the request message is already signed by SS. In fact, the signature will not help much for those nonce versions. We also omit the informative message 1 from original protocol and omit non-critical parts in those messages for conciseness. Following figure shows the scenario of this attack.

α.1. IR (SS) → BS : N_S | Cert (SS) | SIG_{SS}(α.1)

α.2. BS →IR (SS) : N_S | N_B | KU_{SS} (pre-AK, SSID) | Cert (BS) | SIG_{BS} (α.2)

β.1. SS → IR : N_S' | Cert (SS) | SIG_{SS}(β.1)

β.2. IR → SS : N_S' | N_B | KU_{SS} (pre-AK, SSID) | Cert (I) | SIG_I (β.2)

β.3. SS → IR : N_B | SSAddr | AK (N_B | SSAddr)

α.3. IR (SS) → BS : N_B | SSAddr | AK (N_B | SSAddr)

Figure 5. Interleaving Attack on PKMv2

In Fig. 5, α.1 means message 1 in a protocol instance run α. IR (SS) represents an intruder IR impersonating as SS. In run α, IR impersonates as SS and sends message 1 to BS, which is a replayed one sent by SS before. When IR receives α.2 from BS, it needs to reply with α.3 to succeed in this authentication. But IR is not able currently, because it can not decrypt the message encrypted by SS's public key so as to get AK to encrypt the nonce challenge. However, IR can use SS as an oracle to answer this challenge. IR may force SS to run another protocol instance β with it, and replies to SS with the same nonce challenge which BS sent to it. SS will sends β.3 to IR, by which IR can send to BS and finish α.

For IR to succeed in this attack, there are two problems left. First, the AK in PKMv2 is derived from pre-AK with BSAddr and SSAddr. In order that the AKs in α and β are the same, IR should also impersonate BSAddr. This is quite easy in wireless network. Second, PKMv2 uses AA to bind a security session. This also can be forged or replayed by IR to SS.

Similarly as in related works, this attack can be avoided by adding BS's identity (BSID) in message 3, encrypted by AK. From this view, this attack can also be regarded as Attack Due to Name Omission. Moreover, SSAddr is not necessarily encrypted at all. If the message 3 is not modified, i.e. it is encrypted by AK derived from SSAddr, BS should also derived the same AK in order to decrypt it, thus is ensured that SSAddr is not modified. Otherwise, some IR must forge his own AK derived from modified SSAddr, and encrypted the modified SSAddr by that AK. Thus encrypting SSAddr will not give any more security. The designer may want SSAddr to act as a salt, which BSID could do if added. Also, SSID in

[1] The first message is informative and can be ignored, thus we only count the second and third messages.

message 2 is not necessary. The encryption by SS's public key already guarantees that the message is for SS only.

A new attack on original X.509 3-way authentication protocol is found by [17], even if it checks the timestamp. The author calls it Multiplicity Attack, where one agent is mistaken about the multiplicity of sessions. This attack can be eliminated by adding BS's identity also.

Through the discussion above, we can conclude that certificate from BS and signature by SS is critical to all versions of those authentication protocols. Timestamp is also necessary to Basic PKM and Intel version, but it can be omitted in PKMv2 only after the PKMv2 is modified. However, the Simple Replay Attack is still feasible to the modified PKMv2. Comparing with our modified version with timestamp, the modified PKMv2 does not involve timestamp, thus the message may be a little more concise, and will not depend much on synchronized clock. While our modified version only involves two messages compared to 3 messages in PKMv2. Less number of exchanged messages is always essential to authentication protocols. Besides, synchronization is not a problem for applying timestamp here due to the inherence of IEEE802.16, because SS and BS have already synchronized during initial ranging. Moreover, our modified protocol is resistant to Simple Replay Attack.

IV. Formal Analysis using BAN Logic

In this section, we formally verify our analysis on different versions of PKM protocols, and the correctness of our revised version, using BAN logic. Due to space limitation, we can not illustrate the meaning of symbols and postulates in BAN logic in this section. Interested readers can check [10] for details.

The goals of the authentication protocol should be:

$$SS \mid\equiv SS \xleftrightarrow{AK} BS, \qquad BS \mid\equiv SS \xleftrightarrow{AK} BS,$$

$$BS \mid\equiv SS \mid\equiv SS.REQ$$

SS.REQ is implicitly indicated in the request message, which means SS does send the authentication request.

The idealized protocol of Basic PKM Authentication is shown as follows, where A is the authorization server assigning the certificates for SS and BS:

$$SS \rightarrow BS : SS.REQ,\ \{T_{ASS}, \mid \xrightarrow{K_{SS}} SS\}_{K_A^{-1}} \quad (1)$$

$$BS \rightarrow SS : \{SS \xleftrightarrow{AK} BS\}_{K_{SS}} \quad (2)$$

Note that SS.REQ is usually not included in the idealized protocol because the cleartext communication provides no guarantees of any kind. We add it here because it is related to one of the goals we want to achieve after the run of authentication protocol.

The assumptions are listed below:

$$SS \mid\equiv \mid \xrightarrow{K_A} A \quad (3),\ BS \mid\equiv \mid \xrightarrow{K_A} A \quad (4)$$

$$SS \mid\equiv \mid \xrightarrow{K_{SS}} SS \quad (5),\ BS \mid\equiv \mid \xrightarrow{K_{BS}} BS \quad (6)$$

$$SS \mid\equiv A \mid\Rightarrow \xrightarrow{K} BS \quad (7)$$

$$BS \mid\equiv A \mid\Rightarrow \xrightarrow{K} SS \quad (8)$$

$$SS \mid\equiv BS \mid\Rightarrow (SS \xleftrightarrow{K} BS) \quad (9)$$

$$BS \mid\equiv (SS \xleftrightarrow{K} BS) \quad (10)$$

$$SS \mid\equiv \#(T_{ABS}) \quad (11), \qquad BS \mid\equiv \#(T_{ASS}) \quad (12)$$

$$SS \mid\equiv \#(N_{SS}) \quad (13), \qquad BS \mid\equiv \#(N_{BS}) \quad (14)$$

$$SS \mid\equiv \#(T_{BS}) \quad (15), \qquad BS \mid\equiv \#(T_{SS}) \quad (16)$$

Note that not every protocol needs all of these assumptions. As to the Basic PKM, the reasonable assumptions are (4), (5), (8), (9), (10), and (12). The formal analysis of Basic PKM is performed as follows:

BS sees (1), (4) by rule of message meaning, we deduce:

$$BS \mid\equiv A \mid\sim (T_{ASS}, \mid \xrightarrow{K_{SS}} SS) \quad (17)$$

(17), (12) by rule of nonce verification:

$$BS \mid\equiv A \mid\equiv \mid \xrightarrow{K_{SS}} SS \quad (18)$$

(18), (8) by rule of jurisdiction, we deduce:

$$BS \mid\equiv \mid \xrightarrow{K_{SS}} SS \quad (19)$$

SS sees (2), (5) by message meaning rule:

$$SS \triangleleft SS \xleftrightarrow{AK} BS \ \square \quad (20)$$

We have to stop here now because no assertion can be added. The result we get here is that SS sees there is a key AK it could use to communicate with BS, but it does not know whether it is indeed assigned by BS, not to say the freshness of the key.

The Intel Nonce Version PKM is idealized and analyzed as following:

$$SS \rightarrow BS : SS.REQ,\ \{T_{ASS}, \mid \xrightarrow{K_{SS}} SS\}_{K_A^{-1}} \quad (21)$$

$$BS \rightarrow SS : \{N_{SS}, N_{BS}, \{SS \xleftrightarrow{AK} BS\}_{K_{SS}}\}_{K_{BS}^{-1}}, \quad (22)$$

$$\{T_{ABS}, \mid \xrightarrow{K_{BS}} BS\}_{K_A^{-1}} \quad (23)$$

Similar to Basic PKM, we can get:

$$BS \mid\equiv \mid \xrightarrow{K_{SS}} SS \quad (24),\ SS \mid\equiv \mid \xrightarrow{K_{BS}} BS \quad (25)$$

SS sees (22), (25) by message meaning:

$$SS \mid\equiv BS \mid\sim (N_{SS}, N_{BS}, \{SS \xleftrightarrow{AK} BS\}_{K_{SS}}) \quad (26)$$

(26), (13) by nonce verification:

$$SS \mid\equiv BS \mid\equiv \{SS \xleftrightarrow{AK} BS\}_{K_{SS}} \quad (27)$$

This is slightly different from what is desired, because SS can believe that BS does send the encrypted message, but it can not derive whether BS knows the key, which is encrypted. Authors in [10] suggest signing the key before encrypting it, which seems to be the simplest method. From now on, we make an assumption that this problem has been settled and we can get:

$$SS \mid\equiv BS \mid\equiv SS \xleftrightarrow{AK} BS \quad (28)$$

This will also apply to later analysis on other versions. Now we can continue the procedure:

(28), (9) by jurisdiction rule:

$$SS \mid\equiv SS \xleftrightarrow{AK} BS \ \square \quad (29)$$

We achieve two of three goals of the authentication in this nonce version, missing only that BS can not believe SS.REQ, which corresponds to the Simple Replay Attack.

The PKMv2 is idealized and analyzed as follows:

$$SS \rightarrow BS : SS.REQ,\ \{T_{ASS},\ |\xrightarrow{K_{SS}} SS\}_{K_A^{-1}} \quad (30)$$

$$BS \rightarrow SS : \{N_{SS},\ N_{BS},\ \{SS \xleftarrow{AK} BS\}_{K_{SS}}\}_{K_{BS}^{-1}}, \quad (31)$$

$$\{T_{ABS},\ |\xrightarrow{K_{BS}} BS\}_{K_A^{-1}} \quad (32)$$

$$SS \rightarrow BS : \{N_{BS},\ SS.RPL\}_{AK} \quad (33)$$

The analysis is pretty the same as in Intel version, except the last message. By BS sees (33), (10), and (14), we can get:

$$BS \mid\equiv\ SS.RPL\ \square \quad (34)$$

BS believes SS did send the confirmation message. However, BS can still not believe whether it is SS who sent SS.REQ.

Our modified PKM can be idealized and analyzed as follows:

$$SS \rightarrow BS : \{SS.REQ,\ T_{SS}\}_{K_{SS}^{-1}}, \quad (35)$$

$$\{T_{ASS},\ |\xrightarrow{K_{SS}} SS\}_{K_A^{-1}} \quad (36)$$

$$BS \rightarrow SS : \{T_{SS},\ T_{BS},\ \{SS \xleftarrow{AK} BS\}_{K_{SS}}\}_{K_{BS}^{-1}}, \quad (37)$$

$$\{T_{ABS},\ |\xrightarrow{K_{BS}} BS\}_{K_A^{-1}} \quad (38)$$

BS sees (36), (4), (12), (8), as before we get:

$$BS \mid\equiv\ |\xrightarrow{K_{SS}} SS \quad (39)$$

BS see (35), (39) by message meaning:

$$\mathrm{BS}\mid\equiv\ \mathrm{SS}\mid\sim (SS.REQ,\ T_{SS}) \quad (40)$$

(40), (16) by nonce verification:

$$\mathrm{BS}\mid\equiv\ \mathrm{SS}\mid\equiv\ SS.REQ \quad (41)$$

The rest for SS is similar to PKMv2, and we get:

$$SS \mid\equiv\ SS \xleftarrow{AK} BS\ \square \quad (42)$$

Now we achieve all three goals: (10), (41), and (42).

V. Conclusion and Future Work

In this paper, we illustrate various attacks on Basic PKM authentication protocol, Intel Nonce version, and PKMv2. Methods to counter those attacks are suggested and modified protocol is proposed as well. Finally, we use BAN logic to verify our analysis.

Basic PKM has many flaws such that it provides almost no guarantees to SS about the AK. Intel Nonce version apply mutual authentication to settle those problems, but Simple Replay Attack still exists. PKMv2 adds an additional message at the end of the protocol, intending to assure BS the freshness of the first message. However, this goal fails and Interleaving Attack still applies. Our modified protocol can settle all those problems with fewer messages exchanged between the two principles in WMAN.

The future work will be pursued in two directions. First, more work needs to be done about formal analysis of authentication protocols. BAN logic could do verification and help find some flaws, but it also depends much on manual idealization and formalization which are regarded as hard work for researchers. Analysis procedure is also an onerous work, especially for complicated protocols. Thus we want to use some model-checking tools such as FDR (Failures Divergence Refinement) [18], which is suitable for process algebra CSP (Communicating Sequential Processes) [19], to analyze authentication protocols automatically. Second, there are also many security protocols for multicast and mobility in WMAN, which is proposed in 802.16e. We will analyze them as well as soon as they are available.

Acknowledgment

This work is supported in part by an AFRL/DARPA grant (FA8750-04-2-0260).

References

[1] IEEE Std. 802.16-2001: Air Interface for Fixed Broadband Wireless Access Systems, 2002.

[2] IEEE Std. 802.16-2004: Air Interface for Fixed Broadband Wireless Access Systems, 2004.

[3] A. Marks, "A technical Overview of the WirelessMAN Air Interface for Broadband Wireless Access", IEEE C802.16-02/05, 2002.

[4] IEEE 802.16 and WiMax: Broadband Wireless Access for everyone, Intel White Paper, 2004.

[5] Daniel Sweeney, WiMax Operator Manual: building 802.16 Wilreless Networks, Apress, 2005.

[6] D. Johnston, and J. Walker, "Overview of IEEE 802.16 Security", IEEE Security & Privacy, 2004.

[7] S. Xu, M. Matthews, and C.-T. Huang, "Security Issues in Privacy and Key Management Protocols of IEEE 802.16", Proceedings of the 44th ACM Southeast Conference (ACMSE 2006), March 2006.

[8] D. Johnston, J. Walker, "Mutual Authorization for PKMv2", IEEE C802.16e-04/229, 2004.

[9] J. Mandin, "802.16e Privacy Key Management (PKM) version 2", IEEE C802.16e-04/131r1, 2004.

[10] M. Burrows, M. Abadi, and R. M. Needham, "A Logic of Authentication", Proceedings of the Royal Society of London A, vol. 426, pp. 233-271, 1989.

[11] M. Abadi and R. Needham, "Prudent Engineering Practice for Cryptographic Protocols", IEEE Transactions on Software Engineering, 1995.

[12] C. Anson and C. Mitchell, "Security defects in the CCITT recommendation X.509 – the directory authentication framework", Computer Communication Review, 20(2):30-34, April 1990.

[13] CCITT 1987, CCITT X.509 (3), http://www.lsv.ens-cachan.fr/spore/ccittx509_3.pdf, Nov 2002.

[14] W. Stalling, Cryptography and Network Security: Principles and Practices, 3rd edition. Pearson Education, Prentice Hall PTR, 2003.

[15] CCITT draft recommendation X.509 - the directory authentication framework, version 7, 1987.

[16] W. Mao, Modern Cryptography: Theory and Practice. Pearson Education, Prentice Hall PTR, 2004.

[17] G. Lowe, "A Family of Attacks upon Authentication Protocols", Technical Report 1997/5, University of Leicester, 1997.

[18] FDR2 User Manual, Failure-Divergence Refinement, Formal Systems (Europe) Ltd, May 2000.

[19] S. Schneider, "Security Properties and CSP," Proceedings of 1996 IEEE Symposium on Security and Privacy, 1996.