Formal Verification of IEEE 802.16 Security Sublayer Using Scyther Tool

Ahmed M. Taha[1], Amr T. Abdel-Hamid[1], and Sofiène Tahar[2]
[1]Faculty of Information Engineering and Technology
German University in Cairo, Cairo, Egypt
{ ahmed.mohamedtaha, amr.talaat}@guc.edu.eg
[2] Department of Electrical and Computer Engineering
Concordia University, Montreal, Quebec, Canada
tahar@ece.concordia.ca

***Abstract*— WiMAX (IEEE 802.16) is a standard-based wireless technology that provides high-throughput broadband connections over long distances. WiMAX is used for a number of applications, including "last mile" broadband connections, hotspots and cellular backhaul, and high-speed enterprise connectivity. The former IEEE 802.16 standards used the Privacy and Key Management (PKM) protocol which had many critical drawbacks. In IEEE 802.16e, a new version of this protocol called PKMv2 is released. This paper overviews IEEE 802.16e security, formally analyzes its improvements in contrast with IEEE 802.16-2004 security and potential security flaws. The formal analysis has been conducted using a specialized model checker Scyther, that provides formal proofs of the security protocols.**

***Index Terms*— Formal Verification, IEEE 802.16, Protocol Verification, Security Protocol Analysis, WiMAX**

I. INTRODUCTION

Worldwide Interoperability for Microwave Access (WiMAX) was proposed to provide high speed data distribution through Wireless Metropolitan Area Networks (WMANs), with the advantages of rapid deployment, high scalability, and low upgrade cost. It also provides high-throughput broadband connections over long distances.

WiMAX is described in the IEEE 802.16 standard. The IEEE standard 802.16-2001 [1] was first designed to provide the last mile for WMAN with line-of-sight working at 10-66GHz bands. IEEE standard 802.16-2004 [2] consolidates previous standards and supports non-line-of-sight within 2-11GHz bands and mesh nodes. The latest WiMAX standard, IEEE 802.16e-2005 [3], provides full mobility in broadband wireless access.

Yet, all wireless networks suffer from high security risks, due to usage of open air as a broadcast channel for data. This results in different threats for users, such as loss of privacy, and even loss of data, as well as for providers, such as unauthorized usage, and man-in-the-middle attacks. The 802.16 standard specifies a security sub-layer at the bottom of the Medium Access Control (MAC) layer. This security sub-layer plays a key role in authentication, key establishment, as well as information encryption. There are two component protocols in the security sub-layer: an encapsulation protocol for packet data encryption, and a privacy and key management (PKM) protocol for providing the secure distribution of keying data and authorized access to connections between the Subscriber Station (SS) and the Base Station (BS). There are two versions of privacy key management protocols supported in IEEE 802.16e: PKMv1 and PKMv2.

Concerning the IEEE 802.16 security, an analysis of its security in many sides, such as vulnerability in authentication and key management protocols, and failure in data encryption was introduced in [4]. Mutual authentication is the major contribution proposed by [4], which enables SS to authenticate BS as well. In [5], security issues on the basic PKM protocols and some solutions are proposed. In [6] a formal analysis of PKMv1 and PKMv2 is proposed using the Scyther tool [7], however it did not give a clear study for the threat model or intruder capabilities. Also, the author of [6] did not describe the session keys secrecy and uniqueness, thus their forgery can interrupt the communication between SS and BS. The authors of [8] found a new attack on PKMv2 and introduced a formal analysis of PKMv1 and PKMv2 with Burrows-Abadi-Needham (BAN) logic. However, BAN logic is weak at reasoning about confidentiality and it does not clearly formulate the possible actions of an attacker.

In this paper, we analyze and verify the IEEE 802.16 authentication and key management protocols (both PKM version 1 and 2). 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 [7] to extract extra holes or threat that might exist. The above verification process is used to propose different protocols modifications to ensure formally more secured protocols.

The rest of the paper is organized as follows. Section II provides an overview of the IEEE 802.16 authentication protocols and describes various attacks. In Section III, we formally verify the security of the PKMv1 and PKMv2 authentication protocols, using Scyther, and propose some modifications to these protocols to provide higher security. Finally in Section IV, we conclude the paper.

II. PKM PROTOCOLS DESCRIPTION

Ensuring wireless networks protection means that we need to protect this network against different aspects that can

destroy information integrity. The security objectives which are essential to SS are [6]

- ***Pseudonymity*** This means that an outsider, who keeps track of the communication, cannot relate the traffic to a specific SS.
- ***Information Confidentiality*** Only authorized users have access to information.

On the other hand, the security claim which is important for the BS seen as a service provider is:

- ***No theft of Service should be possible,*** where neither an unauthenticated user should gain access to the services provided, nor should an unauthenticated user be able to impersonate another user.

The security objective which is essential to both the BS and the SS is:

- ***Session Key Establishment,*** where the shared session keys must be secret, and identify the protocol sessions, in the sense that there is exactly one execution of every protocol role sharing the session key.

There are several attack techniques that face wireless networks, and some of these attacks are: Traffic analysis, passive eavesdropping, active eavesdropping, man-in-the-middle-attack, replay attack and multiplicity attack.

In the case of IEEE 802.16, message replay and man-in-the-middle attack are the most famous attacks on PKMv1. Multiplicity and simple replay attack are the typical attacks on PKMv2, as well as the interleaving attack.

A. PKMv1Authentication Protocol

According to the authentication protocol of PKMv1 (fixed version) shown in Fig. 1, the SS begins authorization by sending an Authentication Information message which contains the SS manufacturer's X.509 certificate [9] (Cert (Manufacturer (SS))) to the BS to demonstrate that it is a trust worthy device. Afterwards the SS sends its own certificate (Cert (SS)), its cryptographic capabilities and a 16 bit security association identifier (SAID), this message is called Authorization Request message (Auth-REQ). The certificate of the SS contains its RSA Public Key, MAC address, serial number, and manufacturer ID.

In response to an Auth-REQ, the BS validates the requesting SS's identity, uses the certificate of the SS to determine if the SS is authorized, determines the encryption algorithms and protocols to be shared with the SS, generates an

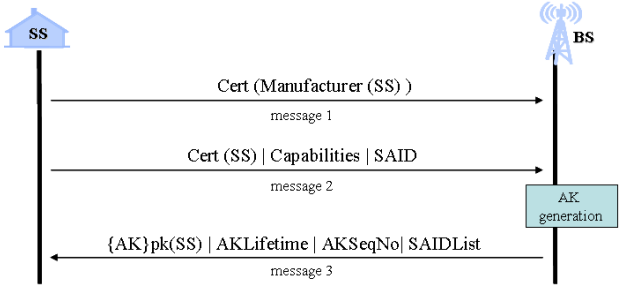

Fig. 1. PKMv1 authentication protocol

Authentication Key (AK), and sends the AK (128 bits), which is RSA encrypted with the received public key of the SS {AK}pk(SS), the lifetime of the AK, a 4-bit sequence number, used to distinguish between successive generations of AKs, and a list of SAIDs which contains the identities and the properties of the SA list the SS authorized to access.

B. Analysis of PKMv1 Attacks

1) Message Replay Attack: If the messages exchanged in an authentication protocol do not carry appropriate freshness identifiers, then an adversary can easily get himself authenticated by replaying messages copied from a legitimate authentication session. BS may face a replay attack from a malicious SS who intercepts and saves the authentication messages sent by a legal SS previously.

Although an adversary eavesdropping the messages cannot derive the AK from messages because it does not have the corresponding private key, the adversary still can replay the message 2 multiple times. Therefore, the Deny of Service occurs to the SS who owns that Cert(SS).

The SS cannot detect a replay attack, because nothing in the protocol leads to this conclusion. To avoid these replay attacks, the authors in [4] proposed adding a nonce to message 2, together with a signature of SS. 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.

Thus, we propose sending a pre-AK to SS instead of AK, and let SS and BS derive AK from the pre-AK at both ends. The pre-AK generation can be modified to bind the SS MAC address and the SS nonce (Ns). Thus, the exposure of the AK may be prevented by adding freshness in the pre-AK, which will be used to derive the AK using the following formula:

$$AK = PRF(pre\text{-}Ak, 128) \quad (1)$$

Where PRF(pre-AK, 128) is a cryptographically secure pseudorandom number function that generates an output of 128-bit length on the input of pre-AK.

2) Man in the Middle Attack: One of the most important security flaws is the one-way authentication, which only authenticates the SS to a BS and not vice versa. SS however has no way of knowing whether the entity sending the AK is a legitimate BS or not. This design lack opens the protocol to forgery attacks, where an unauthorized BS can communicate with a SS. Mutual authentication can eliminate this vulnerability, i.e., SS needs to authenticate BS as well. This can be done by adding BS's certificate in message 3.

C. PKMv2 Authentication Protocol

The latest standard, IEEE 802.16e-2005, includes a new version (PKMv2) of the protocol that caters for the shortcomings of the first version. PKMv2 supports two different mechanisms for authentication: the SS and the BS may use RSA-based authentication or Extensible Authentication Protocol (EAP) -based authentication. We will focus in this paper on RSA based authentication for PKMv2 authentication protocol. The flow of messages exchange in RSA-based authentication is shown as follows:

The SS initiates the RSA-based mutual authentication process by sending two messages. The first message contains the manufacturer X.509 certificate. The second, authorization request message, contains the SS's X.509 certificate, 64-bit SS random number N_s, list of security capabilities that the SS

supports, the SAID and the SS signature. If the SS is authenticated and authorized to join the network, the BS sends an authorization reply message.

In the response message, the BS includes the 64-bit SS random number N_s received, its own 64-bit random number N_b, a 256-bit key pre-primary authorization key (pre-PAK) encrypted with the SS's public key, the pre-PAK key lifetime and its sequence number, a list of SAIDs (one or more), the BS's X.509 certificate and BS's signature in the authorization reply. The SS verifies liveness by comparing the Ns it sent with the received N_s in the authorization response message. It then extracts the PAK, because only the authorized SS can extract the PAK. This can be used as a proof of authorization. Finally, the last message of this authentication is send by the SS to confirm the authentication of the BS. The SS includes the BS random number N_b received in the authorization response message, used to proof liveness, the SS's MAC address and a cryptographic checksum of the message. At the end of the RSA authorization exchange, both SS and BS are authenticated by each other.

D. Analysis of PKMv2 Attacks

1) Interleaving Attack: Without SS signature, the request message is easy to be modified or impersonated. This is similar to what we discussed before on PKMv1 and we refer to it as simple replay attack. Even with the signature from SS served as message authentication, attack still exists. Such attack is similar to the one proposed in [10], which is classified as Interleaving Attack in [8]. This attack uses the messages from previous protocol sessions being run concurrently to the main protocol session, in order to provide the messages in the main protocol session.

2) Multiplicity Attack: A new attack on the original X.509 3-way authentication protocol was found by [11] when one agent is mistaken about the multiplicity of sessions. This attack can be eliminated by adding the BS's identity.

From the above discussion, we can conclude that Basic PKM has many flaws such that it provides almost no guarantees to SS about the AK. 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. So we can conclude that the SS's signature and the BS's certificate are critical to all versions of authentication protocols.

III. FORMAL ANALYSIS OF PKMV1 AND PKMV2

Scyther is an automated security protocol checking, which has proven to be an effective tool for verification, falsification, and analysis of security protocols. It can verify protocols with unbounded number of sessions, with guaranteed termination. As well, it is the only currently existing tool capable of verifying synchronization [12]. Synchronization expresses that the messages are transmitted exactly as prescribed by the protocol description. In other words, whenever an initiator I completes a run of the protocol with responder R, and R has been running the protocol with I. Then, all messages are received exactly as they were sent, in the order as described by the protocol. Synchronization is a strictly stronger property than agreement for the standard intruder model, because it can be used to detect suppress-replay attacks (also known as preplay attacks).

Scyther uses backward symbolic state search technique to analyze security protocols. The backward symbolic state search technique which uses the Arachne engine, based on the Athena method [13]. The Arachne engine finds attacks by searching backwards from the claim that is broken. This technique allows full type flaws and can explore infinite state spaces. Contrary to Athena, Scyther can verify authentication properties such as synchronization as described, and can handle non-atomic keys and multiple key structures. The operational semantics of the Scyther is based on [14].

A. Model Description

In our model, we describe the behavior of the protocol in terms of its roles, either an initiator or a responder. Our system consists of two communicating agents, SS and BS. Each agent performs one or more roles. A role performed by an agent is called a run. Agents execute their runs to achieve their security goals. Every role specification consists of a sequence of events describing the messages the agent shall send and receive, as well as certain security claims. An intruder may try to oppose such security goals. The capabilities of the intruder determine its strength in attacking a protocol run. In order to resist attacks, an agent can make use of cryptographic primitives when constructing messages.

The actual behavior of the entire system, consisting of the intruder and a set of agents executing a number of runs, is encoded in the traces of the system. Every claim event in a trace results in a declaration about the trace that may or may not be true. We focus on three security properties in our work secrecy, secrecy and uniqueness of the session keys and authentication. A secrecy claim event is essentially the statement that something is never known to the adversary. Session-key claim, which is used for session uniqueness and secrecy of the session key, requires that the session key must be secret and act as a session identifier, that is, it must be unique across all runs of the same role of the same protocol. Authentication is captured by the notion of synchronization. Synchronization requires that corresponding send and receive messages have to be executed in the expected order.

The role for which the claim is tested is denoted by *x*, and *y* is the message. The following claims are used:

- *Claim (x,Secret,y)* the agent performing the role *x* know that the intruder will never have knowledge of *y*
- *Claim (x,Nisynch)* agent performing the role *x* knows that the message he received is from an authenticated sender.

Using this description, we can identify the following components of the security protocol model[15].

- *Protocol specification,* describes the behavior of each of the roles in the protocol. Most often, a role in a security protocol is specified as a sequential list of events.

- *Agent model*, agents execute the roles of the protocol. The agent model is based on a closed world assumption. This means that honest agents show only the behaviour described in the protocol specification.
- *Communication model,* describes how the messages are exchanged between the agents. We assume asynchronous communication with a single network buffer because this is the most general approach.
- *Threat model*, a parameter in the semantics of the model based on Dolev and Yao's network threat model [16], where the intruder has complete control over the communication network.
- *Cryptographic primitives,* idealized mathematical constructs such as encryption, using the black box approach. This means that an adversary cannot learn anything from an encrypted message except if he has the key.
- *Security requirements,* expressed as safety properties (i.e. something bad will never happen).

B. Properties Specifications

Pseudonymity, information confidentiality, no theft of service possible [6] and secrecy and uniqueness of the session keys are selected for formal verification. The goals of the authentication protocol should be:

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

Property 1: *{claim (SS, Secret, SsCert)}*

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

Property 2: $\forall_{\alpha \in Msg}$*(claim (SS, Secret,* α*))*

3) No theft of Service possible: 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. This claim is similar to information confidentiality. Its formal definition is given as follows:

Property 3: $\forall_{\alpha \in Msg}$ *(claim (BS, Secret,* α*))*

4) Secrecy and uniqueness of the session keys: 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:

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

C. Formal Verification

Table 1 shows the message elements that are used in our formal model.

TABLE I
MESSAGE TYPES

Message Element	Description
AK	Authorization Key
Nb	BS's Nonce
BsCert	The BS's certificate
Mancert (SS)	Maps an agent SS to its manufacturer certificate
ns1	SS's Nonce
Ns	SS's Nonce
prePAK	pre- primary AK
SAID	Security Association Identifier
SAIDlist	The set of available SAID the SS is allowed to access
SsCert	The SS's certificate
Sig(x)	X's RSA signature over all other attributes of the message
{x}pk (y)	X is encrypted with the public key of y

1) PKMv1 Authentication Protocol: The formal definition of the authentication scenario of PKMv1 described above is shown as follows:

- *SS* $\rightarrow$ *BS: Mancert (SS)*
- *SS* $\rightarrow$ *BS: SsCert, Capabilities, SAID*
- *BS* $\rightarrow$ *SS: {AK}pk(Ss), SAIDlist, AKSeqno*

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

1. *Property 1:* Scyther detected a possible attack, as an intruder eavesdrops the second message and obtains the SS's certificate (SsCert) [6].

The authentication protocol can be fixed by including a public key infrastructure (PKI) certificate for the BS, as shown below. The SS identifier is encrypted with the public key of the BS pk(Bs) and sent to the BS. A nonce (Ns1) can be added for identification.

2. *Property 2:* Scyther identified problems in the authentication protocol, as the SS does not authenticate the BS and so the SS has no way of knowing whether the entity sending the AK is a legitimate BS or not. This design lack opens the protocol to forgery attacks, where an unauthorized BS can communicate with a SS (SS thinks it is communicating with the trusted BS). The result is that the intruder can decrypt the information needed.

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 SS to determine if the SS is authorized, then sends the AK encrypted with the public key of the SS. This guarantees that only the specific SS can decrypt the AK.

4. *Property 4:* It is proved that an adversary cannot obtain the unique AK as it is encrypted with the public key of the SS. 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.

The formal definition of the authentication scenario of the modified PKMv1 described is shown as follows:

- *SS* $\rightarrow$ *BS: Mancert (SS)*
- *SS* $\rightarrow$ *BS: {SsCert,Ns1}pk(Bs), Capabilities, SAID*
- *BS* $\rightarrow$ *SS: {AK}pk(Ss), SAIDlist, AKSeqno, BsCert, Sig(Bs)*

2) PKMv2 Authentication Protocol: The formal definition of the PKMv2 authentication protocol is shown as follows:

- *SS→BS: Mancert (SS)*
- *SS→BS: SsCert, Capabilities, SAID, Sig(Ss), Ns*
- *BS→SS: {prePAK}pk(Ss), SAIDlist, Ns, Nb, prePAKSeqno, prePAKlifetime, Sig(Bs), BsCert*
- *SS→BS: Nb, Sig(Ss)*

This model is going to be challenged with the following requirements using the Scyther tool, as in PKMv1:

1. *Property 1:* A possible attack is detected as in PKMv1, as an intruder eavesdrops the second message and obtains the SS certificate (SsCert).

The authentication protocol can be fixed by including a PKI certificate for the BS. The SS identifier is encrypted with the public key of the BS pk(Bs) and sent to the BS. A nonce (Ns1) can be added for identification as follows:

- *SS→BS: Mancert (SS)*
- *SS→BS:{SsCert, Ns1}pk(Bs), Capabilities,SAID, Sig(Ss), Ns*
- *BS→SS: {prePAK}pk(Ss), SAIDlist, Ns, Nb, prePAKSeqno, prePAKlifetime, Sig(Bs), BsCert*
- *SS→BS: Nb, Sig(Ss)*

2. *Property 2:* In the formal analysis it is proved that the authorization key exchanged in the authentication protocol is secret and not broken if the backward compatibility to PKMv1 is disabled [6].

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:* It is proved that an adversary cannot obtain the unique pre-PAK, which will be used to extract the PAK, as it is encrypted with the public key of the SS. Also pre-PAK is proved to be unique because of synchronization and the fact that pre-PAK is a constant in one of the roles appearing only in one send event, accompanied within a signature by the recipient's nonce.

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 PKMv1 and PKMv2. However, pseudonymity and information confidentiality are broken in both versions of PKM. In addition to attacks, some solutions are introduced to solve those problems.

IV. CONCLUSION

In this paper, we illustrate various attacks on basic PKMv1 and PKMv2 authentication protocols. Basic PKM has many flaws such that it provides almost no guarantees to SS about the AK. Applying mutual authentication by adding nonces to settle those problems was shown, but simple replay attack still exists. A possible solution to counter this attack is proposed and was shown opposing to the introduced attacks especially replay attack. PKMv2 adds an additional message at the end of the protocol, however an interleaving attack still applies. Methods to counter those attacks are suggested and the modified protocol is introduced as well. As seen in the formal analysis, we formally verified the key management protocol of PKMv1 and 2 in terms of the secure session key establishment and distribution, no theft of service, pseudonymity and information confidentiality. Session key establishment and no theft of service are valid in both versions of PKM. However, pseudonymity and information confidentiality are broken in both PKMv1 and 2. Modifications in the protocols were presented to solve those problems.

REFERENCES

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

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

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

[4] D. Johnston and J. Walker, "Overview of IEEE 802.16 Security", IEEE Security and Privacy Magazine, vol. 2, no. 3, pp. 40-48, May-June 2004.

[5] S. Xu, M. M. Matthews, and C. Huang, "Security issues in privacy and key management protocols of IEEE 802.16," Proc. of ACM Southeast Conference, pp.113–118, Melbourne, Florida, USA, March 2006.

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

[7] C. Cremers, "The Scyther tool: Automatic verification of security protocols," http://people.inf.ethz.ch/cremersc/scyther/index.html, 2009.

[8] S. Xu and C. T. Huang, "Attacks on PKM protocols of IEEE 802.16 and its later versions," In Proceedings of 3rd International Symposium on Wireless Communication Systems (ISWCS 2006), Valencia, Spain, September 2006.

[9] R. Housley, W. Polk, W. Ford, and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 3280 (Standards Track), April 2002.

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

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

[12] C. Cremers, S. Mauw, and E. de Vink, "Injective Synchronisation: An Extension of the Authentication Hierarchy," Theoretical Computer Science, vol. 367, no. 1, pp. 139 – 161, 2006.

[13] D. Song, "Athena: A new efficient automatic checker for security protocol analysis," In PCSFW: Proceedings of the Computer Security Foundations Workshop, IEEE Computer Society Press, pp.192, 1999.

[14] C. Cremers and S. Mauw, "Operational semantics of security protocols," In S. Leue and T. Systa, editors, Scenarios: Models, Transformations and Tools Workshop 2003, Revised Selected Papers, vol. 3466 of LNCS, Springer, pp. 66-89 2005.

[15] C. Cremers, "Scyther - Semantics and Verification of Security Protocols," Ph.D. dissertation, Eindhoven University of Technology, Netherlands, 2006.

[16] D. Dolev and A. Yao, "On the Security of Public-Key Protocols," In IEEE Transactions on Information Theory, pp. 198–208, 1983.