

**SECURE AUTHENTICATION SCHEME FOR PRIVACY  
AND KEY MANAGEMENT IN IEEE 802.16e**

**A DISSERTATION**

*Submitted in partial fulfillment of the  
requirements for the award of the degree*

*of*

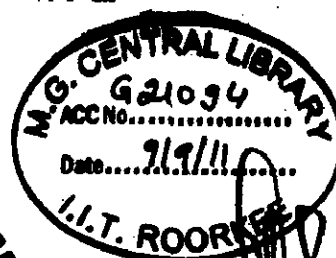
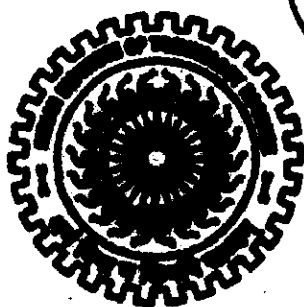
**MASTER OF TECHNOLOGY**

*in*

**COMPUTER SCIENCE AND ENGINEERING**

**By**

**FUDEN TSHERING**



**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE  
ROORKEE -247 667 (INDIA)  
JUNE, 2011**

## CANDIDATE'S DECLARATION

---

I hereby declare that the work, which is being presented in the dissertation entitled "Secure Authentication Scheme for Privacy and Key Management in IEEE 802.16e" towards the partial fulfillment of the requirement for the award of the degree of Master of Technology in Computer Science and Engineering submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand (India) is an authentic record of my own work carried out during the period from July 2010 to June 2011, under the guidance of Dr. Anjali Sardana, Asst. Professor, Department of Electronics and Computer Engineering, IIT Roorkee.

The matter presented in this dissertation has not been submitted by me for the award of any other degree of this or any other Institute.

Date: 1/6/2011

Place: Roorkee

  
(Fuden Tshering)

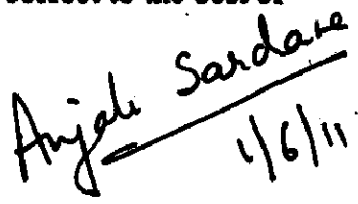
---

## CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 1/6/2011

Place: Roorkee

  
(Dr. Anjali Sardana)

Asst. Professor

Department of Electronics and Computer Engineering

IIT Roorkee.

## ACKNOWLEDGEMENTS

---

I would like to extend my heartfelt gratitude to my guide and mentor **Dr. Anjali Sardana**, Asst. Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, for her valuable advices, guidance, encouragement and sharing her broad knowledge. Her wisdom, knowledge and commitment to the highest standards inspired and motivated me. She has been very generous in providing the necessary resources to carry out my research. She is an inspiring teacher, a great advisor, and most importantly a nice person.

I am greatly indebted to all my friends, who have graciously applied themselves to the task of helping me with ample moral supports and valuable suggestions.

On a personal note, I owe everything to the Almighty and my parents. The support which I enjoyed from my parents, brother and other family members provided me the mental support I needed.

**Fuden Tshering**

## Abstract

---

WiMAX is the fourth generation technology that offers broadband wireless access over long distances to enable pervasive, high-speed mobile Internet access to the widest array of devices. As WiMAX standards expand from considering a fixed line-of-sight propagation and point-to-multipoint infrastructure high frequency system to a lower frequency non-line-of-sight mobile system, WiMAX is open to more security threats than other wireless systems. An authentication and authorization protocol protects the resources of a network from unauthorized access. And after authenticating each other, the Subscriber Station (SS) and Base Station (BS) exchange the key that is used to secure the data communication between them. Although there are standard authentication protocols in IEEE 802.16 but still WiMAX is vulnerable to attacks such as replay attack, denial of service (DoS) attack, interleaving attack etc. The most severe denial of service attack is the attack due to the resource exhausting validation procedure that has to be performed for all SS's authorization request.

In this dissertation entitled "Secure Authentication Scheme for Privacy and Key Management in IEEE 802.16e", a Proxy Base Station based authentication protocol has been proposed. The proposed protocol addresses the major attacks namely DoS attack, interleaving attack, replay attack and downgrade attack. With the introduction of Proxy Base Station (PS), the task of validation is distributed between the PS and BS which resolves the DoS attack due to the resource exhausting validation procedure. The proposed authentication protocol is modeled and verified on Colored PetriNet tool. The results show that our proposed protocol is secure and efficient.

## Table of Contents

<b>Candidate's Declaration &amp; Certificate</b> .....	i
<b>Acknowledgements</b> .....	ii
<b>Abstract</b> .....	iii
<b>Table of Contents</b> .....	iv
<b>List of Figures</b> .....	vii
<b>List of Tables</b> .....	viii
<b>1. Introduction and Statement of the Problem</b> .....	<b>1</b>
1.1 Introduction.....	1
1.2 Motivation.....	2
1.3 Statement of the Problem.....	3
1.4 Organization of the Report.....	4
<b>2. Background and Literature Review</b> .....	<b>5</b>
2.1 Security Requirements.....	5
2.2 Protocol Architecture.....	5
2.3 Security Mechanisms.....	6
2.4 Security Issues in PKMv1.....	8
2.5 Security Issues in PKMv2.....	9
2.5.1 Issues in Protocol Design.....	10
2.5.2 DoS Attack.....	10
2.5.3 Key Space Vulnerability.....	13
2.5.4 Downgrade Attack.....	14
2.5.5 Cryptographic Algorithm Computational Efficiency.....	14
2.5.6 Initial Network.....	15
2.6 Research Gaps.....	17

<b>3. Proxy BS based Authentication Protocol for IEEE 802.16e</b>	<b>19</b>
3.1 Authentication using Proxy BS .....	19
3.1.1 Overall Design .....	19
3.1.2 Message Exchange Details.....	20
3.2 Authorization Request Message.....	21
3.3 Forward Authorization Request Message.....	21
3.4 Authorization Reply Message.....	22
3.5 Key Acknowledgement Message.....	22
3.6 Key Features of Proposed Protocol.....	23
<b>4. Simulation of the Proposed Protocol</b>	<b>24</b>
4.1 Overview of Simulation in CPN .....	24
4.2 Modeling the PKMv2 Protocol .....	26
4.2.1 Modeling PKMv2 without Intruder .....	26
4.2.2 Modeling PKMv2 with Intruder.....	29
4.3 Modeling the Proposed Protocol.....	32
4.3.1 Modeling the Proposed Protocol without Intruder.....	32
4.3.2 Modeling the Proposed Protocol with Intruder.....	36
<b>5. Results and Discussions</b>	<b>39</b>
5.1 Formal Verification Parameters.....	39
5.2 Formal Verification using State Space Analysis Report.....	39
5.2.1 PKMv2 without Intruder.....	39
5.2.2 PKMv2 with Intruder.....	42
5.2.3 PS based Authentication Protocol without Intruder.....	45
5.2.4 PS based Authentication Protocol with Intruder.....	49
5.3 Comparative Analysis .....	52

<b>6. Conclusions and Future Work</b>	<b>54</b>
6.1 Conclusions.....	54
6.2 Future Work.....	55
<b>REFERENCES.....</b>	<b>56</b>
<b>LIST OF PUBLICATIONS.....</b>	<b>59</b>

## LIST OF FIGURES

Figure 1.1	Evolution of WiMAX .....	1
Figure 1.2	Basic Security Approach .....	3
Figure 2.1	Protocol Architecture of IEEE 802.16.....	5
Figure 2.2	Phases in PKM protocol.....	7
Figure 2.3	PKM v1 Protocol .....	9
Figure 2.4	PKM v2 Protocol.....	10
Figure 2.5	Revised authentication protocol.....	11
Figure 2.6	ISNAP .....	11
Figure 2.7	Initial Network Entry with DH Key Agreement .....	15
Figure 2.8	SINEP Scheme .....	16
Figure 3.1	Authentication using Proxy BS .....	19
Figure 3.2	Proposed Authentication Protocol.....	20
Figure 4.1	CPN Top-level Model for PKMv2 with no Intruder .....	26
Figure 4.2	SS page .....	27
Figure 4.3	BS page.....	28
Figure 4.4	Declarations used in the PKMv2 Model with an intruder .....	29
Figure 4.5	CPN Top-level Model for PKMv2 with Intruder .....	30
Figure 4.6	Intruder page.....	31
Figure 4.7	CPN Top-level Model for PKMv2.....	32
Figure 4.8	SS page .....	33
Figure 4.9	PS page .....	34
Figure 4.10	BS page .....	35
Figure 4.11	Declarations used in the Proposed Model with an intruder.....	36
Figure 4.12	CPN Top-level Model for Proposed Protocol with Intruder .....	37
Figure 4.13	Validation of replay message in PS page.....	38



## LIST OF TABLES

Table 2.1	Encryption Strength of RSA and ECC.....	14
Table 2.2	Analysis of Solutions .....	17
Table 3.1	Key Features of Proposed Protocol.....	23
Table 5.1	Analysis of State Spaces.....	49

## Chapter 1

# Introduction and Statement of the Problem

---

### 1.1 Introduction

WiMAX (Worldwide Interoperability for Microwave Access) can be shortly described as: “a telecommunications technology aimed at providing wireless data over long distances in a variety of ways, from point-to-point links to full mobile cellular type access. It is based on the IEEE 802.16 standard”. It is a standards-based technology enabling the delivery of last mile wireless broadband access as an alternative to cable and DSL (Digital Subscriber Line). In order to spread the use of the 802.16 standard solutions, verify the interoperability of 802.16 devices built by different manufacturers and certify interoperable devices, an consortium of wireless device manufacturers was created named as (WiMAX) [1]. WiMAX network has the capability of working on many bands: 2.3 GHz, 2.5 GHz, etc, and provides scalability and mobility with high data rates with NLOS operation. IEEE 802.16 was written in 1999 on Broad Wireless Access (BWA) to develop standards for the global deployment of broadband Wireless Metropolitan Area Networks. The following fig 1.1 shows the timeline of the evolution of WiMAX.

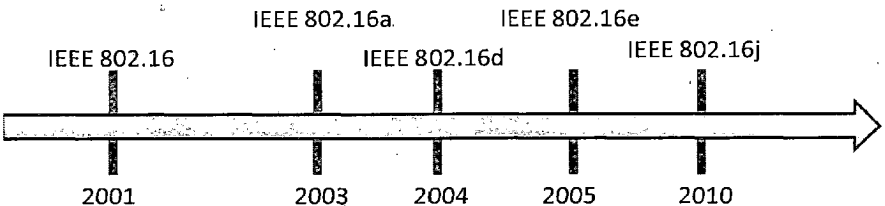


Fig.1.1: Evolution of WiMAX

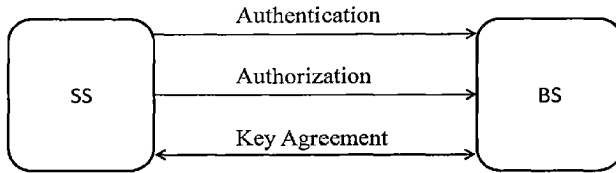
In December 2001, the first 802.16 standard was designed to specialize in point-to-multipoint broadband wireless transmission in the 10-66 GHz spectrum with only a light-of-sight (LOS) capability. But with the lack of support for non- line-of-sight (NLOS) operation, this standard is not suitable for lower frequency applications. Therefore in

2003, the IEEE 802.16a standard was published to accommodate this requirement. Then, after being revised several times, the standard was ended in the final standard: 802.16-2004 [2] which corresponds to revision D. These standards define the BWA for stationary and nomadic use which means that end devices cannot move between Base Stations (BS) but they can enter the network at different locations. In 2005, an amendment to 802.14-2004, the IEEE 802.16e [3] was released to address the mobility which enables mobile stations (MS) to handover between BSs while communicating. This standard is often called "Mobile WiMAX". The privacy and key management protocol requires authentication process for securing the session between BS and subscriber station (SS) and exchange the cryptographic suites and keys.

## 1.2 Motivation

The IEEE 802.16 currently employs the most sophisticated technology solutions in the wireless world, and correspondingly it guarantees performance in terms of covered area, bit-rate, and quality of service. As wireless broadband technology has become very popular, the introduction of WiMAX will increase the demand for wireless broadband access in the fixed and the mobile devices. This development makes wireless security a very serious concern. As the WiMAX technology is touching the sky, it became more ubiquitous among users and eventually prevail the need of more secure and trusted services for better environment. Since there is big need of securing the WiMAX environment so that it can be delivered in public successfully, the privacy and key management work on securing WiMAX.

Fig. 1.2 shows the basic security approach in WiMAX. Authentication addresses establishing the genuine identity of the device or user wishing to join a WiMAX network. Authorization addresses determining whether the authenticated user or device is permitted to join the network. After authorization, key agreement is done between SS and BS.



**Fig. 1.2: Basic Security Approach**

The attacks that we find in this approach are replay, DoS (denial of service) and man in the middle (MITM) attacks. In replay attack the authorization request is replayed multiple times to the BS, which will make the BS ignore the SS. In DoS attack, if a SS sends a lot of false authorization requests to a BS, the BS will use all its resources to calculate whether the certificate is right. This will cause DoS, because BS will not be able to serve any SSs anymore. The MITM is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection when in fact the entire conversation is controlled by the attacker.

### **1.3 Statement of the Problem**

To design and implement a secure authentication protocol for IEEE 802.16e to authenticate and authorize the user and provide the necessary encryption support for the temporal encryption key transfer and data traffic.

This problem can be subdivided into two parts:

- A. Design of proxy BS based authentication protocol for key exchange to reduce the computational overhead of BS.
- B. Enhance the security of authorization request message and ensuring its integrity.

## **1.4 Organization of the Report**

This dissertation report comprises of six chapters including this chapter that introduces the topic and states the problem. The rest of the report is organized as follows.

Chapter 2 describes the background study of IEEE 802.16, description of security mechanisms and brief literature review of related work including research gaps.

Chapter 3 describes the design details of proxy BS based authentication protocol for IEEE 802.16e.

Chapter 4 describes the experimental testbed and simulation details of the proposed protocol.

Chapter 5 describes the result of the simulation results and discussion over the results.

Chapter 6 concludes the dissertation work and gives suggestions for future work.

## Chapter 2

# Background and Literature Review

---

---

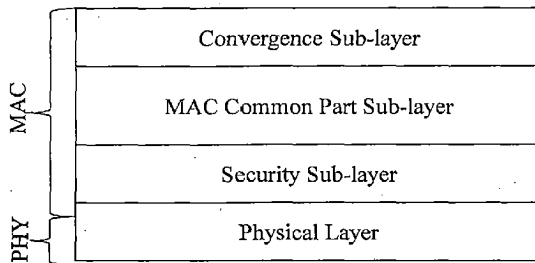
In this chapter, a brief overview of the security mechanisms is described and different issues in the standard PKMv1 and PKMv2 protocols along with different solutions proposed so far.

### 2.1 Security Requirements

Security in 802.16e was thoroughly designed as an important part of the standard architecture due to the additional possible weaknesses that wireless communication endures, especially where the specific network deployment is to cover much larger areas. The security protocol requires mechanisms to ensure confidentiality, integrity and mutual authentication with the implementation of a privacy and key management (PKM).

### 2.2 Protocol Architecture

The protocol architecture of IEEE 802.16 is structured into two main layers: the MAC layer and PHY (physical) layer see Fig 2.1.



**Fig. 2.1: Protocol Architecture of IEEE 802.16**

The MAC layer is divided into three sublayers: Convergence Sublayer (CS), Common Part Sublayer (CPS) and Security Sublayer [4]. The CS sublayer is to converse with higher layers and transform upper-level data services to MAC layer flows and associations. The function of CS sublayer is to receive data from higher layers and to

classify them as ATM cell or packet and forward frames to CPS sublayer [5]. In CPS sublayer, the rules for system access, bandwidth allocation and connection management are defined. Functions like scheduling, connection control and automatic repeat request is defined here. The PHY layer is responsible for receiving MAC frames and transmitting them through coding and modulation of radio frequency signals, providing a two-way mapping. Security sublayer provides secure key exchange and encryption. Security sublayer has two main protocols:

- (a) encapsulation protocol for encrypting packet data across the 802.16 network
- (b) PKM protocol for secure distribution of the key negotiations from the BS to the SS.

### 2.3 Security Mechanisms

The security protocol provides mechanisms to ensure confidentiality, integrity and client authentication with the implementation of a PKM. PKM provides secure key distribution between BS and SS.

The PKM uses security associations (SAs) of which there are two types [6]:

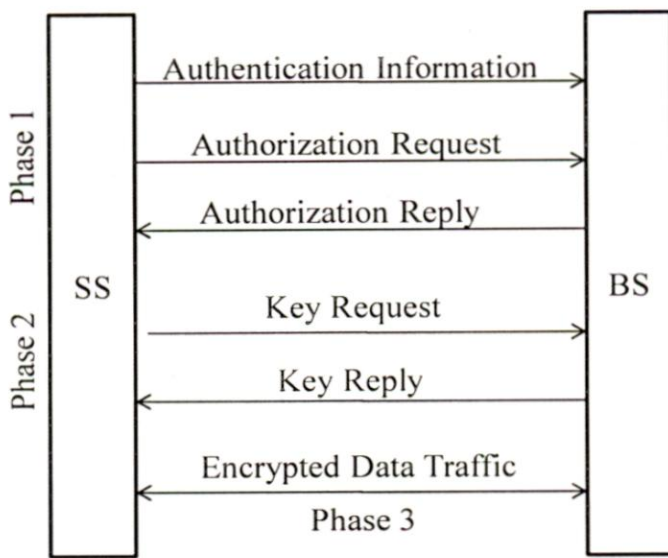
(a) Data SA specifies the messages encryption algorithm and the keys to be used and related information. Each data SA includes an ID (SAID), an encryption algorithm to protect the confidentiality of messages, TEK, and a TEK identifier, a TEK lifetime, an initialization vector for every TEK, and an indication of the type of data SA (primary or dynamic);

(b) Authorization SA includes a credential, an authorization key (AK) to authorize the use of the links, an identifier for the AK, a lifetime for the AK, a key-encryption key (KEK), a downlink hash-based message authentication code (DHMAC), an uplink hash code (UHMAC), and a list of authorized data SAs.

The WiMAX communications follow the security procedure in phases (as shown in fig. 2.2) to ensure secure access of a connection [7].

Phase 1 (SS Authentication and Authorization): To establish the genuine identity of the SS wishing to join BS, the SS sends Authentication Information message containing the X.509 certificate to BS. The X.509 certificate is bound with SS's MAC address. The certificate is issued by the manufacturer or an external authority for the SS. The X.509

authentication service is part of the X.500 series of recommendations that define a directory service. The directory is, in effect, a server or distributed set of servers that maintain a database of information about users. The core of X.509 is the public key cryptography and the digital signatures, and since the standard does not dictate a specific algorithm, RSA (asymmetric cryptography) is recommended [8]. The scheme is complete with the existence of a Certificate Authority (CA). CA issues certificates and binds each entity with a private-public key pair [9]. The certificate contains information like version, a serial number, the certificate issuer, validity period, public key of SS etcetera. The BS may choose to ignore this message. Following first message SS sends authorization request message to BS which contains the X.509 certificate, the list of the cryptographic suite identifiers, each implementing a pair of packet data encryption and authentication algorithms that SS supports, the SS's BCID (basic connection ID), which is the first static CID that BS assigns to SS during initial ranging. After receiving this message, BS authorizes the SS via X.509 certificate and sends authorization reply message back containing AK (authorization key), AK sequence number, AK lifetime and SA descriptors.



**Fig. 2.2: Phases in PKM protocol**

Phase 2 (TEK exchange): After AK exchange the SS derives three keys. (a)KEK for the encryption of the TEK, that BS sends to each SS. TEKs are used for the data encryption to ensure confidentiality. (b)DHMAC key to calculate the HMAC digest for



some of the management messages that it sends to SS, while for the SS it is used to verify the HMAC

Digest from the aforementioned received messages. (c)UHMACH key to calculate the HMAC-Digest for some management messages that it sends to the BS, while the BS uses it to verify the HMAC-Digest of the management messages sent from the SS. The authenticated SS starts a separate TEK process for each SAID. The TEK process periodically sends TEK key request messages to the BS, requesting a refresh of keying material. The BS responds to the key request message with a key reply message, containing the TEK encrypted with KEK, TEK sequence number, TEK's SAID, and the digest of the message with the UHMACH key.

Phase 3 (Encrypted Data Traffic): After the completion of authorization and initial key exchange, data transmission between the BS and the SS starts by using the TEK for encryption. The data encryption [10] is done based on the TEK length, DES in Cipher Block Chaining (CBC) mode using a 6-bit key with 64-bit block encryption along with the 64-bit IV (initialization vector), AES in CCM mode with 128-bit key and 128-bit block size and AES in CBC mode with 128-bit TEK key and 128-bit block size.

## 2.4 Security Issues in PKMv1

The authentication model in 802.16d known as PKM v1 provided one-way authentication, i.e., from SS to BS [11]. It is a three step protocol and uses 1-way authentication. The CertSS is the manufacturer's X.509 certificate preprogrammed in the SS equipment. It is highly informative and contains the public key of SS. The respective authentication protocol can be triggered by any SS using message 1. The message 2 is to request for authentication by the SS using a nonce as token along with X.509 certificate (CerSS) of the SS. In addition, SS also informs the BS of its cryptographic capabilities which determine the supported cryptographic suites and the Basic Connection Identity (BCID). If authentication is successful, BS acknowledges SS with msg. 3 which contains an AK encrypted with SS's public key ( $KU_{SS}(AK)$ ), its sequence number (SeqNo), key lifetime and SAID List. Otherwise, the process is terminated and certificates are discarded. Fig. 2.3 shows the PKM v1 authentication model.

Message 1. SS → BS : Cert(SS.Manufacturer)
Message 2. SS → BS : Cert(SS)   Capabilities   BCID
Message 3. BS → SS : KU <sub>SS</sub> (AK)   SeqNo   Lifetime   SAIDList

**Fig. 2.3: PKM v1 Protocol**

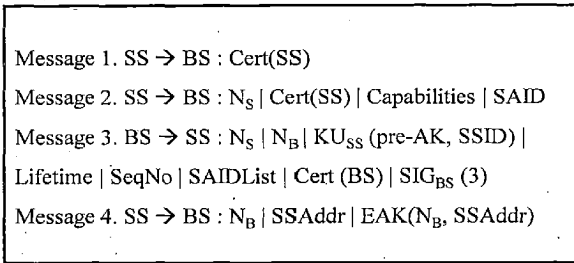
There are open threats in PKM v1 model. If message 1 is captured by an intruder, it can be replayed to drain out the BS capabilities leading to DoS attack. This issue has not been addressed yet. One-way authentication allows for the presence of rogue BS threat. The combination of replay attack and impersonation can lead to an intrusion in the communication of the SS and BS. One serious problem in PKM v1 is that the AK, unique for each SS, is generated solely by the BS. Thus in case of forgery, if AK is leaked out, the TEKs could possibly be generated accordingly.

## **2.5 Security Issues in PKMv2**

Before we start to analyze the authentication protocol of 802.16, we would like to introduce some typical attacks on authentication protocols. Message replay attack is one of the most common attacks on authentication and authenticated key establishment protocols. If the messages exchanged in an authentication protocol do not carry appropriate freshness identifiers, then an intruder can easily get himself authenticated by replaying messages copied from a legitimate authentication session. Man-in-the-middle attack is another classic attack and is generally applicable in a communication protocol where mutual authentication is absent. Other familiar attacks include parallel session attack, reflection attack, interleaving attack, attack due to type flaw and attack due to misuse of cryptographic services. Detailed discussion and examples of these attacks can be found in [12].

### 2.5.1 Issues in the Protocol Design

The PKM v2 model is a four step protocol and uses 3-way authentication. Fig. 2.4 shows the PKM v2 model.



**Fig. 2.4: PKM v2 Protocol**

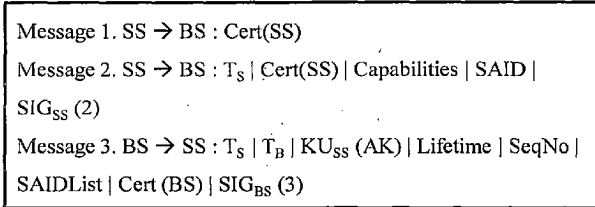
PKM v2 is based on alternating nonce approach as proposed in 802.16e. Although it solves some of the issues in PKM v1, yet a number of these problems remain unresolved. It implements mutual authentication of SS and BS using individual X.509 certificates, CerSS and CerBS, respectively. The incorporation of interchanging nonce helps to link subsequent messages as well as to counter intrusion activity as nonce is random and cannot be easily predicted. The SS Identifier (SSID), unique for each SS in the network, is assigned in msg. 3. Digital Signatures of BS, as in msg. 3, enhance the authenticity of message and the SAID determines the selected security association. An additional fourth step has been introduced in which, the SS acknowledges the authorization reply message with BS's nonce from msg. 3, SS (Physical) address and both these parameters encrypted using the Authorization Key (EAK(N<sub>B</sub>, SSAddr)) [11].

### 2.5.2 Denial of Service Attack

The attacker can easily intercept an authorization request message from a legitimate SS to a BS. Then it replays this message multiple times to the BS, burdening the BS with effect that this declines the legitimate SS. This is a Denial-of- Service attack.

Solution: In [13], the authors have proposed a solution by adding a timestamps in message 2 of basic PKM authentication protocol, together with a digital signature by SS.

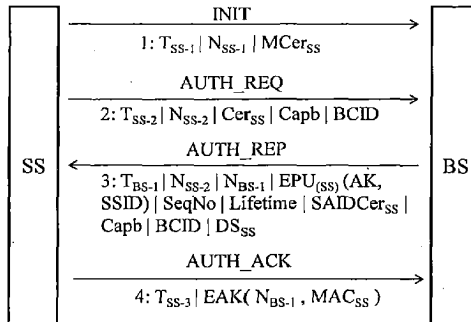
The revised protocol is shown be seen in Fig 2.5. By adding the timestamps and signatures, freshness can be guaranteed for both messages. By adding Cert(BS), mutual authentication is achieved which prevents replay attack from malicious BS. Thus, both the SS and the BS know that the message is fresh and not replayed.



**Fig. 2.5: Revised authentication protocol [8]**

However, the synchronization scheme remains an issue and is yet to be determined. If only timestamps are used, suppress replay attack is likely to occur in which, due to the failure of accurate clock synchronization, messages can be replayed with added delay causing a breach. Nonce does provide a way to link messages, but it is weak if an interleaving attack occurs [13][14]. In this case, the intruder stands in the way between SS and BS, impersonating itself as SS to the BS and vice versa. This attack is explained in detail in [11].

Therefore, the author in [15] proposed a model called Improved Secure Network Authentication Protocol (ISNAP) shown in Fig 2.6.



**Fig. 2.6: ISNAP [15]**

The ISNAP authentication protocol is an extension of the hybrid approach using timestamps together with nonce. The message 1 consists of timestamp (TSS-1), nonce (NSS-1) and MCerSS. BS receives the INIT message and calculates the trip time as:

$$TPROP-1 = TPRESNT - TSS-1 \quad (1)$$

where TPRESNT is the time at which INIT is received. The subsequent messages include AUTH\_REQ and AUTH\_REP which establish mutual authentication between SS and BS along with exchange and validation of their respective credentials. After receiving AUTH\_ACK message, BS calculates the second propagation delay (TPROP-2) as:

$$TPROP-2 = TPRESNT - TSS-3 \quad (2)$$

In optimal environmental conditions, if the whole process of authentication has taken place without any external intrusion, then:

$$|TPROP-1 - TPROP-2| \leq \gamma \quad (3)$$

where  $\gamma$  is the auxiliary parameter introduced to consider the fluctuations in propagation time which occurs due to environmental and multi-path effects. The value of  $\gamma$  must not exceed 3% of the total propagation time (TPROP-1 or TPROP-2), based on empirical analysis considering a Quasi-static Rayleigh Channel. The ISNAP model is robust against the replay attack, DoS attack, interleaving attack, multiplicity attack and man-in-the-middle attack.

In [16], the authors have proposed a technique to counter DoS attacks that uses visual authentication principles. The idea is to perform some pre-authentication steps ensure that the arriving SS is registered and qualified to be considered for the authentication process. This technique considers that the subject unique identifier attribute of the digital X.509 certificate is mandatory rather than to be optional. The value of this field is taken in the form of binary images and registered with Trusted Third Party (TTP) server. This image acts as a piece of secret and unique information that is shared between the SS and BS. With the help of TTP server, both the SS and the BS validate each other. Since the BS does not have to process any X.509 certificates and digital signatures if an SS fails the pre-authentication process, this saves the computational power and resources. If the identity of the SS is validated, the BS continues with the regular authentication process. Not only does this process acts as an additional security measure but also provides an

effective means to counter DoS attacks. This solution is not yet tested in a simulation tool to check and improve its computational and security performance.

A neural network based authentication method has been suggested for the generation of secret key keys in [17] which is based on synchronization of the neural network by mutual learning. The generation process is triggered by competition between stochastic attractive and repulsive forces which act on the weights of the two neural networks. Two dynamical systems synchronizing by mutual signals can prevent an attack as the attacker can only synchronize by listening to exchanged signals. Finally the key is established as the synchronized weights of the two networks. In this method, the attacker has as much knowledge about the process as a node has about another. But it can only listen to the communication and cannot influence the dynamics of the weights of two nodes' neural networks. This helps in securing the secret key sharing.

### **2.5.3 Key Space Vulnerability**

In 802.16e, a 4-bit key sequence number is used to distinguish between successive generations of AKs. Also, a 2-bit key sequence number is used for the same purpose with TEKs. The key reply message contains the sequence number as a part of the TEK parameters. The standard treats the 2-bit key sequence number as a circular buffer, allowing an attacker to interject reused TEKs [18]. An attacker can capture TEK messages and replay them to gain information needed to in order to decrypt the data traffic.

**Solution:** As proposed in [19], the problem can easily be solved by increasing the number of bits for both keys. They could be for example both 8 bits. This would mean sending of a few more bits, but shall not decrease the performance significantly. The main disadvantages are however, that the used encryption and decryption mechanisms will have to be modified. This will probably increase the complexity and will require a standardization action.

#### 2.5.4 Downgrade Attack

In the authorization phase, the authorization request message is an unsecured message from SS to BS specifying the security capabilities SS has. An attacker could, therefore, send a spoofed message to BS containing weaker capabilities in order to convince the BS and the attacked MS to agree on an insecure encryption algorithm.

Solution: As proposed in [19], a possible solution for downgrade attack is that the BS could ignore messages with security capabilities under a certain limit. This could lead to DoS for SSs that does not have the required capabilities though.

#### 2.5.5 Cryptographic Algorithm Computational Efficiency

In the authorization phase, the standard model uses RSA encryption algorithm for encryption which is having a key size of 1024 bits. But RSA is less efficient than ECC as it uses stronger keys (1024 bits) at more cost and ECC is much faster than RSA.

**Table 2.1: Encryption Strength of RSA and ECC**

<b>RSA (key sizes in bits)</b>	<b>ECC (key sizes in bits)</b>
112	512
160	1024
224	2048
256	3072
384	7680
512	15360

Solution: The RSA-based public key cryptography can be replaced with ECC as it is more efficient [20, 21]. ECC can provide the same level of security as RSA with smaller key sizes. For example, 160-bit ECC provides comparable security to 1024-bit RSA. ECC provides faster computational efficiency. Since ECC key size is relatively smaller than RSA key size, thus encrypted message in ECC is smaller, energy and bandwidth efficient. Table 2.1 provides additional information to describe the security level desired.

### 2.5.6 Initial Network Entry Vulnerability

The initial network entry process is the first gate to establish a connection to Mobile WiMAX. When SS first tries to join WiMAX network, it sends a Ranging Request (RNGREQ). BS sends a Ranging Response (RNG-RSP) to SS to change Timing, Power Level, Offset Frequency, Ranging Status, and other Ranging parameters. The attacker can intercept this RNG-RSP message and send the spoofed RNGRSP message by setting the RANGING\_STATUS value to 2 which means “abort”. This leads to a DoS attack.

Solution: To resolve this problem, [22] applies Diffie-Hellman (DH) key agreement scheme to initial ranging procedure as shown in fig 2.7. DH key agreement is a kind of key management method to share an encryption key with global variables known as prime number ‘p’ and ‘q’ a primitive root of ‘p’. After choosing ranging code, SS generates ‘p’ and ‘q’. Then SS sends the global variables along with the ranging code to the BS. After verifying the received key and variables, BS also sends its public key to SS. If the received key and variables are verified, BS also sends its public key to SS. Thus, BS and SS can share global variables and public key with which they generate secret key and establish secret communication channels.

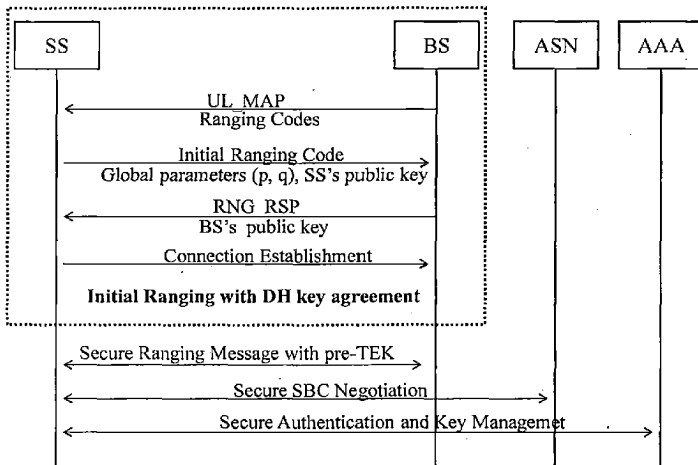


Fig. 2.7: Initial Network Entry with DH Key Agreement [22]



However the original DH key exchange protocol cannot prevent man-in-the-middle attacks [23] since it provides no identity authentication.

To resist man-in-the-middle attacks in this procedure, the authors [23] have enhanced the DH key exchange protocol by introducing identity authentication. In [23], the author assumes that every SS has its own International Subscriber Station Identity (ISSI) and using this ISSI, SS can generate Temporary Subscriber Station Identity (TSSI). This TSSI is used as SS's identity. The author also assumes that legitimate BS has the hash value,  $H(TSSI)$ . The author uses  $H(TSSI)$  as an input parameter of hash authentication function instead of direct usage of TSSI, because in certain situation, one of the legitimate BSs may be captured by attackers, storing  $H(TSSI)$  in BS prevents attackers to achieve the SS's TSSI. The secure initial network entry is shown in Fig 2.9.

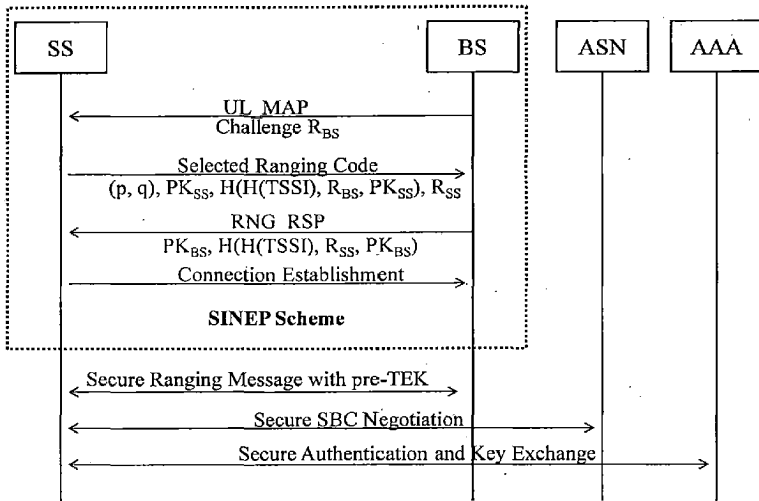


Fig. 2.8: SINEP Scheme [23]

In this protocol, along with the DH key exchange, the SS and BS sends the challenge to each other. The BS sends a challenge  $R_{BS}$  to SS, in turn SS generates hash value using cascade of  $H(TSSI)$ ,  $R_{BS}$  and its public key  $PK_{SS}$  as input. This Hash value is send to BS along with its public key  $PK_{SS}$  and challenge  $R_{BS}$ . Then, the BS calculates the hash value using same inputs and compares it with the SS's response to check identity of SS.

If SS is legitimate, BS calculates hash value using the cascade of  $H(TSSI)$ , RSS and its public key PK<sub>BS</sub> as input and sends it to SS. The SS checks BS's identity using the response that it receives, if the BS is legitimate, the shared key is established and SS continues to communicate with BS; otherwise, SS ceases the communication.

## 2.6 Research Gaps

The following table 2.2 shows the analysis of the solutions addressed for different issues [24].

**Table 2.2: Analysis of Solutions**

S. No.	Solution	Issue addressed	Advantages	Disadvantages
1.	Nonce [13]	Denial-of-Service	synchronization not required	unable to check the freshness of the message
2.	Timestamp [13]		prevents simple replay attack.	requires the time synchronization
3.	timestamp together with nonce [15]		prevents interleaving attack.	difficult to consider the value of $\gamma$
4.	Visual cryptography for pre-authentication [16]		successfully avoids the request from rogue SS	increases the computational overhead by introducing TTP server
5.	Neural cryptography [17]		very secure key exchange	requires complete change in the authentication standards
6.	increase the size of key space [19]	Key Space vulnerability	prevents the circular key space attack	requires modification in the authentication standard and hardware update.
7.	Ignore the cryptographic capabilities beyond certain limit [19]	Downgrade Attack	prevents downgrade attack	vulnerable to DoS
9.	ECC [20,21]	Cryptographic algorithm computational efficiency	ECC requires less key size and computation.	requires modification in the standards.
10.	Diffie-Hellman key exchange [22]	Initial Network Entry	provides key to secure the messages	vulnerable to man-in-the-middle attack
11.	SINEP Scheme [23]		Prevents man-in-the-middle attack	many assumptions and increase in computation cost

The authors in [13, 15, 16, 17] solve the DoS/Reply attacks. They require a reasonable modification to the standards. In [15], computing and analyzing the value of  $\gamma$  increases the complexity. Although [16] counters DoS effectively, it has increased the number of message exchanged thus affecting the performance. In [19], the authors proposed completely new protocol for authentication and authorization process which requires complete modification to the standard. In [19], the author solves the key space vulnerability issue. However experiments are needed to validate the behaviour and performance of this solution. Also, the author [19] solves the downgrade attack but it may create another issue of DoS, so this solution cannot be considered to operate satisfactorily. The author in [20] described that ECC is better than RSA. In [22, 23], the authors solve the initial network entry vulnerability issue but still it is prone to other attacks.

Our proposed proxy BS based authentication protocol addresses the following issues:

- DoS attack
- Downgrade attack
- Interleaving attack (MITM)
- Replay attack

## Chapter 3

# Proxy Base Station based Authentication Protocol for IEEE 802.16e

---

---

### 3.1 Authentication using Proxy Base Station

#### 3.1.1 Overall Design

We introduced proxy base station (PS) between the SS and the BS. Here the PS receives both authentication information and authorization messages sent by SSs. Then PS validates the authorization request sent by the SS and if the request is valid, the PS forwards the authorization request to the BS, otherwise, it does not. Since the PS already validated the authorization request sent by SS, BS simply reads the respective authorization request message and sends authorization reply message to the SS. And then, the SS acknowledges the authorization reply message to the BS. Fig. 3.1 shows the overall design of proposed protocol [25].

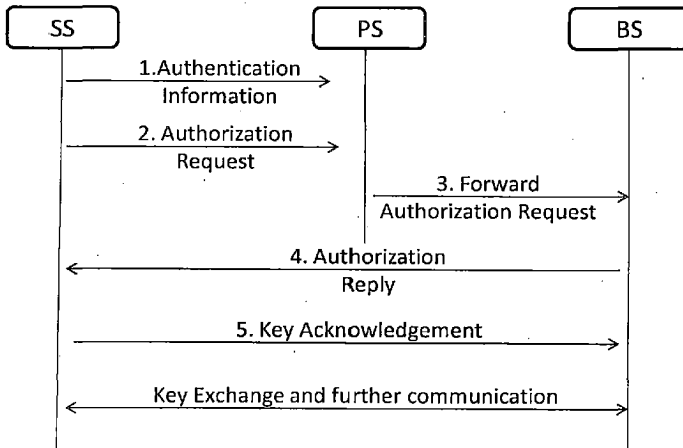
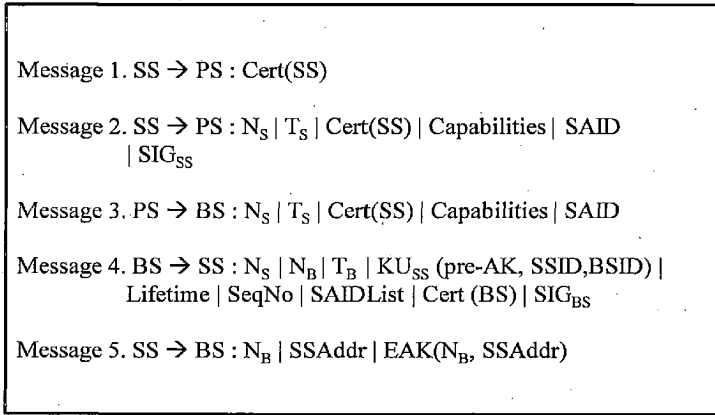


Fig. 3.1: Authentication using Proxy BS

### 3.1.2 Message Exchange Details

The messages exchanged among various stations are shown in figure 3.2. The exchanged messages are as follows:

Message 1: The authorization process (see figure 3.2) begins with the Authentication Information message from SS to PS. The message contains the X.509 certificate (Cert(SS)) which is bound with SS's MAC address. The certificate contains information like version, serial number, the certificate issuer, validity period, public key of SS etc.



**Fig. 3.2: Proposed Authentication Protocol**

Message 2: Following the first message, SS sends the authorization request to the PS. It contains SS's 64 bit nonce ( $N_S$ ) and timestamp ( $T_S$ ), SS's X.509 certificate, Capabilities describing requesting SS's security capabilities, SA identification number (SAID) and signature of the SS over whole message ( $SIG_{SS}$ ) using the private key of SS.

Message 3: After validating the authenticity, uniqueness and freshness of message 2, using the  $SIG_{SS}$ ,  $N_S$  and  $T_S$ , if SS's request is legitimate, PS forwards the SS's authorization request to the BS.

Message 4: BS checks for basic unicast services and possibly additional statically services the SS is subscribed for, and finally, it determines the cryptographic suite from SS's list from the second message. Then, the BS generates the pre-AK. Then BS sends

authorization reply message to the SS containing  $N_S$ ,  $N_B$  (64 bit nonce of BS),  $T_B$  (timestamp), pre-AK (authorization key), SSID and BSID together encrypted by the public key of SS present in the certificate of SS, AK's lifetime and a 4-bit sequence number (SeqNo), SAIDList (contains the identities and the properties of the SA's for which SS is authorized to obtain keying information), BS's certificate (for mutual authentication) and signature of the BS ( $SIG_{BS}$ ) for message integrity.

Message 5: When the SS receives the message; it decrypts the pre-AK with its private key, reads the defined cipher suite and the SAIDs, and proceeds to key exchange with BS. SS sends the acknowledgment message containing  $N_B$  and SSAddr (MAC address) encrypted by AK.

In this way, mutual authentication and three way handshake between SS and BS are achieved.

### **3.2 Authorization Request Message**

In PKMv2 authentication protocol, the authorization request message sent by SS is received by BS. This message is open and can be read and modified easily by intruder without any problem. The integrity of the message can be easily compromised. This message is vulnerable to replay, DoS, MITM and downgrade attacks.

In proposed protocol, the nonce in authorization request message (message 2) prevents the replay attack and the timestamp shows the freshness of the message. So the replay attack is tackled successfully. The signature over the message 2 provides the non-repudiation, integrity and also prevents downgrade attack in which an attacker can send a spoofed message to BS containing weaker capabilities in order to convince the BS and the attacked SS to agree on an insecure encryption algorithm. The use of nonce and timestamp digitally signed by the SS successfully prevents interleaving attack too.

### **3.3 Forward Authorization Request Message**

There is no concept of PS in PKMv2 protocol so there is no forward authorization request message in PKMv2 protocol.

In proposed protocol, since the PS is validating the entire request, it simply decreases the computational overhead of the BS, and thus, the BS can reluctantly provide services

to the legitimate SSs. Both timestamp and nonce are used in the proposed authentication protocol. The PS communicates with the BS through a secured channel so the message<sup>3</sup> forwarded by PS, after validating the SS's authorization request, need not be encrypted. Thus, it saves the computational overhead for BS to validate the SS. BS simply checks for basic unicast services and possibly additional statically services the SS is subscribed for, and finally, it determines the cryptographic suite from SS's list from the second message. Then, the BS generates the pre-AK.

### **3.4 Authorization Reply Message**

In PKMv2 protocol the authorization reply message is signed by the private key of BS which non-repudiation and integrity of the message. But there is no timestamp so the freshness of the message cannot be detect by the SS. This message contains pre-AK and SSID encrypted with the public key of SS but still this message is vulnerable to interleaving attack.

In message 4 of proposed protocol, the  $N_s$ , assures that it is a reply of previous message and  $T_B$  assures the freshness of the authorization reply message. The pre-AK is encrypted using the public key of SS and can be decrypted only using the private key of SS that assures the integrity and confidentiality of pre-AK. The BSID and SSID are also encrypted together with the pre-AK. This adds extra security measure so that none of attacker can use both of the stations as oracle providing complete protection from interleaving attack. The  $\text{Cert}(\text{BS})$  in message 4 completes the mutual authentication and  $\text{SIG}_{\text{BS}}$  provides non-repudiation and integrity of message<sup>4</sup>. Also it adds the fact that the message is intended for the respective SS only because pre-AK is encrypted with public key of that SS.

### **3.5 Key Acknowledgement Message**

The key acknowledgement message is same in both PKMv2 and proposed protocols. The key acknowledgement message acknowledges the successful exchange of the key and completes the three way handshake protocol. The SSAddr (SS's address) is used by the BS to derive AK together with pre-AK and BSAddr.

### 3.6 Key Features of Proposed Protocol

The SSs cannot overwhelm the BS with too many rogue requests because of the presence of PS. Even if the SSs overwhelm the PS with rogue requests, the BS would continue providing services to the authorized SSs. Since the PS is validating the authorization request sent by the SSs, the computational overhead of BS decreases, and hence, the DoS due to the computational overhead is tackled. Here, the BS will not experience any unnecessary computation and can continue providing services to the authorized SS. Although, there is an overhead of adding the proxy station, which increases the number of messages communicated and also the response time of the BS to the SS, it's prepared for the worst DoS attack. To implement this approach in real time, a major amendment in the standard and hardware is required.

**Table 3.1: Key Features of Proposed Protocol**

<b>Attacks</b>	<b>Features to defend attack</b>
Replay Attack	Message 2, contains both timestamp and nonce, is digitally signed by SS.
DoS Attack	Proxy base station validates the message 2.
Interleaving attack	Message 2 digitally signed by SS and the message 4 contains the identity of BS along with pre-AK encrypted by public key of SS.
Downgrade attack	The integrity of message 2 cannot be compromised because of digital signature of SS.



## Chapter 4

# Simulation of the Proposed Scheme

---

In this chapter, we present the models of PKMv2 and proposed schemes with and without intruder using Colored PetriNet (CPN) tool [26]. The intruder model is developed and integrated into the protocol model. Then model checking is performed in CPN Tools.

### 4.1 Overview of Simulation in CPN

CPN based formal techniques are quite suitable for security verification [27, 28] because:

- They have strong formal description capability and well-defined semantics. Their graphical representation and interactive simulation capability help to visually demonstrate concurrency and synchronization of protocol running.
- CPN has many well-studied mathematical analysis methods like reachability tree, matrix equations, place, and transition invariants [29]. They are used to verify whether a system model could provide structural and behavior properties, such as liveness, deadlock, boundedness and fairness properties.

Liveness property assumes that if the authenticator sends Msg1, it will receive Msg4 definitely. It means the handshake executes successfully.

Fairness determines whether the set of transition instances (specified in the list) is impartial or fair.

CPN can test whether the deadlock appears in the modeled system or not. Deadlock means that the protocol will unexpectedly terminate in the case of resource accessing conflict or unlimitedly waiting for acknowledge packets [30].

Boundedness calculates the maximal or minimal number of tokens on a place.

CPN based model checking method executes by three steps:

- To give the CPN model of the protocol. Hierarchical CPNs are always used to demonstrate both protocol framework and functional details.
- To give the formal specification of properties.

• The last step is to run a model checking software to implement the verification and analysis the results. Passing the verification means the protocol can provide corresponding properties.

We use a top-down modeling approach. At the highest level of abstraction, an entity is modeled as a substitution transition. Each substitution transition is defined in a separate subpage that provides a lower level description of the behaviour of the entity.

In modeling a cryptographic protocol, we follow these steps:

**Build a model with no intruder:** In this step,

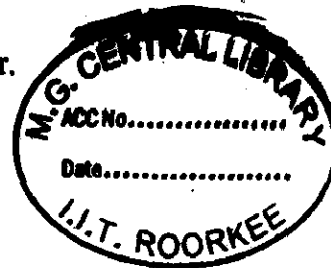
- using CPN ML (modeling language) notation, we declare the color sets and variables that will be used in the net inscriptions of the CPN model.
- we build a top-level model in which the protocol entities are modeled as substitution transitions.
- we define the substitution transitions from the top-level model.

**Add the intruder to the model:** In this step, we

- extend the CPN declarations to include the intruder.
- add the intruder transition to the top-level model.
- define the intruder substitution transition.
- implement a token-passing scheme.
- specify security requirements stated in terms of CPN markings.
- analyze the resulting occurrence graph by using OG queries to locate markings that violate a security requirement.

The simulation is done in the following order:

Do the state space analysis to check the liveness and deadlock-free property.



## 4.2 Modeling the PKMv2 Protocol

The PKMv2 protocol as described earlier is modeled on CPN tool. The model checking is done without and without intruder part.

### 4.2.1 Modeling PKMv2 without Intruder

The following diagram shows the hierarchical CPN – with separate pages (subnets) for the SS, the BS and the intruder part.

#### 4.2.1.1 The Top-Level Model

It presents the model of PKM protocol in a modular way. Thus, the model of a protocol is constructed by using sub-models of its agents. In CPN, this is implemented by using substitution transitions. First, we focus on the messages exchanged between the protocol entities. At this level, protocol entities are modeled as transitions. Fig 4.1 shows a top-level model of the PKMv2 protocol.

The two transitions SS and BS represents the SS and BS of PKMv2 protocol. Here the place A represents the authentication information message, place B represents the

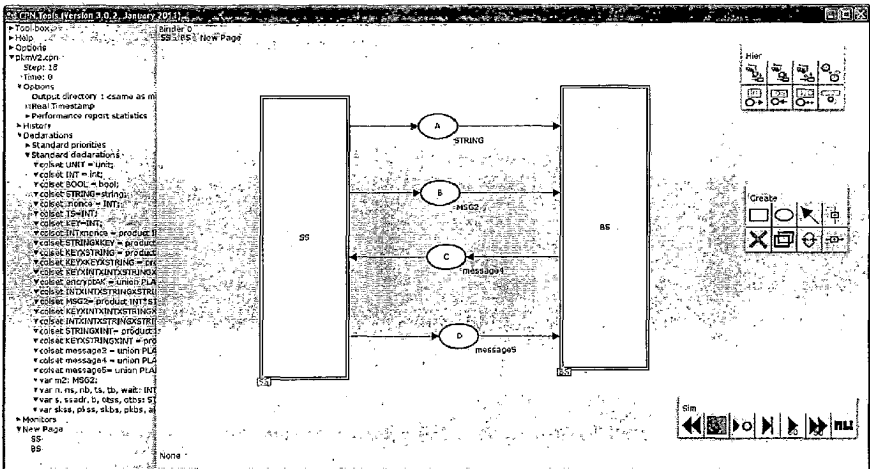


Fig. 4.1: CPN Top-level Model for PKMv2 with no Intruder

authorization request, place C represents authorization reply message and place D represents key acknowledgement message.

#### 4.2.1.2 Defining the Top-Level Substitution Transitions

We consider in detail the model of the SS and the BS. Fig 4.2 shows the CPN model of the SS. It contains three subnets: one models the subtask of SS initiating a protocol run in step 1, the second step sending authorization request and the third one receiving the authorization reply and acknowledging the reply.

In this page, SS sends the authentication information message through SendMSG1 place. The transition generateMsg2 generates the authorization message, using the three places for certificate, nonce and other items such capabilities and SAID, and sends it to BS through sendMSG2 place. The authorization reply is received at place sendMSG3 which is decrypted using the public key of BS and then the validation of this message is done. After validating the message the AK is conceived by decrypting the encrypted AK part within received message. The nonce and AK key are used by SS to generate message 4.

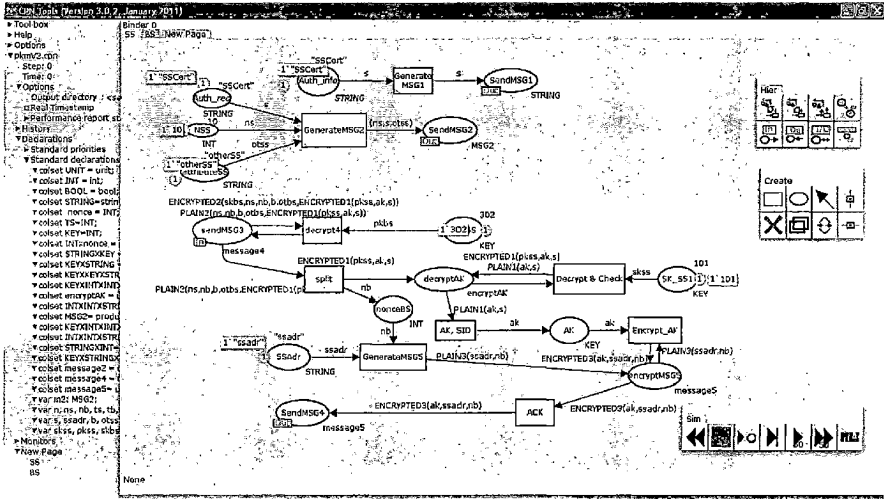


Fig 4.2: SS page

The transition Encrypt\_AK shows the encryption of the message plain message generated at transition GenerateMSG4. The sendMSG4 place is the key acknowledgement message.

Fig. 4.3 shows the CPN model of the BS. It contains three subnets: receiving the authentication information, protocol run in step 1, the second step receiving authorization request, the third one is sending the authorization reply and receiving acknowledgement.

In this page, place sendMSG1 is used to receive the authentication information and the certificate of SS is validated. The BS receives the authorization request through sendMSG2 place, it directly analyses the request for validation and then according to the SS's capabilities it generates AK and encrypts it with public key of SS at place Encrypt\_PKSS using the key from PK\_SS place. Then the transition GenerateMSG3 generates the message. This message is encrypted at transition Encrypt\_BSSK using the private key of BS from BS\_SK place. This message is send to SS through SendMSG3 place. Through place SendMSG4 the BS receives the key acknowledgement message and is decrypted at transition decryptACK using the key provided by GenerateAK place. If decrypted successfully the AK is exchanged successfully.

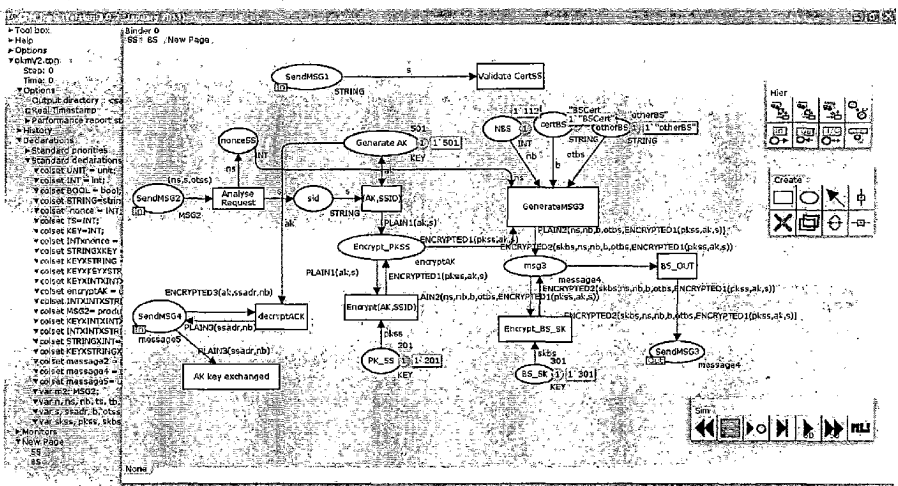
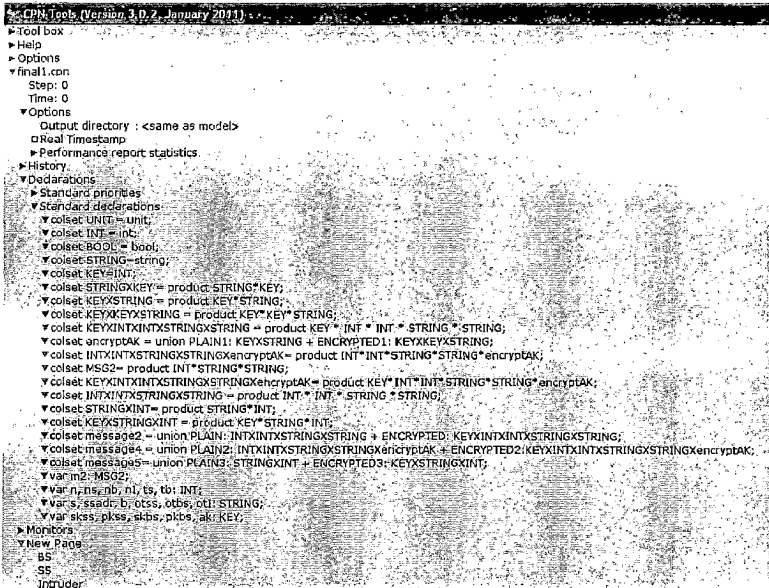


Fig. 4.3: BS page

## 4.2.2 Modeling PKMv2 with Intruder

If we try to analyze a cryptographic protocol by adding a general intruder model, we need to consider a large number of cases; which makes the analysis task infeasible in many situations. The idea is to find the set of modified output data that doesn't affect the acceptance of the data by the legitimate protocol users. Then, we need only to use this set of data in our analysis. This step helps us to identify the vulnerable points in the protocol and to adaptively model the intruder. The inputs to this automated step are the variables that the intruder can modify, and the variables which the legitimate user will check. The simulation will specify the set of vulnerable data which can be modified by the intruder and still will be accepted by the legitimate stations.

In order to add the intruder to the model, one must extend the CPN ML declarations. The final CPN ML declaration for PKMv2 is shown in following fig 4.4.



```
CPN Tools (Version 3.0.2, January 2011)
- Toolbox
- Help
- Options
  - final1.cpn
  - Step: 0
  - Time: 0
  - Options
    - Output directory : <same as model>
    - Real Timestamp
    - Performance report statistics
  - History
  - Declarations
    - Standard priorities
    - Standard declarations
      - colset UNIT = unit;
      - colset INT = int;
      - colset BOOL = bool;
      - colset STRING = string;
      - colset KEY = int;
      - colset STRINGKEY = product STRING*KEY;
      - colset KEYXSTRING = product KEY*STRING;
      - colset KEYKEYXSTRING = product KEY*KEY*STRING;
      - colset KEYXINTXSTRINGXSTRING = product KEY*INT*INT*STRING*STRING;
      - colset encryptPAK = union PLAIN1: KEYXSTRING + ENCRYPTED1: KEYKEYXSTRING;
      - colset INTXINTXSTRINGXSTRINGXencryptPAK = product INT*INT*STRING*STRING*encryptPAK;
      - colset MSG2 = product INT*STRING*STRING;
      - colset KEYXINTXSTRINGXSTRINGXencryptPAK = product KEY*INT*INT*STRING*STRING*encryptPAK;
      - colset INTXINTXSTRINGXSTRING = product INT*INT*STRING*STRING;
      - colset STRINGXINT = product STRING*INT;
      - colset KEYXSTRINGXINT = product KEY*STRING*INT;
      - colset message2 = union PLAIN: INTXINTXSTRINGXSTRING + ENCRYPTED: KEYXINTXSTRINGXSTRING;
      - colset message4 = union PLAIN2: INTXINTXSTRINGXSTRINGXencryptPAK + ENCRYPTED2: KEYXINTXSTRINGXSTRINGXencryptPAK;
      - colset message5 = union PLAIN3: STRINGXINT + ENCRYPTED3: KEYXSTRINGXINT;
      - var m2: MSG2;
      - var n, ns, nb, nl, ts, tbi: INT;
      - var s, ssad, b, ocss, otbs, oli: STRING;
      - var sks: pks, aks, ptbs, ak: KEY;
    - Monitors
      - New Page
      - BS
      - SS
      - Intruder
```

Fig. 4.4: Declarations used in the PKMv2 Model with an intruder

### 4.2.2.1 Top-Level Model with an Intruder

Fig 4.5 shows the top level model of the PKMv2 protocol with an intruder. The substitution transition IR represents the intruder. The intruder is modeled as a separate entity that controls the communication channels between the protocol entities. Thus, it intercepts the exchanged messages and stores them for future use. Then, it attempts to decrypt the encrypted portions of the intercepted messages. Finally, it attempts to modify the message contents, or even to generate new messages to replace the intercepted ones.

Here the messages A,B,C,D are intended to be exchanged between the SS and the BS but the intruder intercept, store, modifies and forwards the spoofed messages. The places A, B, C and D shows the legitimate messages sent by SS and BS and the places IA, IB, IC and ID shows the modified messages sent by the intruder to get the control over communication channel between SS and BS.

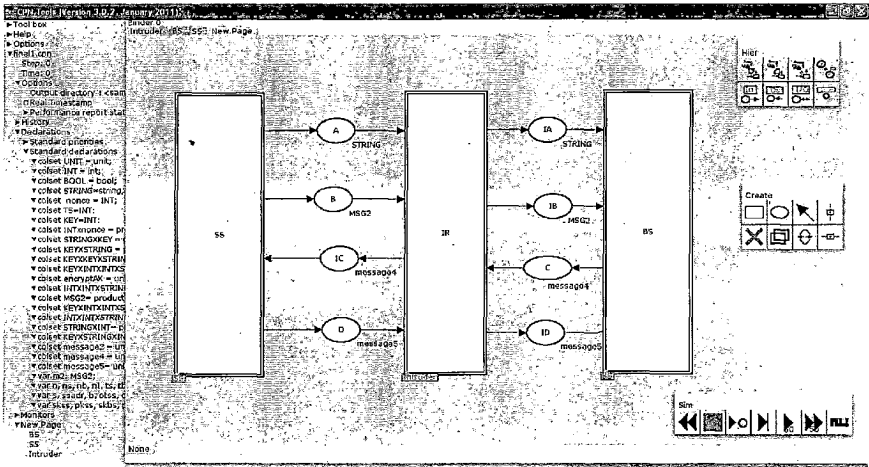


Fig 4.5: CPN Top-level Model for PKMv2 with Intruder

#### 4.2.2.2 Defining the Intruder Substitution Transition

The intruder substitution transition (IR in Fig 4.5) is defined by the subpage intruder shown in Fig 4.6. The intruder first stores the fields of the intercepted message. Then, it tries to decrypt the cipher using one of the public keys stored in its database. Finally, the intruder forms a new message to be sent in place of the intercepted one.

This page contains four subnets. First stores and modifies the authentication information message. Second, stores and modifies the authorization request message. Third, receives the authorization reply message and forwards the modified one to SS. And finally accepts the key acknowledgement. Here the intruder receives the authentication message through sendMSG1 place which it stores and forwards the desired message to BS through sendMIMSG1 place. The authorization request is received at sendMSG2, the intruder stores and modifies this message and sends to BS through sendMSG2. The sendMSG3 is the authorization reply message received from BS and forwarded the modified one to BS through place sendIMSG3. The place sendMSG4 is the key acknowledgement message received from SS and sendIMSG4 is the modified message send to BS.

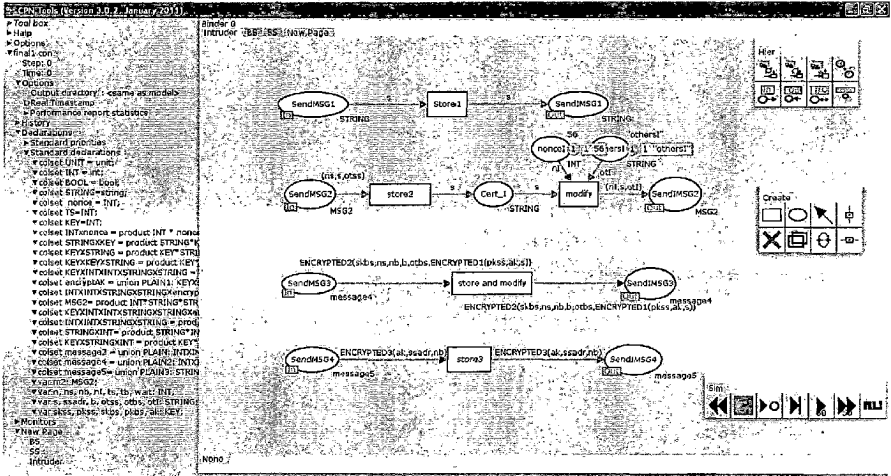


Fig 4.6: Intruder page



### 4.3 Modeling the proposed Proxy BS based Authentication Protocol

The proposed protocol as described earlier is modeled on CPN tool. The model checking is done without and without intruder part. The following sections describe the hierarchical CPN – with separate pages (subnets) for the SS, the BS, the PS and the intruder part.

#### 4.3.1 Modeling the Proposed Protocol without Intruder

The proposed protocol is modeled in a hierarchical approach. The following sections describe the hierarchical CPN – with separate pages (subnets) for the SS, the BS and the PS.

##### 4.3.1.1 The Top-Level Model

It presents the model of proposed protocol in a modular way. Thus, the model of a protocol is constructed by using sub-models of its agents. In CPN, this is implemented by using substitution transitions. First, we focus on the messages exchanged between the protocol entities. At this level, protocol entities are modeled as transitions. Fig 4.7 shows a top-level model of the proposed protocol.

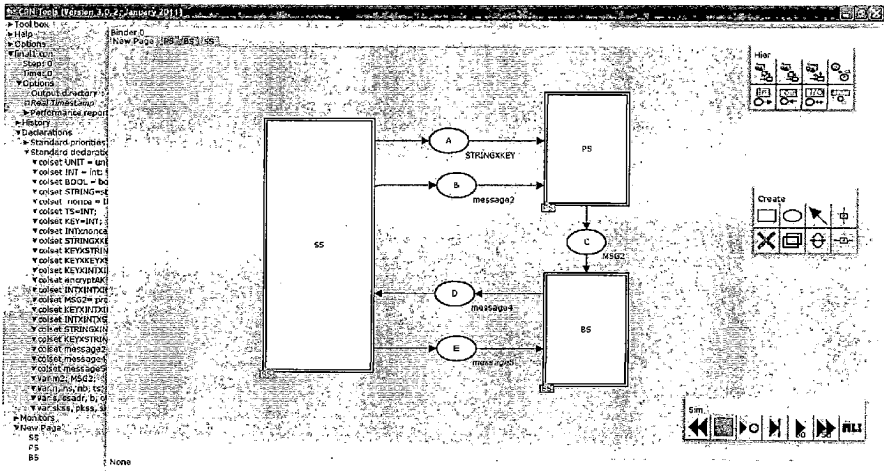


Fig. 4.7: CPN Top-level Model for Proposed Scheme

The three transitions SS, PS and BS represent the SS, PS and BS of the proposed protocol. Here the place A represents the authentication information message, place B represents the authorization request, place C represents forward authorization request message, place D represents the authorization reply message and place E represents key acknowledgement message.

#### 4.3.1.2 Defining the Top-Level Substitution Transitions

We consider in detail the model of the SS, the PS and the BS. Fig 4.8 shows the CPN model of the SS. It contains three subnets: one models the subtask of SS initiating a protocol run in step 1, the second step sending authorization request and the third one receiving the authorization reply and acknowledging the reply.

In this page, SS sends the authentication information message through SendMSG1 place. The transition GenerateMsg2 generates the plain authorization message, using the four places for certificate, nonce, timestamp and other items such capabilities and SAID, and the transition Encrypt\_SK encrypts this plain message using the private key of SS received from place SK\_SS. Then SS sends encrypted authorization request message to PS through sendMSG2 place. The authorization reply is received at place sendMSG3 place which is decrypted using the public key of BS and then the validation of this

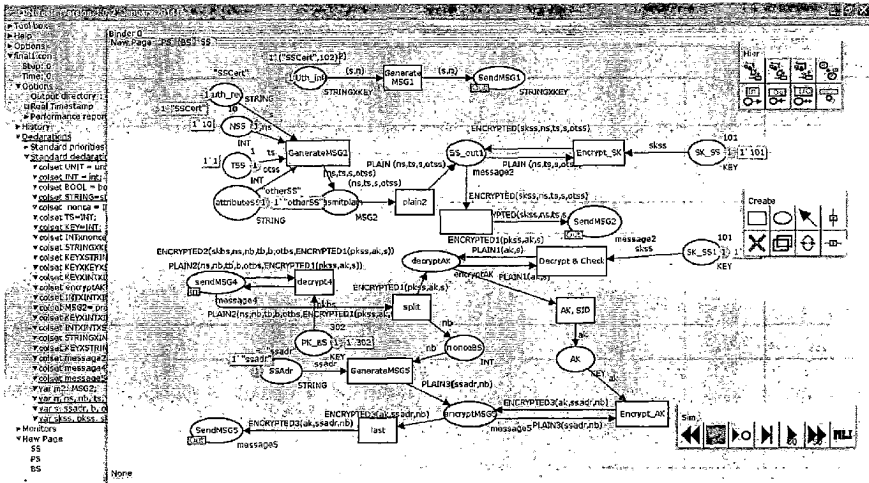


Fig 4.8: SS page

message is done. After validating the message the AK is conceived by decrypting the encrypted AK part within received message. The nonce and AK key are used by SS to generate message 4. The transition Encrypt\_AK shows the encryption of the message plain message generated at transition GenerateMSG4. The sendMSG4 place is the key acknowledgement message.

Fig 4.9 shows the CPN model of the PS. It contains three subparts: receiving the authentication information in step 1, the second step is receiving authorization request, and the third one is forwarding the authorization request after decrypting and validating the authorization request message.

In this page, place sendMSG1 is used to receive the authentication information and the certificate of SS is validated. The PS receives the authorization request through sendMSG2 place and decrypts this message using the public key of SS conceived after validating certificate from place PK\_SS. Then, the PS analyses the request for validation and then forwards the decrypted authorization request message to the BS through the place SendMSG3.

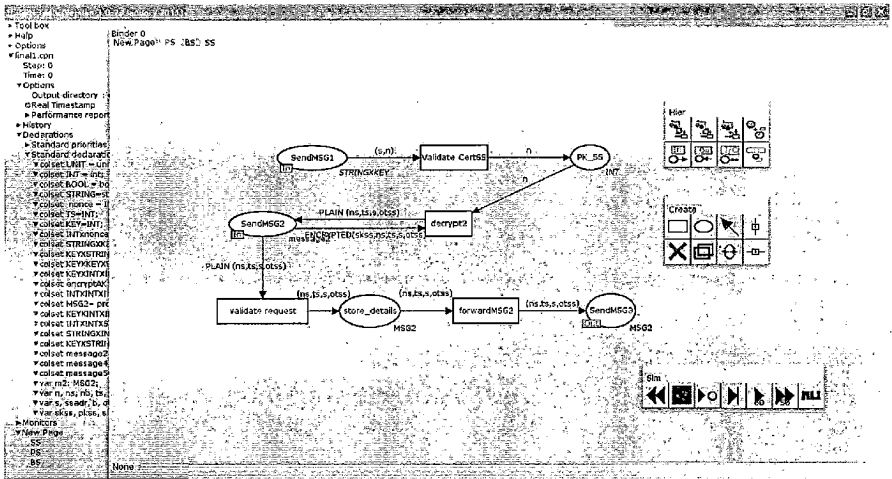


Fig 4.9: PS page

Fig. 4.10 shows the CPN model of the BS. It contains three subnets: receiving the authorization request forwarded by PS, the second one is sending the authorization reply to SS and the last one is receiving acknowledgement from SS.

In this page, the BS receives the authorization request through sendMSG3 place, it directly analyses the request and then according to the SS's capabilities it generates AK and encrypts it with public key of SS along with SSID at transition Encrypt(AK,SSID) using the key from PK\_SS place. Then the transition GenerateMSG3 generates the message incorporating the timestamp, nonce, certificate and other items such as AK\_lifetime, SAID etc. This message is encrypted at transition Encrypt\_BSSK using the private key of BS from BS\_SK place. This message is sent to SS through SendMSG4 place. Through place SendMSG5 the BS receives the key acknowledgement message and is decrypted at transition decryptACK using the AK key provided by GenerateAK place. If decrypted successfully the AK is exchanged successfully.

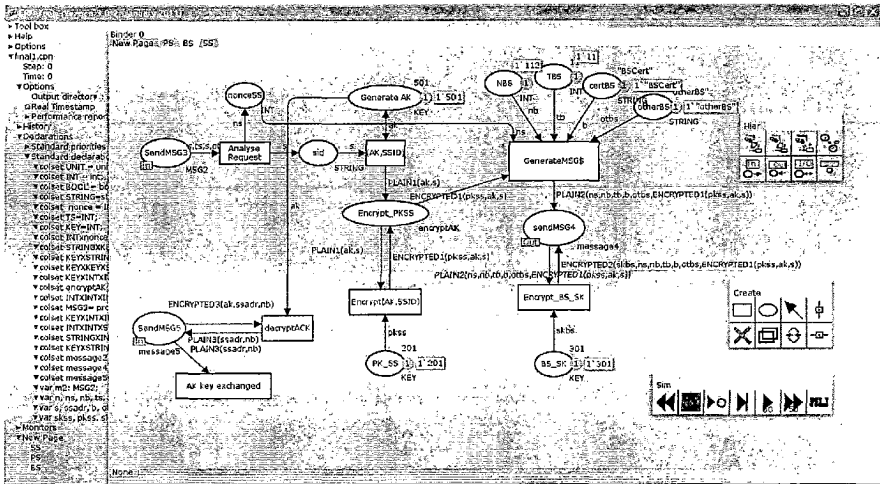


Fig. 4.10: BS page

### 4.3.2 Modeling the Proposed Protocol with Intruder

The proposed protocol is tested with the intruder that is just trying to replay the earlier authorization request message. This stored message cannot be modified because it is encrypted with the private key of SS. The replayed message is easily identified by the timestamp contained in this message. Thus, the model with intruder will not reach to the final state because the PS identifies that this is a replayed message.

In order to add the intruder to the model, one must extend the CPN ML declarations.

The final CPN ML declaration for proposed scheme is shown in following fig 4.11.

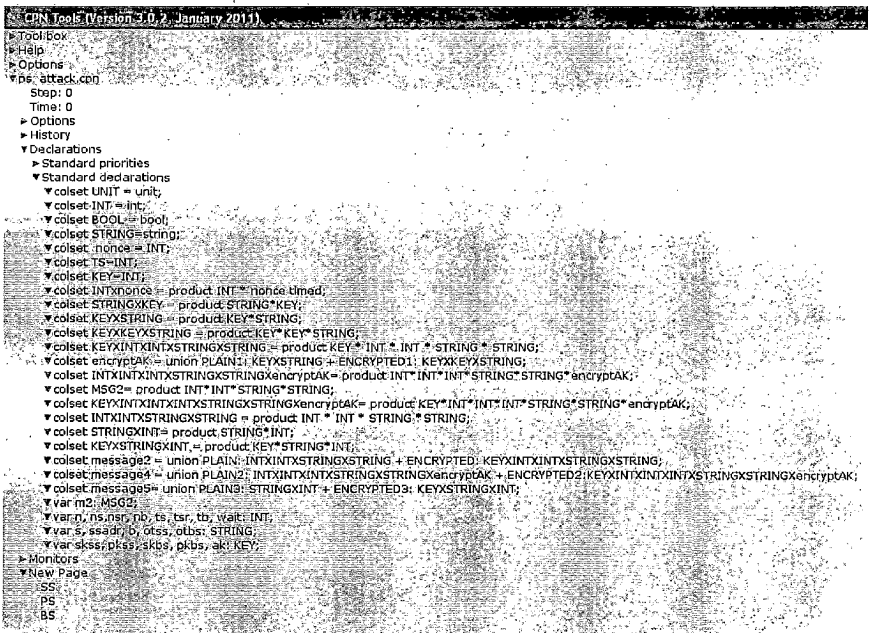


Fig. 4.11: Declarations used in the Proposed Model with an intruder

### 4.3.2.1 Top-Level Model with an Intruder

Fig 4.12 shows the top level model of the proposed protocol with an intruder. The substitution transition IR represents the intruder. The intruder is modeled as a separate entity that controls the communication channels between the protocol entities. Thus, it listens the authorization request message and replays them for future use. Then, it attempts to decrypt the encrypted portions of the intercepted messages. Finally, it attempts to modify the message contents, or even to generate new messages to replace the intercepted ones.

In this page, the transition IR represents the intruder which simply listens the message from place B and replays this message to PS through the place replay. The rest of the transitions and places are as it is in the proposed model without intruder.

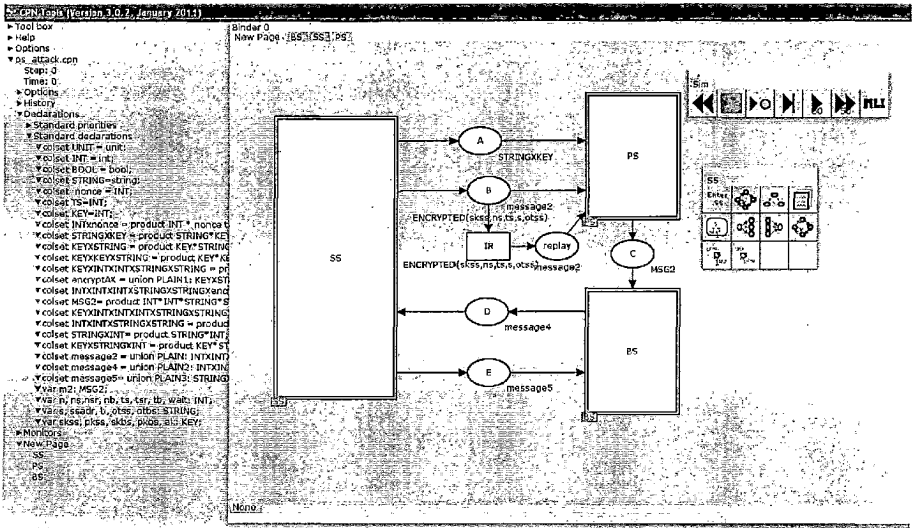


Fig 4.12: CPN Top-level Model for Proposed Protocol with Intruder

### 4.3.2.2. Replay Message Validation by PS

The following fig 4.13 shows the substitution page of PS where PS is validating the replayed authorization request message in proposed protocol.

Here the place sendMSG1 represents the receiver of the message replayed by the intruder. As the replayed message is a encrypted with the private key of SS, it is decrypted with the public key of the SS. After decrypting the message, PS validates the message, at transition named check, by checking the timestamp of the arrived message. If the timestamp of the replayed message is greater than the previous received message then it is validated as legitimate otherwise it is discarded here. Since, the message is the replayed one the timestamp will be less than the timestamp present in the record. Thus this model with intruder fails to reach the final state.

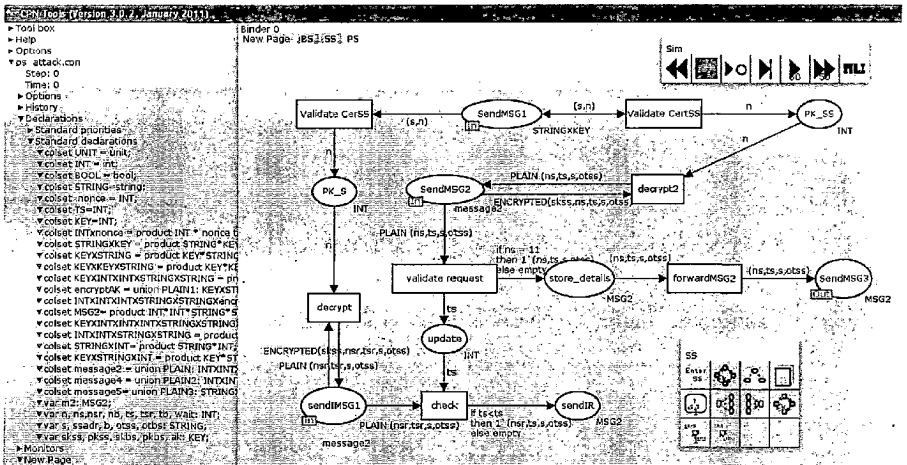


Fig. 4.13 Validation of replay message in PS page

## Chapter 5

### Results

---

#### 5.1 Formal Verification Parameters

The standard formal verification parameters derived from the state space analysis report are:

- Liveness Property: It assumes that if the authenticator sends Msg1, it will receive Msg4 definitely. It means the handshake executes successfully. If the execution reaches to the final state then liveness is satisfied otherwise not.
- Fairness: It determines whether the set of transition instances is impartial or fair.
- Deadlock: CPN can test whether the deadlock appears in the modeled system or not. Deadlock means that the protocol will unexpectedly terminate in the case of resource accessing conflict or unlimitedly waiting for acknowledge packets [28].
- Number of Nodes: The number of nodes is useful to observe the increased number of communicating nodes due to the intruder. With the introduction of intruder the number of increases.
- Number of Arcs: As the number of nodes increases the number of arcs also increases.

#### 5.2 Formal Verification using State Space Analysis Report

To analyze the state space the report is generated using the state space tool in CPN. The following shows the analysis of the report generated for each scenario.

##### 5.2.1 PKMv2 without Intruder

The following is the state space analysis report generated for the PKMv2 model without intruder.

CPN Tools state space report for the PKMv2 model:  
Statistics

-----  
State Space  
Nodes: 57  
Arcs: 98



Secs: 0  
Status: Full

### Boundedness Properties

---

#### Best Integer Bounds

	Upper	Lower
BS'BS_SK 1	1	0
BS'Encrypt_PKSS 1	1	0
BS'Generate_AK 1	1	0
BS'NBS 1	1	0
BS'PK_SS 1	1	0
BS'certBS 1	1	0
BS'msg3 1	1	0
BS'nonceSS 1	1	0
BS'otherBS 1	1	0
BS'sid 1	1	0
New_Page'A 1	1	0
New_Page'B 1	1	0
New_Page'C 1	1	0
New_Page'D 1	1	0
SS'AK 1	1	0
SS'Auth_info 1	1	0
SS'Auth_req 1	1	0
SS'NSS 1	1	0
SS'PK_BS 1	1	0
SS'SK_SS1 1	1	0
SS'SSAdr 1	1	0
SS'attributeSS 1	1	0
SS'decryptAK 1	1	0
SS'encryptMSG5 1	1	0
SS'nonceBS 1	1	0

#### Best Upper Multi-set Bounds

```
BS'BS_SK 1 1^301
BS'Encrypt_PKSS 1 1^PLAIN1((501,"SSCert"))++
1^ENCRYPTED1((201,501,"SSCert")).
BS'Generate_AK 1 1^501
BS'NBS 1 1^112
BS'PK_SS 1 1^201
BS'certBS 1 1^"BSCert"
BS'msg3 1
1^PLAIN2((10,112,"BSCert","otherBS",ENCRYPTED1((201,501,"SSCert"))))++
1^ENCRYPTED2((301,10,112,"BSCert","otherBS",ENCRYPTED1((201,501,"SSCert"
))))
BS'nonceSS 1 1^10
BS'otherBS 1 1^"otherBS"
BS'sid 1 1^"SSCert"
New_Page'A 1 1^"SSCert"
New_Page'B 1 1^((10,"SSCert","otherSS"))
New_Page'C 1
1^PLAIN2((10,112,"BSCert","otherBS",ENCRYPTED1((201,501,"SSCert"))))++
1^ENCRYPTED2((301,10,112,"BSCert","otherBS",ENCRYPTED1((201,501,"SSCert"
))))
```

```

New_Page'D 1      1`PLAIN3(("ssadr",112))++
1`ENCRYPTED3((501,"ssadr",112))
  SS'AK 1         1`501
  SS'Auth_info 1  1`"SSCert"
  SS'Auth_req 1   1`"SSCert"
  SS'NSS 1        1`10
  SS'PK_BS 1      1`302
  SS'SK_SS1 1     1`101
  SS'SSAdr 1      1`"ssadr"
  SS'attributeSS 1 1`"otherSS"
  SS'decryptAK 1  1`PLAIN1((501,"SSCert"))++
1`ENCRYPTED1((201,501,"SSCert"))
  SS'encryptMSG5 1 1`PLAIN3(("ssadr",112))++
1`ENCRYPTED3((501,"ssadr",112))
  SS'nonceBS 1   1`112

```

Best Lower Multi-set Bounds

```

BS'BS_SK 1      empty
BS'Encrypt_PKSS 1 empty
BS'Generate_AK 1 empty
BS'NBS 1        empty
BS'PK_SS 1      empty
BS'certBS 1     empty
BS'msg3 1       empty
BS'nonceSS 1    empty
BS'otherBS 1    empty
BS'sid 1         empty
New_Page'A 1    empty
New_Page'B 1    empty
New_Page'C 1    empty
New_Page'D 1    empty
SS'AK 1         empty
SS'Auth_info 1  empty
SS'Auth_req 1   empty
SS'NSS 1        empty
SS'PK_BS 1      empty
SS'SK_SS1 1     empty
SS'SSAdr 1      empty
SS'attributeSS 1 empty
SS'decryptAK 1  empty
SS'encryptMSG5 1 empty
SS'nonceBS 1   empty

```

Liveness Properties

---

Dead Markings  
[57]

Dead Transition Instances  
None

Live Transition Instances  
None

## Fairness Properties

---

No infinite occurrence sequences.

In the statistics, the state space shows that the number of nodes and arcs are 57 and 98 respectively. The secs variable shows that the time taken for the execution is zero. The status is full which shows that all the nodes in the state space are fully processed. Here the boundedness property is showing the maximum and minimum number of tokens held by each node.

In Liveness property, the dead marking is 57 which means that the node 57 has no further enabled binding elements i.e. the marking of the specified node is dead. There are no dead transition instances because every node occurred starting from the initial marking of the state space. In fairness property, it tells us that we do not have an infinite occurrence sequence unless each transition on the page continues to occur.

Thus the standard PKMv2 protocol satisfies all the desired properties of fairness, liveness and deadlock-free.

### 5.2.2 PKMv2 with Intruder

The following is the state space analysis report generated for the PKMv2 model with intruder.

CPN Tools state space report for PKMv2\_attacked model:

#### Statistics

---

State Space  
Nodes: 92  
Arcs: 165  
Secs: 0  
Status: Full

#### Boundedness Properties

---

#### Best Integer Bounds

	Upper	Lower
BS'BS_SK 1	1	0
BS'Encrypt_PKSS 1	1	0

BS'Generate_AK 1	1	0
BS'NBS 1	1	0
BS'PK_SS 1	1	0
BS'certBS 1	1	0
BS'msg3 1	1	0
BS'nonceSS 1	1	0
BS'otherBS 1	1	0
BS'sid 1	1	0
Intruder'Cert_I 1	1	0
Intruder'nonceI 1	1	0
Intruder'othersI 1	1	0
New_Page'A 1	1	0
New_Page'B 1	1	0
New_Page'C 1	1	0
New_Page'D 1	1	0
New_Page'IA 1	1	0
New_Page'IB 1	1	0
New_Page'IC 1	1	0
New_Page'ID 1	1	0
SS'AK 1	1	0
SS'Auth_info 1	1	0
SS'Auth_req 1	1	0
SS'NSS 1	1	0
SS'PK_BS 1	1	0
SS'SK_SS1 1	1	0
SS'SSAdr 1	1	0
SS'attributeSS 1	1	0
SS'decryptAK 1	1	0
SS'encryptMSG5 1	1	0
SS'nonceBS 1	1	0

Best Upper Multi-set Bounds

```

BS'BS_SK 1      1`301
BS'Encrypt_PKSS 1  1`PLAIN1((501,"SSCert"))++
1`ENCRYPTED1((201,501,"SSCert"))
  BS'Generate_AK 1  1`501
  BS'NBS 1          1`112
  BS'PK_SS 1       1`201
  BS'certBS 1     1`"BSCert"
  BS'msg3 1
1`PLAIN2((56,112,"BSCert","otherBS",ENCRYPTED1((201,501,"SSCert"))))++
1`ENCRYPTED2((301,56,112,"BSCert","otherBS",ENCRYPTED1((201,501,"SSCert"))))
  BS'nonceSS 1    1`56
  BS'otherBS 1   1`"otherBS"
  BS'sid 1       1`"SSCert"
  Intruder'Cert_I 1 1`"SSCert"
  Intruder'nonceI 1 1`56
  Intruder'othersI 1 1`"othersI"
  New_Page'A 1    1`"SSCert"
  New_Page'B 1    1`(10,"SSCert","otherSS")
  New_Page'C 1
1`ENCRYPTED2((301,56,112,"BSCert","otherBS",ENCRYPTED1((201,501,"SSCert"))))
  New_Page'D 1    1`ENCRYPTED3((501,"ssadr",112))
  New_Page'IA 1   1`"SSCert"
  New_Page'IB 1   1`(56,"SSCert","othersI")

```

```

    New_Page'IC 1
1`PLAIN2({56,112,"BSCert","otherBS",ENCRYPTED1({201,501,"SSCert"})})++)
1`ENCRYPTED2({301,56,112,"BSCert","otherBS",ENCRYPTED1({201,501,"SSCert"
}))})
    New_Page'ID 1      1`PLAIN3({"ssadr",112})++)
1`ENCRYPTED3({501,"ssadr",112})
    SS'AK 1      1`501
    SS'Auth_info 1      1`"SSCert"
    SS'Auth_req 1      1`"SSCert"
    SS'NSS 1      1`10
    SS'PK_BS 1      1`302
    SS'SK_SS1 1      1`101
    SS'SSAdr 1      1`"ssadr"
    SS'attributeSS 1      1`"otherSS"
    SS'decryptAK 1      1`PLAIN1({501,"SSCert"})++)
1`ENCRYPTED1({201,501,"SSCert"})
    SS'encryptMSG5 1      1`PLAIN3({"ssadr",112})++)
1`ENCRYPTED3({501,"ssadr",112})
    SS'nonceBS 1      1`112

```

#### Best Lower Multi-set Bounds

```

BS'BS_SK 1      empty
BS'Encrypt_PKSS 1      empty
BS'Generate_AK 1      empty
BS'NBS 1      empty
BS'PK_SS 1      empty
BS'certBS 1      empty
BS'msg3 1      empty
BS'nonceSS 1      empty
BS'otherBS 1      empty
BS'sid 1      empty
Intruder'Cert_I 1      empty
Intruder'nonceI 1      empty
Intruder'othersI 1      empty
New_Page'A 1      empty
New_Page'B 1      empty
New_Page'C 1      empty
New_Page'D 1      empty
New_Page'IA 1      empty
New_Page'IB 1      empty
New_Page'IC 1      empty
New_Page'ID 1      empty
SS'AK 1      empty
SS'Auth_info 1      empty
SS'Auth_req 1      empty
SS'NSS 1      empty
SS'PK_BS 1      empty
SS'SK_SS1 1      empty
SS'SSAdr 1      empty
SS'attributeSS 1      empty
SS'decryptAK 1      empty
SS'encryptMSG5 1      empty
SS'nonceBS 1      empty

```

Liveness Properties

-----  
Dead Markings  
[92]

Dead Transition Instances  
None

Live Transition Instances  
None

Fairness Properties  
-----

No infinite occurrence sequences.

In the statistics, the state space shows that the number of nodes and arcs are 92 and 165 respectively. The secs variable shows that the time taken for the execution is zero. The status is full which shows that all the nodes in the state space are fully processed. Here the boundedness property is showing the maximum and minimum number of tokens held by each node.

In Liveness property, the dead marking is 92 which means that the node 92 has no further enabled binding elements i.e. the marking of the specified node is dead. There are no dead transition instances because every node occurred starting from the initial marking of the state space. In fairness property, it tells us that we do not have any infinite occurrence sequence unless each transition on the page continues to occur.

Thus the standard PKMv2 protocol with intruder also satisfies all the desired properties of fairness, liveness and deadlock-free and the intruder goes undetected.

### 5.2.3 PS based Authentication Protocol without Intruder

The following is the state space analysis report generated for the proposed model with intruder.

CFN Tools state space report for the proposed\_scheme:

Statistics  
-----

State Space  
Nodes: 44  
Arcs: 55  
Secs: 0

Status: Full

Boundedness Properties

---

Best Integer Bounds

	Upper	Lower
BS'BS_SK 1	1	1
BS'Encrypt_PKSS 1	1	0
BS'Generate_AK 1	1	0
BS'NBS 1	1	0
BS'PK_SS 1	1	1
BS'TBS 1	1	0
BS'certBS 1	1	0
BS'nonceSS 1	1	0
BS'otherBS 1	1	0
BS'sid 1	1	0
New_Page'A 1	1	0
New_Page'B 1	1	0
New_Page'C 1	1	0
New_Page'D 1	1	0
New_Page'E 1	1	0
PS'PK_SS 1	1	0
SS'AK 1	1	0
SS'AUTH_info 1	1	0
SS'Auth_req 1	1	0
SS'NSS 1	1	0
SS'PK_BS 1	1	0
SS'SK_SS 1	1	0
SS'SK_SS1 1	1	0
SS'SSAdr 1	1	0
SS'SS_out1 1	1	0
SS'TSS 1	1	0
SS'attributeSS 1	1	0
SS'decryptAK 1	1	0
SS'encryptMSG5 1	1	0
SS'nonceBS 1	1	0
SS'trasmitplain 1	1	0

Best Upper Multi-set Bounds

BS'BS_SK 1	1^301
BS'Encrypt_PKSS 1	1^PLAIN1((501,"SSCert"))++
1^ENCRYPTED1((201,501,"SSCert"))	
BS'Generate_AK 1	1^501
BS'NBS 1	1^112
BS'PK_SS 1	1^201
BS'TBS 1	1^11
BS'certBS 1	1^"BSCert"
BS'nonceSS 1	1^10
BS'otherBS 1	1^"otherBS"
BS'sid 1	1^"SSCert"
New_Page'A 1	1^("SSCert",102)
New_Page'B 1	1^PLAIN((10,1,"SSCert","otherSS"))++
1^ENCRYPTED((101,10,1,"SSCert","otherSS"))	
New_Page'C 1	1^(10,1,"SSCert","otherSS")

```

New_Page'D 1
1`PLAIN2((10,112,11,"BSCert","otherBS",ENCRYPTED1((201,501,"SSCert"))))
++
1`ENCRYPTED2((301,10,112,11,"BSCert","otherBS",ENCRYPTED1((201,501,"SSC
ert"))))
New_Page'E 1      1`PLAIN3(("ssadr",112))++
1`ENCRYPTED3((501,"ssadr",112))
PS'PK_SS 1      1`102
SS'AK 1      1`501
SS'Auth_info 1  1`("SSCert",102)
SS'Auth_req 1  1`"SSCert"
SS'NSS 1      1`10
SS'PK_BS 1     1`302
SS'SK_SS 1     1`101
SS'SK_SS1 1    1`101
SS'SSAdr 1     1`"ssadr"
SS'SS_out1 1   1`PLAIN((10,1,"SSCert","otherSS"))++
1`ENCRYPTED((101,10,1,"SSCert","otherSS"))
SS'TSS 1      1`1
SS'attributeSS 1 1`"otherSS"
SS'decryptAK 1 1`PLAIN1((501,"SSCert"))++
1`ENCRYPTED1((201,501,"SSCert"))
SS'encryptMSG5 1 1`PLAIN3(("ssadr",112))++
1`ENCRYPTED3((501,"ssadr",112))
SS'nonceBS 1   1`112
SS'trasmitplain 1 1`(10,1,"SSCert","otherSS")

Best Lower Multi-set Bounds
BS'BS_SK 1     1`301
BS'Encrypt_PKSS 1 empty
BS'Generate_AK 1 empty
BS'NBS 1      empty
BS'PK_SS 1    1`201
BS'TBS 1      empty
BS'certBS 1   empty
BS'nonceSS 1  empty
BS'otherBS 1  empty
BS'sid 1      empty
New_Page'A 1  empty
New_Page'B 1  empty
New_Page'C 1  empty
New_Page'D 1  empty
New_Page'E 1  empty
PS'PK_SS 1    empty
SS'AK 1       empty
SS'Auth_info 1 empty
SS'Auth_req 1 empty
SS'NSS 1      empty
SS'PK_BS 1    empty
SS'SK_SS 1    empty
SS'SK_SS1 1   empty
SS'SSAdr 1    empty

```



SS'SS_out1 1	empty
SS'TSS 1	empty
SS'attributeSS 1	empty
SS'decryptAK 1	empty
SS'encryptMSG5 1	empty
SS'nonceBS 1	empty
SS'trasmitplain 1	empty

#### Liveness Properties

---

Dead Markings  
[44]

Dead Transition Instances  
None

Live Transition Instances  
None

#### Fairness Properties

---

No infinite occurrence sequences.

In the statistics, the state space shows that the number of nodes and arcs are 44 and 55 respectively. The secs variable shows that the time taken for the execution is zero. The status is full which shows that all the nodes in the state space are fully processed. Here the boundedness property is showing the maximum and minimum number of tokens held by each node.

In Liveness property, the dead marking is 44 which means that the node 44 has no further enabled binding elements i.e. the marking of the specified node is dead. There are no dead transition instances because every node occurred starting from the initial marking of the state space. In fairness property, it tells us that we do not have any infinite occurrence sequence unless each transition on the page continues to occur.

Thus the PS based authentication protocol without intruder satisfies all the desired properties of fairness, liveness and deadlock-free.

## 5.2.4 PS based Authentication Protocol with Intruder

The following is the state space analysis report generated for the proposed model with an intruder.

CPN Tools state space report for the proposed\_attack:  
Statistics

---

State Space  
Nodes: 20  
Arcs: 27  
Secs: 0  
Status: Full

Boundedness Properties

---

Best Integer Bounds

	Upper	Lower
BS'BS_SK 1	1	1
BS'Encrypt_PKSS 1	0	0
BS'Generate_AK 1	1	1
BS'NBS 1	1	1
BS'PK_SS 1	1	1
BS'TBS 1	1	1
BS'certBS 1	1	1
BS'nonceSS 1	0	0
BS'otherBS 1	1	1
BS'sid 1	0	0
New_Page'A 1	1	0
New_Page'B 1	1	0
New_Page'C 1	0	0
New_Page'D 1	0	0
New_Page'E 1	0	0
New_Page'replay 1	1	0
PS'PK_SS 1	1	0
PS'SendMSG2 1	0	0
PS'update 1	1	0
SS'AK 1	0	0
SS'Auth_info 1	1	0
SS'Auth_req 1	1	0
SS'NSS 1	1	0
SS'PK_BS 1	1	1
SS'SK_SS 1	1	0
SS'SK_SS1 1	1	1
SS'SSAdr 1	1	1
SS'SS_out1 1	1	0
SS'TSS 1	1	0
SS'attributeSS 1	1	0
SS'decryptAK 1	0	0
SS'encryptMSG5 1	0	0
SS'nonceBS 1	0	0
SS'transmitplain 1	1	0

Best Upper Multi-set Bounds

```

BS'BS_SK 1 1`301
BS'Encrypt_PKSS 1 empty
BS'Generate_AK 1 1`501
BS'NBS 1 1`112
BS'PK_SS 1 1`201
BS'TBS 1 1`11
BS'certBS 1 1`"BSCert"
BS'nonceSS 1 empty
BS'otherBS 1 1`"otherBS"
BS'sid 1 empty
New_Page'A 1 1`("SSCert",102)
New_Page'B 1 1`ENCRYPTED((101,10,1,"SSCert","otherSS"))
New_Page'C 1 empty
New_Page'D 1 empty
New_Page'E 1 empty
New_Page'replay 1 1`PLAIN((10,1,"SSCert","otherSS"))+
1`ENCRYPTED((101,10,1,"SSCert","otherSS"))
PS'PK_SS 1 1`102
PS'SendMSG2 1 empty
PS'update 1 1`15
SS'AK 1 empty
SS'Auth_info 1 1`("SSCert",102)
SS'Auth_req 1 1`"SSCert"
SS'NSS 1 1`10
SS'PK_BS 1 1`302
SS'SK_SS 1 1`101
SS'SK_SS1 1 1`101
SS'SSAdr 1 1`"ssadr"
SS'SS_out1 1 1`PLAIN((10,1,"SSCert","otherSS"))+
1`ENCRYPTED((101,10,1,"SSCert","otherSS"))
SS'TSS 1 1`1
SS'attributeSS 1 1`"otherSS"
SS'decryptAK 1 empty
SS'encryptMSG5 1 empty
SS'nonceBS 1 empty
SS'trasmitplain 1 1`(10,1,"SSCert","otherSS")

```

Best Lower Multi-set Bounds

```

BS'BS_SK 1 1`301
BS'Encrypt_PKSS 1 empty
BS'Generate_AK 1 1`501
BS'NBS 1 1`112
BS'PK_SS 1 1`201
BS'TBS 1 1`11
BS'certBS 1 1`"BSCert"
BS'nonceSS 1 empty
BS'otherBS 1 1`"otherBS"
BS'sid 1 empty
New_Page'A 1 empty
New_Page'B 1 empty
New_Page'C 1 empty
New_Page'D 1 empty
New_Page'E 1 empty
New_Page'replay 1 empty
PS'PK_SS 1 empty
PS'SendMSG2 1 empty
PS'update 1 empty

```

SS'AK 1	empty
SS'Auth_info 1	empty
SS'Auth_req 1	empty
SS'NSS 1	empty
SS'PK_BS 1	1`302
SS'SK_SS 1	empty
SS'SK_SS1 1	1`101
SS'SSAdr 1	1`"ssadr"
SS'SS_out1 1	empty
SS'TSS 1	empty
SS'attributeSS 1	empty
SS'decryptAK 1	empty
SS'encryptMSG5 1	empty
SS'nonceBS 1	empty
SS'trasmitplain 1	empty

#### Liveness Properties

---

Dead Markings  
[20]

#### Dead Transition Instances

BS'AK\_SSID 1  
 BS'AK\_key\_exchanged 1  
 BS'Analyse\_Request 1  
 BS'Encrypt 1  
 BS'Encrypt\_BS\_SK 1  
 BS'GenerateMSG 1  
 BS'decryptACK 1  
 PS'decrypt2 1  
 PS'validate\_request 1  
 SS'AK 1  
 SS'Decrypt 1  
 SS'Encrypt\_AK 1  
 SS'GenerateMSG5 1  
 SS'decrypt4 1  
 SS'last 1  
 SS'split 1

Live Transition Instances  
None

#### Fairness Properties

---

No infinite occurrence sequences.

In the statistics, the state space shows that the number of nodes and arcs are 20 and 27 respectively. Here the increase in the number of nodes and arcs is seen, because the nodes and arcs of the intruder are also included. The secs variable shows that the time taken for the execution is zero. The status is full which shows that all the nodes in the state space

are fully processed. Here the boundedness property is showing the maximum and minimum number of tokens held by each node.

In Liveness property, the dead marking is 20 which means that the node 20 has no further enabled binding elements i.e. the marking of the specified node is dead. There are no dead transition instances because every node occurred starting from the initial marking of the state space. In fairness property, it tells us that we do not have any infinite occurrence sequence unless each transition on the page continues to occur.

Thus the proposed 3-way handshake protocol with an intruder also does not satisfy the desired properties of liveness and deadlock-free.

### 5.3 Comparative Analysis

The report generated for the four models showing the different desired properties can be consolidated into a single table as shown below (Table 5.1).

**Table 5.1: Analysis of State Spaces**

Approaches	Fairness	Deadlock	Liveness	No. of nodes	No. of arcs
PKMv2 without intruder	yes	no	yes	57	98
PKMv2 with intruder	yes	no	yes	92	165
Proposed Protocol without intruder	yes	no	yes	44	55
Proposed Protocol with intruder	yes	yes	no	20	27

From the above table it can observe that the even after adding the attacker to PKMv2 model the intruder reaches to the final state successfully. There is no deadlock and the liveness property is satisfied. But the number of arcs and nodes are increased almost to the twice of model without intruder. So the only way to detect the intrusion is by noticing the increased number of nodes and arcs in the state space results. This is because the authorization request message sent by the SS is open to everyone and can be modified easily. Thus without the knowledge of SS and BS the intruder can easily compromise the privacy of the communication channel. Hence the intruder goes undetected in the standard PKMv2 protocol.

In the case of the proposed protocol model there is no deadlock and the fairness and liveness properties are satisfied. With introduction of PS in proposed protocol the number of nodes and arcs are increased as compared to standard PKMv2 protocol. As observed from the above table the modeling of proposed model with intruder does not increase the number of nodes and arcs because the replayed message was unable pass through the validation at PS page. So there is unexpected termination which results to deadlock and the liveness property is not satisfied. Hence the intruder fails to compromise the network and reach the final state.

## Chapter 6

# Conclusions and Future Work

---

### 6.1 Conclusions

The standard PKMv2 authentication protocol is vulnerable because the messages exchanged between the SS and BS are not secured. To solve this issue a proxy BS based authentication protocol is proposed, which is efficient in tackling the various security threats such as replay attack, DoS attack, interleaving attack and downgrade attack. In the proposed authentication protocol using proxy base station, the PS performs the task of validating the authorization request messages and relaxes the BS so that the BS can efficiently provide services to legitimate SSs. Thus, DoS attack is successfully tackled.

The proposed authentication protocol is modelled and tested on CPN tool. The state space analysis report shows that the proposed protocol satisfies the desired properties of liveness, fairness and deadlock-free. The attacks that went undetected by the PKMv2 protocol are easily detected and discarded by our proposed protocol. The proposed protocol is more secure against the intruder than the standard PKMv2 protocol. The numbers of messages exchanged are almost same because the message 3 in proposed protocol is openly communicated through secured network. Moreover, in the proposed scheme the BS station can provide better quality of service as compared to the previous one because the task of authorization is distributed among PS and BS. Hence our proposed protocol is more robust against attacks.

## 6.2 Future Work

Designing more secure mechanisms against additional attacks without causing much overhead is a challenging task for further research. There is significant room for improving the security in the proposed protocol. The possible improvements in the future are listed as below:

- real-time implementation of the proposed protocol has to be done
- authentication protocol can be more efficient if we use ECC cryptography which is more efficient than RSA cryptography.
- mechanism to maintain time synchronization throughout the WiMAX environment.
- standardize the threshold for the number of authorization request to be validated by PS.



## REFERENCES

- [1] WiMAX Forum, "Fixed, nomadic, portable and mobile applications for 802.16-2004 and 802.16e WiMAX networks," November 2005.
- [2] IEEE 802.16-2004, "IEEE standard for Local and Metropolitan Area networks-Part 16: Air Interface for Fixed Broadband Wireless Access Systems," IEEE Press, 2004.
- [3] IEEE 802.16-2005, "IEEE standard for Local and Metropolitan Area networks-Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems," IEEE Press, 2005.
- [4] S. Ahson and M. Ilyas, *WiMAX: Standards and Security*: CRC Press, Inc. Boca Raton, FL, USA, 2008.
- [5] J. Hasan, "Security Issues of IEEE 802.16 (WiMAX)," in *School of Computer and Information Science, Edith Cowan University, Australia*, 2006, pp.1-8.
- [6] E. Eren, "WiMAX Security Architecture - Analysis and Assessment," in *4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, 2007, pp. 673-677.
- [7] T. Panagiotis and G. George, "WiFi and WiMAX Secure Deployments," *Journal of Computer Systems, Networks, and Communications*, 2010, vol. 2010, pp.37-64.
- [8] W. Stallings, *Cryptography and Network Security*, Pearson Education, 4th edition, 2006.
- [9] M. Habib and M. Ahmad, "A Review of Some Security Aspects of WiMAX and Converged Network," in *Proceedings of Second International Conference on Communication Software and Networks*: IEEE, 2010, pp. 372-376.
- [10] C. Luo, "A Simple Encryption Scheme Based on WiMAX," in *Proceedings of International Conference on EBusiness and Information System Security*, 2009, pp. 1-4.
- [11] A. Altaf, M. Y. Javed, A. Ahmed, "Security Enhancements for Privacy and Key Management Protocol in IEEE 802.16e-2005", in *Proceedings of 9th ACIS International Conference on software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2008, pp. 335-339.

- [12] Wenbo Mao, *Modern Cryptography: Theory and Practice*, Pearson Education, Prentice Hall PTR, 2004.
- [13] S. Xu, M. Matthews, and C. T. Huang, "Security issues in privacy and key management protocols of IEEE 802.16," in *Proceedings of 44th annual Southeast regional conference*, Melbourne, Florida, 2006, pp. 113-118.
- [14] W. Mao, *Modern Cryptography: Theory and Practice*. Pearson Education, Prentice Hall PTR, 2004.
- [15] Hashmi, R.M.; Siddiqui, A.M.; Jabeen, M.; Shehzad, K.; Zubair, A.; Alimgeer, K.S.; , "Improved Secure Network Authentication Protocol (ISNAP) for IEEE 802.16," in *Information and Communication Technologies, Doha, Qatar*, 2009, pp.101-105.
- [16] A. Altaf, R. Sirhindi, A. Ahmed; "A Novel Approach against DoS Attacks in WiMAX Authentication using Visual Cryptography", in *Proceedings of 2<sup>nd</sup> International Conference on Emerging Security Information*,2008, pp 238-242.
- [17] H. Dong and W. Yu Yan, "Secure Authentication on WiMAX with Neural Cryptography," in *Proceedings of International Conference on Information Security and Assurance*, 2008, pp. 366-369.
- [18] Y. Yang and R. Li, "Toward Winax Security," in *Proceedings of Computational Intelligence and Software Engineering*, Wuhan, China, 2009, pp. 1-5.
- [19] B. Sikkens, "Security Issues and Proposed Solutions Concerning Authentication and Authorization for WiMAX," in *Proceedings of 8<sup>th</sup> Twente Student Conference on IT, Enschede*, January 25th, 2008, pp. 1-7.
- [20] F.Liu and L.Lu, "A WPKI-based Security Mechanism for IEEE 802.16e," in *Proceedings of Int'l Conference on WirelessComm., Networking and Mobile Computing*, 2006, pp. 1-4.
- [21] M. Habib, T. Mehmood, F. Ullah, and M. Ibrahim, "Performance of WiMAX Security Algorithm (The Comparative Study of RSA Encryption Algorithm with ECC Encryption Algorithm)," in *Proceedings of International Conference on Computer Technology and Development. ICCTD '09*, Kota Kinabalu, Malaysia, 2009, pp. 108-112.

- [22] T. Shon and W. Choi, "An analysis of mobile WiMAX security: vulnerabilities and solutions," in *Proceedings of the 1st international conference on Network-based information systems*, Regensburg, Germany, 2007, pp. 88-97.
- [23] T. Han, N. Zhang, K. Liu, B. Tang, and Y. a. L. , "Analysis of Mobile WiMAX Security: Vulnerabilities and Solutions," in *Proceedings of International Conference on Mobile Ad Hoc and Sensor Systems, 5th IEEE Atlanta, GA, 2008*, pp. 828 - 833.
- [24] F. Tshering and A. Sardana. "A Review of Privacy and Key Management Protocol in IEEE 802.16e," in *International Journal of Computer Applications, Published by Foundation of Computer Science*, April 2011, vol.20 (2), pp.25–31.
- [25] F. Tshering and A. Sardana. "A Proxy Base Station based Authentication Protocol for IEEE 802.16e," in *Proceedings of IEEE ICRTIT 2011 Conference, June 3-5, 2011*.
- [26] AIS group, Eindhoven University of Technology, The Netherlands, 2011. <http://cpntools.org/>
- [27] S. Aly, K. Mustafa, "Protocol verification and analysis using colored Petri nets," *Technical Report TR-04-003, The School of Computer Science, Telecommunication and Informations*, August 2004.
- [28] B. Nieh and S. Tavares. "Modelling and analyzing cryptographic protocols using Petri nets," in *Advances in Cryptology-ASIACRYPT '92, Springer*, 1992, vol. 718 pp 275–295
- [29] I. Al-Azzoni, D. G. Down, and R. Khedri, "Modeling and verification of cryptographic protocols using coloured petri nets and design/CPN," *In proceedings of MOMPES'05*, 2005, pp. 1-28.
- [30] Drespe, Wiebke, "Security Analysis of the Secure Authentication Protocol by Means of Coloured Petri Nets," *In proceedings Of Dittmann, Jana, (ed.) Communications and Multimedia Security. Lecture Notes in Computer Science, 3677. Springer, Berlin*, 2005, pp. 230-239.

## LIST OF PUBLICATIONS

---

- [1] **Fuden Tshering** and Anjali Sardana. "A Review of Privacy and Key Management Protocol in IEEE 802.16e," *International Journal of Computer Applications*, vol. 20 (2), pp.25–31, April 2011. Published By Foundation of Computer Science.
- [2] **Fuden Tshering** and Anjali Sardana. "A Proxy Base Station based Authentication Protocol for IEEE 802.16e," *IEEE ICRTIT 2011 Conference Proceedings*, June3-5, 2011 (accepted and registered)