STATUS OF THESIS

Title of thesis	Authentication Mechanism for Ad Hoc Wireless Local Area Network

I, MUHAMMAD AGNI CATUR BHAKTI

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP.
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

☐ Confidential

☑ Non-confidential

If this thesis is confidential, please state the reason:

_____ - _____

The contents of the thesis will remain confidential for ___ - ___ years.

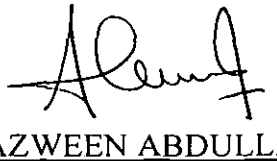
Remarks on disclosure:

_____ - _____

Endorsed by

MUHAMMAD AGNI CATUR BHAKTI
Komplek MABAD II No. 40
RT. 002 / 011, Srengseng Sawah
Jagakarsa, Jakarta, Indonesia

Date: 27/06/08



AZWEEN ABDULLAH
Universiti Teknologi
PETRONAS
Malaysia

Date: 30/6/08

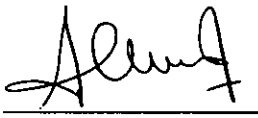
UNIVERSITI TEKNOLOGI PETRONAS

Approval by Supervisor (s)

The undersigned certify that they have read, and recommend to The Postgraduate Studies Programme for acceptance, a thesis entitled "**Authentication Mechanism for Ad Hoc Wireless Local Area Network**" submitted by **(Muhammad Agni Catur Bhakti)** for the fulfillment of the requirements for the degree of Master of Science in Information Technology.

_____

Date

Signature : 

Main Supervisor : Dr Azween Bin Abdullah
Senior Lecturer
Information Technology/Information Systems
Universiti Teknologi PETRONAS
31750 Tronoh
Perak Darul Ridzuan

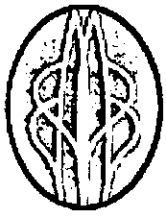
Date : 30/6/08

Co-Supervisor : _____

TITLE PAGE

UNIVERSITI TEKNOLOGI PETRONAS

Authentication Mechanism for Ad Hoc Wireless Local Area Network



By

Muhammad Agni Catur Bhakti

A THESIS

SUBMITTED TO THE POSTGRADUATE STUDIES PROGRAMME

AS A REQUIREMENT FOR THE

DEGREE OF MASTER OF SCIENCE

INFORMATION TECHNOLOGY

BANDAR SERI ISKANDAR,

PERAK

JUNE, 2008

DECLARATION

I hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledge. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Signature : 

Name : MUHAMMAD AGNI CATUR BHAKTI

Date : 27/06/08

ABSTRACT

Wireless networks have grown rapidly over the last decade and they have been deployed in numerous applications due to their advantages over wired networks, specifically for its mobility and convenience. However, due to its wireless nature, some security issues in wireless network need to be addressed, such as unauthorized or rogue wireless devices which are relatively easy to connect to the network because they do not need any physical access. These issues might prevent further acceptance and adoption of wireless network technology.

One of the solutions to overcome the wireless network security is the 802.1X specification. It is a mechanism for port-based network access control, which based on Extensible Authentication Protocol (EAP). It is an authentication framework that can support multiple authentication methods. This research is looking into the possibility of using EAP as a generic authentication mechanism in ad hoc wireless local area networks. One promising advantage of using EAP-based authentication mechanism in a network is its interoperability with other types of networks since EAP is already a platform for various authentication mechanisms.

This thesis studies and explores the feasibility of using EAP in ad hoc wireless local area network and then proposes a mechanism to implement EAP in ad hoc wireless local area network based on EAP multiplexing model. This thesis also proposes an extension to EAP, a mechanism to select a suitable EAP method out of a set of EAP methods to be used in EAP authentication process in heterogeneous mobile devices environment, where the network consists of different types of nodes / devices with different specifications and capabilities, and each node may support different type of EAP authentication method.

Toward the end of this thesis, formal specification and verification of the proposed authentication mechanism are derived and strong final beliefs are obtained. Furthermore, node architecture that can be used in simulation of EAP authentication is designed and the EAP method selection mechanism is simulated.

ABSTRAK

Teknologi rangkaian tanpa wayar telah berkembang pesat dan pantas kebelakangan ini. Antara faktornya adalah daya saingnya yang tinggi berbanding Teknologi rangkaian berwayar dari segi kemudahalihanya dan tahap kesiagaan yang tinggi. Walaubagaimanapun di dalam dunia tanpa wayar ini, isu seperti keselamatan haruslah di titik beratkan, lebih-lebih lagi bagi menangani masalah peralatan rangkaian tanpa wayar yang mudah disambungkan ke mana-mana kumpulan rangkaian tanpa kebenaran atau secara haram. Oleh itu, ini pastinya boleh menimbulkan kemusykilan kepada masyarakat dalam menerima secara baik kegunaan Teknologi rangkaian tanpa wayar ini, justeru boleh menghalang Teknologi ini daripada berkembang maju.

Salah satu daripada cara untuk mengatasi kekurangan dari segi keselamatan Teknologi rangkaian tanpa wayar ini ialah dengan pengenalan satu standard spesifikasi IEEE 802.1X. Mekanisma standard ini adalah dengan mengawal kemasukan data melalui laluan-laluan tertentu atau dengan istilah "port-based network access control" yang berdasarkan "Extensible Authentication Protocol" atau dalam istilah singkatnya EAP. Ia merupakan satu rangka kerja yang boleh menampung pelbagai kaedah pengesahan rangkaian. Oleh yang demikian adalah mungkin EAP juga mampu diadaptasikan sebagai mekanisma pengesahan pelbagai rangkaian seperti rangkaian tanpa wayar setempat yang sepontan atau dengan istilah "ad hoc wireless local area network". Disebabkan EAP telah digunakan sebagai asas kepada pelbagai mekanisma pengesahan, maka ianya dilihat sebagai sungguh berguna untuk diaplikasikan kepada sesuatu rangkaian yang mampu menghubungkannya dengan pelbagai jenis teknologi rangkaian yang lain.

Tesis ini mengkaji dan menghurai kebolehgunaan EAP di dalam rangkaian tanpa wayar setempat secara sepontan dan mencadangkan pengaplikasian "EAP multiplexing model" sebagai mekanisma pengesahan. Ia juga mencadangkan mekanisma tambahan kepada EAP itu sendiri bagi memilih kaedah yang paling sesuai bagi digunakan di dalam proses pengesahan peralatan mudah alih heterogenus, di mana setup rangkaian itu menghubungkan pelbagai jenis peralatan dengan spesifikasi

dan kebolehan yang berbeza-beza. Setiap satu peralatan yang dikenali sebagai "node" ini mampu pula menampung kaedah pengesahan EAP yang berbeza diantara satu sama yang lain.

Di akhir tesis ini, spesifikasi dan pengesahan formal mekanisma pengesahan yang dicadangkan diperbincangkan dan kesimpulan yang ampuh diketengahkan. Oleh yang demikian, reka bentuk "node" yang boleh digunakan dalam simulasi pengesahan EAP diketengahkan dan mekanisma pemilihan kaedah EAP disimulasikan.

ACKNOWLEDGEMENTS

First of all, I would like to say Alhamdulillah praise Allah The Most Gracious and The Most Merciful, for His blessings throughout my life and enabling me to finish this thesis. To my family – my wife, my daughter, my parents, my brothers and sisters – thank you for always support me in my personal or academic endeavors. Their relentless encouragement has enabled me to persevere even in troubled times.

I would like to thank my supervisor Dr. Azween Abdullah and my co-supervisor Low Tang Jung for their supervision and guidance throughout my study and writing up this thesis. I greatly appreciate their encouragement to my work especially in time of lacking self confidence and self discipline.

I also want to thank to Dr. Ahmad Kamil Mahmood for his support and leading as head of Computer and Information Sciences Department Universiti Teknologi PETRONAS. My thanks also go to the lecturers in the department for their sharing of knowledge, discussion, and reviews.

My sincere thanks to all the administrative staffs: Mr. Azful, Mr. Fadzli, Mr. Fadil Ariff, Kak Lia, Kak Norma, Kak Aida, and the rest of the staffs for their assistance during my time in UTP. Last but not least, I would also like to express my appreciation to my friends and colleagues. Thank you for all the experiences, knowledge, and friendship during my stay here at UTP.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization, Accounting
AES	Advanced Encryption Standard
AKA	Authentication and Key Agreement
AMPS	Advanced Mobile Phone Systems
AODV	Ad-hoc On-demand Distance Vector
API	Application Programming Interface
APSS	Access Point Security Service
ARP	Address Resolution Protocol
ASF	Alerting Standards Forum
BAN (Logic)	Burrows, Abadi, Needham
BSS	Basic Service Set
CA	Certificate Authority
CBC	Cipher Block Chaining
CCMP	Counter mode CBC-MAC Protocol
CDMA	Code Division Multiple Access
CHAP	Challenge Handshake Authentication Protocol
CRC	Cyclic Redundancy Check
DSSS	Direct Sequence Spread Spectrum
EAP	Extensible Authentication Protocol
EAPOL	EAP Over LAN
EDGE	Enhanced Data rates for GSM Evolution
FHSS	Frequency Hopping Spread Spectrum
GloMoSim	Global Mobile information systems Simulation library
GPL	General Public License
GPRS	General Packet Radio Service
GSM	Global System for Mobile communication
GUI	Graphical User Interface
HSDPA	High Speed Downlink Packet Access
IAPP	Inter-Access Point Protocol
ICMP	Internet Control Message Protocol

ICV	Integrity Check Value
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IV	Initialization Vector
LEAP	Lightweight EAP
MAC	Medium Access Control
	Message Authentication Code
MD5	Message Digest 5
MIC	Message Integrity Check
MITM	Man-In-The-Middle
NIC	Network Interface Card
NS-2	Network Simulator – 2
OFDM	Orthogonal Frequency Division Multiplexing
OMNet++	Objective Modular Network test bed in C++
OSI	Open System Interconnection
PAE	Port Access Entity
PANA	Protocol for carrying Authentication for Network Access
PAP	Password Authentication Protocol
PDA	Personal Digital Assistant
PDU	Protocol Data Unit
PEAP	Protected EAP
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
RADIUS	Remote Authentication Dial-In User Service
RC4	Ron's Code 4 / Rivest Cipher 4
RFC	Request For Comments
RSNA	Robust Security Network Association
SIM	Subscriber Identity Module
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TKIP	Temporal Key Integrity Protocol
TLS	Transport Layer Security

TTLS	Tunneled Transport Layer Security
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunication System
USIM	UMTS SIM
WCDMA	Wide CDMA
WEP	Wired Equivalent Privacy
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WMN	Wireless Mesh Network
WPA	Wireless Protected Access
WPAN	Wireless Personal Area Network
WWAN	Wireless Wide Area Network

CHAPTER ONE : INTRODUCTION

In this chapter, an introduction to the conducted research is presented. Firstly, an overview of authentication and ad hoc wireless networks is given. Thereafter, an overview of the issue and problem in ad hoc wireless networks and the objectives of the research are given. An outline of the remaining chapters of this thesis is also included here.

1.1 Introduction

For the past few decade, the popularity and the use of wireless network has grown rapidly. Reports in [Horrigan, 2007] and [Horrigan, 2008] show significant growth of wireless connectivity to the internet and mobile access using wireless devices (i.e. laptop, cell phone, wireless-enabled personal digital assistant). This growth is mainly due to the advantages of wireless network over the wired network, such as convenience, mobility, and rapid deployment. However, wireless network also introduces new security issues, such as unauthorized wireless devices (rogue devices) access. Therefore providing a way for the communicating parties in wireless network to validate each other's identity, i.e. authentication, becomes a crucial and important matter.

Ad hoc wireless network is one of the emerging wireless network technologies nowadays and an object of on-going research works. Ad hoc wireless network is a collection of two or more devices equipped with wireless communications and networking capability [Toh, 2002]. It is a type of wireless network that could exist without the support of fixed infrastructures such as access points. For that, ad hoc network is called infrastructure-less wireless network. Therefore the network nodes themselves support the network functionality. Ad hoc nodes can communicate with other nodes immediately within their radio range or other nodes outside their radio range in multi-hop fashion towards the destinations. Typically, ad hoc network will form wireless personal area network or wireless local area network.

Ad hoc wireless network is self-configurable and self-organizing. It can be formed or de-formed on-the-fly without or with very few system administration. Ad hoc nodes or devices should be able to perform the necessary actions to allow communications with other nodes and to do networking functionalities and services, such as packet forwarding or routing.

These features make ad hoc networks very attractive in applications where fixed infrastructures are not available or not adequate, and self-configuration of the network is necessary. Some possible applications of ad hoc network include military monitoring and communications in battlefield, environmental and weather monitoring and information gathering in remote or hazardous or dangerous area using a type of ad hoc wireless network, i.e. wireless sensor network. Figure 1-1 shows a typical multi-hop wireless sensor network topology.

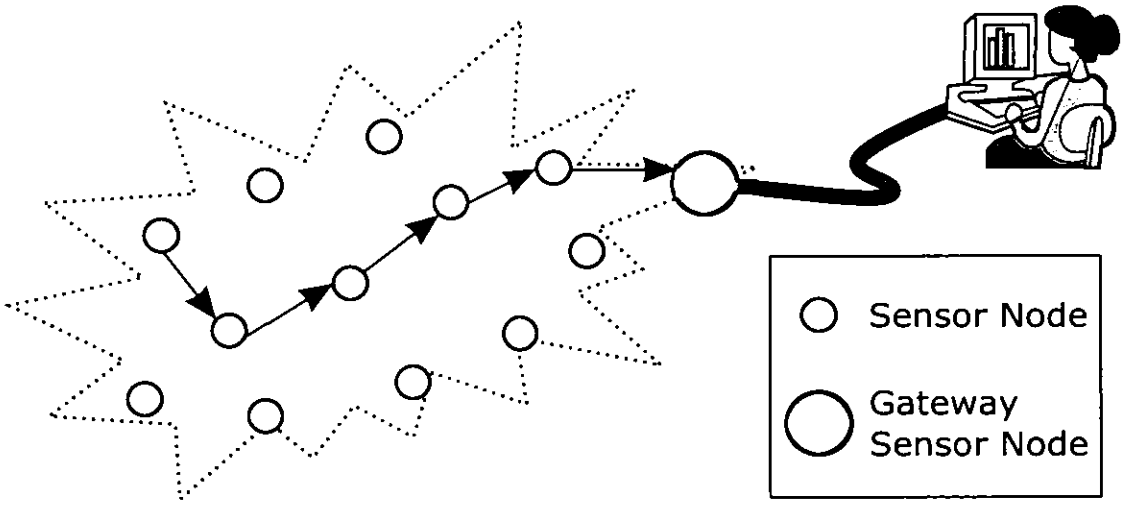


Figure 1-1: Multi-hop wireless sensor network

1.2 Problem Statement

In ad hoc wireless networks, security is one of the most important concerns because it is more vulnerable to various kinds of attacks compared to wired network or infrastructure-based wireless network. However, many challenges restrict the use of conventional security mechanisms in ad hoc network. The wireless channel itself suffers from poor protection and is susceptible to attacks. Most ad hoc routing

protocols assume that all nodes are cooperative in nature [Lou & Fang, 2004] which is not always the case. The lack of fixed infrastructure and centralized system in ad hoc wireless network add up the challenges.

As ad hoc wireless network nodes become mobile, the network associations are dynamically created and torn down. With the nodes joining and leaving the network, it is imperative to distinguish which nodes are trustworthy and which nodes are hostile. A hostile node could disrupt the operation of ad hoc network since each and every node in ad hoc network is required to assist the network operation.

Nodes in ad hoc wireless network are dynamically connected and disconnected, thus authentication becomes very important to ensure the identity of trustworthy nodes. In conventional internet security, authentication is usually carried out using challenge process between two parties; or using a trusted third party, such as certificate authority (CA), for the verification with the understanding that both parties share secrets with the CA, then the challenge process will be carried out between the parties with the CA. However, in ad hoc network, a fixed and centralized party to be trusted may not be available at all time.

One of the solutions to overcome the limitation of wireless network security and providing authentication is the IEEE 802.1X [IEEE802.1X, 2004] specification, a mechanism for port-based network access control, which is based on Extensible Authentication Protocol (EAP) [Aboba et al., 2004]. It is an authentication framework that can support multiple authentication methods. EAP can run over many types of data-link layers and it is relatively flexible in its implementation in the way that it supports many authentication methods or mechanisms.

Extensible Authentication Protocol as one of the authentication mechanisms available today has been used in many types of networks, both wired and wireless, such as Point-to-Point Protocol (PPP), infrastructure model of WLAN / Wi-Fi [IEEE802.11i, 2004], and Worldwide Interoperability for Microwave Access (WiMAX) [IEEE802.16e, 2005], [Nuaymi, 2007]. However, the typical EAP authentication mechanism might not be able to be implemented in ad hoc wireless network due to the characteristics of ad hoc network, e.g. infrastructure-less. Thus

new scheme or mechanism of EAP-based authentication has to be designed and developed for ad hoc wireless network, and it motivated us to carry out this research.

One promising advantage of using EAP-based authentication mechanism in a type of network is interoperability with other types of network since EAP has already become the platform for many authentication mechanisms; and that is one step closer towards interoperability across heterogeneous networks in the near future.

1.3 Objectives

The objectives of this research are to study how EAP can be implemented in ad hoc wireless local area network (WLAN), to design and develop an EAP-based authentication mechanism for ad hoc WLAN using the existing EAP methods. Using the existing EAP methods, we hope that the development and implementation costs (time and effort) of the proposed mechanism will be significantly reduced as compared to the development of a new EAP method.

The mechanism should address the ad hoc network characteristics, i.e. infrastructure-less and heterogeneous mobile devices environment, where ad hoc nodes may exist in different types of device with different specifications and capabilities, and each node may support different types of EAP authentication method.

1.4 Contributions

The expected contributions of this research are as the following:

1. Design of an EAP-based authentication mechanism for ad hoc wireless local area network.
2. Formal specification and formal verification of the proposed mechanism as the proof of correctness.
3. Extension to the existing EAP model or architecture to support heterogeneous mobile devices environment.
4. Simulation model of the mechanism.

1.5 Scope of Research

This research emphasized on the EAP based authentication mechanism of wireless local area network security. We limited the research of the ad hoc wireless network model to the single-hop ad hoc or peer-to-peer model of IEEE 802.11 WLAN network devices, also referred to as Independent Basic Service Set (BSS), because we wanted to develop a more generic model and not restricted by a routing protocol. As we only worked on single-hop ad hoc wireless LAN, we did not emphasize on routing protocol aspect of ad hoc wireless LAN.

1.6 Thesis Structure

This thesis is divided into six chapters. Chapter One introduces the background that comprises the reason of conducting the research, the problems and the approach used to solve the problem. It also describes the objectives, contributions, and scope of this research.

Chapter Two elaborates comprehensive and extensive reviews of enabling technologies used to address the proposed model and implementation. Chapter Two also provides literature review and discussion of related research works.

From the issues highlighted in Chapter One and Chapter Two, Chapter Three presents the proposed mechanism. The model of EAP implementation in ad hoc wireless LAN and algorithm needed are extensively explained.

Formal specification and verification of the proposed model are derived and discussed in Chapter Four.

Chapter Five discusses the simulation design and development, issues and experience gained from simulation study during the research.

Finally, Chapter Six draws the conclusions of the research and recommendation for future works.

CHAPTER TWO : STATE OF THE ART

This chapter presents the background review on wireless network technology and its security aspects, the current methodologies used in EAP implementation, and elaboration on some selected works related to this research.

2.1 Wireless Networks

Wireless networks are usually categorized based on their coverage range, as the following:

1. Wireless Wide Area Network (WWAN)
2. Wireless Metropolitan Area Network (WMAN)
3. Wireless Local Area Network (WLAN)
4. Wireless Personal Area Network (WPAN)

2.1.1 Wireless Wide Area Network

Wireless WAN includes wide coverage area (regional, nation wide or even global scale) technologies such as:

1. The First Generation (1G) systems such as Advanced Mobile Phone Systems (AMPS).

2. The Second Generation (2G) and 2.5G systems such as Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), Global system for Mobile communication (GSM), General Packet Radio Service (GPRS), and EDGE (Enhanced Data rates for GSM Evolution).

3. The Third Generation (3G) and 3.5G systems such as Universal Mobile Telecommunication System (UMTS) using Wideband Code Division Multiple Access (WCDMA), and HSDPA (High Speed Downlink Packet Access).

4. Beyond 3G technologies, e.g. the Forth Generation (4G) system and so forth, that will give better speed, all digital system, convergence of data and voice over IPv6, etc.

2.1.2 Wireless Metropolitan Area Network

Wireless MAN coverage falls intermediately between wireless LAN and wireless WAN. It typically covers an area of the size of a town or city. The standard for the wireless MAN is IEEE 802.16 (commonly known as WiMAX) which defines broadband (high speed) connection / access from fixed or mobile wireless devices.

2.1.3 Wireless Local Area Network

Wireless LAN provides greater flexibility and mobility than the wired LAN with its range, reaching tens to hundreds of meters. The international standards for wireless LAN is the IEEE 802.11 family, providing transmission speeds ranging from 1 – 54 Mbps typically in 2.4 or 5 GHz frequency bands. The standard includes the following (adapted from [IEEE802.11, 2007]):

1. IEEE 802.11 provides 1 or 2 Mbps transmission in the 2.4 GHz radio frequency band using either Frequency Hopping Spread Spectrum (FHSS) or Direct Sequence Spread Spectrum (DSSS).

2. IEEE 802.11a is an extension of 802.11 that provides data rates up to 54 Mbps in the 5 GHz radio frequency band using Orthogonal Frequency Division Multiplexing (OFDM).

3. IEEE 802.11b provides data rates up to 11 Mbps in the 2.4 GHz frequency band. It is backward compatible with 802.11. This is the most widely deployed WLAN.

4. IEEE 802.11d defines specification for operation in additional regulatory domains.

5. IEEE 802.11e adds Quality-of-Service (QoS) enhancements and multimedia support to 802.11b and 802.11a.

6. IEEE 802.11f (trial-use) recommends practice for multi-vendor access point interoperability via an Inter-Access Point Protocol (IAPP) across Distribution Systems supporting 802.11 operations. However the current status of this trial-use standard is withdrawn.

7. IEEE 802.11g provides data rates up to 54 Mbps in the 2.4 GHz frequency band. It is compatible with 802.11b.

8. IEEE 802.11h defines spectrum and transmit power management extensions in the 5 GHz band in Europe.

9. IEEE 802.11i defines a framework and means for supporting security over WLAN. It adds Medium Access Control (MAC) security enhancements.

10. IEEE 802.11j defines specifications for 4.9 – 5 GHz operation in Japan.

There are other 802.11 standards developed or under development as amendments, enhancements, or extensions to WLAN, such as 802.11n draft standard (currently is Draft 3.02) which will be adding specifications for new technologies that will raise WLAN connection speeds to as much as 600 Mbps.

Wireless LAN provides shared radio media for users to communicate with each other and to accomplish the same functionality of wired LAN. Wireless LAN is usually implemented in areas that have no or limited wired network infrastructure, either as an extension of the wired networks (infrastructure WLAN) as shown in Figure 2-1, or as a infrastructure-less (peer-to-peer or Ad-Hoc) WLAN as shown in Figure 2-2.

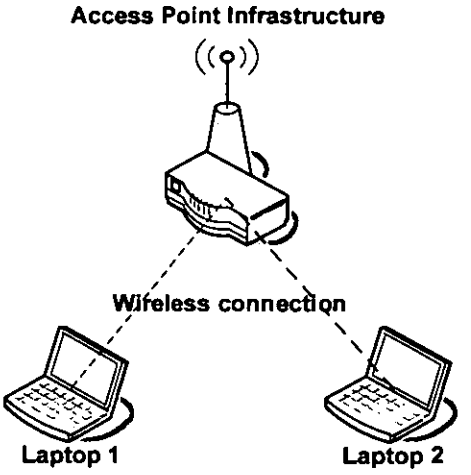


Figure 2-1: Infrastructure WLAN

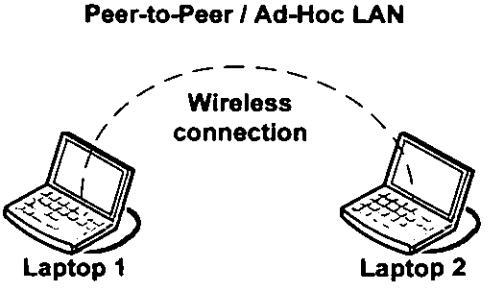


Figure 2-2: Ad Hoc WLAN

2.1.4 Wireless Personal Area Network

Wireless PAN use short-range (typically 10 meters maximum range) and low-power radios to facilitate communication between devices such as laptops, cell phones, and Personal Digital Assistant (PDA). The IEEE standard for wireless PAN is the IEEE 802.15. The well-known and widely-used wireless PAN technology is the Bluetooth technology.

2.2 Wireless Network Security

Network security has many different aspects which includes the following (adapted from [Chen & Zhang, 2004]):

1. Authentication. It is an ability for communicating parties to validate each other identity.
2. Authorization or access control. It is the ability of a party (such as network provider) to determine whether a user should be allowed to access particular network, service, or information.
3. Integrity refers to the protection of information from unauthorized change.
4. Confidentiality or privacy refers to how to keep the information private such that only authorized users can understand it. This is often achieved by encryption.
5. Availability. It is on how to ensure that legitimate access to the network or service will not be blocked by malicious users or attacks.
6. Non-repudiation refers to the ability of the network to supply undeniable evidence to prove that message transmission or network access is performed by a user.

Authentication is one aspect of security which enables network node to ensure and validate the identity of the peer node that it is communicating with. Without proper authentication, an adversary could gain unauthorized access to the network and interfere with the operation of the network.

Just like wired network, wireless network is also subject to threats and attacks. Due to its nature, wireless network is more vulnerable than wired network because attacker can intrude on the wireless network without any physical access to the facilities. Therefore implementing the proper security measurements, including authentication, on wireless network is vital in many aspect.

2.3 Authentication Mechanisms

One of the primary uses of authentication in networking is to implement access control. Controlling access to network is one of the primary defense mechanisms in network security. Authentication implements access control by identifying users or nodes trying to access the network. In the following sections, some mechanisms used in authentication will be discussed.

2.3.1 Address-based Authentication

In this approach, the access control is implemented by allowing only a predetermined set of addresses to access the network. This approach authenticates a node based on its address. The address used in this approach typically is MAC (Medium Access Control) or IP (Internet Protocol) address. Address-based authentication scheme in IP network may be implemented by network devices (switch, router, access point) allowing only a preconfigured set of MAC or IP addresses to access the network.

2.3.2 Password

Passwords are probably the oldest and the most common way of providing authentication, whether for logging into local machines or remote / network machines. Passwords have been used with computers since the early days of computing, back in 1960s. In network authentication, password usually serves as a key to be used in challenge-response systems. The password can be converted to the key using hash method or other ways in a way that the key can be derived only from the password.

2.3.3 Symmetric Key Infrastructure

In symmetric key infrastructure, there is only one key which only the communicating parties know as the shared-key. The shared-key can be used to encrypt and decrypt any message exchanged between parties.

If A wants to be authenticated to B, first A has to send its user name to B. Next B sends a random number (a challenge) to A. Then A encrypts the challenge with the shared-key and sends the result (response) back to B. When B receives the response, B encrypts the challenge it sent with the shared-key and compares it with the received value or B decrypts the response using the shared-key and compares it with the challenge it sent. If both values matched, B can be sure that it is communicating with someone who knows the shared-key. Assuming that only they (A and B) know the key, B has ensured that it is communicating with A.

2.3.4 Asymmetric / Public Key Infrastructure

In asymmetric / Public Key Infrastructure (PKI), there are two keys instead of one single key required. The two keys are private key, which is only known by the user, and public key, which is known by the others. These two keys are complementary; a message encrypted with one of these keys can only decrypted by the other key.

If A wants to communicate with B confidentially, A can ask B's public key and B sends its public key to A. Then A uses B's public key to encrypt the message and send it to B. Only B can decrypt the message because only B has its private key.

For authentication purpose, we can use private key to digitally sign a message. If A signs (encrypts) a message using its private key and then send it to B, B can decrypt it using A's public key and B can be sure that the message was sent by A because only A has the private key.

2.4 Wireless LAN Security Mechanisms

Currently there are security mechanisms that have been developed for wireless LAN using one or more authentication mechanisms discussed in the previous section in order to provide authentication, confidentiality, and integrity. As we stated in our scope of research in Chapter One, we will only discuss the security mechanisms for 802.11 wireless local area networks. The security mechanisms are Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), and WPA version 2 (WPA2).

2.4.1 Wired Equivalent Privacy (WEP)

WEP was developed to work as a part of the IEEE 802.11 standard and was chosen as the first security scheme for WLAN for reasons of cost, flexibility, simplicity, and computational efficiency.

WEP is a symmetric encryption algorithm using a static shared-key of 40 or 104 bits long and added an Initialization Vector (IV) of 24 bits long, for a total of 64

or 128 bits. WEP uses RC4 (Ron's Code 4) stream cipher algorithm to do per packet encryption to provide confidentiality and Integrity Check Value (ICV) using CRC-32 (Cyclic Redundancy Check) checksum process to provide data integrity.

It was soon discovered that WEP has flaws and in 2001 it is proven that WEP can be easily broken [Fluhrer et al., 2001], [Stubblefield et al., 2001]. Some of the flaws in WEP are:

1. WEP uses static shared keys and poor key management, so keys could be compromised easily.
2. Key streams in WEP are repeated / reused which allow easy decryption of data for a moderately sophisticated adversary.
3. Key length was short and it can be discovered after eavesdropping on the network for few hours.
4. Subject to brute-force attack due to its short keys.

2.4.2 Wi-Fi Protected Access (WPA)

WPA was created by the Wi-Fi Alliance as an intermediate solution to address most of the weaknesses of WEP until there was a more secure protocol standard (IEEE 802.11i).

WPA encrypts data using RC4 stream chipper with 128 bit key and 48 bit Initialization Vector (IV). It is almost the same as WEP with an addition of Temporal Key Integrity Protocol (TKIP) which dynamically changes the key as the system is being used and combined with larger IV. For data integrity, WPA uses new algorithm called "*Michael*" as message integrity check (MIC). *Michael* uses no multiplication operations, which can be computation intensive. It relies instead on shift and add operations, which only require much less computation [Chandra, 2005].

Although WPA significantly improves WEP security, there are still concerns on WPA, such as:

1. Design limitations of TKIP and *Michael* in order to support the existing hardware, result in cryptographic weaknesses [Miller & Hamilton, 2002].
2. Data tampering and masquerading are not completely resolved by WPA [Karnik & Passerini, 2005].
3. WPA is susceptible to Denial-of-Service attacks [NETGEAR, 2005], [Maple et al., 2006].

2.4.3 IEEE 802.11i / Wi-Fi Protected Access 2 (WPA2)

IEEE 802.11i / WPA2 can be described as the highest level of wireless network security protocol available currently. It addresses three main security areas: authentication, key management, and data transfer privacy. The 802.11i architecture contains the following components: IEEE 802.1X for authentication (entailing the use of EAP and an authentication server), Robust Security Network Association (RSNA) for keeping track of associations, and Advanced Encryption Standard (AES) based Counter mode CBC-MAC Protocol (CCMP) to provide confidentiality, integrity, and origin authentication.

WPA2 does not use RC4 like WEP and WPA. It uses CCMP to encrypt network traffic. CCMP uses AES as the encryption algorithm. The Counter mode is used for data encryption and the Cipher Block Chaining – Message Authentication Code (CBC-MAC) is used for message / data integrity.

Table 2-1 summarizes and shows the comparison of the three security mechanisms.

Table 2-1: Comparison of WEP, WPA, and WPA2

	WEP	WPA	WPA2
Cipher type	Stream	Stream	Block
Cipher	RC4	RC4	AES
Key Size	40 or 104 bits	128 bits	128 bits
IV size	24 bits	48 bits	48 bits
IV reuse protection	No	Yes	Yes
Security protocol	WEP	TKIP	CCMP
Message/data integrity	ICV	Michael	MAC
Overall security level	Broken	Secure	State-of-the-art

2.5 Extensible Authentication Protocol (EAP)

Extensible Authentication Protocol (EAP) is one of the authentication mechanisms that have been widely used nowadays. It has been used in Point-to-Point Protocol (PPP), wired networks, and wireless networks.

IEEE 802.1X standard is a port-based network access control that makes use of the physical access characteristics of IEEE 802 LAN infrastructures to provide a mean of authenticating and authorizing devices attached to a LAN port, and of preventing access to that port in cases for which the authentication and authorization fails. This IEEE 802.1X standard makes use of EAP.

EAP is an authentication framework which supports multiple authentication methods. EAP was initially used for PPP authentication, but it can also run over other data-link layers such as the IEEE 802 LAN family.

One of the advantages of EAP framework is its flexibility [Aboba et al., 2004]. EAP may be used on dedicated links, switched circuit links, wired and wireless links. EAP also permits the use of back-end authentication server to implement some or all the authentication methods. It is proven that EAP can be implemented in various network access technologies including the 2G technology: Global System for Mobile

communication (GSM), 3G technology: Universal Mobile Telecommunication System (UMTS), Wi-Fi / WLAN, 3G – WLAN internetworking based on the 3rd Generation Partnership Project (3GPP) specification [3GPP, 2006], [Chen et al., 2003], [Kambourakis et al., 2004], [Zhao et al., 2006], and EAP also has been ratified as one of the authentication mechanism for IEEE 802.16e WiMAX.

2.5.1 EAP Methods

EAP methods are authentication methods used in EAP. There are many types of EAP methods available today using many kinds of mechanisms or technologies such as passwords, certificates, challenge-response, hash, smart card, etc. Some of the existing EAP methods are:

1. EAP with MD5 hash (EAP-MD5) [Aboba et al., 2004]
   EAP-MD5 uses Message-Digest algorithm 5 (MD5) hash to authenticate client. This is the basic EAP method and in today's network environment this method gives insufficient wireless network security.

2. EAP with Transport Layer Security (EAP-TLS) [Aboba & Simon, 1999]
   EAP-TLS uses TLS, successor of Secure Socket Layer (SSL) version 3, and requires both the client-side and server-side to have Public Key Infrastructure (PKI) digital certificates in order to provide secure mutual authentication. This method is currently considered as the strongest EAP method (security wise) [Ali & Owens, 2007], [Microsoft, 2007].

3. EAP with Tunneled TLS (EAP-TTLS) [Funk & Blake-Wilson, 2008]
   EAP-TTLS offers strong security while avoiding the complexities of PKI implementation on client's side. EAP-TTLS requires server-side certificate while user-side can use an extensible set of user authentication such as Windows login and password and legacy user authentication methods. EAP-TTLS uses secure TLS record layer channel to set up tunnel to exchange information between client and server. It was co-developed by Funk Software and Certicom.

4. Protected EAP (PEAP) [Kamath et al., 2002], [Palekar et al., 2004]
   PEAP is similar to EAP-TTLS in the way that it only requires server-side certificate and using other way to authenticate client such as Microsoft's Challenge Handshake Authentication Protocol (MS-CHAP). PEAP uses TLS tunnel and it also offers strong security. The main difference of PEAP and EAP-TTLS is in their compatibility with legacy (older) methods and platforms. PEAP is less compatible compared to EAP-TTLS due to PEAP's compatibility only with newer operating systems (Microsoft Windows XP and above) and methods. It was jointly developed by Microsoft, Cisco, and RSA Security.

5. Lightweight EAP (LEAP) [Sankar et al., 2005]
   LEAP is a proprietary EAP method developed by Cisco Systems for their wireless LAN devices. LEAP supports mutual authentication and dynamic security keys changes in every (re)authentication with the hope that the keys will not live long enough to be used by attacker.

6. EAP with Subscriber Identity Module (EAP-SIM) [Haverinen & Saloway, 2006]
   EAP-SIM specifies mechanism for authentication and session key distribution using the 2G GSM network SIM. The EAP-SIM mechanism specifies enhancements to GSM authentication and key agreement and it also includes network authentication, user anonymity support, result indications, and a fast re-authentication procedure.

7. EAP-AKA (Authentication and Key Agreement) [Arkko & Haverinen, 2006]
   EAP-AKA specifies mechanism for authentication and session key distribution that uses the AKA mechanism. AKA is used in the 3G mobile networks UMTS and CDMA2000 with the use of UMTS SIM (USIM).

Table 2-2 provides properties and comparison of several EAP methods.

Table 2-2: Properties of EAP authentication methods (adapted from [Ali & Owens, 2007])

Property	***EAP Authentication Method***				
	MD5	**LEAP**	**TLS**	**TTLS**	**PEAP**
Authentication attributes.	Unilateral	Mutual	Mutual	Mutual	Mutual
Deployment difficulties.	Easy	Easy	Hard	Moderate	Moderate
Dynamic re-keying.	No	Yes	Yes	Yes	Yes
Requires server certificate.	No	No	Yes	Yes	Yes
Tunneled.	No	No	No	Yes	Yes
WPA compatible.	No	Yes	Yes	Yes	Yes
WLAN security.	Poor	Moderate	Strongest	Strong	Strong

2.5.2 EAP Entities

There are three entities defined in the IEEE 802.1X standard that involved in the EAP authentication process:

1. *Supplicant*, an entity in the network that seeks to be authenticated.
2. *Authenticator*, an entity that facilitates authentication of the supplicant.
3. *Authentication Server*, an entity that provides the authentication service to the authenticator.

In Wi-Fi environment, typically mobile / wireless station or device will act as the supplicant. Wireless access point acts as the authenticator. Authentication, Authorization, and Accounting (AAA) server, such as RADIUS (Remote Authentication Dial-In User Service) or Diameter server acts as the authentication server.

EAP permits the Authentication Server to implement some methods, or all methods while the Authenticator only acts as a pass-through entity. The Authenticator and Authentication Server may reside in different devices or collocated in one device. The implementation model of EAP will be explained further in the next section.

2.5.3 EAP Model

Based on its Request For Comments (RFC) document, i.e. RFC 3748, EAP can be illustrated using layer model as in Figure 2-3. It consists of the following components:

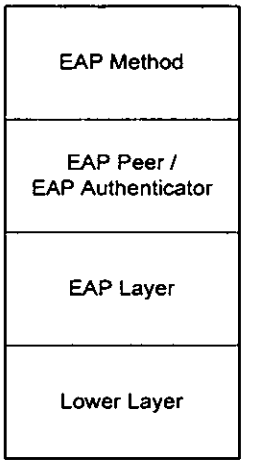


Figure 2-3: EAP layer model

1. *Lower Layer*

   EAP lower layer is responsible for transmitting and receiving EAP frames between the peer and authenticator. This layer includes Point-to-Point Protocol (PPP), wired IEEE 802 LANs, IEEE 802.11 wireless LAN, and other data-link layers.

2. *EAP Layer*

   EAP layer receives and transmits EAP packets via the lower layer; it also implements duplicate detection and retransmission, and delivers and receives EAP messages to and from the EAP peer and authenticator layers.

3. *EAP Peer* or *EAP Authenticator Layer*

   EAP peer or authenticator layer receives and transmits EAP packets via EAP layer, and also delivers and receives EAP messages to and from EAP method layer. Typically implementation on a host will only support either peer or authenticator functionality, but it is possible for a host to act as both peer and authenticator.

4. *EAP Method Layer*

   EAP method layer implements the authentication algorithms, receives and transmits EAP messages via the EAP peer and authenticator layers.

2.5.4 EAP Implementation Model

In the typical EAP implementation, the authenticator acts as a pass-through authenticator. It forwards packets from the peer and destined to its authenticator layer to the back-end authentication server; and vice versa packets received from the back-end authentication server destined to the peer are forwarded to it. Layer model of pass-through behavior model is illustrated in Figure 2-4 and the messages exchange flow is illustrated in Figure 2-5.

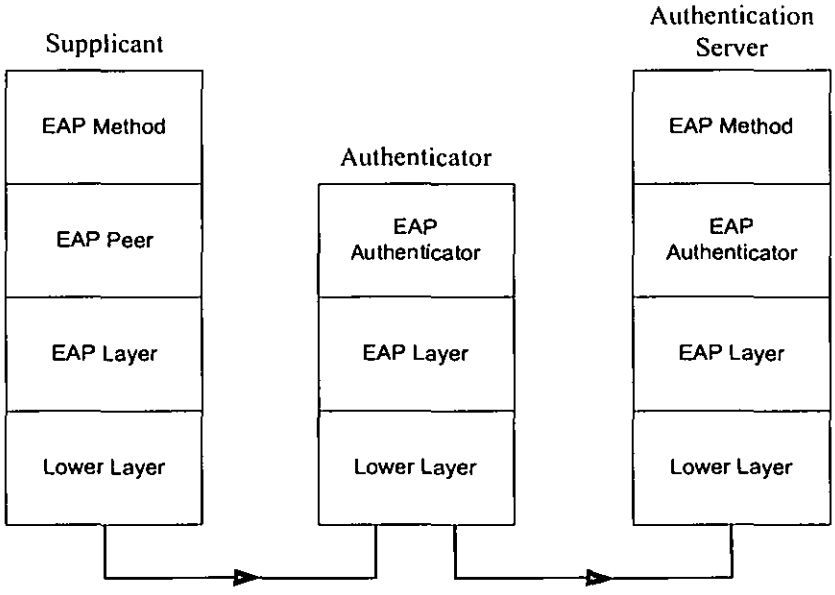


Figure 2-4: Pass-Through Behavior Implementation Model

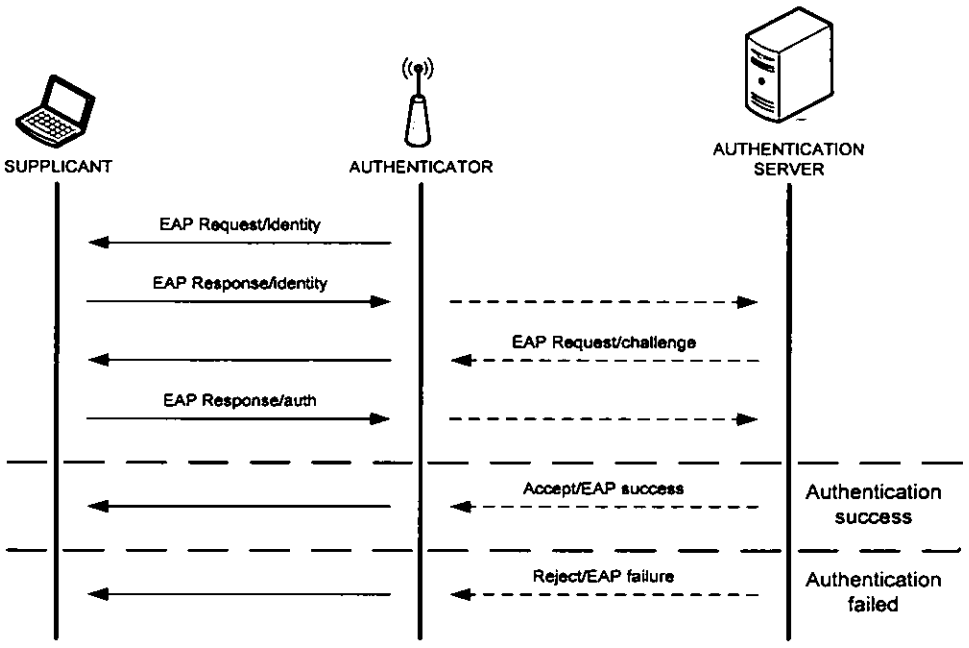


Figure 2-5: Pass-Through Behavior Messages Exchange

Another approach to implement EAP specified in its RFC is the EAP multiplexing model as illustrated in Figure 2-6. In the multiplexing model, there is no authentication server entity since the authenticator will implement all the authentication methods, or the authentication server service is embedded into the authenticator. This may require the node or host that acts as the authenticator to have more computational capabilities in order to support functionality of both authenticator and authentication server and also implement all the authentication methods. A typical authenticator devices, such as access points, might not be able to handle those functionalities, thus computers (PC, server, laptop) are more likely to be used as authenticator.

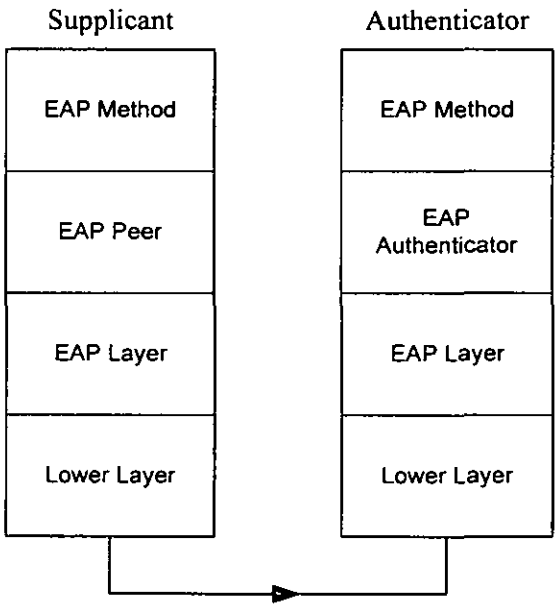


Figure 2-6: EAP Multiplexing Model

2.5.5 EAP Packet Format

EAP packet format is illustrated in Figure 2-7. The fields are transmitted from left to right.

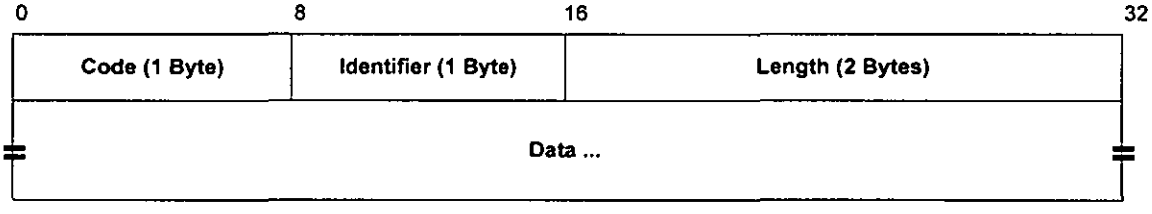


Figure 2-7: EAP Packet Format (adapted from [Aboba et al., 2004])

The *Code* field is 1 octet / byte long and identifies the type of EAP packet. EAP Codes are assigned as the following:

1 Request

2 Response

3 Success

4 Failure

The *Identifier* field is 1 octet long and it aids in matching Responses with Requests. The *Length* field is 2 octets long, and it indicates the length of the EAP packet including the *Code*, *Identifier*, *Length*, and *Data* fields. The *Data* field is 0 or more octets long. The format of the *Data* field is determined by the type of EAP packet (depends on the *Code* field).

2.5.6 EAP Encapsulation Over LAN (EAPOL)

EAP Over LAN (EAPOL) is the encapsulation used to carry EAP packets from Supplicant to Authenticator in LAN environment. The EAPOL encapsulation used in wireless LAN is the EAPOL encapsulation for IEEE 802.3/Ethernet Medium Access Control (MAC). A summary of the Ethernet form of an EAPOL MAC Protocol Data Unit (MPDU) is illustrated in Figure 2-8.

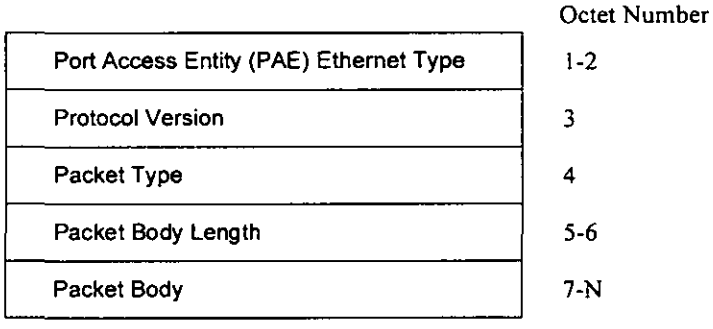


Figure 2-8: EAPOL MPDU format for IEEE 802.3/Ethernet [IEEE 802.1X, 2004]

PAE Ethernet Type field is 2 octets long and it contains the Ethernet Type value assigned for use by the PAE. *Protocol Version* field is 1 octet long and its value identifies the version of EAPOL protocol supported by the sender of the EAPOL frame. *Packet Type* field is 1 octet long and its value determines the type of packet being transmitted. *Packet Body Length* field is 2 octets long and its value defines the

length, in octets, of the *Packet Body* field. *Packet Body* field is present only if the *Packet Type* field contains the value of EAP-Packet, EAPOL-Key, or EAPOL-Encapsulated-ASF (Alerting Standards Forum)-Alert.

2.6 Related Works

Although ad hoc network has attracted great attention for the past few years, most research efforts in ad hoc network have been focused on the development of the network architecture itself, particularly in the network routing protocol and medium access control (MAC) protocol designs [Perkins, 2001], [Ilyas, 2003], [Cheng et al., 2004]. There are relatively little works that have been carried out with security consideration, and most of the ad hoc routing protocols assume that all nodes are cooperative and trustworthy in nature.

As far as we know currently there is no specific EAP method developed for ad hoc network and there are only few EAP mechanisms that have been proposed (in open literature) for ad hoc network authentication.

[Lee & Park, 2003] proposed a user authentication mechanism for mobile ad hoc networks using EAP and Ad-hoc On-demand Distance Vector (AODV) routing protocol. The mechanism defines master node for authentication server and how other nodes acquire authentication from it using MD5 Challenge. This mechanism requires modification / expansion of EAP and AODV hello packet format, and MD5 does not provide sufficient protection.

[Moustafa et al., 2005] proposed architecture for vehicular communication on highways with ad hoc networking support. The architecture adapts an Authorization, Authentication, and Accounting (AAA) scheme using Kerberos, instead of a RADIUS server, EAP-Kerberos and EAP-TLS methods for vehicular communication on highways environment. The proposed architecture still requires fixed network infrastructure and access points on highway entry points.

[Khan & Akbar, 2006] proposed to use EAP-TTLS and Protocol for carrying Authentication for Network Access (PANA) in multi-hop wireless mesh networks (WMN) that can be extended by ad hoc network. This method requires PANA which is still under development and only exists in the form of an Internet Engineering Task Force (IETF) draft for further review.

[Nidjam & Scholten, 2006] proposed the use of virtual Authentication Server in Wi-Fi ad hoc implementation of Access Point Security Service (APSS) with a scenario that comprises of two people communicating for the first time at a conference, both having subscriptions with network service providers. The APSS mechanism still requires the existence of fixed infrastructures, e.g. access points, of the network service providers, both telecommunication and wireless hotspot service providers.

Most of the works above are still in architecture or mechanisms proposal stage. As far as we know, except for [Nidjam & Scholten, 2006], those works have not provided any proof of concept of their proposed mechanisms whether in formal verification, simulation development, test bed or real network implementation.

Related to the idea of supporting heterogeneous mobile devices environment with different supported EAP method, [Ali & Owens, 2007] have pointed out the need of selecting the most suitable authentication method for a particular wireless LAN network environment. The work is useful in selecting one EAP method suitable for a network before implementing the EAP framework in that particular network. It identified the factors to be considered before users or network designers are going to employ EAP in wireless LAN, and it could serve as a foundation for our method selection and negotiation algorithm which will be discussed in Chapter Three.

2.7 Summary

The purpose of this chapter is to provide the reader with sufficient background information to understand the foundations and concepts to be elaborated in the rest of this thesis. This chapter also discussed some selected works related to our work. The major difference between this work and theirs is that this work designs an EAP

authentication mechanism as extension to existing framework and network, using the EAP methods available today without the need to propose a new EAP method or modifying an EAP method. This work also provides the formal verification and simulation development of the proposed mechanism as proof of concept. We also propose a mechanism to select and negotiate the suitable EAP method for the nodes / devices over the existing wireless network and EAP framework, which consists of heterogeneous wireless mobile devices. The proposed mechanism will be discussed in the next chapter.

CHAPTER THREE : PROPOSED AUTHENTICATION MECHANISM

This chapter describes the details of the proposed EAP authentication mechanism for ad hoc wireless LAN. First, an overall description of the mechanism is given. Then consecutively, each phase of the authentication mechanism is elaborated.

3.1 Overview

We designed the mechanism of EAP-based authentication for ad hoc wireless LAN based on EAP multiplexing model where there is no separate authenticator or access point entity, as described in Chapter Two, and master node is used instead. This model is more suitable with the infrastructure-less nature of ad hoc (no access points) and the authentication is carried out between two nodes.

We chose to use the existing EAP methods in the authentication mechanism. There are already numerous EAP methods available today and they are sufficient to provide secure authentication in many network conditions. We just need to select the suitable EAP method for our needs. We believe it can help to reduce the development, implementation, and deployment time of the mechanism significantly as compared to the development of a new EAP method.

The authentication process in the proposed mechanism consists of two parts: initial authentication and operational authentication. The initial authentication is carried out between the master node and the client mobile nodes. The operational authentication is carried out between mobile nodes, without the master node.

The ad hoc network configuration consists of one or some master node(s) and several mobile nodes. The master node is the node that will act as the authentication server that will provide the authentication service in the initial phase. The mobile nodes are the nodes that seek to be authenticated, whether to the master node in the initial phase or to each other in the operational phase. Master node should also have digital certificate service installed to issue certificates for the mobile nodes.

Figure 3-1 illustrates the topology of the ad hoc wireless LAN in which the proposed authentication mechanism can be applied to. The green communication links illustrate node-to-master-node authentication, while the yellow ones illustrate node-to-node authentication.

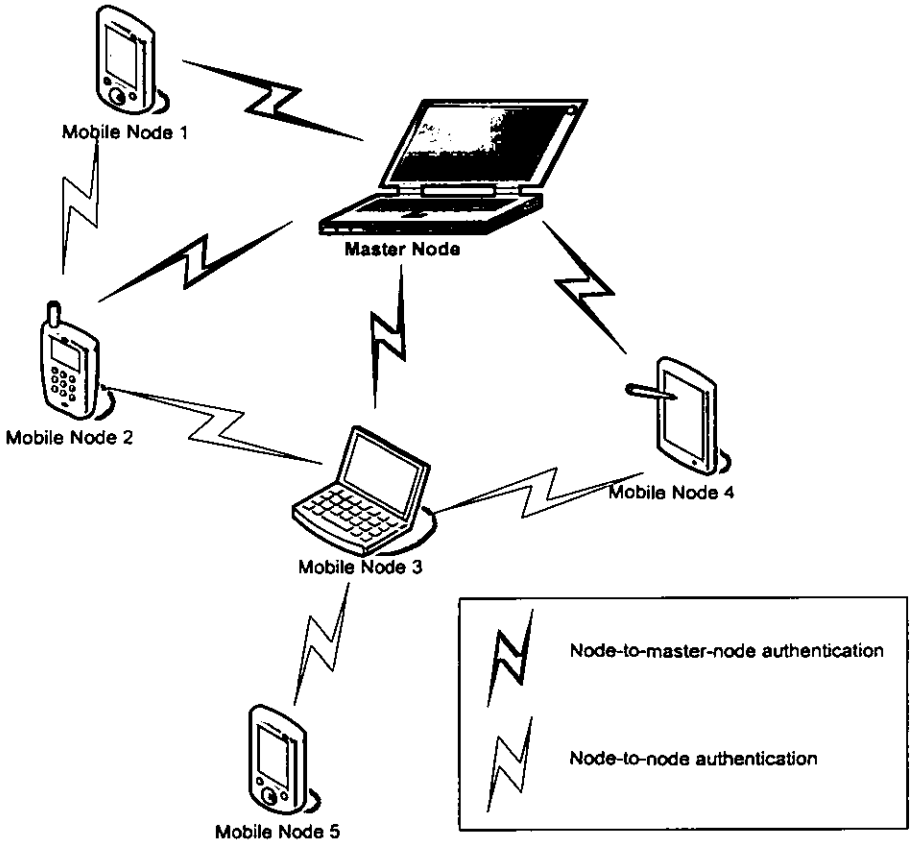


Figure 3-1: Ad hoc wireless LAN topology of the proposed mechanism

3.2 Initial (Node-to-Master-Node) Authentication

In the initial phase, the client mobile nodes are authenticated to the master node which has the authentication server and digital certificate services installed. The mobile node will prove its identity using user name, ID number, or serial number, and password. This method is chosen because mobile node may not have any certificate yet since the Public Key Infrastructure (PKI) may not be available in ad hoc network. EAP method that only requires server-side certificates, i.e. EAP-TTLS and PEAP, is used in this phase.

After successful authentication, the mobile nodes will receive digital certificates generated and signed by the master node. This certificate will have expiry timestamp and will be used in operational authentication. The initial authentication can be implemented using EAP pass-through behavior model or EAP multiplexing model; however in ad hoc network the latter is used. Diagram of the initial phase, i.e. Node to Master Node, authentication is illustrated in Figure 3-2 and the EAP messages exchange is illustrated in Figure 3-3.

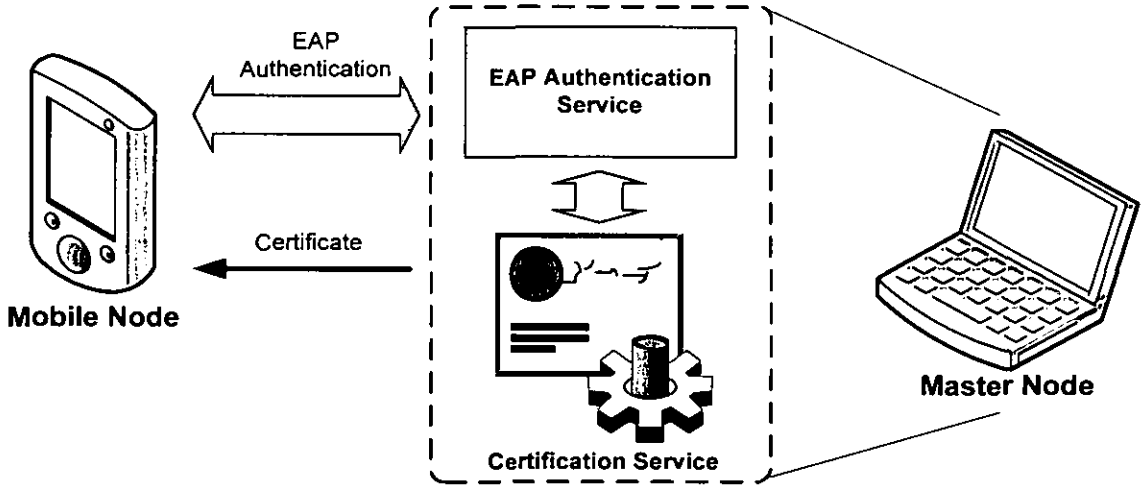


Figure 3-2: Node-to-Master-Node authentication

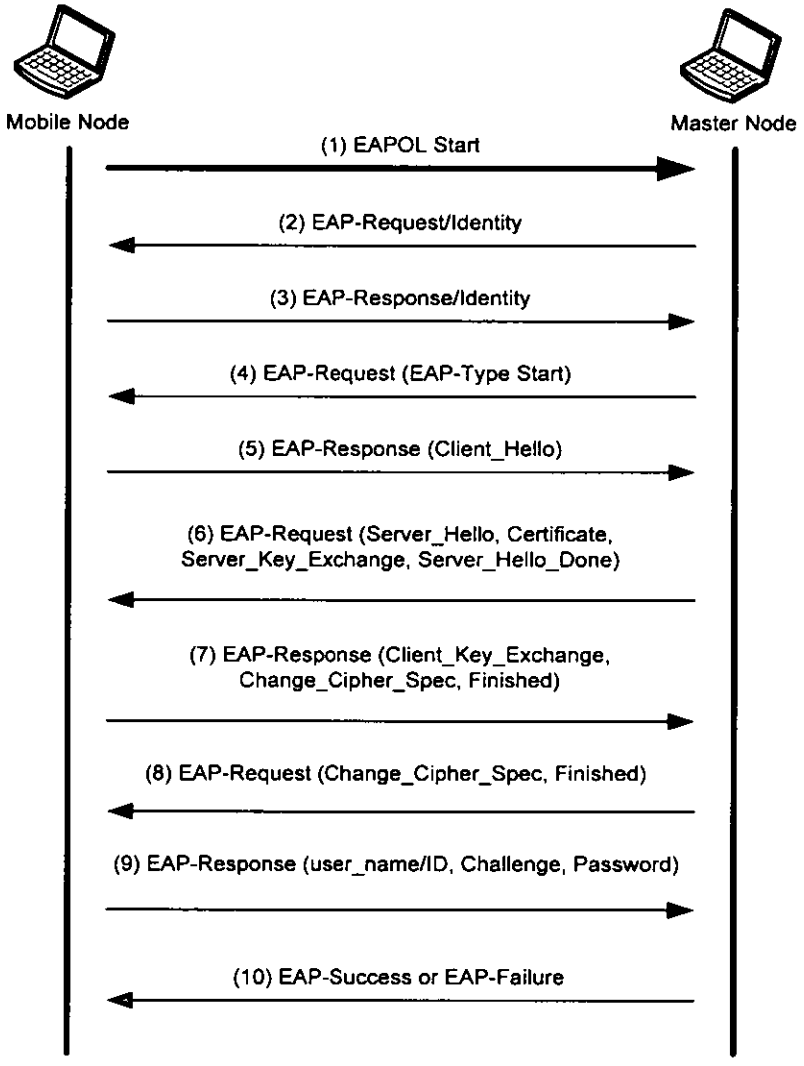


Figure 3-3: Node-to-Master-Node authentication messages exchange

The following is the process executed in the initial authentication:

1. Mobile node (client) sends EAP Over LAN (EAPOL) Start message to the Master node.
2. Master node / authentication server sends an EAP-Request/Identity packet to the client.
3. Client responds with an EAP-Response/Identity packet to the server (containing client's session ID).
4. Server responds with an EAP-Type Start packet (EAP-TTLS or PEAP).

Execute TLS handshake process for server authentication (messages 4 – 7):

5. Client sends a *"Client hello"* message to the server, containing client's random value (nonce).
6. Server responds by sending a *"Server hello"* message to the client, along with the server's random value (nonce), and its certificate.
7. Message 6:

    A. Client creates a random Pre-Master Secret (PMS).
    B. Client encrypts the PMS with the public key from the server's certificate.
    C. Client sends the encrypted PMS to the server, along with *"Change cipher spec"* notification to server to indicate that the client will start using the new session keys for hashing and encrypting messages, and also the *"Finished"* message.

8. Message 7:

    A. Server receives client's response and decrypts the PMS using its private key.
    B. Server and client each generate the Master Secret and session keys based on the Pre-Master Secret using pseudo-random-number function (PRF).
    C. Upon receiving the *"Change cipher spec"* from client, server switches its record layer security state to symmetric encryption using the session keys.
    D. Server sends *"Change cipher spec"* and *"Finished"* messages to the client.

9. Execute client authentication:

    A. Client sends its username or ID, password, and challenge, encrypted with the session key.
    B. If username, password, and challenge are validated by server then go to [EAP-Success] else go to [EAP-Failure].

10. Message 9:

    A. EAP-Success:

        i. Client creates its private and public keys.
        ii. Server creates certificate for client (containing client's identity, public key, and validity period / expiry).
        iii. Server signs client's certificate with server's private key.

B. EAP-Failure:

i. Abort authentication.

ii. Server and client disconnect from each other.

3.3 Operational (Node-to-Node) Authentication

Figure 3-4 illustrates the diagram of the operational (Node-to-Node) authentication.

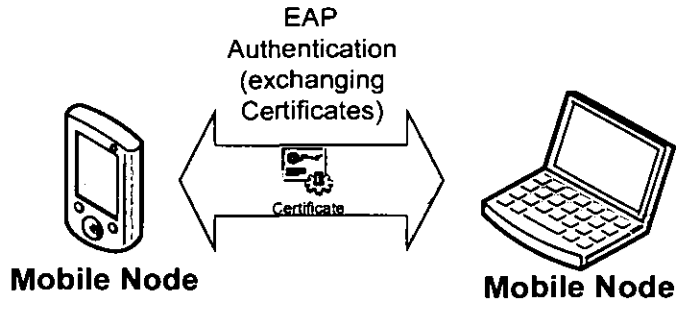


Figure 3-4: Node-to-Node authentication

In the operational phase, the mobile nodes that have been authenticated will authenticate each other using their digital certificates received from master node. The mobile nodes will exchange their certificates, checking the validity and expiry of the certificates, thus proving their identities and then they can proceed to communicate in ad hoc fashion.

We use EAP types that employ authentication requiring or supporting certificates of both sides, i.e. EAP-TLS, EAP-TTLS, and PEAP. The operational phase authentication is implemented using EAP multiplexing model without the need of authentication server / master node support. The EAP messages exchange of the operational phase, i.e. Node to Node authentication is illustrated in Figure 3-5.

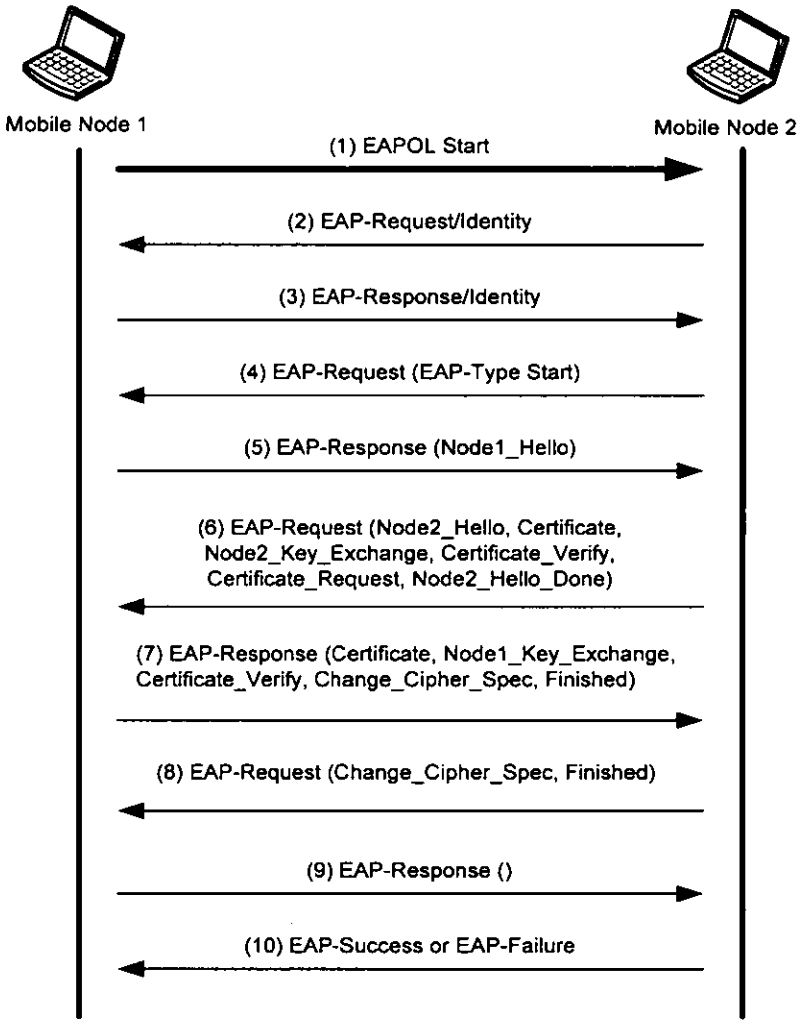


Figure 3-5: Node-to-Node authentication messages exchange

The following is the process executed in the operational phase authentication:

1. Mobile node1 sends EAP Over LAN (EAPOL) Start message to mobile node2.
2. Mobile node2 sends an EAP-Request/Identity packet to mobile node1.
3. Node1 responds with an EAP-Response/Identity packet to node2 (containing Node1's session ID).
4. Node2 responds with an EAP-Type Start packet.

Execute TLS handshake process (messages 4 – 7):

5. Node1 sends a *"Node1 hello"* message to the Node2, containing the Node1's random value (nonce).

6. Node2 responds by sending a *"Node2 hello"* message to Node1, along with Node2's random value (nonce), its certificate, its *"Certificate verify"* message, and request for Node1's certificate.
7. Message 6:
   A. Node1 validates Node2's certificate (using Master Node public key obtained in initial phase).
   B. If Node2's certificate is validated then continue else go to [EAP-Failure].
   C. Node1 creates a random Pre-Master Secret (PMS) and generates the Master Secret and session keys based on the PMS using PRF.
   D. Node1 encrypts the PMS with the public key from Node2's certificate.
   E. Node1 sends the following messages to Node2: the encrypted PMS, its certificate, its *"Certificate verify"* message, *"Change cipher spec"* notification to Node2 to indicate that Node1 will start using the new session keys for hashing and encrypting messages, and also the *"Finished"* message.
8. Message 7:
   A. Node2 decrypts Node1's response and validate Node1's certificate (using Master Node public key obtained in initial phase).
   B. If Node1's certificate is validated then continue else go to [EAP-Failure].
   C. Node2 generates the Master Secret and session keys based on the PMS received from Node1 using PRF.
   D. Upon receiving the *"Change cipher spec"* from Node1, Node2 switches its record layer security state to symmetric encryption using the session keys.
   E. Node2 sends *"Change cipher spec"* and *"Finished"* messages to Node1.
9. Node 1 sends EAP-Response () message.
10. Message 9:
    A. EAP-Success:
       i. Node1 and Node2 can start data communication.
    B. EAP-Failure:
       i. Abort authentication.
       ii. Node1 and Node2 disconnect from each other.

In typical client – server TLS handshake process, there is no requirement for server to send 'certificate verify' message to client, only client sends it to server. However in this authentication process, both nodes are clients, thus both nodes need to send their 'certificate verify' messages in order to prove their identities by signing the message with their private keys.

3.4 EAP Method Selection and Negotiation Mechanism

We propose an extension to the existing EAP architecture / framework in order to support heterogeneous wireless devices environment. The proposal is to have an EAP method selection and negotiation process which is executed before the EAP authentication process, as illustrated in Figure 3-6. The extended EAP architecture is illustrated in Figure 3-7.

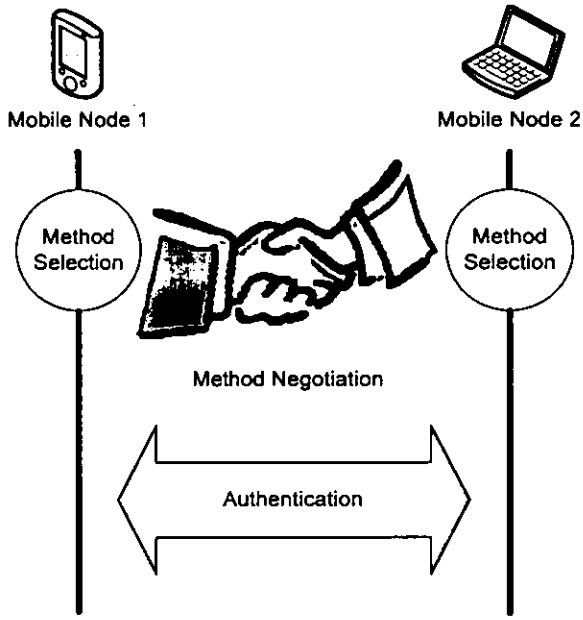


Figure 3-6: Overview of EAP authentication with EAP method selection and negotiation

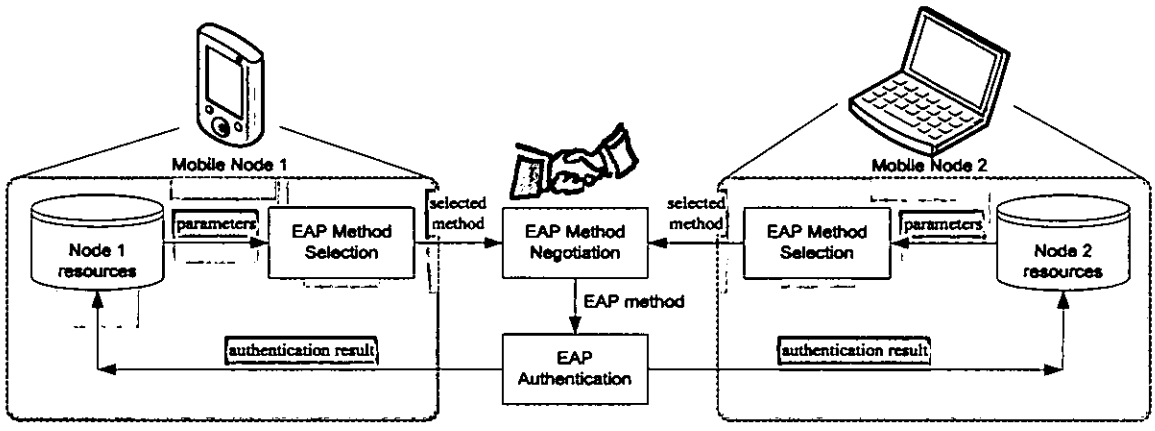


Figure 3-7: EAP authentication with EAP method selection and negotiation

The purpose of the EAP method selection and negotiation mechanism is to recommend the suitable EAP method to be used between two nodes. The selection is based on the following criteria:

1. The nodes resources: the node specifications, certificates availability, operating system, etc. Each EAP method may require different resources; therefore it is important to select the suitable EAP method based on this criterion.

2. Previous authentication records: these records will provide data about previous authentications, such as list of nodes that have authenticated or have been authenticated, time of authentications, the used methods, authentication results, etc. These data can be used to obtain useful information for the method selection, such as the last successful authentication method and the most successful authentication method.

3. Previous communication records: these records will provide data about previous network communications, such as previous malicious packets / traffic from other nodes. These records could be an output or a log file from a network communication analyzer program which monitors and analyzes the network activities and records them in a log file. The collected data can provide information as to whether a node to be authenticated is considered

harmless or otherwise by examining whether that node has history of sending malicious packets.

In this work, we used a set of EAP methods that have been widely used and considered to give strong security protections. They are EAP-TLS, EAP-TTLS, and PEAP [Ali & Owen, 2007]. We put the highest priority to EAP-TLS since it provides the strongest security protection [Ali & Owens, 2007], [Microsoft, 2007]. The requirement of using EAP-TLS is the digital certificates in both nodes, thus we have to check their availability first.

If only one party has digital certificate, then we have to use EAP-TTLS or PEAP. Both EAP-TTLS and PEAP give strong level of protection. However PEAP has the disadvantage of less compatibility and flexibility compared to EAP-TTLS because PEAP only supports newer Microsoft Windows operating systems, (Windows XP and above), and Microsoft mechanisms, e.g. Microsoft – Challenge Handshake Authentication Protocol version 2 (MS-CHAPv2) [Khan & Akbar, 2006], while EAP-TTLS can be used in different platforms or operating systems and supports many mechanisms including the legacy (older) mechanisms, such as Password Authentication Protocol (PAP), CHAP, MS-CHAP, and MS-CHAPv2.

Although EAP-TTLS has the advantage over PEAP, for better compatibility, we put EAP-TTLS as the last option of method when conditions for the other methods are not met. We can do the selection by checking the operating system of the node. Thus, the flow diagram of the EAP authentication with EAP method selection can be illustrated as in Figure 3-8.

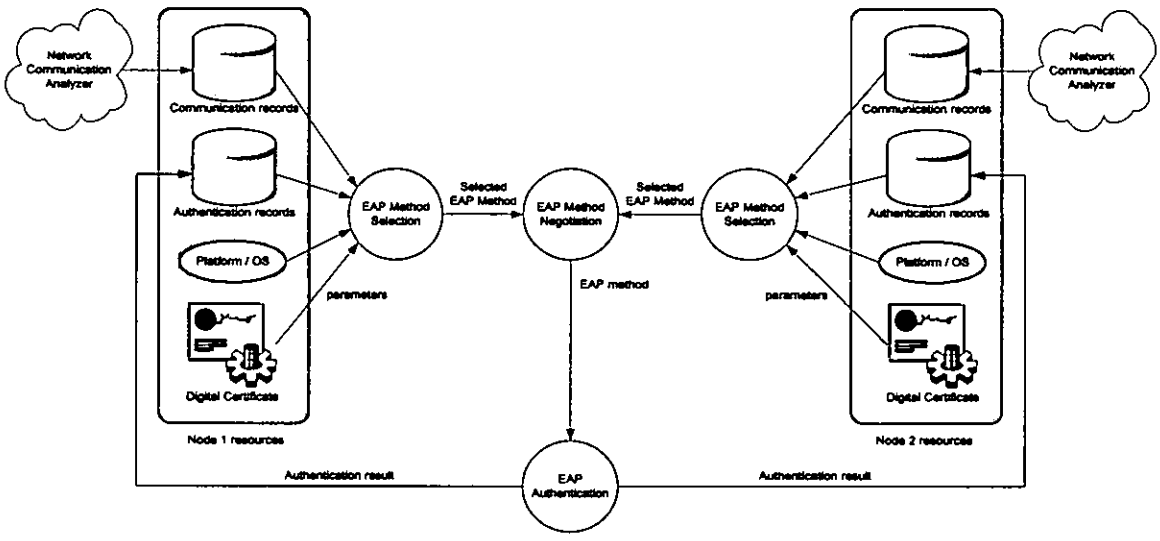


Figure 3-8: Flow diagram of EAP authentication with EAP method selection and negotiation

The algorithm of the EAP method selection mechanism is illustrated in the following pseudo code and flowchart (Figure 3-9). In this algorithm, if there is any history of malicious packet from the other node trying to be authenticated, the authentication process will be aborted immediately.

```
1.  Start
2.  if communication record available then {
3.      get communication record data;
4.      if any malicious packets came from the other node then {
5.          abort authentication process;
6.          go to End;
7.      }
8.      else if authentication record available then {
9.          get authentication record data;
10.         if successful authentication record available then {
11.             check current node resources;
12.             if current resources comply with last successful authentication
                method then
13.                 use last successful authentication method;
```

```
14.             else if current resources comply with most successful
                authentication method then
15.                 use most successful authentication method;
16.             else
17.                 go to step 25
18.         }
19.         else
20.             go to step 25
21.     }
22.     else
23.         go to step 25
24. }
25. else {
26.     check current node resources
27.     execute method selection based on current resources
28.     use selected method
29. }
30. End
```

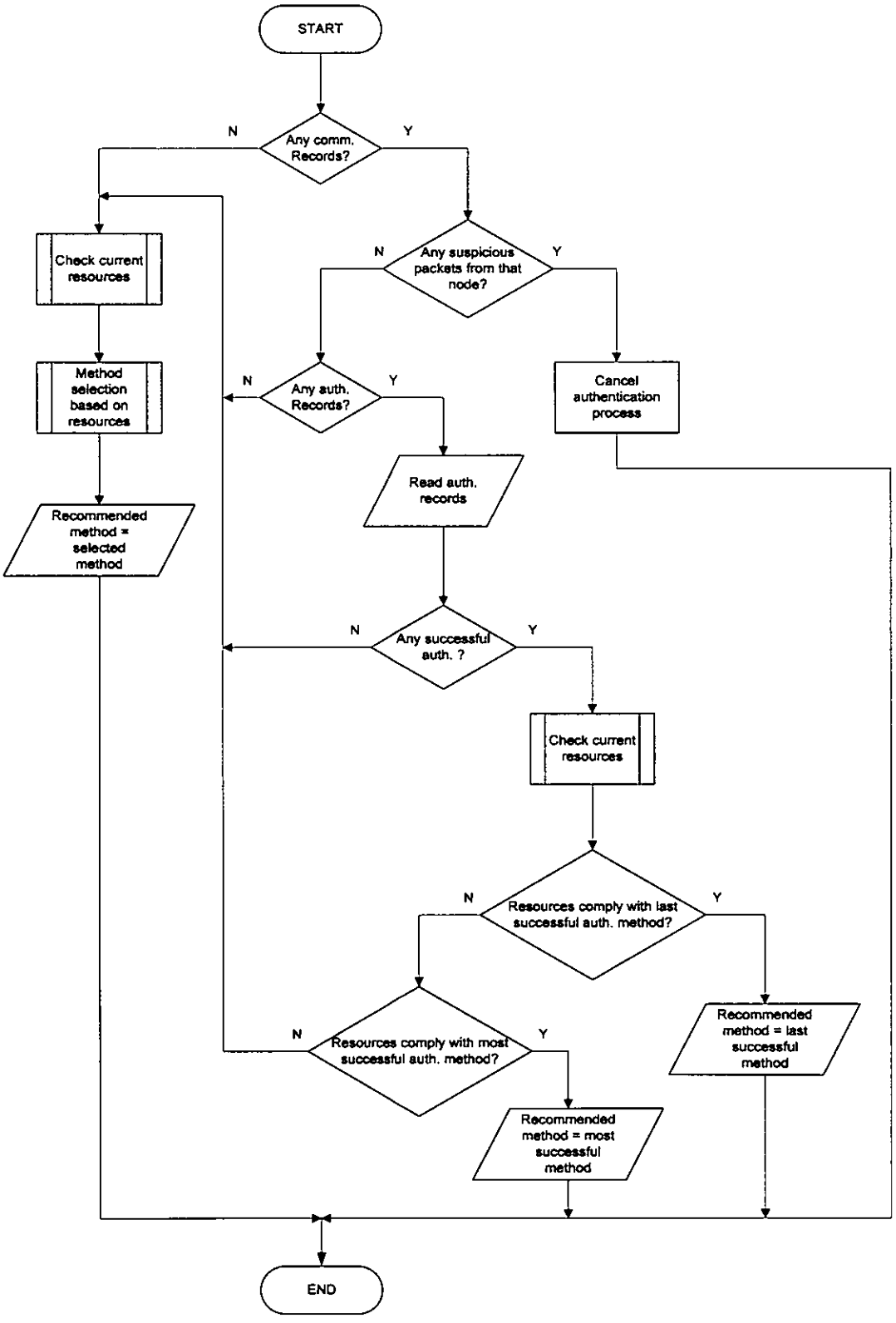


Figure 3-9: Flowchart of the EAP method selection mechanism

The following is the pseudo code of the EAP method selection algorithm based on node's current resources, and the flowchart is illustrated in Figure 3-10:

```
1. Start
2. if (certificate exists) then    //requirement for EAP-TLS
3.     method = EAP-TLS
4. else if (OS == MS Windows) then {
5.     if (Windows is or above Windows XP) then    //requirement for PEAP
6.         method = PEAP
7.     else
8.         method = EAP-TTLS
9. }
10. else
11.     method = EAP-TTLS
12. End
```

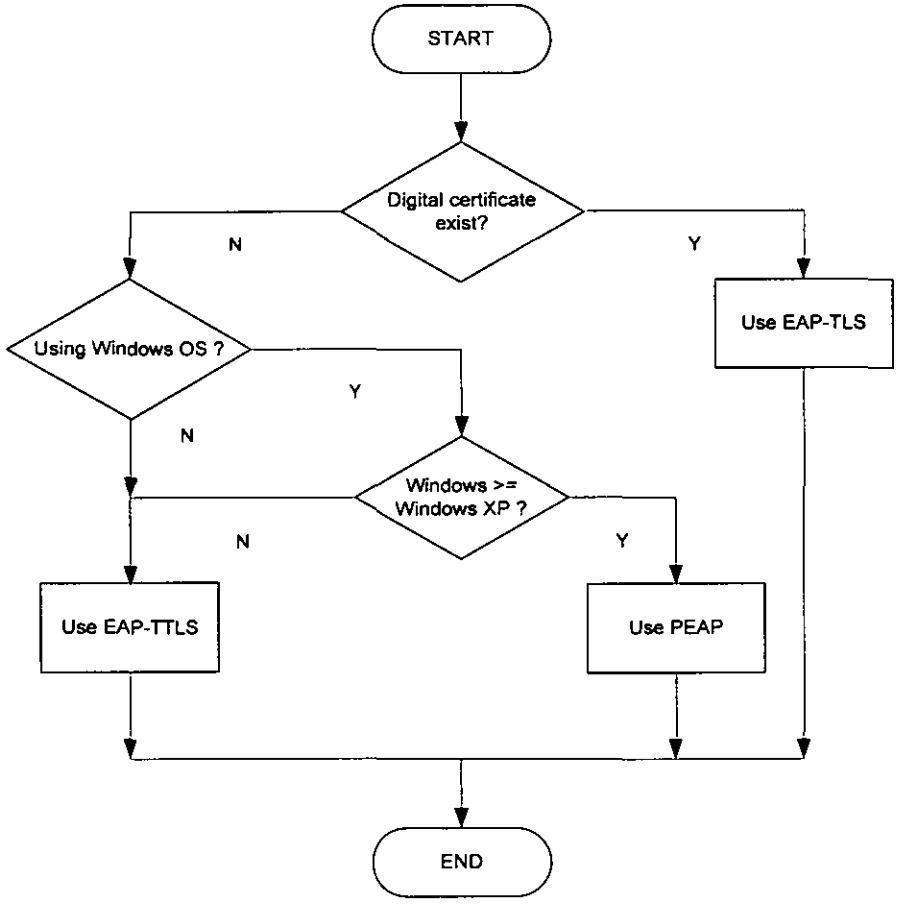


Figure 3-10: Flowchart of the EAP method selection based on node current resources

The nodes will then negotiate to use the method that is suitable for both nodes, or use the lower method in hierarchy out of the two selected methods. As described earlier, the hierarchy is as the following: EAP-TLS, then PEAP, and then EAP-TTLS. The negotiation of the lower method in hierarchy will not compromise the authentication security since the last method in hierarchy (and the most compatible method), i.e. EAP-TTLS, still provides a strong level of security.

3.5 Summary

This chapter has presented the concept and the design of EAP authentication mechanism for ad hoc wireless LAN based on EAP multiplexing model. This chapter also presented the proposed extension to the existing EAP framework / architecture in order to select an EAP method out of a set of EAP methods in heterogeneous wireless devices environment. With the proposed mechanism, it is feasible that the implementation can use the existing TLS-based EAP methods, i.e. EAP-TLS, EAP-TTLS, and PEAP, which provide strong security protection. However, the EAP multiplexing model needs the master node to support all authentication methods, thus the master node might need to have more computational capabilities.

The proposed EAP method selection and negotiation mechanism should be able to select the suitable EAP method for each node and negotiate the one to be used. Thus it should reduce the probability of nodes that could not be authenticated due to unsuitable or unsupported EAP method. This will be elaborated further in Chapter Five.

CHAPTER FOUR : FORMAL SPECIFICATION AND VERIFICATION

In this chapter, BAN Logic [Burrows, et al, 1990] is used to specify and prove the abstract model of the proposed authentication mechanism. BAN Logic is a logic of authentication to express / describe the beliefs of entities / parties / *principals* (people, computers, services) involved in authentication process.

4.1 BAN Logic

BAN Logic was introduced by Michael Burrows, Martin Abadi, and Roger Needham to answer questions with the help of formal method, such as:

- Does the protocol work, or can it be made to work?
- What does the protocol achieve?
- Does the protocol need more assumptions than another protocol?
- Does the protocol do anything unnecessary?

BAN Logic focuses on the beliefs of trustworthy parties involved in authentication protocol and on the evolution of these beliefs as a consequence of communication during authentication process. If proof that a protocol is correct cannot be obtained, then the protocol deserves to be treated with precaution. BAN Logic has been used to analyze and improve many authentication protocols such as in [Anderson, 1997], [Agray, 2001], [Agray et al., 2002], [Chiu-Man, 2002]. Refer to Appendix A for more details about BAN Logic including its formulas and postulates.

The steps in analyzing a protocol using BAN Logic are as follow:

1. The idealized form of the protocol is derived from the original form.
2. Assumptions about the initial states are written.
3. Logical formulas are attached to the statements / messages of the protocol, as assertions about the state of the system after each message.

4. The logical postulates of BAN Logic are applied to the assumptions and the assertions in order to discover the beliefs held by the parties in the protocol.

The BAN Logic considers that authentication process is complete between A and B if there is a K such that:

A believes $A \xleftrightarrow{K} B$, B believes $A \xleftrightarrow{K} B$

Or possibly more than the above state can be achieved, as the following:

A believes B believes $A \xleftrightarrow{K} B$, B believes A believes $A \xleftrightarrow{K} B$

Some public key protocols are not intended to result in the exchange of shared keys, but instead transfer other data. For example, the interaction of a principal with certification authority (CA) might be intended to transfer a public key, or to establish shared secret or nonce. In our case, the goal of the authentication is to establish shared session key generated from shared nonces, pre-master secret (PMS), and master secret (M).

4.2 Formal Specification

In this section we formally specify the messages exchanged in the proposed authentication mechanism in Chapter Three using BAN Logic and obtain the idealized form.

We refer to the client mobile nodes as 'A' and 'B', the master node as 'S' (Server). Sid is Session ID; Na, Nb, Ns, are nonces (random numbers / values); Ka, Kb, and Ks are public keys; Ka^{-1}, Kb^{-1}, and Ks^{-1} are the related private keys; PMS is pre-master secret, a random string generated by 'A'; Kas and Kab are shared session keys generated from master secret (M) and the nonces. The master secret is a 48-bytes secret calculated from PMS and the nonces.

4.2.1 Initial Phase (Node-to-Master-Node) Authentication

These are the messages in initial phase authentication:

1. A → S: EAPOL_start
2. S → A: request_id
3. A → S: A, Sid
4. S → A: EAP_start
5. A → S: A, Na
6. S → A: Ns, Sid, certificate(S, Ks)
7. A → S: $\{PMS\}_{Ks}$, $\{finished\}_{Kas}$
8. S → A: $\{finished, N's\}_{Kas}$
9. A → S: $\{user\_name, password, N's\}_{Kas}$
10. S → A: EAP_success / EAP_failure

We can omit the messages that do not contribute to the logical properties of the mechanism and any clear text communications since they provide no guarantees of any kind [Burrows et al., 1990], thus we omitted messages 1 – 5, and 10. In message 6, server sends its certificate signed by itself (self-signed certificate) with an assumption that the server is already known to and trusted by the clients as the certificate authority. Pre-master secret (PMS) is transformed into N'a as PMS is a random string generated by A. The 'finished' message should contain the hashed of the master secret and all previous handshake messages [Paulson, 1998], [Dierks & Allen, 1999], thus it is transformed into H(M, Sid, A, Na, S, Ns). N's is a random challenge, considered as another nonce generated by S to be used in subsequent messages. Xa and Ya are user data which is the user name and password pair to be used in client authentication. And if the authentication succeeds, the process will continue with key and certificate generation. The certificate will contain the identity of A, the public key of A, and the certificate validity time, signed with S's private key.

Thus we can obtain the idealized form as the following:

6. S → A: $\{\overset{Ks}{\mapsto} S\}_{Ks}^{-1}$
7. A → S: $\{N'a\}_{Ks}$, $\{H(M, Sid, A, Na, S, Ns)\}_{Kas}$
8. S → A: $\{H(M, Sid, A, Na, S, Ns), N's\}_{Kas}$
9. A → S: $\{\langle Xa\rangle_{Ya}, N's\}_{Kas}$

4.2.2 Operational Phase (Node-to-Node) Authentication

These are the messages in operational phase authentication:

1. A → B: EAPOL_start
2. B → A: request_id
3. A → B: A, Sid
4. B → A: EAP_start
5. A → B: A, Na
6. B → A: Nb, Sid, certificate(B, Kb, Tb), certificate_verify
7. A → B: certificate(A, Ka, Ta), $\{PMS\}_{Kb}$, certificate_request, certificate_verify, $\{finished\}_{Kab}$
8. B → A: $\{finished\}_{Kab}$
9. A → B: EAP_response()
10. B → A: EAP_success / EAP_failure

Again, we omit the messages that do not contribute to the logical properties of the mechanism and any clear text communications since they provide no guarantees of any kind (messages 1 – 5, and 9 – 10). In message 6, B sends its certificate signed by S in initial phase. As mentioned in Chapter Three, in typical client – server TLS handshake process, there is no requirement for server to send 'certificate verify' message to client, only client sends it to server. However in our case, both nodes are clients. Thus both nodes need to send their 'certificate verify' messages, containing all previous handshake messages signed by their private key. Therefore, in message 6 the 'certificate verify' message from B is added.

In message 7, A must send its certificate along with the pre-master secret it generates encrypted with B's public key, its 'certificate verify' message and followed by the 'finished' message. In message 8, B also responds with 'finished' message to confirm that both parties have agreed on the same parameters (secrets and keys).

Thus we can obtain the idealized form as the following:

6. B → A: $\{B, Kb, Tb\}_{Ks}^{-1}$, $\{H(A, Na, B, Nb, Sid)\}_{Kb}^{-1}$
7. A → B: $\{A, Ka, Ta\}_{Ks}^{-1}$, $\{N'a\}_{Kb}$, $\{H(A, Na, N'a, B, Nb, Sid)\}_{Ka}^{-1}$, $\{H(M, Sid, A, Na, B, Nb)\}_{Kab}$
8. B → A: $\{H(M, Sid, A, Na, B, Nb)\}_{Kab}$

4.3 Formal Verification

In this section we make assumptions about the initial state, and then we apply BAN Logic formulas and postulates to the assumptions and the assertions in order to discover the beliefs held by the parties in the authentication scheme.

4.3.1 Initial Phase (Node-to-Master-Node) Authentication

Initial assumptions:

A believes $\overset{Ks}{\mapsto} S$	S believes $\overset{Ks}{\mapsto} S$
A believes # (Na)	S believes # (Ns)
A believes # (N'a)	S believes # (N's)
S believes (A controls $A \overset{N}{\longleftrightarrow} S$)	
A believes $A \overset{Ya}{\Leftrightarrow} S$	S believes $A \overset{Ya}{\Leftrightarrow} S$

A knows the public key of certification agent S, and S knows its own keys. Each principal believes that the nonce they generate is fresh. A will invent a new nonce as pre-master secret and S trusts A to invent good / valid nonce that is likely to make good encryption key. Each principal believes that they shared a secret.

The authentication process analyzed as the following:

Message 6: **S → A:** $\{\overset{Ks}{\mapsto} S\}_{Ks}^{-1}$

Message 6 will give a belief that A can be assured that it is communicating with S since only S can encrypt the message with Ks^{-1}.

A believes $\overset{Ks}{\mapsto} S$

Message 7: **A → S:** $\{N'a\}_{Ks}$, $\{H(M, Sid, A, Na, S, Ns)\}_{Kas}$

A sends message 7 to S. A can be sure that only S can decrypt $\{N'a\}_{Ks}$ and see N'a since only S knows the Ks^{-1}. Therefore A believes that it shares N'a as a secret with S.

A believes $A \overset{N'a}{\Leftrightarrow} S$

Since N'a is the PMS and from it A can calculate master secret M and Kas, therefore we obtain:

A believes $A \overset{M}{\Leftrightarrow} S$

A believes $A \overset{Kas}{\longleftrightarrow} S$

S receives message 7 which will yield:

S $\triangleleft$ $\{N'a\}_{Ks}$

S $\triangleleft$ $\{H(M, Sid, A, Na, S, Ns)\}_{Kas}$

S sees N'a, since S can decrypt $\{N'a\}_{Ks}$ using its private key Ks^{-1}. S then can calculate the master secret M and Kas. Using Kas, S can decrypt $\{H(M, Sid, A, Na, S, Ns)\}_{Kas}$, thus:

S $\triangleleft$ $A \overset{N'a}{\Leftrightarrow} S$

S $\triangleleft$ $A \overset{M}{\Leftrightarrow} S$

S $\triangleleft$ $A \overset{Kas}{\longleftrightarrow} S$

S $\triangleleft$ H(M, Sid, A, Na, S, Ns)

At this point, we can not obtain better belief for S. S still can not be sure yet that it is communicating with A or that message 7 was sent by A recently, since other party, e.g. attacker C, might be able to intercept the messages exchanged between A and S (acts as Man-in-The-Middle). C can replace N'a with N'c, encrypt it with Ks, and send it along with hash of the intercepted messages. Ks is the master node's public key which is likely available in public. Actually, this problem can be addressed using 'certificate verify' from A, a hash of relevant items signed by A. However, in the initial phase, the node might not have a certificate with private key yet.

Message 8: $\mathbf{S \rightarrow A: \{H(M, Sid, A, Na, S, Ns), N's\}_{Kas}}$

S will respond with message 8, sending its 'finished' message and N's. A supposed to receive message 8 and we can obtain:

$A \triangleleft \{H(M, Sid, A, Na, S, Ns), N's\}_{Kas}$

Using message-meaning rule, we can obtain:

A believes S said {H(M, Sid, A, Na, S, Ns), N's}

Since A believes #(Na), thus:

A believes #{H(M, Sid, A, Na, S, Ns), N's}

Using nonce-verification rule, we can obtain:

A believes S believes {H(M, Sid, A, Na, S, Ns), N's}

A believes S believes $A \overset{M}{\Leftrightarrow} S$

A believes S believes $A \overset{Kas}{\longleftrightarrow} S$

If an attacker C intercepts message 8, C will not be able to pass this message to A since the 'finished' message will be different with the one calculated by A because C cannot obtain the value of N'a from message 7 (it is encrypted with Ks and can only be decrypted with Ks^{-1}), thus C cannot calculate the correct M and 'finished' message. If A does not receive the correct 'finished' message within a time frame, it will disconnect and abort the authentication process session. Thus there is no security breach.

Message 9: **A → S: $\{\langle Xa\rangle_{Ya}, N's\}_{Kas}$**

A responds with message 9, sending its identity and secret along with N's. S receives it, which will give:

S ◁ $\{\langle Xa\rangle_{Ya}, N's\}_{Kas}$

Since S believes $A \overset{Ya}{\Leftrightarrow} S$, $\langle Xa\rangle_{Ya}$ serves as proof of A's identity. Thus we obtain:

S believes A said ($\{\langle Xa\rangle_{Ya}$, N's, $A \overset{Kas}{\longleftrightarrow} S$)

S believes A said($A \overset{Kas}{\longleftrightarrow} S$)

Since S believes #(N's), thus:

S believes #($A \overset{Kas}{\longleftrightarrow} S$)

Using nonce-verification rule, we obtain:

S believes A believes $A \overset{Kas}{\longleftrightarrow} S$

And finally, using jurisdiction rule, we obtain:

S believes $A \overset{Kas}{\longleftrightarrow} S$

Attacker C will not be able to create message 9 since C needs to provide A's identity and secret, i.e. $\langle Xa\rangle_{Ya}$, which is unlikely to be obtained by C. Therefore, Man-in-The-Middle (MiTM) and replay attacks will not work in the authentication scheme.

The final beliefs of the initial phase authentication are:

A believes $A \overset{Kas}{\longleftrightarrow} S$

S believes $A \overset{Kas}{\longleftrightarrow} S$

A believes S believes $A \overset{Kas}{\longleftrightarrow} S$

S believes A believes $A \overset{Kas}{\longleftrightarrow} S$

4.3.2 Operational Phase (Node-to-Node) Authentication

Initial assumptions:

A believes $\overset{Ka}{\mapsto} A$

A believes $\overset{Ks}{\mapsto} S$

A believes (S controls $\overset{Ka}{\mapsto} A$)

A believes (S controls $\overset{Kb}{\mapsto} B$)

A believes # (Na)

A believes # (N'a)

B believes (A controls $A \overset{N}{\longleftrightarrow} B$)

B believes $\overset{Kb}{\mapsto} B$

B believes $\overset{Ks}{\mapsto} S$

B believes (S controls $\overset{Ka}{\mapsto} A$)

B believes (S controls $\overset{Kb}{\mapsto} B$)

B believes # (Nb)

B believes # (N'b)

Each principal knows the public key of certification agent S, and each knows its own keys. Each principal trusts the certification agent to sign digital certificates. A will invent a new nonce as pre-master secret and B trusts A to invent good / valid nonce. Each principal believes that the nonces they generate are fresh.

The authentication process analyzed as the following:

Message 6: **B → A: $\{B, Kb, Tb\}_{Ks}^{-1}$, $\{H(A, Na, B, Nb, Sid)\}_{Kb}^{-1}$**

We apply message-meaning and jurisdiction rules to message 6 and obtain:

A believes $\overset{Kb}{\mapsto} B$

Applying message-meaning rule to the 'certificate verify' message, we obtain:

A believes B said H(A, Na, B, Nb, Sid)

A believes B said (A, Na, B, Nb, Sid)

Since A believes #(Na):

A believes #(A, Na, B, Nb, Sid)

Using nonce-verification rule, we obtain:

A believes B believes(A, Na, B, Nb, Sid)

Message 7: **A → B: $\{A, Ka, Ta\}_{Ks}^{-1}$, $\{N'a\}_{Kb}$, $\{H(A, Na, N'a, B, Nb, Sid)\}_{Ka}^{-1}$, $\{H(M, Sid, A, Na, B, Nb)\}_{Kab}$**

A responds with message 7, containing its digital certificate, pre-master secret (N'a), its 'certificate verify', and 'finished' messages. A can be sure that only B can decrypt $\{N'a\}_{Kb}$ and see N'a since only B knows the Kb^{-1}. Therefore A believes that it shares N'a as a secret with B.

A believes $A \overset{N'a}{\Leftrightarrow} B$

A believes $A \overset{M}{\Leftrightarrow} S$

A believes $A \overset{Kab}{\longleftrightarrow} B$

B receives message 7. Using message-meaning and jurisdiction rules we obtain:

B believes $\overset{Ka}{\mapsto} A$

B sees N'a, since B can decrypt $\{N'a\}_{Kb}$ using its private key Kb^{-1}. B then can calculate the master secret M and Kab. Using Kab, B can decrypt $\{H(M, Sid, A, Na, S, Ns)\}_{Kab}$, thus:

B $\triangleleft$ $A \overset{N'a}{\longleftrightarrow} B$

B $\triangleleft$ $A \overset{M}{\longleftrightarrow} B$

B $\triangleleft$ $A \overset{Kab}{\longleftrightarrow} B$

B $\triangleleft$ H(M, Sid, A, Na, B, Nb)

Applying message-meaning rule to the 'certificate verify' message, we obtain:

B believes A said H(A, Na, N'a, B, Nb, Sid)

B believes A said (A, Na, N'a, B, Nb, Sid)

Since B believes #(Nb):

B believes #(A, Na, N'a, B, Nb, Sid)

Using nonce-verification rule, we obtain:

B believes A believes(A, Na, N'a, B, Nb, Sid)

B believes A believes $A \overset{N'a}{\Leftrightarrow} B$

B believes A believes $A \overset{M}{\Leftrightarrow} S$

B believes A believes $A \overset{Kab}{\longleftrightarrow} B$

And using jurisdiction rule, we obtain:

B believes $A \overset{Kab}{\longleftrightarrow} B$

As seen in the beliefs obtained from message 7, we can obtain stronger beliefs for B due to the 'certificate verify' from A. Using 'certificate verify' message, B can be sure that the pre-master secret (N'a) came from A. Unlike in the initial phase, in the operational phase the node already has digital certificate thus it can produce 'certificate verify' message.

Message 8: **B → A: $\{H(M, Sid, A, Na, B, Nb)\}_{Kab}$**

B will respond with message 8, sending its 'finished' message. A supposed to receive message 8 and we can obtain:

A $\triangleleft$ $\{H(M, Sid, A, Na, B, Nb)\}_{Kab}$

Using message-meaning rule, we obtain:

A believes B said H(M, Sid, A, Na, B, Nb)

A believes B said (M, Sid, A, Na, B, Nb)

Since A believes #(Na), thus:

A believes # (M, Sid, A, Na, B, Nb)

Using nonce-verification rule, we obtain:

A believes B believes (M, Sid, A, Na, B, Nb)

A believes B believes $A \overset{M}{\Leftrightarrow} B$

A believes B believes $A \overset{Kab}{\longleftrightarrow} B$

The final beliefs of the operational phase authentication are:

A believes $A \xleftrightarrow{Kab} B$

B believes $A \xleftrightarrow{Kab} B$

A believes B believes $A \xleftrightarrow{Kab} B$

B believes A believes $A \xleftrightarrow{Kab} B$

4.4 Summary

This chapter has elaborated the formal specification and verification of the proposed authentication mechanism as a proof of correctness. After applying BAN Logic formulas and postulates, strong final beliefs for the proposed authentication mechanism can be obtained. It indicates that the proposed authentication mechanism can provide secure authentication, and after authentication the two parties are entitled to believe that they are communicating with each other and not with intruder.

CHAPTER FIVE : SIMULATION STUDY

Simulation development study of the proposed authentication mechanism is presented in this chapter. A widely used open source network simulation tool was chosen to design the simulation prototype. The simulation results were analyzed and discussed.

5.1 Network Simulators

Some of the network simulator tools that have been surveyed and studied in this research are discussed in this section. These network simulator tools have been widely used by research community for research purposes. They are: ns-2, GloMoSim, OPNET, and OMNeT++.

5.1.1 Network Simulator – ns-2

Network Simulator (NS) is a discrete event simulator targeted at networking research under GPL (General Public License) license. NS provides substantial support for simulation of almost all variants of TCP, routing (including several ad hoc routing protocols and propagation models), data diffusion, and multicast protocols over wired and wireless (local and satellite) networks.

NS began as a variant of the REAL network simulator in 1989 and has evolved substantially over the past few years. In 1995 ns development was supported by DARPA (Defense Advanced Research Projects Agency) through the VINT (Virtual InterNetwork Testbed) project at LBL, Xerox PARC, UC Berkeley, and USC/ISI. Currently ns development is supported through DARPA with SAMAN (Simulation Augmented by Measurement and Analysis for Networks) and through NSF (National Science Foundation) with CONSER (Collaborative Simulation for Education and Research). Both are in collaboration with other researchers including ICIR (The ICSI Center for Internet Research), formerly ACIRI (AT&T Center for

Internet Research at ICSI: International Computer Science Institute). NS has included substantial contributions from other researchers, including wireless code from the UC Berkeley Daedelus and CMU Monarch projects and Sun Microsystems.

5.1.2 GloMoSim

Global Mobile Information Systems Simulation Library (GloMoSim) provides a scalable simulation environment for wireless and wired network systems. It is being designed using the parallel discrete-event simulation capability provided by Parsec (Parallel Simulation Environment for Complex Systems) of UCLA Parallel Computing Laboratory. GloMoSim currently supports protocols for a purely wireless network. In the future, it will be added with functionality to simulate a wired as well as a hybrid network with both wired and wireless capabilities.

GloMoSim is built using layered approach that is similar to the OSI (Open System Interconnection) seven layers network architecture, same as most of network systems. Standard API (Application Programming Interface) will be used between the different simulation layers, which will allow rapid integration of models developed at different layers by different people. The protocols being shipped with the GloMoSim library are shown in Table 5.1.

Table 5-1: Protocols implemented in GloMoSim (adapted from [GloMoSim])

Layers	Protocols
Mobility	Random waypoint, Random drunken, Trace based
Radio Propagation	Two ray and Free space
Radio Model	Noise Accumulating
Packet Reception Model	SNR bounded, BER based with BPSK/QPSK modulation
Data Link (MAC)	CSMA, IEEE 802.11 and MACA
Network (Routing)	IP with AODV, Bellman-Ford, DSR, Fisheye, LAR scheme 1, ODMRP, WRP
Transport	TCP and UDP
Application	CBR, FTP, HTTP and Telnet

5.1.3 OPNET

OPNET Technologies is one of the providers of solutions for network modeling and simulation, application performance management, network configuration management, network capacity planning and management, network engineering and operations, network research and development. Some of OPNET network simulation tool solutions are the OPNET IT Guru and OPNET Modeler, which are licensed and commercial products. OPNET also provides a free version of OPNET IT Guru, i.e. OPNET IT Guru Academic Edition, for academic introductory level networking courses.

5.1.4 OMNet++

The Objective Modular Network Testbed in C++ (OMNeT++) is a public and open source, free (for academic and non-profit use), component-based, modular and open-architecture discrete event simulation environment with strong Graphical User Interface (GUI) support and an embeddable simulation kernel. Its primary application area is the simulation of communication networks. Due to its generic and flexible architecture, it has been successfully used in other areas like the simulation of IT systems, queuing networks, hardware architectures, and business processes as well. OMNet++ allows modeling systems which can be mapped into components / modules that communicate by passing messages.

OMNet++ is rapidly becoming a popular simulation platform in the scientific community as well as in industrial settings. Several open source simulation models and frameworks have been published, in the field of internet simulations (IP, IPv6, MPLS, etc), mobility and ad-hoc simulations, and other areas. One of the simulation model frameworks developed for OMNet++ is the INET Framework. INET Framework is suited for simulation of wired, wireless, and ad-hoc networks. Other than IP and TCP/UDP protocols, there are 802.11, Ethernet, PPP, IPv6, OSPF, RIP, MPLS with LDP and RSVP-TE signaling, and several other protocols.

5.2 Simulation Design of EAP Authentication

After extensively studying and searching the available frameworks, modules, and simulations, unfortunately we were unable to find anything equipped with EAP framework. Initially, we intended to develop the EAP framework ourselves. However later on it was realized that to simulate EAP authentication, many other frameworks of simulator are required. These include the IEEE 802.1X and IEEE 802.11i frameworks which are also unavailable. It would require a lot of time and efforts to achieve those feats which is beyond constrains of this research's time and resources.

[OMNet++ Wiki] discusses how OMNet++ could be incorporated with a security framework to simulate and analyze security protocols. [Hachana, 2006] surveyed the existing network simulators and then designed a new wireless network simulation environment which will integrate the IEEE 802.11i protocol since the existing network simulation environments are insufficient. However, at the moment of writing, there is no working framework or simulation has been produced yet.

Therefore in this section we designed a model of mobile node that can support EAP framework based on OMNet++ simulation environment. OMNet++ was chosen due to its advantages on open source and GUI support. It is hoped that this design can be used by others to develop the complete simulation of EAP authentication protocol.

5.2.1 Node Modeling

The mobile node design is based on *MobileHost* compound module from OMNet++ INET Framework which models a mobile host with 802.11b wireless card in ad hoc mode. This model contains the IEEE 802.11 implementation, and IP, TCP, and UDP protocols. The *MobileHost* module is illustrated in Figure 5-1.

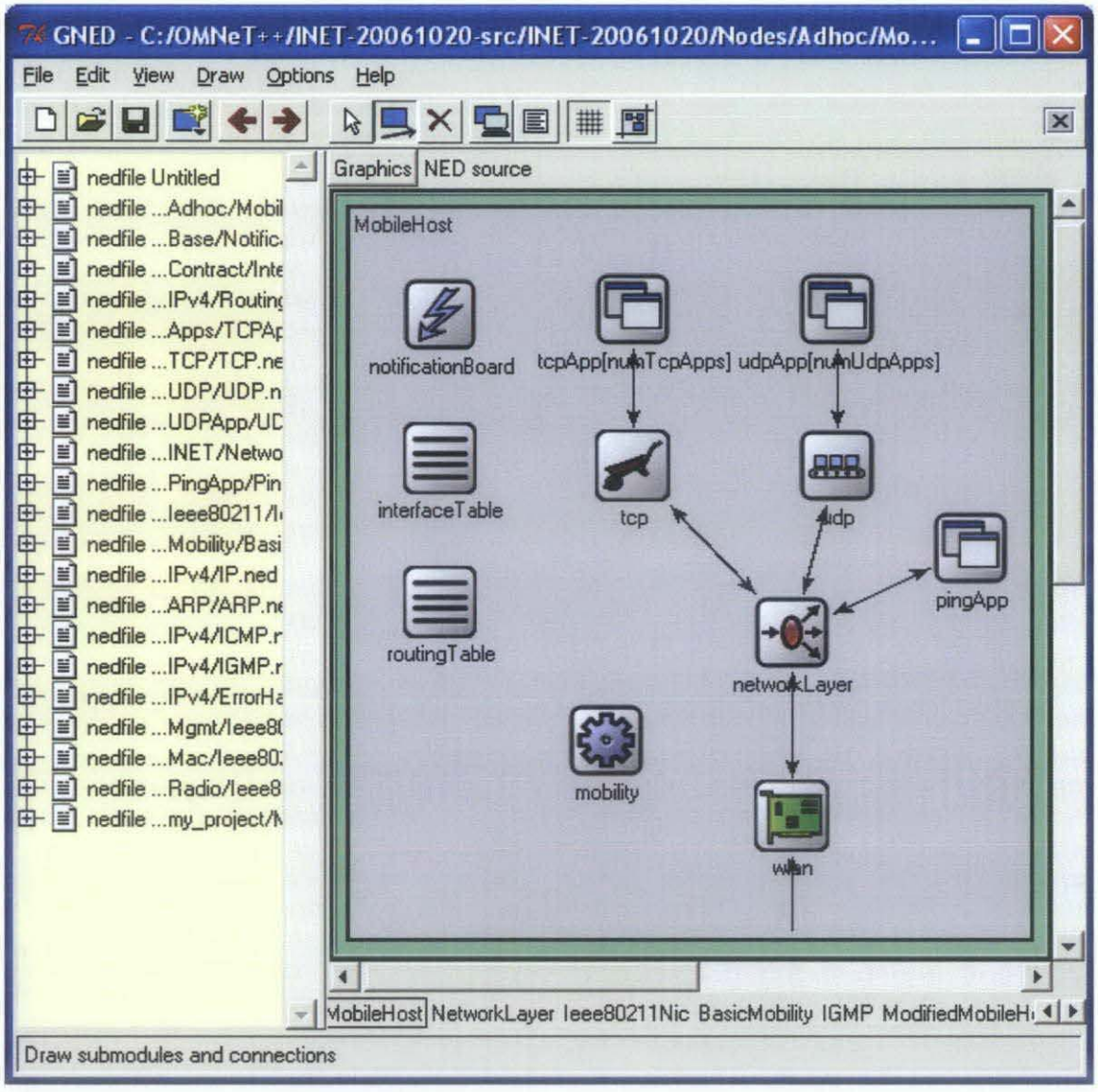


Figure 5-1: *MobileHost* compound module (adapted from [OMNet++ INET])

The *MobileHost* compound module contains the following modules:

- *BasicMobility*: a prototype for mobility models.
- *Ieee80211NicAdhoc*: this NIC (Network Interface Card) module implements an IEEE 802.11 network interface card in ad-hoc mode.
- *InterfaceTable*: keeps the table of network interfaces.
- *NetworkLayer*: network layer of an IP node.
- *NotificationBoard*: using *NotificationBoard*, modules can notify each other about "events" such as routing table changes, interface status changes (up/down),

interface configuration changes, wireless handovers, changes in the state of the wireless channel, mobile node position changes, etc.

- *PingApp*: generates ping requests and calculates the packet loss and round trip parameters of the replies.
- *RoutingTable*: stores the routing table (Per-interface configuration is stored in *InterfaceTable*).
- *TCP*: TCP protocol implementation. Supports RFC 793, RFC 1122, RFC 2001. Compatible with both IPv4 and IPv6.
- *TCPApp*: Template for TCP applications.
- *UDP*: UDP protocol implementation, for IPv4 and IPv6.
- *UDPApp*: Template for UDP applications.

The *NetworkLayer* compound module contains the following modules, as illustrated in Figure 5-2:

- *ARP*: implements the Address Resolution Protocol (ARP) for IPv4 and IEEE 802 6-byte MAC (Medium Access Control) addresses.
- *ErrorHandling*: handles error notifications that arrive from other protocol modules.
- *ICMP*: the Internet Control Message Protocol (ICMP) implementation.
- *IGMP*: placeholder for the Internet Group Management Protocol (IGMP) protocol.
- *IP*: implements the Internet Protocol (IP). The protocol header is represented by the *IPDatagram* message class.

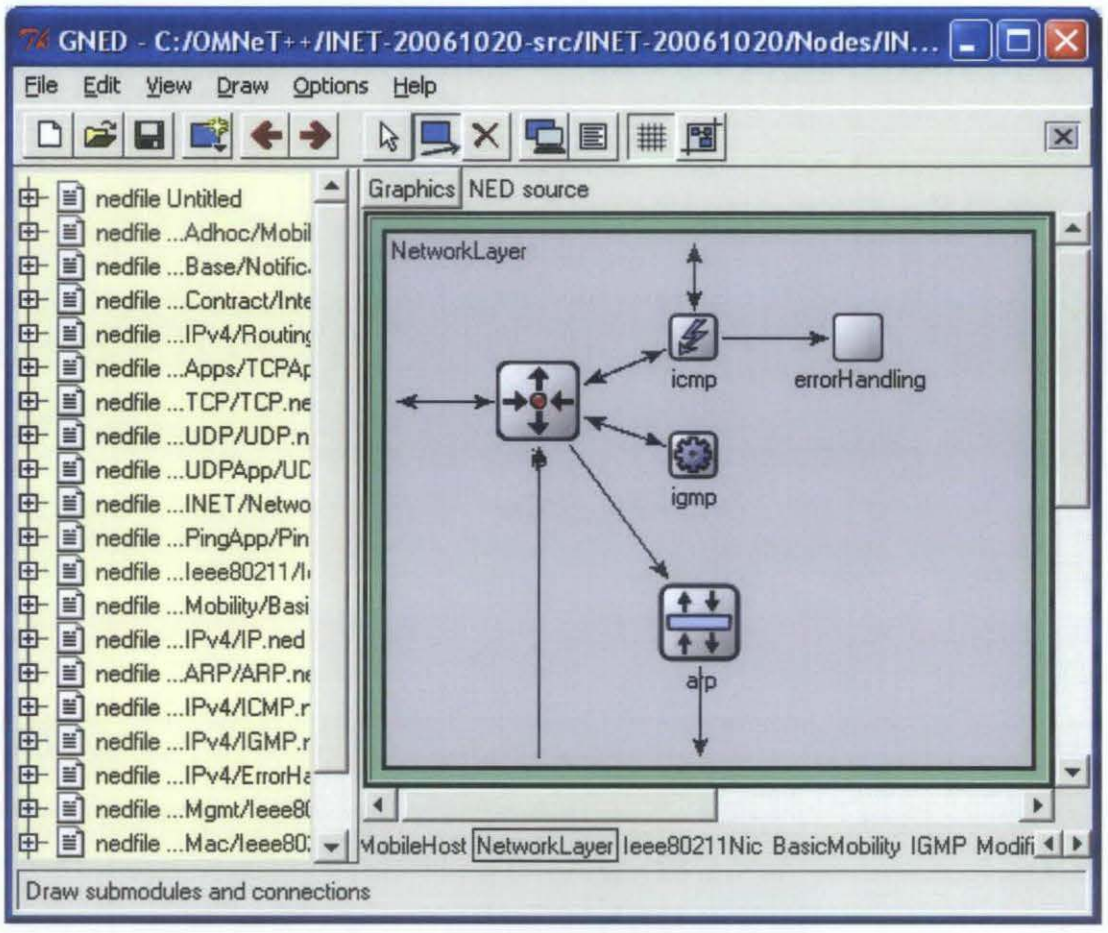


Figure 5-2: *NetworkLayer* compound module (adapted from [OMNet++ INET])

The *Ieee80211NicAdhoc* compound module contains the following modules, as illustrated in Figure 5-3:

- *Ieee80211Mac*: implementation of the 802.11b MAC protocol. This module is intended to be used in combination with the Ieee80211Radio module as the physical layer.
- *Ieee80211MgmtAdhoc*: 802.11 management module used for ad-hoc mode. Relies on the MAC layer (*Ieee80211Mac*) for reception and transmission of frames.
- *Ieee80211Radio*: physical layer for the IEEE 802.11 models. Its external interface (including gates and how it communicates with other modules) is the same as the *Radio* module.

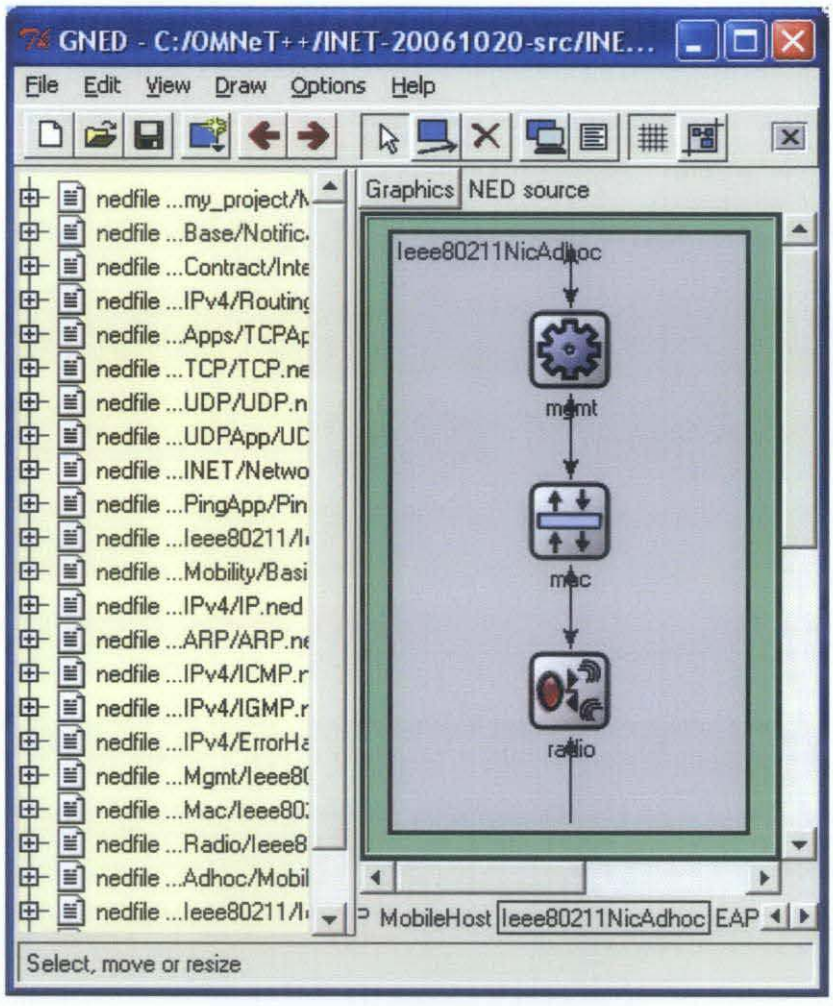


Figure 5-3: *Ieee80211NicAdhoc* compound module (adapted from [OMNet++ INET])

The *MobileHost* compound module needs to be modified to support EAP by adding the EAP module between *Ieee80211NicAdhoc* module and *NetworkLayer* module since EAP resides above the data-link layer. Other modules also need to be added to store communication records, authentication records, digital certificates, and platform / operating system information to support the EAP method selection algorithm. The modified *MobileHost* module is illustrated in Figure 5-4.

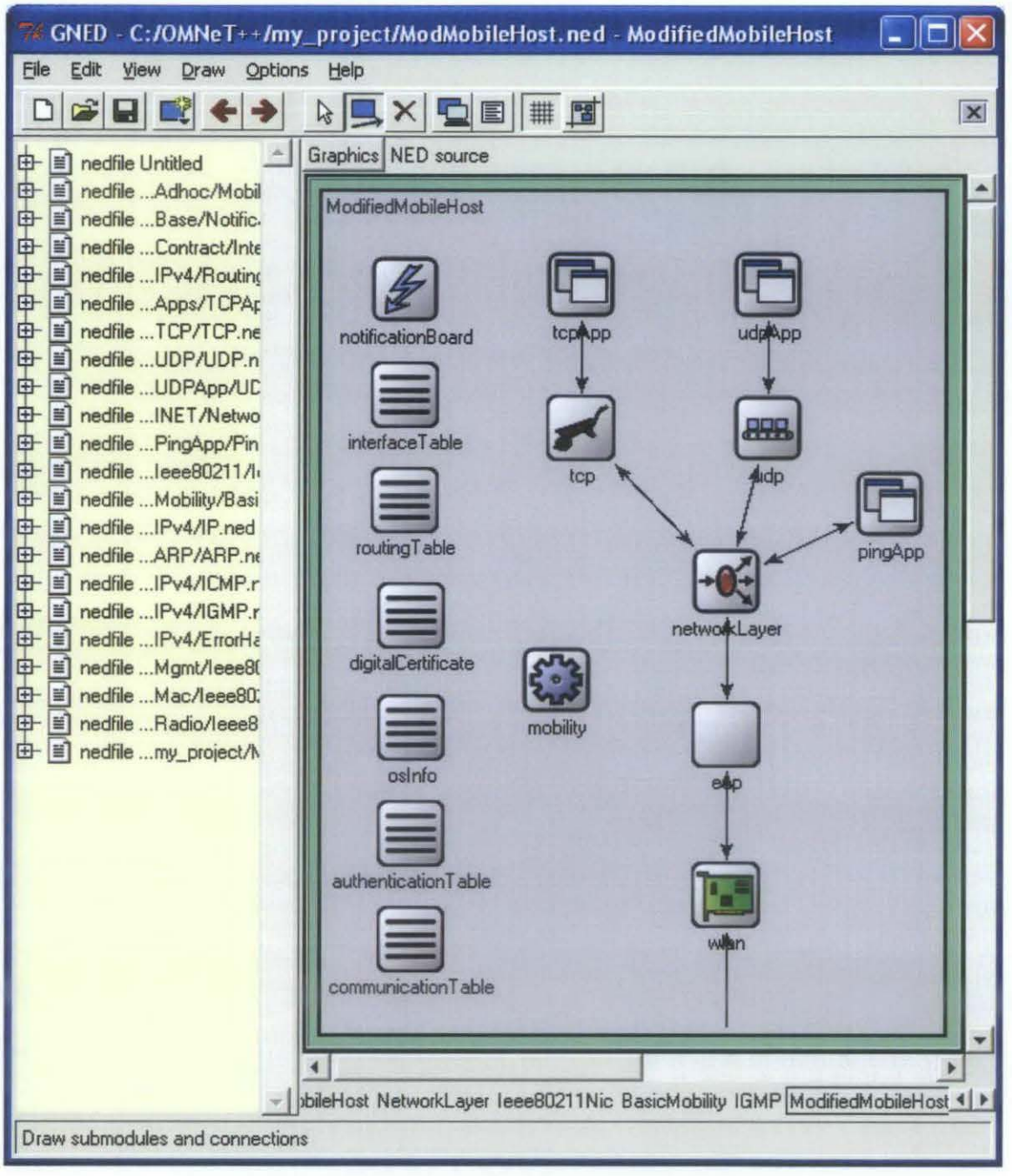


Figure 5-4: Modified *MobileHost* compound module

The *EAP* compound module contains the following modules, as illustrated in Figure 5-5:

- *eapLayer*: implementation of EAP Layer.
- *eapPeerAuth*: implementation of EAP Peer and EAP Authenticator layer.

- *eapMethod*: implementation of EAP Method layer, which contains the EAP methods and authentication algorithms. The EAP method selection algorithm should also be programmed into this module.

For EAP lower layer, the existing data link layer module, i.e. the *Ieee80211NicAdhoc* compound module which implements an IEEE 802.11 network interface card in ad hoc mode, can be used.

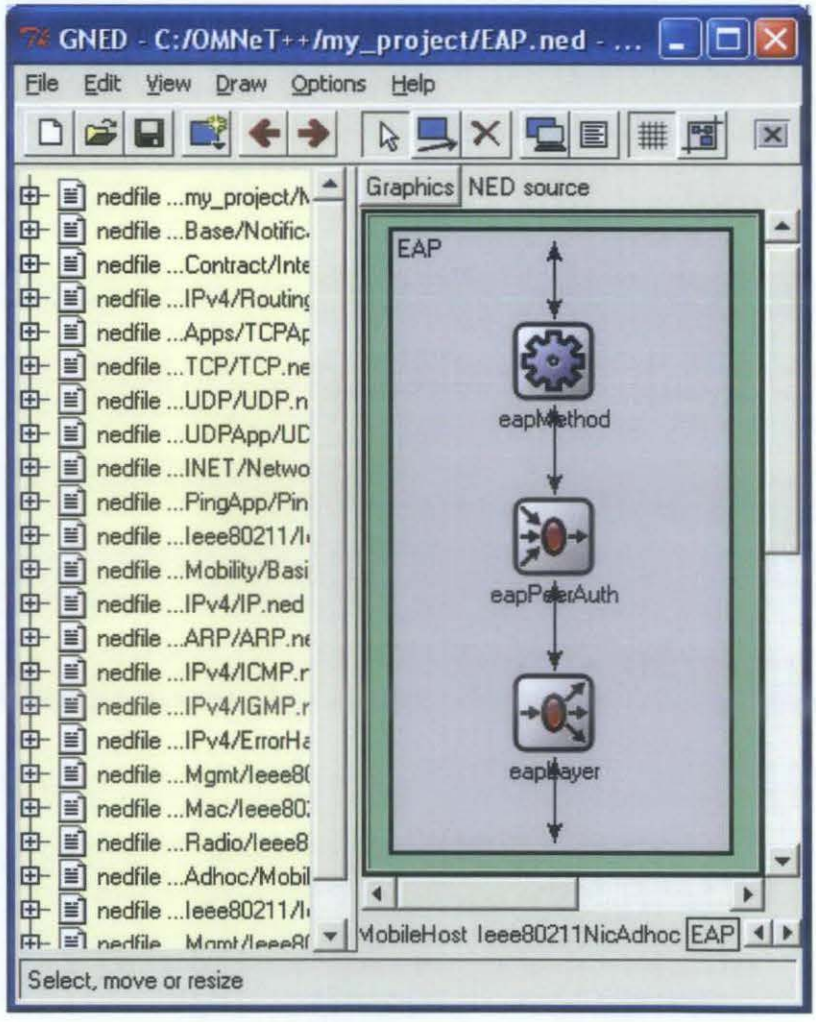


Figure 5-5: *EAP* compound module

5.3 Simulation Development of EAP Method Selection & Negotiation

The ad hoc network configuration for simulation of the EAP method selection is illustrated in Figure 5-6. The network consists of one master node and several different mobile nodes. Each node will be assigned with random resources (operating system and certificate). The master node must have digital certificate since it is the requirement of master node, as discussed in Chapter Three. Each mobile node then will execute the EAP method selection and negotiation algorithm, and attempt to be authenticated to master node using the selected EAP method. Description of the simulation prototyping and some screenshots of the simulation results are presented in Appendix B.

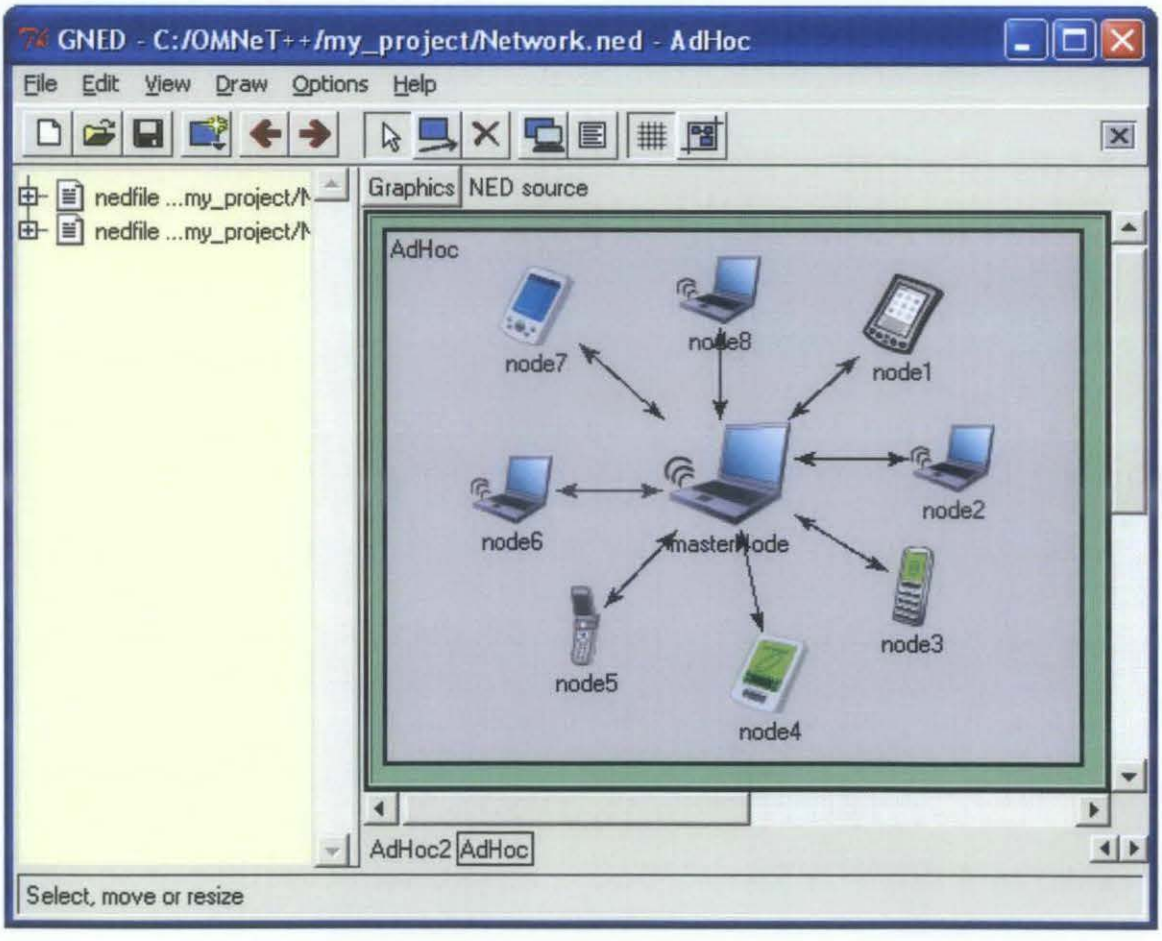


Figure 5-6: Ad hoc network configuration for the simulation

5.4 Results and Discussion

Table 5-2 shows some simulation results of the EAP method selection and negotiation mechanism in node-to-master-node authentication (scenario #1). There are 8 mobile nodes (named as Node 1 to Node 8) and 1 master node (named as Node 0), and the simulation is repeated 10 times. The operating system of the master node is set to Windows Server 2003. The master node will always prioritize the use of EAP-TLS since it has digital certificate.

Table 5-2: Simulation results #1

Nodes	**I**	**II**	**III**	**IV**
Node 1	EAP-TTLS	PEAP	EAP-TTLS	EAP-TLS
Node 2	EAP-TLS	EAP-TLS	PEAP	EAP-TLS
Node 3	EAP-TLS	EAP-TLS	EAP-TLS	PEAP
Node 4	PEAP	EAP-TLS	EAP-TTLS	EAP-TLS
Node 5	EAP-TTLS	EAP-TLS	EAP-TLS	EAP-TTLS
Node 6	EAP-TLS	EAP-TTLS	EAP-TTLS	EAP-TLS
Node 7	EAP-TLS	PEAP	EAP-TLS	EAP-TLS
Node 8	PEAP	EAP-TLS	PEAP	EAP-TLS
Nodes	**V**	**VI**	**VII**	**VIII**
Node 1	EAP-TTLS	EAP-TTLS	EAP-TLS	PEAP
Node 2	EAP-TTLS	EAP-TLS	EAP-TTLS	EAP-TTLS
Node 3	PEAP	EAP-TTLS	EAP-TLS	EAP-TTLS
Node 4	EAP-TLS	EAP-TLS	EAP-TTLS	EAP-TTLS
Node 5	EAP-TLS	EAP-TTLS	PEAP	EAP-TTLS
Node 6	EAP-TTLS	EAP-TLS	PEAP	EAP-TTLS
Node 7	PEAP	EAP-TLS	EAP-TLS	EAP-TTLS
Node 8	EAP-TTLS	EAP-TLS	EAP-TLS	EAP-TLS
Nodes	**IX**	**X**	**% of match**	**% of not match**
Node 1	EAP-TLS	EAP-TLS	0.4	0.6
Node 2	EAP-TLS	EAP-TLS	0.6	0.4
Node 3	EAP-TTLS	EAP-TLS	0.5	0.5
Node 4	PEAP	EAP-TLS	0.5	0.5
Node 5	EAP-TTLS	PEAP	0.3	0.7
Node 6	EAP-TLS	PEAP	0.4	0.6
Node 7	EAP-TLS	EAP-TTLS	0.6	0.4
Node 8	EAP-TLS	EAP-TLS	0.7	0.3

Based on the results of the simulation, due to the random resources assigned to the mobile nodes, there are chances that the EAP method supported by the node is different (not matched) with the EAP method of master node, i.e. EAP-TLS.

Without EAP method negotiation, the node with different EAP method to that of the EAP method of master node cannot be authenticated. However, with the method selection and negotiation mechanism, the master node can select the lower EAP method and match the EAP method supported by the mobile node. The negotiated EAP method is always the same as the mobile node's method. Thus the EAP authentication between master node and the mobile node can be carried out.

For the node-to-master-node authentication scenario above, where master node's operating system is set to Windows Server 2003, the negotiated EAP method is always the same as the mobile node's method whether it is TLS, PEAP, or TTLS because the master node supports all the three methods. However, if master node's operating system is also set to random, the results can be different. One example is shown in the nodes specifications in Table 5-3 and the simulation result in Table 5-4.

Table 5-3: Nodes specifications (node-to-master-node authentication)

Nodes	OS	Certificate
0	Unix	Available
1	Windows 2000	Not Available
2	Linux	Available
3	Windows NT	Not Available
4	Windows 98	Not Available
5	Windows 2003	Not Available
6	Windows ME	Not Available
7	Windows XP	Not Available
8	Windows 98	Available

Table 5-4: Simulation results #2

Nodes	Selected Methods		Negotiated Method
0 & 1	EAP-TLS	EAP-TTLS	EAP-TTLS
0 & 2	EAP-TLS	EAP-TLS	EAP-TLS
0 & 3	EAP-TLS	EAP-TTLS	EAP-TTLS
0 & 4	EAP-TLS	EAP-TTLS	EAP-TTLS
0 & 5	EAP-TLS	PEAP	EAP-TTLS
0 & 6	EAP-TLS	EAP-TTLS	EAP-TTLS
0 & 7	EAP-TLS	PEAP	EAP-TTLS
0 & 8	EAP-TLS	EAP-TLS	EAP-TLS

In the second simulation scenario above, master node's operating system is UNIX. Compared to the first simulation scenario, now the master node only supports TLS and TTLS since UNIX does not support PEAP method. Therefore, if the client mobile node supports PEAP (node 5 & node 7) then the master node will negotiate to use EAP-TTLS instead (see authentications of node 0 & node 5, and node 0 & node 7 in Table 5-3).

In the proposed authentication scheme, the node-to-master-node authentication is followed by the node-to-node authentication. The mobile nodes are given digital certificate at the end of node-to-master-node authentication, thus all mobile nodes will then select and negotiate to use EAP-TLS in the node-to-node authentications. However, that will not be the case if node-to-node authentication occurs without preceded by node-to-master-node authentication. It is shown in the third scenario below where node 1 tries to authenticate the other nodes (node 2 to node 8) without preceded by node-to-master-node authentication. The simulation results are shown in Table 5-5 and Table 5-6.

Table 5-5: Nodes specifications (node-to-node authentication)

Nodes	OS	Certificate
1	Windows NT	Not Available
2	Windows Vista	Available
3	Windows XP	Not Available
4	Windows NT	Available
5	Linux	Not Available
6	Windows ME	Available
7	Windows XP	Available
8	Windows 2000	Not Available

Table 5-6: Simulation results #3

Nodes	Selected Methods		Negotiated Method
1 & 2	EAP-TTLS	EAP-TLS	EAP-TTLS
1 & 3	EAP-TTLS	PEAP	*Insecure*
1 & 4	EAP-TTLS	EAP-TLS	EAP-TTLS
1 & 5	EAP-TTLS	EAP-TTLS	*Insecure*
1 & 6	EAP-TTLS	EAP-TLS	EAP-TTLS
1 & 7	EAP-TTLS	EAP-TLS	EAP-TTLS
1 & 8	EAP-TTLS	EAP-TTLS	*Insecure*

As shown in the results, if one of the two authenticating nodes supports TLS then the authentication can be carried out using the negotiated EAP method, which is similar to the second simulation scenario. However, if neither nodes support TLS, then they cannot use PEAP or TTLS because in order to use PEAP or TTLS at least one node must support TLS. Therefore authentication cannot be carried out securely (marked *insecure*), and the nodes must use other methods of authentication, e.g. MD5 and LEAP, which may be weak, security wise.

5.5 Summary

This chapter has presented the simulation study conducted for this research. Although simulation of the whole EAP authentication process could not be developed due to unavailability of most simulation components of EAP framework, an EAP supporting node design based on OMNet++ simulator has been developed. The EAP method selection mechanism has been simulated using C++ language platform. The results showed that the method selection mechanism is able to select and negotiate the most suitable method for the authenticating nodes. They also showed that in order to execute secure node-to-node authentication, a successful node-to-master-node authentication is required. Otherwise, the authentication process will have to use weak authentication method.

CHAPTER SIX : CONCLUSION AND RECOMMENDATIONS

This final chapter is organized into two sections. The first section provides conclusion of the results and knowledge gained through this research. The second section provides some recommendations for further work.

6.1 Conclusion

This thesis presents the study on EAP-based authentication for ad hoc wireless LAN. The existing EAP implementation models were investigated and then an authentication mechanism for ad hoc wireless LAN based on EAP multiplexing model was proposed. The multiplexing model is found to be more suitable for ad hoc network since it defines two separate entities involved in the authentication without the access point infrastructure.

As implied by the name, Extensible Authentication Protocol, EAP can be extended to support the growing and expanding needs of the network, such as supporting new authentication method or new network type. The EAP framework provides extensible environment where it is possible to customize or grow the framework. It is shown in this work that EAP can be extended to support environment of heterogeneous network devices by adding a mechanism to select and negotiate an EAP method out of a set of EAP methods based on certain parameters.

The simulation framework for EAP authentication process is not yet available in any of the existing network simulators. Thus the proposed authentication mechanism could not be simulated. Instead, specification and verification of the proposed authentication mechanism using BAN Logic was provided as a formal proof of correctness. After applying BAN Logic's postulates, strong final beliefs for the proposed authentication mechanism were obtained. It indicates that the proposed authentication mechanism can provide secure authentication, and after authentication process the two parties are entitled to believe that they are communicating with each other and not with intruder.

The EAP method selection and negotiation algorithm was simulated in this work. The results showed that the method selection and negotiation algorithm is able to select and negotiate the suitable method for the authenticating nodes. The simulation also showed that in order to execute secure node-to-node authentication, a successful node-to-master-node authentication is required. Otherwise, the authentication process will have to use weak authentication method which should not be considered.

EAP has been used as the authentication framework in many types of networks. Therefore, enabling EAP authentication in a network will provide interoperability with other types of networks and enable network users to be authenticated across heterogeneous environment using EAP as the enabling technology, though more works and other protocols will also be required.

6.2 Recommendations for Future Works

There are a number of challenges that still need to be addressed. These may include the followings:

The authentication mechanism proposed in this work did not focus on any ad hoc routing protocol since the work only discussed single hop ad hoc WLAN. Further research is needed to incorporate the proposed authentication mechanism into ad hoc routing protocol and analyze it in multi hop ad hoc networks.

The work presented in this research is tested in an OMNet++ simulation environment with only the basic necessary method selection functionality implemented. It means that the reliability and maturity of the model is not fully evaluated yet since the simulation framework for testing EAP authentication is not yet available. Further work is required to develop a complete EAP simulation framework to include the IEEE 802.1X and 802.11i frameworks. Future works could focus on implementation of EAP authentication in test-bed environment or real ad hoc network, possibly also in extensions of ad hoc network, such as vehicular network and sensor network, incorporating other protocols such as ad hoc routing protocols.

The method selection algorithm can be further researched and extended in order to support more EAP methods. This will require more parameters as inputs for the method selection component that will increase the complexity of the algorithm. Eventually the selection mechanism might require artificial intelligence (AI) implementation, such as Decision Support System (DSS), to yield a more accurate and efficient selection.

The formal verification in this work used BAN Logic which was derived manually by applying the logic formulas and postulates. Future work may use a computer based approach to provide an automated formal proof.

Other possible research subject in order to achieve network interoperability is to study the implementation of EAP across heterogeneous network and to study how users can be authenticated seamlessly across heterogeneous network types. For example wireless LAN / Wi-Fi, WiMAX, and 3G. The solution might involve the use of Protocol for carrying Authentication for Network Access (PANA) as network-layer transport for EAP. PANA will carry EAP which can carry various authentication methods. By PANA's feature of enabling transport of EAP above internet protocol (IP), any authentication method that can be carried as an EAP method is made available to PANA thus to any data link-layer technology. Currently, PANA is still an IETF draft, but it is likely that PANA will be developed further to make it a scalable and practical protocol.

REFERENCES

[3GPP, 2006] *3rd Generation Partnership Project (3GPP) Technical Specification Group Service and System Aspects, 3G Security, Wireless Local Area Network (WLAN) internetworking security (Release 7)*, 3GPP TS 33.234, 2006.

[Aboba & Simon, 1999] B. Aboba, D. Simon, "RFC 2716, PPP EAP TLS Authentication Protocol," The Internet Society, 1999.

[Aboba et al., 2004] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz, Ed., "RFC 3748, Extensible Authentication Protocol (EAP)," The Internet Society, 2004.

[Agray, 2001] Nesria Agray, "The BAN Approach to Formal Verification: Authentication in GSM and SET," Master Thesis, Utrecht University, Leidschendam, 2001.

[Agray et al., 2002] Nesria Agray, Wiebe van der Hoek, and Erik de Vink, "On BAN Logic for Industrial Security Protocols," in *Proceeding of the 2nd International Workshop of Central and Eastern Europe on Multi-Agent Systems (CEEMAS 2001)*, Springer-Verlag, Berlin, Heidelberg, 2002.

[Ali & Owens, 2007] K. M. Ali and T. J. Owens, "Selection of an EAP Authentication Method for a WLAN," *Int. J. Information and Computer Security*, Vol. 1, No. 1/2, pp. 210-233, Inderscience Enterprises Ltd., 2007.

[Anderson, 1997] Ross J. Anderson, "The Formal Verification of a Payment System," Computer Laboratory, Cambridge, UK, 1997.

[Arkko & Haverinen, 2006] J. Arkko, H. Haverinen, "RFC 4187, Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)," The Internet Society, 2006.

[Brian & Hamilton, 2002] Brian R. Miller, Booz Allen Hamilton, "Issues in Wireless Security (WEP, WPA, & 802.11i)", presented at 18th Annual Computer Security Applications Conference, Las Vegas, Nevada, December 2002.

[Burrows et al., 1990] M. Burrows, M. Abadi, R. Needham, "A Logic of Authentication," *ACM Transaction on Computer Systems*, Vol. 8, No. 1, 1990.

[Chandra, 2005] P. Chandra, *Bulletproof Wireless Security: GSM, UMTS, 802.11, and Ad Hoc Security*, Elsevier Inc., 2005.

[Chen & Zhang, 2004] Jyh-Cheng Chen and Tao Zhang, *IP-based Next-Generation Wireless Networks: Systems, Architectures, and Protocols*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2004.

[Chen et al., 2003] Hong Chen, Miroslav Zivkovic, Dirk-Jaap Plas, "Transparent End-User Authentication Across Heterogeneous Wireless Network," IEEE 58th Vehicular Technology Conference (VTC), 2003.

[Cheng et al., 2004] Xiuzhen Cheng, Xiao Huang, and Ding-Zhu Du (Editors), *Ad Hoc Wireless Networking, Network Theory and Applications*, Vol. 14, Kluwer Academic Publishers, 2004.

[Chiu-Man, 2002] Yu Chiu-Man, "Secure Execution of Mobile Agents on Open Networks using Cooperative Agents," Master Thesis, The Chinese University of Hongkong, 2002.

[Dierks & Allen, 1999] T. Dierks, C. Allen, "RFC 2246, The TLS Protocol Version 1.0," The Internet Society, 1999.

[Fluhrer et al., 2001] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," *Selected Areas in Cryptography 2001*, Vol. 2259 of Lecture Notes in Computer Science, pp. 1-24, Springer, 2001.

[Forsberg et al., 2007] D. Forsberg, Y. Ohba (Ed.), B. Patil, H. Tschofenig, A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)," Internet-Draft, The IETF Trust, 2007.

[Funk & Blake-Wilson, 2008] P. Funk, S. Blake-Wilson, "EAP Tunneled TLS Authentication Protocol Version 0 (EAP-TTLSv0)," Internet-Draft, The Internet Trust, March 2008.

[GloMoSim] GloMoSim, Global Mobile Information Systems Simulation Library [Online]. Available: http://pcl.cs.ucla.edu/projects/glomosim/

[Hachana, 2006] Safaa Hachana, "Design and Implementation of A Wireless Communication Networks Simulation Environment," End of Studies Project Report, ENSI, Manouba, Tunisia, 2006.

[Haverinen & Saloway, 2006] H. Haverinen, Ed., J. Saloway, Ed., "RFC 4186, Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)," The Internet Society, 2006.

[Horrigan, 2007] John Horrigan, "Wireless Internet Access," Pew Internet & American Life Project, Feb. 2007.

[Horrigan, 2008] John Horrigan, "Mobile Access to Data and Information," Pew Internet & American Life Project, Mar. 2008.

[IEEE 802.11, 2007] *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11, 2007.

[IEEE 802.11i, 2004] *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 6: Medium Access Control (MAC) Security Enhancements*, IEEE Standard 802.11i, 2004.

[IEEE 802.16e, 2005] *IEEE Standard for Local and metropolitan area networks, Part 16, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operations in Licensed Bands*, IEEE Standard 802.16e, 2005.

[IEEE 802.1X, 2004] *IEEE Standard for Local and metropolitan area networks, Port-Based Network Access Control*, IEEE Standard 802.1X, 2004.

[Ilyas, 2003] Mohammad Ilyas (Editor), *The Handbook of Ad Hoc Wireless Networks*, CRC Press, 2003.

[Kamath et al., 2002] V. Kamath, A. Palekar, M. Wodrich, "Microsoft's PEAP version 0 (Implementation in Windows XP SP1)," Internet-Draft, The Internet Society, Oct. 2002.

[Kambourakis et al., 2004] G. Kambourakis, A. Rouskas, G. Kormentzas, and S. Gritzalis, "Advanced SSL/TLS-based Authentication for Secure WLAN-3G Internetworking," *IEE Proc.-Commun.*, Vol. 151, no. 5, 2004.

[Karnik & Passerini, 2005] Ankush Karnik and Katia Passerini, "Wireless Network Security – A Discussion from A Business Perspective," Wireless Telecommunication Symposium, Pomona, California, 2005.

[Khan & Akbar, 2006] Kaleemullah Khan and Muhammad Akbar, "Authentication in Multi-Hop Wireless Mesh Networks," *Transactions on Engineering, Computing and Technology*, Vol. 16, World Enformatika Society, Nov. 2006.

[Lee & Park, 2003] Jong-Hoon Lee and Ho Jin Park, "A User Authentication Protocol Using EAP for Mobile Ad Hoc Networks," in *Proceedings of the IASTED International Conference: Communication, Network, and Information Security*, New York, USA, 2003.

[Lou & Fang, 2004] Wenjing Lou and Yuguang Fang, "A Survey of Wireless Security in Mobile Ad Hoc Networks: Challenges and Available Solutions," *Ad Hoc Wireless Networking, Network Theory and Applications*, Vol. 14, Kluwer Academic Publishers, 2004.

[Maple et al., 2006] Carsten Maple, Helen Jacobs, and Matthew Reeve, "Choosing The Right Wireless LAN Security Protocol for the Home and Business User," in *Proceedings of the First International Conference on Availability, Reliability, and Security (ARES 2006)*, 2006.

[Microsoft, 2007] Microsoft Corp., "IEEE 802.11 Wireless LAN Security with Microsoft Windows", Jan. 2007 [Online]. Available: http://www.microsoft.com/downloads/

[Moustafa et al., 2005] H. Moustafa, G. Bourdon, Y. Gourhant, "AAA in Vehicular Communication on Highways with Ad hoc Networking Support: A Proposed Architecture," The 2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2005), Cologne, Germany, 2005.

[NETGEAR, 2005] NETGEAR, Inc., "Wireless Networking Basics," v1.0, Oct. 2005.

[Nidjam & Scholten, 2006] Mark Nidjam and Hans Scholten, "Access Point Security Service for wireless ad-hoc communication," Technical Report TR-CTIT-06-66 Centre for Telematics and Information Technology, University of Twente, Enschede, Netherlands, 2006.

[ns-2] The Network Simulator – ns-2 [Online]. Available: http://www.isi.edu/nsnam/ns/

[Nuaymi, 2007] Loutfi Nuaymi, *WiMAX, Technology for Broadband Wireless Access*, John Wiley & Sons, Ltd., 2007.

[OMNet++ Wiki] OMNet++ Wiki, Main / Security Framework Discussion [Online]. Available: http://www.omnetpp.org/pmwiki/index.php?n=Main.SecurityFrameworkDiscussion

[OMNet++] OMNet++, Discrete Event Simulation System [Online]. Available: http://www.omnetpp.org/

[Palekar et al., 2004] A. Palekar, D. Simon, J. Salowey, H. Zhou, G. Zorn, S. Josefsson, "Protected EAP Protocol (PEAP) Version 2," Internet-Draft, The Internet Society, Oct. 2004.

[Pardoe & Snyder Jr., 2005] Terry D. Pardoe and Gordon F. Snyder Jr., *Network Security*, Thomson Delmar Learning, 2005.

[Paulson, 1998] Lawrence C. Paulson, "Inductive Analysis of the Internet Protocol TLS," *Security Protocols*, Springer-Verlag, Berlin, Heidelberg, 1998.

[Perkins, 2001] Charles E. Perkins, *Ad Hoc Networking*, Addison-Wesley, 2001.

[Sankar et al., 2005] Khrisna Sankar, Sri Sundaralingam, Darrin Miller, Andrew Balinsky, *Cisco Wireless LAN Security*, Cisco Press, 2005.

[Stubblefield et al., 2001] A. Stubblefield, J. Ioannidis, and A. D. Rubin, "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP," AT&T Labs Technical Report TD-4ZCPZZ Revision 2, August 2001.

[Toh, 2002] C.-K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems*, Prentice Hall PTR Prentice-Hall, Inc., 2002.

[Zhao et al., 2006] Yao Zhao, Chuang Lin, Hao Yin, "Security Authentication of 3G-WLAN Internetworking," in *Proceeding of the 20th International Conference on Advanced Information Networking and Applications (AINA)*, 2006.

PUBLICATIONS

1. Muhammad Agni Catur Bhakti, Azween Abdullah, Low Tan Jung, "EAP-based Authentication for Ad Hoc Network," in *Proceedings of National Seminar on Information Technology Application (SNATI 2007)*, Yogyakarta, Indonesia, June 2007.

2. Muhammad Agni Catur Bhakti, Azween Abdullah, Low Tan Jung, "EAP-based Authentication with EAP Method Selection Mechanism," in *Proceedings of International Conference on Intelligent and Advanced Systems (ICIAS 2007)*, Kuala Lumpur, Malaysia, November 2007.

3. Muhammad Agni Catur Bhakti, Azween Abdullah, Low Tan Jung, "EAP-based Authentication with EAP Method Selection Mechanism: Simulation Design," in *Proceedings of The 5th Student Conference on Research and Development (SCOReD 2007)*, Permata Bangi, Malaysia, December 2007.

4. Muhammad Agni Catur Bhakti, Azween Abdullah, Low Tan Jung, "Simulation of EAP Method Selection and Negotiation Mechanism," accepted to be presented at *International Symposium on Information Technology (ITSIM 2008)*, Kuala Lumpur, Malaysia, August 2008.

APPENDIX A : BAN LOGIC

A.1 Basic Symbols / Notation

In BAN logic, there are three objects distinguished: principals (parties involved in authentication protocol), encryption / decryption keys, and formulas (called statements). Messages are identified with statements.

Typically, BAN logic uses these symbols:

- A, B, S : denote specific principals
- K_{ab}, K_{as}, K_{bs} : denote shared key between principals
- K_a, K_b, K_s : denote public key of principal
- K_a^{-1}, K_b^{-1}, K_s^{-1} : denote the corresponding secret / private key
- N_a, N_b, N_s : denote specific statements (nonce, etc)

The symbols P, Q, and R range over principals; X and Y range over statements; and K ranges over encryption / decryption keys.

The only propositional connective in BAN Logic is conjunction, denoted by a comma, and properties such as associative and commutative are also taken for granted. In addition to conjunction, the following constructs are used:

- **P *believes* X**: P believes X, or P would be entitled to believe X. The principal X may act as though X is true.
- $P \triangleleft X$: **P *sees* X**: P can read and repeat X (possibly after doing some decryption).
- **P *said* X**: P once said X. The principal P at some time sent a message including the statement X.
- $P \Rightarrow X$: **P *controls* X**: P has jurisdiction over X. The principal P is an authority on X and should be trusted on this matter.
- ***#(X): fresh*(X)**: the formula X is fresh, in a way that X has not been sent in message at any time before the current of the protocol.

- $P \xleftrightarrow{K} Q$: P and Q may use the shared-key K to communicate. The key K is good, in that it will never be discovered by any principal except P or Q, or a principal trusted by either P or Q.
- $\overset{K}{\mapsto} P$: P has K as public key. The matching private key (K^{-1}) will never be discovered by any principal except P, or a principal trusted by P.
- $P \overset{X}{\Leftrightarrow} Q$: The formula X is a secret known only to P and Q, and possibly to principals trusted by them. An example of a secret is a password.
- $\{X\}_K$: This represents formula X encrypted under the key K.
- $\langle X \rangle_Y$: This represents X combined with formula Y. It is intended that Y be a secret and that its presence is proof of origin for X.

A.2 Logical Postulates

(1) The message-meaning rules concern the interpretation of messages. They explain how to derive beliefs about the origin of messages.

For shared keys:

$$\frac{P\ believes\ Q \xleftrightarrow{K} P,\ P \triangleleft \{X\}_K}{P\ believes\ Q\ said\ X}$$

For public keys:

$$\frac{P\ believes \overset{K}{\mapsto} Q,\ P \triangleleft \{X\}_{K^{-1}}}{P\ believes\ Q\ said\ X}$$

For shared secrets:

$$\frac{P\ believes\ Q \overset{Y}{\Leftrightarrow} P,\ P \triangleleft \langle X \rangle_Y}{P\ believes\ Q\ said\ X}$$

(2) The nonce-verification rule expresses the check that a message is recent, thus the sender still believes in it:

$$\frac{P\ believes\ fresh(X),\ P\ believes\ Q\ said\ X}{P\ believes\ Q\ believes\ X}$$

(3) The jurisdiction rule states that if P believes that Q has jurisdiction over X, then P trusts Q on the truth of X:

$$\frac{P\ believes\ Q \Rightarrow X,\ P\ believes\ Q\ believes\ X}{P\ believes\ X}$$

(4) A necessary property of the ***belief*** operator is that P believes a set of statements if and only if P believes each individual statement separately. This justifies the following rules:

$$\frac{P\ believes\ X,\ P\ believes\ Y}{P\ believes\ (X,Y)}, \qquad \frac{P\ believes\ (X,Y)}{P\ believes\ X}, \qquad \frac{P\ believes\ Q\ believes\ (X,Y)}{P\ believes\ Q\ believes\ X}$$

(5) Similar rule applies to the ***said*** operator:

$$\frac{P\ believes\ Q\ said\ (X,Y)}{P\ believes\ Q\ said\ X}$$

(6) If a principal see a formula, then he also sees its components, provided he knows the necessary keys:

$$\frac{P \triangleleft (X,Y)}{P \triangleleft X}, \qquad \frac{P \triangleleft \langle X \rangle_Y}{P \triangleleft X}, \qquad \frac{P\ believes\ Q \overset{K}{\longleftrightarrow} P,\ P \triangleleft \{X\}_K}{P \triangleleft X}$$

$$\frac{P\ believes \overset{K}{\mapsto} P,\ P \triangleleft \{X\}_K}{P \triangleleft X}, \qquad \frac{P\ believes \overset{K}{\mapsto} Q,\ P \triangleleft \{X\}_{K^{-1}}}{P \triangleleft X}$$

(7) If one part of a formula is fresh, then the entire formula must also be fresh:

$$\frac{P\ believes\ \#(X)}{P\ believes\ \#(X,Y)}$$

(8) The same key is used between a pair of principals in either direction. The following two rules reflect this property:

$$\frac{P\ believes\ R \overset{K}{\longleftrightarrow} R'}{P\ believes\ R' \overset{K}{\longleftrightarrow} R}, \qquad \frac{P\ believes\ Q\ believes\ R \overset{K}{\longleftrightarrow} R'}{P\ believes\ Q\ believes\ R' \overset{K}{\longleftrightarrow} R}$$

(9) A secret can also be used between a pair of principals in either direction. The following two rules reflect this property:

$$\frac{P \text{ believes } R \overset{X}{\Leftrightarrow} R'}{P \text{ believes } R' \overset{X}{\Leftrightarrow} R} \qquad \frac{P \text{ believes } Q \text{ believes } R \overset{X}{\Leftrightarrow} R'}{P \text{ believes } Q \text{ believes } R' \overset{X}{\Leftrightarrow} R}$$

Given the above postulates, proofs in the logic can be constructed. A formula X is provable in the logic from a formula Y if there is a sequence of formulas Z_0, ..., Z_n where Z_0 = Y, Z_n = X, and each Z_{i+1} can be obtained from previous ones by the application of a rule.

A.3 The Formalized Goals of Authentication

The BAN Logic deemed that authentication process is complete between A and B if there is a K such that:

A believes $A \overset{K}{\longleftrightarrow} B$, B believes $A \overset{K}{\longleftrightarrow} B$

Some authentication protocols can achieve more than the above:

A believes B believes $A \overset{K}{\longleftrightarrow} B$, B believes A believes $A \overset{K}{\longleftrightarrow} B$

Some public key protocols are not intended to result in the exchange of shared key, but instead transfer other data. For example, the interaction of a principal with certification authority (CA) might be intended to transfer a public key.

$$A \text{ believes } \overset{K}{\mapsto} B$$

Or principals may establish shared secrets or nonces.

$$A \text{ believes } A \overset{Na}{\Leftrightarrow} B$$

APPENDIX B : SIMULATION PROTOTYPING

This appendix presents a brief description of simulation prototyping of the proposed method selection and negotiation mechanism. Some screenshots of the simulation result are also provided.

B.1 Method Selection Simulation Prototype

Figure B-1 illustrates the diagram of method selection and negotiation pilot simulation. *method_Selection* and *method_Negotiation* are the core modules. *random_OS* and *random_Cert* are the node specification manipulation modules. *method_Selection* and *method_Negotiation* modules interact with the authentication records data. *method_Selection* reads authentication records data and gets input from *random_OS* and *random_Cert* modules to obtain the selected method. *method_Selection* modules then delivers the selected method to *method_Negotiation* module. The *method_Negotiation* then stores the negotiated method to authentication records database.

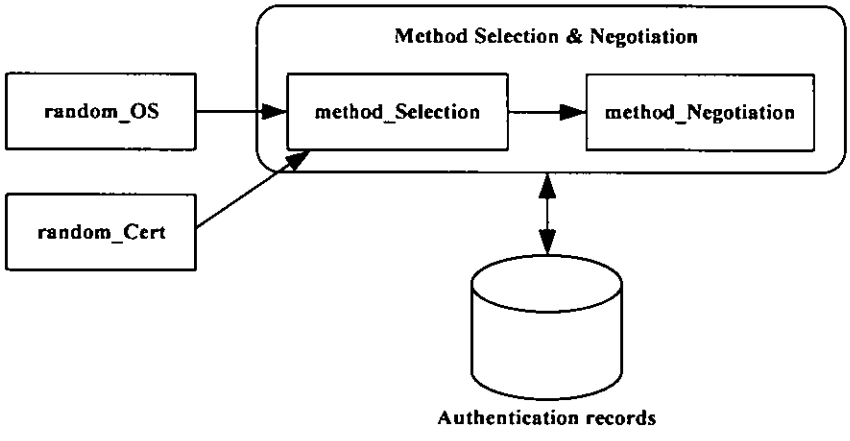


Figure B-1: Simulation prototyping diagram

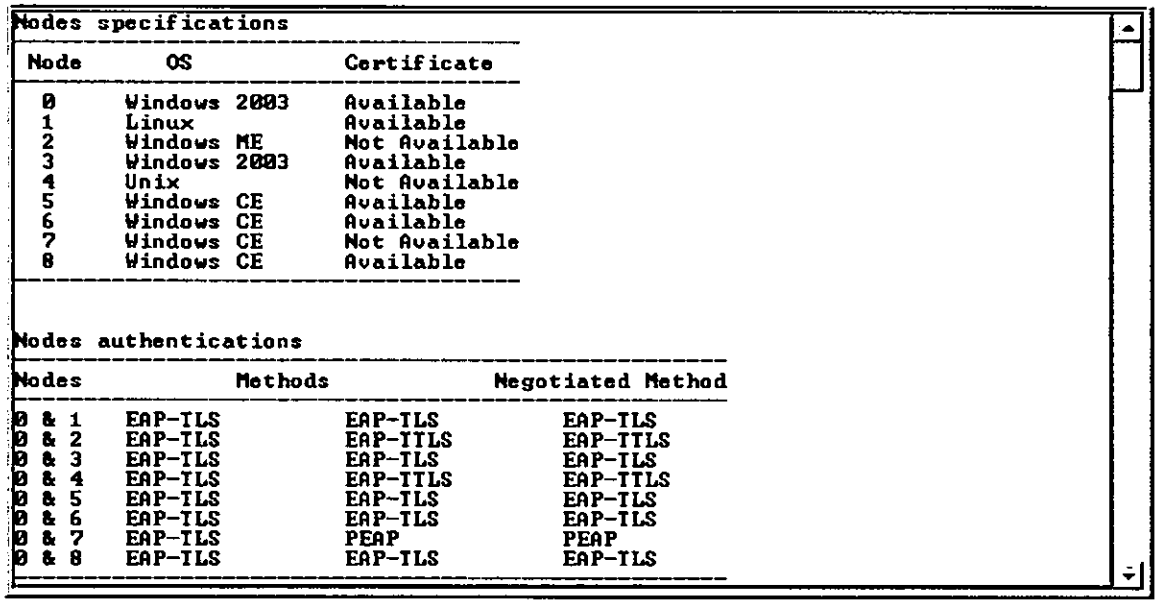
Figure B-2 illustrates skeleton code of the method selection and negotiation simulation which include the node structure, *random_OS* function, and *random_Cert* function.

```
//#include all the needed libraries

struct Node { //structure of node
  void init() { //node initialization function
       //assigning random operating system
       int random_OS() {
           //insert code here
       }

       //assigning random certificate availability
       bool random_Cert() {
            //insert code here
       }
  }
}

string method_Selection($inputData) { //method selection function
     //insert code here
}

string method_Negotiation($inputData) { //method negotiation function
     //insert code here
}

int main() {
  //initialize the nodes
  Node.init();

  //method selection
  method_Selection($inputData);

  //method negotiation
  method_Negotiation($inputData);
}
```

Figure B-2: Skeleton code of the simulation program

B.2 Simulation Screenshots

Figure B-3, Figure B-4, and Figure B-5 show screenshots of some of the simulation results.



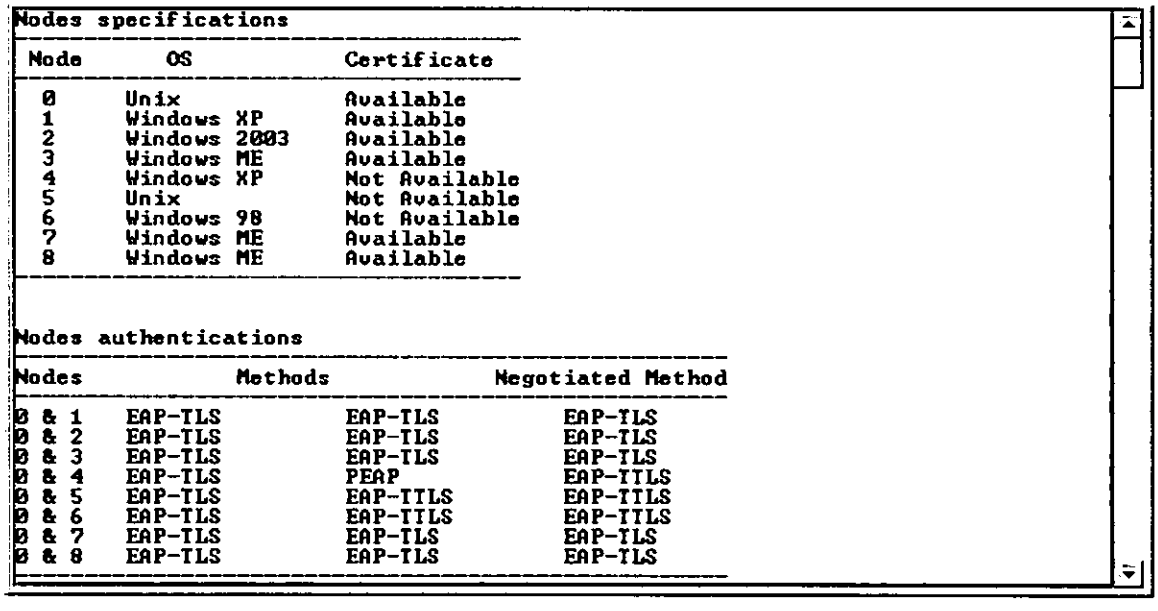
Nodes specifications

Node	OS	Certificate
0	Windows 2003	Available
1	Linux	Available
2	Windows ME	Not Available
3	Windows 2003	Available
4	Unix	Not Available
5	Windows CE	Available
6	Windows CE	Available
7	Windows CE	Not Available
8	Windows CE	Available

Nodes authentications

Nodes	Methods		Negotiated Method
0 & 1	EAP-TLS	EAP-TLS	EAP-TLS
0 & 2	EAP-TLS	EAP-TTLS	EAP-TTLS
0 & 3	EAP-TLS	EAP-TLS	EAP-TLS
0 & 4	EAP-TLS	EAP-TTLS	EAP-TTLS
0 & 5	EAP-TLS	EAP-TLS	EAP-TLS
0 & 6	EAP-TLS	EAP-TLS	EAP-TLS
0 & 7	EAP-TLS	PEAP	PEAP
0 & 8	EAP-TLS	EAP-TLS	EAP-TLS

Figure B-3: Node-to-master-node simulation result (scenario #1)



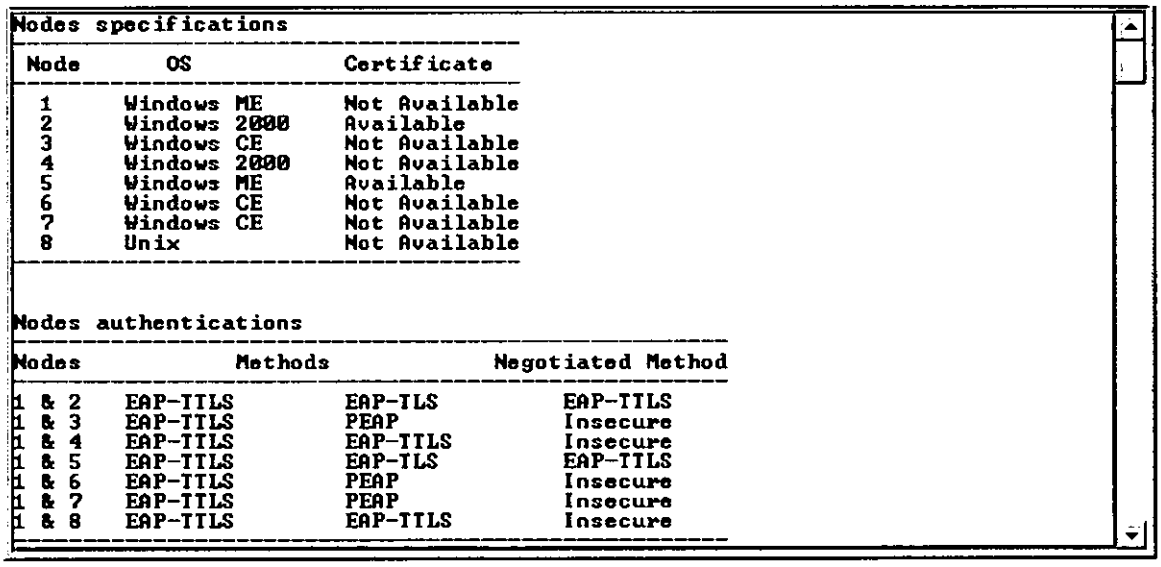
Nodes specifications

Node	OS	Certificate
0	Unix	Available
1	Windows XP	Available
2	Windows 2003	Available
3	Windows ME	Available
4	Windows XP	Not Available
5	Unix	Not Available
6	Windows 98	Not Available
7	Windows ME	Available
8	Windows ME	Available

Nodes authentications

Nodes	Methods		Negotiated Method
0 & 1	EAP-TLS	EAP-TLS	EAP-TLS
0 & 2	EAP-TLS	EAP-TLS	EAP-TLS
0 & 3	EAP-TLS	EAP-TLS	EAP-TLS
0 & 4	EAP-TLS	PEAP	EAP-TTLS
0 & 5	EAP-TLS	EAP-TTLS	EAP-TTLS
0 & 6	EAP-TLS	EAP-TTLS	EAP-TTLS
0 & 7	EAP-TLS	EAP-TLS	EAP-TLS
0 & 8	EAP-TLS	EAP-TLS	EAP-TLS

Figure B-4: Node-to-master-node simulation result (scenario #2)



Nodes specifications

Node	OS	Certificate
1	Windows ME	Not Available
2	Windows 2000	Available
3	Windows CE	Not Available
4	Windows 2000	Not Available
5	Windows ME	Available
6	Windows CE	Not Available
7	Windows CE	Not Available
8	Unix	Not Available

Nodes authentications

Nodes	Methods		Negotiated Method
1 & 2	EAP-TTLS	EAP-TLS	EAP-TTLS
1 & 3	EAP-TTLS	PEAP	Insecure
1 & 4	EAP-TTLS	EAP-TTLS	Insecure
1 & 5	EAP-TTLS	EAP-TLS	EAP-TTLS
1 & 6	EAP-TTLS	PEAP	Insecure
1 & 7	EAP-TTLS	PEAP	Insecure
1 & 8	EAP-TTLS	EAP-TTLS	Insecure

Figure B-5: Node-to-node simulation result (scenario #3)