Toward Authentication Mechanisms for Wi-Fi Mesh Networks

Mohammad Salim Saay

Thesis presented in fulfilment
of the requirements for the degree of
Master of Science
at the University of the Western Cape

Supervisor: Dr William Tucker

May 2011

Declaration

I, Mohammad Salim Saay, declare that this thesis "*Toward Authentication Mechanisms for Wi-Fi Mesh Networks*" is my own work, that it has not been submitted before for any degree or assessment at any other university, and that all the sources I have used or quoted have been indicated and acknowledged by means of complete references.

Signature: .... Date: ....

Mohammad Salim Saay.

Contents

List of Figures

List of Tables

Acknowledgments

This thesis is a compilation of the efforts of many people that helped and guided me through the years. I would first like to thank my supervisor Dr William Tucker for supporting and encouraging me during my Masters study. Without his help, this work would not have been possible. At this time I would like to extend a very special thanks to Prof. I. M. Venter and Verna Connan—without their help I would certainly not be where I am today.

I would also like to thank USAID for their unwavering financial assistance without which our efforts would have been impossible.

We appreciate the help of Dr Jan Plane from the University of Maryland who assisted me with my academic writing in a weekly seminar for developing my English, and who also instructed me in research methods. Thanks to all the members of the Computer Science department of the University of the Western Cape. My thanks also goes to Dr Maria Beebe and Kabul University for planning this program.

Also thanks to Servar Danish, minister of higher education and Prof. Babury, deputy minister, Dr Ashraf Ghani, the previous Chancellor of Kabul University, Prof. Hamidullah Amin the current Chancellor of the University of Kabul for his interest and support, and to Prof. M. Homayoun Naseri, the Dean of the Computer Science Faculty for organizing this program and organizing our trip to South Africa.

Abstract

Wi-Fi authentication mechanisms include central authentication, dynamic and distributed authentication and some encryption methods. Most of the existing authentication methods were designed for single-hop networks, as opposed to multi-hop Wi-Fi mesh networks.

This research endeavors to characterize and compare existing Wi-Fi authentication mechanisms to find the best secure connection mechanism associated with Wi-Fi mesh network fragmentation and distributed authentication. The methodology is experimental and empirical, based on actual network testing. This thesis characterizes five different types of Wrt54gl firmware, three types of Wi-Fi routing protocols, and besides the eight Wi-Fi mesh network authentication protocols related to this research, it also characterizes and compares 14 existing authentication protocols.

Most existing authentication protocols are not applicable to Wi-Fi mesh networks since they are based on Layer 2 of the OSI model and are not designed for Wi-Fi mesh networks. We propose using `TincVPN` which provides distributed authentication, fragmentation, and can provide secure connections for backbone Wi-Fi mesh networks.

Key words

Wi-Fi

Mesh

Real

Network

Distributed

Authentication mechanism

Fragmentation

Characterization

OLPC

PANA

`TincVPN`

Glossary

AAA	Authentication, authorization and availability *Authentication* ensures that an entity is what it claims to be. *Authorization* ensures that an entity is authorized to do an action. *Availability* ensures that authorized access to a resource is available.
AES	Advanced encryption standard
AGE	Authentication on the edge
AODV	Ad-hoc on-demand distance vector
AP	Access point
AS	Authentication server
BATMAN	Better approach to mobile ad-hoc networking
BSSID	Basic service set identifier
CA	Certificate authority
CBC	Cipher block chaining
CBC-MAC	Cipher block chaining message authentication code
CCMP	Counter mode with cipher block chaining message authentication code
CHAP	Challenge handshake authentication protocol
CRC	Cyclic redundancy check
CRL	Certificate revocation list
CTR	Counter mode

DD-WRT	DD-WRT is Linux-based firmware that can be used for Linksys routers
DHCP	Dynamic host configuration protocol
DoS	Denial of service
DSR	Dynamic source routing
DSSS	Direct Sequence spread spectrum
EAP	Extensible authentication protocol
EAP	Extensible authentication protocol
EAP-PANA	Extensible authentication protocol–protocol for carrying authentication for network access
EP	Enforcement point
ESSID	Enhanced service set identifier
ExOR	Extremely opportunistic routing
Freifunk	Freifunk is an extended version of OpenWrt
GIANT	Global-scale Internet access infrastructure
HMAC	Hashed message authentication code
HNA	Host and network association
IAPP	Inter-access point protocol
IBSS	Independent basic services set
ICV	Integrity check value
IEEE	Institute of Electrical and Electronics Engineers
IEEE 802.1X	IEEE standard for port-based network access control
IETF	Internet Engineering Task Force
IV	Initialization vector

IP	Internet protocol
LAN	Local area network
LCMP	Light client management protocol
LCP	Link control protocol
LEAP	Lightweight extensible authentication protocol
MAC	Media access control
MANET	Mobile ad-hoc network
MD5	Message design 5
MIC	Message integrity code
MiTM	Man-in-the-middle
MPR	Multipoint relays
NAS	Network authentication server
NS2	Network simulator 2
NIC	Network interface card
OLSR	Optimized link state routing
OLPC	One laptop per child
OpenWrt	OpenWrt is Linux-based firmware
OSI	Open systems interconnection
OSPF	Open shortest path first
PAA	PANA authentication agent
PaC	PANA client
PANA	Protocol for carrying authentication for network access
PAP	Password authentication protocol

PKI	Public key infrastructure
PMK	Pairwise master key
PPP	Point-to-point protocol
PPPoE	Point-to-point protocol over Ethernet
PSK	Pre-shared key or phase-shift keying
RADIUS	Remote authentication dial in user service
RC4	Rivest cipher 4
RF	Radio frequency
RSA	Rivest Shamir Adelman
SRP	Secure remote password
SSID	Service set identifier
TAP	Transient access point
TCP	Transmission control protocol
TBRPF	Topology broadcast based on reverse-path forwarding
TKIP	Temporal key integrity protocol
TLS	Transport layer protocol security
TTL	Time To Live
TTLS	Tunnel transport layer protocol security
UDP	User datagram protocol
USB	Universal serial bus
WAN	Width area network
WEP	Wired equivalent privacy
WDS	Wireless distributed system

Wi-Fi	Wireless fidelity
WiMAX	Worldwide interoperability for microwave access
WLAN	Wireless local area network
WMAN	Wireless metropolitan area network
WMC	Wireless mesh client
WMN	Wireless mesh network
WMR	Wireless mesh router
WPAN	Wireless personal area network
WPA	Wi-Fi protected access

Chapter 1

Introduction

Wireless fidelity (Wi-Fi) networks have become ubiquitous but are mostly dependent on central access points or routers. Wi-Fi mesh networks extend a Wi-Fi network's accessibility. Authentication on single-hop Wi-Fi networks has been standardized. We tackle the problems of authentication in multi-hop Wi-Fi mesh networks in this thesis. Besides describing the background to the thesis in Section 1.1, this chapter motivates the thesis topic in Section 1.2 and poses research questions in Section 1.3. The approach of the thesis regarding methods, design of and implementation of the project is discussed in Section 1.4 and the complete thesis is outlined in Section 1.5.

1.1 Background

Wireless networks can provide easy and inexpensive network connections for different environments (Akyildiz and Wang, 2005). Providing network connections for cities is not very difficult. We can provide urban network connections with cables as well, but networks with cables can be very difficult to install in rural areas. As wireless networks can easily provide network connectivity for rural and difficult environments, and can bridge the digital gap between rural and urban areas, they are more cost efficient and scalable than wired networks. For these reasons wireless networks have become very popular in recent years (Galperin, 2005). Wireless network equipment is available in most countries. Worldwide over 200 companies provide wireless networking equipment (Fourty et al., 2005).

Wireless networks are based on different technologies, such as Wi-Fi and *wireless interoperability for microwave access* (WiMAX) that can be used in diverse areas for various purposes. We will focus on Wi-Fi technology in this thesis. There are different types of wireless network based on their range, such as *wireless local area networks* (WLANs), *wireless personal area networks* (WPANs), and *wireless* *metropolitan area networks* (WMANs) and Wi-Fi (Fourty et al., 2005). Wireless networks are a promising network technology. They have been widely used in *developed* countries. Therefore, due to their good performance and low cost, *developing* countries have also been motivated to use wireless networks (Pentland et al., 2004).

There are many standards defined by the *Institute of Electrical and Electronics Engineers* (IEEE) for wireless networks. These standards include IEEE 802.11/a/b/g/n and 802.16. The term Wi-Fi, generally used for the IEEE 802.11 standard, is a very popular wireless technology. By 2005 42 million Wi-Fi devices had been installed worldwide. Wi-Fi is designed for short distance coverage. It can be distributed in ad-hoc, peer-to-peer, and wireless mesh networks (Akyildiz and Wang, 2005).

Currently Afghanistan has 24 governmental universities, and in 2010 18 of these 24 universities already use wireless networks. The *Ministry of Higher Education in Afghanistan* (MoHE) installed WiMAX for campus-to-campus connections and Wi-Fi for on site connections between faculties (MoHE Afghanistan, 2010). The *one laptop per child* (OLPC) projects that are running for Afghanistan schools (OLPC, 2009), and Village Telco project running in the Bo-kaap in Cape Town and Dili in East Timor, are the some examples of Wi-Fi mesh networks in developing countries (Song, 2011). A good example of a mesh network applied in a developed country is the use of Freifunk in Berlin (Johnson, 2007).

User access management of Wi-Fi *mesh* networks is more difficult than for wired and wireless networks, because the data in wireless mesh networks is transferred through different devices and over networks whose structure is under continual change. *Wi-Fi mesh networks* (WMNs) are more complex than Wi-Fi infrastructure networks and user access control in WMNs is very critical (Akyildiz and Wang, 2005). Wi-Fi networks have different modes, such as *point-to-point* or *peer-to-peer* mode, *point-to-multi-point* or *infrastructure* mode and *multi-point-to-multi-point* or *meshed* mode (Flickenger et al., 2006, p. 54). Each of the Wi-Fi network modes requires a specific mechanism for user access management.

Wireless mesh networks do not have a single point of failure, because if one node experiences a failure another node can be used to pass data. Therefore, wireless mesh networks have better redundancy, reliability, scalability, and flexibility than other types of wireless network. Wi-Fi mesh networks are composed of mesh nodes and mesh clients. Nodes act like a backbone which is connected to clients and other nodes. Each client acts as a router that can pass data to other clients and nodes. The resulting network can transmit data over long distances (Flickenger et al., 2006, pp. 51–56). Wi-Fi mesh networks are very similar to ad-hoc networks (Schollmeier et al., 2002). However ad hoc networks are peer-to-peer networks that do not need any backbone.

The reasons for the popularity of wireless mesh networks include their easy installation, scalability, cost effective deployment, high redundancy, availability, reliability and low cost (Tsai and Chen, 2005). Often wireless mesh networks are deployed in high density areas and areas where cabling is impractical.

An unmeshed Wi-Fi network uses an *access point* (AP) as a central device for communication between network users. Each AP needs to be connected to a wired network as a backhaul, so it has a single point of failure, a higher cost and less scalability than a Wi-Fi mesh network. Nevertheless, in Wi-Fi mesh networks only some nodes need to be connected to a wired network, in order to provide Internet services. There are two types of mesh nodes in the Wi-Fi mesh network namely:

- *Transient access points* (TAPs): Nodes with no wired Internet connection. They simply forward the data from one node to other nodes.

- *Hot spots*: Nodes having a wired Internet connection that can pass data to the Internet.

There are two types of mesh network: the partial mesh network and the full mesh network. A partial mesh network is a mesh network between routers to routers or clients to clients. A full mesh network is a mesh network between all routers and clients. A mesh network may be composed of several ad-hoc networks. Figure 1.1 illustrates the structure of an ad-hoc network and Figure 1.2 shows the structure of a mesh network.

Wi-Fi mesh networks have several benefits. Most of these benefits are related to self-management features of mesh networks. Wi-Fi mesh networks enable self-managing systems. Self-management consists of self-configuration, self-tuning, self-healing, and self-monitoring (Flickenger et al., 2006; Akyildiz and Wang, 2005).

Overall Wi-Fi mesh networks look very promising compared with other standards of wireless networks, even though the security and performance of Wi-Fi mesh networks need some improvement. Different methods have been introduced for wireless network security, such as encryption of data flowing through the network, user access management by filtering the MAC addresses and IP addresses of devices, and user access management by authentication (Flickenger et al., 2006). However, the existing security policies and user access management cannot necessarily provide guarantees on a wireless mesh network.

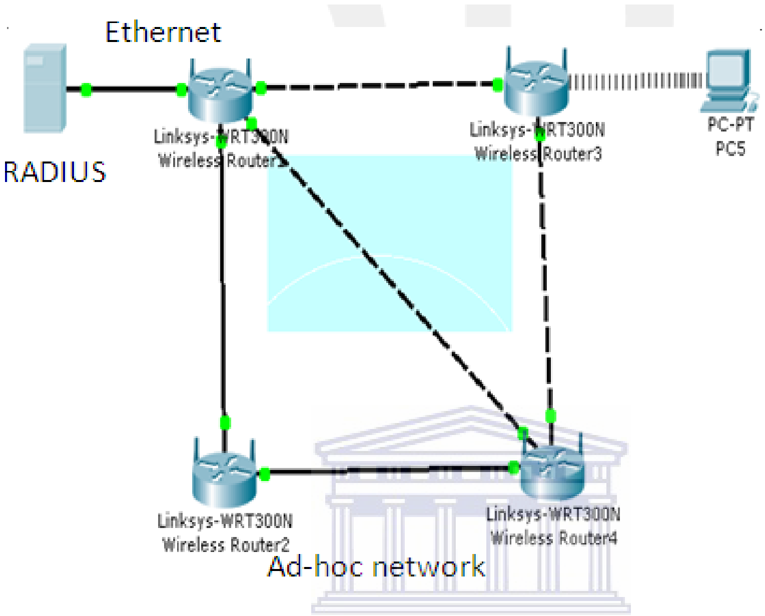


Figure 1.1: An ad-hoc network

Ad-hoc Wi-Fi and Wi-Fi mesh networks are a confusing topic. Ad-hoc Wi-Fi networks are networks where clients such as laptops, PDAs or sensors transfer data to neighboring nodes to form arbitrary network topologies. If we have mobility in such networks, they form a network class known as a *mobile ad hoc network* (MANET). The Wi-Fi topology can change rapidly in MANET network. Wireless sensor networks are a good example of a static Wi-Fi ad-hoc network. Static Wi-Fi mesh networks have dedicated and static wireless routers and clients that carry out the function of routing packets through the networks. Broadband community wireless networks or municipal wireless networks are good examples of Wi-Fi mesh networks (Johnson et al., 2008). The provision of authentication and security can therefore be challenging in static and mobile mesh networking. Our research covers efficient user access mechanisms in infrastructure or backbone Wi-Fi mesh networks. The hardware, software and wireless routing protocol selection process is also important, because these technologies are related to each other.

We are mainly concerned with wireless mesh network routing protocols.

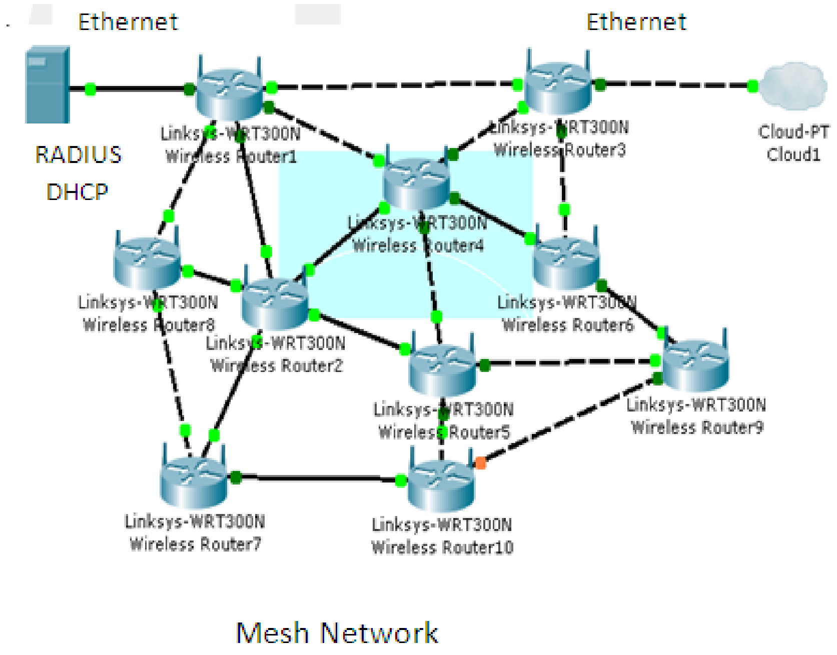

Ethernet
Ethernet
RADIUS
DHCP
Linksys-WRT300N Wireless Router1
Linksys-WRT300N Wireless Router3
Cloud-PT Cloud1
Linksys-WRT300N Wireless Router4
Linksys-WRT300N Wireless Router8
Linksys-WRT300N Wireless Router2
Linksys-WRT300N Wireless Router6
Linksys-WRT300N Wireless Router5
Linksys-WRT300N Wireless Router9
Linksys-WRT300N Wireless Router7
Linksys-WRT300N Wireless Router10
Mesh Network

Figure 1.2: A mesh network

Wireless mesh routing protocols provide the connection facilities, select the best route for packages and transfer the packages to those neighboring mesh points. There are several routing protocols for wireless and wireless mesh networks such as *topology broadcast based on reverse-path forwarding* (TBRPF), *dynamic source routing* (DSR), *extremely opportunistic routing* (ExOR), *optimized link-state routing* (OLSR), *ad-hoc on-demand distance vector* (AODV), and *better approach to mobile ad-hoc networking* (BATMAN).

BATMAN, OLSR and AODV, are the most well-known Wi-Fi mesh routing protocols which are used in the OpenWrt, Freifunk, and DD-WRT. We discuss the characteristics of these protocols briefly below, and review them in more detail in Chapter 4 with respect to authentication mechanisms.

AODV

AODV is a distance vector routing protocol used for mobile ad-hoc networks that can be adapted dynamically. AODV uses resources efficiently and works with slow processors and at low network speeds. AODV is dynamically self starting, uses multi-hop routing and can establish links quickly. It is loop free and converges quickly. AODV uses three types of *user datagram protocol* (UDP) messages: route request, route reply, and route error for communication between ad-hoc nodes (Sun et al., 2003).

AODV is a reactive routing protocol which can find the route when it needs to send packets somewhere. AODV considers the shortest route as the best one. But sometimes the shortest route has more delay, more latency and can use more bandwidth. Usually it is used in small networks because in bigger networks repeated route discovery causes more error reports which results in the loss of bandwidth. The AODV protocol does not employ a system of *multipoint relays* (MPRs). The lack of MPRs results in multi-route advertisement and this results in repeated route discovery and overhead in the network. The AODV protocol needs to discover the route first in order to send the actual data, so the search latency affects the AODV protocol and increases the overhead. The AODV protocol works better in networks with static traffic, and few wireless source and destination points. In bigger networks some other routing protocols are preferable in order to be able to manage more routes and more resources, and in order to consider the best route based on the state of the links (Huhtonen, 2004).

OLSR

OLSR is a wireless network routing protocol which is based on the concept of a link state routing protocol, where each node selects a set of its neighboring MPR nodes to reduce the overhead. The topology of wireless networks continually changes and a routing protocol needs to be compatible with a changeable structure. In the MPR concept, the host must have information about more than one neighbor symmetrically. The MPR is responsible for controlling the traffic during the forwarding of data. OLSR also relies on the MPR calculation to select the shortest path. OLSR has a list of routes with high priority and another list of routes for redundancy and load balancing.

A topology control message is broadcast to all nodes and only the MPR is allowed to forward the message to others. In an OLSR topology, each MPR needs to have at least two neighbors to exchange the message. The message has a sequence number. The message with the highest number has been updated more than others. If a node gets a message with a smaller sequence number then it will be discarded automatically (Clausen and Jacquet, 2003).

The screenshot in Figure 1.3 shows the topology of OLSR for four routers included in the backbone of the experimental network we used for our research which will be addressed in more detail in Section 3.4.2 on mesh network design.

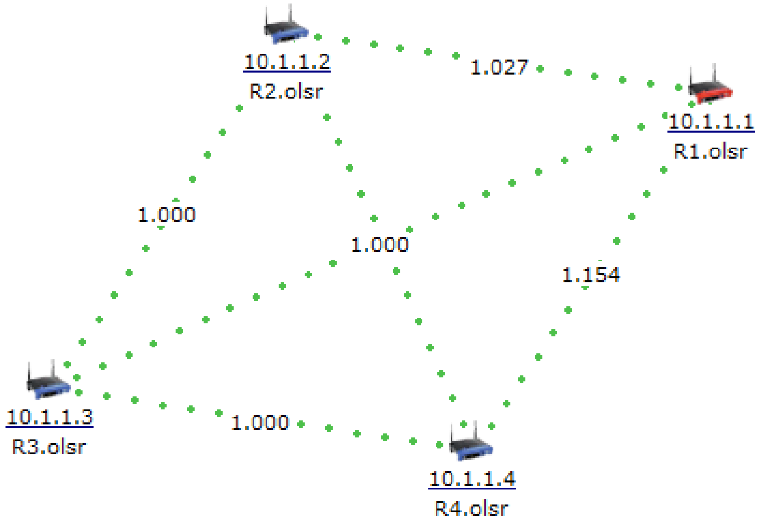


Figure 1.3: OLSR topology

OLSR reduces the overhead of a network because it works with MPRs and MPRs are the only points that can forward messages. Since OLSR maintains the entire route in the routing table, it needs more powerful hardware than AODV. OLSR is mostly used for dense wireless networks such as schools, airports, hospitals and hostels (Ge et al., 2003).

OLSR produces higher routing efficiency than AODV because the updates are done periodically and no additional overhead occurs for finding new routes. In OLSR the overhead is independent of the traffic profiles, so it has a fixed upper bound for the overhead in a network regardless of the network's traffic. OLSR uses more bandwidth and resources than AODV. Thus, OLSR cannot be used in resource critical solutions (Ge et al., 2003). OLSR does not need to do extra work for the discovery of the route so it provides low single packet transmission latency (Huishan et al., 2003).

The one great advantage of the OLSR protocol is that it knows the status of the link and it is possible to extend the quality of service information using this protocol, but AODV does not have this facility (Ge et al., 2003).

BATMAN

Control of the packet and routing in wireless mesh networks is different from other ad-hoc types of network, because wireless mesh networks tend to have less mobility than other types of ad-hoc network. In wireless mesh networks the route changes less than in ad-hoc or mobile ad-hoc networks. BATMAN, which like OLSR, is a proactive routing protocol for establishing multi-hop routes in mobile ad-hoc networks. In this protocol each node maintains information only about the best next hop towards all other nodes, which avoids unnecessary knowledge about the global topology and reduces the signaling overhead.

Based on the BATMAN algorithm, mesh nodes broadcast a *hello* packet to inform neighbor nodes about their existence. The neighbors rebroadcast the hello packet to inform their neighbors about the existence of this node. These small packets contain the address of the original node, the address of the node rebroadcasting the packet, a *time to live* (TTL) and a sequence number. Each node rebroadcasts this packet at most once and only if it is received by the current best next hop towards the original initiator of the packet. BATMAN does not maintain the full route to the destination. Each node along the route maintains the information only about the next link through which the best route can be found. BATMAN never checks the quality of the links, it just checks the link existence. The links are compared in terms of the number of hello packets that have been received within the current sliding window. So the overhead of BATMAN is much less than OLSR (Johnson et al., 2008).

1.2 Motivation

Wireless mesh networks have proven to be cost effective. They have self management features, and they are more extendable than other types of wireless networks. Wireless mesh networks do not need extra network administrator interaction. Wireless mesh networks are used mostly in dense areas such as schools, airports, hospitals and hostels. The strategic plans (MoHE Afghanistan, 2010) of the Afghanistan ministry of higher education and ministry of education are to equip schools and university campuses with computers and Internet access to improve the education and research quality. Wireless mesh networks are a promising technology for these environments and most of the universities in Afghanistan are already connected using wireless networks.

Freifunk firmware is *open source* Linux-based firmware which can can be adapted as needed. For example, it can be localized by changing the English or

German interface of Freifunk to Dari to make it easier to understand in Afghanistan. The Freifunk project which started in Berlin is a good example of using wireless for mesh networks. The OLPC project in Afghanistan uses Freifunk . A team of software developers are working to localize this firmware for use in small laptops in schools in Afghanistan (OLPC, 2009). Later we will explain how Freifunk firmware can provide us with valuable insights into the practical application of WMNs.

It follows that security in wireless mesh networks is a big challenge.

1.3 Research questions

Wi-Fi mesh networks are a promising approach to networking in addition to non-mesh wireless networks, but user access management of mesh networks is more difficult. There are many mechanisms for user access management on wireless infrastructure mode networks, such as filtering mechanisms, cryptographic protocols, and authentication protocols to control the access of users. Each of them has advantages and disadvantages. In our research we characterize and compare the existing cryptographic protocols, basic user access management, and authentication mechanisms for infrastructure mode wireless to identify options for Wi-Fi mesh networks.

The goal of our research is to determine the best authentication mechanism to address the problem of Wi-Fi mesh network authentication using open source software and protocols. To deal with authentication protocols that address secure access of Wi-Fi mesh networks we need to answer the following questions.

1. How can we characterize the existing Wi-Fi authentication mechanisms?

   We did a literature review and feasibility study to answer this question. We compared 5 types of firmware for the Linksys Wrt54gl router to investigate their authentication capabilities and we characterized 14 types of authentication mechanisms to answer this question. However they do not work for Wi-Fi mesh networks. The reasons are detailed in Chapter 2.

2. What is the best option to pursue for Wi-Fi mesh network authentication?

   We designed and installed real mesh networks, tested several authentication protocols, and collected results to answer the second question. We also tried to implement two specific solutions. We describe the results and efforts in detail in Chapter 4, based on methods described in Chapter 3.

Both questions are important for this study because in order to understand the existing authentication methods it is important to analyze existing authentications. The authentication methods are summarized in Chapter 4 in Table 4.6. Furthermore, understanding the similarities, differences, capabilities and limitations of existing authentication mechanisms, is important for identifying the best one for wireless mesh networks.

1.4 Overall approach

Most of the research in ad-hoc and mesh networks have been carried out using simulation tools such as NS2 and a packet tracer (Andel and Yasinac, 2006). This has the disadvantage that most of the simulation tools are limited by the physical layer and do not fully support all the types of protocols which we needed in our research. The simulations do not always reveal which implementations in the network do not work in real networks. For example connecting wireless devices to a packet tracer is much easier in a real mesh network because most of the configuration is done automaticaly. We did our research using real infrastructure in a mesh network with five Wrt54gl routers and one Laptop computer.

As mentioned in Section 1.3, the aim of this thesis is to pursue an authentication mechanism that addresses secure connections in Wi-Fi mesh networks. We focus on analyzing existing authentication mechanisms, comparing the capabilities and limitations of authentication protocols and finding the most appropriate of those authentication protocols to improve the security of Wi-Fi mesh networks. Our methodology is discussed in depth in Chapter 3.

We used the wireless Linksys Wrt54gl router, because it has open source firmware that we can adapt. During our experimentation we tested 14 types of authentication mechanisms, used the freeRADIUS server as authentication server, we used VPN, and we used five types of firmware. We used the `openssl` library as a *certification authority* (CA). We employed five Linksys Wrt54gl routers, with several wireless clients for implementing and verifying these protocols. Finally, we propose Freifunk firmware, the OLSR routing protocol, `TincVPN` , and `openssl` for securing connections and user access management.

1.5 Thesis outline

This section outlines the entire thesis as a guide for the reader. The thesis is organized in five chapters. This introductory chapter gives a background of wireless mesh networks and authentication in wireless mesh networks in Section 1.1, introduces the motivation in Section 1.2, the research questions in Section 1.3 and the overall approach in Section 1.4.

Chapter 2 discuses related work focusing on Wi-Fi infrastructure and Wi-Fi mesh network authentication protocols. Section 2.1 covers 14 types of infrastructure Wi-Fi authentication protocols and Section 2.2 discusses the Wi-Fi mesh network authentication protocols and describes central authentication mechanisms and distributed authentication mechanisms for Wi-Fi mesh networks as proposed by several researchers such as Lee et al. (2008), Luo et al. (2004) and by Thompson et al. (2007).

Chapter 3 goes into the methodology and Section 3.4 describes the experimental designed of Wi-Fi and Wi-Fi mesh network. Section 3.5 discusses the firmware choices for Linksys Wrt54gl routers, Section 3.6 describe the protocol modification, Section 3.7 summarizes the contents of Chapter 3.

Chapter 4 covers the analysis of the thesis using several tables for comparison. Section 4.1 characterizes firmware and routing protocols in Tables 4.1–4.2. Section 4.2 discusses and tabulates authentication protocols in Tables 4.3–4.7 and Section 4.3 covers the protocols that are used with EAP in Table 4.6 and Table 4.7. TincVPN also detailed in this chapter.

Chapter 5 summarizes the entire thesis in Section 5.1. Section 5.1.4 points out the result of thesis. Section 5.1.5 discusses the conclusions the thesis. Section 5.2 points out the limitation of proposed authentication protocols and Section 5.3 discusses the future work.

Chapter 2

Related work

This chapter covers work related to authentication mechanisms for wireless mesh networks. The chapter contains a survey of authentication mechanisms for Wi-Fi networks primarily for infrastructure mode. Researchers have proposed different kinds of authentication protocols for Wi-Fi mesh networks. We discuss and analyze those proposed authentication protocols in this chapter. Section 2.1 discusses 14 types of Wi-Fi network authentication protocols. Section 2.2 discusses Wi-Fi *mesh* network protocols, centralized authentication protocols based on certificate authentication and distributed authentication. Section 2.3 summarizes this chapter on related work.

2.1 802.11 authentication mechanisms

Access to Wi-Fi networks should be controlled by a strong policy that defines their accessibility. Different methods are used for controlling user access in Wi-Fi networks and each of these methods has benefits and drawbacks. Authentication is a method used to control the access of users to the network. Access rights for those users who are allowed to use the network can be pre-assigned, and different authentication methods for Wi-Fi networks handle this in different ways.

Sections 2.1.1–2.1.14 discuss basic user access control, cryptographic protocols, and several authentication protocols. Each of these help in controlling access to Wi-Fi networks. Keeping in mind that different encryption and authentication capabilities depend on different firmware, we analyze several existing standards based on different firmware. Firewall based security, VPN and patching are also used for the security of Wi-Fi networks. Most network administrators use a combination of them in one Wi-Fi network.

There are various methods for access management of users in Wi-Fi networks. Sections 2.2.1–2.2.2 overview some well known methods of user access management for Wi-Fi networks.

2.1.1 Closed network

A closed network is a mechanism that gives access only to those who know the *service set identifier* (SSID) of the Wi-Fi network. Since the inception of closed networks, IEEE 802.11 has provided security mechanisms to reduce the potential security threat this extra freedom brings, e.g. IEEE 802.11 APs, or sets of APs, can be configured with a single SSID, which is known to the *network interface card* (NIC) in order to associate it with an AP and then continue with data transmission and reception on the network. Whether the association is allowed when the SSID is unknown can be controlled by the NIC/driver locally without using any encryption. This is a very weak security system because the SSID is known by all NICs and APs and the SSID is transmitted over the network in clear text.

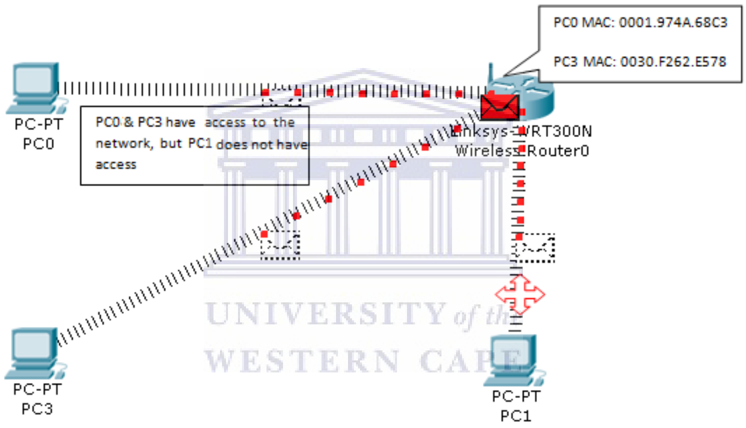
A closed network mechanism was introduced to decrease these threats. Normally routers and access points broadcast the *enhanced service set identifier* (ESSID) or SSID many times to the clients that are served by an AP/router. Clients can easily find this AP/router based on the SSID. In a closed network router/access points do not broadcast the SSID of the network to clients, and each client who wants to access the network should know the complete name or code of the SSID. Otherwise, the user cannot get access to the network. The good point of this method is to allow only those who know the SSID to use the network.

Closed networks also have drawbacks. Other network administrators do not know which channel is already in use in this environment. Therefore, a closed network can cause interference to other networks in the same environment. When a legal user types the SSID, malicious users can sniff the packet that is exchanged between the router/access point and the client because it is in clear text. Forgetting the SSID leads to administrative problem for users and administrator (Flickenger et al., 2006).

2.1.2 MAC address Filtering

The *media access control* (MAC) address is a 48-bit address which is hard coded in the NIC of each network device. This address can be used to control user access to the network. Router or access points can keep a list of client MAC addresses and when a client tries to access the network, the client MAC address is matched with the MAC address database as is illustrated in Figure 2.1. If the MAC address is found, access to the network is granted, otherwise, access is denied. However, it is not a very secure method because malicious users can spoof a MAC address.

This method works well for small networks and temporary solutions. For example, a computer that generates viruses can be included in the black list and its connections can be denied. However, for large networks, it is difficult to keep an updated list of MAC address that are allowed to use the network or are denied access to the network, because we can assign only a limited number of client MAC addresses in the filter lists (Flickenger et al., 2006). MAC address filtering can also work in point-to-point mesh networks because blockage of connections of any mesh router can be performed even if that router has other related clients.



PC0 MAC: 0001.974A.68C3

PC3 MAC: 0030.F262.E578

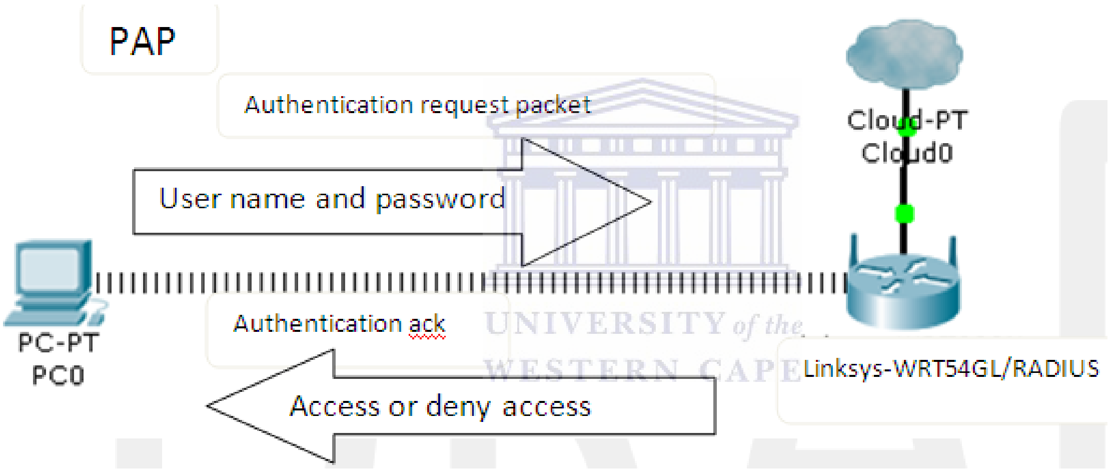
PC0 & PC3 have access to the network, but PC1 does not have access

Figure 2.1: MAC address filtering system

2.1.3 Password Authentication

The *password authentication protocol* (PAP) is a simple authentication protocol that is used for remote authentication servers. Almost all of the network operating systems support PAP. PAP transmits a username and password across the network unencrypted and it is a point-to-point protocol. Figure 2.2 illustrates the PAP authentication process. PAP is an insecure protocol compared with newer protocols, because it does not have an encryption system. In the PAP method, the user sends an access request packet containing a username and password together to the authentication server, then the authentication server validates the username and password or rejects the request (Kim and Choi, 2004).

When PAP is used with wireless networks, it should not be used alone, but rather with other authentication methods such as *tunnel transport layer protocol security* (TTLS) to ensure that the password is not revealed. In 2006, the Ma group (Ma et al., 2006) reworked PAP and developed another version of PAP called M-PAP. PAP and M-PAP were mostly used in Wi-Fi point-to-point authentication. M-PAP has integrated security and is better than PAP. However, M-PAP still has drawbacks. For example, it is vulnerable to off-line password guessing attacks (Yoon and Yoo, 2006). Even Yoon and Yoo's paper was soon shown to be flawed by Lee et al. who showed that Yoon and Yoo's protocol is vulnerable to both the off-line password guessing attack and to the replay attack (Lee et al., 2007).



3:

Figure 2.2: The PAP authentication process

2.1.4 CHAP: Challenge-handshake authentication protocol

The *challenge-handshake authentication protocol* (CHAP) was originally designed for wired networks. A version of CHAP is proprietary to Microsoft but has a documented and updated RFC as well (Lloyd and Simpson, 1996, RFC 1334). CHAP is a three way handshake authentication protocol that is used by the authentication server to identity the remote client that wants to access the network. CHAP performs the following steps:

1. After the *link control protocol* (LCP) phase is complete, and CHAP has been negotiated between both devices, the authenticator sends a challenge message

to the peer.

2. The peer responds with a value calculated using a one-way hash function.
3. The authenticator checks the response against its own calculation of the expected hash value. If the values match, the authentication is successful, otherwise, the connection is terminated.
4. The authenticator sends a new challenge to its peers randomly and then steps 1–3 are repeated.

The client is authenticated during the initial link establishment phase and the process is repeated until the source and destination are disconnected. Verification of point-to-point nodes is based on the device host name and password, and also uses a hash value. In the CHAP system, the user sends an access request together with a password to the authentication server, but without a username, while in the PAP method the user sends the request packet with a username and a password (Simpson, 1996, RFC 1994). CHAP has a variable challenge value, and uses repeated challenges, and it can prevent attacks because it has complete control over access and authentication. CHAP has a one-way authentication system that requires a plain text password because an encrypted password database cannot be used so it is not suitable for large networks.

Advantages of CHAP

CHAP changes the hashed-key identifier periodically. The use of repeated challenges can reduce the vulnerability of networks. The authenticator is in control of the frequency and timing of the challenges. This authentication method depends upon a "secret" known only to the authenticator and the specific peer. The secret is not sent over the link. Although the authentication is only one-way, by negotiating CHAP in both directions the same set of secret keys may easily be used for mutual authentication. Since CHAP may be used to authenticate many different systems, name fields may be used as an index to locate the proper secret in a large table of secrets. This also makes it possible to support more than one name or secret pair per system, and to change the secret in use at any time during the session (Lloyd and Simpson, 1996, RFC 1334).

Disadvantages of CHAP

CHAP requires that the secret be available in plain text form and an encrypted password database cannot be used. It is not useful for large installations, since every possible secret is maintained at both ends of the link. CHAP is not a mutual authentication protocol and is therefore not suitable for multi-hop networks (Lloyd and Simpson, 1996, RFC 1334).

2.1.5 Shared key authentication

Shared key and *open system* are the two oldest authentication mechanisms for IEEE 802.11 standard. “Open system authentication” does no authentication because each client that is in the wireless coverage area can access and can use the network. “Shared key authentication” does its authentication by using *wired equivalent privacy* (WEP) encryption. In this method, the router or access point broadcasts its SSID to the coverage environment using a 128-bit random number. All clients that are in the coverage area can detect the wireless network, but each client that has the shared key can access the network otherwise it will be denied. If the client sends the key to the router or access point, and the shared key is correct then client is allowed to access the network or else it will be denied. Figure 2.3 shows the *open system* authenticatiom mechanism. The process of *shared key* authentication is a four-way handshake, illustrated in Figure 2.4.

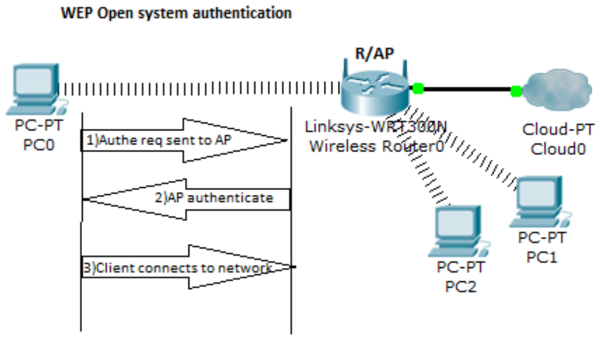


Figure 2.3: Open system authentication

Shared keys have some limitations as well. The attacker just needs the shared key, which can be found in different ways—maybe from a disloyal user or by intercepting the packet that passes between clients and nodes. No mutual authentication method is available where the user does not have clear information that he / she is connected to the right network or not. All of the users use the same shared key. When the shared key needs to be changed an advertisement will be made to all of the legitimate users. Shared key authentication has been deprecated since 2004 by the IEEE standard (IEEE Std 802.11i, 2004, Page 34).

Encryption is another mechanism for user access management, and provides better security for wireless networks, several cryptographic protocols already exist which we cover in Sections 2.1.6–2.1.8.

2.1.6 WEP: Wired equivalent privacy

Wired equivalent privacy (WEP) is the first encryption protocol which was used for wireless network security. It encrypts the data which is exchanged between router/access point and clients using a symmetric key algorithm. WEP has two types of key, a 64-bit key and a 128-bit key. 24 bits are occupied by the *initialization vector* (IV) which is transmitted as clear text, and the other 40 bits of the 64-bit key or 104 bits of the 128-bit key are occupied by the actual key used in WEP. After altering 24 bits to the IV in every new data frame it is passed together with the shared key to the *Rivest cipher 4* (RC4) algorithm to generate a pseudo random stream.

RC4 is used in the WEP encryption method, and the *cyclic redundancy check* (CRC-32) encodes and decodes the data for integrity purposes. The packet passed from from the NIC to the router has three major parts: (1) the initialization vector, (2) the actual data and an *integrity check value* (ICV), and (3) the actual data and the ICV encrypted by CRC-32.

WEP provides two types of authentication for user access management to the network: (1) *open system* authentication mechanisms and (2) *shared key* authentication mechanisms. In *open system* authentication the client does not need any key during authentication and after authentication. WEP is used for encrypting the frame and in this case the client requires a key. The *shared key* authentication that WEP uses is a four-way handshake for authentication, that is shown in Figure 2.4.

1. The client sends an authentication request to the AP/Router.

2. The router or access point returns a clear text challenge.

3. The client has to encrypt the challenge text using the WEP key and send it back in another authentication request.
4. The AP/router decrypts the packet and sends a positive or negative response to the client.

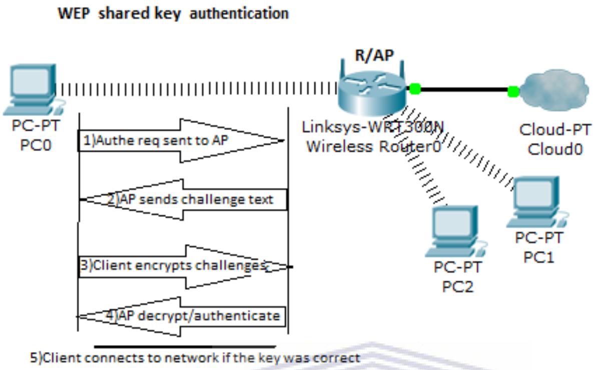

WEP shared key authentication
R/AP
PC-PT
PC0
Linksys-WRT300N
Wireless Router0
Cloud-PT
Cloud0
PC-PT
PC1
PC-PT
PC2
1)Authe req sent to AP
2)AP sends challenge text
3)Client encrypts challenges
4)AP decrypt/authenticate
5)Client connects to network if the key was correct

Figure 2.4: The WEP shared key authentication process

WEP has some weaknesses as well. 24 bits of IV is in clear text and it is not a very big number. After passing $2^{24} = 16777216$ packets, the use and reuse of the same key pose the main problem of WEP. An attacker can easily find the IV key by capturing 2^{24} packets. Weak keys for RC4 algorithm pose another problem for WEP. The cracker can easily crack this key using `AIRCRACK`, software that is freely available. It has several tools for breaking the WEP key, such as `AIRPLAY`. It has some tools that can capture network frames and can easily find the WEP key (Maple et al., 2006; Rigney et al., 2000, RFC 2865). WEP has been deprecated since 2004 by the IEEE standard (IEEE Std 802.11i, 2004, Page iv).

2.1.7 WPA: Wi-Fi protected access

Wi-Fi protected access (WPA) is an encryption and authentication method in Wi-Fi networks that was developed by the Wi-Fi Alliance to solve the security problems of WEP. WPA was developed by the 802.11i working group, and works with all the IEEE 802.11/a/b/g standards.

The *temporal key integrity protocol* (TKIP) is a key scheduling algorithm concept which is used in WPA. In the real world WPA is an enhanced type of

WEP. WPA not only provides strong data encryption to correct WEP weaknesses, but it adds user authentication which was largely missing in WEP as well (LaRosa, 2004).

The CRC-32 algorithm that was used for WEP has been replaced with *message integrity code* (MIC) for data integrity and header integrity on WPA. Based on WPA, a 128-bit dynamic key is used by the RC4 algorithm to finalize the encryption, and 64 bits are used for authentication. WPA can use both shared and private keys for authentication, it employs 802.1X authentication with one of the standard *extensible authentication protocol* (EAP) types as well. WPA can also use a third party database such as *remote authentication dial in user service* (RADIUS). While WPA is more secure then WEP, it also possible to crack it by using the cloud service WPA Cracker on the `www.wpacracker.com` website. It uses a brute-force dictionary-based attack which can check the captured network traffic using its cluster of 400 CPUs and 135 million word dictionary set up for WPA passwords in 20 minutes—at a cost of \$17. The same job is still feasible on a fast quad-core PC and should take only a few days. On our 1.4GHz notebook we can check 16–18 passwords/second. It is therefore recommended that pass phrases should be at least 20 characters long and not contain any dictionary words.

Both WPA and WPA2—see Section 2.1.8—are vulnerable to *denial of service* (DoS) attacks. WPA has a technique that if a host receives at least two wrong packets sent per second from a router or AP the WPA shuts down the network. Nevertheless malicious users can take advantage of this because the WPA shuts down the network for a minute and during this time all the links are disconnected.

WPA and WPA2 both have two modes, namely *enterprise* mode and *personal* mode. Enterprise mode uses a RADIUS server for authentication, and IEEE 802.1X/EAP to process the information. WPA enterprise mode is mostly used on bigger networks. WPA personal mode is used for home and small networks. It uses a combination of *pre-shared key* (PSK), TKIP and MIC (Maple et al., 2006).

WPA-PSK can be hacked by coWPAtty tools. Cracking of WPA-PSK is done by systematically testing numerous passwords and combinations of characters. It is estimated that on a Pentium 4 3.8GHz system, coWPAtty can try 70 words per second, however it would take over 3452 days to test all the possible eight letter passwords (over 208×10^9 combinations) if they are not in a dictionary. Therefore, WPA has stronger encryption than WEP (Acharya et al., 2009), but as we have pointed out before WPA is vulnerable to attack by cloud clusters.

2.1.8 WPA2: Wi-Fi protected access II

Wi-Fi protected access II (WPA2) was introduced in September 2004. It was developed by the 802.11i working group. WPA2 was created to replace the RC4 algorithm and its weaknesses by using a strong encryption algorithm. WPA2 uses the *advanced encryption standard* (AES) for data encryption, and the *Counter mode with cipher block chaining message authentication code protocol* (CCMP) for data integrity and header integrity, it is more secure than WEP and WPA. The AES algorithm needs only 128 bits for authentication and EAP is used as the protocol for key management and centralized mutual authentication (Maple et al., 2006).

The personal version of WPA is typically referred to as WPA-PSK or WPA2-PSK, which is a fancy term for a password. The Enterprise versions are commonly referred to as WPA-RADIUS and WPA2-RADIUS because they require a RADIUS server employing one of five different EAP standards, which are described in Sections 2.1.10–2.1.14.

Authentication is a process that takes place between a client and an AP/router that can be based on CA or based a on filtering mechanism (Huber and Jordan, 2005). It is a process that identifies authorized and unauthorized users. Each wireless user who wants to access a wireless network should be controlled by a network administrator. With a good access mechanism, isolation of the legal and illegal users and prevention of network attacks can be performed. Different mechanisms exist for authentication in a wireless network. Each of these has benefits and some weaknesses (Akyildiz and Wang, 2005). PAP, CHAP, IEEE 802.1X, EAP, EAP-TLS, and EAP-TTLS are the most well-known protocols that are used in wireless networks. The remainder of this section covers EAP and its extensions.

2.1.9 EAP: Extensible authentication protocol—IEEE 802.1X

EAP is a flexible protocol that works with other authentication protocols. It can support and can get help from other upper layer authentication protocols, such as *transport layer security* (TLS), and *tunnel transport layer security* (TTLS). Originally EAP was an 802.1X standard that allowed developers to pass security authentication data between the authentication server, authenticator and supplicant.

It was originally designed for dial-up *point-to-point* (PPP) connections, but is used with upper layer protocols such as *protocol for carrying authentication for network access* (PANA) and TTLS to work on wireless and wireless mesh networksr. EAP 802.1X resides in the access point or router and keeps the network port disconnected until authentication is completed. Depending on the results of the EAP message, either the port is made available to the user, or the user is denied access to the network. In these protocols, four types of messages are exchanged between the client and authenticator server. Figure 2.5 shows how EAP works over IEEE 802.1X in four main steps.

1. Request identity message from authenticator.
2. Reply message from the client.
3. Success message from authenticator when authentication was successful.
4. Failure message from authenticator when authentication was unsuccessful.

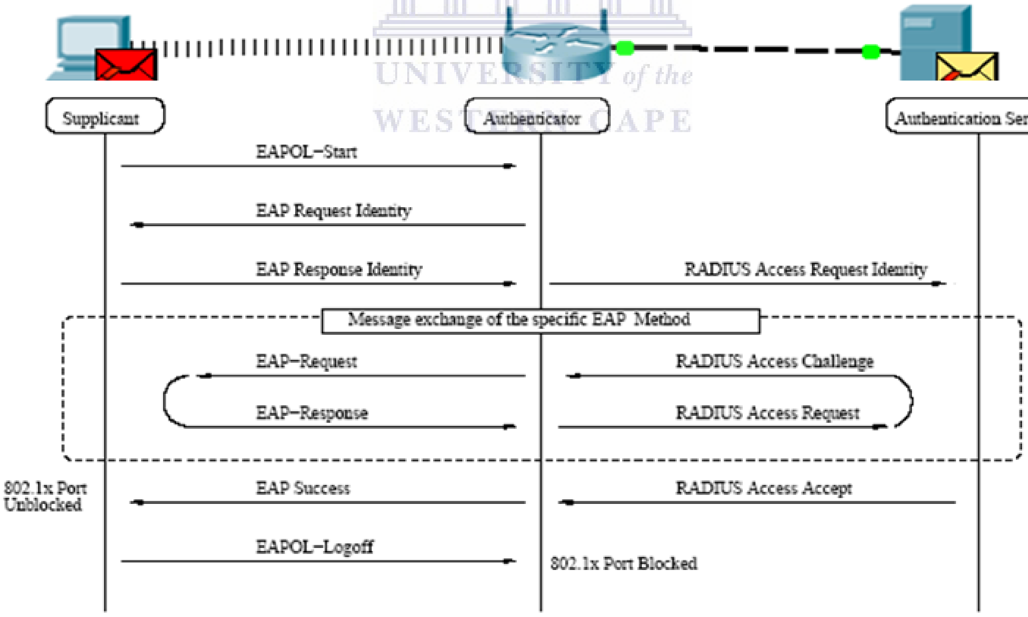


Figure 2.5: EAP over IEEE 802.1X with RADIUS

EAP has some benefits and some weaknesses. It is extensible and changeable and it can be modifed by adding other protocols, but EAP always needs another protocol to complete the authentication process. Each access point or router that uses EAP also needs to support IEEE 802.1X, because it is the requirement of EAP. EAP has a more complex authentication mechanism than other protocols (Blunk et al., 2004, RFC 3748). EAP has three main entities (Frank, 2006):

Supplicant: is usually a client who is trying to access the network.

Authenticator: is usually an AP or router, which forwards the authentication key / username and password or certificate to authentication server.

Authentication server: is normally a RADIUS / free radius server, which analyzes the received authenticator data.

IEEE 802.1X is an authentication method which was formed in the early 1990s. It is a port based authentication which carries the EAP message between supplicant, authenticator, and authentication server. It was originally designed for point-to-point networks. IEEE 802.1X is not suitable for authentication in a multi-hop wireless mesh network because it is a port-based OSI Layer 2 authentication protocol, and multi-hop networks need an OSI Layer 3 security protocol. It is not a mutual authentication method, but if we use it with WPA and EAP it can do mutual authentication as well (Khan and Akbar, 2006).

EAP is flexible with numerous authentication protocols, and new authentication protocol can be added to it. It can limit the possible attacks and it is independent of the network layer protocol like *Internet protocol* (IP) addresses, because it is a link layer protocol. It can support retransmission and if the communication is problematic, retransmission is possible and it can associate again. It can dynamically send keys to clients. Refreshing WEP/WPA keys at short intervals provides a good defense against many of the attacks. EAP-IEEE 802.1X can be used with several other authentication protocols such as RADIUS, TLS, TTLS, PANA and CHAP (Zhang et al., 2009).

2.1.10 EAP-RADIUS: Remote authentication dial in user service

Remote authentication dial in user service is also called RADIUS; it was designed to solve the authentication problem in a network environment. It was originally designed to be used in a wired network, but it is also being used on wireless networks. The big difference between wired and wireless networks is that the communication is processed at different layers of the OSI protocol stack.

There are two main authentication points:

1. Network authentication server (NAS)/(Router / AP).

2. Authentication server (AS)/(RADIUS server). RADIUS is mostly used with 802.1X, EAP-TLS, EAP-MD5, PAP and CHAP.

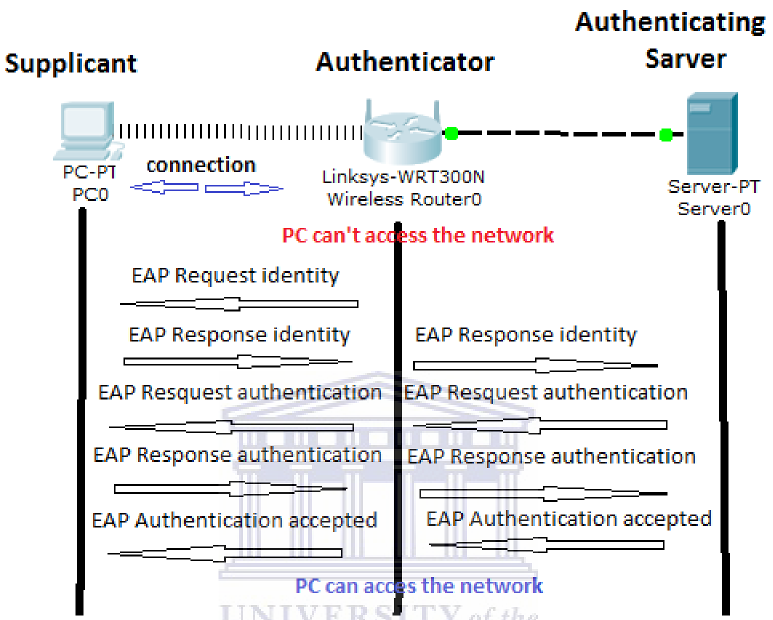


Figure 2.6: The full cycle of the RADIUS authentication process

As shown in Figure 2.6, there are four types of general message transferred by EAP:

1. The client sends an EAP start packet to the authenticator.

2. The authenticator replies with an EAP identity request.

3. The supplicant sends an EAP identity response to the authenticator. Then the authenticator transfers the received identity response to the authentication server. After receiving this packet the communication between the supplicant and authentication server starts.

4. The authentication server analyzes the identity packet and then, if the authentication was successful, the authentication server sends the RADIUS access accept message, otherwise it sends a RADIUS access reject message to the authenticator.

When the supplicant gets the RADIUS access accept message from the authentication server, the 802.1X port unblocks, and when the supplicant wants to end the session it will send an *extensible authentication protocol over LANs* (EAPOL) log off message to the authentication server, resulting in the blocking of the 802.1X port (Husseiki, 2006). The RADIUS protocol uses UDP for packet transfer through port number 1812. UDP is preferable because it is much faster than *transmission control protocol* (TCP) (Rigney et al., 2000, RFC 2865). The RADIUS server can also provide connectivity with smartcards. EAP messages are exchanged between clients and the authenticator server using a smartcard. In this method, a modem port or a *universal serial bus* (USB) port can be used for installing the smart card, but a USB port is better because many network devices have a USB port. Smartcards support IP addresses and are installed on a RADIUS server so that the authentication process can be controlled by the RADIUS server (Urien and Badra, 2006).

Das et al. (2004a) proposed a dynamic authentication system using smartcards. The advantage of this system is that the users can change the password freely. Nevertheless, it also has some drawbacks in that it cannot prevent guessing attacks completely, it cannot authenticate mutually and the password can be found by remote systems (Liao et al., 2006).

2.1.11 EAP-TLS: EAP-transport layer security

Extensible authentication protocol-transport layer security, defined in (Aboba and Simon, 1999, RFC 2716). EAP-TLS is a mutual authentication protocol which was developed by Microsoft in 1999 and is based on public key certificate authenticationr. EAP-TLS supports link layer fragmentation and reconnects rapidly. TLS uses an X.509 client/server certificate. This certificate is based on *public key infrastructure* (PKI), and it works on any hardware or software, such as Microsoft operating systems, Apple and Linux.

EAP-TLS was the first authentication method to meet three goals for wireless networks according to (Gast, 2005). Certificates provide strong authentication of both the users to the network, and the network to users. Mutual authentication provides a strong guard against rogue access points by enabling clients to determine that an AP has been configured by the right department, rather than an attacker who is intent only on stealing passwords. TLS also establishes a master secret key that can be used to derive keys for link layer security protocols (Gast, 2005).

However, EAP-TLS has some disadvantages; it is a point-to-point protocol that does only single-hop authentication (Aboba and Simon, 1999, RFC 2716). Another main drawback of using EAP-TLS is the overhead caused by the authentication procedurer. The encryption and decryption are time consuming, but using a certificate authority is a sound method for EAP-TLS authentication (Frank, 2006).

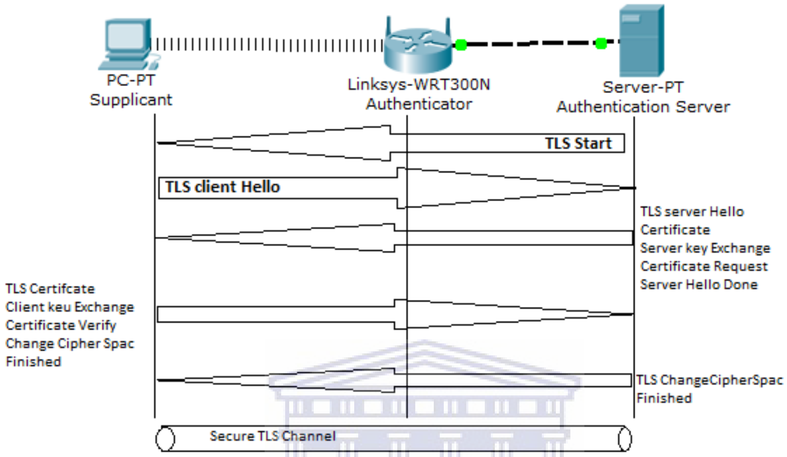


Figure 2.7: The EAP-TLS cycle

Figure 2.7 illustrates the EAP-TLS process. The EAP-TLS authentication process exchanges and analyzes packets in several steps (Frank, 2006):

1. The supplicant sends the client "hello" to the authentication server to initiate the session and exchange the identity request and reply.

2. The server replies to the request with a server "hello" message, which consists of a server certificate.

3. This certificate is checked by the supplicant. If the server-side authentication was successful, the supplicant sends the client certificate which is then checked similarly. Mutual authentication is accomplished if this last step is successful.

2.1.12 EAP-TTLS: EAP-tunnel transport layer security

The EAP-TTLS protocol was developed by (Funk and Blake-Wilson, 2002). It has an optional mutual authentication protocol, but commonly only one-sided authentication is used because mutual authentication can remove the sequence number of EAP-TTLS messages and can be the cause of overloading (Frank, 2006). It uses two authenticated layers that are external and internal. The external authentication uses a TLS handshake protocol for security, and internal authentication is for isolating users via EAP or password authentication protocols such as PAP or CHAP. EAP-TTLS is very flexible and it can be used together with other authentication protocols. EAP-TTLS is enabled by a third party server (Funk and Blake-Wilson, 2002). Figure 2.8 shows a typical network configuration using EAP-TTLS. Table 2.1 lists the acronyms used in EAP-TTLS authentication and not defined in the glossary.

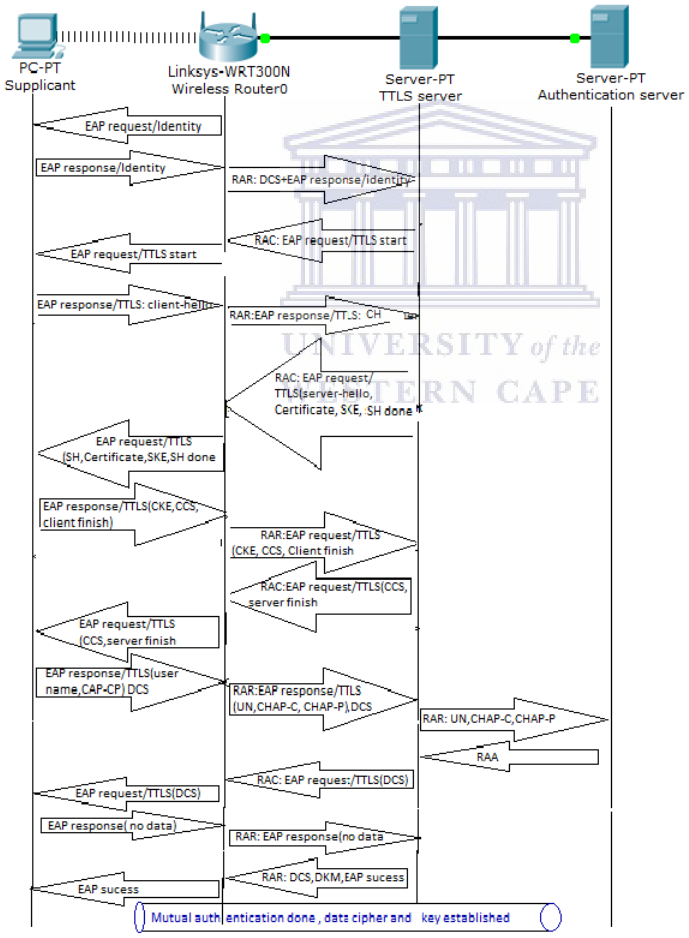

PC-PT
Supplicant
Linksys-WRT300N
Wireless Router0
Server-PT
TTLS server
Server-PT
Authentication server
EAP request/Identity
EAP response/Identity
RAR: DCS+EAP response/identity
RAC: EAP request/TTLS start
EAP request/TTLS start
EAP response/TTLS: client-hello
RAR:EAP response/TTLS: CH
RAC: EAP request/ TTLS(server-hello, Certificate, SKE, SH done
EAP request/TTLS (SH,Certificate,SKE,SH done
EAP response/TTLS(CKE,CCS, client finish)
RAR:EAP request/TTLS (CKE, CCS, Client finish
RAC:EAP request/TTLS(CCS, server finish
EAP request/TTLS (CCS,server finish
EAP response/TTLS(user name,CAP-CP) DCS
RAR:EAP response/TTLS (UN,CHAP-C, CHAP-P),DCS
RAR: UN,CHAP-C,CHAP-P
RAA
RAC: EAP reques:/TTLS(DCS)
EAP request/TTLS(DCS)
EAP response(no data)
RAR: EAP response(no data
RAR: DCS,DKM,EAP sucess
EAP sucess
Mutual authentication done , data cipher and key established

Figure 2.8: The EAP-TTLS cycle

Table 2.1: EAP-TTLS acronymns

Acronym	**Meaning**
RAR	Radius access request
RAC	Radius access challenge
CH	Client hello
SKE	Server key exchange
SH	Server hello
CKE	Client key exchange
CCS	Common channel signaling
CHAP-CP	CHAP-challenge and password
DCS	Data-cipher suite
UN	User name
CHAP-C	CHAP-client
CHAP-P	CHAP-password
RAA	Radius access accept
DKM	Data keying material

2.1.13 EAP-MD5: EAP-Message design 5

MD5 stands for Message-Digest 5 developed by Ronald Rivest in 1991 (Rivest, 1992, RFC 1321). MD5 was designed to work with the EAP protocol and it is an enhanced type of MD4. EAP-MD5 is specified in RFC 1994 (Simpson, 1996). and it was widely used in the 802.1X wired Ethernet switch but now it also used in wireless networks. EAP-MD5 is useful for public applications in which encryption is used at the application level. It uses a hash value for authentication. It has some security drawbacks, e.g. MD5 does not have any security key in the case of wireless networks. It cannot support mutual authentication, and conflicts of hash values can be another problem of this protocol. EAP-MD5 collects a username and a password from the user to be authenticated, encrypts that via the MD5 message hashing algorithm, and passes that data on to a RADIUS server (Blunk et al., 2004, RFC 3748).

2.1.14 LEAP: lightweight extensible authentication protocol

Cisco provides an apropiate wireless LAN security protocol called *lightweight extensible authentication protocol* (LEAP). It works with an authentication server like a RADIUS server. It is a password-based authentication that also can use a shared key mechanism. LEAP uses mutual authentication and can provide dynamic encryption keys. LEAP was a big step forward from the system of WEP manual keys to a dynamic encryption key system, but it uses MS-CHAP that is prone to dictionary-based attacks and it is proprietary to Cisco and can be used only with Cisco devices (Virendra et al., 2005) and (Gast, 2005).

2.2 Wi-Fi mesh network authentication

A wireless mesh network consists of mesh nodes and mesh clients, without any specific infrastructure. Mesh networks are a type of ad-hoc network which we introduced in Section 1.1. Mesh networks have a multi-hop architecture. User access control on multi-hop networks is more difficult than on single-hop wireless networks. Therefore user access management in wireless mesh networks is a challenge for the network administrator. Wireless mesh networks must therefore consider additional configurations for the authentication.

Access control in ad-hoc networks is a persistent challenge for several reasons. First, it is unlike wired networks or wireless cellular networks where access control can be deployed at a router or base station. Ad-hoc networks have a loosely structured architecture under continual change. Second, the access control of mesh networks has distribution problems. If all authentication is performed by a central device then it will tend to have bandwidth and overhead problems. Third, ad-hoc nodes moving from place to place need to have access to the network all the time and everywhere (Lee et al., 2008). Approaches include centralized authentication and distributed authentication. The centralized authentication mechanisms are detailed in Section 2.2.1 and distributed authentication mechanisms are described in Section 2.2.2.

2.2.1 Centralized authentication

Centralized authentication is a server and client mechanism where only the central device can issue certificates for the clients. If the central device is not available there is no way to renew or revoke other members. Because of this central authentication has a single point of failure, and is also not very scalable in wireless mesh networks.

A *certificate authority* (CA) is an entity which issues digital certificates such as public and private key pairs. The CA uses third party software for digital signature certification. In this method `openssl` mostly works as third party software. `openssl` is software that provides secure access to other networks as well. It encrypts the key which passes between the client and the server. The client and server share a secure ticket otherwise they cannot associate with each other. So the server needs to authenticate the key or ticket of the client trying gain access to the network with this pre-assigned key.

IEEE 802.1X and X.509 certificates

The IEEE 802.1X/EAP scheme uses a centralized server playing the role of an authenticator. However, considering the architecture of a wireless mesh, which is based on ad-hoc links between nodes, the need for multi-hop authentication has not been addressed. Some nodes have no single-hop access to the server. Since the mobility of the nodes within the WMN requires regular establishment of links with new neighbors, there is a definite need to provide multi-hop (re)authentication mechanisms in order to allow nodes to authenticate with the central server through a path of multiple authenticated nodes. Furthermore, the multi-hop authentication process between a client and the server, which is usually a combination of an EAP method and an AAA carrier protocol, should not reveal crucial information, e.g. the PSK, to the other MPs on the multi-hop path. This imposes some alterations, adaptations and restrictions on the EAP method and the AAA protocol (Cheikhrouhou et al., 2006).

Tung et al. researched CA using 802.1X and X.509 (Tung et al., 2006). They set up a CA server under RADIUS where the user can apply for an X.509 public key certificate through a webpage interface that is installed in RADIUS. RADIUS receives the application for the certificate and passes it to a CA manager. If the CA Manager approves the certificate application, it produces the user public key certificate and passes it back to the RADIUS database for users. After the user completes the certificate application, the certificate can be looked up in the RADIUS database and then obtain its own public key certificate and the public key certificate of the CA. In the case of more than one CA server, users obtain all CA certificates, after these CA servers mutually authenticate each other (Tung et al., 2006).

Authentication in a layered security approach

Komninos et al. (2007) explored integrated cryptographic mechanisms in the first and second phases to design multiple lines of authentication defense and further protect ad-hoc networks against malicious attacks. Based on the project of Komninos et al., symmetric key, asymmetric key, and elliptic curve cryptosystems were implemented to offer a complete analysis of the authentication protocols. AES and MD5 were implemented as symmetric key algorithms, and the *Rivest Shamir Adelman* (RSA), and Menezes Vanstone cryptosystems were used as asymmetric key algorithms. The key size was based on the X9.30 standard specifications (Komninos et al., 2007). However, this strong encryption mechanism can reduce the performance of networks, especially in a large wireless mesh network, but dynamic authentication and encryption is much better than static authentication.

EAP-PANA: Protocol for carrying authentication for network access

EAP-Protocol for carrying authentication for network access (EAP-PANA) has been under development by the *Internet Engineering Task Force* (IETF) since 2001 (Forsberg et al., 2008, RFC 5191). It is a type of authentication that assists clients to access the network. It works on multi-hop WMNs. PANA uses a similar authentication scheme as 802.1X, which works on the IP layer for multi-access and point-to-point links. PANA carries the authentication between the client and the server. PANA messaging involves several processes. Figure 2.9 shows the PANA framework authentication process. When clients are connected to the network they can get their IP address named the "pre-PANA address" through a DHCP server and can then get accredited by a PANA server called a *PANA authentication agent* (PAA). The PAA forwards request messages to the *authentication server* (AS) using an *enforcement point* (EP) for verification. The AS has a database of authorized and unauthorized clients, which checks the accreditation of clients in the database and if the accreditation is correct, the client can access the network or else will be denied (Khan and Akbar, 2006).

PANA is a framework that passes authentication messages around the network. The sequence of messages exchanged during a successful authentication process has several parts. The mesh access router sends an EAP-request/identity message encapsulated in the PANA-authorization-request message to the *PANA client* (PaC). This message initiates the process of authentication and then the authentication proceeds as follows:

1. Upon receiving the EAP-request/identity message, the PaC returns an identity, e.g. username, hostname, etc., in an EAP-response/identity message encapsulated into a PANA-authorization-answer message.

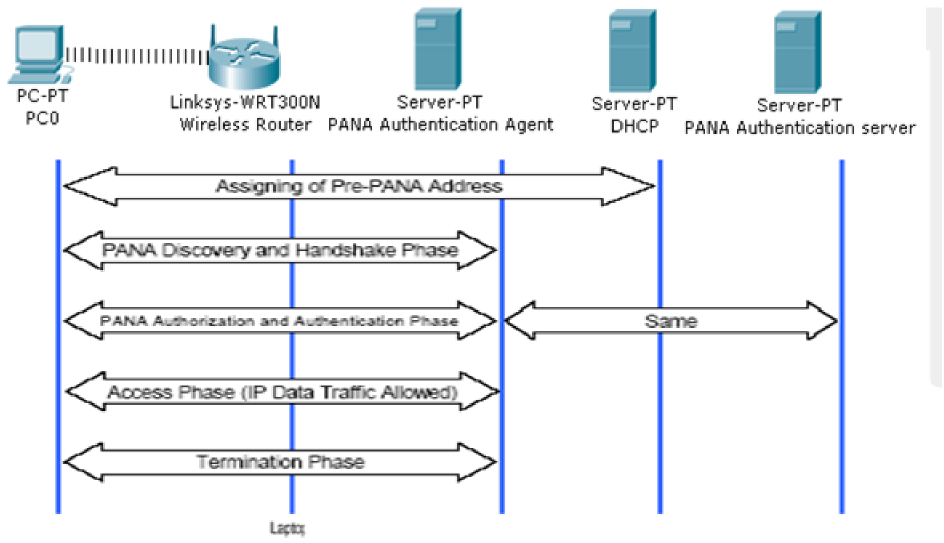

PC-PT
PC0
Linksys-WRT300N
Wireless Router
Server-PT
PANA Authentication Agent
Server-PT
DHCP
Server-PT
PANA Authentication server
Assigning of Pre-PANA Address
PANA Discovery and Handshake Phase
PANA Authorization and Authentication Phase
Same
Access Phase (IP Data Traffic Allowed)
Termination Phase
Laptq

Figure 2.9: The EAP-PANA-TLS cycle

2. Once having received the PaC's identity, the mesh router forwards this message to the AS. From this point, the mesh router acts as a pass-through between the PaC and the AS.

3. The AS then sends an EAP-TLS/start packet to start the EAP-TLS conversation with the PaC.

4. The PaC responds by sending a TLS client "hello" handshake message which contains the TLS version number, a TLS session ID, a random number, and a set of supported cipher encryption algorithms.

5. The AS then sends an EAP-Request packet containing a TLS server "hello" handshake message followed by a TLS certificate, server key exchange, and certificate request and server "hello done". The server "hello" handshake message contains the AS's TLS version number, another random number, a session ID, and the selected cipher encryption algorithms.

6. The PaC sends an EAP-response packet containing a client certificate, a client key exchange (which determines the session key—Master Session Key—with the server key exchange) and, verify certificate, which is a digital signature of the authentication response.

7. Upon receiving this EAP-Response packet, the AS proceeds by verifying the PaC's certificate and the digital signature. If the test succeeds, it sends

an EAP-Request packet containing the TLS change cipher specification and finished handshake messages which include a keyed hash over the message. By verifying the keyed hash, the PaC can authenticate the AS (EAP server). If the authentication is successful, the PaC and AS exchange EAP-response and EAP-success messages.

EAP-TTLS over PANA was proposed by Khan and Akbar. EAP-TTLS is an extension of TLS and it can be used with PAP, CHAP and MD5. Authentication protocols are mostly applied in the link layer and the network layer. Therefore, the operations of either the link or the network layer can enable one of the two phases to take place. In phase one, for example, the node authentication procedure attempts to determine the true identity of the communicating nodes through challenge-response protocols based on *symmetric key* techniques. Likewise, in phase two the authentication procedure again seeks the identities of the communicating nodes through challenge-response protocols based on *public key* techniques. In the first phase, the node identification procedure assumes that the secret is known to the verifying node, and this secret is used to verify the response with symmetric techniques. In the second phase of the authentication, the secret is not actually known to the verifying node. Asymmetric techniques can be applied before private information is exchanged between communicating nodes (Khan and Akbar, 2006).

OpenVPN

OpenVPN is a free, open source VPN built on SSL encryption. OpenVPN can support a wide range of operating systems such as Linux, Windows 2000/XP and later, OpenBSD, FreeBSD, NetBSD, Mac OS X, and Solaris. OpenVPN encapsulates all traffic in an encrypted tunnel, OpenVPN has higher latency than distributed authentication or other authentication mechanisms. OpenVPN can use public and private keys and also shared secret keys. The following benefits of OpenVPN that are of interest to us have led us to test it in our research:

1. It uses SSL and RSA which are strong encryption systems.
2. Configuration is easy.
3. A wide range of hardware and software support OpenVPN .

4. Freifunk has a well documented open source version of OpenVPN .

5. It is completely free and has changeable source code making it amenable to localization.

6. We can use OpenVPN in TCP or assign it to a UDP port and it works on the network and data-link layer.

OpenVPN is a client-server based mechanism that has a single point of failure, more load in a single point, and if we apply it to wireless mesh networks we have to generate key pairs for each client. Therefore it is not suitable for wireless mesh network authentication, especially in the backbone. The interface for creating OpenVPN certificates and keys is shown in Chapter 3, Figures 3.11 and 3.12. OpenVPN operates in the link layer and the network layer of the OSI model. It uses TLS for authentication and it can transport datagrams and OSI Layer 2 frames. OpenVPN is a most secure VPN as it uses SSL/TLS. It uses a sequence number and encrypts the datagram of the frame in the channel (Snader, 2007).

2.2.2 Distributed authentication

Distributed authentication is necessary in a wireless mesh network mechanism in which each mesh node can function like a central device, but using a shared key remains a risk, because a shared key can have problems which we already pointed out in Section 2.1.5. Mesh networks can grow and fragment. Therefore centralized authentication can cause extra load and overhead to a central authentication server.

Distributed authentication and key management

Husseiki's (2006) thesis proposed a certificate authority system as a general approach for distribution authentication of WMNs. His thesis proposed an hierarchical schema for wireless mesh networks to have an authentication server, supervisor and members of the WMN, that seems a very useful approach, for wireless mesh network fragmentation, but unfortunately it is not applicable on wireless mesh networks.

The changeability of structure of a WMN requires the possibility of obtaining certificates that are valid in multiple fragmented sections of the network. Therefore, certificates that are issued locally between mesh points in the WMN should also be valid when the a device holding such a certificate moves to the neighbor of another mesh points broadcasting the SSID of the same wireless mesh network. Even if it works, the hierarchical CA mechanisms do not solve the problem of single point of failure. Any compromised CA can cause the failure of the entire security system (Husseiki, 2006).

Wireless distributed system

The distributed authentication algorithm allows small enterprises to use a shared-secret-key mechanism and it allows a multi-hop environment to grow beyond a single AP in a *wireless distributed system* (WDS). This method simplifies the installation and deployment of the WDS. If an organization were to use the existing *independent basic services set* (IBSS) authentication for the wireless mesh network, then the administrator would have to provide a key for every mesh point—which is not easy. In a dynamic organization, this management burden would be intolerable. The only solution, currently, is to install an AAA server and perform centralized authentication. Yet, this management burden may be too heavy for the central server.

Distributed authentication scheme for wireless LAN based

Lee et al. proposed a distributed authentication mechanism for wireles network with a secure shared key which can significantly reduce the load and overhead of a central authentication server and speed up the mesh network (Lee et al., 2008). Lee et al. proposed a network where the administrator only needs to establish one PSK between two mesh points within the WMN and the clients from each WDS can roam freely between the two systems. This is a distributed authentication mechanism where a small enterprise network purchases a single AP to support one or more stations and each access point can work as an authentication server. This works without the need for an AAA server, but it cannot grow easily beyond the single AP scenario without tedious manual key management. This algorithm is designed to allow the enterprise to add APs easily and still provide the same degree of security as the single AP case without any additional work for the administrator.

The algorithm used in this distributed authentication system supports two isolated WDSs joined to form a single WMN. The system administrator need only establish a single shared secret between the two connecting APs, then clients from each WDS will be able to roam freely between the two systems. This method is a combination of a modified Otway-Reese protocol with broadcasting for a novel distributed authentication algorithm within dynamic topologies that easily integrates into EAP and the IEEE 802.11i protocols. This protocol uses a reactive routing protocol which is a mandatory routing protocol of the IEEE 802.11s standard which defines WLAN-based mesh networks. Lee et al. used the NS-2 simulator for testing their approach. They used AODV as a routing protocol. They also tested the delay and found a lower authentication delay compared to the existing scheme with a centralized architecture. Generally, the objective of increasing the number of nodes with the authentication functionality is to distribute the load over the nodes, therefore, to enhance the performance of the authentication operation and the performance of the network accordingly.

Lee et al. have a very good approach for wireless mesh networks, but this is only for wireless LAN and it does not work in the backbone of a wireless mesh network since they used IEEE 802.11i standards that are limited to OSI Layer 2, and the security weakness pointed out in Section 2.1.7.

URSA: ubiquitous and robust access control for mobile ad hoc networks

Luo et al. in 2004 proposed *ubiquitous and robust access control for mobile ad hoc networks* (URSA), that is a localized and ubiquitous authentication mechanism. URSA which is a user access mechanism based on the network layer, is fully localized to provide ubiquitous and robust access control for mobile ad-hoc networks. The proposed solution takes a ticket-based approach. Each well-behaving node uses a certified ticket to participate in routing and packet forwarding. Nodes without a ticket cannot access the network and will be denied. If a node moves from one mesh point's coverage to another coverage it needs to be certified by the new cover or parent. The tickets issued by a mesh point are valid for a fixed period after which they expire. The expiring ticket of a well-behaving node will be renewed collectively by its local monitoring neighbors, while a misbehaving node will be denied ticket renewal or its ticket will be revoked.

A mobile ad-hoc network which is protected by URSA cannot issue tickets to new users. First time users need to purchase a ticket from the central authentication office. After the ticket is granted, the mobile ad-hoc network can renew that ticket, before it expires.

URSA implements ticket certification services through multiple-node consensus and fully localized instantiation. It uses tickets to identify and grant network access to well-behaving nodes. In URSA, no single node monopolizes the access decision or is completely trusted. Instead, multiple nodes jointly monitor a local node and certify/revoke its ticket. Furthermore, URSA ticket certification services are fully localized into each node's neighborhood to ensure service ubiquity and resilience. The NS-2 simulator was used in the Luo et al. (2004) project. URSA is an efficient mechanism for authentication between two router-router networks.

URSA seems a promising mechanism for Wi-Fi mesh networks, but still the networks are related to a central server.

AGE: Authentication on the edge

Authentication on the edge (AGE) is a localized method for Wi-Fi and global open Wi-fi network access. AGE uses EAP-TLS as authentication. In the AGE project (Thompson et al., 2007) proposed distributed authentication on the edge, and implemented it in the Internet as a network between SPs. Thompson et al. compared EAP-TLS with EAP-AGE and found the latter to be much better. In EAP-AGE each node can operate without central server intervention. AGE uses a certification mechanism in which each device that wants to access the network should have a certificate. AGE localizes the authentication on each AP so that once a client gets a certificate then the client certificate does not need to be authenticated in the central server. The local AP can authenticate the client CA.

This authentication mechanism requires little user interaction since it does not work like a user-name-and-password mechanism. It is an automatic mechanism. The central server gives a certificate an expiration time and the client just needs to update the certificate. AGE is a localized and distributed authentication method proposed by (Thompson et al., 2007) for the *global-scale Internet access infrastructure* (GIANT) network to avoid Internet authentication. Three main ideas are used to address edge authentication

1. The use of certificate-based authentication,
2. the distribution of certificate revocation list segments to all entities, and
3. the self organization of access points into a social look-up network.

Authentication on the edge combines centralized administration and operator assistance with distributed algorithms to confine the authentication to the edge of the network. These methods achieve the scalability needed for the overwhelming size and volume of a global network and increase resiliency against intrusion. (Thompson et al., 2007). This method, like URSA, works based on CA, but it uses TLS and *certificate revocation lists* (CRLs) instead of private keys. A trusted third party is responsible for managing user credentials and handling billing. Table 2.2 lists the acronyms used in EAP-AGE authentication and not defined in the glossary. AGE localizes and completely decentralizes the authentication process itself while relying on a central server to manage, maintain, administer and disseminate updates of authentication material as a task separate from authentication itself. It is a mechanism that lies between fully centralized and fully distributed.

AGE's mechanisms make it well suited to GIANT networks. AGE supports a single authentication authority allowing clients to access the service anywhere with the same user ID and authentication credentials. Authentication in AGE proceeds with as little user interaction as possible—the user only has to select the GIANT SSID for association—and AGE is resilient to the variable network conditions in GIANT including potential loss of connectivity to the central server.

AGE is similar to EAP-TLS authentication. As we have mentioned in Section 2.1.11 EAP-TLS is a certificate-based method which uses public and private keys for certification. A central server operates the AGE CA which manages the certificates for all GIANT users. The central server pushes updates to all relevant parties when authentication material changes. To continue operation in the face of server failure and avoid delays caused by accessing an authentication server in the Internet, each AGE AP runs a self contained authenticator, confirming the authen-

Table 2.2: EAP/AGE acronymns

Acronym	**Meaning**
Assoc	Associate
Req	Request
OW	Open Wi-Fi
Auth	Authentication
CRL	Certificate revocation list
AGE	Authentication on the edge
AA	Access accept
AR	Access request
AC	Access challenge

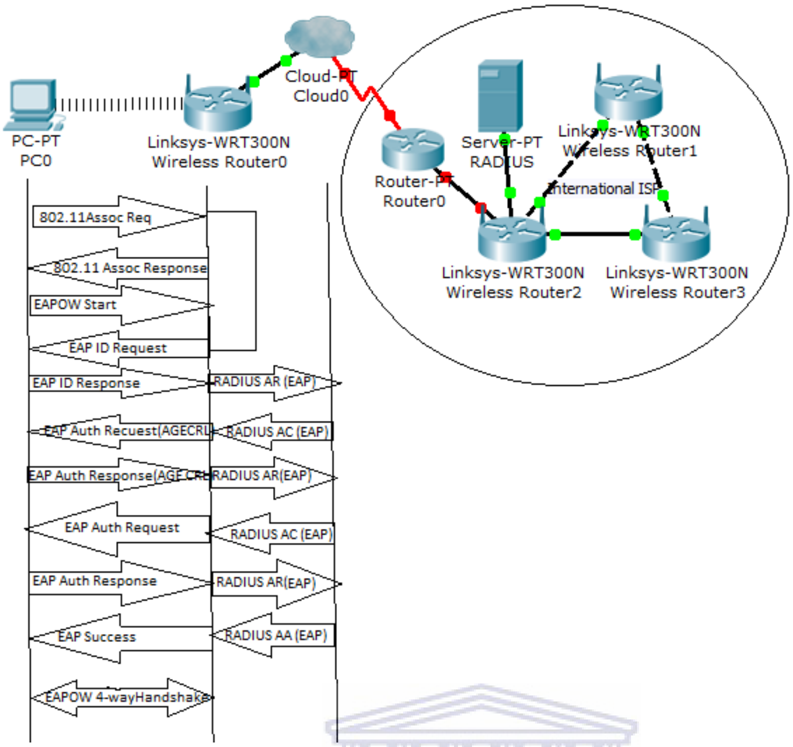


Figure 2.10: AGE authentication process

tication process to the wireless link. Only the CA root certificate is embedded at every entity allowing clients and access points to verify each other's certificates locally. AGE uses CRLs to inform AGE entities when a certificate has been revoked before its expiration. The CRL is also maintained by the AGE central server.

AGE was implemented as a new EAP module for the FreeRADIUS server and wpa_supplicant Linux software packages. It is also uploaded to the OpenWrt open source router firmware for the LinksysWRT home wireless router.

Measurement results comparing EAP-AGE to EAP-TLS in the GIANT scenario show that AGE satisfies requests with between 49.7% and 71.6% lower delay, around 490 msec and 1614 msec, providing a faster and more predictable authentication (Thompson et al., 2007).

Thompson et al.'s project is a useful project that localized the authentication and uses a local authentication server instead of an Internet authentication server, but can also use a central server.

TincVPN

TincVPN is a *virtual private network* (VPN) daemon that uses tunneling and encryption to create a secure private network (Tinc, 1998). `TincVPN` supports an encrypted tunnel between two hosts, and it can have the VPN between edges, too. `Tinc` does auto routing and manages the routing itself. Packet exchange in `TincVPN` is similar to the *open shortest path first* (OSPF) routing protocol. This information is continually updated as nodes join or leave the network or become unreachable. `Tinc` uses two channels for each VPN, i.e. UDP and TCP. `Tinc` operates at both the Link layer and IP layer. Conceptually in `Tinc` each node behaves as a VPN gateway. In addition, `Tinc` features encryption, authentication and compression where traffic is optionally compressed using `zlib` or `LZO` and the `openssl` library but it does not use the SSL protocol itself. `Tinc` protects the message from tampering and alterations with message authentication codes and sequence numbers (Snader, 2007, Pages 1–7).

`TincVPN` supports automatic full mesh routing. Regardless of how `Tinc` daemons have been connected to each other, since it supports multi-hop authentication, VPN traffic goes directly to the destination without going through intermediate hops.

We can easily expand a VPN through `Tinc` to several wireless mesh networks. In order to add nodes to the VPN, all we have to do is add an extra configuration file. There is no need to start new daemons or create and configure new devices or network interfaces. It automatically creates a virtual interface and we can apply the security to that virtual interface. `TincVPN` has the ability to bridge ethernet segments. We can link multiple ethernet segments together to function like one segment so it is very useful for segmentation and fragmentation of wireless mesh networks.

`TincVPN` supports many operating systems, and also supports IPv6. Currently Linux, FreeBSD, OpenBSD, NetBSD, MacOS/X, Solaris, Windows 2000, XP, Vista and Windows 7 platforms are supported. `Tinc` also has full support for IPv6, providing both the possibility of tunneling IPv6 traffic over its tunnels and of creating tunnels over existing IPv6 networks.

Although `Tinc` uses the `openssl` library, it does not use the SSL protocol to establish connections between daemons. The reason for is that when `Tinc` was created, SSL was starting to gain popularity for use outside web browsers and servers. SSL at that time did not make it easy to have both sides of the connection to authenticate each other. SSL requires a TCP-like transport layer to function, whereas a VPN works much more efficiently if it can send encapsulated packets via a UDP-like transport layer(Tinc, 1998).

When Tinc encrypts UDP packets, it uses the *cipher block chaining* (CBC) block cipher mode with a 32-bit counter. This was chosen to avoid the overhead of a full random IV for every packet. However, due to the predictable IV, an attacker could launch a chosen-plaintext attack (Katz and Lindell, 2007). Tinc distinguishers known plaintexts from each other. The main problem of TincVPN is the restricted memory of devices. TincVPN has a large overhead on the router for authentication and every router has to function as both server and client.

By default, Tinc uses *hashed message authentication code* (HMAC) to authenticate packets that are trunctated to 32 bits. This default was chosed to avoid the overhead of a full 160-bit hash for every packet. An attacker on a high-speed network connection could inject a forged packet by sending it 2^{31} times on average with different HMACs. It is possible to change the strength of the HMAC with the MAC length option. The default length will change in the future.

Tinc uses RSA without padding. Padding schemes are designed to prevent attacks when the size of the plaintext is not equal to the size of the RSA key. However, Tinc always encrypts random numbers that have the same size as the RSA key. This should be safe. There are *timing attacks* against RSA—Tinc does not protect against those. Tinc uses RSA encryption to send symmetric cipher keys to its peer. Then, a challenge/response exchange is done to verify that each peer indeed has the private key. However, MiTM attack is possible where an attacker that has the public key of one of the peers gains control over one side of the communication between two peers. The MiTM cannot decrypt messages between peers, but it can send messages to the peer that initiated the connection. If the MiTM knows enough about the VPN, it could trick peers into sending it packets that it can decrypt. However, the MiTM cannot send packets to other peers (Snader, 2007).

2.3 Summary

This chapter covered the IEEE 802.11 authentication protocols that are used in Wi-Fi infrastructure networks and Wi-Fi mesh networks. Section 2.1 covered 14 types of authentication protocols that are used in Wi-Fi infrastructure networks, they are support single-hop Wi-Fi networks. However, the security of Wi-Fi mesh networkis is difficult and problematic for several reasons. Wi-Fi mesh networks have unstable architecture and are under continual change. Access control of mesh networks has distribution problems and central devices have a performance overhead. Wi-Fi mesh points move around from place to place and need to have access to the network all the time and everywhere (Lee et al., 2008).

Section 2.2 covered the Wi-Fi mesh network authentication that includes central authentication mechanisms such as IEEE-802.1X (Cheikhrouhou et al., 2006), PANA (Khan and Akbar, 2006) and OpenVPN (Snader, 2007). It also covered distributed authenticator URSA (Luo et al., 2004), EAP-AGE (Thompson et al., 2007), distributed authentication and key management with TincVPN .

Chapter 3 discusses the methodology and Experimental network.

Chapter 3

Methodology

This chapter first discusses the challenges of existing Wi-Fi mesh networks in Section 3.1. The research questions are defined in Section 3.2. Section 3.3 deals with the overall approach and implementation of the research. We cover the experimental design for analyzing authentication mechanisms of Wi-fi and Wi-Fi mesh networks. Section 3.4 covers the experimental design. Section 3.5 discusses the choices for mesh firmware and Section 3.6 describes protocol modification. Finally, Section 3.7 summarizes the contents of this chapter. The results are presented in Chapter 4.

3.1 Challenges for Wi-Fi mesh network authentication

Single-hop wireless networks are difficult to extend but are more secure and more easily controled than wireless mesh networks. Mesh networks are more easily extended and are more scalable than wired and single-hop wireless networks, but they are more vulnerable to different types of attack. Less security is one of the important drawbacks of wireless mesh networks. Several mechanisms are defined for the security of wireless mesh networks such as various type of authentication, encryption and filtering systems. However, each of the current methods entails some limitations with respect to fragmentation and dynamicaly distributed authentication. Several user access management methods were discussed in Chapters 1 and 2. We compare and summarize these protocols in Chapter 4. The access policies of most wireless mesh networks are vulnerable to attack and most of them only work on single-hop wireless networks.

Extension of a mesh network can complicate security administration and reduce the performance of the network, because of added computation required for authentication services causing device latencies. Most of the authentication methods were designed for a single-hop wireless environment such as IEEE 802.1X, but mesh networks are multi-hop networks, requiring protocols that can support multi-hop networks and multi-hop network authentication. Encryption of protocols such as WEP encryption is done only on the authenticator side and passwords from the client's side are in clear text, which an attacker can capture and use to gain access to the network. Some of the other protocols such as EAP-TLS or EAP-TTLS have mutual authentication. The encryption and decryption of the packets causes overhead and reduces the performance of the network. These protocols only work on the OSI link layer. Thus, most of the protocols such as *medium access control* (MAC) filtering or IEEE 802.1X, PAP and CHAP work on the link layer, and they are vulnerable to different attacks and cannot support a backbone wireless mesh network.

A protocol such as EAP-RADIUS was originaly designed for wired networks but is currently also used in wireless networks. RADIUS is based on IP addresses but the security of this method is guaranteed over a one-hop wireless connection with a central RADIUS server. Multi-hop wireless mesh networks are still vulnerable to unauthorized access (Frank, 2006; Aboba and Calhoun, 2003, RFC 3579). RADIUS also works on the link layer and cannot support a backbone wireless mesh network.

WMNs can be compromised more easily than wired networks due to several factors: (1) their distributed network architecture, (2) the vulnerability of channels and nodes in the shared wireless medium, and (3) dynamic changes of network topology. Attacks on routing protocols and MAC protocols is possible on WMNs. Wireless links are vulnerable to attacks that other wireless media are also prone to, so suitable cryptographic protection has to be setup for WMNs. A dynamic and distributed authentication method to support multi-hop or mesh wireless networks, which can circumvent single failures and reduce the load of the central authentication server, is required.

A centralized authentication scheme is not suitable for WMNs where the network topologies are dynamic and distributed, due to mobility and network failure that arises from their ad-hoc nature. Moreover, key management in WMNs is much more difficult than in infrastructure wireless networks, because it's more complicated for a central authority to handle distributed networks. The dynamic characteristic of WMNs also makes key management more complicated. Key management in WMNs needs to be performed in a distributed but secure manner.

Therefore, a distributed authentication and authorization scheme with secure key management is needed for WMNs. Distributed authentication with a public key infrastructure is straightforward for the implementers. It is, however, a major management and operational hurdle for end users (Frank, 2006).

The IEEE 802.11i standard defines the security architectures for protecting the link layer between the client and the AP. It provides the security architecture such as authentication, confidentiality, key management, data origin authenticity and replay protection. The authentication framework of this standard is for both infrastructure and ad-hoc modes. This authentication framework uses a combination of several protocols such as IEEE 802.1X and TLS. Authentication is performed through the interaction of three entities—client, AP, and authentication server. Authentication is performed to give access to the network only for legitimate nodes. For infrastructure WLANs, this is performed through a centralized server such as RADIUS.

WPA, which was developed by the 802.11i group, is able to encrypt the data transferred between mesh nodes, it also provides authentication simultaneously. WPA works together with a *network authentication server* (NAS), and other protocols such as EAP-802.1X, and RADIUS (Frank, 2006), but it does not function on the backbone wireless mesh network which works on OSI Layer 3.

Based on our analysis of mesh network authentication, dynamic authentication and combinations of several existing authentication protocols in wireless mesh networks are very important for preventing attacks and securing the network. A hybrid wireless mesh network has two parts (1) a mobile ad-hoc network or clients and (2) an infrastructure or backbone wireless mesh network (Akyildiz and Wang, 2005). Existing authentication protocols cannot secure the connections between routers.

Wireless networks are vulnerable to various types of attack and the following are the most common attacks on wireless mesh networks:

DoS is an attack where the attacker hampers the normal functioning of a server by flooding it with repeated requests for services that it cannot cope with because of the volume, making it virtually impossible for the server behave normally. Recently the website "`wikileaks.com`" has been forced to distribute its website all over the internet to many sites such as "`wikileaks.za.org`" to counter the DoS attacks. On 11 January 2011 the latter site was functional but because of DoS attacks by persons unknown the `wikileaks.com` site advertized an apology for not being available.

Unfortunately, this type of attack cannot be completely avoided as the attacker can do this easily when the IP address of the router is known. DoS is normaly maliciously accomplished in the link layer of wireless mesh networks where the attacker sends loose packets to network devices that have to respond but take up so much processing power on the device that delays and latencies are caused in networks. Cryptographic systems can reduce the impact of such attacks.

Unauthorized access is another attack that often occurs in wireless mesh networks. This is known as the *man-in-the-middle* or MiTM attack. It usually occurs when the security mechanism implemented does not provide mutual authentication. Other attacks known to occur include *session hijacking*, *reflection attacks* and other attacks due to the abuse of cryptographic services. There are different authentication mechanisms to avoid this attack, but each of these have some drawbacks that the attacker can exploit. The password authentication mechanisms, *MAC address filtering*, and *IP address filtering* are rather vulnerable to MiTM attacks.

3.2 Research questions

The challenges for Wi-Fi mesh network authentication discussed in Section 3.1 lead us toward answering the two questions that we gave in Section 1.3 namely,

1. How can we characterize and compare existing Wi-Fi authentication mechanisms?
2. What are the best options for Wi-Fi mesh network authentication?

These questions lead us to study the characterization and features of existing authentication protocols, and learn how they can be applied to authentication on wireless mesh networks. Sections 3.3–3.6 and Tables 4.1–4.2 answer the first question. Sections 4.1–4.2 and their tables address the second question.

3.3 Overall approach

The aim of this project is to identify authentication mechanisms that addresses secure user access to the wireless mesh network, based on existing standards, mechanisms, protocols and current solutions under consideration in the standardization process of wireless mesh networks. Therefore, in this thesis, a gradual approach was followed. A literature review was first conducted in order to gather general information about wireless mesh networks, their specifications and characteristics, along with their application trends. Established definitions, descriptions and analysis of wireless mesh networks characteristics were spotted on the internet, papers, reviews, documents, standards and other materials focusing on security aspects in wireless mesh networks were highlighted for an in-depth study.

After we studied many issues related to wireless mesh network authentication, a wireless mesh network was designed and implemented physically for experimental purposes. Five Linksys Wrt54gl routers were used for practical work. Several designs of wireless mesh networks such as infrastructure wireless networks, mobile ad-hoc wireless mesh networks, infrastructure wireless *mesh* networks[1] and hybrid wireless mesh networks were designed and built for the testing of several authentication protocols.

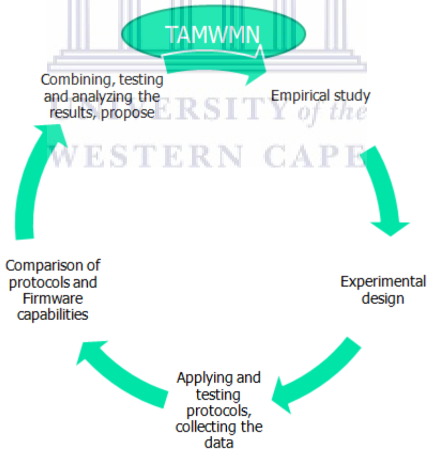

TAMWMN
Empirical study
Experimental design
Applying and testing protocols, collecting the data
Comparison of protocols and Firmware capabilities
Combining, testing and analyzing the results, propose

Figure 3.1: Outline of project design

Figure 3.1 outlines the steps followed in our project design.

3.4 Experimental design

As the Linksys Wrt54gl router allows firmware changes, five types of Linksys firmware were compared. Each of these has weaknesses and limitations, but also has advantages in respect of wireless mesh networks and available wireless mesh network

[1] Infrastructure wireless mesh networks are also known as *backbone wireless mesh networks*.

packages. Unfortunately, all of them have the limitation of no authentication and user access management on the wireless mesh network. We compared 14 types of user access mechanisms on a single-hop wireless network and several designs of wireless mesh network. In order to identify suitable solutions we tested various packages using OpenWrt and Freifunk firmware. We selected Freifunk firmware for our experimental network and applied BATMAN and OLSR to the wireless routers as routing protocols. We compared the facilities and features as listed in Chapter 4.

In order to evaluate and characterize the 14 authentication mechanisms described in Section 2.1 we designed a wireless mesh network in the main building of Kabul University Computer Science Faculty which has two floors. Two routers were installed on the first floor and three others on the second floor. On this physical mesh network we tested the connections and user access management. We installed OLSR and used the visualization package of OLSR which can calculate the route, show the metrics, show the accessibility of each point, and show the quality of each link as well. Figure 3.2 shows the network topology, IP addresses, name of routers and quality of links generated by the OLSR visualization packages.

3.4.1 Network Design

Our first wireless network design was not a mesh network. We found that the Linksys Wrt54gl has changeable firmware but its default firmware does not support mesh networks. However, the open source OpenWrt firmware is capable of supporting mesh networks. This makes OpenWrt amenable to experimentation. We went on to test the features of Freifunk, Meshcom OpenAP, and DD-WRT.

We tested authentication protocols and user access management protocols in infrastructure wireless networks and several types of wireless mesh network. Chapter 2 described each protocol theoretically. Section3.4.2 provides more practical detail of various designs of wireless mesh networks, using Linksys firmware.

3.4.2 Mesh network design

The study required us (1) to select the devices to use for wireless mesh networks, (2) to understand the software and hardware requirements, and (3) to study the design of wireless mesh networks. In our research, we used Wrt54gl wireless routers because of the ease of altering the firmware for the purposes of comparison and because we are familiar with configuring Linux-based firmware used on these routers.

Five types of firmware were used for comparing and testing mesh network authentication capabilities. After designing and setting up a real mesh network, we tested and compared the firmware and protocols. The steps below were followed in designing a wireless mesh network:

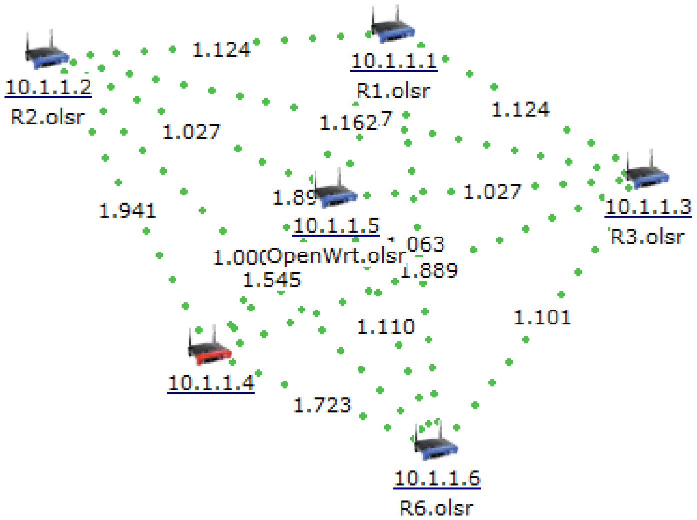


Figure 3.2: Visualization of mesh network with OLSR

1. Upload firmware which supports wireless mesh networks, such as OpenWrt, Meshcom OpenAP, DD-WRT, or OpenWrt-g-Freifunk.

2. Upload the recommended and necessary packages to support the mesh network.

3. Plan the IP addresses.

4. Design the network and the location of routers.

5. Every router needs the same SSID or ESSID, same *basic service set identifier* (BSSID), same channel, and the same setting to be wireless mesh points.

We can use the web interface of the firmware, but we can do only a limited amount of configuration of the networks and protocols from the web interface. We have to use `ssh` or `telnet` to do the configuration, and use a cross compiler to compile missing firmware packages.

There are three types of wireless mesh network: (1) Mobile ad-hoc networks in which mobile clients are connected to each other without using any router. (2)

In infrastructure wireless mesh networks the routers are connected to each other without being connected to the clients. In this type of wireless mesh network, routers are connected in ad-hoc mode and to an AP directly by cable and are connected to the router mesh points for access of clients in the network. (3) There are hybrid wireless mesh networks in which all the routers and mobile clients are connected to one another (Akyildiz and Wang, 2005). Figure 3.3 illustrates the design of a hybrid wireless network.

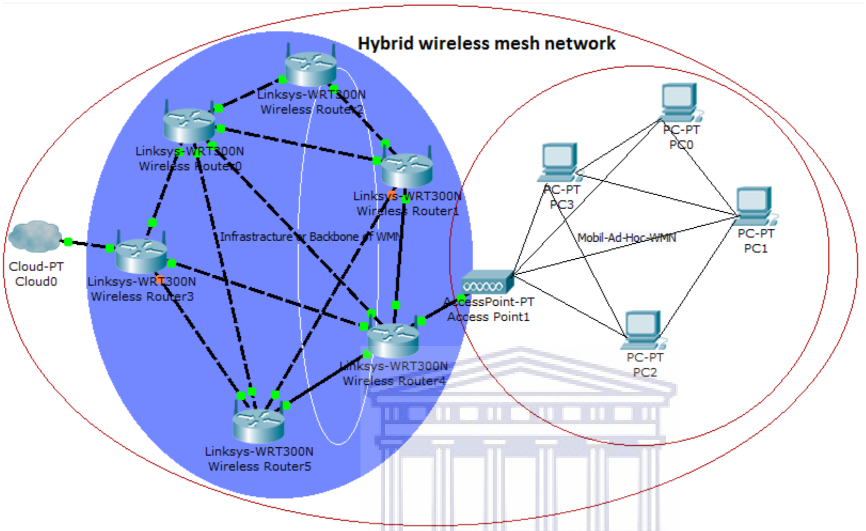


Figure 3.3: Design of a hybrid wireless mesh network

User access management has specific software and hardware requirements to be able to access wireless networks. If the devices do not have the necessary software and hardware then access will be denied to devices The connections between wireless routers and clients are also based on a specific policy. Figure 3.4 shows required modes for wireless routers to be able to associate and communicate with each other. Table 3.1 shows the type of connection between different wireless modes. Mesh networks can be used without any wireless AP, but we need to set up a computer with a wireless card based on the authentication that is used in mesh points. Most of the wireless card drivers only support basic encryption and basic authentication mechanisms such as WEP shared key, WPA, and 802.1X. If some other authentication protocols such as EAP, RADIUS, TLS, TTLS are needed, then additional software must be installed. For example, think vantage access software is useful for wireless connections but it does not have any authentication mechanism for WPA or RADIUS.

Table 3.1: Types of network connections

Type of network	Connection
Ad-hoc to ad-hoc "ad-hoc only can talk with ad-hoc"	Yes
Ad-hoc to client	No
Client to access point master	Yes
Access point master to access point master	No
Ad-hoc to access point master	No
Client managed to client managed	Yes

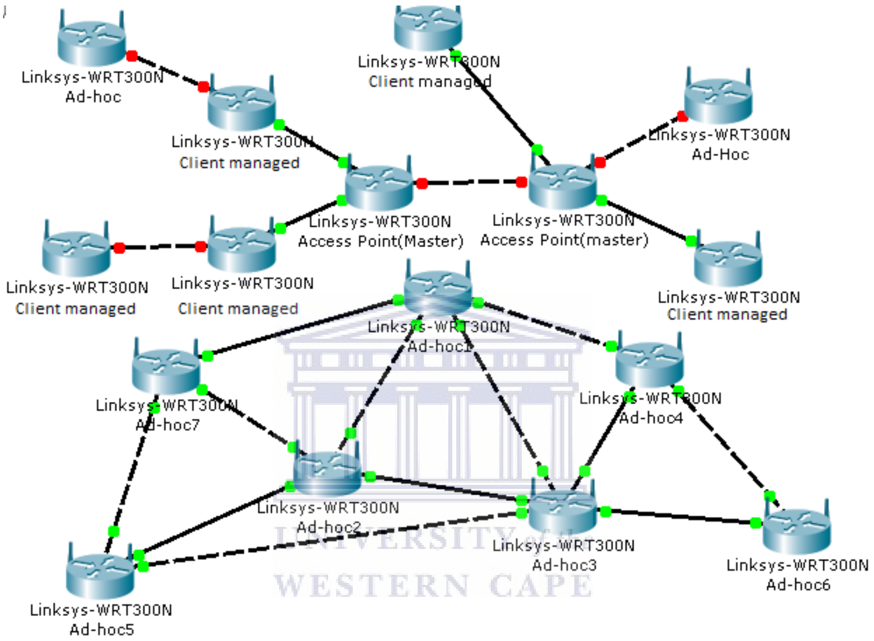


Figure 3.4: Wireless network connection modes

The hardware manufacturers usually provide specific drivers and software online for each operating system to support ad-hoc authentication. For Intel computers the website: `/www.intel.com/support/wireless/wlan/sb/cs-010623.htm` is very useful.

We also investigated VPN in a backbone wireless mesh network, and applied OpenVPN but it is based on a central server and clients which cannot be implemented in wireless mesh networks, because all the mesh routers have to connect to the central router and pass the packets through that router. They also have to generate key pairs for each client. A wireless router can never function as a central server for this type of authentication due to resource limitations. Figure 3.5 illustrates the design of our backbone wireless mesh network using an implementation of OpenVPN and an example of the key pairs generated is illustrated in Appendix A.

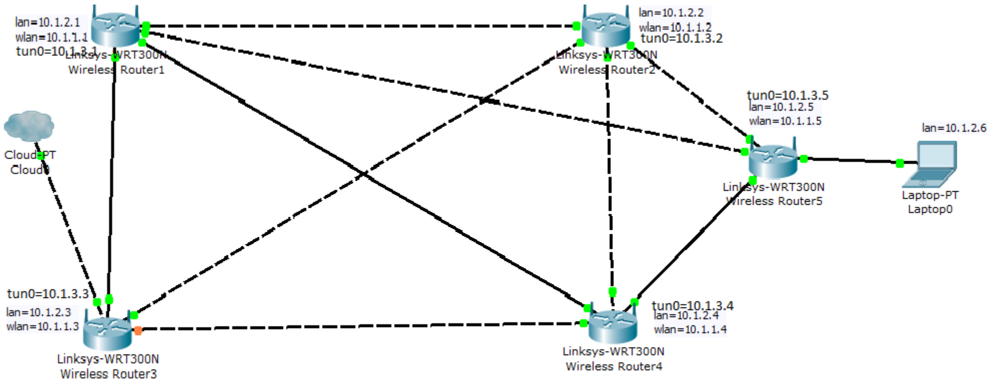


Figure 3.5: Design of backbone wireless mesh network

TincVPN is another type of VPN supporting OpenWrt, OpenWrt kamikaze, DD-WRT and Freifunk. It is open source and the packages are free available on the Internet. The facilities and features of TincVPN which can support distributed authentication and fragmentation of backbone wireless mesh networks is discussed in Section 3.6. We proposed an implementation of TincVPN to the backbone wireless mesh network as the best solution in respect of supporting wireless mesh networks because (1) it does not require a central server, (2) it has strong encryption, (3) it has dynamic authentication and (4) supports fragmentation. The configuration of TincVPN and an example of its public and private keys are given in Appendix B.

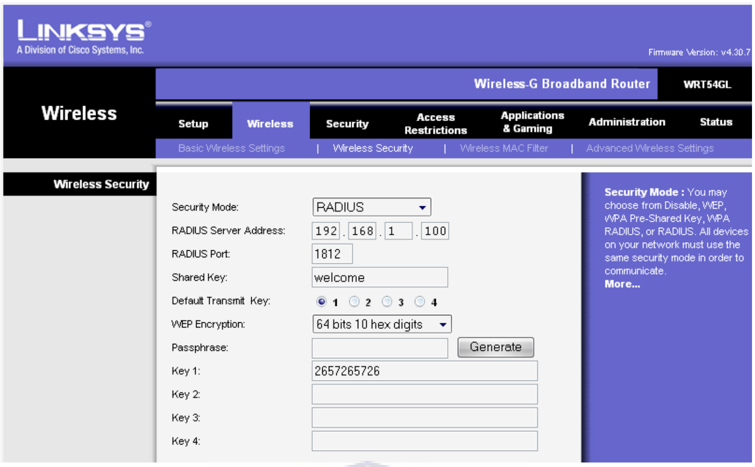
3.5 Mesh firmware choices

Linksys Wrt54gl supports different types of firmware. We compared five different types of firmware to identify those suitable for wireless mesh networks. A specific requirement was that the software must have open source code, and must be able to support authentication protocols that can be used in Wi-Fi mesh networks. These mesh capabilities and security features are characterized in Table 4.1.

3.5.1 Wrt54gl default firmware:

The default firmware for Linksys routers allows several protocols for authentication and encryption. It can act as a DHCP server, and it has a proper user interface, but for extending the network to large areas, wireless devices to ad-hoc mode, i.e.

peer-to-peer mode, is needed. This firmware does not support mesh networks. In order to adapt it for wireless mesh networks, the Linksys firmware of the router must be changed. Figure 3.6 illustrates the interface for Linksys Wrt54gl default firmware.


LINKSYS®
A Division of Cisco Systems, Inc.
Firmware Version: v4.30.7
Wireless
Wireless-G Broadband Router
WRT54GL
Setup
Wireless
Security
Access Restrictions
Applications & Gaming
Administration
Status
Basic Wireless Settings | Wireless Security | Wireless MAC Filter | Advanced Wireless Settings
Wireless Security
Generate
More...

Security Mode: RADIUS

RADIUS Server Address: 192 . 168 . 1 . 100

RADIUS Port: 1812

Shared Key: welcome

Default Transmit Key: ● 1 ○ 2 ○ 3 ○ 4

WEP Encryption: 64 bits 10 hex digits

Passphrase:

Key 1: 2657265726

Key 2:

Key 3:

Key 4:

Security Mode : You may choose from Disable, WEP, WPA Pre-Shared Key, WPA RADIUS, or RADIUS. All devices on your network must use the same security mode in order to communicate.

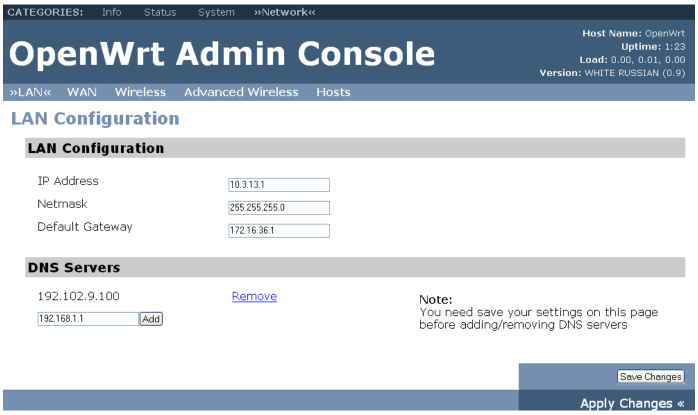
Figure 3.6: Web page of Wrt54gl default firmware

3.5.2 OpenWrt firmware:

OpenWrt is Linux-based Linksys firmware that has four wireless modes: (1) client, (2) client bridge, (3) ad-hoc and (4) AP. It has different authentication and encryption protocols for each mode. WEP, WPA and WPA2 pre-shared keys are available on client and client bridge mode. Only WEP is available for ad-hoc mode but it does not support authentication. In the OpenWrt AP mode the following are all available: WPA pre-shared key, WPA RADIUS, WEP, WPA2 pre-shared key only, WPA RADIUS only, WPA2 pre-shared key mixed, WPA2 RADIUS mixed and RADIUS. Figure 3.7 illustrates the interface for OpenWrt firmware.

3.5.3 Meshcom OpenAP firmware

Meshcom OpenAP is Linux-based firmware that provides limited security services. Meshcom OpenAP currently supports three types of access management, namely, Open, Closed, and a secure mode. In *open* mode the identity of the peer is not checked and the link becomes automatically authenticated and everyone in the coverage area can access the wireless network. In *closed* mode links are never authenticated. In this mode, initially the fixed node is set up to use open authen-


CATEGORIES: Info Status System »Network«
»LAN« WAN Wireless Advanced Wireless Hosts
Remove
Add
Save Changes
Apply Changes «

OpenWrt Admin Console

Host Name: OpenWrt
Uptime: 1:23
Load: 0.00, 0.01, 0.00
Version: WHITE RUSSIAN (0.9)

LAN Configuration

LAN Configuration

IP Address 10.3.13.1
Netmask 255.255.255.0
Default Gateway 172.16.36.1

DNS Servers

192.102.9.100
192.168.1.1

Note:
You need save your settings on this page before adding/removing DNS servers

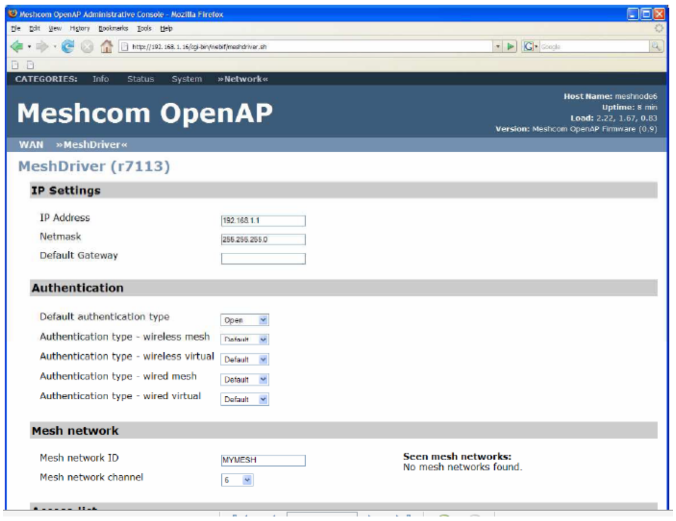
Figure 3.7: OpenWrt firmware web page

tication with all links. *Secure remote password* (SRP) protocol is used to verify authorization of the parties by proving the mutual knowledge of a secret phrase or key—pre-shared key authentication. In addition, a *pair-wise master key* (PMK) is derived for link level encryption. Authentication mode can be set separately for different types of links by selecting a type other than "Default" from the pull down Menu. Note that SRP authentication is not available for virtual links, because SRP authentication requires a running Meshcom OpenAP mesh driver at both end points.

Meshcom OpenAP supports user access management by setting an *access control list* (ACL) in order to control which mesh nodes are allowed to communicate with this fixed node. The access list may either be a white list—with only allowable nodes in the list—or a blacklist—which allows all nodes except the listed nodes. Nodes can easily be added or remove by using the interface. Figure 3.8 illustrates the Meshcom OpenAP web page.

3.5.4 Freifunk firmware

Freifunk is Linux-based and open source firmware that has changeable source code from which other versions can be developed. Freifunk firmware is an extension of OpenWrt. Using this firmware, packages can be installed and configured from a list of uninstalled packages. Details for installation and configuration are available on the Freifunk webpage, as can be seen in Figure 3.9. All available Freifunk packages can be viewed and the ones needed can be installed to the router. Freifunk firmware has been localized in many languages. We have started localizing Freifunk into Dari.


Meshcom OpenAP Administrative Console - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.1.16/cgi-bin/webif/meshdriver.sh
CATEGORIES: Info Status System »Network«
Meshcom OpenAP
WAN »MeshDriver«

Host Name: meshnode6
Uptime: 8 min
Load: 2.22, 1.67, 0.83
Version: Meshcom OpenAP Firmware (0.9)

MeshDriver (r7113)

IP Settings

IP Address	192.168.1.1
Netmask	255.255.255.0
Default Gateway	

Authentication

Default authentication type	Open
Authentication type - wireless mesh	Default
Authentication type - wireless virtual	Default
Authentication type - wired mesh	Default
Authentication type - wired virtual	Default

Mesh network

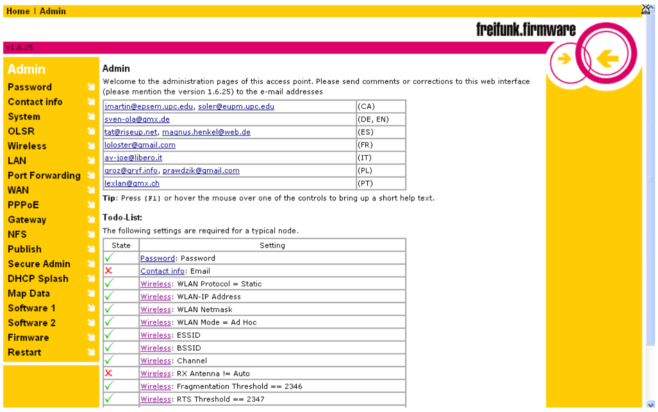
Mesh network ID	MYMESH
Mesh network channel	6

Seen mesh networks:
No mesh networks found.

Figure 3.8: The Meshcom OpenAP web page

Freifunk has three wireless modes namely: master access point, managed client and ad-hoc mode. Freifunk can support closed networks as well, a list of IP or MAC addresses can be blocked, and it also has a gateway filtering tool that can filter all incoming and outgoing traffic.

Freifunk has OLSR routing to provide communication between other network devices that are in wireless range, and have the same channel and the same mode. In our research `Openwrt-g-freifunk-1.7.4-en` firmware for mesh routers was used. Freifunk has several packages freely available on the Freifunk web site.



Admin

Welcome to the administration pages of this access point. Please send comments or corrections to this web interface (please mention the version 1.6.25) to the e-mail addresses

jmartin@epsem.upc.edu, soler@eupm.upc.edu	(CA)
sven-ola@gmx.de	(DE, EN)
tat@riseup.net, magnus.henkel@web.de	(ES)
loloster@gmail.com	(FR)
av-joe@libero.it	(IT)
groz@gryf.info, prawdzik@gmail.com	(PL)
lexlan@gmx.ch	(PT)

Tip: Press [F1] or hover the mouse over one of the controls to bring up a short help text.

Todo-List:

The following settings are required for a typical node.

State	Setting
✓	Password: Password
✗	Contact info: Email
✓	Wireless: WLAN Protocol = Static
✓	Wireless: WLAN-IP Address
✓	Wireless: WLAN Netmask
✓	Wireless: WLAN Mode = Ad Hoc
✓	Wireless: ESSID
✓	Wireless: BSSID
✓	Wireless: Channel
✗	Wireless: RX Antenna != Auto
✓	Wireless: Fragmentation Threshold == 2346
✓	Wireless: RTS Threshold == 2347

Figure 3.9: The Freifunk firmware web page

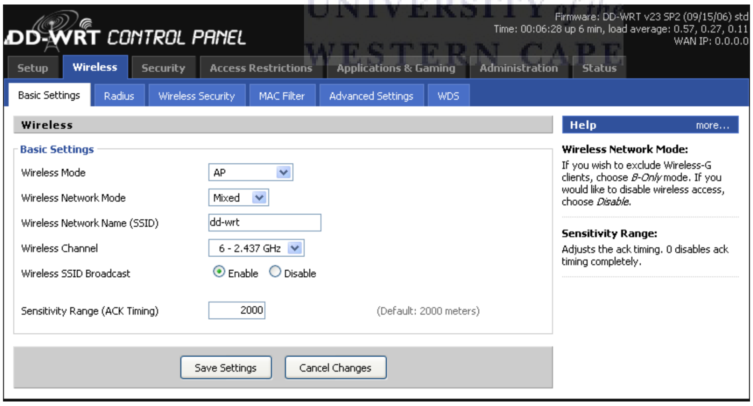
Its packages can be downloaded and installed on the router, followed by proper configuration of the mesh network. Packages can be installed through `ssh` or the command line user interface. With `ssh`, updating the list of available packages is feasible and the packages can be installed faster than via the web interface. For updating and installing the packages the following comments should be followed:

```
# ipkg update  "update list of available packages"
# ipkg install "install this list of packages"
```

Providing Internet connections with a non hot spot wireless router that has Freifunk firmware is not very difficult but it differs from the other firmware. *OLSR* and HNA4 should be configured properly.

3.5.5 DD-WRT Firmware

DD-WRT is free Linux-based firmware with good security capabilities that provides many encryption and authentication capabilities. DD-WRT supports EAP, WPA pre-shared key, RADIUS, MAC filtering, IP address filtering, closed network, and other protocols that are shown in Table 4.1. It also supports IPv6. It can use RIPv2, BGP, OLSR, Tinc, AODV and OSPF as routing protocols. DD-WRT works on different wireless modes such as ad-hoc, client, client bridge, and access point mode. Figure 3.10 shows the interface of DD-WRT .



Firmware: DD-WRT v23 SP2 (09/15/06) std
Time: 00:06:28 up 6 min, load average: 0.57, 0.27, 0.11
WAN IP: 0.0.0.0

Wireless

Basic Settings

Wireless Mode: AP
Wireless Network Mode: Mixed
Wireless Network Name (SSID): dd-wrt
Wireless Channel: 6 - 2.437 GHz
Wireless SSID Broadcast: ● Enable ○ Disable
Sensitivity Range (ACK Timing): 2000 (Default: 2000 meters)

Wireless Network Mode:
If you wish to exclude Wireless-G clients, choose *B-Only* mode. If you would like to disable wireless access, choose *Disable*.

Sensitivity Range:
Adjusts the ack timing. 0 disables ack timing completely.

Figure 3.10: The DD-WRT firmware web page

In the African *building a rural wireless mesh network* project DD-WRT was used in the non backbone wireless routers or APs (Johnson, 2007). They connected two Linksys routers back-to-back where one of them works as a backbone wireless mesh router that is connected to other mesh points in ad-hoc mode and the second one is configured in AP mode for the easy connection of non mesh clients.

3.6 Protocol modification

We addressed the first research question in Section 3.3–3.5 and Tables 4.1–4.2, in this section and Section 4.2 we address the second research question. Section 3.4 describes our experiments that use five Linksys Wrt54gl routers and a laptop to test the security and mesh features of firmware, to apply various wireless routing protocols to test the capability and compatibility of authentication protocols for Wi-Fi backbone mesh networks.

As a result of our attempts to adapt existing authentication protocols for Wi-Fi backbone mesh and distributed fragmentated networks, we found that existing authentication protocols also characterized in Section 4.1 are not suitable or applicable in a Wi-Fi backbone mesh network.

3.6.1 WPA_upplicant

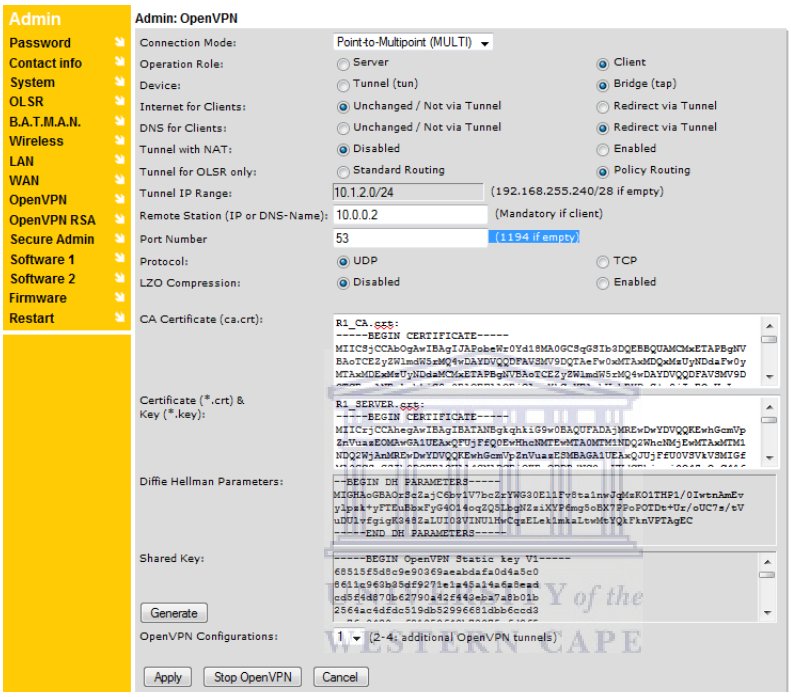
WPA_supplicant is the supplicant for BSD or Windows client machines. WPA and WPA2 need this supplicant for key negotiation and roaming of authentication and association. WPA_supplicant is licensed software and users are expected to abide by the licensing terms. Thompson et al. also used WPA_supplicant and EAP-edge and Free RADIUS as an authentication server in their experimental network as pointed out in Section 2.2.2 on Page 38. Free RADIUS is free software which can be installed on the router and can be used for authentication. The router that is loaded with Free RADIUS functions as the authentication server.

We tested several of wireless authentication protocols and chose WPA_supplicant to adapt EAP-edge, WPA with WPA_supplicant and `openssl` to make them compatible with fragmented backbone wireless mesh networks. We replaced the EAP-TLS of the Freifunk WPA_supplicant package to EAP-edge and combined it in a Linux operating system. After modification of WPA_supplicant we used a cross compiler to compile the package and adapted the package to the Freifunk format. We uploaded the package for the routers and tested it. The analysis and results are described in Section 4.2.

3.6.2 OpenVPN and tincVPN

Our experiments revealed that only VPN can provide a secure connection for the backbone of a wireless mesh network. In Section 2.2.1, we pointed out that OpenVPN is free software available on Freifunk firmware. In this subsection we tested

OpenVPN and TincVPN . We installed OpenVPN and tested it using six routers and created a tunnel for a backbone Wi-Fi mesh network and exchanged keys between routers. The results of our experiments are described in Section 4.2. Figure 3.11 shows the graphic interface of OpenVPN in Freifunk with the private and public keys of the server. An example of some certificate keys is shown in detail in Appendix A, and the results are described in Chapter 4. Figure 3.12 shows the



Admin: OpenVPN

Connection Mode: Point-to-Multipoint (MULTI)

Operation Role: ○ Server ● Client

Device: ○ Tunnel (tun) ● Bridge (tap)

Internet for Clients: ● Unchanged / Not via Tunnel ○ Redirect via Tunnel

DNS for Clients: ○ Unchanged / Not via Tunnel ● Redirect via Tunnel

Tunnel with NAT: ● Disabled ○ Enabled

Tunnel for OLSR only: ○ Standard Routing ● Policy Routing

Tunnel IP Range: 10.1.2.0/24 (192.168.255.240/28 if empty)

Remote Station (IP or DNS-Name): 10.0.0.2 (Mandatory if client)

Port Number 53 (1194 if empty)

Protocol: ● UDP ○ TCP

LZO Compression: ● Disabled ○ Enabled

CA Certificate (ca.crt):

```
R1_CA.crt:
-----BEGIN CERTIFICATE-----
MIICSjCCAbOgAwIBAgIJAPobeWr0Yd18MA0GCSqGSIb3DQEBBQUAMCMxETAPBgNV
BAoTCEZyZWlmdW5rMQ4wDAYDVQQDFAVSMV9DQTAeFw0xMTAxMDQxMzUyNDdaFw0y
MTAxMDExMzUyNDdaMCMxETAPBgNVBAoTCEZyZWlmdW5rMQ4wDAYDVQQDFAVSMV9D
```

Certificate (*.crt) & Key (*.key):

```
R1_SERVER.crt:
-----BEGIN CERTIFICATE-----
MIICrjCCAhegAwIBAgIBATANBgkqhkiG9w0BAQUFADAjMREwDwYDVQQKEwhGcmVp
ZnVuazEOMAwGA1UEAxQFUjFfQ0EwHhcNMTEwMTA0MTM1NDQ2WhcNMjEwMTAxMTM1
NDQ2WjAnMREwDwYDVQQKEwhGcmVpZnVuazESMBAGA1UEAxQJUjFfU0VSVkVSMIGf
```

Diffie Hellman Parameters:

```
--BEGIN DH PARAMETERS-----
MIGHAoGBAOr8cZajC6bv1V7bcZrYWG30E11Fv8ta1nwJqMzKO1THP1/0IwtnAmEv
ylpzk+yFTEuBbxFyG4O14oqZQ5LbgNZziXYP6mg5oBX7PPoPOTDt+Ur/oUC7s/tV
uDUlvfgigK348ZaLUI03VINU1HwCqzELek1mkaLtwMtYQkFknVPTAgEC
-----END DH PARAMETERS-----
```

Shared Key:

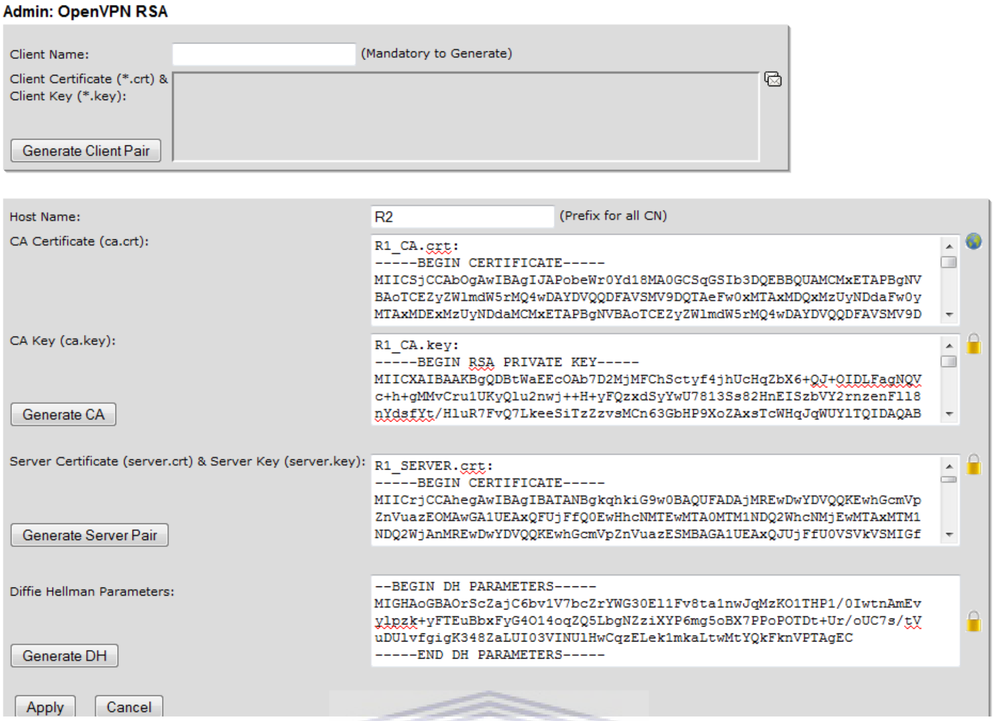
```
-----BEGIN OpenVPN Static key V1-----
68515f5d8c9e90369aeabdafa0d4a5c0
8611c963b35df9271e1a45a14a6a8ead
cd5f4d870b62790a42f443eba7a8b01b
2564ac4dfdc519db52996681dbb6ccd3
```

OpenVPN Configurations: 1 (2-4: additional OpenVPN tunnels)

Figure 3.11: Interface for creating OpenVPN certificates

interface OpenVPN uses to create a certificate authority, a server key and client key pairs based on RSA and Hellman parameters.

We tried TincVPN which supports distributed authentication on backbone wireless mesh networks. We used six Linksys routers for TincVPN and installed Tinc from the list of Freifunk packages. We created a tunnel for the experimental Wi-Fi backbone mesh network. We configured the tinc.conf, tinc-up, tinc-down and olsr.tinc. We introduced the IP address of virtual ports of the routers to each other and we created public and private keys for each router and exchanged the public keys. ping, traceroute, and the OLSR visualization software was used for testing of connections. We detailed the TincVPN authentication results in Chapter 4.
]



Admin: OpenVPN RSA

Client Name: (Mandatory to Generate)

Client Certificate (*.crt) & Client Key (*.key):

Generate Client Pair

Host Name: R2 (Prefix for all CN)

CA Certificate (ca.crt):

```
R1_CA.crt:
-----BEGIN CERTIFICATE-----
MIICSjCCAbOgAwIBAgIJAPobeWr0Yd18MA0GCSqGSIb3DQEBBQUAMCMxETAPBgNV
BAoTCEZyZWlmdW5rMQ4wDAYDVQQDFAVSMV9DQTAeFw0xMTAxMDQxMzUyNDdaFw0y
MTAxMDExMzUyNDdaMCMxETAPBgNVBAoTCEZyZWlmdW5rMQ4wDAYDVQQDFAVSMV9D
```

CA Key (ca.key):

```
R1_CA.key:
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQDBtWaEEcOAb7D2MjMFChSctyf4jhUcHqZbX6+QJ+OIDLFagNQV
c+h+gMMvCru1UKyQlu2nwj++H+yFQzxdSyYwU7813Ss82HnEISzbVY2rnzenFl18
nYdsfYt/HluR7FvQ7LkeeSiTzZzvsMCn63GbHP9XoZAxsTcWHqJqWUYlTQIDAQAB
```

Generate CA

Server Certificate (server.crt) & Server Key (server.key):

```
R1_SERVER.crt:
-----BEGIN CERTIFICATE-----
MIICrjCCAhegAwIBAgIBATANBgkqhkiG9w0BAQUFADAjMREwDwYDVQQKEwhGcmVp
ZnVuazEOMAwGA1UEAxQFUjFfQ0EwHhcNMTEwMTA0MTM1NDQ2WhcNMjEwMTAxMTM1
NDQ2WjAnMREwDwYDVQQKEwhGcmVpZnVuazESMBAGA1UEAxQJUjFfU0VSVkVSMIGf
```

Generate Server Pair

Diffie Hellman Parameters:

```
--BEGIN DH PARAMETERS-----
MIGHAoGBAOrScZajC6bv1V7bcZrYWG30El1Fv8ta1nwJqMzKO1THP1/0IwtnAmEv
ylpzk+yFTEuBbxFyG4O14oqZQ5LbgNZziXYP6mg5oBX7PPoPOTDt+Ur/oUC7s/tV
uDUlvfgigK348ZaLUI03VINUlHwCqzELek1mkaLtwMtYQkFknVPTAgEC
-----END DH PARAMETERS-----
```

Generate DH

Apply Cancel

Figure 3.12: Interface for creating RSA certificate authorities

3.7 Summary

This chapter described our research method. We discussed the security challenges of Wi-Fi mesh networks in Section 3.1. Existing authentication protocols have several problems such as centralization of authentication, several types of attack and most of the existing authentication protocols do not support Wi-Fi mesh networks. We formulated two research questions based on the problems identified in Section 3.2. Section 3.3 discussed our overall approach to try to discover authentication mechanisms that address the problem of Wi-Fi mesh networks. We used real wireless equipment instead of simulators to test existing authentication mechanisms and categorize them. Section 3.4 described our experimental infrastructure and mesh Wi-Fi network for testing and identifying Wi-Fi authentication mechanisms. In Section 3.5 we installed five types of firmware and tested them. After analyzing 14 types of Wi-Fi authentication protocols in Chapter 4 and studying of eight more Wi-Fi mesh network authentication protocols in Section 2.1 and Section 3.6 described our attempts at modifying WPA_supplicant and VPN to solve the authentication problem for Wi-Fi backbone mesh networks.

Our results are analyzed in Chapter 4.

Chapter 4

Analysis

In this chapter we tabulate the data we previously collected and reviewed. The comparison tables make it easy to read and understand the security capabilities of the firmware and authentication protocols. Section 4.1 including Tables 4.1–4.2 answers the first research question and Sections 4.2–4.3 answer the second research question with Tables 4.3–4.7.

4.1 Characterization of firmware and routing protocols

Wi-Fi mesh firmware and routing protocols are characterized and compared in this section. Table 4.1 compares the security features and ad-hoc capability of five types of firmware which we tested in the Linksys Wrt54gl routers using a real Wi-Fi network, Table 4.2 characterizes the three types of Wi-Fi routing protocols.

4.1.1 Wrt54gl firmware

The right firmware choice determines the wireless mesh network and authentication protocols that can be applied. We selected the Wrt54gl routers for our experimental network, because their firmware can be altered. Furthermore, the Wrt54gl open source firmware is Linux-based and is freely available on the Internet. We tested *five* types of Linksys Wrt54gl firmware, namely (1) the Linksys Wrt54gl default firmware, (2) OpenWrt, (3) DD-WRT, (4) Freifunk firmware, and (5) Meshcom OpenAP firmware. We characterized them to find the best one that has mesh network capability and security features. See Table 4.1.

4.1.2 Wi-Fi mesh routing protocols

The selection of route and packet forwarding between mesh points needs a Wi-Fi mesh network routing protocol that can find the route, select the best route and send the packet through the best route. When selecting a protocol during

Table 4.1: Comparison of Firmware

Security Features		Name of firmware				
		Linksys	OpenWrt	Meshcom	Freifunk	DD-WRT
User Access Control and Filtering	MAC filter	Yes	Yes	Yes	Yes	Yes
	IP filter	Yes	Yes	Yes	Yes	Yes
	Closed network	Yes	Yes	Yes	Yes	Yes
	ACL	No	Yes	Yes	Yes	Yes
Encryption and authentication	WEP	Yes	Yes	Yes	Yes	Yes
	WPA	Yes	Yes	Yes	Yes	Yes
	RADIUS	Yes	Yes	Yes	Yes	Yes
	SRP	No	No	Yes	No	No
	Open	Yes	Yes	Yes	Yes	Yes
Wi-Fi mode	Ad-hoc	No	Yes	Yes	Yes	Yes
	AP	Yes	Yes	Yes	No	Yes
	Client	No	Yes	No	No	Yes
	Client bridge	No	Yes	No	No	Yes
	Master access point	No	No	No	Yes	No
	Managed client	No	No	No	Yes	No

the design of a Wi-Fi mesh network, it must be compatible with authentication mechanism and have a low overhead. We characterized BATMAN, OLSR and AODV in Section 1.1 on Pages 5–7. This comparison lead us to select OLSR as an experimental wireless mesh network routing protocol, because: (1) it has less overhead, than AODV; (2) it can support larger wireless networks, and (3) it has good features for monitoring the networks. A team of developers in Afghanistan are already working to localize this protocol with Freifunk for the OLPC project which is being used in schools (OLPC, 2009). We installed both BATMAN and OLSR and tested them on a real wireless mesh network, using the testbed of Abolhasan et al. (2009).

Our characterization is discussed in Section 4.2. The testbed of M. Abolhasan (Abolhasan et al., 2009), and the performance analysis of BABEL by Juliusz Chroboczek (2010) show that current wireless routing protocols have limitations—they lack security, and suffer from slow convergence. Finding an efficient wireless routing protocol still remains an open research problem.

4.2 Characterization of Wi-Fi authentication protocols

Analysis of existing protocols is very important to determine the most suitable one for wireless mesh networks. To find the similarities, differences, capabilities

Table 4.2: Comparison of wireless mesh network routing protocols

Protocol	**Pros**	**Cons**
Ad hoc On-Demand Distance Vector (AODV)	–Works with critical hardware –Works with low bandwidth –Efficient in quality of services –Performs better in networks with static traffic	–Overhead in the network –Higher latency to find the route –No information about quality of routes –Flooding of packets occurs when searching for routes
Optimized Link-State Routing (OLSR)	–Reduces the overhead –Used in high density environment –Higher routing efficiency –Low latency to find the router –Knows the quality of link –Overhead independent of traffic	–Scalability is limited –Need enhanced hardware to store entire routing table –Slow convergence –Does not produce reliable and secure routing
Better Approach to Mobile Ad Hoc Networks (BATMAN)	–Less network overhead –Simple route selection –Specially designed for Wi-Fi mesh networks –It has good stability	–Slow convergence –Slow route selection and slow error checking –Possibility of routing loops –No aggregation –Possibility of collision

and limitations of different protocols we compared them. Tables 4.3–4.7 show a detailed comparison of the existing protocols.

4.2.1 Basic user access control

Table 4.3 shows a list of features and limitations of basic user access management, which we studied in Chapter 2.

The four access control mechanisms compared in Table 4.3 help us to control user access and manage who can / cannot access the network. Each has its own drawbacks, as discussed in Sections 2.1.1–2.1.14, and they can be used only in very small networks.

4.2.2 WEP and WPA

Table 4.4 shows detailed comparisons of WEP, WPA and WPA2—it compares different encryption methods, and shows that WEP has some security leaks such as a short key, key management reply attack, and weak encryption methods. WPA was developed to solve these problems of WEP—it supports a new encryption

Table 4.3: Characterization of basic user access control

Access control methods	**Pros**	**Cons**
Closed network	–Prevent from war driving –Hide location of network –Only authorized know the SSID	–Interference to others –Clear text messages –Difficult to memorize the SSID
MAC addressing filtering	–Easy to set up –Permit white list and deny black list –Wide hardware support	–Number of MAC addresses is limited in most firmware –Attacker can easily spoof the MAC address –MAC must be set manually
Shared key	–Supports many of the IEEE 802.11 standards –Users need know the key –WEP and WPA used for encryption –It uses a four-way handshake authentication –It supports a wide range of hardware	–Vulnerable to inside and packet spoofing attack –No mutual authentication –One shared key for all users –When changing the key we must advertise it to all users –Weak and deprecated authentication
User ID and password authentication	–All OSs support it –Easy to setup and manage –No need additional hardware –It can be changed by user's choice	–Vulnerable to dictionary attack –Central server failure –Weak security and plain text password –Complexity of maintenance

algorithm that has good security features and it can also support the *extensible authentication protocol* (EAP). Today WPA2 is very popular and is used on new hardware that has all of the WPA security features.

WPA2 can be used with ad-hoc links which is very useful for wireless mesh networks, since it has better security features than the two previous encryption methods, but its management is more difficult than WEP and WPA. Encryption and decryption between every mesh node takes time and adds to the overhead in mesh nodes (Maple et al., 2006).

The WPA protocol can work in mesh networks. It uses the strong encryption protocol namely AES and can also use the weaker TKIP. WPA_supplicant is a supplicant of WPA for the client side. The mesh point needs to work both as client and server. We can use WPA with Free RADIUS as well.

There are four types of WPA, each with its pros and cons as shown in Table 4.5. WPA supports dynamic authentication and encryption which is more

Table 4.4: Comparison of WEP, WPA and WPA2

Features	**WEP**	**WEP(802.11i)**	
		WPA	**WPA2**
Roles of calculation	RC4	RC4	AES
Key Size	40 /104 bit	128 bits encryption 64 bits authentication	128 bits
Data integrity	CRC-32	MIC	CCMP
Header integrity	None	MIC	CCMP
Key life	24 bit IV	48 bit IV	48 bit IV
Authentication	Shared key	PSK, RADIUS+ EAP	PSK, RADIUS+ EAP
Key management	None	EAP based	EAP based
Reply to attack	None	IV sequence	IV sequence
Dynamic encryption and authentication	None	Yes	Yes

useful for mesh networks and we can avoid administrator intervention.

Table 4.5: The pros and cons of WPA

Version of WPA	**Encryption**	**Authentication**	**Pros**	**Cons**
WPA-personal	TKIP	PSK	Easy to set up/ wide hardware support	Weak encryption, weak password and vulnerable to attack
WPA-Enterprise	TKIP	RADIUS+EAP	Robust authentication	Weak encryption, require RADIUS setup
WPA2_personal	AES	PSK	Easy to set up, strong encryption	Weak password and vulnerable to attack, needs new hardware
WPA2-enterprise	AES	RADIUS+EAP	Robust authentication and strong encryption	Needs new hardware, needs RADIUS setup by default.

Table 4.5 summarizes our analysis and experiments and shows which WPA is the most suitable authentication and encryption protocol for wireless mesh networks in respect of fragmentation and dynamic authentication. Unfortunately WPA does not work in backbone wireless mesh networks because it is an OSI link layer protocol. Furthermore, as described in Section 2.1.7 on Page 20, it is vulnerable to a dictionary attack.

4.2.3 Extensible authentication protocols

EAP is a protocol that is compatible with several other authentication protocols and supports a wide range of hardware and software. It can be used for several types of network. In this thesis we characterized and compared seven types of Wi-Fi authentication protocols that are used over EAP. Table 4.6 and Table 4.7 show the characterization of EAP over seven other authentication protocols.

Table 4.6: Characterization of authentication protocols

Authentication	**Pros**	**Cons**
EAP-IEEE 802.1X	–It is flexible with other protocols –Gets help from upper layer protocols –Uses four-way handshake authentication –Is extensible to new protocols –Has dynamic re-keying	–Single-hop authentication –always needs another protocol to support authentication –Physical port-based authentication –Does not support wireless backbone
EAP-RADIUS	–Works with different layers of the OSI model –Can be used with different authentication protocols –It uses UDP for fast data transmission	–Always needs a supporter for authentication –It is a central server mechanism –Does not support fragmentation
EAP-TLS	–It is a mutual authentication protocol –Supports link layer fragmentation –Wide hardware and software support –Resistant to attack –Strong authentication –Dynamic rekeying	–It is a point to point protocol –Does not support Layer 3 authentication –Needs client and server setup –Increased maintenance cost –Causes overhead in authentication time
EAP-TTLS	–Has optional mutual authentication –Dynamic rekeying –Resistant to attack –Strong authentication	–Mutual authentication causes removal of the packet sequence number –Can use less secure authentication such as MD5, PAP and CHAP
EAP-MD5	–Password-based authentication –Easy to implement –Supports a wide range of hardware and software	–Vulnerable to several attacks –Clashing hash values –Slow reconnection –No mutual authentication –No self protection
EAP-LEAP	–Dynamic rekeying –Mutual authentication –Symmetric key for data encryption –Self protecting	–Cisco proprietary software –Less device support –Vulnerable to dictionary attack –Slow reconnection
EAP-PANA	–Supports multi-hop wireless networks –Works on IP layer of OSI model –It supports mutual authentication –Fast reconnection	–Not resistant to attack –Lease device support –It is used in the IP layer which is still unsolved in wireless mesh networks (Zhang et al., 2009).

This table was collated from (Gast, 2005, Page 656), (Dantu et al., 2007), and (Zhang et al., 2009, Page 280).

Characterization and comparison of authentication protocols illustrate that current authentication protocols are not efficient and are unsuitable for secure connection of Wi-Fi mesh networks and adapting and developing new authentication protocols for multi-hop Wi-Fi mesh networks are still an open research question.

4.3 Analysis of Wi-Fi mesh network authentication

We did a deep literature review of Wi-Fi mesh network authentication. Selection of the hardware and software was important for us to support the Wi-Fi mesh network and have the facility of altering the code. We selected the Linksys Wrt54gl router, because its firmware is changeable.

We tested five types of firmware such as Linksys default firmware, OpenWrt, DD-WRT, Meshcom OpenAP and Freifunk. We characterized the Wi-Fi mesh capability and security features of the mentioned firmware. We selected Freifunk for our experimental network because it is open source firmware. Freifunk supports mesh networks. It has already been used in the Village Telco project in South Africa (Song, 2011) to provide low cost Internet and telephone access. Freifunk has also been used in the OLPC project in Afghanistan schools (OLPC, 2009).

The Afghanistan ministry of education (MoE) has a strategic plan to implement the OLPC project to most Afghanistan schools to improve the computer skill of students and provide then with Internet access. The secure connection between ad-hoc points is still an open question. The MoE is faced with the problem of securing network access, because they provide the bandwidth, which is limited, and there is barely enough bandwidth to accommodate everyone and certainly not enough to support bandwidth piracy due to the insecurity of access. The Village Telco problem has to deal with similar problems (Song, 2011).

Based on our characterization of authentication protocols we found the features and limitations of the existing authentication protocols. We tested the existing authentication protocols in a real network to determine if they work or not and what the reason is if they do not work.

We divided the existing authentication protocols into several groups such basic authentication mechanisms in Wi-Fi mesh networks, cryptographic protocols

Table 4.7: EAP over other protocols comparison table

	EAP-802.1x	EAP-RADIUS	EAP-TLS	EAP-TTLS	EAP-MD5
Mutual Authentication	Yes	Yes[1]	Yes	Yes	No
Identity privacy	No (RFC 2486)	Not enough privacy	No	Yes	No
Reply attack resistance	No (RFC 3748)	No	Yes	Yes	No
Dictionary attack resistance	Yes (RFC 3748)	Not enough resistance	Yes	Yes	No
Derivation of strong session keys	Yes	Yes	Yes	Yes	No
Server Authentication	User name, password and public key	User name, password	Public key certificate	Public key Certificate	None
Supplicant Authentication	User name, password and public key	User name, password	Public key Certificate	EAP,MS-CHAP,CHAP	Password Hash
Dynamic key generation	Yes	No	Yes	Yes	No
Ease of deployment	No	Yes	No	No	Yes
Over all security performance	Related to security protocol	Not enough security	Good	Good	Poor
Software support	Multiple Operating system support	Multiple Operating system support	Multiple Operating system support	Multiple OS support, needs 802.11 Cisco wireless card	Multiple operating system support
Wi-Fi mesh authentication	No	No	No	No	No

[1] Mutual authentication can work between the supplicant and the AP, but it does not operate between the AP and the RADIUS server.

and extensible authentication protocols.

4.3.1 WPA and WPA_supplicant

As a first step we selected WPA_supplicant and WPA to modify and adapt it for Wi-Fi mesh networks. Since WPA can support mutual authentication, dynamic encryption and authentication, it can also support roaming profiles, and it supports a wide range of hardware.

Because of the promising attributes of WPA we tried to adapt EAP-edge, which originated from TLS to WPA_supplicant, combined EAP-edge, and openssl with the WPA_supplicant package and used it with WPA. We combined them using a cross compiler to generate firmware compatible with the Freifunk package, but we could not successfully implement it because WPA_supplicant only works in Layer 2 of the OSI model whereas ad-hoc network connections require network layer protocols.

4.3.2 TincVPN

VPN is a public-private key based mechanism for securing connections in backbone Wi-Fi mesh networks, and it supports the IP Layer of the OSI model. We used OpenVPN but it did not work as pointed in Chapter 3, but TincVPN seems to be a promising solution for Wi-Fi mesh networks.

OpenVPN is not suitable for mesh networks in respect of distributed authentication and fragmentation. TincVPN supports encryption, distributed authentication and fragmentation. We used six routers for testing configurations and pair keys configured according to the configuration described in Appendix B.

Figure 3.5 on Page 54 illustrates the concept of implementing of TincVPN in backbone wireless mesh networks and fragmented wireless mesh networks.

Every mesh point in TincVPN can function as a gateway and the routing can be handled by Tinc itself, which is interesting for our research. But if a mesh point does not have a direct connection to the destination, the source will send the packet through the route which is nearest to the destination. The policy routing of Tinc is similar to the OSPF routing protocol.

During message exchange, mesh points, i.e. the routers, are aware only of their directly connected neighbors whose public keys are known to them. Tinc operates in the link layer and the IP layer. This characteristic of Tinc can be

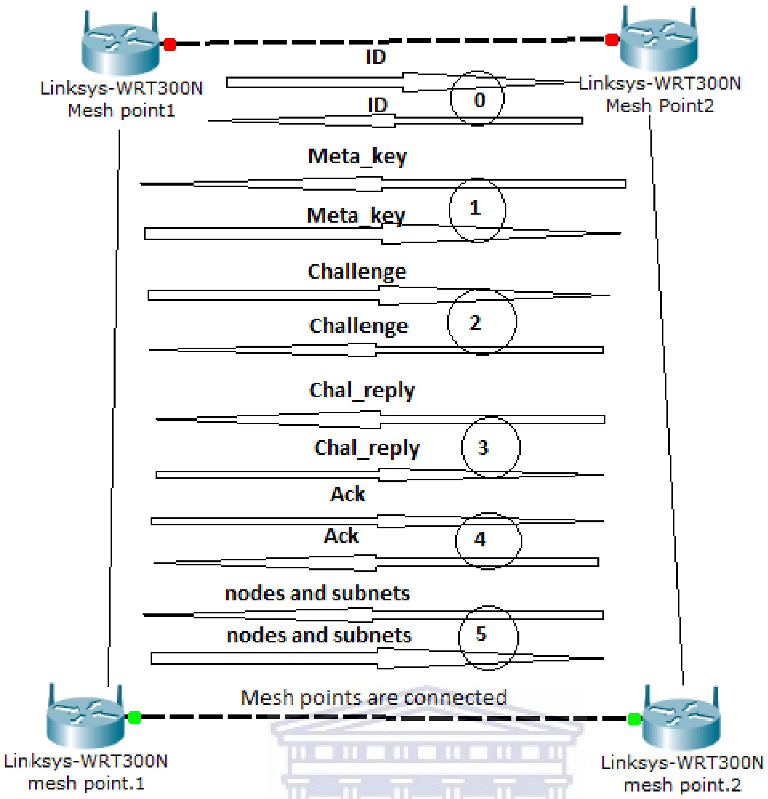


Figure 4.1: The TincVPN mesh point authentication process

applied to restrict the transmission of datagrams and frames.

Other wireless authentication protocols use the link layer of the OSI model and cannot support secure backbone wireless mesh networks connecting via the IP layer. So the link layer authentication protocols are not used for backbone wireless mesh networks any more.

Two channels are used by Tinc UDP and TCP. UDP is mostly used for carrying bulk data and TCP is used to carry some control data like key-exchange and routing information.

The exchange message uses a sequence number to prevent mesh points from attacks, because if two or more packages are received with same sequence number it is a sign that the package is not normal and this sequence number also works as an IV for Tinc .

Tinc also uses a traditional IV for the first block, but it is set at key-generation time and is the same for every packet. Tinc uses a metaprotocol over the TCP connection to exchange the control message between mesh points. Table 4.8 shows the numeric value and description of messages which we use in this

Table 4.8: Types of messages

Value	**Message**	**Purpose**
0	ID	host identification
1	Meta_key	keying material for metaconnection
2	Challenge	authentication challenge
3	Challenge_reply	reply to authentication challenge
4	Ack	acknowledgment of correction authentication
5	Status	status string for logging
6	Error	error notification
7	Temreq	request to terminate connection
8	Ping	keep-alive echo request
9	Pong	keep-alive echo reply
10	Add_subnet	add a subnet to graph
11	Del_subnet	delete a subnet from graph
12	Add_edge	add a node (host) to graph
13	Del_edge	delete a node (host) from graph
14	Key_changed	node has changed its key
14	Req_key	request a node's key
15	Ans_key	reply to key request
16	Packet TCP data	packet length and data

chapter.

Tinc supports VPN using TincVPN software for providing an authentication on wireless mesh networks, each mesh point behaves as a server and client simultaneously. Tinc exchanges six types of message from the start to the end of the authentication process, which we illustrate in Figure 4.1.

At the end of the authentication protocol, each Tinc point will inform its peer of all the other points and subnets that it knows about. As other points enter and leave the VPN network, each Tinc point will likewise inform their peers about the event.

Tinc uses openssl to provide encryption primitives but does not use the SSL protocol itself. Tinc can use any of the encryption algorithms or hash methods that openssl provides, so a Tinc user has a large variety of cryptographic systems available. By default, Tinc uses Blowfish and SHA-1, but algorithms such AES is also available.

As can be seen in Figure 4.1, the authentication protocol, after the ID message, the peers exchange the keys they will use for metaprotocol encryption. There are two criticisms of this procedure. First, each side determines its transmit key by itself, violating the rule that neither side should specify a key completely. On the other hand, the key is used only to transmit data from the point that specified it, so a point can't be tricked into using a weak key by its peer, which is what the rule is meant to prevent. That leaves open the possibility that one peer has a weak random number generator, but because both sides use the `openssl` random number generator, their results will most likely be similar.

The fact that UDP does not use the metaprotocol, and keys are encrypted in raw bits, are some weaknesses of `Tinc` but they do not affect the security of `Tinc`.

4.4 Summary

The characterization and comparisons in Tables 4.1–4.2 answer the first research question. This characterization led us to use Freifunk firmware in our experimental network since it is Linux-based and open source and used in several related projects such the OLPC project in Afghanistan (OLPC, 2009), universities of Afghanistan (MoHE Afghanistan, 2010), and rural wireless mesh network project in South African (Johnson, 2007). Characterization of Wi-Fi mesh routing protocols in Table 4.2 led us to select OLSR as routing protocols in our experimental network.

Characterization and comparison of 14 types of authentication protocols in Section 4.2 lead us to answer the second question. We found the limitation and challenge of existing authentication protocols in the Wi-Fi mesh networks. `TincVPN` is a very promising mechanism for backbone Wi-Fi mesh networks, it can encrypt the data, support authentication, and we can fragment the Wi-Fi mesh network. It works based on public key and private key and uses the `openssl` library.

Next, Chapter 5 concludes the thesis by summarizing it and discusses the results and conclusions and suggests future work.

Chapter 5

Conclusions

This chapter summarizes the thesis. The results and proposals for future work are also covered. In Section 5.1, the thesis summary, covers the background, the related work, the methodology, the results of the thesis and conclusion. Section 5.2 discusses the limitations of this research and future work is suggested in Section 5.3.

5.1 Thesis summary

Here we give a final synopsis of the entire thesis to conduct the reader briefly through the entire contents of the thesis for a clearer understanding of the research. We conclude by highlighting the results and limitations of this research, and with a view to the future work, we address an approach to surmount the limitations to assist other researchers in the field of authentication in wireless mesh networks.

5.1.1 Background

Wi-Fi networks are suitable for dense network coverage and we can extend Wi-Fi networks to Wi-Fi mesh networks in very densely covered areas. Wi-Fi mesh networks are cost effective but their access management remains a challenge (Lee et al., 2008).

Section 1.1, besides introducing the thesis, describes wireless networks, wireless mesh networks and their types. Wi-Fi routing protocols such OLSR (Clausen and Jacquet, 2003), BATMAN (Johnson et al., 2008), and AODV (Perkins et al., 2003) are also covered in Section 1.1. Section 1.2 motivates the thesis. There were several aspects we needed to study, namely, first, the efficiency of Wi-Fi mesh for extending network accessibility, and second, the requirements of security and access management of Wi-Fi mesh networks, based on the requirements of several Wi-Fi mesh projects such as the Afghanistan OLPC initiative (OLPC, 2009), and the Afghanistan universities wireless network (MoHE Afghanistan, 2010), the Village Telco project in the Bo-kaap (Song, 2011), and the South African wireless mesh network (Johnson, 2007). Section 1.3 covers the research questions. Section 1.4

explains the research methodology and finally Section 1.5 outlines the entire thesis.

5.1.2 Related Work

Chapter 2 on related work covers Wi-Fi infrastructure network authentication mechanisms and Wi-Fi mesh network authentication protocols and has two main parts, namely, 802.11 authentication mechanisms in Section 2.1 and Wi-Fi mesh network authentication in Section 2.2.

Section 2.1 covers 14 types of authentication protocols that are used in Wi-Fi infrastructure networks, such as closed networks in Section 2.1.1, MAC address filtering in Section 2.1.2, password authentication in Section 2.1.3, challenge-handshake authentication protocol in Section 2.1.4, shared key authentication Section 2.1.5, wired equivalent privacy in Section 2.1.6, Wi-Fi protected access in Section 2.1.7, Wi-Fi protected access II in Section 2.1.8, extensible authentication protocol in Section 2.1.9, EAP-remote authentication dial in user service in Section 2.1.10, EAP-transport layer security in Section 2.1.11, EAP-tunnel transport layer security in Section 2.1.12, EAP-message design 5 in Section 2.1.13, and lightweight extensible authentication protocol in Section 2.1.14. They only support single-hop Wi-Fi networks and other protocols must be used for Wi-Fi mesh networks.

The security of Wi-Fi multi-hop networks is much more complicated than Wi-Fi single hop networks because their architecture is unstable and is under continual change. Access control of mesh networks has distribution problems and central devices have a performance overhead. Wi-Fi mesh points moving from one location to another need to have access to the network all the time and everywhere (Lee et al., 2008).

Section 2.2 discusses Wi-Fi mesh network authentication protocols in two parts, under (1) central authentication mechanisms such as IEEE-802.1X (Cheikhrouhou et al., 2006), PANA (Khan and Akbar, 2006) and OpenVPN (Snader, 2007), and under (2) distributed authentication mechanisms such as URSA (Luo et al., 2004), EAP-AGE (Thompson et al., 2007), distributed authentication and key management (Husseiki, 2006), TincVPN (Tinc, 1998), and distributed authenticating mechanisms for Wi-Fi mesh networks proposed by several researchers such as Luo et al. (2004), Thompson et al. (2007), and by Lee et al. (2008).

5.1.3 Methods

We studied 14 types of Wi-Fi and eight more types of Wi-Fi mesh networks in Chapter 2 in Sections 2.1.1–2.1.14 and in Section 2.2. Chapter 3 on methods discusses the security challenges of Wi-Fi mesh networks in Section 3.1 where several challenges are pointed out. The security challenges prompted us to do this research and we organized the thesis around two research questions in Section 3.2. We designed experimental real Wi-Fi and Wi-Fi mesh networks in Section 3.4. The selection of firmware was covered in Section 3.5, based on the testing of five types of Linksys Wrt54gl firmware such as the Linksys default firmware, DD-WRT, Meshcom OpenAP, OpenWrt, and Freifunk were tested in this section. We also tested three types of Wi-Fi routing protocols such as OLSR, AODV, and BATMAN in Section 3.5.

After studying Wi-Fi and Wi-Fi authentication protocols in Chapter 2 and analyzing authentication protocols in Section 3.6 we attempted to modify the WPA_supplicant protocol and adapted it with EAP-AGE and tried use with WPA for Wi-Fi mesh networks. The analysis and results of this are discussed in Chapter 4 and Chapter 5.

5.1.4 Results

The empirical study was important to gain a deeper understanding of Wi-Fi mesh networks and their features and challenges.

We used Linksys Wrt54gl routers because we had access to the code for the firmware and we could use Linux-based open source firmware such as OpenWrt, DD-WRT, Meshcom OpenAP, Open AP, and Freifunk. The testing and characterization of five types of firmware in Section 4.1 was important to enable us to find the best one to support Wi-Fi mesh networks with adequate security features. We selected Freifunk firmware because its firmware is open source. Freifunk supports mesh networks. It is already being used in the Village Telco project (Song, 2011) in South African which provides Internet and telephone network access at a low cost. Freifunk is also used in the OLPC project in Afghanistan schools, where based on the MoE in Afghanistan's strategic plan, the OLPC initiative will be implemented in most of Afghanistan schools to improve the computer skills of students and provide them with Internet access (OLPC, 2009).

We analyzed three Wi-Fi routing protocols such as OLSR (Clausen and

Jacquet, 2003), AODV (Perkins et al., 2003) and BATMAN (Johnson et al., 2008) in Table 4.2, and we selected OLSR (Clausen and Jacquet, 2003), because it is an *open source* wireless mesh network routing protocol with a low overhead. It can support larger wireless networks. It has good features for the monitoring of networks, and Freifunk has a complete and compatible package to support OLSR. The Afghanistan OLPC initiative has a team that is localizing the Freifunk OLSR package to support Dari and Pashto (OLPC, 2009). Based on our characterization in Table 4.2 and Abulhasan's analysis (Abolhasan et al., 2009) developing a secure routing protocol for Wi-Fi mesh networks remains an open question.

The connections between ad-hoc points needs to be secured otherwise unauthorized users can access to the network. We characterized and compared 14 types of authentication protocols in Tables 4.3–4.7 to find the best option for access management and securing connections in Wi-Fi mesh networks. We tested the authentication protocols on five types of firmware in real mesh networks in Chapter 3 also we studied eight more authentication protocols related to this thesis in Section 2.2. We analyzed the authentication protocols in Chapter 4. The Wi-Fi ad-hoc network designed to be accessible for everyone and our analysis in Chapter 4, the analysis of Abolhasan et al. (2009) shows that the existing authentication protocols are not able to secure the backbone of Wi-Fi ad-hoc networks because most of them use OSI Layer 2 protocols and our aim was to secure the backbone of Wi-Fi mesh networks which rely on OSI Layer 3 protocols.

EAP-TTLS over PANA proposed by Khan and Akbar (2006) seems a promising authentication protocol, because PANA works on Layer 3 of OSI model and it supports multi-hop wireless networks, however, it requires a central server that restricts the extension of Wi-Fi mesh network. For backbone Wi-Fi mesh networks the routers are required to function simultaneously as client and server.

Wi-Fi mesh networks are extensible networks that grow over time. Several researchers such as (Khan and Akbar, 2006; Lee et al., 2008) proposed distributed authentication mechanisms to prevent overloading the central authentication server.

We tried to adapt EAP-AGE with WPA_supplicant as proposed by Thompson et al. (2007) and mentioned in Section 3.6 to localize the authentication and avoid Internet authentication. We tried to use it for distributed authentication, but we were not able to get the network to work using distributed authentication and fragmentation, because WPA_supplicant only works in OSI Layer 2 and we need to secure the network layer.

The existing authentication is not able to provide distributed authentication in respect of fragmentation, because we propose VPN for the backbone wireless mesh networks—it should be able to provide distributed authentication and fragment the backbone wireless mesh networks. Our experiments revealed that only VPN can provide a secure connection for the backbone of a wireless mesh network. In Section 2.2.1 we pointed out that OpenVPN is free software available on Freifunk firmware. We installed OpenVPN and described our tests in Chapter 3 using six routers. Based on our tests OpenVPN is not a suitable access management mechanism for Wi-Fi mesh networks because one VPN can secure the link between only two nodes per tunnel, and if we must secure all the links then we need a key pair for each link. This is difficult to achieve on a network with many nodes since the central server needs to store the public key of every node connected to it, overloading the central server.

We tested `TincVPN` on the backbone wireless mesh network and it seems to be the best solution that supports Wi-Fi backbone mesh networks because (1) it does not require a central server, (2) it has strong encryption, (3) it has dynamic authentication and (4) it supports fragmentation. The configuration of `TincVPN` is shown in Appendix B on Page 93 and an example of the public and private keys are given in Appendix B on Page 97. Finally, based on our experiments `TincVPN` is the best solution for backbone Wi-Fi mesh network. However, EAP-TTLS over PANA also can be used for smaller Wi-Fi mesh networks.

5.1.5 Conclusion

Wi-Fi mesh networks are extensible and promising networks in high density areas, but the security of Wi-Fi mesh network is still a big challenge (Akyildiz and Wang, 2005). Several routing protocols exist that can help to connect mesh points in Wi-Fi mesh networks (Johnson et al., 2008), but there are no security mechanisms for connecting mesh points. In Chapter 2 the existing authentication protocols are categorized into two major groups. Authentication protocols that work only in single-hop Wi-Fi networks and authentication protocols that work in Wi-Fi mesh networks.

Wi-Fi ad-hoc networks were designed and tested experimentally with various Wi-Fi authentication protocols. We found several limitations of authentication protocols by applying them to Wi-Fi mesh networks.

In Chapter 4 we characterized the 14 types of authentication protocols besides studying eight other related Wi-Fi mesh authentication mechanisms. `TincVPN` seems a promising mechanism for backbone Wi-Fi mesh networks as detailed in Chapter 4.

The existing authentication protocols are designed to work at the OSI link layer. We tried to find an authentication solution for backbone Wi-Fi mesh networks. PANA, designed for the IP layer of Wi-Fi networks is based on a central server, but that is unsuitable for distributed mesh networks. We propose using `TincVPN` as the way forward because it supports distributed authentication, fragmentation of the networks, supports the IP layer of the OSI model and enables a secure mesh mesh authentication mechanism.

5.2 Limitations

`TincVPN` provides encryption, distributed authentication, and fragmentation of the network, and seems a promising security mechanism, but it has some limitations, which are summarized as follows.

1. Figure 5.1, shows a screen shot of our implementation of an experimental network to test `TincVPN` in a Wi-Fi mesh network. Routers R1, R2, R6 and R3 are connected and send the packets to each other through unsecure wireless ports. The same routers also have another virtual port or tunnel that

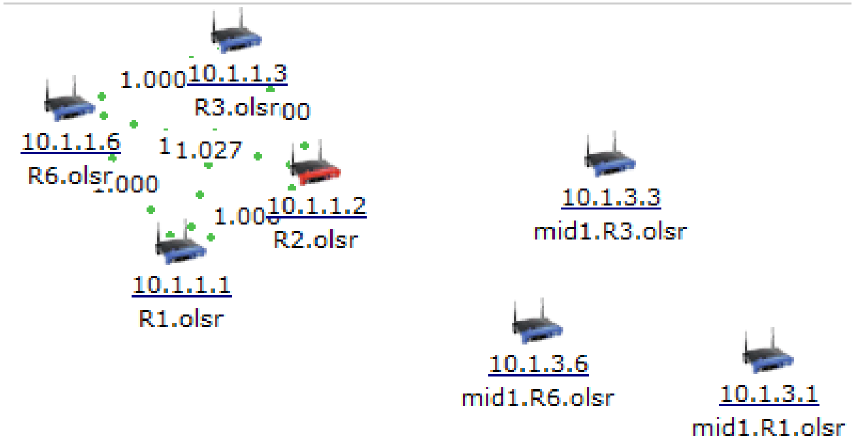


Figure 5.1: `TincVPN` connections

are created by TincVPN, but they are only connected through wireless ports, and packets are not forwarded through the secure tunnel despite the configuration files, IP tables, OLSR settings and firewall configurations. Applying `traceroute` and `ping` show that they packets are forwarded through wireless ports that are insecure in ad-hoc mode. This demonstrates that TincVPN still needs some improvement.

2. The manual exchange of public keys for mesh points are time consuming and needs a lot of network administrator intervention.

3. It causes overload in the devices, since they have to hold the keys of all the mesh points that are part of the mesh network.

5.3 Future work

VPN is a good solution for securing the connections of the backbone Wi-Fi mesh network, but it should support multi-hop mesh points. It must support distributed authentication and it must support network fragmentation. TincVPN is uses public and private keys, supports distributed authentication and fragmentation and supports multi-hop mesh points where each of the mesh points behave as client and server at the same time.

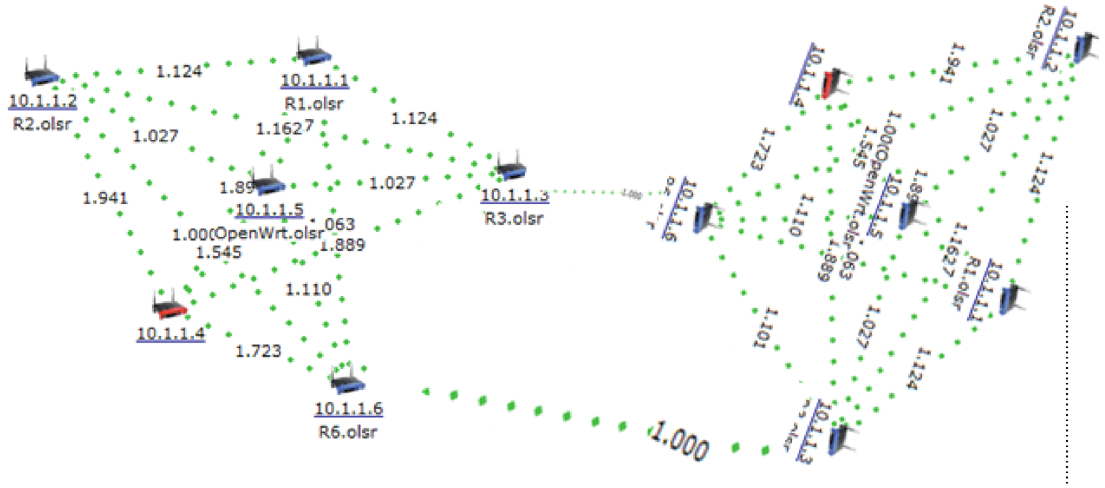


Figure 5.2: Fragmentation with TincVPN

If we want to implement TincVPN we have to exchange the public keys of the routers on all mesh points included the network. Figure 5.2 illustrates the concept of implementation of TincVPN in Wi-Fi backbone mesh networks.

There are two groups of routers in Figure 5.2. Six routers are in the first group and six routers are in the second group and there are only two connections between the two groups. All the routers which are in the one group must share their public keys with each other and only one of them has to share the public key with one of other routers in the second group. In order to introduce some redundancy to prevent a bottleneck more than one connection between the groups of routers can be introduced. We tried this setting but as we did not have more time to work on this research we couldn't connect the routers through a tunnel, and we faced some problems in the IP security protocol because it did not allow the routing protocol to send or receive the packets through the tunnel. So we suggest using TincVPN for securing the connection between backbone routers and fragment the network to spread the load of routers. It can provide distributed authentication by default.

Bibliography

Aboba, B. and Calhoun, P. (2003). RADIUS (Remote authentication dial in user service) support for extensible authentication protocol (EAP). RFC 3579.

Aboba, B. and Simon, D. (1999). PPP EAP TLS authentication protocol. RFC 2716.

Aboba, B., Simon, D., and Eronen, P. (2008). Extensible authentication protocol (EAP) key management framework. RFC 5247, updates RFC 3748.

Abolhasan, M., Hagelstein, B., and Wang, J. C.-P. (2009). Real-world performance of current proactive multi-hop mesh protocols. In *IEEE Asia-Pacific Conference on Communications (APCC)*, Shanghai, China.

Acharya, R., Vaitiynathan, V., and Chelliah, P. R. (2009). Wireless LAN security—challenges and solutions. *International Journal of Computer and Electrical Engineering*, 1(3):258–265.

Akyildiz, I. F. and Wang, X. (2005). A survey on wireless mesh networks. *IEEE Communications Magazine*, 43(9):23–30.

Andel, T. R. and Yasinac, A. (2006). On the credibility of MANET simulations. *IEEE Computer*, 39(7):48–54.

Blunk, L., Vollbrecht, J., Aboba, B., Carlson, J., and Levkowetz, H. (2004). Extensible authentication protocol (EAP). RFC 3748.

Cheikhrouhou, O., Laurent-Maknavicius, M., and Chaouchi, H. (2006). Security architecture in a multi-hop mesh network. In *Proceedings of 5th Conference on Security and Network Architectures (SAR 2006)*.

Chroboczek, J. (2010). The babel routing protocol. URL `http://tools.ietf.org/html/draft-chroboczek-babel-routing-protocol-05`.

Clausen, T. and Jacquet, P. (2003). Optimized link state routing protocol (OLSR). RFC 3626.

Congdon, P., Aboba, B., Smith, A., Zorn, G., and Roese, J. (2003). IEEE 802.1X remote authentication dial in user service (RADIUS). RFC 3580.

Dantu, R., Clothier, G., and Atri, A. (2007). EAP methods for wireless networks. *Computer Standards & Interfaces*, 29:289–301.

Das, M. L., Saxena, A., and Gulati, V. P. (2004a). A dynamic ID-based remote user authentication scheme. *IEEE Transactions on Consumer Electronics*, 50(2):629–631.

Das, M. L., Saxena, A., and Gulati, V. P. (2004b). A novel remote user authentication scheme through dynamic login identity. In *Proc. of the International Workshop on Distributed Computing (IWDC)*. Springer-Verlag, India. LNCS 3326.

D.Forsberg, Ohba, Y., Patil, B., and Tschofenig, H. (2007). Protocol for carrying authentication for network access (PANA). URL `ftp://ftp.rfc-editor.org/in-notes/internet-drafts/draft-ietf-pana-pana-18.txt`.

Flickenger, R., Aichele, C. E., Fonda, C., Forster, J., Howard, I., Krang, T., and Zennaro, M. (2006). *Wireless Networking in the Developing World*. Limehouse E-Book. URL `http://wndw.net`.

Forsberg, D., Ohba (Ed.), Y., Patil, B., Tschofenig, H., and Yegin, A. (2008). Protocol for carrying authentication for network access (PANA). RFC 5191.

Fourty, N., Val, T., Fraisse, P., and Mercier, J.-J. (2005). Comparative analysis of new high data rate wireless communication technologies “from Wi-Fi to WiMAX”. In *IEEE ICAS’05*, pages 1–6.

Frank, R. (2006). Authentication in wireless mesh networks. Master’s thesis, Université Joseph Fourier.

Funk, P. and Blake-Wilson, S. (2002). EAP Tunneled TLS authentication protocol (EAP-TTLS). URL `http://tools.ietf.org/html/draft-ietf-pppext-eap-ttls-01`.

Galperin, H. (2005). Wireless networks and rural development: Opportunities for latin america. *Information Technologies & International Development*, 2(3):47–56.

Gast, M. (2005). *802.11 Wireless Networks: The Definitive Guide.* O'Reilly, Sebastopol, CA.

Ge, Y., Kunz, T., and Lamont, L. (2003). Quality of service routing in ad-hoc networks using OLSR. In *Proceeding of the 36th Hawaii International Conference on System Science (HICSS'03)*, Washington, DC, USA. IEEE Computer Society.

Huber, R. and Jordan, N. (2005). An experimental study of next generation authentication mechanisms for wireless LAN and content service provider. In *Proceedings of Fifth IASTED International Conference on Wireless and Optical Communications 2005 (WOC 2005), Banff, Alberta, Canada.* Paper ID 474-038.

Huhtonen, A. (2004). Comparing AODV and OLSR routing protocols. In *Seminar on Internetworking*, pages 1–9, Sjökulla, Finland.

Huishan, K., Huimin, C., and Nam, K. Y. (2003). Routing protocols in ad hoc wireless networks. National University of Singapore.

Husseiki, R. (2006). Distributed authentication and key management. Master's thesis, Royal Institute of Technology, Stockholm, Sweden.

Hwang, K.-F. and Liao, I.-E. (2005). Two attacks on a user friendly remote authentication scheme with smart cards. *Operating Systems Review*, 39(2):94–96.

IEEE Std 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) specification.

Johnson, D. (2007). Evaluation of a single radio rural mesh network in South Africa. In *Proceedings 2nd IEEE/ACM International Conference on Information and Communication Technologies and Development 2007*, pages 1–9.

Johnson, D., Ntlatlapa, N., and Aichele, C. (2008). A simple pragmatic approach to mesh routing using BATMAN. In *2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries.* CSIR, Pretoria.

Katz, J. and Lindell, Y. (2007). *Modern Cryptography*. Chapman and Hall/CRC Press.

Khan, K. and Akbar, M. (2006). Authentication in multi-hop wireless mesh networks. In *World Academy of Science, Engineering and Technology*, volume 22, pages 100–105.

Kim, I.-G. and Choi, J.-Y. (2004). Formal verification of PAP and EAP-MD5 protocols in wireless networks: FDR model checking. In *Advanced Information Networking and Applications, International Conference on*, volume 2, page 264, Los Alamitos, CA. IEEE Computer Society.

Kim, I.-G. and Choi, J.-Y. (2005). Model checking of radius protocol in wireless networks. *IEICE Transactions*, 88-B(1):397–398.

Komninos, N., Vergados, D. D., and Douligeris, C. (2007). Authentication in a layered security approach for mobile ad hoc networks. *Computers and Security*, 25(5):373–380.

LaRosa, J. A. (2004). WPA: A key step forward in enterprise-class wireless LAN (WLAN) security. URL `http://www.meetinghousedata.com/landing/wp.shtml`.

Lee, I., Lee, J., Arbaugh, W., and Kim, D. (2008). Dynamic distributed authentication scheme for wireless LAN-based mesh networks. In Vazlo, T., Freire, M., and Chong, I., editors, *ICOIN 2007*, pages 649–658. Springer Verlag, Berlin.

Lee, Y.-C., Hsieh, Y.-C., and You, P.-S. (2007). A secure password authentication protocol for wireless networks. *International Journal of Computers*, 1(3):125–128.

Liao, I.-E., Lee, C. C., and Hwang, K.-F. (2005). Security enhancement for a dynamic ID based remote user authentication scheme. In *Proceedings of the national conference on Next Generation Web Services Practices (NWeSP'05)*, pages 4–4.

Liao, I.-E., Lee, C.-C., and Hwang, M.-S. (2006). A password authentication scheme over insecure networks. *J. Comput. Syst. Sci.*, 72(4):727–740.

Lloyd, B. and Simpson, W. (1996). Challenge handshake authentication protocol (CHAP). RFC 1334, obsoleted by RFC 1994, updated by RFC 2484.

Luo, H., Kong, J., Zerfos, P., Lu, S., and Zhang, L. (2004). URSA: Ubiquitous and robust access control for mobile ad-hoc networks. *IEEE/ACM Transactions on Networking*, 12:1049–1063.

Ma, X., McCrindle1, R., and Cheng, X. (2006). Verifying and fixing password authentication protocol. In *Proceedings of the Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2006)*, pages 324–329, Las Vegas, NV.

Malek, M., Fernández-Medina, E., and Hernando, J., editors (2006). *SECRYPT 2006, Proceedings of the International Conference on Security and Cryptography, Setúbal, Portugal, August 7–10, 2006, SECRYPT is part of ICETE - The International Joint Conference on e-Business and Telecommunications*. INSTICC Press.

Maple, C., Jacobs, H., and Reeve, M. (2006). Choosing the right wireless LAN security protocol for the home and business user. In *The First International Conference on Availability, Reliability and Security, 2006*, pages 1025–1032.

MoHE Afghanistan (2010). The Ministry of Higher Education. URL `http://www.mohe.gov.af`.

OLPC (2009). One laptop per child. URL `//http:olpc.af`.

Parthasarathy, M. (2005). Protocol for carrying authentication and network access (PANA) threat analysis and security requirements. Technical Report RFC 4016, IETF.

Pentland, A. S., Fletcher, R., and Hasson, A. (2004). DakNet: Rethinking Connectivity in Developing Nations. *Computer*, 37(1):78–83.

Perkins, C., Belding-Royer, E., and Das, S. (2003). Ad hoc on-demand distance vector (AODV) routing. RFC 3561.

Rigney, C., Willens, S., Rubens, A., and Simpson, W. (2000). Remote authentication dial in user service (RADIUS). RFC 2865.

Rivest, R. (1992). The md5 message-digest algorithm. RFC 1321.

Schollmeier, R., Gruber, I., and Finkenzeller, M. (2002). Routing in mobile ad-hoc and peer-to-peer networks. A comparison. In *Revised Papers from the NETWORKING 2002 Workshops on Web Engineering and Peer-to-Peer Computing*, pages 172–186, London, UK. Springer Verlag.

Simpson, W. (1996). PPP Challenge handshake authentication protocol (CHAP). RFC 1994.

Snader, J. C. (2007). *VPNs Illustrated: Tunnels, VPNs, and IPsec.* Addison Wesley.

Song, S. (2011). Bo-Kaap Village Telco. URL `http://www.villagetelco.org`.

Sun, S., Reilly, S., and Lannom, L. (2003). Handle system namespace and service definition. RFC 3651.

Thompson, N. A., Yin, Z., Luo, H., Zerfos, P., and Singh, J. P. (2007). Authentication on the edge—distributed authentication for a global open Wi-Fi network. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, MobiCom '07, pages 334–337, New York, NY, USA. ACM.

Tinc (1998). URL `http://www.tinc-vpn.org/`.

Tsai, T. J. and Chen, J. W. (2005). IEEE 802.11 MAC protocol over wireless mesh networks: Problems and perspectives. In *AINA 2005*, pages 60–63.

Tung, C. H., Chen, Y. Q., Chen, Z. M., and Tsai, S. R. (2006). Implementation of security mechanism for adhoc wireless networks based on X.509 and IEEE 802.1x. In *International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, volume 1, pages 562–563, Los Alamitos, CA, USA. IEEE Computer Society.

Urien, P. and Badra, M. (2006). Secure access modules for identity protection over the EAP-TLS—smartcard benefits for user anonymity in wireless infrastructures. In (Malek et al., 2006), pages 157–163.

Virendra, M., Upadhyaya, S., Kumar, V., and Anand, V. (2005). SAWAN: A survivable architecture for wireless LANs. In *Proceedings of the Third IEEE International Workshop on Information Assurancer*, pages 71–82, Washington, DC, USA. IEEE Computer Society.

Yeager, W. and Williams, J. (2002). Secure peer-to-peer networking: the JXTA example. *IT Professional*, 4(2):53–57.

Yoon, E.-J. and Yoo, K.-Y. (2006). Secure password authentication protocol in wireless networks. In *2006 International Conference on Next Generation Web Services Practices (NWeSP'06)*, pages 149–154.

Zhang, Y., Zheng, J., and Hu, H. (2009). *Security in Wireless Mesh Networks*. Auerbach Publications, Boca Raton, FL.

Appendix A

Certificate keys generated with openVPN

A.1 R1_CA.crt

```
-----BEGIN CERTIFICATE-----
MIICSjCCAbOgAwIBAgIJAPobeWr0Yd18MA0GCSqGSIb3DQEBBQUAMCMxETAPBgNV
BAoTCEZyZWlmdW5rMQ4wDAYDVQQDFAVSMV9DQTAeFw0xMTAxMDQxMzUyNDdaFw0y
MTAxMDExMzUyNDdaMCMxETAPBgNVBAoTCEZyZWlmdW5rMQ4wDAYDVQQDFAVSMV9D
QTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAwbVmhBHDgG+w9jIzBQoUnLcn
+I4VHB6mW1+vkCfjiAyxWoDUFXPofoDDLwq7tVCskJbtp8I/vh/shUM8XUsmMFO/
Nd0rPNh5xCEs21WNq583pxZZfJ2HbH2Lfx5bkexb0Oy5Hnkok82c77DAp+txmxz/
V6GQMbE3Fh6iallGJU0CAwEAAaOBhTCBgjAdBgNVHQ4EFgQUBr+RpYJU1jobBCZH
8Y4iCe2XFcMwUwYDVR0jBEwwSoAUBr+RpYJU1jobBCZH8Y4iCe2XFcOhJ6QlMCMx
ETAPBgNVBAoTCEZyZWlmdW5rMQ4wDAYDVQQDFAVSMV9DQYIJAPobeWr0Yd18MAwG
A1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEFBQADgYEAOZwDRoXhfcSWyOIR1/TtMTVy
y4Z1Th1XraAPFqx8y4jMMkcxvk6GPrR8G/8iCZPdLYjxyPPgb1omA8uzLBBriXyo
Xu7Aa2oIj2loBEDwIAGHMenxUBdfHTd9qRi1jxfEEXKy+5nMDtdVy8ovVAui1UX6
NOEdFvXWQ4fjm5IcoL8=
-----END CERTIFICATE-----
```

A.2 R1_SERVER.crt

```
-----BEGIN CERTIFICATE-----
MIICrjCCAhegAwIBAgIBATANBgkqhkiG9w0BAQUFADAjMREwDwYDVQQKEwhGcmVp
ZnVuazEOMAwGA1UEAxQFUjFfQ0EwHhcNMTEwMTA0MTM1NDQ2WhcNMjEwMTAxMTM1
NDQ2WjAnMREwDwYDVQQKEwhGcmVpZnVuazESMBAGA1UEAxQJUjFfU0VSVkVSMIGf
MA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDDRdNC3vcVUdCBhiggi3O4ZzOnC416
F/h2kQYnLGkyV5C6yRd/7LAnlxkA2YyZbSaT9z2q4XiVG+Nuh9sfDNIhCI6xqvhv
```

```
9XEaPjTGGzyrQ8t1OEyR3ZLfK7q0pdSQFABo64OIr83n8Xrlvq0jTNyQqrBuRd5x
xqgU1aZOEVPntwIDAQABo4HtMIHqMAkGA1UdEwQCMAAwEQYJYIZIAYb4QgEBBAQD
AgZAMDQGCWCGSAGG+EIBDQQnFiVFYXN5LVJTQSBHZW5lcmF0ZWQgU2VydmVyIENl
cnRpZmljYXRlMB0GA1UdDgQWBBT8PXtz93Oz5DejQVVwn1EJPK0DQjBTBgNVHSME
TDBKgBQGv5GlglTW0hsEJkfxjiIJ7ZcVw6EnpCUwIzERMA8GA1UEChMIRnJlaWZ1
bmsxDjAMBgNVBAMUBVIxX0NBggkA+ht5avRh3XwwEwYDVR0lBAwwCgYIKwYBBQUH
AwEwCwYDVR0PBAQDAgWgMA0GCSqGSIb3DQEBBQUAA4GBAHzu36OzU5jmpTQKLB67
+8BE+ehTtJNOmK6N8x8qf3Dybh8YClXPPRO1ItND08XYBPcewPMygKP9mISxzZhl
Dge325vP50/mRxP0WCCYEMwIYGCk2ELm1XAfCFRKE9YyfmYQf3cl2F+x0ebqvTuF
qzNSqFOApRO11H1xWRZ4Xhtv
-----END CERTIFICATE-----
```

A.3 R1_SERVER.key

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQDDRdNC3vcVUdCBhiggi3O4ZzOnC416F/h2kQYnLGkyV5C6yRd/
7LAnlxkA2YyZbSaT9z2q4XiVG+Nuh9sfDNIhCI6xqvhv9XEaPjTGGzyrQ8t1OEyR
3ZLfK7q0pdSQFABo64OIr83n8Xrlvq0jTNyQqrBuRd5xxqgU1aZOEVPntwIDAQAB
AoGBAJK9BeRaaJI8yNKMXdsW7/7Cjq4YoFf/a1at5FHLrcGGy9uRAXfABR+7C673
zA//Vd4bDNw8UsC6FE4bCS5sE/ot9SptSIg8TkXzriEjVOVyDN1FLU1xou1ip+vu
+N+bj8QqERu+trxsW+Fbw1u66UH2L4Ux4v2nmD3mZ0jJBokBAkEA7/UPE65aceBJ
nYHNUouCungs6T0oIjtdi2OWfImcuX41b8n0Pn+I5w3FgKKuV6MaqSBPx/Wg+mTK
j0su4ydGswJBANBT+j2IWbBcZ9J/TDJe8JwG2hjHH7cZIh374v2vrYzUiq2u511q
V4oM6LpGDSttCw0dszGurnn556aFUH/EvO0CQQCUwHLnPXd+Do80IApbeHbDB8q3
kBz4E5sq1MKuU1SfvPh9Y1GRCNiie2wLEU5Ir6jQXWJ3z/JKJv9VmlnHoUp3AkAH
aoXk/N6cfZ2gi7GNtX2BQGkxKp2Bah6hzJ63AzL9aW+KYUiRNcDqBC2gu++qFRao
n98KLLq/UtshPWuxMLK9AkBlit/YP3M1UslHajd+ccK8L4n3ga7orScmUBCpLSEW
0jfVkuDjorbpQfScc2H/atF5UrR8dJAxWwa5KHxdNfM0
-----END RSA PRIVATE KEY-----
```

Appendix B

Configuration and certificate keys for TincVPN

B.1 Setting up Freifunk

Following are the step for server:

```
#ipkg update
#ipkg install tinc
#mkdir -p /etc/tinc/R3/hosts
#vim /etc/tinc/tinc.conf
# Symbolischer Name f?r diese Verbindung. (nur alphanumerische Werte und der _ s
Name = R3

# Mit welchen anderen Tinc-Daemons soll sich bei Programmstart verbunden werden.
# Entsprechende Host-Konfiguation Dateien in "hosts/" m?ssen vorhanden sein.
ConnectTo = R1
ConnectTo = R2

Device = /dev/net/tun

# Name des Tunnelinterfaces, der vergeben werden soll z.B. tun0
Interface = tinc0
AddressFamily = ipv4

Hostnames = yes
Mode = switch
PrivateKeyFile = /etc/tinc/R3/rsa_key.priv
PingTimeout = 30
```

We write the script of tinc up and down:

Vim/etc/tinc/R3/tinc-Down

```
======================================================
iptables -D INPUT -i $WANDEV -p udp --dport $TINCPRT -j ACCEPT
iptables -D INPUT -i $WANDEV -p tcp --dport $TINCPRT -j ACCEPT

# Input / Output auf dem Tinc-Interface verbieten
iptables -D INPUT  -d $WIFIADR -i $TINCDEV -j ACCEPT
iptables -D INPUT  -d $TINCADR/$TINCPRE -j ACCEPT
iptables -D OUTPUT -s $WIFIADR -o $TINCDEV -j ACCEPT
iptables -D OUTPUT -s $TINCADR/$TINCPRE -j ACCEPT

# Weiterleitungen zwischen Wifi- und Tinc-Interface verbieten
iptables -D FORWARD -i $WIFIDEV -o $TINCDEV -j ACCEPT
iptables -D FORWARD -i $TINCDEV -o $WIFIDEV -j ACCEPT

# Weiterleitungen zwischen LAN- und Tinc-Interface verbieten
iptables -D FORWARD -i $LANDEV -o $TINCDEV -m state --state NEW,RELATED,ESTABLIS
iptables -D FORWARD -i $TINCDEV -o $LANDEV -m state --state RELATED,ESTABLISHED

# LAN-Nat-Regel l?schen
iptables -t nat -D POSTROUTING -o $TINCDEV -s $LANNET/$LANPRE -j MASQUERADE

# Interface dekonfigurieren
ip addr del $TINCADR/$TINCPRE brd $TINCBRC dev $TINCDEV

===========================================================
```

Vim/etc/tinc/R3/tinc-up

```
#/bin/sh
#this file is for tinc startup

#  Start of configuration --
```

```
TINCPRT=655
TINCADR=10.1.1.1       # <-- IP is the network address of router")
TINCBRC=10.1.1.255
TINCPRE=24
TINCDEV=$INTERFACE
#TINCDEVhost=10.1.1.3/24
# --end of configuration --

# Network load parameters
eval $(/usr/bin/netparam)

# Tinc Interface configuration
ip addr add $TINCADR/$TINCPRE brd $TINCBRC dev $TINCDEV

# Input to tinc permit from the WAN port
iptables -I INPUT -i $WANDEV -p udp --dport $TINCPRT -j ACCEPT
iptables -I INPUT -i $WANDEV -p tcp --dport $TINCPRT -j ACCEPT

# input/output interface on tinc
iptables -I INPUT  -d $WIFIADR -i $TINCDEV -j ACCEPT
iptables -I INPUT  -d $TINCADR/$TINCPRE -j ACCEPT
iptables -I OUTPUT -s $WIFIADR -o $TINCDEV -j ACCEPT
iptables -I OUTPUT -s $TINCADR/$TINCPRE -j ACCEPT

# redirect between Wi-Fi and tinc interface allow
iptables -I FORWARD -i $WIFIDEV -o $TINCDEV -j ACCEPT
iptables -I FORWARD -i $TINCDEV -o $WIFIDEV -j ACCEPT

# forwarding between LAN and tinc interface allow
iptables -I FORWARD -i $LANDEV -o $TINCDEV -m state --state NEW,RELATED,ESTABLISHE
iptables -I FORWARD -i $TINCDEV -o $LANDEV -m state --state RELATED,ESTABLISHED -j

# LAN netting direction of tinc
iptables -t nat -A POSTROUTING -o $TINCDEV -s $LANNET/$LANPRE -j MASQUERADE
```

```
# active  interface
ip link set dev $TINCDEV up

for shutdown of tinc VPN:
root@R1:/etc/tinc/R1# vim tinc-down
#!/bin/sh
# This file closes down the tunnel device und removes corresponding firewall rules.

# --Start of configuration--
TINCPRT=655
TINCADR=10.1.1.0       # <-- ip should customize!
TINCBRC=10.1.1.255
TINCPRE=24
TINCDEV=$INTERFACE
# --End of configuration --

# Network load parameters
eval $(/usr/bin/netparam)

# Tinc interface configuration
ip link set $TINCDEV down

# Input to the tinc band from the WAN port
iptables -D INPUT -i $WANDEV -p udp --dport $TINCPRT -j ACCEPT
iptables -D INPUT -i $WANDEV -p tcp --dport $TINCPRT -j ACCEPT

# Input / Output interface ban in tinc
iptables -D INPUT  -d $WIFIADR -i $TINCDEV -j ACCEPT
iptables -D INPUT  -d $TINCADR/$TINCPRE -j ACCEPT
iptables -D OUTPUT -s $WIFIADR -o $TINCDEV -j ACCEPT
iptables -D OUTPUT -s $TINCADR/$TINCPRE -j ACCEPT

# forwarding between Wifi interface and tinc interface ban
```

```
iptables -D FORWARD -i $WIFIDEV -o $TINCDEV -j ACCEPT
iptables -D FORWARD -i $TINCDEV -o $WIFIDEV -j ACCEPT

# forwarding between LAN and tinc interface ban
iptables -D FORWARD -i $LANDEV -o $TINCDEV -m state --state NEW,RELATED,ESTABLISHE
iptables -D FORWARD -i $TINCDEV -o $LANDEV -m state --state RELATED,ESTABLISHED -j

# LAN-Nat-role delete
iptables -t nat -D POSTROUTING -o $TINCDEV -s $LANNET/$LANPRE -j MASQUERADE

# Interface unconfigure
ip addr del $TINCADR/$TINCPRE brd $TINCBRC dev $TINCDEV
```

We created a private key and a public key for the server and clients:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA3z3fXVFp9G3tJcakcmUB7yuogH7rtHSTPxnUbx3HWAXLeM9l
BXJ3ztKH0Ht06SlIXmoAwPEU66CARajqLeFrrK8ZV+JUbXH9lnP2q2wlgdhnBDhP
m2q9mTe8L3WZ+yAdcyDil8PbbKa3UQ6micvA6EL3mOEP0+oJ+X0SaocGInk5k4Pl
49Y+OxxuaGvJca+TaQs6DRyr+2+WqLwaNhXiNb5nf6jtgXXDFQmEwxPFfxhN9NLI
PEizzgQ+c0UwmBAIVjs5YztEaB1zXS26c/N4MWX8BkIACj3jrwiwVWa7Z2+BaDAb
F8JFKCflzbmA4Z9tpIIMJLlumQ8k9JXbcuHLtQIDAQABAoIBAQDQWWGXKX2218Vl
5T/jV+v3OQrZahs33zjxwnyme+dpencNOJqy+8NfbtO8uwxA8/uLymkLqbT5uGEf
Imf1DBa/IuiHQj/DqYigG3iTQqq6xapXoOqDRZrQXdkvfhWI788sfPud4vhuLiaW
6xJ5NoXC/4Uv1JU+FYbbp2RBseUC09xBKJrdsQGpzCbopmu0rI+9ynlbEfItlEQ/
NMXVp8OyexWIylNXaWKkUkNjr019xIWm62xiQHUAovn8FcV40kLv+XcII3e98a+N
3UDzaPnf2EvMkOYlLJcMEwDpjNxEdGfP5keQW1Q6AIMFKjt4dLVwMusiZ4PzAQYw
jFM4rR0hAoGBAPnvTmNP8hl6yVcYR1DLqEQfxJI4DSRyjUtX/mS8J+izxjIGBtsH
Ztt0bRI3C6sB9MfJPHdCFlUQyXgyECCGdqb7D5PmB2qTaaKeEmSPQopURmGtzroS
2lvEup78nJXqDQfRteswe5bmifTlQ1SJJy6417oDthXcO71RgBzyqquZAoGBAOSo
vPulxH8LMZvtpFI6+Sg6As0gR4dvrXiTB4KHoU0CdDJ2UI12MSqPwLLLCShPCEeS
Ulo+wzz+/8H39rtA2WfGioM//j+9TM+FpoFJ+VO3WosPngHoQnpW1y1mNX+gD+RG
zKiS/9PqSAPCrL1kVaZTULSwiT2DekFHlwYUXVJ9AoGAVSyGbDXMbX54qu0tOg2N
dFnXJXVQlUEsgaalHTM57oWFX3rKT05AcT1GBlUX9tkd0A/2OTYeYjLsVbfJ4yuv
t8wLTP2xPVKrRy3TWs2vgXCi45rVvne6TlfGuXOg8T3n3uhKtLG7DHvYO8r9nQt4
```

```
fOWlLtEg9mYtPaFOpY7VNtECgYBOYlG2alL7whiVfVdxIOo8kfZX3xLmKLBsvuo6
Ish2n7q5ebmPZM7h6jmCkPtop/8MmepFKXLAPCHsmlYoRsjAPF1LNOqxavEjyosT
kwo3WoldVlMyEC6Jxa8pzvnT8EjCVkreFtZADPX25AtXOUfrJnBqIIJLZgDI1AO/
f8el+QKBgQDvr+bpoSh4Kf/sQCzm6yj4gGiEOPDnBhU2Z4fGMtNGKwRnVpaVt6kD
iYXV9OtsQSsWpkd5ZNGHQCLbKiWDZarBHVnmMOpjiTgF47KCMjg2OW+TGkNJH7z3
gKfyhyPHP36m4WPZazUQwepXNwr5KvBOVdiaVhxybeUleLv305v5jA==
-----END RSA PRIVATE KEY-----

compression=9
subnet=10.1.1.0/24
address=10.1.1.1

-----BEGIN RSA PUBLIC KEY-----
MIIBCgKCAQEA3z3fXVFp9G3tJcakcmUB7yuogH7rtHSTPxnUbx3HWAXLeM9lBXJ3
ztKHOHtO6SlIXmoAwPEU66CARajqLeFrrK8ZV+JUbXH9lnP2q2wlgdhnBDhPm2q9
mTe8L3WZ+yAdcyDil8PbbKa3UQ6micvA6EL3mOEPO+oJ+XOSaocGInk5k4Pl49Y+
OxxuaGvJca+TaQs6DRyr+2+WqLwaNhXiNb5nf6jtgXXDFQmEwxPFfxhN9NLIPEiz
zgQ+cOUwmBAIVjs5YztEaB1zXS26c/N4MWX8BkIACj3jrwiwVWa7Z2+BaDAbF8JF
KCflzbmA4Z9tpIIMJLlumQ8k9JXbcuHLtQIDAQAB
-----END RSA PUBLIC KEY-----
```