Deanship of Graduate Studies
Al-Quds University

Ticket Authentication Wireless Mesh Networks Protocol

Anas Maher Ibrahim Melhem

M.Sc. Thesis

Jerusalem - Palestine

Year (1433) / Year (2012)

Ticket Authentication Wireless Mesh Networks Protocol

Prepared By:

Anas Maher Ibrahim Melhem

B.Sc.: Palestine Technical University- Palestine

Supervisor: Dr. Rushdi Hamamreh

A thesis submitted in partial fulfillment of requirements for the degree of Master of Electronics and Computer Engineering - faculty of Engineering
Al - Quds University

Year (1433) / Year (2012)

Al Quds University

Deanship of Graduate Studies

Electronics and Computer Engineering Program

Thesis Approval

Ticket Authentication Wireless Mesh Networks Protocol

Prepared By: Anas Maher Ibrahim Melhem Melhem

Registration No: 20611001

Supervisor: Dr. Rushdi Hamamreh

Master thesis submitted and accepted, Date:

The names and signatures of the examining committee members are as follows:

1-Head of Committee: Dr. Rushdi Hamamreh Signature ………

2-Internal Examiner: Dr. Raid Saghal Signature ……….…

3-External Examiner: Dr. Aiman Abu Samra Signature …………….

Jerusalem - Palestine

Year (1433) / Year (2012)

Dedication

To my beloved Parents who gave me endless love and support.

Daddy, you always told me to “reach for the stars.” I think I got my first one. Thanks for inspiring my love for knowledge

Mom, you have given me so much, thanks for your faith in me, and for teaching me that I should never surrender.

Brothers, for their encouragement;

My wife Nagham, for being with me during the most difficult moments.
A very special thanks for your practical and emotional support.

My precious son, Maher who gave me the spirit and strength to finish this dissertation

Thank you all, I love you!

Signature:……………………….

Anas Maher Ibrahim Melhem

Date:……………………………

Declaration:

I Certify that this thesis submitted for the degree of master, is the result of my own research, except where otherwise acknowledged, and that this thesis (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed: ……………………….

Anas Maher Ibrahim Melhem

Date: ……………………….

Acknowledgements

I would like to express my greatest gratitude to the following people that helped me to turn a dream into reality, be it through scientific as well as moral support. Without their guidance and help the work presented in this thesis would not have been possible.

The first person that comes to mind is my adviser Dr. Rushdi Hamamreh, who believed in me from the very beginning and allowed me to develop professionally as well as individually. With his patience, guidance, deep vision, support, constant encouragement, this thesis has been done what it is today. I am very glad and it has been an honor to work with him over the past year.

Thanks go out to all my teachers at AL-Quds University who do their best to provide me with their help, encouragement and experience. Also, I would like to express my gratitude to all those who gave me the possibility to complete higher studies especially my colleagues at Palestine Technical University.

Abstract

Wireless Mesh Networks (**WMNs**) are multi-hop wireless networks consists of Mesh Clients (**MCs**) and Mesh Routers/Access Points (**MAPs**), where **MCs** are mobile, battery constrained, and connecting among themselves over possibly multi-hop paths with or without the involvement of **MAPs**. The **MAPs** are typically stationary and energy rich devices and form the backbone of **WMNs**. Shared medium access, open peer-to-peer network topology, severe resource constraints, and highly dynamic topology, are reasons to make WMNs vulnerable to various kinds of security threats.

Hybrid Wireless Mesh Protocol (**HWMP**) the standard routing protocol for **WMNs** defines secure links between MPs, but it does not provide end-to-end security. It's also doesn't specify routing security or forwarding functionality. Routing information elements are transferred in plain text and are prone to various types of routing attacks like flooding, route re-direction, spoofing.

In this thesis, we developed a secure protocol (**TAWMP**) that provides end-to-end authentication to **WMN** by dividing authentication process into two phases. In first phase (**MAP Phase**) we provide protection for **MAPs** using asymmetric authentication to generate a secret session key in order to provide node-to-node protection; while in second phase (**MC Phase**) we depend on symmetric authentication to generate **Ticket** contains necessary keys for transferring data between communicated MCs using (EAP-TTLS) which is more suitable for wireless communications in order to provide end-to-end protection. **TAWMP** depends on adding authentication server to the WMN; which will increase end-to-end delay in the network. **TAWMP** provides link protections between hops in addition to end-to-end authentication. **TAWMP** has been tested using Network Simulator NS-2 version 2.34 in order to evaluate our proposed protocol and compare it to **HWMP**.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Meaning
WMN	Wireless Mesh Network
MAP	Mesh Access Point
MC	Mesh Client
AS	Authentication Server
DH	Diffie-Hellman
TAWMP	Ticket Authentication Wireless Mesh Network Protocol
HWMP	Hybrid Wireless Mesh Network Protocol
AODV	Ad hoc On-demand Distance Vector
DSDV	Distance Sequenced Distance Vector
DSR	Dynamic Source Routing
LAN	Local Area Network
MAC	Media Access Control
MANET	Mobile Ad hoc Network
MPR	Multipoint Relaying
OLSR	Optimized Link state Routing
PDR	Packet Delivery Rate
QoS	Quality of Service
RERR	Route Error
RREP	Route Reply
RREQ	Route Request
SNR	Signal-to-Noise-Ratio

Chapter One

Introduction

Table of Contents

1.1 Introduction

With rapid growth of internet, WMNs is emerging as a practical solution for providing community broadband connection. These networks are dynamically self-organized and self-configured with the existing nodes in the network by automatically establishing and maintain mesh connectivity among the nodes [1].

Typical WMNs consist of MAPs and MCs [2], MAPs are static and power-rich devices it form WMNs backbone through multi-hop communications, MAPs cover same area as conventional routers do with much less transmission power. On the other hand MCs are battery dependent devices that have necessary mesh functions for behaving like mesh routers and for transmission of data in the network [2-4].

The architecture for WMN can be classified into three main categories [1, 5]:

Infrastructure/Backbone WMN: This type of WMNs can be build using various type of radio technology in addition to IEEE 802.11 technologies. With gateway functionality, this type of WMNs can be connected with internet and with various types of wire/wireless networks [6, 7].

Client WMNs: MCs form network through peer-to-peer connectivity using on type of radios on client devices. In this type of WMNs, client nodes perform routing and configuration functionality in addition to end user applications to customer which of course increases the requirements on end-user devices. In this type of architecture a packet destined to a node in the network hops through multiple nodes to reach the destination [3,8].

Hybrid WMNS: this type of WMNs is combination of **infrastructure** and **client** networks, were MCs can access the network through MAPs as directly connect with nearby MCs [5]. Since **Hybrid** architecture combines both type of devices; energy-rich MAPs and battery dependent MCs, Hybrid Wireless Mesh Protocol (HWMP) is the default protocol for IEEE 802.11s WLAN mesh networking, and combine reactive and proactive components, which uses MAC address to identify destination and to construct routes. HWMP is based on AODV routing protocol [9, 10]. In reactive component, when a source wants to send data to a destination for which it does not have a path yet, it initiates a *path discovery*, in which the source node broadcasts a path request message (PREQ). The destination responds with a path reply message (PREP), which is sent to the source by unicast. The accumulated path metric, stored in the PREQ and the PREP respectively, determines the best path according to the used path selection metric [9, 11]. The HWMP sequence number in path selection messages prevents loops that would have been caused by outdated path selection information. On the other hand proactive mode spread the broadcasts periodically in the mesh. This generates a root selection in tree structure using PREQ or RANN [10, 12].

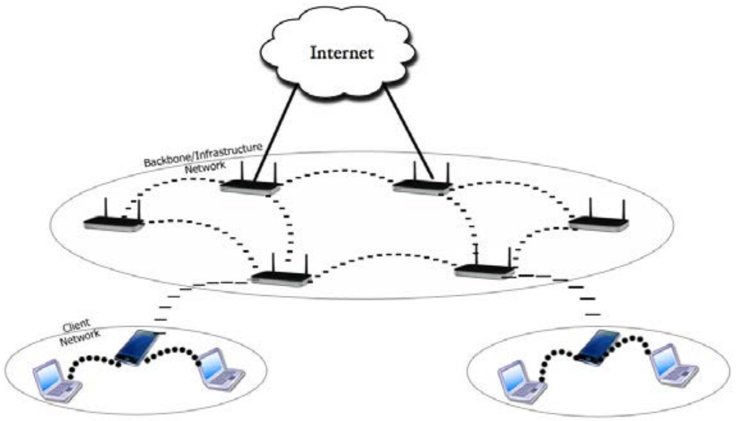

Figure1.1. Hybrid wireless Network.

Some features such as the multi-hop connection, dynamic mesh topological changes and the attack prone wireless medium access of WMNs are making it vulnerable to different security threats both active and passive attacks [13, 14]. Were passive attacks would compromise confidentiality; active attacks would violate availability, integrity, authentication, and non-repudiation. Wireless networks have high probability for physical threat, unlike wired networks were routers are protected and hard to reach, wireless routers and access points may installed on roofs of building or street lamps which make physical protection harder and cause like stealing private key for authentication stored in the router or replacing the router with malicious one [15].

1.2 Problem Statement

Hybrid Wireless Mesh Protocol (HWMP) is the standard security protocol for WMN, HWMP defines security for multi-hop network using node-to-node authentication which is fast and suitable for wireless scalable networks.

Dependency in node-to-node authentication in HWMP leaves the network vulnerable to various types of routing attacks such as flooding, route disruption and diversion; in addition HWMP doesn’t protect mutable fields in the routing information elements.

1.3 WMNs Security Challenges

Detecting malicious nodes since router/access points in wireless networks are highly prone to physical attacks and because of dynamic mesh topological changes, mesh network’s nodes have highly possibility to passive or active attack, which can seriously disrupt routing process, this make it critical for WMNs to identify compromised nodes[15].

Protecting multi-hop mechanism different type of threats attacks routing mechanism and the functionality of WMN by changing the routing message or the state of one or more of the nodes[16].

1.4 WMNs Security Requirements

To ensure security of WMN, the following requirements must be achieved:

Confidentiality which means that certain data is only accessible by authorized users. Confidentiality cannot be achieved without authenticity i.e. without knowing you are talking to right person. After achieving authenticity, confidentiality becomes matter of encrypting transferring data[17].

Authenticity authentication is vital at any type of communication to make sure the connection is done with true destination, without authentication impersonators can gain access to confidential data [15].

Congestion since several nodes in WMN are working as repeater. Bandwidth is dependent on the number of nodes surrounding repeater node in addition to the throughput of this repeater node. Some attacks aims to increase the message path to increase the latency or to change the path to specific node in order to cause congestion and reduce bandwidth share [18].

Integrity it's crucial for WMN to guarantee message integrity since a transferred message in wireless medium can be subjected to altering. [17] has defined two ways in which integrity can be done in as

Malicious altering, which can be done by adversary.

Accidental altering, such as transmission error.

Availability one of the most dangerous attacks wireless network can be prone to is Denial of Service (DoS) attacks, in which all nodes in the network can be attack target, therefore some selfish nodes make some of the network services unavailable [15]. This type if attacks can be done at any layer of the network. It can be done by launching jamming signal to interfere with communication on physical channels. Or an adversary can interrupt the routing protocols and disconnect the network. Availability ensures the survivability of network services despite attacks.

1.5 Security Attacks of WMNs

Wireless mesh networks proposed to several kinds of attacks which can be categorized into [15, 16, 18, 19]:

1. **Passive attacks** in this type of attacks, attacker will not involve any disruption servers, it will only listen and analyze the network traffic to capture sensitive information.

   1.1. **Eavesdropping** the eavesdropper can get topological information or capture and listen to on-going traffic of communication channels, eavesdropper can get benefit of sniffer which is application or device which is able to read and capture network packets. This attack is less severe for the wireless network.

1.2. **Network analysis** in this type of attacks, network traffic is intercepted and analyzed for many reasons such as

- To locate the source and destination nodes.
- People in communication.
- Military intelligence.

WMN is vulnerable to such attack since its multi-hop nature compared with IEEE 802.11 which is single-hop.

2. **Active attacks** in this type of attacks, the attacker harms the network intentionally be altering data and flooding network or cut of some nodes from their network so they cannot use its services effectively anymore.

2.1. **Denial of Service (DoS)** this type of attacks normally can be done by flooding centralized resource in the network, so that it no longer operate correctly or stop completely. It also can be done on MAC layer which called *Selfish attack*, in which selfish node reduces the resource of wireless channel, thereby affect network performance or interrupt network service.

2.2. Impersonate attack in this type of attacks, compromised node manage to join the network, so it will be able to send false routing information or succeed to access the management system of the network.

1.6 Research Objective

1. Develop an efficient and reliable solution for node-to-node vulnerability in HWMP authentication.

2. Evaluate the effectiveness of the proposed protocol for addressing security threats facing node-to-node protection.

3. Demonstrate the effectiveness of performance of the proposed protocol in WMN environment.

1.7 Thesis Contribution

To achieve the main goal of this thesis we developed a new routing protocol Securing Ticket Authentication Wireless Mesh Networks Protocol (TAWMP) that provide end-to-end protection for transferred messages between authenticated devices in the network in addition to provide node-to-node protection for between MAPs in the WMN.

The contributions of this thesis can be summarized as follows:

1. We developed a new routing protocol that provides both an end-to-end protection for transferred data between authenticated devices and also provides a node-to-node protection for between MAPs in the WMN.

2. We specified a threat model by determining several attacks that WMNs are prone to; these attacks are passive and active ones such as flooding attack, route disruption attack, route redirection attack, and routing loops attack. We defined these attacks in order to limit our scope to specific issues so we can develop a suitable authenticating protocol that can address these issues

3. We adopted hybrid approach in designing TAWMP by dividing the authenticating protocol to two phases, where the first phase called MAP phase which depends on Diffie-Hellman authentication based protocol to provide authentication for new connected MAPs in the WMN, and the second phase, MC phase depends on server-side certificate authentication protocol such as EAP-TTLS.

4. We used an authentication server in order to provide authentication for authorized MAPs and MCs in the WMN.

1.8 Thesis Organization

The remaining chapters of this thesis are organized as follows:

Chapter 2 presents an overview of WMN. This chapter describes the concepts used in WMNs, architecture and types of WMNs, and routing protocols used in WMNs such as On-Demand routing protocols and proactive routing protocols of WMN,

Chapter 3 presents an overview of cryptography. This chapter explains the basic cryptography concepts which are essential for the clarity of this thesis. These concepts include symmetric cryptography, asymmetric cryptography, hash functions, authentication and digital signatures.

Chapter 4 presents the design and analysis of TAWMP protocol. We present the protocol basic approach, protocol properties, and authentication process in details and how TWMP protocol protects WMN from different security threats.

Chapter 5 presents protocol simulation and results. We explain how to generate and simulate our network by ns-2 simulator and discuss the parameters used to evaluate our protocol such as: packet delivery ratio, network throughput, and end to end delay.

Chapter Two

Wireless Mesh Networks

Table of Contents

2.1 Overview of Wireless Mesh Networks

Demand on wireless networks is expanding and the desired for ubiquitous wireless connectivity is deriving the demand for coverage extension of WLANs. IEEE 802.11 relies on wired backbone to interconnect to non-208.11 networks and on bridging functionalities, this dependency limits deployment of wireless infrastructure and its coverage [20]. This dependency cause several issues since its costly and inflexible as WLAN coverage cannot go beyond backhaul deployment, in addition that centralized structures work inefficiently with new applications, such as wireless gaming, require peer-to-peer connectivity. Also, a fixed topology prohibits stations from choosing a better path for communication [21].

Conventional WLANs relay on IP layer for multi-hop communication, this dependency has a number of disadvantages since wireless links are less reliable than wired ones, and IP layer cannot perceive the radio environment [22]. To address these issues the Mobile Ad hoc Networks (MANET), the group of the Internet Engineering Task Force's (IETF's) [23] developed special ad hoc routing protocols that constantly broadcast routing messages to acquire reasonable metrics. MAC-based multi-hop address different IP-layer issues so an integrated mesh networking solution developed in IEEE 802.11Task Group S which deals with mesh support [20].

2.2 Architecture and Network Design

In the following section we describe both IEEE 802.11 and 802.11s architectures.

2.2.1 IEEE 802.11 Architecture

The most basic entity in 802.11 is a station (**STA**), which is any device satisfy requirement of an IEEE 802.11 standard compliant Medium Access Control (**MAC**) and

Physical Layer (**PHY**). A basic service set (**BSS**) can be formed using two stations, where the station provides an integration service to other stations referred to as an access point (**AP**). In IEEE 802.11 the term link is defined from the MAC layer's point of view.

As the BSS forms a wireless single-hop network where all participating stations send and receive frames via the AP, the AP operates as relay between them. Deferent AP can get benefit from Distribution System Service (DSS) to interconnect their BSS so that stations can roam within their BSS.

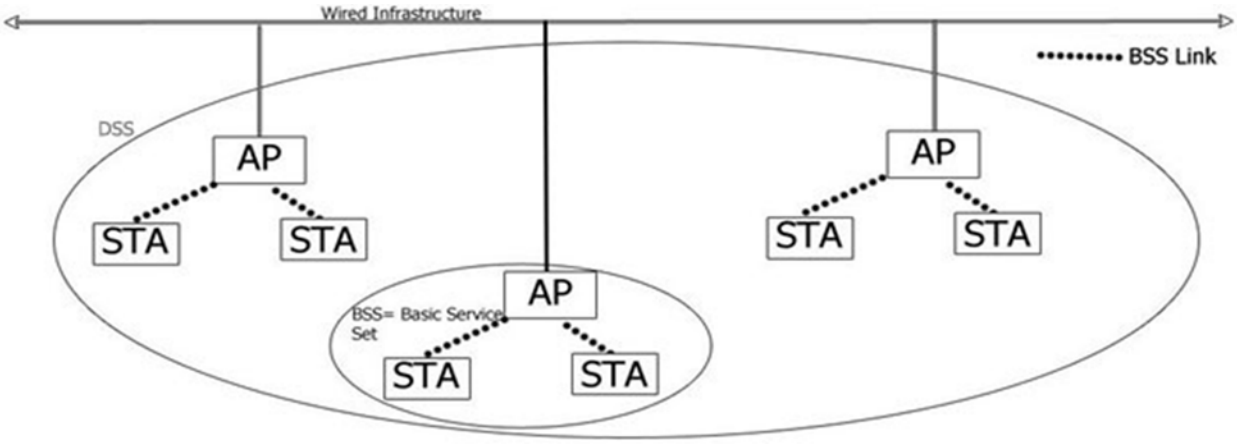

Figure 2.1. IEEE 802.11 Architecture

2.2.2 IEEE 802.11s Architecture

The basic entity in IEEE 802.11s is the Mesh Point (**MP**), which is every IEEE 802.11-based entity (AP or STA) that fully or partially supports mesh functionalities like neighbor discovery, channel selection and association forming with its neighboring MPs. Number of MPs interconnected to each other, enabling automatic topology learning and dynamic path configuration forms an Extended Service Set (**ESS**). Mesh connectivity is established by applying multi-hop mesh techniques to specify a wireless distribution system (**WDS**) building a wireless infrastructure among nodes.

MAPs are specific MPs but can act as APs as well. Mesh Portal Point (**MPP**) is another type of MPs that has the ability of acting as a bridge or gateway between the mesh cloud and external network.

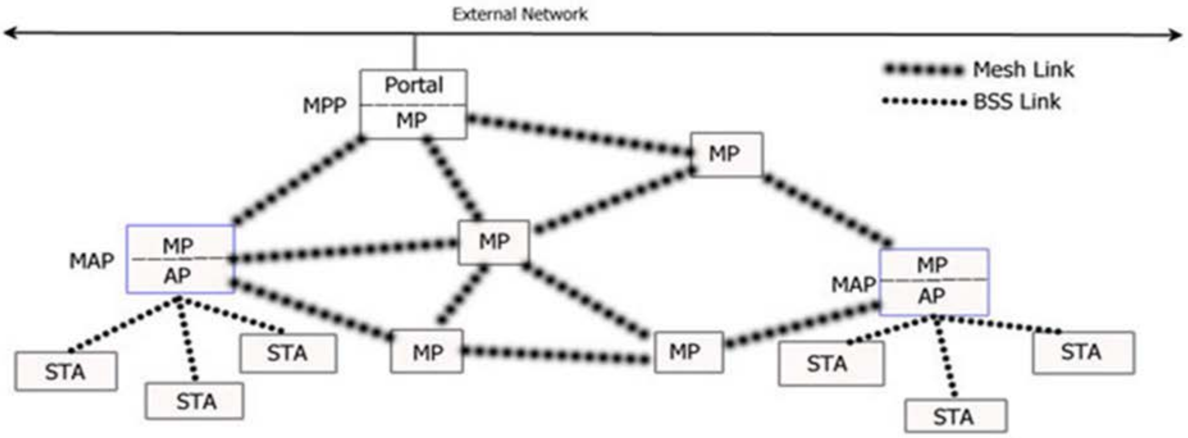

Figure 2.2. IEEE 802.11s Architecture

2.3 Routing Protocols

Routing protocols for WMNs are mostly based on protocols designed for mobile ad hoc networks [24]. These can be classified in the three categories, proactive, reactive, and hybrid routing protocols [25]

2.3.1 Proactive Routing Protocols

Proactive routing protocols maintain a table for each node representing the entire network topology which is regularly updated in order to maintain the freshness of routing information [26]. At any given time, any node knows how to reach another node of the network. This approach minimizes the route discovery delay at the cost of exchanging data periodically, which consumes network bandwidth. Proactive protocols are preferred for small networks because of low routing, table lookups. Destination Sequenced Distance Vector

(DSDV) [27], Optimized Link State Routing (OLSR) [28], Topology dissemination Based on Reverse-Path Forwarding (TBRPF) [28] are some of proactive routing protocols. [9] and [10] approve that OLSR has minimum delay over DSDV and less power consumption.

Optimized Link State Routing (OLSR)

Optimized Link State Routing (OLSR) protocol is a proactive routing protocol where the routes are always immediately available when needed [29]. OLSR is an optimization version of a pure link state protocol in which the topological changes cause the flooding of the topological information to all available hosts in the network [30]. OLSR may optimize the reactivity to topological changes by reducing the maximum time interval for periodic control message transmission. Furthermore, as OLSR continuously maintains routes to all destinations in the network, the protocol is beneficial for traffic patterns where a large subset of nodes are communicating with another large subset of nodes, and where the source-destination pairs are changing over time.

OLSR protocol is well suited for the application which does not allow the long delays in the transmission of the data packets [31]. The best working environment for OLSR protocol is a dense network, where the most communication is concentrated between a large numbers of nodes[32]. OLSR reduce the control overhead forcing the Multi Point Relays (MPR) to propagate the updates of the link state, also the efficiency is gained compared to classical link state protocol when the selected MPR set is as small as possible [33]. But the drawback of this is that it must maintain the routing table for all the possible routes, so there is no difference in small networks, but when the number of the mobile host increases, then the overhead from the control messages is also increasing. This constrains the scalability of the OLSR protocol [24], the OLSR protocol work most efficiently in the dense networks [21].

2.3.2 Reactive Routing Protocols

In reactive routing protocols [34], nodes are not aware of the network topology. A routing is constructed on-demand. They find routes by flooding network with route requests. This leads to higher latency due to the fact that the route has to be discovered, however it minimizes control traffic overhead. Usually, reactive routing protocols are better suited in networks with low node density [21]. Proactive protocols are more efficient in dense networks with burst traffic due to the continuous exchange of topology information, reducing route discovery delay. Reactive protocols are preferred for high mobility networks [35]. Dynamic Source Routing (DSR) [36], Ad hoc On-Demand Vector (AODV) [37] and some other extensions derived from AODV are reactive routing protocols

Ad-Hoc on Demand Distance Vector (AODV) Protocol

AODV is one of the most popular on-demand routing protocol and it’s the base for HWMP, routes to the destination are only discovered when required thus avoiding memory overhead and less power [38].

A node using AODV does not need to discover and maintain a route to another node until the two nodes need to communicate with each other [39]. The routing messages do not contain information about the whole route path, but only about the source and destination [40]. Therefore, routing messages are not increasing in size. All these features enable AODV to be a suitable routing protocol for MANET.

AODV uses a destination sequence number, which is generated, by the destination itself for each route entry [37]. The destination sequence number ensures loop freedom and if two similar routes to a destination exist, then the node chooses the one with the highest sequence number. AODV uses Route Request (RREQ), Route Reply (RREP) and Route Error (RERR) messages for route discovery and maintenance.

The routing operations of AODV generally consist of two phases: route discovery and route maintenance. In figure 2.3, Route discovery is performed through broadcasting RREQ messages.

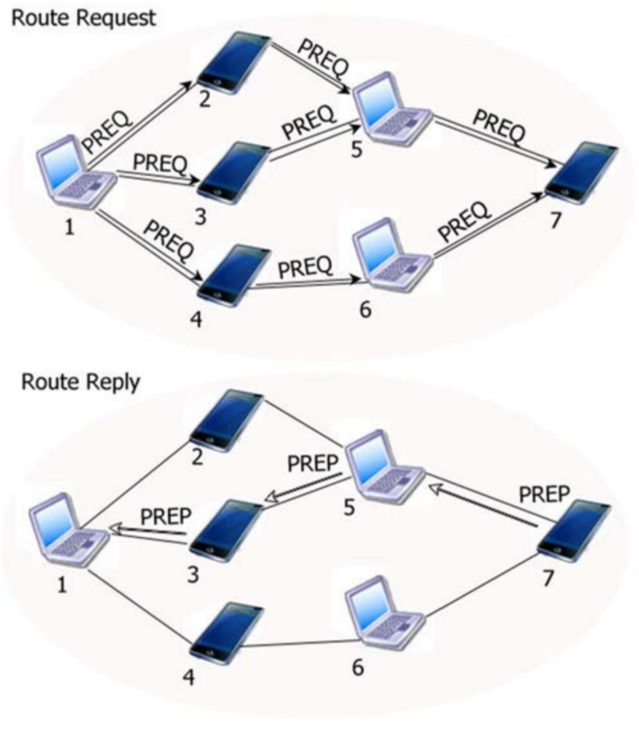

Figure 2.3. Route Discovery in AODV Protocol

When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet across the network. RREQ carries Source ID, Destination ID, Source Sequence Number, Destination Sequence Number and a Broadcast

ID. When an intermediate node receives a RREQ, it sends a route reply (RREP) if it is either the destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. The intermediate node also stores the previous node information in order to forward the data packet to this next node towards the destination. When the RREQ reaches the destination, a RREP will be generated by the destination node as a response to the RREQ. The RREP will be transmitted back to the originator of the RREQ in order to inform the route. If an intermediate node has an active route towards the destination, it can reply the RREQ with a RREP, which is called Gratuitous Route Reply. The intermediate node will also send an RREP to the destination node. The RREP will be sent in reverse route of RREQ if a bidirectional link exists. Whenever there is a link break in the routing path, the RERR message will be broadcasted by the link break identifying node to the neighbor nodes to update or delete the routes through that node and the source initiates another RREQ broadcast to find fresh routes to the destination.

2.3.3 Hybrid Routing Protocols

Hybrid routing protocols are mixed design of two approaches mentioned above [31]. The protocols typically use a proactive approach to keep routes to neighborhood nodes (nodes within the vicinity of the source). But for the nodes beyond the vicinity area the protocol behaves like a reactive one. Alternatively, multiple algorithms can be used simultaneously, if WMN is segmented into clusters. Within each cluster a proactive algorithm is used, whereas between clusters a reactive algorithm is used. The challenge is to choose a point, a point from which the protocol should change from proactive to reactive.

2.4 Hybrid Wireless Mesh Protocol (HWMP)

IEEE 802.11s specifies multi-hop MAC functions for mesh nodes using a mandatory path selection mechanism named HWMP (Hybrid Wireless Mesh Protocol) [1] and also provide a path selection framework for alternative mechanisms and future extensions.

Routing in HWMP

HWMP is the default routing protocol for IEEE 802.11s WLAN mesh networking. It has both reactive and proactive components. The foundation of HWMP is an adaptation of Ad hoc On-demand Distance Vector (AODV) protocol. It has four messages frames path request (PREQ), path replay (PREP), path error (PERR) and root announcement (RANN) [20]. HWMP uses destination sequence numbers in order to detect outdated routing information. Newly received routing information with a smaller sequence number than the sequence number of the corresponding information already known to the mesh point will be discarded, because it’s outdated. HWMP has two types of routing, proactive and reactive [3].

Reactive Routing in HWMP

The foundation of HWMP is an adaption of the reactive Ad hoc On-demand Distance Vector routing protocol (AODV) called Radio-Metric AODV (RM-AODV) [33]. While AODV works on layer 3 with IP addresses and uses the hop count as routing metric, RM-AODV works on layer 2 with MAC addresses and uses a radio-aware routing metric for the path selection [1]. In RM-AODV, it is assumed that each node has some mechanism to determine the metric cost of the link to each of its neighbors. In order to propagate the metric information between nodes, a metric field is used in the RREQ and RREP messages.

If a source MP needs to find a route using the on demand routing mode, it broadcasts a RREQ with the destination MP specified in the destination list and the metric field initialized to zero [6]. When a MP receives a RREQ, it creates a route to the source or updates its current route if the RREQ contains a greater sequence number, or the sequence number is the same as the current route and the RREQ offers a better metric than the current route. If a new route is created or an existing route modified, the RREQ is also forwarded. Each MP may receive multiple copies of the same RREQ that originated in the source, each RREQ traversing a unique path from the source to the MP. Whenever a MP forwards a RREQ, the metric field in the RREQ will be updated to reflect the cumulative metric of the route to the RREQ's source. After creating or updating a route to the source, the destination MP sends a unicast RREP back to the source. Intermediate MPs create a route to the destination on receiving the RREP, and also forward the RREP toward the source. When the source receives the RREP, it creates a route to the destination. If the destination receives further RREQs with a better metric, then the destination updates its route to the source to the new route and also sends a fresh RREP to the source along the updated route. Thus a bidirectional, best metric end-to-end route is established between the source and destination

Proactive Routing in HWMP

There are two mechanisms for proactively spreading routing information for reaching the root MP. The first method uses a proactive Route Request (RREQ) message and is intended to create routes between the root and all MPs in the network proactively. The second method uses a Root Announcement (RANN) message and is intended to distribute route information for reaching the root but the actual routes to the root can be built on-demand.

Proactive PREQ Mechanism

The RREQ tree building process begins with a proactive Route Request message sent by the root MP, with the destination address set to all ones (broadcast address). The RREQ contains the distance metric (set to 0 by the root) and a sequence number. The proactive RREQ is sent periodically by the root, with increasing sequence number [11, 29]. Any MP hearing a proactive RREQ creates or updates its forwarding information to the root MP, updates the metric and hops count of the RREQ, records the metric and hops count to the root, and then transmits the updated RREQ. Each MP may receive multiple copies of a proactive RREQ, each traversing a unique path from the root to the MP. A MP updates its current route to the root if and only if the RREQ contains a greater sequence number, or the sequence number is the same as the current route and the RREQ offers a better metric than the current route to the root [29].If the proactive RREQ is sent with the “Proactive RREP” bit set to 0, the recipient MP may send a gratuitous RREP if required (for example, if the MP has data to send to the root and requires establishing bidirectional route with the root). If the RREQ is sent with a “Proactive RREP” bit set to 1, the recipient MP shall send a gratuitous RREP. The gratuitous RREP establishes the route from the root to the MP. When the route from an MP to a root changes, and the root RREQ was sent with a “Proactive RREP” bit set to 1, it shall send a gratuitous RREP to the root containing the addresses of the MPs which have established a route to the root through the current MP [41].

Proactive RANN Mechanism

The root periodically floods a RANN message into the network. The information contained in the RANN is used to disseminate route metrics to the root. Upon reception of a

RANN, each MP that has to create or refresh a route to the root will send a unicast RREQ to the root via the MP from which it received the RANN. The unicast RREQ will follow the same processing rules defined in the on demand mode [33]. The root sends a RREP in response to each RREQ. The unicast RREQ creates the reverse route from the root to the originating MP, while the RREP creates the forward route from the MP to the root. When the route from an MP to a root changes, it may send a RREP with the addresses of the MPs which have established a route to the root through the current MP [42].

Chapter Three

Cryptography

Table of Contents

3.1 Introduction

Security and networking has very constrained bonds, since wireless networks are prone to various networks attacks because of its multi-hop communicating nature, dynamic mesh topological changes, and its shared wireless medium access.

Cryptography is the scientific study of techniques for securing digital information, transactions, and distributed computations [43]. Cryptography aims to transfer readable data into a form which cannot be understood for the purpose of securing data. Cryptography involves different distinct mechanisms [48]: encryption, authentication, digital signatures and hashing.

In this chapter we describe some cryptography elements that will be used throughout this thesis that include symmetric and asymmetric cryptography, Diffie-Hellman key exchange protocol, and digital signatures.

3.2 Encryption

The fundamental task for cryptography is to provide **confidentiality** by encryption methods. Encryption provides protection against **passive attacks** such as eavesdropping. The message to be transmitted is called **plaintext.** The sender applies **encryption algorithms** to encrypt the message into **cipher-text** and sends it to the receiver in order to **decrypt** the cipher-text.

3.2.1 Symmetric Encryption

Symmetric cryptography is based on using same key for encrypting and decrypting [44]. In the process of exchanging cipher-text over network sender encrypts data with a key normally referred as symmetric key or secret key **K** and using and encrypting algorithm $\mathbf{E_k(x)}$, on the other hand receiver is using decrypting algorithm $\mathbf{D_k(x)}$ and the same key **K** to decrypt cipher-text **C** and restore the plaintext **m**.

$$\mathbf{C= E_k(m)}$$

$$\mathbf{D_k(C)=D_k(E_k(m))= m}$$

This leads us to keep in mind two important requirements to use symmetric cryptography:

1. In symmetric encryption, it is essential that the sender and receiver have a secure channel to exchange secret keys.

2. It’s vital to use a strong encryption algorithm. What this means is that if someone has a cipher text, a corresponding plain text message and the encryption algorithm, they still cannot determine the key or decrypt another cipher text. In other words, someone who has a plain text for a given cipher text and the algorithm should not be able to break the cipher, however. Sharing a key with each possible communicating entity, even in a closed group of entities, is a very high constraint, and rapidly leads to a big number of keys to be managed. Thus, it is better automating the establishment of these keys [45].

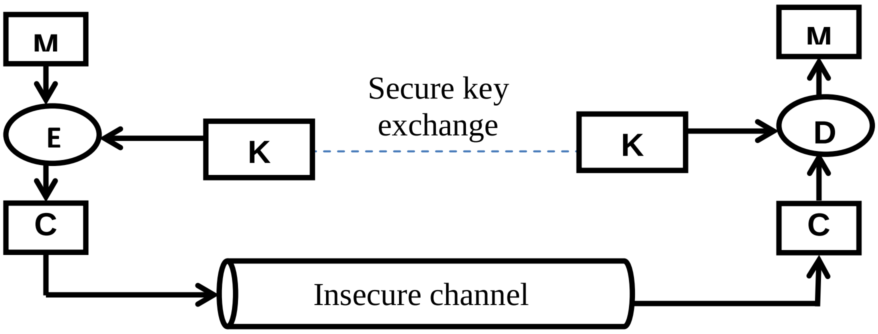

Figure3.1. Symmetric Cryptography

The most well-known symmetric algorithms are, DES (Data Encryption Standard), 3DES (“Triple DES”), and AES (Advanced Encryption Standard) [46]. DES was invented by IBM as the public encryption algorithm with secret keys of 56 bits and input of 64-bits data blocks. DES is based on combinations of mechanisms and exclusive OR gates. These fast operations make DES highly efficient, but brute force attacks are still able to crack the 56-bits keys by trying any combinations of keys [47].

The 3DES algorithm was more robust and was successor of DES; 3DES applies DES three times, one after the other; the 3DES key is maximum 168 bits (3*56=168) and applies to the same input block size (64 bits). The 3DES algorithm is not always efficient from an encryption rate point of view, robustness to brute-force attacks, so AES algorithm was selected for its fast processing time, several supported key lengths, and it relies on inputs of 128-bit blocks and key lengths of 128, 192 or 256 bits [48].

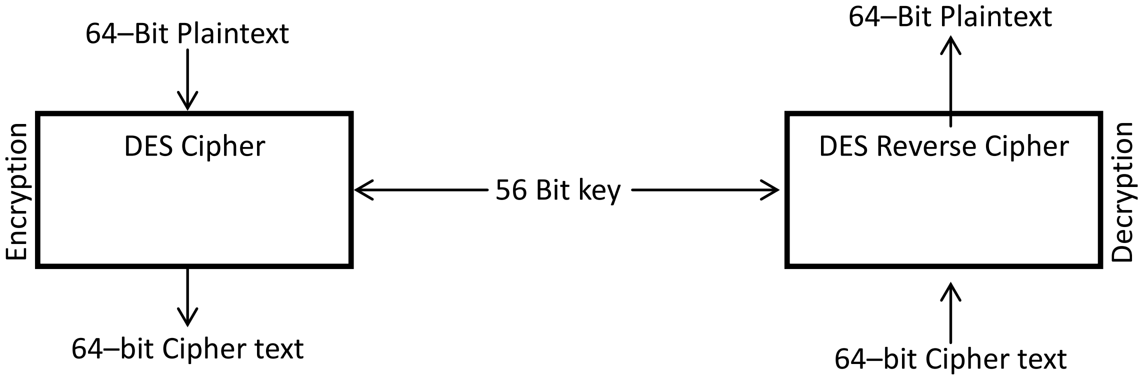

Figure3.2. DES Encryption Process

On the other hand, AES permits fast implementations. It doesn’t require much memory, which makes it suitable for small mobile devices. AES has not known to have been any successful attacks against it. However, the main disadvantage of symmetric encryption is the large number of keys required and the distribution of these keys between all parties in the network where each two parties have to exchange their keys before starting secure communications. For example, assume a network with N nodes, each need to communicate securely, then the number of required keys is $\left(N \times \frac{(N-1)}{2}\right)$. This is a big number, in particular, for large networks [48].

However, symmetric encryption can be used to implement the authentication security services. Symmetric encryption is fast which make it compatible with mobile devices, the main problem with symmetric cryptography is to have secure channel in order to exchange secret keys between communicating devices.

3.2.2 Asymmetric Encryption

Asymmetric or public key cryptography relies on two different keys, called "asymmetric keys". Both keys are generated at the same time and play a complementary role in the encryption process, the encryption is done using one of the keys needs to be decrypted with the other key. Each key plays a specific role [48]. The private key $\mathbf{K_r}$ must be known by only one entity and can be used for authenticating itself for instance. The public key $\mathbf{K_u}$ can be largely published and it is better that public keys are largely published in order that any other entity can perform authentication. Knowledge of $\mathbf{K_u}$ does not enable us to deduce $\mathbf{K_r}$. Therefore, asymmetric cryptography is the opposite of symmetric cryptography in safety, since it doesn't require sharing the secret key between the sender and the receiver [49]. This is the main difference between symmetric and asymmetric encryption.

$$C= E_{ku}(m)$$

$$D_{kr}\ (C)=D_{kr}(E_{ku}(m))= m$$

Sender encrypts the message with $\mathbf{K_u}$ of receiver and then forwards that encrypted message to the receiver over the network. Now on receiving that encrypted data, only receiver can decrypt it with the help of its corresponding $\mathbf{K_r}$. No other user can decrypt that message, until, unless, has the knowledge about the secret key of the receiver.

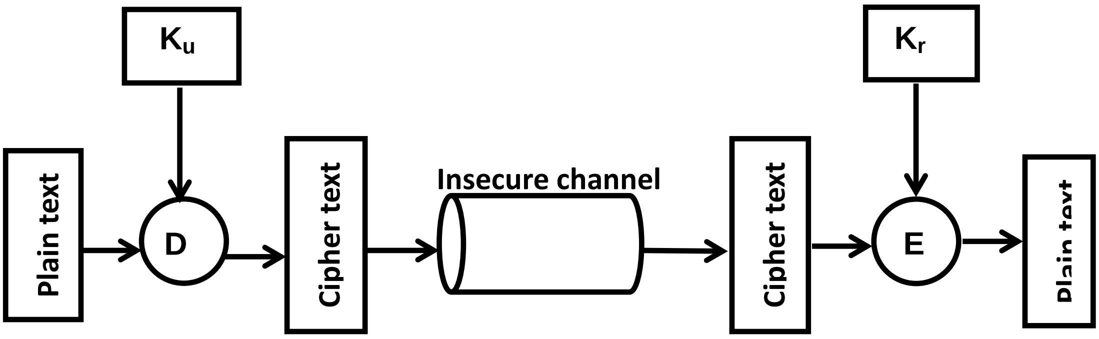

Figure3.3. Asymmetric Cryptography

Sender encrypts the message with $\mathbf{K_u}$ of receiver and then forwards that encrypted message to the receiver over the network. Now on receiving that encrypted data, only receiver can decrypt it with the help of its corresponding $\mathbf{K_r}$. No other user can decrypt that message, until, unless, has the knowledge about the secret key of the receiver.

We must understand that all the difficulty is related to the guarantee provided that a public key is truly associated with the unique identified entity. This association entity/public key is fundamental. With no such reliable guarantee, it is useless implementing security over a network. Different solutions have been developed to guarantee the entity/public association

1. First idea would be that each entity previously registers every public key of its correspondents.

2. A second solution consists of defining a trusted third party, that is, an entity in which a large number of entities have trust. This trust is created by the knowledge of a public key associated with the trusted third party [35]. This trusted third party can for instance take the form of a certification authority **CA** whose role is to issue electronic, i.e. some data structures binding a public key to an entity, and signed by the certification authority.

On the other hand, **asymmetric encryption** deals with **plaintext** as a group of numbers which are manipulated in mathematics, while **symmetric encryption** deal with plaintext as group of symbols and characters. The encryption process may change these symbols or may substitute one symbol by another. Therefore, asymmetric encryption is slower and very complicated in calculations than symmetric encryption. The advantage of asymmetric encryption is that the key used for encryption can be published for use by anyone, while the key used for decryption must be kept secret. Also, the number of keys is much smaller than that in symmetric encryption since, in **N** nodes each node needs **N** public keys which much smaller than the number of $\left(N \times \frac{(N-1)}{2}\right)$ key required in symmetric encryption [48]. However, asymmetric encryption can be used to implement the authentication and digital signature, and for integrity assurance.

In the following we describe two public-key algorithms **Diffie-Hellman Key Exchange** and **RSA Algorithm**.

Diffie-Hellman key Exchange

Whitfield Diffie and Martin Hellman provide a solution for key generation and distribution in symmetric cryptography. Diffie-Hellman algorithm depends for its effectiveness on the difficulty of computing discrete logarithms. Diffie-Hellman defines two publicly known numbers: a prime number **q** and an integer **α** that is a primitive root of **q**.

Diffie-Hellman model supposes user **A** and user **B**, were user **A** selects a random integer $\mathbf{X_A < q}$ and computes $\mathbf{Y_A = \alpha^{XA} \bmod q}$. Similarly, user **B** independently selects a random integer $\mathbf{X_B < q}$ and computes $\mathbf{Y_B = \alpha^{XB} \bmod q}$. Each side keeps the **X** value private and makes the **Y** available publicly to the other side. User **A** computes the key as $\mathbf{K = (Y_B)^{XA} \bmod q}$ and user **B** computes the key as $\mathbf{K = (Y_A)^{XB} \bmod q}$. these two calculations produce identical results:

$$\begin{aligned} K &= (Y_B)^{XA} \bmod q \\ &= (\alpha^{XB} \bmod q)^{XA} \bmod q \\ &= (\alpha^{XB})^{XA} \bmod q \\ &= \alpha^{XBXA} \bmod q \\ &= (\alpha^{XA})^{XB} \bmod q \\ &= (\alpha^{XA} \bmod q)^{XB} \bmod q \\ &= (Y_A)^{XB} \bmod q \end{aligned}$$

Since **X** values are private, an adversary only has {**q, α,** and **Y**} values to work with. Thus the adversary is forced to take a discrete logarithm to determine the key.

The following table depicts Diffie-Hellman Key Exchange protocol.

Table 3.1. Diffie-Hellman Generation Key Model

Global Public Elements	
q	prime number
α	$\alpha < q$ and **α** is a primitive root of **q**
User **A** Key Generation	
Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A= \alpha^{XA} \bmod q$
User **B** Key Generation	
Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B= \alpha^{XB} \bmod q$
Generation of Secret Key by user **A**	
$K = (Y_B)^{XA} \bmod q$	
Generation of Secret Key by user **B**	
$K = (Y_A)^{XB} \bmod q$	

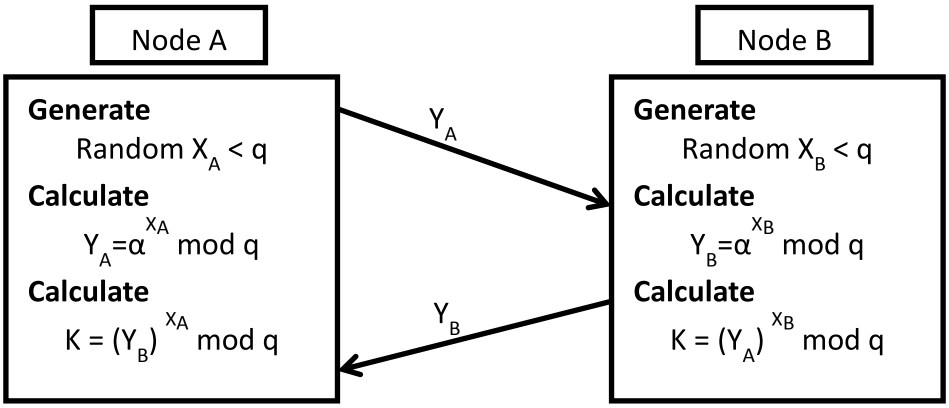

Figure3.4. Diffie-Hellman generating key between node A and B

RSA Algorithm

RSA encrypts **plaintext** in blocks, with each block having a binary value less than some number **n**. the block size must be equal or less than $\log_2(n) + 1$. **RSA** makes use of an expression with exponentials. Encryption and decryption are of the following form, for some plaintext block **m** and cipher-text block **C**:

$$C = m^e \bmod n$$

$$m = C^d \bmod n = (m^e)^d \bmod n = m^{ed} \bmod n$$

Both sender and receiver must know the value on **n**. the sender knows the value of **e**, and only the receiver knows the value of **d**. then the public key of **Ku = {e,n}** and a private key of $K_r = \{d,n\}$. **RSA** must satisfy the following requirements:

1. It is possible to find values of **e, d, n** such that $m^{ed} \bmod n = m$ for all $m < n$.
2. It is relatively easy to calculate $m^e \bmod n$ and $C^d \bmod n$ for all values of $m < n$.
3. It is infeasible to determine **d** given **e** and **n**.
4. In **RSA** every device computes a public and private key which can be used to encrypt/decrypt messages and to digitally signed messages after exchanges keys with other devices in the network, on the other hand Diffie-Hellman aims in to exchange shared secret key with other device on the network without the need to secure channel between them.

The following table depicts the process of generation keys in **RSA** algorithm.

Table 3.2. RSA Public key generation model

Key Generation	
Select **p,q**	**p** and **q** both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\varphi(n) = (p-1)(q-1)$	
Select integer **e**	$gcd(\varphi(n),e) = 1;\ 1< e < X_A < \varphi(n)$
Calculate **d**	$de \bmod \varphi(n) = 1$
Public key	$Ku = \{e,n\}$
Private key	$Kr = \{d,n\}$
Encryption	
Plain text Cipher-text	$m < n$ $C = m^e \bmod n$
Decryption	
Cipher-text Plain	C $m = C^d \bmod n$

Table 3.3. Public key algorithms comparison

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
Diffie-Hellman	No	No	Yes
RSA	Yes	Yes	Yes

3.3 Digital Signature

Digital signature provides message authentication in terms of source authentication and data integrity. The **digital signature** uses a pair of asymmetric keys. Sender **A** uses his private key $\mathbf{K_{rA}}$ to encrypt the hashed message, while the receiver **B** uses **A** s public key $\mathbf{K_{uA}}$ to decrypt the cipher-text. This proves the following to **B**:

1. The message has been encrypted by **A**.
2. The message has not been altered since no one has **A**’s private key.
3. The message has been authenticated in terms of source and data integrity.

In the previous scheme the entire message has been encrypted which means that the entire message represents a **digital** signature. In the other hand the encryption can be done to the hash of the message and append it to the message in this case the **digital signature** with provide authentication without confidentiality.

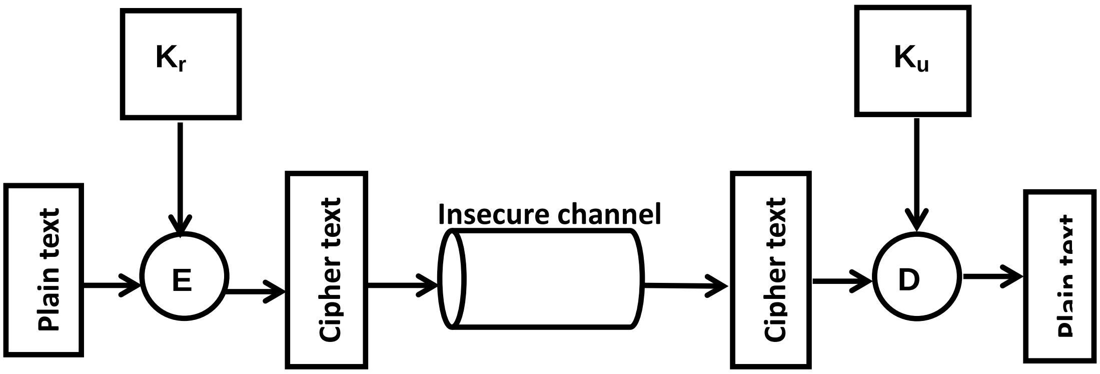

Figure3.5. Digital Signature Scheme

3.4 Integrity

Hash functions are a cryptographic building block used mostly in ensuring message integrity [35]. A hash function takes a variable length input of plain text, and computes a fixed-length output string called the hash, digest or fingerprint value. The main purpose of hash functions is to produce a value that indicates whether the content of sent message has modified or not or to check the integrity of received message [48]. Since it's computationally infeasible to find two different messages with the same hash values, the application of hash functions in cryptography is mostly used in techniques that are applied to guarantee message authentication such as digital signatures.

According to [49] secure hash function most generally has the following six properties:

1. Hash function **(H)** can be applied to a block of data of any size.
2. **H** produces a fixed-length output.
3. **H (M)** is relatively easy to compute for any given message **M**, making both hardware and software implementations practical.
4. For any given digest **m**, it is computationally infeasible to find **M** such that **H(M)=m**
5. For any given message **M**, it is computationally infeasible to find another message **M'≠ M** with **H (M') = H (M).**
6. It is computationally infeasible to find any pair **(M, M')** such that **H(M) = H(M').**

Examples of well-known hash functions are Message Digest 5 (MD5)[51] and Secure Hash Algorithm (SHA-1) [52].

3.5 Authentication

Authentication is any process through which one proves and verifies certain information. Sometimes one may want to verify the origin of a document, the identity of the sender, the time and date a document was sent and/or signed, the identity of a computer or user, and so on.

Authentication mechanisms are becoming more and more sophisticated on the information system security market. They are designed to provide users and administrators with a certain ease of use, a minimal administration, great robustness, high reliability and ubiquity of its usage.

In order to diversify the authentication methods, the IETF has standardized a generic authentication protocol called EAP for [53]. This protocol is generic, in that it is independent of the authentication method, and its role is limited to the transportation of authentication data between a client and a server. The content of these exchanges is not interpreted by the software layer EAP, but by the selected EAP method. As such, it brings the advantage that an EAP method suddenly detected as vulnerable can easily be changed to another more robust method while keeping the same EAP protocol. This makes the security equipment more flexible and able to evolve at low cost.

The EAP protocol is mainly operated in PPP or 802.11 (wireless) environments. Because of its limited role in encapsulation of authentication data, it is extremely simple and includes only four types of messages request, response, success and failure. Today, there are more than 40 EAP methods, but few of them are standardized like EAP-TLS, EAP-MD5 or EAP-SIM

EAP model has defined three types of entities involved in the authentication process:

Supplicant This is the port that wishes to or requests access to the services offered by the authenticator's system. Typically, the supplicant would be the client system, such as a laptop or a PDA.

Authenticator This is the port or device that enforces authentication before allowing access to services that are accessible via that port. In the basic WLAN configuration, the AP would typically be the authenticator.

Authentication Server (AS) This is the entity that performs the authentication function necessary to check the credentials of the supplicant, on behalf of the authenticator. The resulting decision consists of whether or not the supplicant is authorized to access the authenticator's services.

The supplicant converses with the AS as the other endpoint of the handshake, an authenticator must function as a pass-through entity. This means that the pass-through authenticator must relay the EAP packets to its corresponding peer. Depending on the direction of the conversation, the peer of the authenticator will be the supplicant on one side and the AS on the other side. The pass-through authenticator must verify the fields of the packets before forwarding them. The forwarding model for a pass-through authenticator is shown in Figure 3.6 [2].

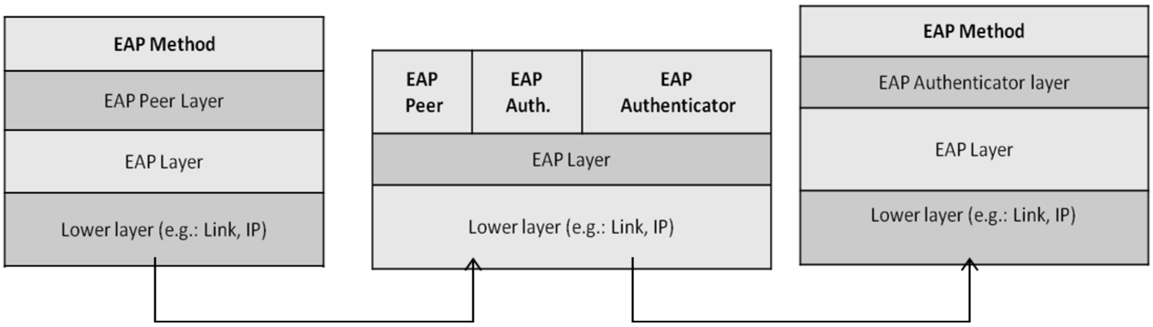

Figure3.6. EAP Layer Model

Lower layer It is responsible for transmitting and receiving EAP frames between the peer and authenticator. This layer includes Point-to-Point Protocol (PPP), Ethernet wired LAN, wireless LAN, etc.

EAP layer It receives and transmits EAP packets via the lower layer; implements duplicate detection and retransmission, and deliver and receive EAP messages to and from the EAP peer and authenticator layers.

EAP peer or EAP authenticator layer The EAP layer demultiplexes incoming EAP packets to the EAP peer and authenticator layers. Typically implementation on a host only will support either peer or authenticator functionality, but it is possible for a host to act as both.

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

There are many types of EAP methods available today using different kinds of mechanisms or technologies such as passwords, digital certificates, challenge-response, hash message, smart card, etc. Some of the available EAP methods are the following.

3.5.1 Ticket-based authentication

Massachusetts Institute for Technology (MIT) in the early 1980s has developed an authentication protocol called Kerberos [63]. Kerberos is a network authentication system based on using ***tickets*** [62]. Authority in Kerberos is built on two servers:

The Key Distribution Server (**KDS**) is responsible for issuing a ticket to the client. The ticket enables the client to contact the Ticket Granting Server (**TGS**) server securely by certifying the request is authentic and by establishing a session key $\mathbf{K_{MC\text{-}TGS}}$ between the client and the **TGS**. To do so, the **KDS** sends a randomly generated key $\mathbf{K_{MC\text{-}TGS}}$ encrypted with the client's public key and communicates the same key to the **TGS** using the ticket that is encrypted with the public **TGS** key. The ticket, which is not readable by the client, is forwarded unchanged by the client to the **TGS**. The ticket also contains the identifiers of the client and the **TGS**;

The **TGS** issues another ticket to the client. The principle of the ticket remains the same as before, i.e. the **TGS** receives from the client the ticket generated by the **KDS** and the ID encrypted with the key $\mathbf{K_{MC\text{-}TGS}}$. The **TGS** decrypts the ticket, deduces the key $\mathbf{K_{MC\text{-}TGS}}$, then decrypts the message built by the client and verifies the consistency between identifiers specified by the client and the **KDS** in the ticket.

In the case of consistency, this means that the request issued by the client is authentic because the client knows the key $\mathbf{K_{MC\text{-}TGS}}$, and to decipher $\mathbf{K_{MC\text{-}TGS}}$ it was necessary to know its private key.

The **TGS** server then proceeds similarly to the **KDS** generating a key $\mathbf{K_{MC\text{-}APP}}$. Likewise, **TGS** communicates this key to the client and application by sending the client the key **K_{MC-APP}** encrypted with **K_{MC-TGS}**, and issuing a ticket for the application encrypted with the public key of application. This Kerberos architecture, which is useful in establishing a secure channel between a client and an application, is very heavy: it requires implementing two Kerberos servers. For the first access by Kerberos, five exchanges of messages are needed with several encryption/decryptions. When accessing a second application, the procedure is lighter with only three messages between the client and the **TGS**, and the client and the application.

Some applications suggest controlling the access with a Kerberos ticket. This requires two Kerberos server architecture to be operational and that these servers are first contacted by the client.

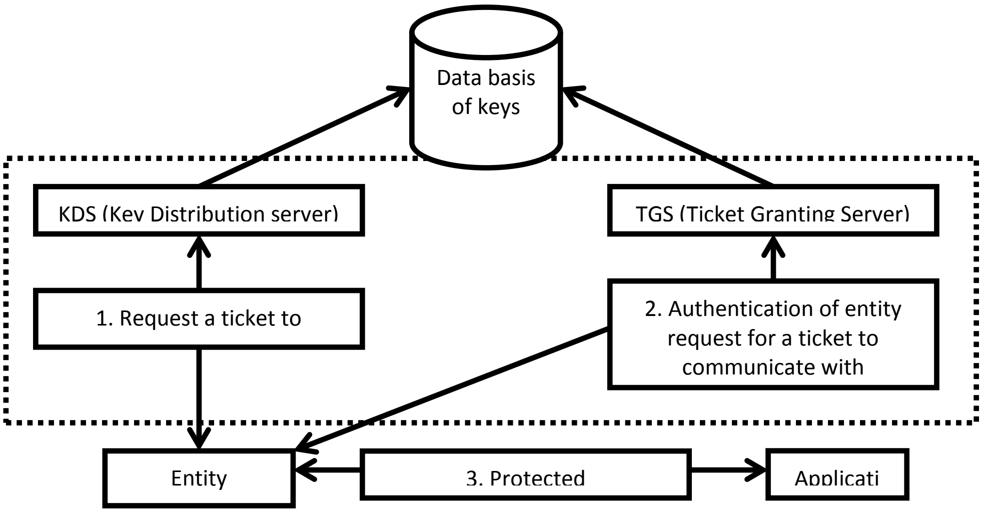
Data basis
of keys
KDS (Key Distribution server)
TGS (Ticket Granting Server)
1. Request a ticket to
2. Authentication of entity request for a ticket to communicate with
Entity
3. Protected
Applicati

Figure3.7. Ticket-based Authentication

3.5.2 Certificate-based Authentication

This type of authentication is based on asymmetric cryptography and makes it necessary to manage private keys of the entities and usually managing electronic certificates through PKIs.

Several EAP methods have been derived from the TLS protocol. The EAP-TLS method [53] standardized by the IETF supports the mutual authentication of client and server by using their private keys and certificates. The difficulty of EAP-TLS is in providing each user with private/public keys and ensuring good maintenance of them, in particular by ensuring that the private key of the client remains confidential.

EAP with Transport Layer Security (EAP-TLS)

EAP-TLS provides mutual authentication between Authentication server and client through an encrypted transport layer and the capability to dynamically change the keys. On the other hand, EAP-TLS is based on digital certificates and thus requires an infrastructure to manage issue, revoke, and verify certificates and keys.

EAP-TLS was also the first TLS-based EAP method for authentication shipped by Microsoft for the Windows platform, thereby achieving wider availability compared to other more recent proposals. EAP-TLS has found a strong following in many enterprise networks that seek strong security through authentication using strong credentials such as digital certificates. As implemented by Microsoft, EAPTLS requires both client-side and server-side certificates. For enterprises already running a PKI either internally or using a public CA and issuing employee certificates, the step to choosing EAP-TLS is a natural one. The choice of EAPTLS by enterprises is also made easier by the fact that the

Microsoft Windows Server (version 2000 or later) comes shipped with a CA Server, which allows enterprises to run their own private CA internally, and have seamless integration with the Microsoft directory-related products. This method considered as the strongest EAP method currently.

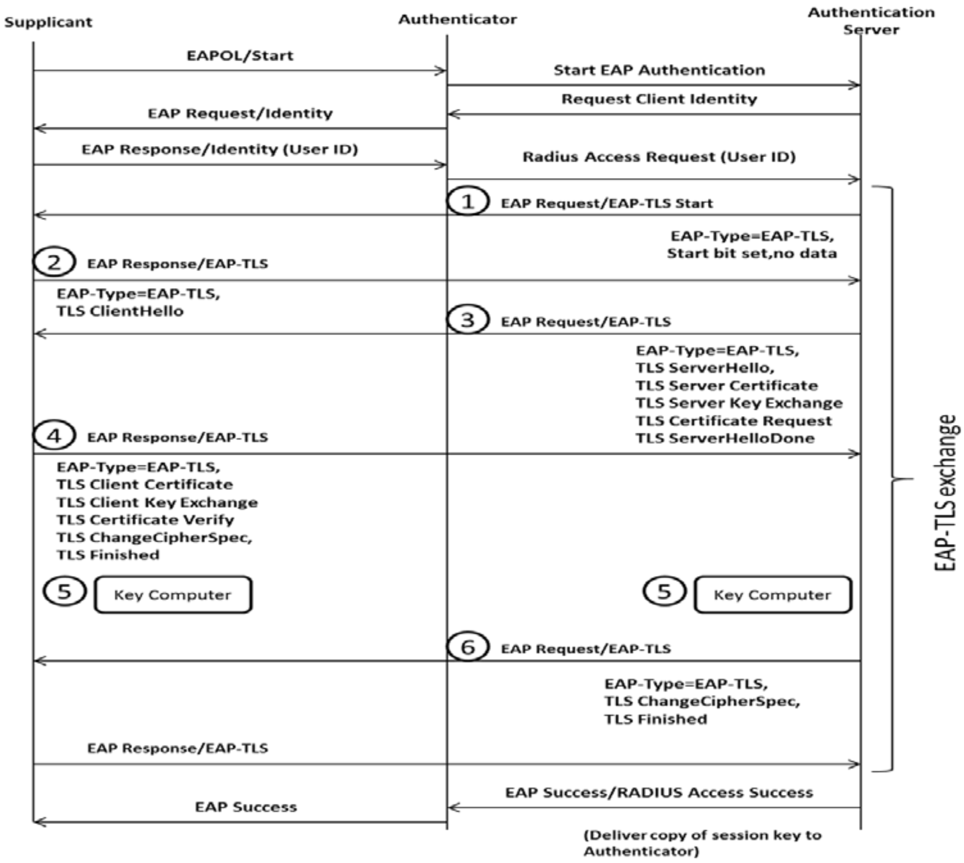
Supplicant
Authenticator
Authentication Server
EAPOL/Start
Start EAP Authentication
Request Client Identity
EAP Request/Identity
EAP Response/Identity (User ID)
Radius Access Request (User ID)
1 EAP Request/EAP-TLS Start
EAP-Type=EAP-TLS, Start bit set,no data
2 EAP Response/EAP-TLS
EAP-Type=EAP-TLS, TLS ClientHello
3 EAP Request/EAP-TLS
EAP-Type=EAP-TLS, TLS ServerHello, TLS Server Certificate TLS Server Key Exchange TLS Certificate Request TLS ServerHelloDone
4 EAP Response/EAP-TLS
EAP-Type=EAP-TLS, TLS Client Certificate TLS Client Key Exchange TLS Certificate Verify TLS ChangeCipherSpec, TLS Finished
5 Key Computer
5 Key Computer
6 EAP Request/EAP-TLS
EAP-Type=EAP-TLS, TLS ChangeCipherSpec, TLS Finished
EAP Response/EAP-TLS
EAP Success/RADIUS Access Success
EAP Success
(Deliver copy of session key to Authenticator)
EAP-TLS exchange

Figure3.8. EAP-TLS Authentication steps

Protected EAP (PEAP).

PEAP is actually EAP over TLS for the wireless domain. The PEAP is similar to EAP-TLS. The main difference is that PEAP does not require client authentication. The protected EAP (PEAP) method offers a solution to this need of user identity protection.

The other main difference is in compatibility with older methods and platforms which PEAP is less compatible compared to EAP-TTLS. PEAP allows users to submit their credential which may contain their identity after the TLS session has been established, and therefore have their credential passed to the server under the protection of the TLS session. PEAP also allows the server to request various forms of credentials from the client.

PEAP session can be divided into two phases:

Phase1: Here a TLS session is negotiated and established. The server authenticates itself to the client using server-side certificate, and optionally the client to the server. The resulting key is used to encipher the exchanges in phase2.

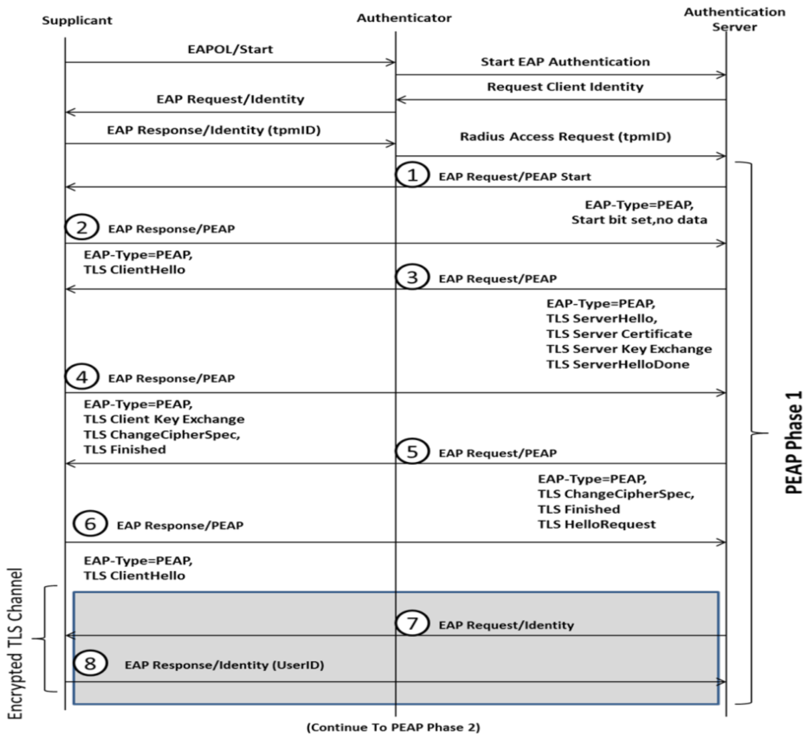

Figure3.9. EAP-PEAP Authentication Steps-phase1

Phase 2: Within the TLS session, zero or more EAP methods are carried out between the client and server, with a success/failure indication protected by the TLS session. Identity establishment is part of this phase.

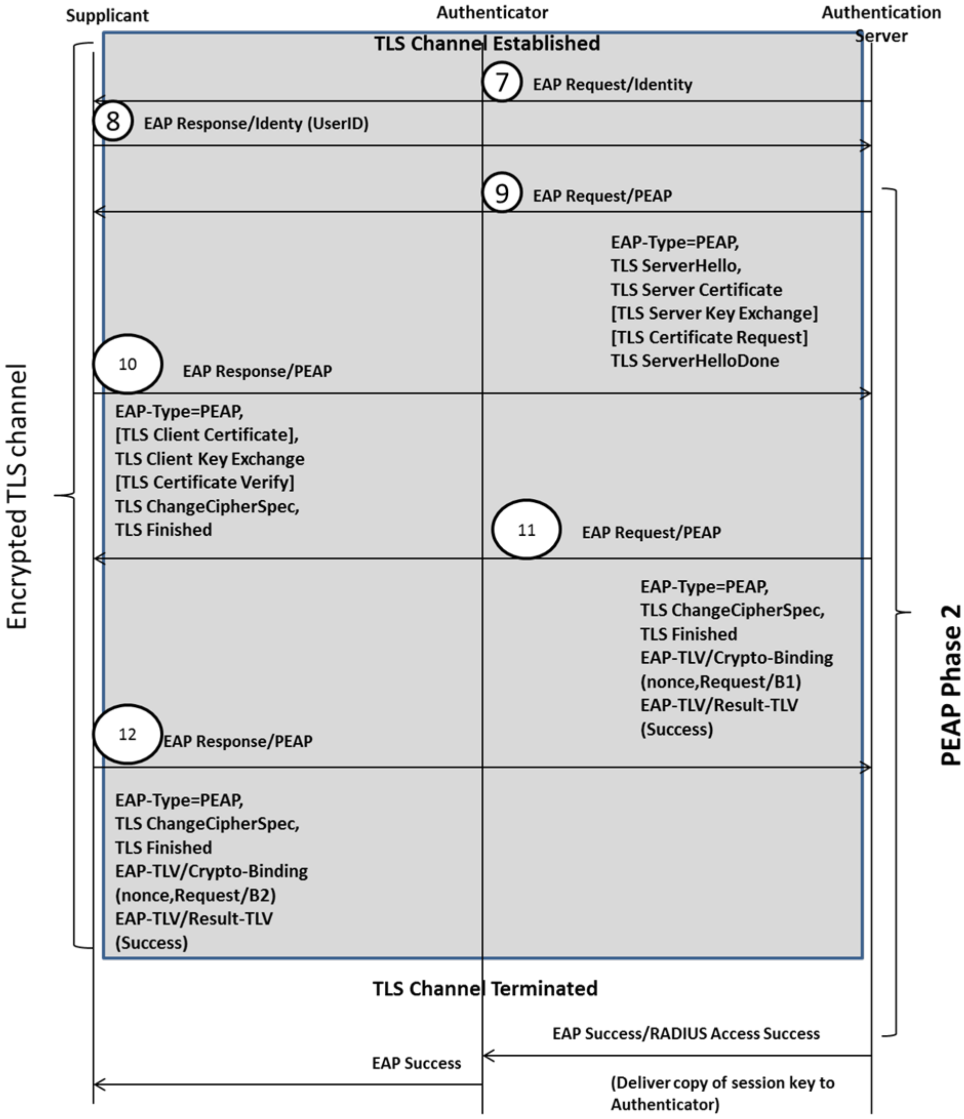

Figure3.10. EAP-PEAP

EAP-TTLS (Tunneled Transport Layer Security)

The EAP-TTLS extends EAP-TLS to exchange additional information between client and server by using the secure tunnel established by TLS negotiation. Then the client can be authenticated using any of the methods like username/PW, CHAP, and MSCHAPv2. This is called tunneled authentication. What this achieves is that the client does not require a digital certificate; only the authentication server needs one. This capability simplifies the client credential management. Organizations can also use currently available/legacy authentication methods (usually password-based schemes). An EAP-TTLS negotiation comprises two phases: **the TLS handshake phase** and **the TLS tunnel phase**.

During **handshake phase**, TLS is used for the client to authenticate the server. Optionally, the server can also authenticate the client. Similarly as in EAP-TLS, the authentication is done by using certificates. A secure TLS tunnel is also established after the phase-one handshake.

In **TLS tunnel phase**, the secure TLS tunnel can be used for other information exchanges, such as additional user authentication key, communication of accounting information, and so forth. In a WLAN environment, the EAP-TTLS usually is used as follows. In phase one, TLS is used as a supplicant to authenticate the authentication server by using a certificate.

Once the authentication server is authenticated, the authentication server authenticates the supplicant by using the supplicant’s username and password in phase two. The message exchanges are protected by the TLS tunnel established in phase one. The authentication of supplicant in phase two can use any non-EAP protocols such as PPP Authentication Protocols, PPP Challenge Handshake Authentication Protocol, Microsoft PPP CHAP Extensions, or Microsoft PPP CHAP Extensions, Version 2. Because only the authentication server needs to have a valid certificate, EAP-TTLS is more manageable than EAP-TLS.

Table 3.4. Comparison of authentication mechanisms.

	EAP-TLS	**EAP-TTLS**	**PEAP**
Server authentication	Public key (certificate)	Public key (certificate)	Public key (certificate)
Supplicant authentication	Public key (certificate or smart card)	Certificate, EAP, or non EAP protocols	Certificate or EAP protocols
Mutual authentication	Yes	Yes	Yes
Dynamic key deliver	Yes	Yes	Yes
Basic protocol architecture	Establish TLS session and validate certificates for both client and server	1. Establish TLS between client and TTLS server 2. Exchange attribute-value pairs between client and server	1. Establish TLS between client and PEAP server 2. Run EAP exchanges over TLS tunnel
Server certificate	Required	Required	Required
Client certificate	Required	Optional	Optional
Protection of user identity	NO	protected by TLS	protected by TLS
Security Risks	Identity Exposed	MITM Attack	MITM Attack and identity exposed in phase 1

3.6 Security in IEEE 802.11s

IEEE 802.11s ensures link security by using Mesh Security Association (MSA) services. 802.11s extends the security concept of 802.11 by a key hierarchy, inherits functions of 802.11i and uses 802.1X for initial authentication [2]. The operation of MSA relies on mesh key holders, which are functions that are implemented at MPs within the WMN.

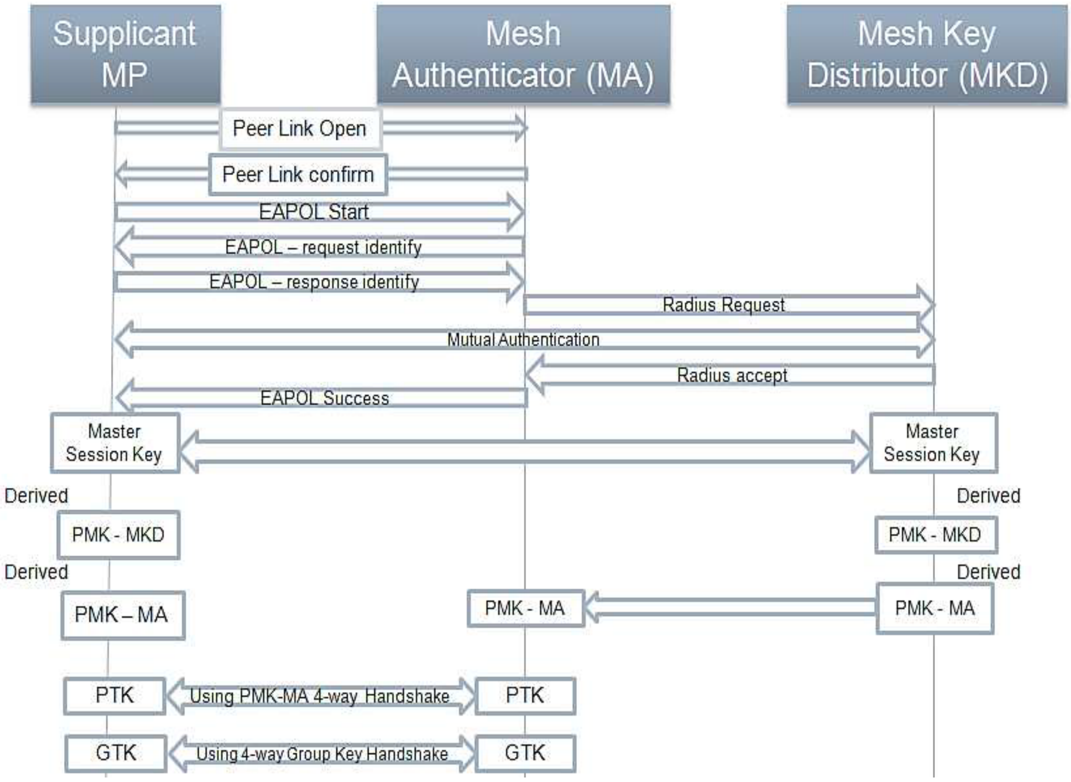

Figure3.11. Key Establishment procedure in 802.11s

Two types of mesh key holders are defined: mesh authenticators (MAs) and mesh key distributors (MKDs). A single MP may implement both MKD and MA key holders, an MA alone and no key holders. Fig. 7 depicts the key establishment procedure between two MPs in IEEE 802.11s. The first level of link security branch, PMK-MKD is mutually derived by the supplicant MP and MKD, from the Master Session Key (MSK) that is created after the initial authentication phase between supplicant MP and MKD or from a pre-shared key (PSK) between MKD and supplicant MP, if exists. The second level of link security branch PMK-MA is also derived by the supplicant MP and MKD. MKD then delivers the PMK-MA to the MA and thus permits to initiate MSA 4-way handshake which results in deriving a PTK of 512 bits between supplicant MP and MA. During the MSA 4-way handshake, an MA receives the GTK of the supplicant MP. After the completion of MSA 4-way handshake, a group handshake is used to send the GTK of the MA to the supplicant MP. The GTK is a shared key among all supplicant MPs that are connected to the same mesh authenticator (MA).

3.7 Summery

In this thesis we have used **asymmetric cryptography** depending on **Diffie-Hellman** exchange key model to obtain a secret shared session key between **MAPs**.

We have developed or protocol based on **Diffie-Hellman** rather than **RSA** since **Diffie-Hellman** provides **perfect forward secrecy** which means that if either node's key are compromised, the past sessions keys remain secure, while **RSA** can only provide **perfect forward secrecy** if a one-time RSA key pair is generated by one node who sends the public key to the other, and the other node returns a symmetric key encrypted with the public key which take much time than **Diffie-Hellman** [19].

We have used **symmetric encryption** using **EAP-TTLS** to authentication session between connected **MCs,** we have used **EAP-TTLS** over **PEAP** and **TLS**, since **EAP-TLS** requires client certificate which is not available all the time in mobile clients, and **PEAP** uses transport mode which provide protection for upper-layer protocols keeping the original IP header, but in **EAP-TTLS** encapsulates an entire IP packet within an IP packet to ensure that no part of the original packet is changed as it is moved through a network.

Chapter Four

Ticket Authentication Wireless Mesh Networks Protocol (TAWMP)

Table of Contents

4.1 Introduction

IEEE 802.11s standard defines secure links between MPs, but it does not provide end-to-end security. It also doesn’t specify routing security or forwarding functionality. Routing information elements are transferred in plain text and are prone to various types of routing attacks like flooding, route misdirection, spoofing. The main reason is that intermediate nodes need to modify mutable fields such as hop count, TTL, and metric in the routing elements before forwarding and re-broadcasting them. Since other nodes will act upon those added information, these must also be protected somehow from being forged or modified.

However, only source authentication does not solve this problem, because the information is added or modified in intermediate nodes. This motivates us to include node-to-node authentication in our proposal. More specifically, each node that adds information in the control packet should authenticate the added information in such a way that each other node acts upon that information should be able to verify its authenticity.

We developed a Ticket-based Authentication Wireless Mesh Protocol (TAWMP) for providing end-to-end security. TAWMP take into consideration WMN architecture and hybrid nature of routing protocol for mesh network, TAWMP protects WMN using hop-by-hop in addition to end-to-end authentication.

TAWMP divides the authentication process into two phases: the MAP phase in which a new MAP conducts the network, and the MC phase in which a new MC conducts the network. At the MAP phase, the protection is done by using authenticated Diffie–

Hellman key exchange model. On the other hand, MC devices in the second part of the authentication use an EAP-TTLS server-side certificate.

4.2 Threat Model

In this section we describe several attacks IEEE 802.11s is vulnerable to:

4.2.1 Flooding Attack

It is very easy for a malicious node to flood the network with a **PREQ** messages destined to an address which is not present in the network. As the destination node is not present in the network, every intermediate node will keep forwarding the **PREQ** messages. As a result, a large number of **PREQ** messages in a short period will consume the network bandwidth and can degrade the overall throughput. As shown in Figure 4.1, the malicious node **M** initiates route discovery with a **PREQ** for a destination that is not in the network. So that intermediate nodes rebroadcasts **PREQ** and within a short time the network is flooded with fake requests.

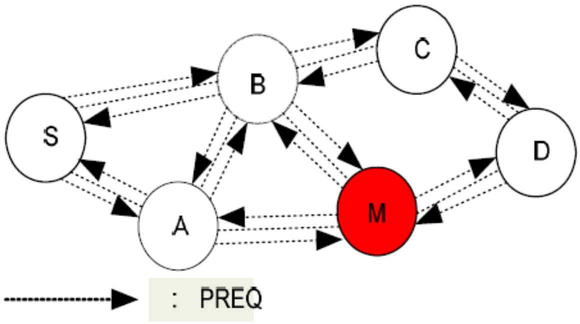

Figure 4.1 Flooding Attack

4.2.2 Route Disruption

By launching a route disruption attack, an adversary can prevent discovering a route between two legitimate nodes. In other word, if there an established route between the two victim nodes, but due the malicious behavior of the attacker, the routing protocol cannot discover it. In HWMP, route-disruption attacks can easily be launched by a malicious node as shown in figure 4.2. The malicious node **M** can prevent the discovery of routes between nodes **S** and **D**. **M** can modify the metric field value to zero on the **PREQ** message it receives from **A** or **B** and re-broadcast. So, after receiving the modified **PREQ**, **D** will choose **M** as the next hop in the reverse path and unicast **PREP** to **M**. Now, **M** can prevent the route discovery by dropping the valid **PREP** message destined for **S**.

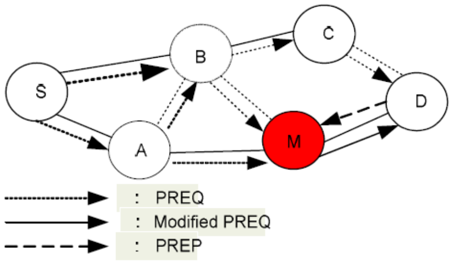

Figure 4.2 Route Disruption

4.2.3 Route Misdirection Attack

A malicious node **M** may divert traffic to itself by advertising a route to a destination with a destination sequence number (**DSN**) greater than the one it received from the destination. For example, the malicious node **M** in Figure 4.3 receives a **PREQ** from **A** which was originated from **S** for a route to node **D**.

As HWMP allows intermediate **PREP**, **M** may unicast **PREP** to A with a higher **DSN** than the value last advertised by **D**. So, **A** will misdirect all subsequent traffic destined for **D** to the malicious node **M**. Route misdirection attack can also be launched by modifying the mutable metric field used in the HWMP's **PREQ** messages. **A** malicious node can modify the mutable metric field to zero to announce a better path to a destination. As depicted in Figure 4.4, **M** can decrease the metric field in the **PREQ** to zero and re-broadcasts it to the network. So, the reverse path created should go through the malicious node **M**. As a result, all traffics to the destination **D** will be passed through the attacker.

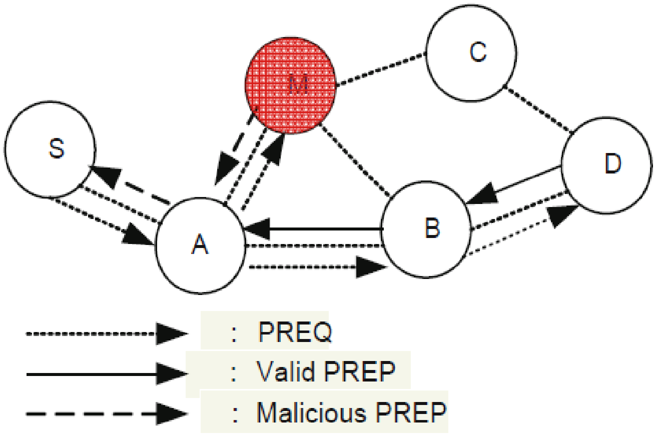

Figure 4.3 Route Misdirection Attack – Increasing DSN

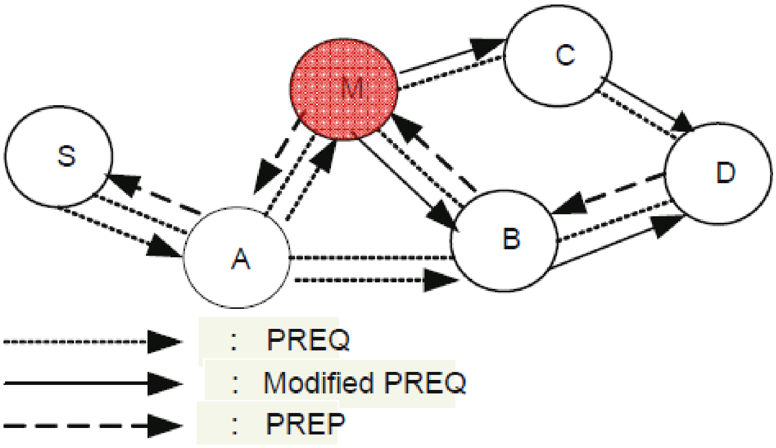

Figure 4.4 Route Misdirection Attack- Decreasing Metric

4.2.4 Routing Loops

A malicious node may create routing loops in a mesh network by spoofing MAC addresses and modifying the value of the metric field. Consider the following network scenarios in Figure 14 where a path exists between the source S and destination X that goes through node B and C. Also, there exists a path from A to C through D.

Assume that a malicious node **M**, as shown in Figure 4.5a, is present in the vicinity where it can listen to the **PREQ**/**PREP** messages that pass through **A**, **B**, **C** and **D** during route discovery process. It may create a routing loop among the nodes **A**, **B**, **C** and **D** by impersonation combined with modification of metric field in **PREP** message. First, it impersonates node **A**'s MAC address and moved out of the reach of node **A** and closer to node **B**. And then it sends a **PREP** message to node **B** indicating a better metric value than that of the value received from **C**. So, node **B** now re-establishes its route to **X** that should go through **A** as shown in Fig 4.5b. At this point, the malicious node impersonates node **B** and moves closer to node **C** and sends **PREP** to node **C** indicating a better metric then the one received from **E**. So, node **C** will now choose **B** as the next hop for its route to the destination **X** as shown in Figure 4.5c. Thus a loop has been formed and the destination **X** is unreachable from all the four nodes.

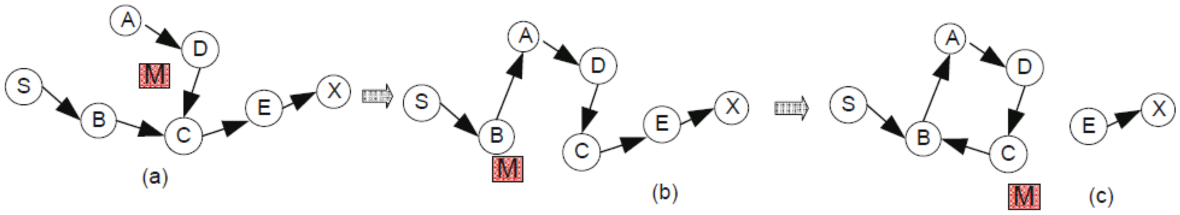

Figure 4.5 Routing Loops Information

4.3 Ticket Authentication Wireless Mesh Network Protocol (TAWMP)

TAWMP is an end-to-end authentication protocol that provides hop-to-hop security in addition to end-to-end authentication, in this integration protection TAWMP provides protection for the end-to-end communication between the device and its home network in general, and to protect the application content from being eavesdropped or modified during its transmission in particular. WMN as conventional WLAN is prone to several security threats and attacks because of its shared medium access, open peer-to-peer network topology, severe resource constraints, and highly dynamic topology, in addition to its multi-path connection [64]. TAWMP provides this protection by two phase authentication process which we describe in the following sections.

4.3.1 Assumptions

In our proposed protocol we assume an existing WMN that have at least:

1. One Authentication server (AS) attached to Mesh Access Point that has high processing capabilities in order to handle high bit rate and throughput during authentication process.

2. This Authentication Server has a pre-defined list of the valid MAP's MAC address who's allowed to connect the WMN.

3. This Authentication Server has a database of a hashed value for (username/password) for each MC who's allowed to the WMN.

4.3.2 MAP Phase

This phase aims to provide protection for backbone network by using Diffie–Hellman function for the key agreement and Harn's scheme [65], when a MAP is connected to a WMN during setup stage; it has to share a secret key with AS in order to exchange public keys:

1. MAP sends its details including the type (1 for MAP / 0 for MC) to an Authentication Server (AS).

$$\text{MAP} \longrightarrow \text{AS:} \quad \{\text{Type}\|\ \text{ID}_{\text{MAP}}\ \|\ \text{N}_{\text{once}}\}\ _{\mathbf{H}(MAC|\ Type\|\ IDMAP\ \|\ Nonce)}$$

MAP sends this massage using hashed value of MAC address to guarantee integrity of the message since the AS already has the MAC address of specific ID_{MAP}, so the MAP will not send MAC within the message.

2. After receiving this message AS check the integrity of the message by calculating the hash function of (MAC, ID of the MAP, Type, and Nonce), after verifying the integrity of the message AS selects a short-term private key $\mathbf{K_{r(AS)}}$ and computes a short-term public key $\mathbf{K_{u(AS)}}$ using the equation.

$$\mathbf{K_{u(AS)}} = \mathbf{(\alpha)}^{\boldsymbol{Kr(AS)}}\ \mathbf{mod}\ \boldsymbol{p} \qquad \boldsymbol{(4.1)}$$

Where p is a prime number, α is generator with order q, q is factor of p-1.

After computing $\mathbf{K_u}$, AS starts to transmit the following message to MAP.

$$\text{AS} \longrightarrow \text{MAP:} \quad \{\mathbf{K_{u(AS)}}\ \|\ \text{N}_{\mathbf{once}}\}\ _{\mathbf{H}(MAC\|\ \boldsymbol{Ku(AS)}\ \|\ Nonce))}$$

AS sends $\mathbf{K_{u(AS)}}$ with a hashed value of (MAP's MAC address and $\mathbf{K_{u(AS)}}$) in order to guarantee the integrity of the message.

3. In response to receiving $\mathbf{K_{u(AS)}}$. MAP selects his short-term private key $\mathbf{K_{r(MAP)}}$ and computes the following values
   - A short-term public key $\mathbf{K_{u(MAP)}}$.
   - A shared secret key between AS and connecting MAP $\mathbf{K_{r(AS\text{-}MAP)}}$which will be used to decrypt messages from AS to MAP.
   - A long-term private key $\mathbf{K_{R(MAP)}}$ in order to obtain MAP's signature $\mathbf{S_{(MAP)}}$ which will be used later.
   - A long-term public key $\mathbf{K_{U(MAP)}}$ which will be send to AS.

$$\mathbf{K_{r(AS\text{-}MAP)} = (K_{u(AS)})^{Kr(MAP)} \bmod \mathit{p}} \qquad \textit{(4.2)}$$

$$\mathbf{S_{(MAP)}} = (\mathbf{K_{r(AS\text{-}MAP)}})^{-1} \, \mathbf{(K_{R(MAP)} - K_{u(MAP)}\, K_{r(MAP)}) \bmod \mathit{q}} \qquad \textit{(4.3)}$$

$$\mathbf{K_{R(MAP)}} = \mathbf{K_{u(MAP)}\, K_{r(MAP)}} + \mathbf{S_{(MAP)}\, K_{r(AS\text{-}MAP)} \bmod \mathit{p}} \qquad \textit{(4.4)}$$

$$\mathbf{K_{U(MAP)}} = \mathbf{(\alpha)}^{\boldsymbol{\mathit{KR(AS)}}} \mathbf{\bmod \mathit{p}} \qquad \textit{(4.5)}$$

4. MAP sends $\mathbf{K_{u(MAP)}}$ along with $\mathbf{K_{U(MAP)}}$ and signed the message using $\mathbf{S_{(MAP)}}$.

$$\text{MAP} \longrightarrow \text{AS: } \mathbf{S_{(MAP)}}\{\, \mathbf{K_{u(MAP)}} \,\|\, \mathbf{K_{U(MAP)}} \,\}\|\, \text{N}_{\mathbf{once}}$$

5. AS computes $\mathbf{K_{u(AS\text{-}MAP)}}$ that used to encrypt messages from AS to MAP.

$$\mathbf{K_{u(AS\text{-}MAP)} = (K_{U(MAP)})^{Kr(AS)} \bmod \mathit{p}} \qquad \textit{(4.6)}$$

Then AS verifies $\mathbf{S_{(MAP)}}$ and $\mathbf{K_{u(AS\text{-}MAP)}}$ by checking if the following equation is valid

$$\mathbf{K_{U(MAP)} = (K_{U(MAP)})^{Ku(MAP)} (\alpha)^{\mathit{S(MAP)Kr(AS)}} \bmod \mathit{p}} \qquad \textbf{\textit{(4.7)}}$$

After a successful verification AS convinced that $\mathbf{S_{(MAP)}}$ signed by MAP and that $\mathbf{K_{u(MAP)}}$and $\mathbf{K_{U(MAP)}}$ is selected by MAP, since $\mathbf{S_{(MAP)}}$ is a signature of both values.

AS computes A shared secret key between connecting MAP and AS $\mathbf{K_{r(MAP\text{-}AS)}}$which will be used to decrypt messages from MAP to AS.

$$\mathbf{K_{r(MAP\text{-}AS)} = (K_{u(MAP)})^{Kr(AS)} \bmod \mathit{p}} \qquad \textbf{\textit{(4.8)}}$$

AS computes $\mathbf{K_{R(AS)}}$ which used to obtain $\mathbf{S_{(MAP)}}$ as the following equations:

$$\mathbf{K_{R(AS)}} = \mathbf{K_{u(AS)}\, K_{r(AS)}} + \mathbf{S_{(AS)}\, K_{r(MAP\text{-}AS)} \bmod \mathit{p}} \qquad \textbf{\textit{(4.9)}}$$

$$\mathbf{S_{(AS)}} = (\mathbf{K_{r(MAP\text{-}AS)}})^{-1}\, \mathbf{(K_{R(AS)} - K_{u(AS)}\, K_{r(AS)}) \bmod \mathit{q}} \qquad \textbf{\textit{(4.10)}}$$

6. AS sends $\mathbf{K_{U(AS)}}$ signed in the message using $\mathbf{S_{(AS)}}$.

$$\text{AS} \longrightarrow \text{MAP: } \mathbf{S_{(AS)}}\{\ \mathbf{K_{U(AS)}}\ \} \| \text{N}_{\mathbf{once}}$$

7. MAP computes $\mathbf{K_{u(MAP\text{-}AS)}}$ that used to encrypt messages from MAP to AS.

$$\mathbf{K_{u(MAP\text{-}AS)} = (K_{U(AS)})^{Kr(MAP)} \bmod \mathit{p}} \qquad \textbf{\textit{(4.11)}}$$

MAP verifies $\mathbf{S_{(AS)}}$ and $\mathbf{K_{u(MAPS\text{-}AS)}}$ by checking if the following equation is valid

$$\mathbf{K_{U(AS)} = (K_{U(AS)})^{Ku(AS)} (\alpha)^{\mathit{S(AS)Kr(MAP\text{-}AS)}} \bmod \mathit{p}} \qquad \textbf{\textit{(4.12)}}$$

After a successful verification AS convinced that $\mathbf{S_{(AS)}}$ signed by AS and that $\mathbf{K_{U(AS)}}$ is selected by MAP, since $\mathbf{S_{(AS)}}$ is a signature for $\mathbf{K_{U(AS)}}$. Furthermore since MAP calculates $\mathbf{K_{u(MAP\text{-}AS)}}$ based on $\mathbf{K_{U(AS)}}$, MAP is convinced that $\mathbf{K_{u(AS)}}$ has been sent be AS. As we can see in figure4.6 a flowchart depicts the process of conducting a new MAP to WMN.

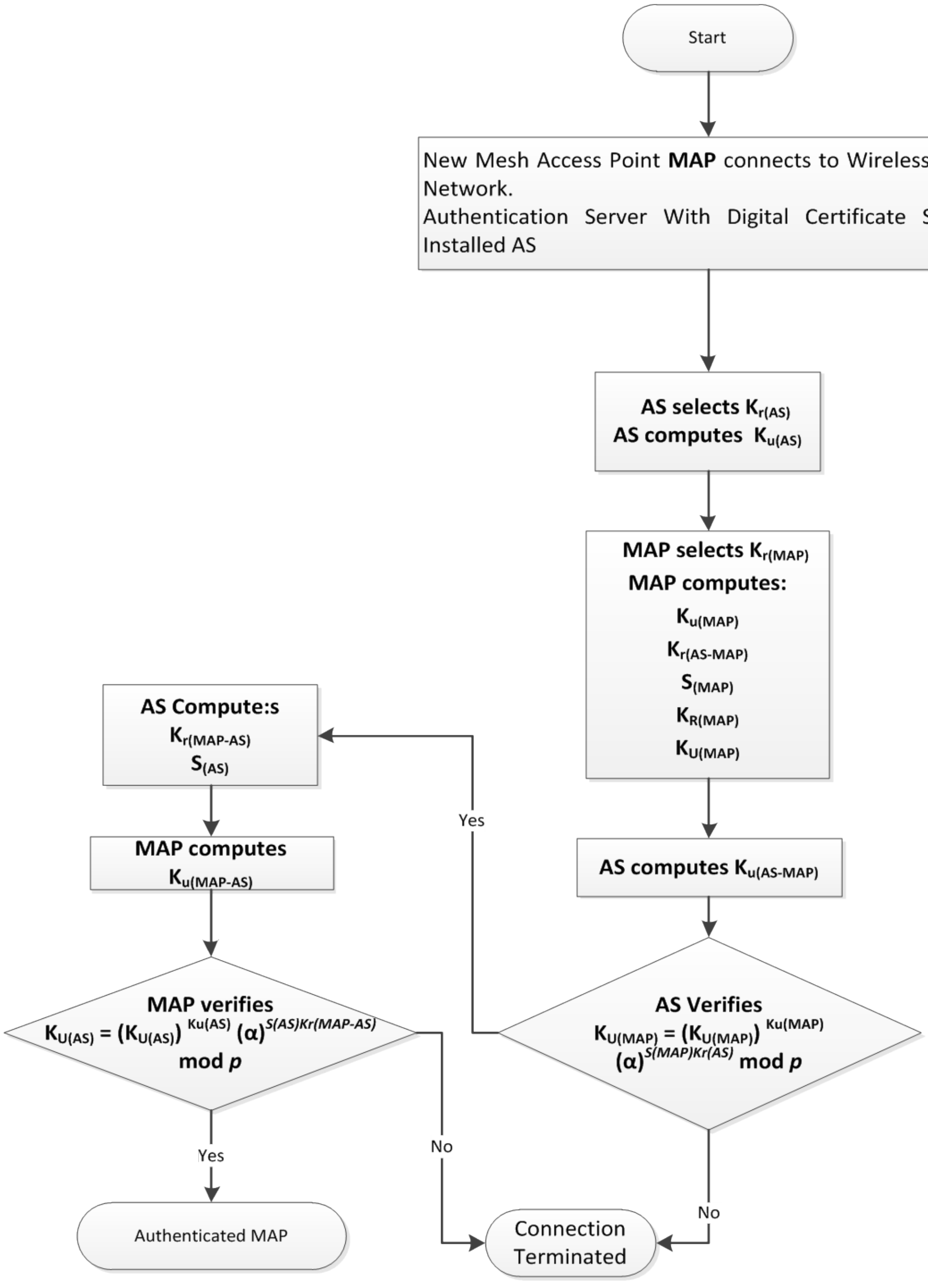
Start
New Mesh Access Point **MAP** connects to Wireless Mesh Network.
Authentication Server With Digital Certificate Service Installed AS
AS selects $K_{r(AS)}$
AS computes $K_{u(AS)}$
MAP selects $K_{r(MAP)}$
MAP computes:
$K_{u(MAP)}$
$K_{r(AS\text{-}MAP)}$
$S_{(MAP)}$
$K_{R(MAP)}$
$K_{U(MAP)}$
AS computes $K_{u(AS\text{-}MAP)}$
AS Verifies
$K_{U(MAP)} = (K_{U(MAP)})^{Ku(MAP)} (\alpha)^{S(MAP)Kr(AS)} \bmod p$
Yes
No
AS Compute:s
$K_{r(MAP\text{-}AS)}$
$S_{(AS)}$
MAP computes
$K_{u(MAP\text{-}AS)}$
MAP verifies
$K_{U(AS)} = (K_{U(AS)})^{Ku(AS)} (\alpha)^{S(AS)Kr(MAP\text{-}AS)} \bmod p$
Yes
No
Authenticated MAP
Connection Terminated

Figure 4.6 Flow chart depicts MAP Phase

4.3.3 MAP-to-MAP authentication

As it has been mentioned before, MAP depends on proactive protocols in order to build routing table through periodical exchanges of connectivity information, when a MAP discovers a new neighboring MAP, a secure route must be established by doing the following steps:

1. The first MAP sends both its identifier and the identifier of destination MAP to the AS, which in turn looks up both identifiers in its database in order to verify the validity of both clients.

$$\text{MAP1} \rightarrow \text{AS: } \{ID_{\mathbf{MAP1}}, ID_{\mathbf{MAP2}}\}\ \mathbf{K}_{\mathbf{u(MAP1\text{-}AS)}} \,\|\, N_{\mathbf{once}}$$

2. AS sends a shared secret key $\mathbf{K}_{\mathbf{MAP1\text{-}2}}$ and primitive values required for generating shared keys between two MAPs along with ***Authenticating_Data*** which contains $\mathbf{K}_{\mathbf{MAP1\text{-}2}}$ and $ID_{\mathbf{MAP1}}$ is encrypted with $\mathbf{K}_{\mathbf{u(AS\text{-}MAP2)}}$. This ***Authenticating_Data*** provides MAP_2 with the shared key and proof that this is the right shared key to use with MAP_2 at this time.

$$\text{AS} \rightarrow \text{MAP1: } \{\mathbf{K}_{\mathbf{MAP1\text{-}2}} \| \mathbf{ID}_{\mathbf{MAP2}}\}\ \mathbf{K}_{\mathbf{u(AS\text{-}MAP1)}} \,\|\, \{\mathbf{K}_{\mathbf{MAP1\text{-}2}} \| \mathbf{ID}_{\mathbf{MAP1}}\}\ \mathbf{K}_{\mathbf{u(AS\text{-}MAP2)}} \,\|\, \mathbf{N}_{\mathbf{once}}$$

MAP_1 sends a message to MAP_2 contains $ID_{\mathbf{MAP1}}$, $ID_{\mathbf{MAP2}}$, ***Authenticating_Data***, and primitive values required to generate keys encrypted with $\mathbf{K}_{\mathbf{MAP1\text{-}2}}$.

$$\text{MAP1} \rightarrow \text{MAP2: } \{\mathbf{K}_{\mathbf{MAP1\text{-}2}} \| \mathbf{ID}_{\mathbf{MAP1}}\}\ \mathbf{K}_{\mathbf{u(AS\text{-}MAP2)}} \,\|\, \alpha, \boldsymbol{q}, \boldsymbol{p} \,\|\, \mathbf{ID}_{\mathbf{MAP1}}, \mathbf{ID}_{\mathbf{MAP2}} \,\|\, \mathbf{N}_{\mathbf{once}}$$

3. After receiving this message MAP$_2$ decrypts ***Authenticating_Data*** with $\mathbf{K_{r(AS\text{-}MAP2)}}$ to obtain $\mathbf{K_{MAP1\text{-}2}}$ which in turn used to secure connection between the two MAPs while they generating authenticating keys.

MAP$_2$ selects a short-term private key $\mathbf{K_{r(MAP2)}}$ and computes a short-term public key $\mathbf{K_{u(MAP2)}}$ using equation ***(4.1)***, Then MAP$_2$ starts to transmit $\mathbf{K_u}$ to MAP$_1$.

$$\text{MAP2} \longrightarrow \text{MAP1:}\{\mathbf{K_{u(MAP2)}} \,||\, \text{N}_{\mathbf{once}}\}\ \mathbf{K_{MAP1\text{-}2}}$$

4. In response to receiving $\mathbf{K_{u(MAP2)}}$. MAP$_1$ selects his short-term private key $\mathbf{K_{r(MAP1)}}$ and computes the following values:

   - A short-term public key $\mathbf{K_{u(MAP1)}}$ using equation ***(4.1)***, and a shared secret key $\mathbf{K_{r(MAP2\text{-}MAP1)}}$ using equation ***(4.2)*** which will be used to decrypt messages from MAP$_2$ to MAP$_1$.
   - A long-term private key $\mathbf{K_{R(MAP1)}}$ using the equation ***(4.4) .***
   - A long-term public key $\mathbf{K_{U(MAP1)}}$ using equation ***(4.5)*** which will be send to MAP$_2$.

5. MAP$_1$ sends $\mathbf{K_{u(MAP1)}}$ along with $\mathbf{K_{U(MAP1)}}$.

$$\text{MAP1} \longrightarrow \text{MAP2: }\{\{\mathbf{K_{u(MAP1)}} \,||\, \mathbf{K_{U(MAP1)}}\ \}||\ \text{N}_{\mathbf{once}}\}\ \mathbf{K_{MAP1\text{-}2}}$$

6. MAP$_2$ computes $\mathbf{K_{u(MAP2\text{-}MAP1)}}$ that used to encrypt messages from MAP$_2$ to MAP$_1$.

$$\mathbf{K_{u(MAP2\text{-}MAP1)} = \ (K_{U(MAP1)})^{Kr(MAP2)}\ mod}\ \boldsymbol{p} \qquad \textit{(4.13)}$$

7. MAP$_2$ sends $\mathbf{K_{U(MAP2)}}$ to MAP$_1$.

$$\text{MAP2} \longrightarrow \text{MAP1:}\{\{\ \mathbf{K_{U(MAP2)}}\ \}||\ \text{N}_{\mathbf{once}}\}\ \mathbf{K_{MAP1\text{-}2}}$$

8. MAP$_{\mathbf{1}}$ computes $\mathbf{K_{u(MAP1\text{-}MAP2)}}$ that used to encrypt messages from MAP1 to MAP2 using equation ***(4.11).***

As we can see in figure 4.7 a flowchart depicts the steps required to connect two MAPs in a WMN.

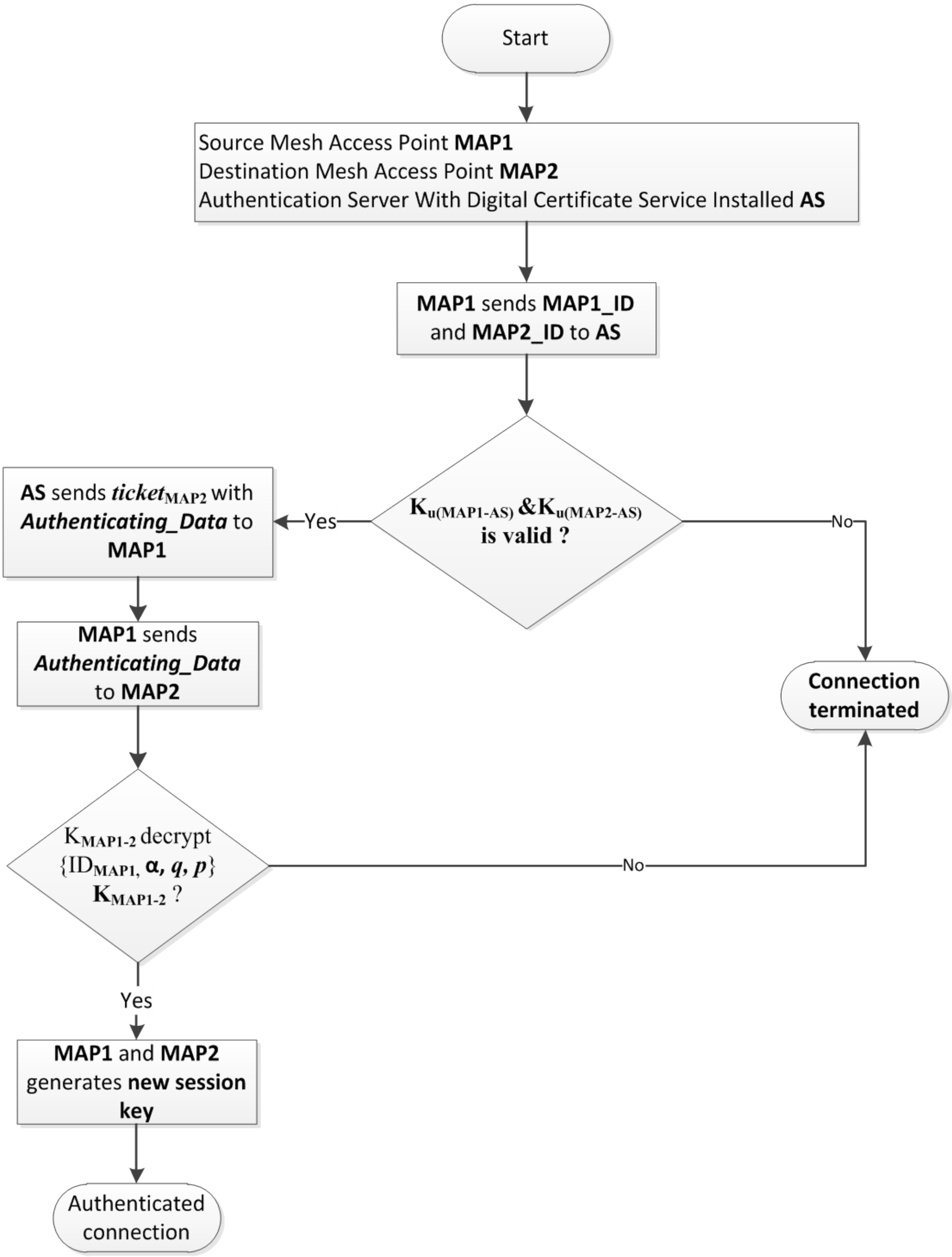
Start
Source Mesh Access Point MAP1
Destination Mesh Access Point MAP2
Authentication Server With Digital Certificate Service Installed AS
MAP1 sends MAP1_ID and MAP2_ID to AS
$K_{u(MAP1-AS)}$ & $K_{u(MAP2-AS)}$ is valid ?
Yes
No
AS sends $ticket_{MAP2}$ with Authenticating_Data to MAP1
MAP1 sends Authenticating_Data to MAP2
Connection terminated
K_{MAP1-2} decrypt $\{ID_{MAP1,}\ \alpha, q, p\}$ K_{MAP1-2} ?
No
Yes
MAP1 and MAP2 generates new session key
Authenticated connection

Figure 4.7 Flow Chart depicts MAP-MAP Authentication

4.3.4 MC phase

When a new MC is connected to the WMN, it has to provide credentials to the AS. These credentials can be user-name/ID-number and password via EAP-TTLS server-side certificate. After successful authentication, the mobile node will have a secret key that shares with the AS.

The following are the steps that MC has to go through in order to be authenticated:

1. MC sends **EAPOL** to AS via authenticator (MAP)

   MC → AS:**EAPOL_**start

2. AS sends an **EAP-Request/Identity** packet to MC

   AS → MC:**req_id**

3. MC responds with an **EAP-Response/Identity** packet to AS containing the client's userID/sessionID

   MC → AS:**MC_id|| S_id**

4. AS responds with an EAP-type Start packet (EAP-TTLS)

   AS → MC:**EAP_**start

5. MC starts TLS handshake process by sending "Client hello" message to the server, along with the client's random value (nonce).

   MC → AS:**MC_hello||Nonce**

6. AS responds with “Server hello” message to the client, along with the server’s random value (nonce), its certificate signed by itself (self-signed certificate) with an assumption that the server is already known to trusted by the clients as the certificate authority , and the “Server hello done” message.

$$\text{AS} \longrightarrow \text{MC:}\mathbf{AS\_hello}\|\ \mathbf{S\_id}\|\ \text{Cert}\{\mathbf{AS\_id},\ \mathbf{K_{u(AS)}}\}\|\mathbf{Nonce}$$

7. Client generates a random Pre-Master Secret (PMS) which is treated as Nonce since it’s randomly generated.

MC encrypts the PMS with the public key from the server’s certificate and sends it to AS.

MC generates a shared secret key depending on PMS.

$$\text{MC} \longrightarrow \text{AS:}\mathbf{\{PMS\}ku_{(AS)}}\|\{\mathbf{H(_{M,\ S\_id,\ MC\_id,\ AS\_id,\ Nonce})}\}\ \mathbf{ku_{(MC\text{-}AS)}}$$

8. Server and Client each generate the Master Secret and session keys based on the Pre-Master Secret and the nonce using pseudo-random-number function.

$$\text{MC} \longrightarrow \text{AS:}\{\mathbf{H(_{M,\ S\_id,\ MC\_id,\ AS\_id,\ Nonce})}\|\ \mathbf{Nonce}\}\ \mathbf{ku_{(MC\text{-}AS)}}$$

9. Client sends “Client cipher spec” notification to Server, to indicate that the client will start using the new session key for hashing and encrypting messages. After AS receives “Client cipher spec” it switches to symmetric encryption using session keys.

In figure 4.8 we can see a flowchart depicts the steps of conducting new MC to a WMN

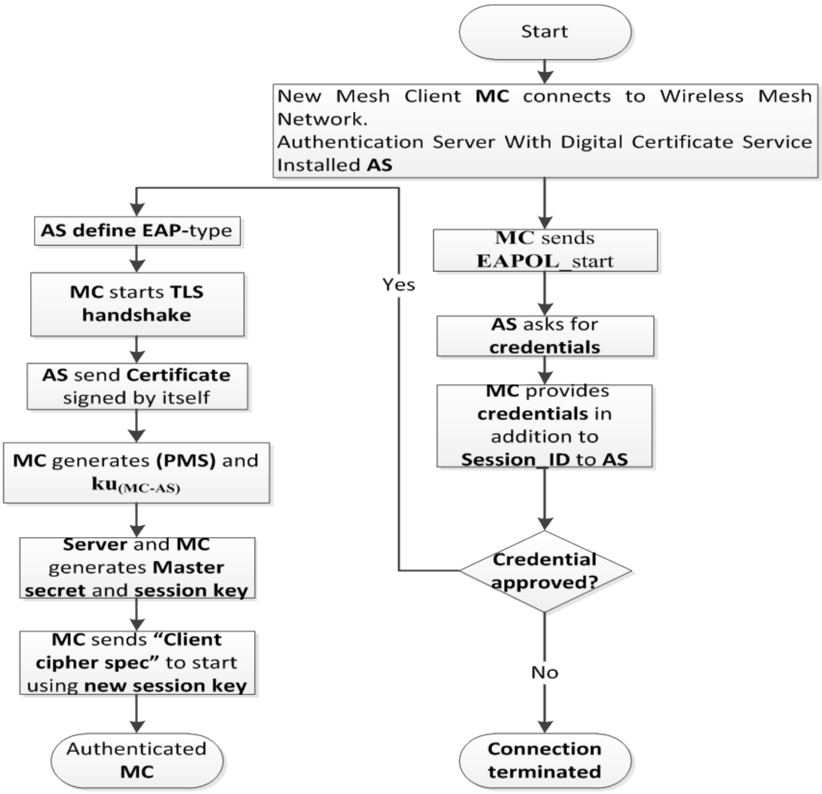
Start
New Mesh Client MC connects to Wireless Mesh Network.
Authentication Server With Digital Certificate Service Installed AS
MC sends EAPOL_start
AS asks for credentials
MC provides credentials in addition to Session_ID to AS
Credential approved?
No
Connection terminated
Yes
AS define EAP-type
MC starts TLS handshake
AS send Certificate signed by itself
MC generates (PMS) and ku(MC-AS)
Server and MC generates Master secret and session key
MC sends "Client cipher spec" to start using new session key
Authenticated MC

Figure 4.8 Flow chart depicts MC Phase

4.3.5 Client-to-Client authentication

For Client–to-Client Authentication, our proposed model uses EAP authentication with a modified version of a scheme known as a four-pass Kerberos protocol.

Whenever an MC wants to establish a secure connection with another MC it approaches the AS and follows the protocol as following steps:

1. the first Client MC1, sends its identifier and the identifier of destination client MC2 to the AS, which in turn looks up both MCs in its database in order to verify the validity of both clients.

$$\text{MC1} \rightarrow \text{AS: } \{ \text{ID}_{\mathbf{MC1}} \,||\, \text{ID}_{\mathbf{MC2}} \mathbf{\}} \, \mathbf{ku}_{\mathbf{(MC1\text{-}AS)}} || \, \text{N}_{\mathbf{once}}$$

2. AS sends ***ticket***$_{\mathbf{MC2}}$ which contains K$_{\mathbf{MC12}}$ and the lifetime of that key, this ticket will be sent to MC1 along with the ***Authenticating_Data*** which provides MC1 with the shared key and proof that this is the right shared key to use with MC2 at this time.

$$\text{AS} \longrightarrow \text{MC1: } \textbf{\textit{ticket}}_{\mathbf{MC2}} \,\|\, \text{ID}_{\mathbf{MC1}} \,\|\, \{\text{K}_{\mathbf{MC12}}, \text{lifetime, N}_{\mathbf{once}}, \text{ID}_{\mathbf{MC2}}\}\ \mathbf{ku}_{\mathbf{(MC1\text{-}AS)}}$$

3. MC1 decrypts ***Authenticating_Data*** in order to validate its information and creates a new message; this message contains both identifiers in addition to ***ticket***$_{\mathbf{MC2}}$

$$\text{MC1} \longrightarrow \text{MC2: } \textbf{\textit{ticket}}_{\mathbf{MC2}} \,\|\, \{\text{ID}_{\mathbf{MC1}} \,\|\, \text{ID}_{\mathbf{MC2}}\}\ \mathbf{k}_{\mathbf{MC12}} \| \text{ N}_{\mathbf{once}}$$

4. MC2 decrypts ***ticket***$_{\mathbf{MC2}}$ with **K**$_{\mathbf{MC2}}$ to obtain **K**$_{\mathbf{MC12}}$, MC2 generates a new session key and encrypt it with **K**$_{\mathbf{MC12}}$. And sends it to MC1

$$\text{MC2} \longrightarrow \text{MC1: } \{\mathbf{K}_{\mathbf{MC1}}\text{-MC2}\}\ \text{K}_{\mathbf{MC12}}$$

In figure 4.9 we can see the steps required to connect two MCs in a WMN.

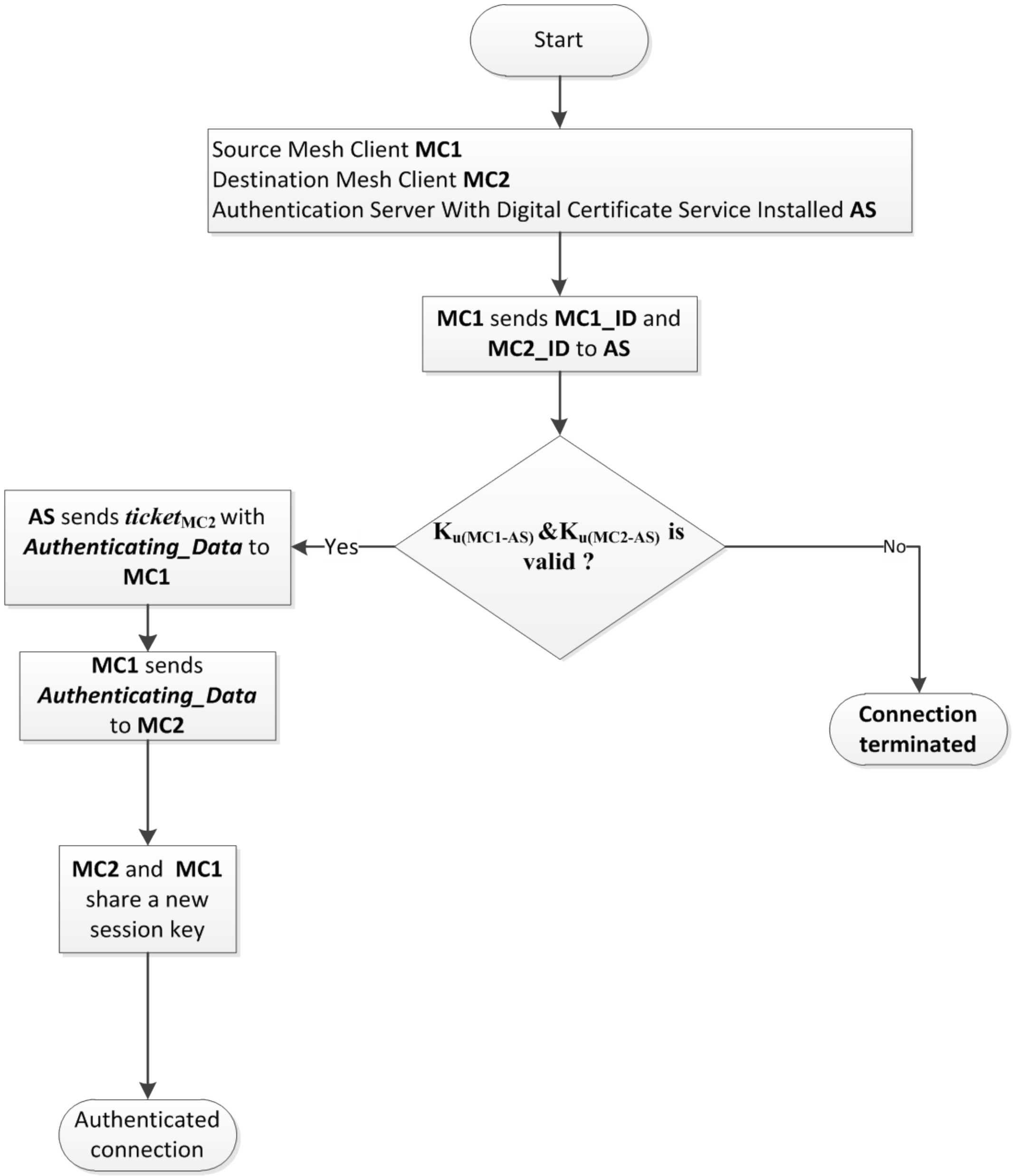
Start
Source Mesh Client **MC1**
Destination Mesh Client **MC2**
Authentication Server With Digital Certificate Service Installed **AS**
MC1 sends **MC1_ID** and **MC2_ID** to **AS**
$K_{u(MC1-AS)}$ & $K_{u(MC2-AS)}$ is valid ?
Yes
No
AS sends ***ticket***$_{MC2}$ with ***Authenticating_Data*** to **MC1**
MC1 sends ***Authenticating_Data*** to **MC2**
MC2 and **MC1** share a new session key
Authenticated connection
Connection terminated

Figure 4.9 Flow chart depicts MC-MC Authentication

4.4 TAWMP Protection Integrity

TAWMP provides a secret session key between every communicated MAPs, this session key provides **node-to-node** authentication and secure links between MAPs. In addition, TAWMP provides a secret session key between communicated MCs in order to provide **end-to-end** protection. The integration protection between **end-to-end** and **node-to-node** protects WMN from different several attacks as ones we described in section 4.2.

MC Phase uses EAP-TTLS authentication that provides protection for the origin source and destination address by encapsulating under Encapsulating Security Payload (ESP), EAP-TTLS includes the original IP header in the encapsulation process, and a new IP header is created for tunnel routing information. This method protects the mutable fields in HWMP and protects the origin message during routing in the mesh network. Figure 4.10 depicts EAP-TTLS authentication process.

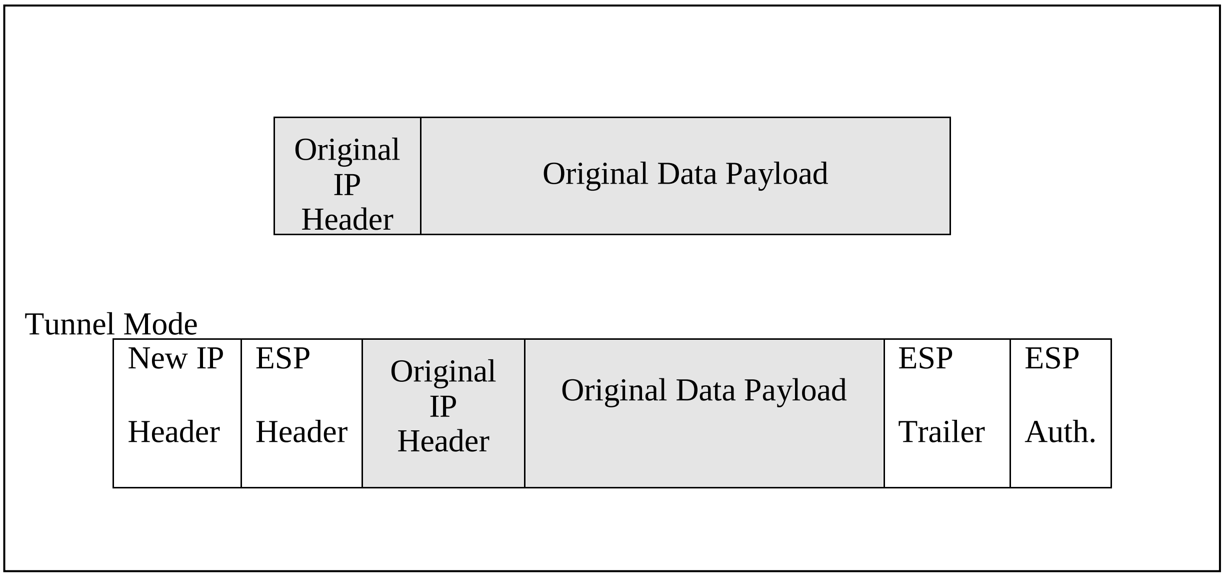

Figure 4.10 EAP-TTLS Packet Format.

As shown in figure 4.11, TAWMP uses a symmetric encryption in MAP Phase in order to provide Node-to-Node protection for backhaul network, while MC Phase uses symmetric encryption between source MC and destination one in order to provide End-to-End authentication.

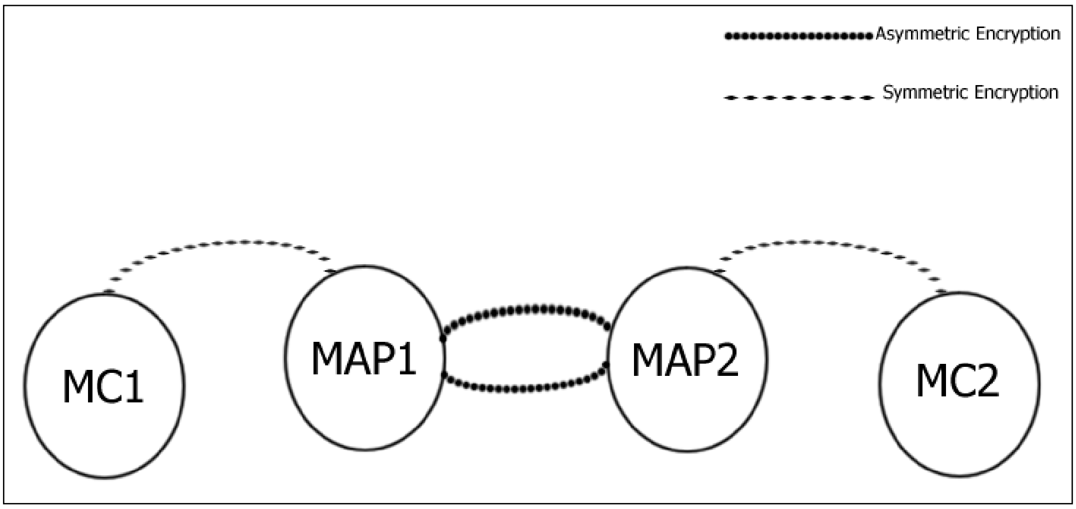

Figure 4.11 **Symmetric and Asymmetric encryption in TAWMP.**

When the source MC sends a packet to the destination MC through two MAPs, it will encrypt the packet using EAP-TTLS and encapsulate the packet by adding a new header, then the source MC sends this encapsulated message to the neighbor MAP which in turn encrypt the message using Asymmetric encryption.

4.5 Security Analysis

Preventing Flooding: In the proposed TAWMP, a node can participate in the route discovery process only if it has successfully establishes a secure link with the server and pass through steps of key distribution mechanism of TAWMP in case of MC device or MAP device, Thus it will not be possible for a malicious node to initiate a route discovery process with a destination address that is not in the network. Again, as the PREQ message is encrypted during transmission, a malicious node cannot insert new destination address.

2) Preventing Route Disruption: This type of attack is caused by the malicious behavior of a node through modification of a mutable field and dropping routing information elements. Note that, in our proposed scheme only authenticated nodes can participate in the route discovery phase. Moreover, routing information elements are authenticated and verified per hop. So, it is not possible to launch a route disruption attack in TAWMP.

3) Preventing Route Diversion: The root cause of route diversion attack is the modification of mutable fields in routing messages. These mutable fields are authenticated in each hop. If any malicious node modifies the value of a field in transit, it will be readily detected by the next hop while comparing the new MAC with the received one. It will find a miss-match in comparing the message authentication code (MAC) and the modified packet will be discarded.

4) Avoiding Routing Loops: Formation of routing loops requires gaining information regarding network topology, spoofing and alteration of routing message. As all the routing information is encrypted between nodes, an adversary will be unable to learn network topology by overhearing routing messages. Spoofing will not benefit the adversary as it will require authentication and key establishment to transmit a message with spoofed MAC. Moreover, fabrication of routing messages is detected by integrity check. So, proposed mechanism ensures that routing loops cannot be formed.

4.6 Complexity

In this section we analyze the complexity of TAWMP against network attacks demolish the secure protocol. TAWMP depends on user to provide certain credentials to access the network. So user password is the first line of defense which impose to be immune against cracks, according to [57] who reported the panel discussion at RSA 2005 conference, "*Password will be with us forever*", because "*We've got to make security simpler to use if it's going to be effective*", [58] mentioned that long and complex passwords makes it less likely that attackers will guess or crack them, but it also makes the passwords harder for users to remember, and thus more likely to be stored insecurely which increase the possibility that users will store their passwords insecurely and expose them to attackers. [58] defines the quality of a password by "how different it is from the dictionary words, how long it is, and how big the password character set is". We can define ten character set to generate a password as sown in table 4-1, with [a-z] we denote all the lower case alphabet letters, and with [A-Z] all the upper case alphabet letters. Considered symbols are ! " # $ % & ' * + , - . / : ; \ < = > ? @ ( ) [ ] ˆ _ { }| ~

We notice that the number of passwords equals the number of characters (N) in the character set raised to the length of password (L)

$$\textbf{\# of possible passwords} = N^L \qquad \textbf{(4.14)}$$

This formula shows that increasing the length of a password has a greater effect than increasing the number of possible characters

Table 4.1 Character Set to generate passwords

Character Set #	Set	Chars
1	Digits	10
2	[a-z] or [A-Z]	26
3	Symbols	32
4	Digits + ([a-z] or [A-Z])	36
5	Symbols + Digits	42
6	[a-z] + [A-Z]	52
7	Symbols + ([a-z] or [A-Z])	58
8	Digits + ([a-z] + [A-Z])	62
9	Symbols + ([a-z] + [A-Z])	84
10	Digits + Symbols +([a-z] + [A-Z])	94

One of major attacks on password based authentication is brute force attack, in which every possible must be tried until the password is reviled, while long complex password make it harder to brute force attack to expose the password, it’s so important to change the password every 2-3 months.

After valid credentials have been provided an EAP-TTLS authentication process took place, we have mentioned in table 3.4 that EAP-TTLS is vulnerable to Main-in-the-Middle Attack, EAP-TTLS is a transformation of EAP-TLS.

According to [60] whenever the server is authenticated, the channel is secure against man-in-the-middle attacks, but completely anonymous sessions are inherently vulnerable to such attacks. [61] mentioned that tunneling approach is vulnerable to Man-in-the-Middle attack in case the legacy client authentication protocol is used in other environments, e.g., plain EAP without any tunneling, or without any encapsulation at all.

TAWMP specifies in step 7 in MC Phase from this chapter that Authentication Server (AS) sends a self-signed certificate to the connecting MC, so if a legacy client tries to connect it fails and connection terminate at this step. By this way TAWMP is protected against Man-in-the-Middle attack.

In MAP phase, we have kept in mind several attacks threats during designing stage, at the first step MAP1 sends to AS a massage using hashed value of MAC address to guarantee integrity of the message.

Since the AS already has the MAC address of specific IDMAP, so the MAP will not send MAC within the message, by doing so we guarantee the message safety during transmission.

In response, AS reply with Ku(AS) encrypted using hashed of its MAC address and its ID to guarantee confidentiality and integrity during transmission time.

At the end of MAP phase, connected MAP has private and public keys, these keys will be valid for a period of time (lifetime/expiration time) then keys will be revoked and regenerated again in order to protect keys from brute force attacks. Lifetime calculations are out of scope of this thesis.

In MAP-MAP phase and MC-MC phase, the connection will be secured using keys between MAPs and AS. TAWMP provides **forward secrecy**, which means that if either node's key are compromised, the past sessions keys remain secure. This will keep connection secure even if one of the keys revealed.

Chapter Five

Simulations and Results

Table of Contents

5.1 Introduction

Due to the difficulty of providing a realistic environment to evaluate our protocol, we can use network simulators as an alternative solution. These simulators are considered as a prominent and effective tool for analyzing and evaluating wide variety of researches problems that require considerable network infrastructure to perform huge number of experiments. However, the simulation tools have a significant role in evaluating proposed network protocols.

In this chapter, we use the Network Simulator version 2 (NS-2) [62] to conduct our simulation and evaluate the TAWMP protocol based on some performance metrics such as packet delivery ratio, network throughput, protocol overhead and the average end to end delay and then compare our results with HWMP protocols.

5.2 Network Simulation

Network simulators are now widely used in a variety of civilian and military applications to evaluate the performance of network infrastructure or the implemented mechanisms to secure routing protocols. There are many network simulators such as NS-2, J-sim [64] and OMNeT++ [65].

In our thesis, we use NS-2 to perform network simulation because it’s widely deployed network simulator for network protocols evaluation. It supports the AODV and OLSR protocol, Also, its free source for researchers, according to [66] NS-2 has less time consumption in implementation over other wireless network simulators and include more routing protocol such as DSR. In order to generate a reasonable router-level representation of the internal AS in the Internet, we create our topology by Georgia Tech Internetwork

Topology Models (GT-ITM) [63] topology generator to generate the topology of the 802.11 based WMNs here exists three types of nodes, MAPs, MPs and STAs.

Without routing functionality, STAs should associate with MAPs to obtain network access. MAPs provide connectivity to STAs and relay traffic together with MPs among STAs or between STAs and other networks. This means, the topology of WMNs is hierarchical. The generated topology in a 1000*m*×1000*m* area consists of 30 MAPs, 20 MPs, and 300 STAs which are randomly distributed in the area.

MAPs are equipped with two wireless interfaces, one access interface and one relay interface. MPs are equipped with just one relay interface. STAs are equipped with one interface to connect with MAPs. One of the MAPs is connected to the Internet through wired cable. The maximum transmission range of all the nodes is 150m. In our rate adaptation strategy, we use a simple wireless channel model in which the data transmission rate depends only on the distance between transmitters and receivers. Specifically, the data rate is 1Mbps.

Our simulation is implemented with network simulator NS-2 version 2.34, running on a PC with the following main specifications: OS: Linux Ubuntu 10, CPU: Intel Core- 2 Duo 2.2GHz and RAM: 2GB. In our simulation, we choose the AODV as routing protocol between MAPs and OLSR between MCs. The network simulation parameters for our simulation are summarized in Table 4. In addition, each node implements the TAWMP protocol to detect malicious nodes in the network.

So, according to the malicious nodes ratio in the network, we can evaluate the network performance based on some performance metrics as shown in the next section. On the other hand, to measure the running time of MAP phase authentication with 195 bit key size for MC keys and 256 bit key size for MAP key, we run TAWMP with 1500 Bytes on our PC.

In order to ensure the rationality, each simulation experiment is repeated 15 times and their average is used to calculate the simulation results. However, we use the Tool Command Language (TCL) script files to configure NS-2 simulator and setup the parameters for our simulation experiments.

Also, the simulation results can be resolved from the *trace files* produced by the NS-2 simulator. These trace files contains a huge number of lines. So, it's nontrivial to deal with these files directly. Instead, we can use other tools such as the Linux command "grep" to extract the intended information from trace files. This technique minimizes

The size of original trace files and makes them more convenient to read and analyze. In addition, we use the "awk" files to analyze trace files and get the required results and graphs based on network performance metrics. The overall steps of simulation model are shown in figure 5.1.

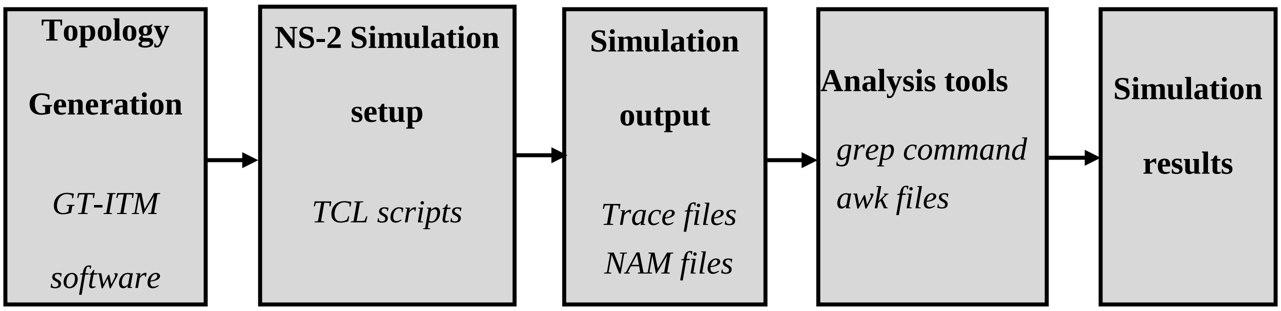

Figure 5.1 The overall steps of the simulation model

Note that "nam" files are not used in the analysis. They are only used to visualize the network traffic.

Table 5.1. Network Simulation Parameter

Parameters	**Value**
Reactive Routing Protocol	AODV
Proactive Routing Protocol	OLSR
Number of MAPs	30
Number of MPs	20
Number of STA	300
Data packet size	1500 Byte
Data rate	1Mbps.
Queue type	DropTail/PriQueue
Traffic type	Constant Bit Rate (CBR)
Simulation time	1000 s

5.3 Performance Evaluation Metrics

To evaluate the performance of the network in the presence of malicious nodes and compare it with the performance of our protocol and other proposed protocols, we can use some performance metrics such as packet delivery ratio, throughput, protocol overhead and average end-to-end delay. This section provides an overview of these metrics.

5.3.1 Packet Delivery Ratio

Packet Delivery Ratio (PDR) is the ratio of the number of data packets received at the destinations to the number of data packets generated by the CBR sources. It in turn determines the efficiency of the protocol to discover routes successfully Mathematically, PDR can be expressed as:

$$PRD = \frac{\sum_{i=1}^{N} r_i}{\sum_{i=1}^{N} s_i} \tag{5.1}$$

Where r_i is the received data packet, s_i is the sent data packet and N is the number of packets.

This metric measures the reliability, effectiveness and efficiency of protocols. Therefore, the protocol that has better PDR is considered more reliable.

5.3.2 Throughput

Throughput (TH) is the total size of data packets delivered to the destinations divided by the time interval. The throughput is usually measured in bits per second (bit/s or bps). Mathematically, TH can be expressed as:

$$\mathrm{TH} = \frac{\sum_{i=1}^{N} r_i \times p_s}{\mathrm{T}} \tag{5.2}$$

Where r_i is the received data packet, $p_s$is the packet size and T is the time interval.

This metric measures the speed or actual data rate in the channel and indicates the effectiveness of a protocol.

5.3.3 Protocol Overhead

Communication overhead or protocol overhead (OH) is defined by the number of routing messages required to establish a secure path between sender and receiver. Mathematically, OH can be expressed as:

$$OH = \cos t\,\alpha + \boldsymbol{n} \times \cos t\,\beta \qquad (5.3)$$

Where $\cos t\,\alpha$ the number of routing messages sent in MC phase, $\cos t\,\beta$ the number of routing messages sent in MAP phase, and $\boldsymbol{n}$ the number of MAPs in the secure path.

This metric reflects how much cost will be added by the protocol over available bandwidth. Therefore, it’s desired to keep the OH as low as possible to increase the network throughput.

However, the control packets added by our protocol are the *RREQ* and *RREP* packets which are used for the route authentication process.

5.3.4 Average End-to-End Delay

Average End-to- End Delay (E2ED) is the average delay for data packets to traverse from the source nodes to the destination nodes. This includes all possible delays caused by the sender buffer, the delay in the interface queue, the delay of links between routers in the route and the hardware latency. In general, this metric measures the total delay time from a source to a destination. Mathematically, E2ED can be expressed as:

$$E2ED = \frac{\sum_{i=1}^{N} Tr_i - Ts_i}{\sum_{i=1}^{N} r_i} \qquad (5.4)$$

Where Tr_i is the received time, Ts_i is the sent time and r_i is the received data packet.

5.4 Simulation Results

In our simulation, we create several network environments and investigate the performance metrics (PDR, TH, E2ED and OH) as a function of Malicious Nodes Ratio **(MNR)** which is the total number of malicious nodes divided by the total number of nodes in the network. Mathematically, E2ED can be expressed as:

$$MNR = \frac{\sum_{i=1}^{M} \ m_i}{\sum_{i=1}^{N} \ n_i} \tag{5.5}$$

Where: n is a network node, m is a malicious node, N is the total number of nodes and M is the total number of malicious nodes.

This metric can be used to measures the effect of malicious nodes on the network performance, in particular, the effect on PDR, TH and E2ED.

5.4.1 Packet delivery ratio

As shown in Figure 5.2, when the network is free from malicious nodes, the PDR achieves its highest value which is rather stable at 0.96. While, the PDR decreases from 1 to 0.96 due to packet dropping resulting from normal network congestion and buffers overflow. When the malicious nodes appear in the network, the PDR starts decreasing and fluctuating. When the network has 5% malicious nodes, the PDR decreases to the mean value 0.93 and fluctuates between (0.941 - 0.922). When the network has 10% malicious nodes, the PDR decreases to the mean value 0.835 and fluctuates between (0.806 - 0.852).

We noticed in figure 5.2 that the malicious nodes can strongly affect the PDR. Besides, as the number of malicious nodes increases the PDR decreases and the network becomes less stable. This is due to malicious nodes behavior, whereas, each malicious node floods network with PREQ causing traffic congestion and less-bandwidth and more collisions. Therefore, the packet dropping will increase because the links buffers will overflow.

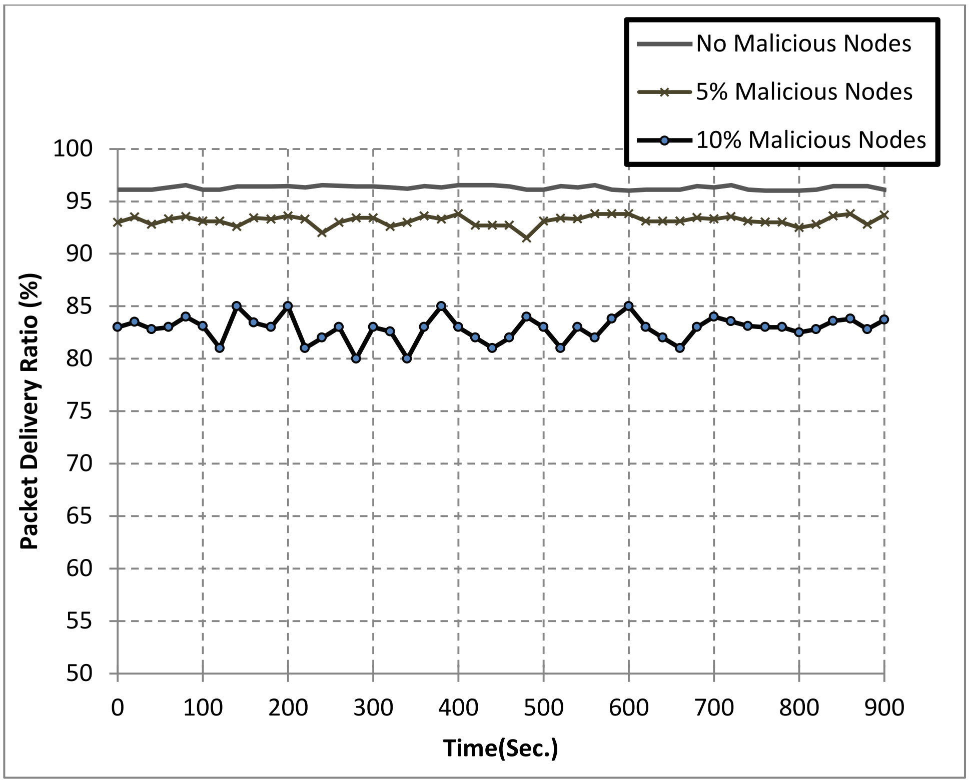

Figure 5.2 PDR as a function of time with the impact of malicious nodes in the network

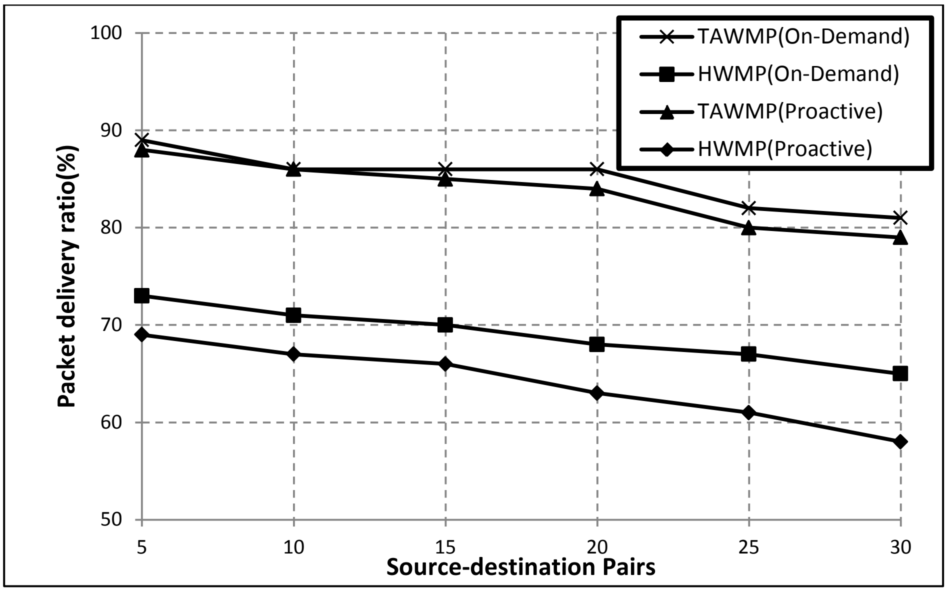

Figure 5.3 PDR as a function of time with the impact of increasing number of communicating pairs in the wireless mesh network.

On the other hand, figure 5.3 shows the impact of malicious nodes on the PDR in different environments. This figure compares the average PDR with number of connected pairs in WMN contains 10% malicious nodes in two cases: the first case shows the impact of malicious nodes over PDR in the network running HWMP while; the second case shows the PDR after applying the TAWMP protocol. In both cases, the impact of malicious nodes was investigated with different communicated nodes ranging from 5 to 30 pair. In this test we run HWMP and TWMP in proactive and reactive routing protocol.

In the first case, one can notice that as the communicated nodes increase the PDR decreases. For example, when the network includes 5 communicated pairs, the average PDR decreases by 28.7%. Also, when the network includes 10 communicated pairs, the average PDR decreases by 30.2%. However, the PDR remains decreasing until 30 communicated pairs. At this point the PDR becomes 65.4%.

In the second case, keeping the same conditions and simulation environments, it’s noticed that the average PDR is remarkably improved when applying the TAWMP protocol. By comparing with HWMP when the network includes 5 communicated pairs. The average PDR increases by 14.8%. Also when the network includes 10 communicated pairs, the average PDR increases by 18.5%. Moreover, when increasing the communicating pairs, the improvement caused by TAWMP increases since the flooding packets dropped and the decreases in PDR just caused from normal network congestion and buffers overflow.

5.4.2 Throughput

The impact of malicious nodes on the network throughput is shown in figure 5.4. As illustrated in the PDR cases, this figure compares the average TH with 10% malicious nodes in the network in two cases: the first case shows the impact of malicious nodes over TH in the HWMP protocol. While, the second case shows the TH after applying TAWMP protocol. In both cases, the impact of malicious nodes was investigated at different number of communicated pairs ranging from 5 to 30 pair.

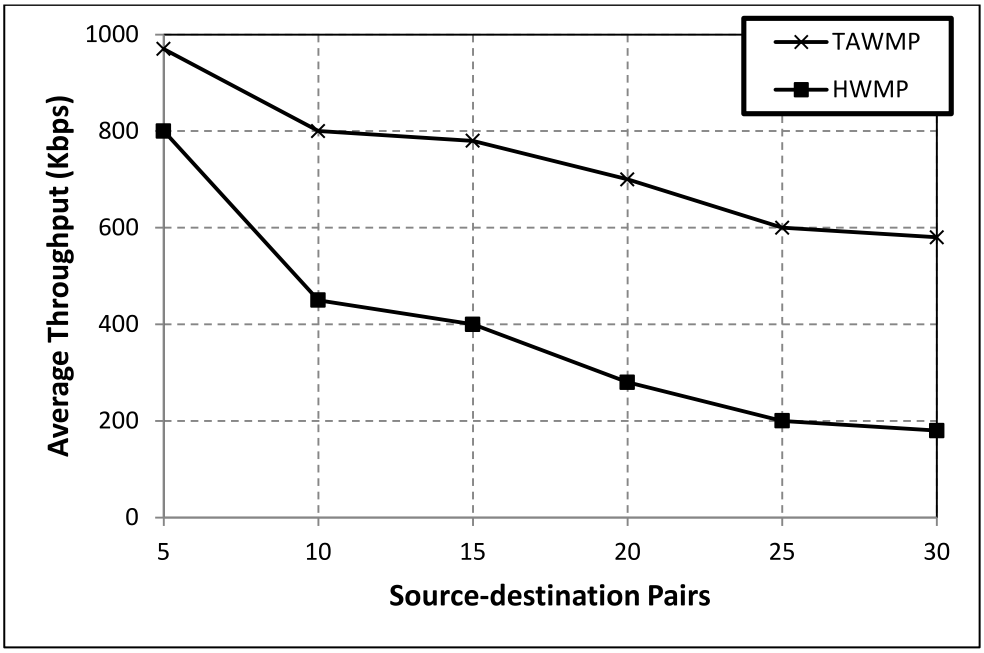

Figure 5.4 The impact of malicious nodes ratio on Average throughput in different environments with 10% of malicious nodes in the wireless mesh network.

When the network has 5 communicated pairs, the TH achieves its highest value which equals to 800 Kbps. (Note, the data rate in our simulation is set to 1Mbps for each connection) While, the decreases from 1Mbps to 800 kbps caused by normal network congestion which leads to decreasing of connections data rate. When the communicated pairs increase in the network, the TH starts decreasing. For example when the network includes 10 communicated pairs, the TH decreases by 60.3% Also, when the network includes 30 communicated pairs, the TH decreases to 80.1% Whereas, the average PDR decreases by 30.2% when the network includes 10 communicated pairs and 65.4% when the network includes 30 communicated pairs. It’s clear that the malicious nodes can strongly affect the network TH and PDR, but their impact on TH is greater than that on PDR since the delay time of received packets will increase as the malicious nodes flood the network with PREQ packets leading to high traffic and to congested links.

Now, when applying the TAWMP protocol and keeping the same conditions and simulation environment, it’s noticed that the TH is improved remarkably. For example, by comparing with HWMP case, the TH increases by 37.4% when the network includes 10 communicated pairs. Also, the TH increases by 38.5% when the network includes 30 communicated pairs.

5.4.3 End-to-End delay

According to table 5.1 and table 5.2 the time required to authenticate a MAP is 42.73µs while the time required establishing a secure connection between 2 MAPs is 38.68µs. On the other hand the time required to authenticate MC is 36.71µs while the time required establishing a secure connection between 2 MCs is 54.26µs. on the other hand, table 5.3 and table 5.4 shows authentication and encryption time in HWMP which is less than TAWMP since there is a delay caused by adding AS and the process of authentication in TAWMP which longer than HWMP. According to equation 5.3 the authentication is depended on the number of between MAPs in the routing path.

Table 5.2. Authentication time in TAWMP

	(MAP-AS)	(MC-AS)	(MAP-MAP)	(MC-MC)
Authentication time (µs)	42.73	36.71	38.68	54.26

Table 5.3. Encryption time in TAWMP

	MAP	MC
Encryption time (µs)	29.66	35.88

Table 5.4. Authentication time in HWMP

	MAP	MC
Authentication time (µs)	32.31	28.45

Table 5.5. Encryption time in HWMP

	MAP	MC
Encryption time (µs)	21.65	18.23

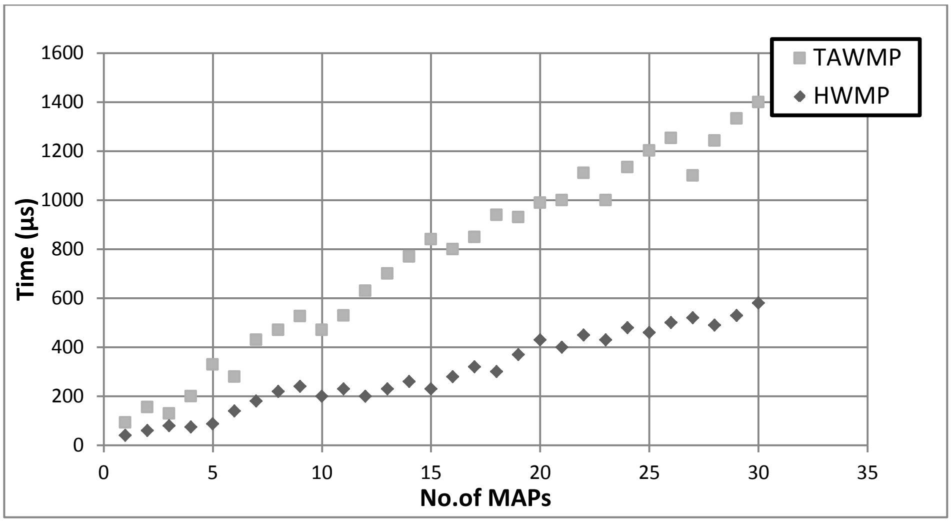

Figure 5.5 Client Authentication time as a function of No. of MAPs

As shown in figure 5.5, client authentication time required for mesh client in network that run TAWMP will be longer than time required in the same network runs HWMP. We also recognize from figure 5.5 that the authentication time in TAWMP increases rapidly and proportional with number of MAPs since the authentication and encryption time for MAPs is higher than HWMP. On the other hand in figure 5.6 we see that encryption time in TAWMP is higher than HWMP since delivered message will be encrypted using EAP-TTLS in mesh client and using asymmetric encryption in every passing MAP, while in HWMP the delivered message will be encrypted and decrypted in every single hop using symmetric keys which is faster than process used in TAWMP. We have used MATLAB in order to make a curve fitting to the simulated results in figures 5.5 and 5.6 and we got equation 5.3 to the End-to-End delay.

$$T_E = 2\,(T_{MAP})\,n + 2(T_{MC}) \qquad (5.6)$$

Where…

T_E is End-to-End delay,

T_{MC} is encryption time required by MC

n The number of MAPs the message passing through it

(T_{MAP}) is encryption time required by MAP

The figure below depicts the relationship between End-to-End delay and number of hops

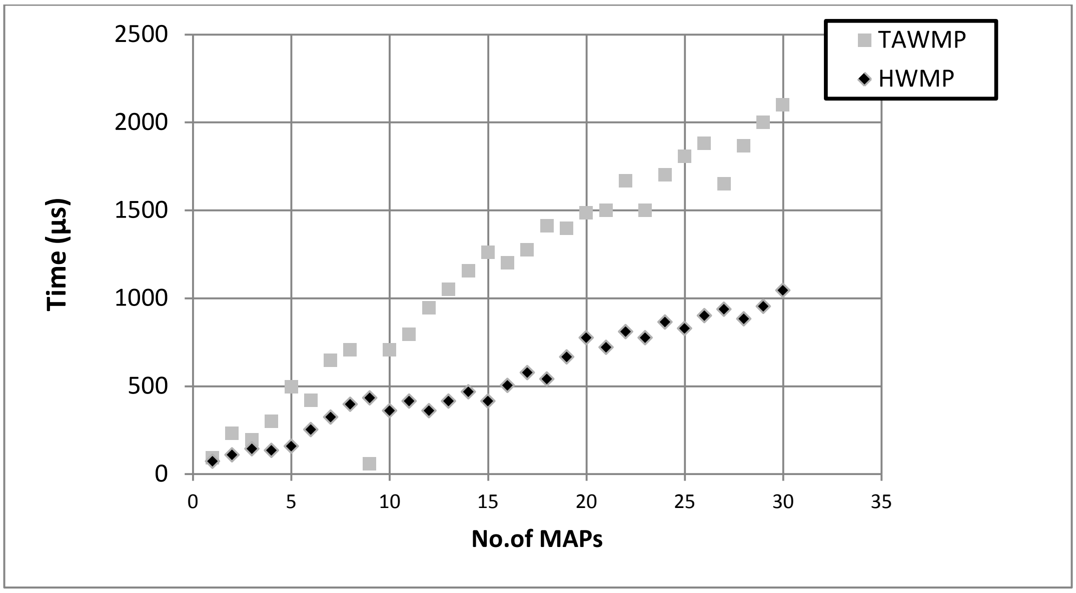

Figure 5.6 Message encryption time as a function of number of MAPs

5.4.4 Path Acquisition Delay

As shown in figure 5.7 when we run the test in two cases (HWMP and TAWMP) without interfering of malicious nodes we notice that path acquisition time in HWMP and TAWMP increase proportional to the number of nodes in the network since the time required to build tables in MAPs and to send PREQ and PREP will increase. On the other hand it’s clear in figure 5.7 that path acquisition time in TAWMP is higher than HWMP because, in addition to normal routing operation of HWMP, the proposed TAWMP scheme requires computing end-to-end encryption and decryption algorithms to verify the authenticity of a received packet, which require extra processing delay.

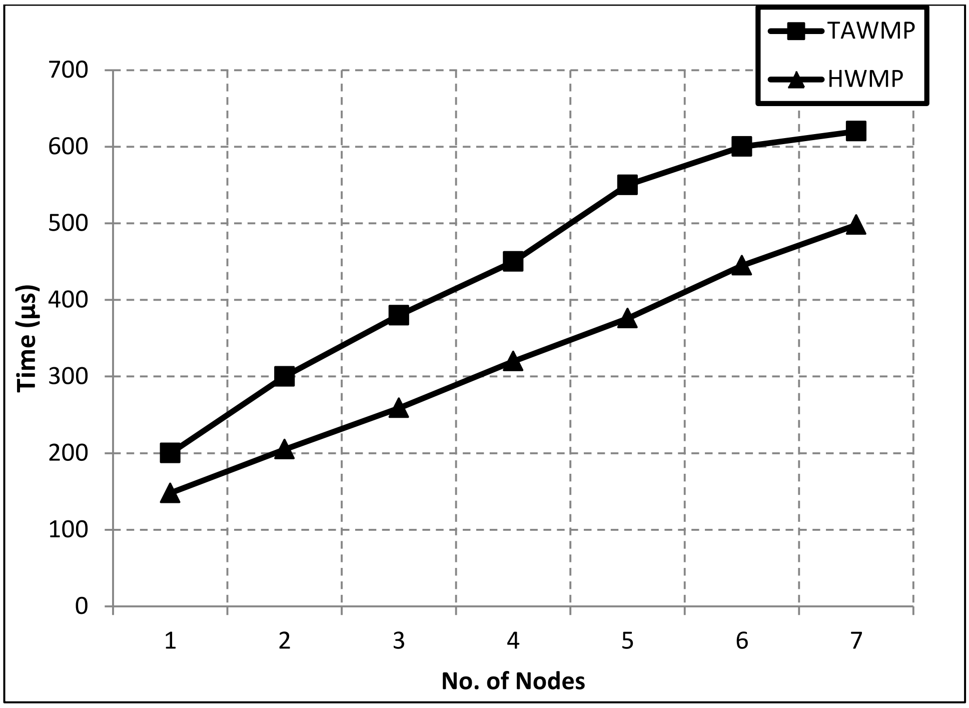

Figure 5.7 Path acquisition delay as function of NO. of nodes.

5.5 Conclusion

Security in WMNs in considered one of the top hot research subjects; it is well known that there is a tradeoff between the quality and performance of a secure algorithm. In this thesis a secure authenticating protocol for WMN has been developed to provide an end-to-end protection. Consequently we summarize a number of conclusions from this thesis:

1. WMN is a multi-hop network with hybrid routing protocol, WMN authenticating protocol (HWMP with nod-to-node authentication) is vulnerable to various kinds of security threats.
2. In this thesis we developed a secure authentication protocol (TAWMP) that provides end-to-end authentication in addition to node-to-node authentication.

3. TAWMP authentication process is divided in two phases to applicable with hybrid routing nature in WMN.
4. Ticket authentication process in TAWMP depends on Authentication Server which is a single point of failure.
5. Adding authentication server to the network and additional authentication process (end-to-end) increases delay times in network such as end-to-end time delay and path acquisition delay.
6. The additional security by TAWMP increases packet delivery ration and throughput.

5.6 Future Work

1. TAWMP depends on authentication server, which is single point of failure; more work is needed to overcome this issue.
2. TAWMP has a slight delay over HWMP which is need to be enhanced.
3. Handoff process has not been address in this work, more work need to be done to provide an integrated system.

References

1. Akyildiz, I.F. and X. Wang, Wireless mesh networks2009: Wiley.

2. Aggelou, G., Wireless Mesh Networking2008: McGraw-Hill.

3. Akyildiz, I.F. and W. Xudong, A survey on wireless mesh networks. Communications Magazine, IEEE, 2005. **43**(9): p. S23-S30.

4. Held, G., Wireless mesh networks2005: Auerbach Publications.

5. I. F. Akyildiz, X.W., and W. Wang, Wireless mesh networks: A survey, Computer Networks Journal,. March 2005. **vol. 47, no. 4,**: p. 445–487.

6. Misra, S., S.C. Misra, and I. Woungang, Guide to wireless mesh networks2008: Springer.

7. Arumugam, M., et al., Stabilizing Interference-Free Slot Assignment for Wireless Mesh Networks

8. Pirzada, A.A. and M. Portmann. High Performance AODV Routing Protocol for Hybrid Wireless Mesh Networks. in Mobile and Ubiquitous Systems: Networking & Services, 2007. MobiQuitous 2007. Fourth Annual International Conference on. 2007.

9. Kai, Y., M. Jian-feng, and M. Zi-hui. Hybrid Routing Protocol for Wireless Mesh Network. in Computational Intelligence and Security, 2009. CIS '09. International Conference on. 2009.

10. Cornils, M., M. Bahr, and T. Gamer. Simulative analysis of the Hybrid Wireless Mesh Protocol (HWMP). in Wireless Conference (EW), 2010 European. 2010.

11. Bahr, M. Update on the Hybrid Wireless Mesh Protocol of IEEE 802.11s. in Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on. 2007.

12. Minseok, O. A hybrid routing protocol for wireless Mesh Networks. in Broadband Multimedia Systems and Broadcasting, 2008 IEEE International Symposium on. 2008.

13. Muogilim, O.E., K.-K. Loo, and R. Comley, Wireless mesh network security: A traffic engineering management approach. Journal of Network and Computer Applications, 2011. **34**(2): p. 478-491.

14. Siddiqui, M.S., et al. MHRP: A Secure Multi-Path Hybrid Routing Protocol for Wireless Mesh Network. in Military Communications Conference, 2007. MILCOM 2007. IEEE. 2007.

15. Ping, Y., et al. Security in Wireless Mesh Networks: Challenges and Solutions. in Information Technology: New Generations, 2009. ITNG '09. Sixth International Conference on. 2009.

16. Hamid, M.A., M.S. Islam, and H. Choong Seon. Developing Security Solutions for Wireless Mesh Enterprise Networks. in Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE. 2008.

17. Redwan, H. and K. Ki-Hyung. Survey of Security Requirements, Attacks and Network Integration in Wireless Mesh Networks. in Frontier of Computer Science and Technology, 2008. FCST '08. Japan-China Joint Workshop on. 2008.

18. Hong, J., et al. Securing Wireless Mesh Network with Mobile Firewall. in Wireless Communications and Signal Processing (WCSP), 2010 International Conference on.

19. Shushan, Z., et al. A Secure Routing Protocol in Proactive Security Approach for Mobile Ad-Hoc Networks. in Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE. 2008.

20. Hiertz, G.R., et al., IEEE 802.11s: The WLAN Mesh Standard. Wireless Communications, IEEE, 2010. **17**(1): p. 104-111.

21. Changling Liu, J.K., A survey of mobile ad hoc networks routing protocols, 2005, University of Magdeburg.

22. Michael, B., Proposed routing for IEEE 802.11s WLAN mesh networks, in Proceedings of the 2nd annual international workshop on Wireless internet2006, ACM: Boston, Massachusetts.

23. http://www.ietf.org/rfc/rfc2501.txt. December 2011.

24. Waharte, S., et al., Routing protocols in wireless mesh networks: challenges and design considerations. Multimedia Tools and Applications, 2006. **29**(3): p. 285-303.

25. Zhang, Y., J. Luo, and H. Hu, Wireless mesh networking: architectures, protocols and standards2006: Auerbach Publications.

26. Abolhasan, M., T. Wysocki, and E. Dutkiewicz, A review of routing protocols for mobile ad hoc networks. Ad Hoc Networks, 2004. **2**(1): p. 1-22.

27. Perkins, C.E. and P. Bhagwat, Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. SIGCOMM Comput. Commun. Rev., 1994. **24**(4): p. 234-244.

28. Jacquet, P., et al. Optimized link state routing protocol for ad hoc networks. in Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International. 2001.

29. Ghannay, S., et al., Comparison of Proposed Path Selection Protocols for IEEE 802.11s WLAN Mesh Networks, Wireless and Mobile Networking, 2008, Springer Boston. p. 17-28.

30. Khushboo, T., P. Manjusha, and V. Shekhar, Comparison of reactive and proactive routing protocols for different mobility conditions in WSN, in Proceedings of the 2011 International Conference on Communication, Computing, and Security, ACM: Rourkela, Odisha, India.

31. Liqiang, Z., et al. A Hybrid Routing Protocol for Hierarchy Wireless Mesh Networks. in Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on.

32. Boukerche, A., et al., Routing protocols in ad hoc networks: A survey. Computer Networks, 2011. **55**(13): p. 3032-3080.

33. Nassereddine, B., A. Maach, and S. Bennani. The scalability of the hybrid protocol in wireless mesh network 802.11s. in Microwave Symposium (MMS), 2009 Mediterrannean. 2009.

34. Misra, S., I. Woungang, and S.C. Misra, Guide to Wireless Ad Hoc Networks2009: Springer.

35. Huang, S., D. MacCallum, and D.Z. Du, Network Security2010: Springer.

36. Johnson, D.B., D.A. Maltz, and J. Broch, DSR: the dynamic source routing protocol for multihop wireless ad hoc networks, in Ad hoc networking2001, Addison-Wesley Longman Publishing Co., Inc. p. 139-172.

37. Kumar, S. and J. Sengupta. AODV and OLSR routing protocols for Wireless Ad-hoc and Mesh Networks. in Computer and Communication Technology (ICCCT), 2010 International Conference on. 2010.

38. Yi, L., et al. Study of distance vector routing protocols for mobile ad hoc networks. in Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on. 2003.

39. Bertsekas, D.P. and R.G. Gallager, Data networks1992: Prentice Hall.

40. Kannhavong, B., et al., A survey of routing attacks in mobile ad hoc networks. Wireless Communications, IEEE, 2007. **14**(5): p. 85-91.

41. Zhong, H., B. Zhang, and J. Cheng. Implementation of Wireless Mesh Network Protocol Research Platform. in Communications and Mobile Computing (CMC), 2010 International Conference on. 2010.

42. Peppas, N. and D. Turgut. A Hybrid Routing Protocol in Wireless Mesh Networks. in Military Communications Conference, 2007. MILCOM 2007. IEEE. 2007.

43. Nichols, R.K. and P.C. Lekkas, Wireless security: models, threats, and solutions2002: McGraw-Hill.

44. Maiwald, E., Network security: a beginner's guide2003: McGraw-Hill/Osborne.

45. Bragg, R., M. Rhodes-Ousley, and K. Strassberg, Network security: the complete reference2004: McGraw-Hill/Osborne.

46. Seung-Jo, H., O. Heang-Soo, and P. Jongan. The improved data encryption standard (DES) algorithm. in Spread Spectrum Techniques and Applications Proceedings, 1996., IEEE 4th International Symposium on. 1996.

47. Chaouchi, H. and M. Laurent-Maknavicius, Wireless and Mobile Networks Security2009: ISTE.

48. Stallings, W., Cryptography and network security: principles and practice2010: Prentice Hall.

49. Zhang, Y., J. Zheng, and M. Ma, Handbook of Research on Wireless Security2008: IGI Global.

50. Sklavos, N. and X. Zhang, Wireless security and cryptography: specifications and implementations2007: CRC Press.

51. Agarwal, A.K., et al., An Experimental Study on Security Protocols in Wlans

Wireless Network Security, 2007, Springer US. p. 295-322.

52. Zhang, Y., J. Zheng, and H. Hu, Security in Wireless Mesh Networks2008: CRC Press.

53. Aboba, B. and D. Simon, PPP EAP TLS Authentication Protocol, 1999, IETF.

54. Simpson, W., PPP Challenge Handshake Authentication Protocol (CHAP), 1996, IETF.

55. Aboba, B., et al., Extensible Authentication Protocol (EAP). RFC 3748, 2004.

56. Bo, S., et al., Intrusion detection techniques in mobile ad hoc and wireless sensor networks. Wireless Communications, IEEE, 2007. **14**(5): p. 56-63.

57. Wanli, M., J. Campbell, et al. (2010). Password Entropy and Password Quality. Network and System Security (NSS), 2010 4th International Conference on.

58 National Institute of Standards and Technology., DRAFT Guide to Enterprise Password Management. 2009

59 Dell'Amico, M., P. Michiardi, et al. (2010). Password Strength: An Empirical Analysis. INFOCOM, 2010 Proceedings IEEE.

60 http://www.ietf.org/rfc/rfc2246.txt, December 2011.

61 N. Asokan, Valtteri Niemi, Kaisa Nyberg. (2002). Man-in-the-Middle in Tunnelled Authentication. Nokia Research Center, Finland

62 Garman, J. (2003). Kerberos: the definitive guide. O'Reilly

63 Jennifer G. Steiner , Clifford Neuman , Jeffrey I. Schiller. (1988). Kerberos: An Authentication Service for Open Network Systems. Usenix Conference Proceedings.

64. Guide to Wireless Mesh Networks, S. Misra, S.C. Misra, and I. Woungang, Editors. 2009, Springer London. p. 77-118.

65. Harn, L.; Hsin, W.-J.; Mehta, M.; , "Authenticated Diffie-Hellman key agreement protocol using a single cryptographic assumption," *Communications, IEE Proceedings-* , vol.152, no.4, pp. 404- 410, Aug 2005

66. Lessmann, J.; Janacik, P.; Lachev, L.; Orfanus, D.; , "Comparative Study of Wireless Network Simulators," *Networking, 2008. ICN 2008. Seventh International Conference on* , vol., no., pp.517-523, 13-18 April 2008

Appendices

J
O
R
D
A
N

The International Journal of ACM JORDAN

ISSN: 2078-7952 (print)

Date: Sep 10, 2011

Subject: Acceptance Letter

Dear Dr. Rushdi Hamamreh and Anas Melhem ,

We gladly inform you that your paper titled "SWMPT: Securing Wireless Mesh Networks Protocol Based on Ticket Authentication" was accepted for publication in the **International Journal of ACM JORDAN,** and will be published in **Volume II, Number IV** and thank you for your cooperation in performing all the changes as requested by the IJJ reviewers.

Thank you again for submitting your article to the **International Journal of ACM JORDAN.**

Sincerely Yours,

Dr. Amjad M daoud,

Editor-in Chief

The International Journal of ACM Jordan

SWMPT: Securing Wireless Mesh Networks Protocol Based on Ticket Authentication

Rushdi A. Hamamreh
Computer Engineering Department, Faculty of Engineering
Al-Quds University
Jerusalem, Palestine
rhamamreh@eng.alquds.edu

Anas M. Melhem
Computer Engineering Department, Faculty of Engineering
Palestine Technical University
Tulkarm, Palestine
amelhem@ptuk.edu.ps

ABSTRACT

Wireless mesh network (WMN) consists of two parts: mesh access points which are relatively static and energy-rich devices, and mesh clients which are relatively dynamic and power constrained. In this paper, we present a new model for WMN end-to-end security which divides authentication process into two phases: Mesh Access Point which is based on asymmetric cryptography and Mesh Client which is based on a server-side certificate such as EAP-TTLS.

General Terms

Algorithms, Performance, Design, Reliability, Experimentation, Security, Standardization, Theory.

Keywords

Hybrid mesh; network security; end-to-end authentication; server mobile; mobile router.

1. INTRODUCTION

Wireless mesh networks have appeared as a promising design model for next generation wireless networks which have grown rapidly due to recent developments such as easy installation and low setup cost when compared to wired networks [1]. WMN is a promising new technology which has been adopted as the wireless internetworking solution for the near future due to their self-healing, self-configuring and self-optimizing capabilities [2]. The most commercial form of WMN is called hybrid mesh networks [3], shown in Figure 1. Hybrid mesh networks contain mesh access points (MAP) and mesh clients (MC). MAPs are relatively static and energy-rich devices that have multiple wireless network interfaces. On the other hand, Mesh Clients are relatively mobile and power constrained devices such as notebook, Smartphone, and smart pad [4].

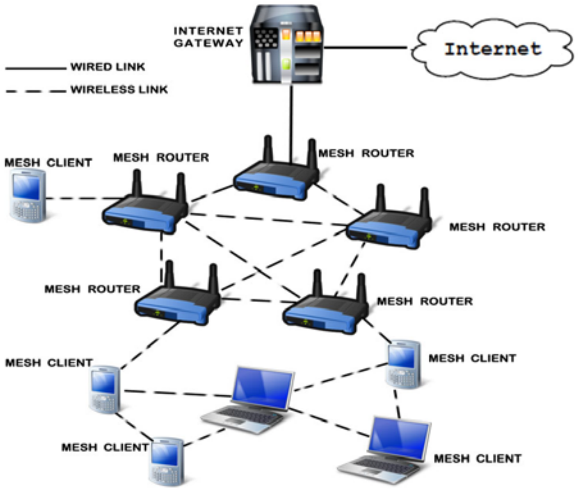

Figure 1. Hybrid Wireless Mesh Network.

The routing protocols used for WMNs can be classified into two types: Reactive Routing Protocols in which routes are established only when required and generally via flooding of Route Request packets in the network, and Proactive Routing Protocols in which routes are established before actual usage through periodical exchanges of connectivity information [5] [6] [7]. Both protocols have their individual advantages. Reactive protocols focus on minimizing control packet overhead such as Ad hoc On Demand Distance Vector (AODV) [8], Dynamic Source Routing (DSR) [9],Temporally-Ordered Routing Algorithm (TORA) [10] etc. while the proactive protocols attempt to minimize the route establishment delays such as OLSR [9], DSDV [10].

However, since these routing protocols have been designed for relatively homogenous MANETs, they will not provide optimum security for hybrid WMNs. An important security goal of a wireless mesh network is to protect the end-to-end communication between the device and its home network in general, and to protect the application content from being eavesdropped or modified during its transmission in particular.

Copyright ©2011 IJJ: The Research Bulletin of Jordan ACM - ISWSA; ISSN: 2078-7952 (print); 2078-7960 (online)

2. RELATED WORK

2.1 KAMAN

Please Kerberos Assisted Authentication in Mobile Ad-hoc Networks [11] uses multiple Kerberos servers for distributed authentication and load distribution. In Kaman only the users know the secret key or passwords and the servers know a cryptographic hash of these passwords. All Kaman servers share a secret key with each other server. In Kaman all servers periodically, or on-demand, replicate their databases with each other. Kaman uses an election based server selection mechanism.

2.2 TAODV

Ticket Based Ad-hoc On Demand Distance Vector [12] is a ticket-based security protocol foe WMNs that is based upon the AODV protocol, which is a cross layer protocol which works at network layer but also provides security for data exchange and avoids transfer of ARP messages for finding MAC addresses of source and destination.

2.3 Secure Extension to the OLSR protocol

Use The Secure Extension to the OLSR protocol [13] has only provided integrity and not confidentiality by signing each OLSR control packet with digital signature for authenticating the message. The digital signature is based on symmetric keys [14]. All OLSR control traffic is signed for every hop. This doesn't provide end-to-end signatures.

3. Our Proposed Model

Our proposed model aims to achieve an end-to-end authentication in WMN. In order to achieve such a goal we have divided the authentication process into two phases: the MAP phase in which a new MAP conducts the network, and the MC phase in which a new MC conducts the network.

At the MAP phase, we aim to use asymmetric cryptographic sine MAP is an energy rich device [14] on the other hand, MC devices in the second part of the authentication use server-side certificate such as EAP-TTLS and PEAP.

3.1 MAP Phase

When a MAP is connected to a WMN during setup stage, it has to do the following steps: (1) MAP sends its details including the type (1 for MAP / 0 for MC) and MAC address to an Authentication Server (AS). (2) AS will send key generation mechanism back to the MAP after checking MAC address in a stored list. (3) MAP will generate its public and secret keys, and then sends its public key ($PK_{\mathbf{MAP}}$) to the AS. Then AS generates a shared secret key ($K_{\mathbf{MAP}}$) for new MAP and AS on the basis of public key of MAP and its secret key by using Fixed Diffie-Hellman key exchange protocol. (4) AS generates a ticket for new MAP with required info (MAP ID, IP, issue time, expiration time) and sign it with its private key. Then, after signing, AS will encrypt that ticket with the shared secret key and then forward this encrypted ticket to new MAP. After receiving encrypted ticket, new MAP will first generate a shared secret key on the basis of AS's public key and its secret key (as AS generated) and then will decrypt the ticket. For future communication (route discovery request/reply) MAP will use this ticket.

(1) MAP → AS: Type|| MAC|| $N_{\mathbf{once}}$
(2) AS → MAP: key generation mechanism|| $N_{\mathbf{once}}$
(3) MAP → AS: $PK_{MAP} \| N_{once}$
(4) AS → MAP: $\{\textbf{\textit{ticket}}_{\mathbf{MAP\text{-}AS}}\}K_{MAP} \| N_{once}$

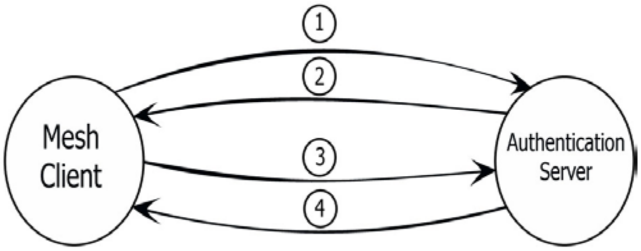

Figure 2. MAP Phase

3.2 MC Phase

When a new MC is connected to the WMN, it has to provide credentials to the AS. These credentials can be user-name/ID-number and password (via PAP, CHAP, or MD5 challenges) [15]. In this phase server-side certificate such as EAP-TTLS can be used. After successful authentication, the mobile node will receive a secret key that shares with the authentication server (AS).

3.3 MAP –to- MAP Authentication

As it has been mentioned, MAP depends on proactive protocols such as OLSR in order to build routing table through periodical exchanges of connectivity information, when a MAP discovers a new neighboring MAP, a secure route must be established. In order to do so, the first MAP sends both its identifier and the identifier of destination MAP to the AS, which in turn looks up both identifiers in its database in order to verify the validity of both clients.

MAP1 → AS: $\{ID_{\mathbf{MAP1},} ID_{\mathbf{MAP2}}\} K_{\mathbf{MAP1}} \| N_{\mathbf{once}}$

AS sends $\textbf{\textit{ticket}}_{\mathbf{MAP2}}$ along with the Authenticator $\{K_{\mathbf{MAP1}}, K_{\mathbf{MAP12}}, ID_{\mathbf{MAP1}}, T\}$ in which $K_{\mathbf{MAP12}}$ is the secret shared key between two MAPs and T is the lifetime of that key, this Authenticator provides MAP1 with the shared key and proof that this is the right shared key to use with MAP2 at this time.

AS → MAP1: $\textbf{\textit{ticket}}_{\mathbf{MAP2}} \| ID_{\mathbf{MAP1}} \| \{K_{\mathbf{MAP12}}, \text{times}, N_{\mathbf{once}}, ID_{\mathbf{MAP2}}\}$ KMAP1

MAP1 decrypts Authenticator in order to validate its information and then creates a new message with a fresh timestamp; this message contains both identifiers in addition to $\textbf{\textit{ticket}}_{\mathbf{MAP2}}$ and encrypted values that express MAP2 identifier with the fresh timestamp. And then send this message to MAP2.

MAP1 → MAP2: $\textbf{\textit{ticket}}_{\mathbf{MAP2}} \| ID_{\mathbf{MAP1}}, ID_{\mathbf{MAP2}} \|$timestamp

After receiving this message, MAP2 decrypts $\textbf{\textit{ticket}}_{\mathbf{MAP2}}$ with $K_{\mathbf{MAP2}}$ to obtain $K_{\mathbf{MAP12}}$ which in turn is used to get the encrypted values, and then MAP2 validates timestamp by comparing it to local time. In case the verification succeeds, MAP2 sends a new encrypted message with $K_{\mathbf{MAP12}}$, this message contains the timestamp sent before by MAP1 and a new key instead of $K_{\mathbf{MAP12}}$ called **subkey** used as a shared key between two clients in their communications. When the message received MAP1 decrypts it and verifies timestamp. If the verification succeeded, MAP1 knows that MAP2 has received the previous message

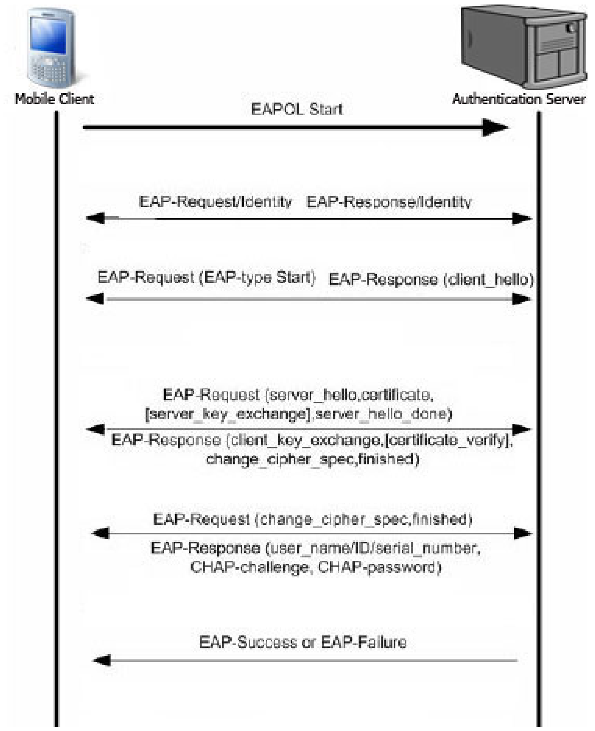
Mobile Client
Authentication Server
EAPOL Start
EAP-Request/Identity EAP-Response/Identity
EAP-Request (EAP-type Start) EAP-Response (client_hello)
EAP-Request (server_hello,certificate, [server_key_exchange],server_hello_done)
EAP-Response (client_key_exchange,[certificate_verify], change_cipher_spec,finished)
EAP-Request (change_cipher_spec,finished)
EAP-Response (user_name/ID/serial_number, CHAP-challenge, CHAP-password)
EAP-Success or EAP-Failure

Figure 3. MC Phase

a) MAP1 ➡ AS: {ID$_{MAP1}$, ID$_{MAP2}$ } K$_{MAP1}$||***ticket***$_{MAP1}$||N$_{once}$
b) AS ➡ MAP1: ***ticket***$_{MAP2}$ || ID$_{MAP1}$ || {K$_{MAP12}$, lifetime, N$_{once}$, ID$_{MAP2}$}K$_{MAP1}$
c) MAP1 ➡ MAP2: ***ticket***$_{MAP2}$ || ID$_{MAP1}$, ID$_{MAP2}$ ||timestamp
d) MAP2 ➡ MAP1: {timestamp, subkey}K$_{MAP12}$

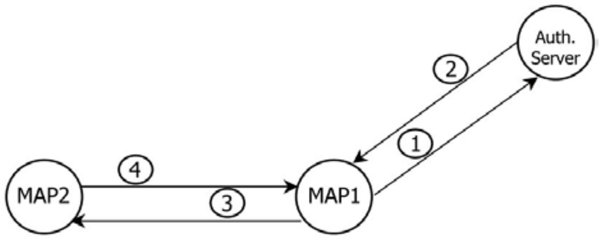

Figure 4. MAP-to-MAP Authentication

3.4 Client–to-Client Authentication

For Client–to-Client Authentication, our proposed model uses EAP authentication with a modified version of a scheme known as a four-pass Kerberos protocol [16][17].

When a new MC is connected to the WMN, it approves itself to the Authentication Server (AS) in order to get a secret key shared with the AS in addition to a unique identifier ID.

Whenever an MC wants to establish a secure connection with another MC, it approaches the AS and does the protocol as following steps:

The first Client MC1, sends both its identifier and the identifier of destination client MC2 to the AS which in turn searches for both MCs identifiers in its database in order to verify the validity of both clients.

MC1 ➡ AS: ID$_{MC1}$ || ID$_{MC2}$ || N$_{once}$

AS sends ***ticket***$_{MC2}$ which contains K$_{MC12}$ and the lifetime of that key, this ticket is sent to MC1 with the Authenticator which provides MC1 with the shared key and proof that this is the right shared key to use with MC2 at this time.

AS ➡ MC1: ***ticket***$_{MC2}$ || ID$_{MC1}$ || {K$_{MC12}$, lifetime, N$_{once}$, ID$_{MC2}$}K$_{MC1}$

MC1 decrypts Authenticator in order to validate its information. It then creates a new message with a fresh timestamp. This message contains both identifiers in addition to ***ticket***$_{MC2}$ and encrypted values that express MC2 identifier with the fresh timestamp. And then send this message to MC2.

MC1 ➡ MC2: ***ticket***$_{MC2}$ || Authenticator

After receiving this message, MC2 decrypts ***ticket***$_{MC2}$ with K$_{MC2}$ to obtain K$_{MC12}$ which in turn is used to get the encrypted values. Then MC2 validates timestamp and local time comparing the life time sent from MC1.In case the verification succeeds, MC2 sends a new encrypted message with K$_{MC12}$. This message contains both the **timestamp** sent before by MC1 and a new key called **subkey** instead of K$_{MC12}$ which is used as a shared key between the two clients in their communications. When the message is received, MC1 decrypts it and verifies timestamp. If the verification succeeds, then MC1 knows that MC2 has received the previous message in proper form and decrypt the shared key correctly.

MC2 ➡ MC1: {timestamp, subkey}K$_{MC12}$

a) MC1 ➡ AS: ID$_{MC1}$ || ID$_{MC2}$ || N$_{once}$
b) AS ➡ MC1: ***ticket***$_{MC2}$ || ID$_{MC1}$ || {K$_{MC12}$, lifetime, N$_{once}$, ID$_{MC2}$}K$_{MC1}$
c) MC1 ➡ MC2: ***ticket***$_{MC2}$ || Authenticator
d) MC2 ➡ MC1: {timestamp, subkey}K$_{MC12}$

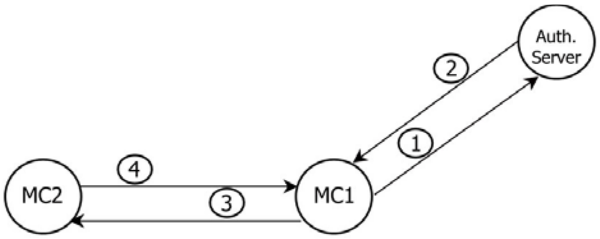

Figure 5. Client-to-Client authentication

We notice that all routes between MAP's are all secured through MAP-to-MAP authentication steps, so that when MC1 sends a message to MC2, this message is encrypted by the shared secret key **subkey** between every single MAP pair, and this provides both node–to–node and end-to-end security.

4. Simulation

We have used ns-2 simulator to simulate our proposed model (THWMP) protocol and to compare it with existing protocols HWMP and SHWMP[19]. We have simulated 50 static mesh nodes in a 1500 x1500 m2 area. We use 5 to 10 distinct source-destination pairs that are selected randomly. Traffic source are CBR (constant bit-rate). Each source sends data packets of 512 bytes at the rate of four packets per second during the simulation period of 900 seconds.

In order to compare HWMP with SHWMP, both protocols were run under identical traffic scenario. Both on-demand and proactive mode were simulated. We consider Packet delivery ratio and End-to-end delay as performance metrics.

As shown in Figure 7, the packet delivery ratio is better in SHWMP for both on demand and proactive mode than that of HWMP. We assume that 10% misbehaving nodes are present in the network. Since the misbehaving nodes participate in the route discovery process, in HWMP sometimes packets are intentionally dropped by the misbehaving nodes. But, in the proposed protocol, misbehaving nodes cannot participate in the route discovery process and thus always achieve a higher packet delivery ratio.

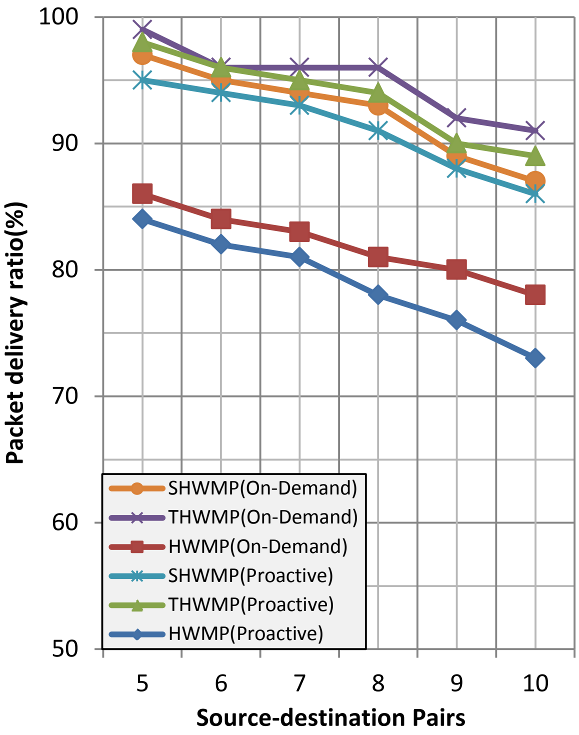

Figure 7. Packet delivery ratio

Figure 8 depicts that the average end-to-end delay of data packets for both protocols are almost equal. We run the simulation using 5 and 10 source-destination pairs, and as the traffic load increases, end-to-end delay also increases. It is also evident that the effect of route acquisition delay on average end-to-end delay is not significant.

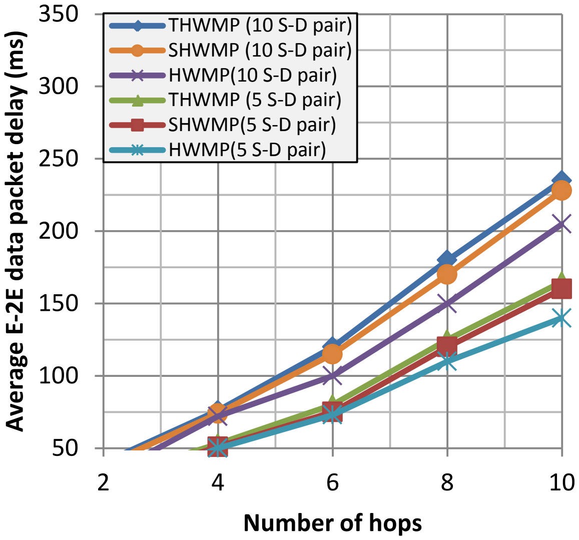

Figure 8. Control overhead

5. Conclusion

In this paper, we presented a new model for securing end-to-end wireless mesh network with ticked based-authentication. This model divides the authentication process into two phases: MAP phase and MC phase. In the first, our proposed model authenticates MAP using asymmetric cryptography [19] depending on MAP's MAC address. This phase ensures the securing of all network paths by establishing ticket based between every single MAP pair. Whereas in the second phase, the authentication process is done by proving the new MC to the AS using preconfigured credentials. This is required because the MC doesn't have any certificate yet. After that, the AS uses a server-side certificate to authenticate the MC. This is a secure method that saves MC battery. Our proposed model uses a modified version of a scheme known as four-pass Kerberos protocol in MAP-to-MAP authentication and MC-to-MC authentication. By doing this, we ensure the providing of a secure node-to-node routes for all routes in the network in addition to the end-to-end security message that cannot be decrypted without the secret key at the receiver MC with reasonable consuming to the battery at MC side. .

6. REFERENCES

[1] A. A. Pirzada, anad M. Portmann, 2007. High Performance AODV Routing Protocol for Hybrid Wireless Mesh Networks, *Proceedings of the 2007 Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking&Services (MobiQuitous)*, p.1-5, August 06-10, 2007.

[2] Stephan Miry, Asad Amir Pirzada and Marius Portmannz, 2008. HOVER: Hybrid On-demand Distance Vector Routing for Wireless Mesh Networks, *Proceedings of the thirty-first Australasian conference on Computer science* - Volume 74, Wollongong, Australia, 2008.

[3] I. F. Akylidiz, X. Wang and W. Wang, 2005. *Wireless Mesh Network: A Survey' in Computer Network ans ISDN Systems*, Volume 47, Issue 4, March 2005.

[4] Ping Yi; Tianhao Tong; Ning Liu; Yue Wu; Jianqing Ma; , Security in Wireless Mesh Networks: Challenges and Solutions, *Information Technology: New Generations*, 2009. ITNG '09. Sixth International Conference on , vol., no., pp.423-428, 27-29 April 2009.

[5] M. S. Azad, F. Anwar, M. A. Rahman, A. H. Abdalla, A. U. Priantoro, O. Mahmoud, 2006. *Performance Comparison of Proactive and Reactive Multicast Routing Protocols over Wireless Mesh Networks*, Vol. 9 No. 6 pp. 55-62, June 2006.

[6] A. jmal, M.M.; Mahmood, K.; Madani, S.A.; , 2010. Efficient routing in wireless mesh network by enhanced AODV, *Information and Emerging Technologies (ICIET), 2010 International Conference on* , vol., no., pp.1-7, 14-16 June 2010

[7] Mbarushimana, C.; Shahrabi, A.; , 2007. Comparative Study of Reactive and Proactive Routing Protocols Performance in Mobile Ad Hoc Networks, *Advanced Information Networking and Applications Workshops*, 2007, AINAW '07. 21st International Conference on , vol.2, no., pp.679-684, 21-23 May 2007

[8] C. Perkins, E. Belding-Royer and S. Das, 2003. Ad hoc On demand Distance Vector (AODV) Routing, *IETF RFC 3561*, July 2003.

[9] S. Hamma, E. Cizeron, H. Issaka, and J.-P. Guèdon, 2006 Performance Evaluation of Reactive and Proactive Routing Protocol in IEEE 802.11 Ad hoc Network. *in the proceedings of SPIE, Next-Generation Communication and Sensor Networks* 2006, Volume 6387, October 2006.

[10] J. Broch, D. A. Maltz, D. B. Johnson, Y –C. Hu, and J. Jetcheva, 1998. “A Performance Comprison of Multi-Hop Wireless Ad Hoc Network Routing Protocols” *in the proceedings of the Fourth Annual International Conference on Mobile Computing and Networking (MobiCom`98)*, Oct, 1998, pp: 85-97.

[11] A.A. Pirzada and C. McDonald, 2004. Kerberos Assisted Authentication in Mobile Ad Hoc Networks, *Proc. 27th Australasian Computer Science Conf. (ACSC)*, vol. 26, pp. 41-46, 2004.

[12] Qazi, S.; Yi Mu; Susilo, W.; , 2008. Securing wireless mesh networks with ticket-based authentication, *Signal Processing and Communication Systems, 2008. ICSPCS 2008. 2nd International Conference on* , vol., no., pp.1-10, 15-17 Dec. 2008.

[13] A. Hafslund, A. Tønnesen, J. Andersson, R. Rotvik, Ø Kure, 2004. Secure Extension to OLSR *Currently under review for the OLSR Interop and Workshop*, 2004.

[14] R. A. Hamamreh, and M. Farajallah, 2009. Design of a Robust Cryptosystem Algorithm for Non-Invertible Matrices Based on Hill Cipher. *IJCSNS International Journal of Computer Science and Network Security*, VOL.9 No.5, May 2009.

[15] Bhakti, M.A.C.; Abdullah, A.; Jung, L.T.; ,2007. EAP-based Authentication with EAP Method Selection Mechanism: Simulation Design. *Research and Development, 2007. SCOReD 2007. 5th Student Conference on* , vol., no., pp.1-4, 12-11 Dec. 2007

[16] D. W. Carman, P. S. Kruus and B. J.Matt, 2000. Constraints and Approaches for DistributedSensor Network Security. *dated September 1, 2000.NAI Labs Technical Report.*

[17] P. Langendoerfer, and K. Piotrowski, 2005. More Privacy in Context-aware Platforms: User Controlled Access Right Delegation using Kerberos, *Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers, Tenerife, Spain*, December 16-18, 2005.

[18] Y.M.; Senouci, S.-M.; Agoulmine, N.; , 2006, P-SEAN: A Framework for Policy-based Server Election in Ad hoc Networks. *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP* , vol., no., pp.271-281, 3-7 April 2006.

[19] Ahmed, A.; Yasumoto, K.; Shibata, N.; Kitani, T.; Ito, M.; ,2009. DAR: Distributed Adaptive Service Replication for MANETs. *Wireless and Mobile Computing, Networking and Communications, 2009. WIMOB 2009. IEEE International Conference on* , vol., no., pp.91-97, 12-14 Oct. 2009.

[20] Md. Shariful Islam, Md. Abdul Hamid and Choong Seon Hong, 2009. SHWMP: A Secure Hybrid Wireless Mesh Protocol for IEEE 802.11s *Wireless Mesh Networks, Lecture Notes in Computer Science, 2009,* Volume 5730/2009.

[21] *William Stallings; Cryptography and Network Security: Principles & Practice (5th ed.) Pearson/Prentice* Hall, 2010.

ملخص

صممت شبكات الاتصالات اللاسلكية لتعمل ضمن ظروف بيئية غير موثوقة وقابلة للتوسع. حيث تشكل الشبكات اللاسلكية التشعبية (Mesh Wireless Networks (MWN)) جزءاً اساسيا من شبكات الاتصالات في العالم، و تتكون من موجهات لاسلكية (Mesh Access Points) و اجهزة اتصال طرفية لاسكية تشعبية (Mesh Clients) مثل الكمبيوتر المحمول او الخلوي الذكي.

ان الاتصال في هذه الشبكة يتم بين الاجهزة الطرفية بعضها ببعض او من خلال الموجهات اللاسلكية. فتقوم الموجهات اللاسلكية بتقوية الاشارة و تكون ثابتة في الغالب ومتصلة بالكهرباء على عكس اجهزة الاتصال الطرفية التي تكون متحركة، والتي تعتمد في حصولها على الطاقة من خلال مصدر محدود (مثل البطارية)، مما جعلها عرضة للهجمات والاختراقات من قبل الهاكر.

ان طبيعة الاتصال اللاسلكية بين هذه الاجهزة المختلفة واعتمادها على وسط ناقل مفتوح (Shared Medium)، كما ان بنية الند- للند (Peer to Peer) في الاتصال بين الاجهزة الطرفية واعتمادها على مصادر محدودة الطاقة جعل منها عرضة للهجمات والاختراقات من قبل الهاكر والفايروسات المختلفة.

ان بروتوكول التحقق في هذه الشبكات المعروف بـ(Hybrid Wireless Mesh Protocol(HWMP)) يعمل على حماية خطوط نقل البيانات (Node-to-Node) بين كل جهازين اتصال متجاورين ، دون ان يحمي المعلومات المنقولة بين الجهازين المستقبل والمرسل (End-to-End)، كما لم يحمي عملية إعادة ارسال البيانات (Data Forwarding) مما يعرض البيانات لهجمات منها : الانتحال (Spoofing)، التوجيه الخاطئ (Misdirection) ، الافاضة (Flooding) الذي يغرق الوسط الناقل بالبيانات و يؤدي بدوره الى الازدحام (Congestion) .

في هذه الأطروحة، قمنا بالتركيز على حل مشكلة حماية البيانات المرسلة خلال نقلها من المرسل الى المستقبل (End-to-End protection)، وحماية خطوط النقل بين اجهزة الاتصال المختلفة (Node-to-Node). حيث اقترحنا بروتوكول: Ticket Authentication Wireless Mesh Networks Protocol (TAWMP).

ان هذا البروتوكول يتكون من مرحلتين (Two Phases) : المرحلة الاولى يقوم بتوفير الحماية لاجهزة التوجية اللاسلكية بالاعتماد على طريقة تشفير غير تماثلية (Asymmetric Encryption)، اما المرحلة الثانية تعتمدعلى طريقة التشفير التماثلية (Symmetric Encryption) من خلال من خلال توليد (ticket) يحتوي على مفتاح لتشفير البيانات المرسلة وذلك باستخدام EAP-TTLS والتي تتلائم مع اجهزة الاتصال المحمولة . البروتوكول المقترح يستخدم خادم التحقق (Authentication Server) للمصادقة على الاتصال مما يضيف عبئا على اداء الشبكة بشكل ضئيل جدا، إلا أنه آمن وفعال.

قد اعتمدنا في تصميم هذا البروتوكول نهجا ينطوي على التكامل من خلال توليد مفتاح لتشفير البيانات بين المرسل و جهازالتوجيه اللاسلكي (Node-to-Node). و ارسال (Ticket) تحتوي على مفتاح يستعمل لتشفير البيانات المرسسلة (End-to-End)

البروتوكول المقترح يستطيع الرد على هجمات التوجيه الخاطئ السلبية (Passive Attack) والفعالة (Active Attack) على حد سواء، ان بروتوكول (TAWMP) هو خطوة مهمة نحو توفير حماية (End-to-End) للبيانات المنقولة في الشبكات اللاسلكية التشعبية.