Addis Ababa University

College of Natural Sciences

Securing the Transmission of Group Addressed Data Frames by Enhancing the IEEE 802.11i Security Protocol

Meareg Abreha Hailemariam

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

November, 2016

Addis Ababa University

College of Natural Sciences

Meareg Abreha Hailemariam

Advisor: Dejene Ejigu (PhD)

This is to certify that the thesis prepared by Meareg Abreha Hailemariam, titled: *Securing the Transmission of Group Addressed Data Frames by Enhancing the IEEE 802.11i Security Protocol* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

	Name	Signature	Date
Advisor:	Dejene Ejigu (PhD)		
Examiner:	Mulugeta Libsie (PhD)		
Examiner:	Fekade Getahun (PhD)		

Abstract

Security in Wireless LAN technology is evolving from time to time. This progress can be easily visualized by recounting the success stories achieved through the years since the modification of its first security protocol, WEP, then the WPA and finally the WPA2. In spite of this improvement, there are still weaknesses that need protocol level modification. One such example is the Hole 196 vulnerability which was discovered in 2010. This vulnerability is inherent in the WPA/WPA2 security protocols design and can be used for several serious insider attacks as it allows authenticated clients to misuse the Group Temporal Key (GTK) to attack other clients.

So this thesis proposes an enhancement to the IEEE 802.11i security protocol to address the Hole 196 vulnerability.

Keywords: IEEE 802.11i Security, WLANs security, GTK security, Hole 196 vulnerability

Acknowledgments

I am deeply grateful to the source of harmony that sustains and supports me. I want to thank a lot my advisor Dejene Ejigu (PhD) for his wise intellectual guidance during the course of this thesis. It‟s been a pleasure working with him.

I am also much indebted to my professors and examiners; Mulugeta Libsie (PhD) and Fekade Getahun (PhD) for their thorough and critical comments which enabled me upgrade this thesis to what it is now.

Finally I would like to take this opportunity to thank my family and friends for their continuous care and love they provide me. Thank you all

Table of Contents

LIST OF TABLES iii
LIST OF FIGURES iv
LIST OF ALGORITHMS vi
LIST OF ACRONYMS vii
CHAPTER 1: INTRODUCTION 1
1.1 Overview 1
1.2 Motivation 2
1.3 Statement of the Problem 3
1.4 Objective 4
1.5 Methods 5
1.6 Scope and Limitations 5
1.7 Application of Results 6
1.8 Organization of the Rest of the Thesis 6
CHAPTER 2: LITERATURE REVIEW 7
2.1 Wireless LAN Technology and Wireless Networks 7
2.2 Layered Approach Network Security on WLANs 8
2.3 802.11 Protocol 9
2.4 802.1x Protocol 10
2.5 EAP over LAN 12
2.6 The 802.11 Security Protocols: WEP, WPA and WPA2 13
2.6.1 Wired Equivalent Privacy (WEP) 13
2.6.2 Wi-Fi Protected Access (WPA) 15
2.7 IEEE 802.11i 16
2.7.1 Wi-Fi Protected Access 2 (WPA2) 17
2.7.2 Pre-shared key (PSK) Mode 18
2.7.3 Enterprise Mode 19
2.7.4 Key Configuration and Hierarchy in 802.11i 20
2.7.5 The 4-way Handshake 21
2.7.6 The Group Key Handshake 23
2.7.7 Hole 196 Vulnerability (GTK Misuse among Authenticated Clients) 24

2.8 The Role of an Access Point in terms of Security 28
2.9 Summary 29
CHAPTER 3: RELATED WORK 30
CHAPTER 4: SECURING THE TRANSMISSION OF GROUP ADDRESSED DATA FRAMES BY ENHANCING IEEE 802.11i 38
4.1 Proposed Solution Requirement 38
4.2 Overview 38
4.3 Proposed Architecture of the Initial Distribution of the AP‟s Public Key 39
4.4 Proposed System Architecture of Group Addressed Data Frames Communication between the AP and an Associated Client during a Session 48
CHAPTER 5: IMPLEMENTATION AND VALIDATION 60
5.1 Overview 60
5.2 Development Environment 60
5.3 Implementation Details 60
5.4 Implementation of the Proposed IEEE 802.11i Enhancement 63
5.5 Validation Results 66
5.6 Computational Cost 68
CHAPTER 6: CONCLUSION AND FUTURE WORKS 73
6.1 Conclusion 73
6.2 Future work 73
References 74
Annexes 78

LIST OF TABLES

Table 3.1: Summary of Related Works……... 37

Table 5.1: Proposed solution results…… 69

Table 5.2: Existing mechanism results…… 70

LIST OF FIGURES

Figure 2.1: IEEE 802.11 MAC Frame Format 9

Figure 2.2: IEEE 802.11 Frame Control Field 10

Figure 2.3: IEEE 802.1x Model from the IEEE 802.1x Specification 11

Figure 2.4: WEP Encryption 14

Figure 2.5: WEP Decryption 14

Figure 2.6: Construction of TKIP per-frame keys 16

Figure 2.7: The 802.11i Authentication Procedures in PSK Mode 19

Figure 2.8: The 802.11i Authentication Procedures in Enterprise Mode 20

Figure 2.9: The 4-way Handshake 22

Figure 2.10: The Group key Handshake 23

Figure 2.11: Injection of Forged Group Addressed Data Traffic 25

Figure 2.12: ARP Poisoning in Wireless LAN 26

Figure 2.13: Normal Vs. Stealth ARP Poisoning in Wired LAN 26

Figure 2.14: Frame number Transfer from AP to Client 27

Figure 3.1: Client Isolation 34

Figure 4.1: A High Level Pictorial Representation of Integration of the Proposed Initial Key Sharing Architecture within the Initial Authentication Phases of the IEEE 802.11i Standard 40

Figure 4.2: Proposed architecture of Initial Distribution of the AP‟s Public key along its MIC 42

Figure 4.3: Architecture of the Proposed Session between the Associated Clients and the AP 49

Figure 5.1: Snapshot of the Network Topology in the Simulator (similar for both simulations) 62

Figure 5.2: Snapshot of an Access Point Broadcasting a Data Frame to Receiver Clients (clients 2, 3, 4, 5) 62

Figure 5.3: Snapshot of Client 3(malicious client) Sending Fake Group Addressed Data Frame to Client 2 63

Figure 5.4: Performance Evaluation for Existing Mechanism and Proposed Solution 70

Figure 5.5: Relative Speed Comparison 71

Figure 5.6: Event Density Measurement for the Existing Mechanism and Proposed Solution 71

LIST OF ALGORITHMS

Algorithm 4.1**:** Proposed public key and its MIC sending process from AP to authenticating client 45

Algorithm 4.2: Proposed initial authentication of public key sharing process, with AP, at authenticating client..... 47

Algorithm 4.3: Process of Proposed Broadcasting or Multicasting of Modified Group Addressed Data Frames from AP to Receiver Clients 54

Algorithm 4.4: Proposed Mechanism of Verifying and Accepting Group Addressed Data Frames from AP; at Receiver Clients..... 56

LIST OF ACRONYMS

AAA: Authentication, Authorization, and Accounting

AES-CCMP: Advanced Encryption Algorithm Counter Mode with Cipher Block Chaining Message Authentication Code Protocol

AP: Access Point

ARP: Address Resolution Protocol

AS: Authentication Server

DoS: Denial of Service

EAP: Extensible Authentication Protocol

ECC: Elliptic Curve Cryptography

GMK: Group Master Key

GTK: Group Temporal Key

IV: Initialization Vector

IDS: Intrusion Detection System

MIC: Message Integrity Code

MPDU: Medium access control (MAC) protocol Data Unit

MSK: Master Session Key

PAE: Port Access Entity

PMK: Pairwise Master Key

PRF: Pseudo Random Function

PTK: Pairwise Transient Key

RADIUS: Remote Authentication Dial-In User Service

RC4: Rivest Cipher version 4

RSA: Rivest, Shamir and Adelman

RSN: Robust Security Network

RSNA: Robust Security Network Association

TKIP: Temporal Key Integrity Protocol

WEP: Wired Equivalent Privacy

Wi-Fi: Wireless Fidelity

WIDS: Wireless Intrusion Detection System

WIPS: Wireless Intrusion Prevention System

WLAN: Wireless Local Area Network

WPA: Wi-Fi Protected Access

WPA2: Wi-Fi Protected Access Version 2

CHAPTER 1: INTRODUCTION

1.1 Overview

Wired Equivalent Privacy (WEP) is the first security protocol the IEEE 802.11 committee deployed for WLANs and has been in place since the adoption of the IEEE 802.11 standard in 1997. It is based on the RC4 encryption algorithm, an algorithm that was designed by Ronald Rivest in 1987. By 2001, WEP‟s cryptographic weaknesses had become well-known. Series of studies from various academic and commercial institutions had shown that, having the proper tools which are also available freely and a moderate amount of technical knowledge, intrusion is possible on WLANs with WEP protection [1].

The WEP protocol proved to be vulnerable to RC4 weak keys issue as described by Wagner [2]. Scott *et al.* [3] showed that RC4 is completely insecure in a common mode of operation which is used in the widely deployed WEP in which a fixed secret key is concatenated with known four modifiers in order to encrypt different messages. A new passive ciphertext-only attack on this mode can recover an arbitrarily long key in a negligible amount of time which grows only linearly with its size, both for 24 and 128 bit IV modifiers.

Cracking tools like Aircrack [4] or WepLab [5] are used to implement attacks and can recover a 128-bit WEP key in less than 10 minutes or slightly longer, depending on the specific access point and wireless card.

To correct the flaws in WEP, the IEEE 802.11 Task Group I (TGi) introduced the Wi-Fi Protected Access (WPA), a transitional security protocol till a stable one is designed. The WPA security structure contains the Temporal Key Integrity Protocol (TKIP) and operates in two modes: Pre shared Key (PSK) and Enterprise [6]. The WPA-PSK offers less security than the Enterprise version, as it requires a shared secret; however, it is easier to install. The TKIP is a WEP patch, designed to run on existing hardware, wrapping the WEP protocol with three new elements: a message integrity code (MIC) named Michael, a data frame sequencing procedure, and a per data frame key mixing function. Encryption is still carried out using the RC4 Stream Cipher [7].

Tews and Beck [8] showed a TKIP Michael attack that an attacker, who has about 12-15 minutes access to the network, is then able to decrypt an ARP request or response and send 7 data frames with custom content to the network. Furthermore, Beck showed an improved TKIP Michael attack that allow more and longer data frames to be injected by the attacker and also an attack on the Michael message integrity code [9]. With the increase in use of wireless networks, the initial protocols, WEP first then WPA, used to secure wireless communications were found inadequate due to many proven vulnerabilities so a new protocol was implemented, the Wi-Fi Protected Access 2 (WPA2) protocol.

The WPA2 standard has two components, encryption and authentication which are crucial in network security. The encryption piece of WPA2 mandates the use of AES (Advanced Encryption Standard) but TKIP is available for backward compatibility with existing WPA hardware. The authentication piece of WPA2 has two modes: Personal and Enterprise.

The Pair Transient Key (PTK) in WPA/WPA2 can be cracked based on a dictionary attack on a weak passphrase [11]. The Group Temporal Key-based (hole 196) [11] is also another possible attack on the WPA2 protocol found by Sohail [11], from Airtight Networks [12], in 2010.

Generally, a brief history of the WPA2 attacks can be categorized based on year:

- (2003-2004) Pre-Shared Key
- (2008) Protected Extensible Authentication Protocol
- (2008) Temporal Key Integrity Protocol
- (2010) Group Temporal Key

The first three attacks have been fixed or can be prevented by following recommendations in [11]. But the last one, the Group Temporal Key misuse vulnerabilities, also named Hole 196, since its discovery in 2010 has lasted to this time basically because it is a problem ingrained in the protocol itself.

1.2 Motivation

Year-after-year security studies show that insider security breaches continue to be the biggest source of loss to businesses and sensitive information leak or compromise, whether from

disgruntled employees or spies who steal and sell confidential data. This is very dangerous to organizations or users in general with sensitive information or business secrets that must not be revealed to attackers. To elaborate this motivation with facts on the real world let us see few of the published statistics of cyber-crime categories. A study conducted in 2010 by Baker *et al.* [15] states that 48% of the total data breaches were caused by insiders. The next year‟s, 2011, report of Verizon [16] shows 17% of the attacks were by implicated insiders.

Another investigation report [17] of the year 2010 reported that 26% of the total cyber-attacks were committed by insiders, 24% were unknown attacks and the remaining 50% were done by outsiders. This second report confirms too that how prevalent the attacks from insiders are; let alone the other unknown 24% attacks which can possibly include insider attacks. Statistically Cyber-attack reports through consecutive years show flip-flops in the trend of the categories. But such irregularity, as explained by the publishers in the reports, were mostly due to the joining or leaving of the highly contributing organizations like United States Secret Service (USSS) to share lists of cyber-attacks in the reports which can make a difference in the accuracy of statistics that get published.

All these statistical investigation results signal to us how urgently we need better security solutions in the wireless cyber world so that malicious attacks with various causes and intentions could be minimized as much as possible.

1.3 Statement of the Problem

WPA2 uses two types of keys: 1) Pairwise Transient Key (PTK), which is unique to each client, for protecting unicast traffic; and 2) Group Temporal Key (GTK) to protect broadcast data sent to multiple clients in a network. PTKs can detect address spoofing and data forgery. "GTKs do not have this property," according to page 196 of the IEEE 802.11 standard.

Caneill and Gilis [13] state that it's a man-in-the-middle attack, and it works because everyone can build and broadcast fake data frames with the GTK (shared group key). Because a client has the GTK for receiving broadcast traffic within the specific wireless network, the user of that client device could exploit GTK to create its own broadcast data frame.

Central to this vulnerability is the Group Temporal Key that is shared among all authorized clients in a WPA2 network. In the standard behavior, only an AP is supposed to transmit group-addressed data traffic encrypted using the GTK and clients are supposed to decrypt that traffic using the GTK. However, nothing in the standard stops a malicious authorized client from injecting spoofed GTK-encrypted data frames. Exploiting the vulnerability, an insider (authenticated client) can sniff and decrypt data from other authenticated clients, install malware and possibly compromise those devices. In short, this vulnerability means that inter-user data privacy among authorized users is inherently absent over the air in a WPA2-secured network [14].

1.4 Objective

General Objective

The general objective of this research work is to design and develop a secure group addressed (broadcast /multicast) data frames transmission architecture among authenticated clients in the 802.11i security protocol.

Specific Objectives

To achieve the general objective, the following specific objectives need to be satisfied:

- Study and analyze the current Group Temporal Key distribution mechanism of the WPA2 protocol.
- Study the applicability of cryptographic techniques to secure the group addressed data frame transmission in the IEEE 802.11i.
- Analyze the structure of the 802.11 group addressed data frames.
- Design an architecture that secures group addressed data frames transmission among authenticated clients in the 802.11i security protocol using relevant cryptographic techniques.
- Develop a prototype of the designed architecture and also partial prototype of the existing WPA2 (only parts of the protocol that are modified in the proposed architecture) for comparison in a simulated environment.

- Make performance measurement and comparison of the simulated proposed solution versus the existing protocol in the simulation environment.

1.5 Methods

- **Literature Review**

  Research papers that are related with infrastructure based wireless technologies, IEEE 802.11i security protocols, attacks on wireless protocols will be studied thoroughly.

- **Requirement Analysis**

  A thorough requirement analysis will be made to improve data frames to solve the GTK vulnerability in the 802.11i protocol.

- **Architecture Design**

  During the design phase, the proposed system architecture specified in the specific objective will be thoroughly designed.

- **Prototype Simulation and Validation**

  The proposed design will be implemented as per the specifications put during the development of the architecture in a simulator. We have chosen to use a simulation environment to test our design since it is infeasible to modify the 802.11i protocol and implement it in real devices. The prototype‟s validity will be evaluated by comparing it with another simulation of the existing protocol. Existing protocol will be simulated partially, by only simulating the parts that our proposed design is going to modify, so that the result obtained from the comparison between the existing mechanism and the modified one (our proposed architecture) will enable us to measure the validity of the proposed solution.

1.6 Scope and Limitations

This research will only provide an architectural level fix to the GTK‟s vulnerability exploitation in the WPA2 protocol in a new architectural level design. Other possible weaknesses of the WPA2 protocol or weaknesses of a known public key cryptography algorithm are out of this

research‟s scope. So this thesis will give emphasis to the architectural validity of the proposed solution to solve the GTK vulnerability.

Another limitation of this research is that it does not concern with malicious group addressed frames that have come to the receiver clients through the legitimate access point. Such attacks can succeed if the access point (or a server if available) does not have an intrusion detection or prevention system. Our proposed solution takes into account that such systems are installed in the server or AP.

1.7 Application of Results

The proposed solution will be designed with a consideration that it will motivate other researchers and the authorities of the IEEE 802.11 standard to initiate the fix in the 802.11i protocol either using the solution suggested in this research or coming up with alternative better solutions to end the vulnerability.

1.8 Organization of the Rest of the Thesis

The rest of this thesis is organized as follows. The next chapter deals with literature review. It will try to give a detailed and summarized report on the state of art in relation to the 802.11, 802.11i security protocols and weaknesses. Chapter 3 reviews research works that are related to our study and more specifically works related with proposed solutions to the Hole 196 vulnerability. The proposed design of securing the transmission of group addressed data frames among authenticated clients will be discussed thoroughly in Chapter 4 and validation and evaluation will be discussed in Chapter 5. The last Chapter will give conclusion and also indicate future works.

CHAPTER 2: LITERATURE REVIEW

2.1 Wireless LAN Technology and Wireless Networks

Wireless LANs (WLANs) use Infrared (IR) light or Radio Frequency (RF) electromagnetic waves to transmit and receive data unlike wired LANs which rely on the physical connections of copper wire or optical fiber to transport information.

This simplifies and speeds up network installation, increases flexibility and scalability, and is cost effective while allowing greater user mobility. These advantages, combined with the ever-increasing data bandwidth offered by wireless technology, make WLANs an attractive alternative for individuals and organizations that plan to implement or expand a LAN without having to install or move wires [26].

It was only after 1985, after the ISM (Industrial, Scientific, and Medical) band‟s restriction for spread-spectrum technology for public use was lifted that wireless data networks gained much popularity. As a result, many vendor companies started building wireless hardware which were compatible with the then existing data networks but incompatible with each other as they were vendor specific. This indicated that a standard was needed if devices of two or more vendors were to work together.

So in 1988, the IEEE established a committee to develop the 802.11 standard. All of the 802 standards deal with the data link layer and physical layer of the OSI reference model. Part 11, or 802.11, defines all of the specifications for wireless local area networks [27].

The Wi-Fi (Wireless Fidelity) Alliance was formed in 1999 to certify interoperability of wireless local area network products based on IEEE 802.11 specification. Today the use of Wi-Fi (IEEE 802.11 based) technologies has spread throughout the globe and became very crucial to control the security of the data traffic on these devices.

The 802.11 wireless technologies have been implemented with security mechanisms, or commonly called security protocols, to protect users‟ data traffic. These security protocols have been scrutinized by cryptographers, researchers in the field and even enthusiasts. This made a better quality security protocols, on wireless LAN devices, be realized and it is a continuing process as long as security weaknesses or holes don‟t cease to exist in the protocols.

2.2 Layered Approach Network Security on WLANs

Stanley [28] puts the ultimate requirements of wireless security in two categories as Encryption and Data Privacy which aims to provide data privacy along data integrity; and the other, Authentication and Access control to grant right level of access by providing a mutual authentication mechanism.

Securing WLANs is a difficult job than securing on wired LANs/networks since the medium, air, is shared by all unlike a cable and an associated port which allows traffic only from the particular source in case of wired LANs. The medium, air, makes it possible to any attackers to be able to send data traffic to the WLAN without having to attach a device, physically, to the networking infrastructure. This makes the security of WLANs harder as the wireless security protocol designs have to consider all the threats of wired networks and threats that solely endanger wireless networks since air-based traffic enables attackers to find devices directly and send payloads straight, otherwise would not be easy in case of wired networks as data traffic are directed through cables.

The OSI model of a network communication has 7 layers [29] and each layer has its own designated tasks to achieve. A given network layer requests a service from the layer below it and answers to service request from the layer above it. In network security, the security protocol is embedded on one of the network layers. Choosing an optimal network layer for the security protocol‟s implementation is a difficult task which demands understanding and visualizing all the potential security problems and attacks which can be done, on the system to be implemented, in the frame of time extending from the present to the future.

Implementations in the past have shown that data security is possible on various layers of the OSI network model. These include data link layer security, transport layer security, network layer security and application layer security.

Why Layer-based security matters: All the possible attack scenarios of networked environment happen at one of the available network layers. Each layer is vulnerable to various kinds of attacks; so any security protection must be thoroughly studied on which layer it must be implemented before deciding to implement it. Such pre-analysis will have a crucial effect on the

future of the system‟s security. A well-studied security implementation will be more resistant to attacks.

Generally, securing a data communication on a network is considered possible at one of the previously mentioned layers but the core issue would be to understand on which layer it would be more effective and relevant with the context of the system design. Though security implementation is application dependent, security at lower layer ensures security of the layers above it and better performance in terms of speed and bandwidth is achieved as a result of less overhead on higher layers [31, 32].

802.1x is a port-based access standard which protects upper layer attacks by denying access to the network before authentication at the data-link layer is completed [30]. The rest of IEEE 802 standards too, including the 802.11, are implemented at the data-link layer.

2.3 802.11 Protocol

According to the IEEE 802 standards committee definition, all of the components in the 802.11 architecture fall into either the media access control (MAC) sub layer of the data-link layer, as the data-link layer is further separated into the Logical Link Control (LLC) and media access control, or the physical layer. The IEEE 802.11 wireless standard defines the specifications for the physical layer and the media access control (MAC) layer that communicates up to the LLC layer.

At the physical (PHY) sub layer, IEEE 802.11 defines a series of encoding and transmission schemes for wireless communications the most common of which are the Frequency Hopping Spread Spectrum (FHSS), Direct Sequence Spread Spectrum (DSSS), and Orthogonal Frequency Division Multiplexing (OFDM) transmission schemes while the MAC sub layer consists of a MAC header, the frame body, and a frame check sequence (FCS). Figure 2.1 [33] and 2.2 [49] show the IEEE 802.11 MAC Frame Format and Frame Control Field respectively.

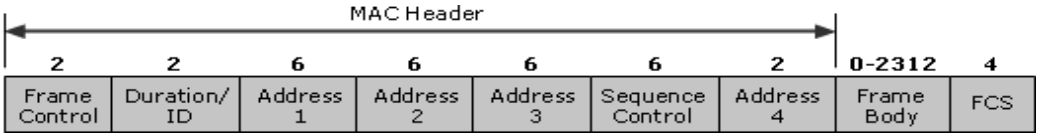

Figure 2.1: IEEE 802.11 MAC Frame Format

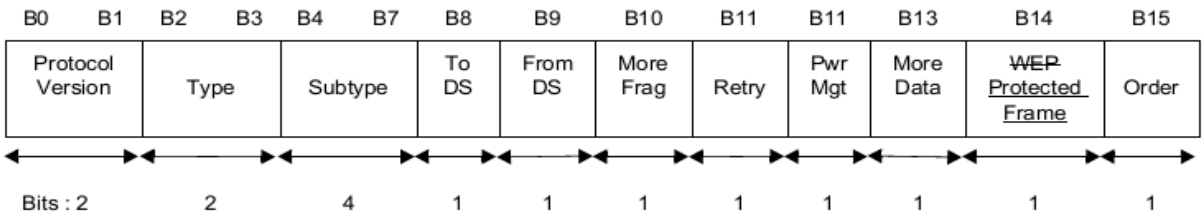

Figure 2.2: IEEE 802.11 Frame Control Field

2.4 802.1x Protocol

The IEEE 802.1x standard defines port-based network access control to provide authenticated network access in Ethernet networks. This port-based network access control uses the physical characteristics of the switched LAN infrastructure to authenticate devices attached to a LAN port. Access to the port can be denied if the authentication process fails. Although this standard was designed for wired Ethernet networks, it has been adapted to 802.11 wireless LANs. The following are components of the IEEE 802.1x:

- **Port Access Entity:** A LAN port, also known as *Port Access Entity (PAE)*, is the logical entity that supports the IEEE 802.1x protocol that is associated with a port. A PAE can adopt the role of the authenticator, the supplicant, or both.
- **Authenticator:** Is the end component in the link initiating EAP authentication [34]. An *authenticator* is a LAN port that enforces authentication before allowing access to services accessible using that port. For wireless connections, the authenticator is the logical LAN port on a wireless AP through which wireless clients in infrastructure mode gain access to other wireless clients and the wired network.
- **Supplicant:** Is the end of the link that responds to the authenticator [34]. The *supplicant* is a LAN port that requests access to services accessible on the authenticator. For wireless connections, the supplicant is the logical LAN port on a wireless LAN network adapter that requests access to the other wireless clients and the wired network by associating with and then authenticating itself to an authenticator.
- **Authentication Server**: Is an entity that provides an authentication service to an authenticator [34]. To verify the credentials of the supplicant, the authenticator uses an

authentication server, which checks the credentials of the supplicant on behalf of the authenticator and then responds to the authenticator, indicating whether or not the supplicant is authorized to access the authenticator's services.

The authenticator's port-based access control defines the following different types of logical ports that access the wired LAN by means of a single physical LAN port:

- **Uncontrolled Port:** The uncontrolled port allows an uncontrolled exchange between the authenticator (the wireless AP) and other networking devices on the wired network regardless of any wireless client's authorization state. Frames sent by the wireless client are never sent using the uncontrolled port.
- **Controlled Port:** The controlled port allows data to be sent between a wireless client and the wired network only if the wireless client is authorized by 802.1x. Before authentication, the switch is open and no frames are forwarded between the wireless client and the wired network. When the wireless client is successfully authenticated using IEEE 802.1x, the switch is closed, and frames can be sent between the wireless client and nodes on the wired network.

Below, Figure 2.3 [41], explains the IEEE 802.1x model in terms of component level communication.

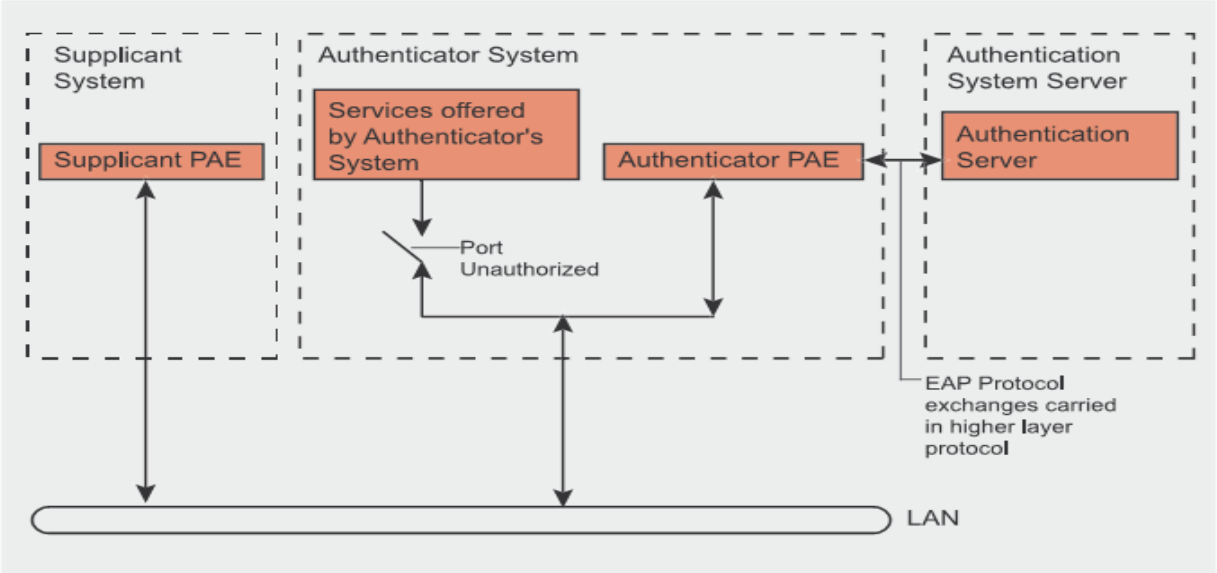

Figure 2.3: IEEE 802.1x Model from the IEEE 802.1x Specification

On an authenticating Ethernet switch, the wired Ethernet client can send Ethernet frames to the wired network as soon as authentication is complete. The switch identifies the traffic of a specific wired Ethernet client using the physical port to which the Ethernet client is connected. Typically, only a single Ethernet client is connected to a physical port on the Ethernet switch.

In wireless LANs, because multiple wireless clients contend for access to the same frequency channel and send data using the same channel, an extension to the basic IEEE 802.1x protocol is required to allow a wireless AP to identify the secured traffic of a particular wireless client. The wireless client and wireless AP do this through the mutual determination of a per-client unicast session key. Besides the AP only authenticated wireless clients have knowledge of their per-client unicast session key. Without a valid unicast session key tied to a successful authentication, a wireless AP discards the traffic sent from the wireless client.

2.5 EAP over LAN

Extensible Authentication Protocol (EAP) is used as a standard authentication mechanism for IEEE 802.1X. EAP is a Point-to-Point Protocol (PPP) based authentication mechanism that was adapted for use on point-to-point LAN segments.

EAP messages are normally sent as the payload of PPP frames. To adapt EAP messages to be sent over Ethernet or wireless LAN segments, the IEEE 802.1x standard defines EAP over LAN (EAPOL), a standard encapsulation method for EAP messages. EAP typically runs directly over data link layers such as Point-to-Point Protocol (PPP) or IEEE 802, without requiring IP**.**

In 802.11i, EAP is a component of an 802.1x network. EAP is designed to create a mechanism to provide authentication types that leverage existing Authentication, Authorization, and Accounting (AAA) solutions. EAP messages can be transferred from the 802.1x supplicant to the authenticator or authentication server.

In the case where no backend authentication server is used, the EAP server is part of the authenticator. In the case where the authenticator operates in pass-through mode, the EAP server is located on the backend authentication server [34].

A backend authentication or AAA server is an entity that provides an authentication service to an authenticator. When used, this server typically executes EAP methods for the authenticator.

AAA protocols with EAP support include RADIUS [35] and Diameter (DIAM-EAP). 802.1X does not specify what kind of back-end authentication server must be present, but RADIUS (Remote Authentication Dial-In User Service) is the "de-facto" back-end authentication server used in 802.1x.

EAP is now in the 802.11i protocol as an authentication mechanism to be used in integration with 802.1X. Within IEEE 802.11i, EAP is used for both authentication and key exchange between the EAP peer and server [36]. One of the advantages of the EAP architecture is its flexibility. EAP is used to select a specific authentication mechanism, typically after the authenticator requests more information in order to determine the specific authentication method to be used. Rather than requiring the authenticator to be updated to support each new authentication method, EAP permits the use of a backend authentication server, which may implement some or all authentication methods, with the authenticator acting as a pass-through for some or all methods and peers [34]. EAP authentication methods suitable for Wireless LANs must satisfy the mandatory requirements criteria as listed in the RFC 4017 [36].

Benton [27] explains that the WPA and WPA2 standards define the EAP (Extensible Authentication Protocol) methods which are allowed for 802.1x authentication. The 802.11i revisions only specified that EAP method had to mutually authenticate the user and the server. There are various kinds of authentication mechanisms; widely implemented include the EAP-TTLS, EAPTTLS/MSCHAPv2, PEAPv0/EAP-MSCHAPv2, PEAPv1/EAP-GTC, EAP-SIM, EAP-AKA, EAP-FAST [27] etc. A research paper [37] surveys authentication protocols and provides a good write up on their desired properties.

2.6 The 802.11 Security Protocols: WEP, WPA and WPA2

2.6.1 Wired Equivalent Privacy (WEP)

WEP was the first protocol designed to provide wireless security in terms of confidentiality, access control and data integrity for users implementing 802.11 wireless networks. WEP was developed by a group of volunteer IEEE members [27].

WEP employs RC4 [38] algorithm, designed in 1987 by Ron Rivest for RSA Security, for encryption and uses two key sizes: 40 bit and 104 bit; to each is added a 24-bit initialization

vector (IV) which is transmitted directly. At the transmitter side the plaintext is XOR'ed with the key stream, generated after KSA and PRGA process of RC4 and cipher text is obtained. These steps take place in the reverse order at the receiver side using the same key. WEP uses CRC-32 [13] hash algorithm for data integrity [39]. Figure 2.5 [39] and 2.6 [39] show the WEP encryption and decryption processes respectively.

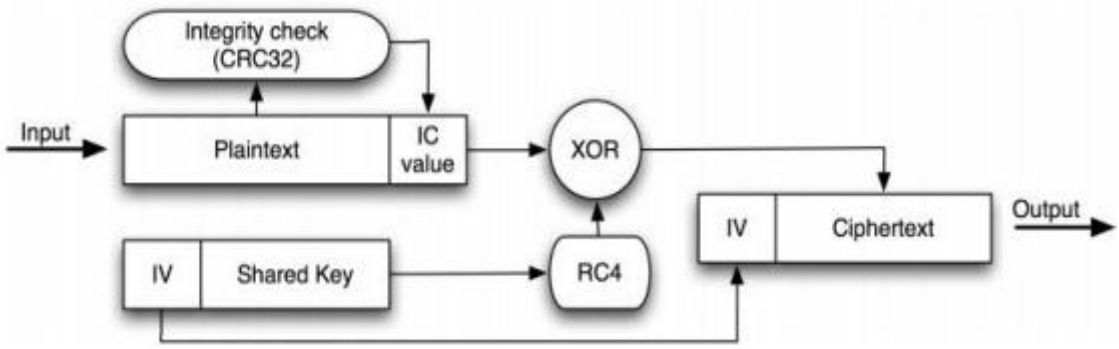

Figure 2.4: WEP Encryption

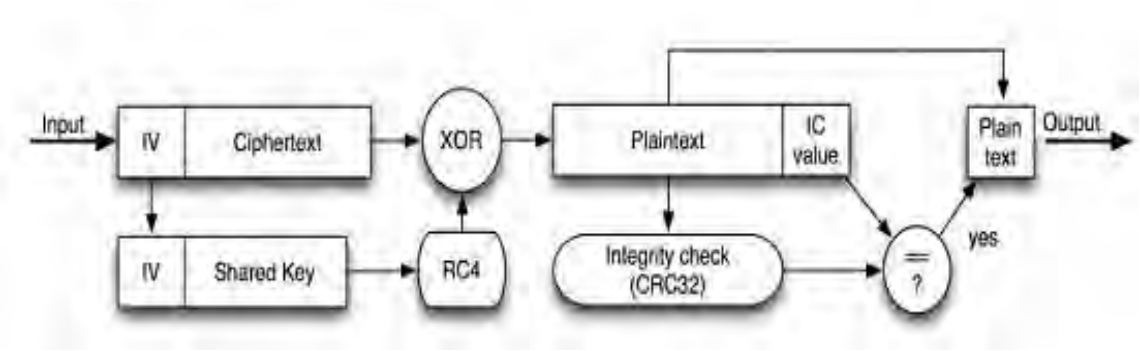

Figure 2.5: WEP Decryption

The WEP protocol quickly proved vulnerable to RC4 issues described by Wagner in 2001 [2]. In 2001, Scott *et al.* [3] showed two vulnerabilities in the RC4 encryption algorithm on WEP: invariance weaknesses and known IV attacks.

The integrity check stage also suffers from a serious weakness due to the CRC32 algorithm used for this task. CRC32 is commonly used for error detection, but was never considered cryptographically secure due to its linearity, as Nikita *et al.* stated back in 2001 [21].

Gutjahr *et al.* [40] and Scott *et al.* [3] list weaknesses of WEP such as that it does not prevent forgery of frames or replay attacks, it uses weak RC4 keys and reuses Initialization Vectors (IV) which made data decryption possible with cryptanalytic methods or data modification without knowing encryption key and lack of key management. Another paper [41] also reassures that the WEP reuses IVs since the length of the IVs it uses are too short. It is indicated also that ICV algorithm is not a good choice for cryptographic hash [42].

There are various off-the-shelf tools that exploit these vulnerabilities, allowing WEP keys to be recovered by analyzing the traffic [10]. Through the years several attack techniques had been successfully implemented on the WEP. Some of the famously known attacks are: FMS attack [45], koreK attack, Fragmentation attack [43], PTW attack [46] and Café-latte attack [44].

2.6.2 Wi-Fi Protected Access (WPA)

The utter weakness of the WEP to provide adequate security to its users has led the IEEE 802.11 committee to design and come up with an improved security standard that avoids most of the weaknesses that had previously doomed the WEP to failure. WPA addresses all known vulnerabilities in WEP by using a greatly enhanced encryption scheme, Temporal Key Integrity Protocol (TKIP) together with 802.1x/ EAP authentication.

TKIP is a major enhancement over traditional WEP protocol. Since legacy APs and wireless interface cards are equipped with hardware necessary for WEP, TKIP is introduced to work on the same hardware for backward compatibility but with software enhancement for additional security [47]. A research work in [1] explains that TKIP uses a key hierarchy and key management methodology, by leveraging the 802.1x\EAP framework, and thus removes the predictability which intruders relied upon to exploit the WEP key. Moreover, it also uses the Message Integrity Check (MIC), which is also commonly called Michael, on the data frames it sends to check the integrity of the data received. The steps for building of TKIP per-frame keys are showed in Figure 2.6 [47].

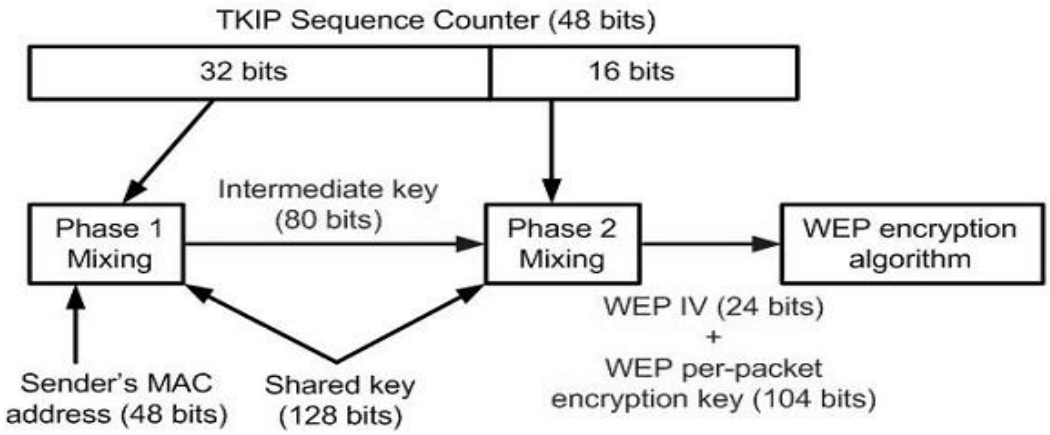

Figure 2.6: Construction of TKIP per-frame keys

Stanley in [28] summarizes the benefits of WPA over WEP concisely. It summarizes that WPA applies strong network access control through mutual authentication, it supports 802.1x\EAP framework or pre-shared keys, adopts dynamic keys in TKIP which improves the key management, enforces data integrity through Michael MIC and provides forward compatibility to 802.11i.

The WPA security protocol, nevertheless avoided several of the WEP‟s weaknesses, it has been subject to various attacks. Some of these successful vulnerabilities include Beck and Tews' Improved Attack on RC4 [8], Ohigashi-Morii Attack (Beck-Tews + Man-in-the-middle) [48], the Hole 196 Vulnerability [11] and Dictionary based attacks.

2.7 IEEE 802.11i

Benton [27] and Naamany *et al.* [47] explain that the IEEE task Group I of the 802.11 was formed to replace the original authentication and privacy, the WEP algorithm, provided by the initial 802.11 standard with an enhanced security as well as support to legacy protocols for backward compatibility. IEEE802.11i is based on IEEE 802.11 standard with security enhancement in the MAC layer [48]. The final draft was ratified on the 24th of June, 2004 as 802.11i [49].

According to paper work [47] several of the 802.11 standards (a, b, d, e, g, h, i, j) were rolled up into the new base 802.11 standard "IEEE 802.11-2007" on March 8th, 2007. Networks compatible with the new security protocols are referred to as Robust Security Networks (RSNs).

Nowicki [50] defines Robust Security Network (RSN) as the term applied to the strongest security model that 802.11i uses to authenticate, authorize, and protect the connection between the STA and AP. It further explains the robust parts of the 802.11i standard: 802.1x for authentication and authorization, EAP for authentication transport, and support for stronger message encryption and integrity mechanisms such as CCMP, and, optionally, TKIP.

Pre-RSN stations cannot connect to an RSN network. An association between two RSN stations is referred to as Robust Security Network Association (RSNA). Each RSNA has its own unique set of keys and key lifetimes. This is necessary because RSN networks introduce an entirely new key management and authentication protocol in addition to new encryption algorithms [27].

2.7.1 Wi-Fi Protected Access 2 (WPA2)

The final IEEE 802.11i security protocol which fully implements the requirements of the 802.11i amendment is called Wi-Fi Protected Access Version 2 (WPA2). The predecessor, WPA, was only designed as a transitional protocol to address the weaknesses found in WEP so it didn‟t fully contain all the requirements of the 802.11i as stated in [49] but it is supported in 802.11i‟s WPA2 for backward compatibility purpose. WPA2 differs from WPA because it includes specification for IBSS (Independent Basic Service Set), pre-authentication, CCMP and WRAP (optional).

The authentication piece of 802.11i (which include both WPA and WPA2) operates in two modes: Personal and Enterprise [10].
The Personal mode requires the use of a PSK (Pre-Shared Key) and does not require users to be separately authenticated while the Enterprise mode, which requires the users to be separately authenticated through the Authentication server based on the IEEE 802.1x authentication standard, uses the Extended EAP (Extensible Authentication Protocol) which offers other EAP standards to choose from [50].

So both WPA2 and WPA enjoy two modes of operations which made them fit enough to organizational-level security while at the same time they are feasible to be used for Small Office/Home Office (SOHO) environments. According to [18], there are five specific key types that are of particular interest in the 802.11i amendment; which are the AAA key (or MSK),

Pairwise Master Key (PMK), Pairwise Transient Key (PTK), Group Master Key (GMK) and Group Temporal Key (GTK).

The AAA key is jointly negotiated between the Supplicant and the Authentication Server (AS). This key information is transported via a secure channel from the AS to the Authenticator. The pairwise master key (PMK) may be derived from the AAA key [49]. The PTK is a key value used to protect unicast Medium access control (MAC) protocol Data Units (MPDUs) from that source and it is derived from PMK and other input values which will be explained in more detail in Section 2.7.5. A GTK is random value, assigned by the broadcast/multicast source, which is used to protect broadcast/multicast medium access control (MAC) protocol data units (MPDUs) from that source. It may be derived from GMK which is an auxiliary key used to derive a GTK [49].

Generally, a successful authentication process means that the station and the access point verify each other‟s identity and generate some shared secret for subsequent secure data communication. In case of Enterprise mode, as we explained in the 802.1x standard section, the authentication server can be implemented either in an access point, or through a separate server.

2.7.2 Pre-shared key (PSK) Mode

In this mode of operation as shown in Figure 2.7 [20], the 802.1x based authentication mechanism is totally avoided. This mode of operation is meant only for SOHO environments where users don‟t have to install the Authentication server. This mode is not safe to be used in an organizational mode. Since except for the 802.1x authentication all the phases are similar with the Enterprise‟s mode, these common phases will be explained under the Enterprise mode‟s

section.

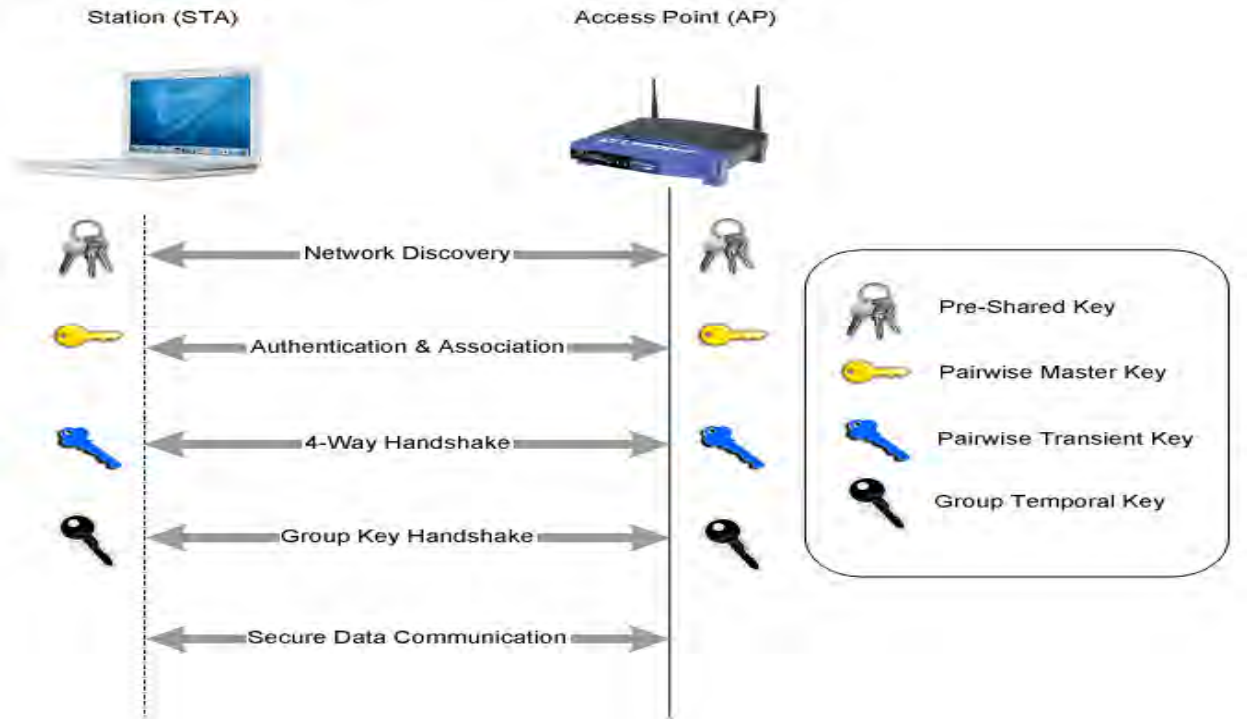

Figure 2.7: The 802.11i Authentication Procedures in PSK Mode

2.7.3 Enterprise Mode

In this mode the 802.1x authentication framework is deployed. Clients are authenticated separately. This mode is safe to be deployed for organizational-level security. A research work in [20] explains the 6 distinct phases in the 802.11i authentication procedures, shown in Figure 2.8 [20], in detail as follows:

1. **Discovery Phase**: the access point periodically advertises its IEEE 802.11i security policy in a certain channel through the Beacon frame. Station passively monitors the Beacon frame and uses the frame to identify the access point.
2. **Authentication and association phase**: the station selects one access point from the list of available access points and attempts to authenticate and associate with that access point. However, this authentication requires to be supplemented by further mutual authentication.
3. **EAP/802.1x/RADIUS authentication:** the station and the authentication server perform mutual authentication and thus some common secret (i.e., Master Session Key-MSK) is

generated between the station and the authentication server**.** This step does not exist if a static Pre-Shared Key (PSK) is pre-installed over the station and the access point.

4. **4-way Handshake:** Is used to confirm the mutual possession of the PMK and to distribute the initial GTK between the AP and clients [49]. The PMK is used to derive a fresh Pairwise Transient Key (PTK) which is shared between the station and the access point.
5. **Group Key Handshake**: due to the requirement of broadcast/multicast traffics, the access point uses it to issue a new Group Temporal Key (GTK) to peers with whom the local station (STA) has already formed security associations [49].
6. **Secure Data Communication:** by using PTK or GTK, the station and the access point construct a secret transmission channel and thus accomplish robust data confidentiality.

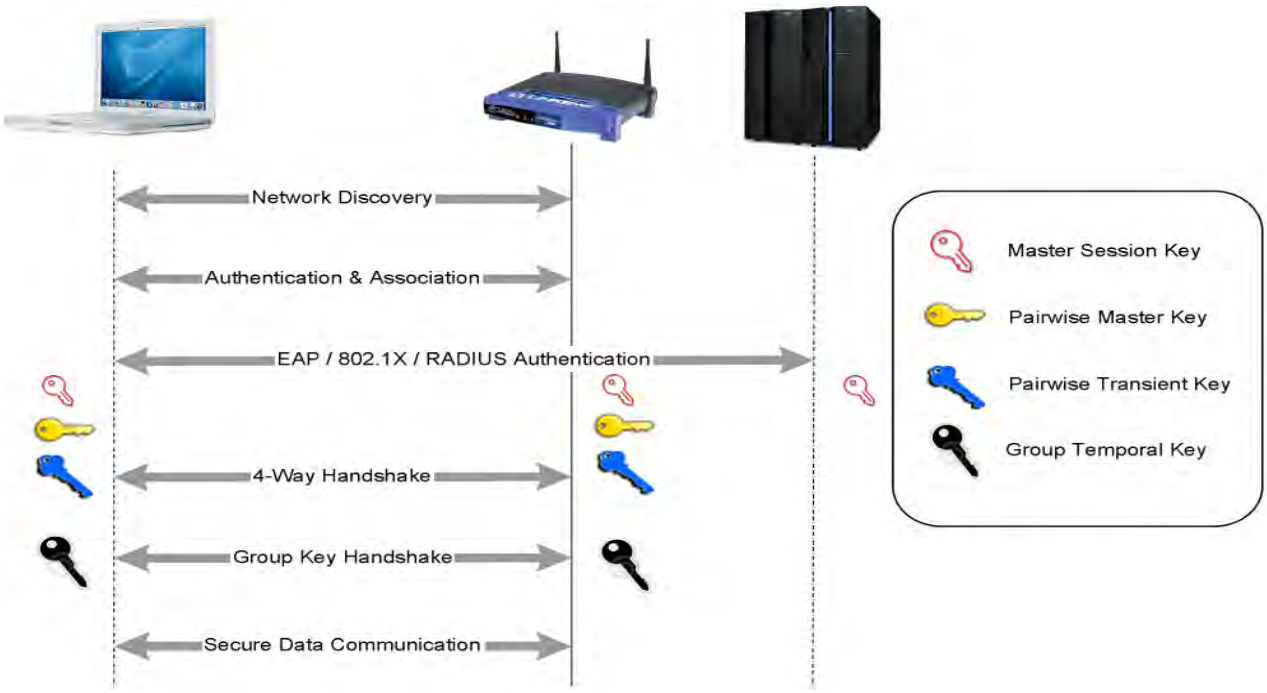

Figure 2.8: The 802.11i Authentication Procedures in Enterprise Mode

2.7.4 Key Configuration and Hierarchy in 802.11i

One of the major changes introduced in the 802.11i standard was the way encryption keys are handled and generated. With WEP, the pre-shared key was immediately appended to the IV and given to RC4 to generate the key-stream. As a result this puts a limit on the number of frames that could be encrypted from one pre-shared key without repeating a key-stream.

To fix this limitation, a new key hierarchy was introduced for RSN network that prevents the re-use of any keys regardless of the chosen encryption algorithm [27]. In the IEEE 802.11i the means to provide these fresh keys is achieved through the use of 4-way Handshake and Group key Handshake protocols [49].

There are essentially two possible top level keys that are used to generate the rest of the keys in the hierarchy [18, 27]. In the case of a pre-shared key based network, the top key is simply the pre-shared key (PSK). On the other hand, if the network uses the 802.1x authentication, the top key is the master session key (MSK) or AAA key. These keys are used to derive the next key in the list, called the pairwise master key (PMK).

In the PSK scenario, the PSK simply becomes the PMK. In the 802.1x scenario, the PMK is derived from a section of the MSK. This section is dependent on the Extensible Authentication Protocol (EAP) method used for 802.1x. Once the PMK has been derived, all of the subsequent key derivations are the same for both 802.1x and PSK networks.

The next key in the hierarchy is the Pairwise Transient Key (PTK). This key is specific to the client and the AP that it communicates with. Therefore, if all of the clients use the same PMK (i.e., a PSK- based network) they will still derive different PTKs. The PTK is also unique to each association; so, every time a client associates with an AP, it will derive a different PTK.

The problem with maintaining an individual key with each client becomes apparent when dealing with multicast and broadcast traffic**.** If N clients are associated, the AP would have to retransmit the frame N times, encrypting it with a different key each time. To avoid this, the AP generates a random Group Master Key (GMK). Every time a client associates or disassociates, the AP derives a new Group Transient Key (GTK) from the GMK. This GTK is delivered to each one of the clients to be used to encrypt and decrypt multicast and broadcast traffic.

The final keys in the hierarchy are the EAPOL-Key Key Confirmation Key (KCK), EAPOL-Key Key Encryption Key (KEK), and the Temporal Key (TK). The KCK and KEK are used to protect EAPOL-Key frames. The Temporal Key is the one that is used with either TKIP or CCMP to protect regular network traffic.

2.7.5 The 4-way Handshake

The pairwise key hierarchy utilizes PRF-384 (Pseudo Random Function-384) to generate a 348 bits PTK during CCMP or PRF-512 in case of TKIP to derive a 512 bits PTK from a PMK

which shall be 256 bits [49]. A PTK is comprised of three types of keys: KCK (Key Confirmation Key –128 bits) used to check the integrity of EAPoL-Key frames, KEK (Key Encryption Key–128 bits) used to encrypt the GTK and the TK (Temporal Keys–128 bits) used to secure data traffic [10].

NB: In this thesis the terms *authenticator* and *supplicant* will be used interchangeably with the terms *AP* and *client* respectively.

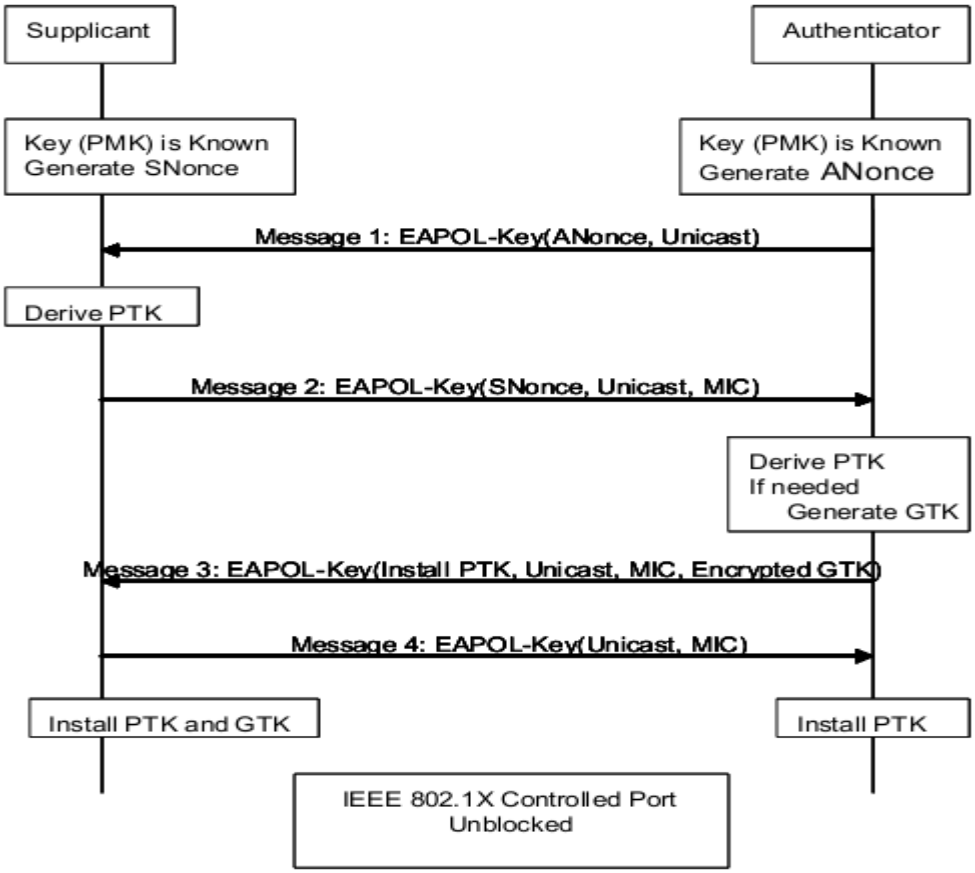

Figure 2.9: The 4-way Handshake

In Figure 2.9 [49], a pictorial representation of the 4-way handshake, Message 1 delivers ANonce to the Supplicant and initiates negotiation for a new PTK. At the supplicant side the PTK generator PRF function is used. Message 2 delivers SNonce to the Authenticator so that it can derive the same PTK using the same PRF function used at the supplicant. The MIC prevents undetected modification of Message 2 contents. Message 3 also sends a GTK encrypted with the PTK and confirms to the supplicant that there is no man-in-the-middle attack by sending the

MIC of the message. The supplicant verifies that the authenticator derived the same PTK by comparing the MIC it received and a MIC it generates from the received message. If comparison proves valid it decrypts the GTK and installs it. Message 4 has no cryptographic purposes but just to assure the authenticator that it has installed the PTK and GTK.

The 4-way Handshake accomplished by four EAPoL-Key messages between the client and the AP is initiated by the AP. The tasks performed by the 4-way Handshake can be summarized as; confirmation of the client‟s knowledge of the PMK, derivation of a fresh PTK, installation of encryption and integrity keys, encryption transport of the GTK and confirmation of the cipher suite selection.

2.7.6 The Group Key Handshake

This process is initiated to change the GTK value with a new one during disassociation or deauthentication of a client or at a time configured into the AP to reduce data exposure if GMK is ever compromised and uses the KEK generated during the 4-Way Handshake to encrypt the GTK [10, 49]. Without a separate Group Key Handshake, a new PTK would have to be generated and installed by the authenticator and every supplicant every time the GTK needed to be changed [18].

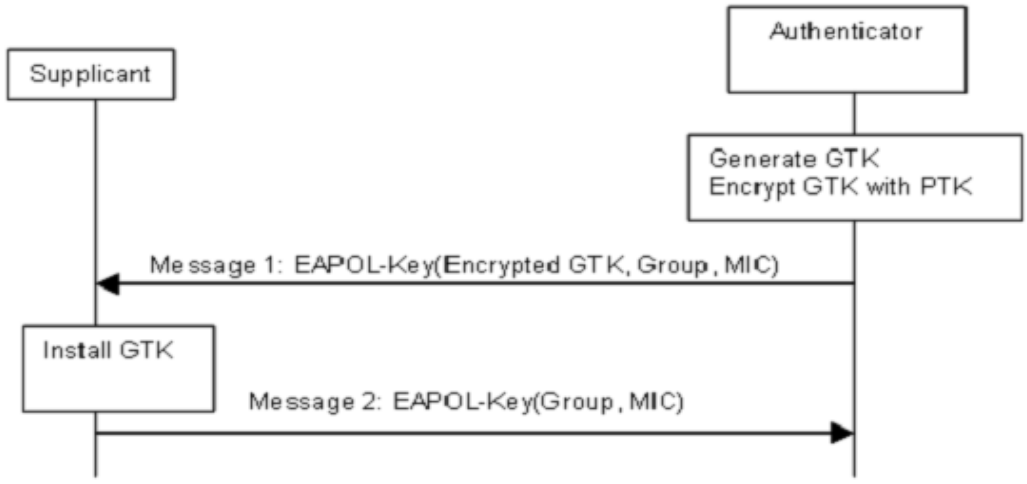

Figure 2.10: The Group key Handshake

Figure 2.10 [49] depicts the steps in the Group Key Handshake process. First the authenticator derives the GTK using PRF-128 for CCMP or PRF-256 for TKIP and outputs 128 bits GTK for CCMP or 256 bits for TKIP. Then it encrypts the derived GTK with one of the PTKs generated during the 4-way Handshake and sends it to the particular client along the message‟s MIC so that the supplicant can detect any modification attempt on the message during transmission. Then the supplicant sends back the authenticator an acknowledgment along MIC. This process is repeated for every associated client.

The finalized security protocol of the 802.11i, the WPA2, too didn‟t escape from the weaknesses inherent in the way it is built which make it vulnerable as its predecessor, the WPA, though it is much stronger. Some of the weaknesses include:

- Dictionary based attacks on the PTK during the PSK mode [41]
- Attack on Message 2 of the 4-way Handshake [41]
- Hole 196 vulnerability attack (the GTK misuse among authenticated clients) [11, 41]

More details on the Hole 196 attack will be explained in the next section as it is the sole aim of this research to come up with a solution that equips the WPA/WPA2 security protocols with a mechanism that enable to prevent the GTK misuse among authenticated clients.

2.7.7 Hole 196 Vulnerability (GTK Misuse among Authenticated Clients)

In the 802.11i standard data origin authenticity is only applicable to unicast data frames. The protocols do not guarantee data origin authenticity for broadcast/multicast data frames [49]. This inherent weakness of the standard allows authenticated clients to send fake group addressed data frames to other peers. Group Temporal Key (GTK) is designed to be used as an encryption key, to encrypt group addressed data frames, by the AP and as a decryption key to decrypt, such data frames, by receiver clients. Parameters (GTK, KeyID and PN) required for sending group addressed MPDUs are known to all connected clients. A malicious user can always create fake frames. Because of this WPA2 secured Wi-Fi networks are vulnerable to insider attacks [8, 11]. The 802.11i standard assumes group addressed data frames are always broadcasted or multicast to receiver clients by the access point only. But there is no mechanism to enforce and prevent associated clients from forging and sending fake group addressed frame by using the required parameters. This design flaw, which allows GTK misuse among authenticated clients, was

discovered in 2010 [8] and is named “Hole 196” since it was found in the last line on page 196 of the 1232-page IEEE 802.11 Standard (Revision, 2007). Figure 2.11 [11] shows the general case scenario of how the Hole 196 based vulnerability can be used to attack clients.

Figure 2.11: Injection of Forged Group Addressed Data Traffic

Hole 196 Vulnerability-based Attacks

- **Stealth mode ARP Poisoning/Spoofing attack**

An attacker can snoop victim‟s data traffic by setting the gateway address in the victim‟s ARP cache to his/her own so that victim will send data communication to the attacker‟s gateway which will be forwarded through the AP. The attacker can either send the victim‟s data it intercepted to its true destination (actual gate) which is also commonly known as Man in the Middle (MITM) attack and keep playing the gateway‟s role to the victim or it can drop it which causes the victim IP layer Denial of Service since the attack was caused by poisoning the victim‟s ARP cache. Figure 2.12 [11] shows a Hole 196 based attack where the attacker sniffs the address of the Gateway and then changes target‟s ARP table of the gateway address to its own. This will direct all of the target‟s communication destined to or passing through the gateway to be forwarded to the attacker.

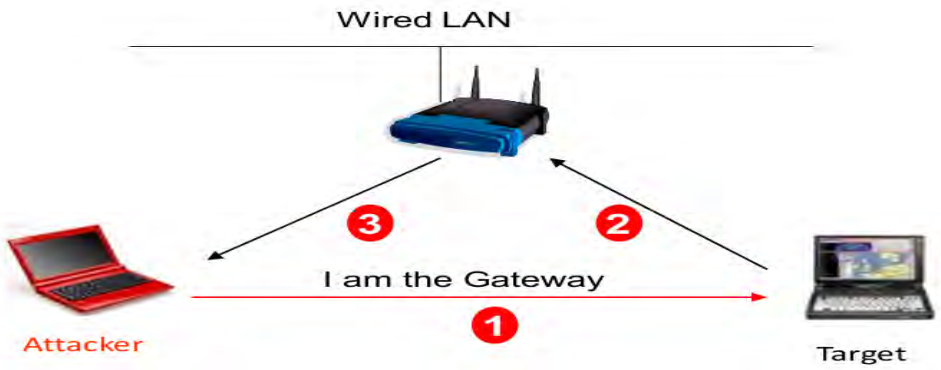

Figure 2.12: ARP Poisoning in Wireless LAN

Steps in ARP Poisoning:

1. Attacker injects fake ARP frame to poison client‟s cache for gateway. (For example attacker can poison victim‟s ARP cache of Gateway to his machine‟s address.)
2. Victim sends all traffic to attacker
3. Now attacker can either drop traffic or forward it to actual gateway

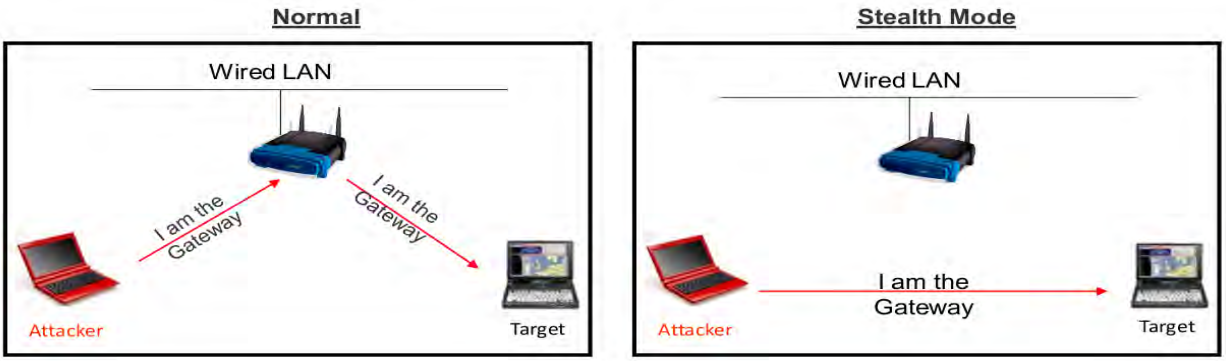

Figure 2.13: Normal Vs. Stealth ARP Poisoning in Wired LAN

In Normal case: Shown in the left side in Figure 2.13 [11] ARP poisoning frames appear on wire through AP. Chances of being caught is high as attack would be visible to WIPS or WIDS at can reside in the AP.

In Stealth Mode case: Shown in the right side in Figure 2.13 [11] ARP poisoning frames are invisible to AP, never go on wire as they are done directly implemented on clients. So cannot be detected by any ARP cache poison detection tool.

- **IP level targeted attack**

The attacker can spoof AP‟s MAC and IP and can reset or indirect victim‟s TCP, scan port, inject malware, privilege escalation, etc. as the data frame it sent is received as a valid data frame since it fools the victim with a spoofed MAC and IP and valid session‟s GTK, KeyID and Data frame Number (shared information in the WLAN is known to the attacker; he/she is also an authenticated and associated client).

- **Wireless DoS attack**

Victim‟s cache memory can be overflowed by a storm attack of fake Group addressed data frames sent to it by the attacker and this can block the client‟s downlink broadcast reception as there won‟t be enough cache to receive the real data frames sent from the network.

- **Replay Attack Detection in WPA2**

All clients learn the PN (Data frame Number) or Frame Number in CCMP Header associated with a GTK at the time of association. A 48 bit Frame Number is present in all CCMP encrypted data frames. In Figure 2.14 [11] an AP sends group addressed data frames to all clients with a new PN. If new PN is greater than locally cached PN then frame is decrypted and after successful decryption, old PN is updated with new PN.

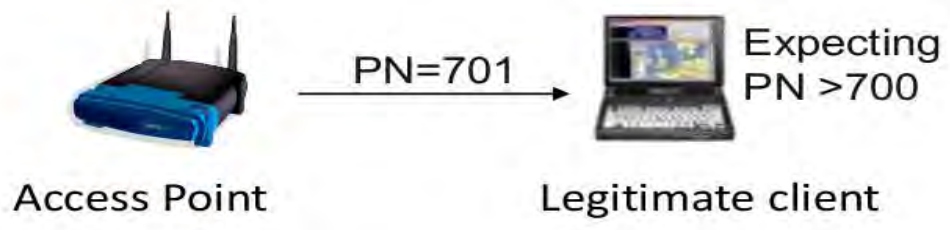

Figure 2.14: Frame number Transfer from AP to Client

Malware Injection Scenario

A research paper in [22] claims to have demonstrated an example scenario of creation and injection of valid 802.11i frames with malware payload by encrypting the frame with the GTK shared among authenticated and associated clients. In their procedure they capture a broadcast or multicast frame then by taking important information like frame sequence number and initialization vector and increment these values in the forged frame they sent so that the frame won‟t be dropped by the target/victim as a result of incorrect sequence number and IV value.

2.8 The Role of an Access Point in terms of Security

In infrastructure based WLANs the AP does not only serve as a pass to communication between the associated clients but it serves also as a device where the security of frames that pass through it are checked.

Earlier we have stated that wired LANs were easier to secure than Wireless LANs for a reason that communication in Wired LANs was directed through cables and as a result every data transmission‟s source, destination and threat behavior, on the LAN, was detectable since the data would be exposed to scrutiny of Intrusion Detection System (IDS) and Intrusion Prevention Systems (IPS) that are installed at key points in the LAN.

In contrast in Wireless LANs data transmission is not specifically directed as the medium is air. So in order for a Wireless LAN system to test a given data on transmission for its threat level it must pass through a device which have the Wireless IDS or Wireless IPS are installed to examine captured data. But this is not always the case; which is also the basis for the major success of the Hole 196 attacks. If the same attacks were going to be sent through the normal WLANs route, which means sending them through the AP, the success of the attacks would entirely depend on the AP security mechanism strength assuming no WIDS or WIPS on the clients‟/receiver‟s side.

In reality most APs, in infrastructure based WLANs, are equipped with a WIDS or WIPS. Effective WIPS can help a lot in preventing GTK misuse attacks or other types of attacks that pass through the AP. There are dozens of researches and vendors that propose detection or prevention solutions to detect or prevent attacks on Wireless LANs.

2.9 Summary

WLANs have passed through generations of security protocols which proved to have weaknesses in terms of achieving secure data communication. As a result several security protocols and modifications have been proposed and resulted in an evolutionary improvement of WLAN security; nevertheless, there are still weaknesses such as the Hole 196 which need to be addressed.

CHAPTER 3: RELATED WORK

Rajotiya and Arora in [23] indicated that the WPA2 as the strongest security protocol of the IEEE 802.11i standard since it has implemented the block cipher AES which is much secure than the RC4 algorithm which was used in previous security protocols. But it is still vulnerable to several attacks due to transmission of unencrypted management and control frames and misuse of shared Group Temporal Key (GTK) among peers connected to the WLAN.

They propose a solution to the Hole 196 vulnerability as follows: "First the authenticator can assign a random and unique GTK to every peer in the network. Then the access point generates a random and unique GTK and during a multicast or broadcast the sender sends encrypted text using its key to the access point. The access point then transmits the encrypted text along with the originating station‟s GTK to the recipient stations. The recipient stations then decrypt the text at their end using this GTK. At the end of the session, a new GTK is assigned to the sender station." The authors claim that each peer connected to the network is unaware of the GTK of the rest of the peers.

The problem with this proposed solution is that there is no mechanism described that can possibly prevent any of the peers in the network from forging a valid group addressed data frame using a particular peer‟s (or its own or a fake) GTK and broadcast it directly to other peers along the GTK value it used. Thus, distributing unique GTKs to each peer and, when peers want to send a broadcast or multicast message, the AP‟s sending of these GTK keys along the encrypted group addressed message to receiver peers does not prevent authenticated peers from forging a valid group addressed message.

In this research paper there is no mechanism to control group addressed messages replay attacks. The worst scenario is in this research clients can even generate a fake GTK keys and use it to encrypt a malicious data frame, they prepared, also which they want to send/ broadcast it as a group addressed data frame. This makes the existing Hole 196 vulnerability get worse because in the current case forging group addressed data frames possibility is limited to authenticated clients but according to this research‟s proposal the Hole 196 vulnerability would also be open to outsiders or unauthenticated clients which makes the current hole only bigger and more dangerous than it is already. The reason, unauthenticated clients are able to inject fake group

addressed frames is because there is no notion of shared GTK within the associated clients of the WLAN. When a client receives a broadcast or multicast message, all it does is use the GTK that came along the received encrypted message and use it for decrypting the message. Even if the authors do not state it, if we assume clients check for the origin of received group addressed data frames, then spoofing the address of the access point would be sufficient. In general, this research does not provide a valid and thoroughly studied solution at all; it widens the Hole 196 vulnerabilities.

Matej in [19] proposes a new approach of Wi-Fi protection; the wireless intrusion detection system based on reputation system with suspicious and malicious behavior detection of wireless devices. According to the author‟s argument the approach is based on using a system of cataloging, which looks at the profile of client behavior on the Wi-Fi networks, categorizing the reputation of the devices and then deciding whether a device or Wi-Fi network‟s user is risky or not.

In order to collect traffic data communication within the Wi-Fi which will be fed to the server for analysis, they claimed to use a tool from aircrack-ng [12] called Airserv-ng which is a wireless card server. This tool allows multiple wireless application programs to use a wireless card independently via client-server TCP network connection. So it must be installed on every client node so that it can send the captured data frame to the server.

This can be infeasible in corporate networks where there can be many clients because it will need to process each client‟s action in the WLAN to build trust profiles and also it learns only little about client‟s profiles or behaviors in smaller networks where the clients can be random (every time new clients joining and others leaving the WLAN; in places like Internet cafe).

Moreover, the author didn‟t indicate what kind of approach is exactly used to analyze a data frame in order to categorize it as a risky or non-risky and also didn‟t explain how the analysis can work to categorize/rate clients based on the reputation to score for any of the vulnerabilities including Hole 196.

Sohail [11] and Sohail *et al.* [24] provides suggestion on how to prevent the WPA2 protocol from the Hole 196. The suggestion is to deprecate use of GTK and group-addressed data traffic

and send broadcast or multicast data frames as unicast data frames because of the following arguments:

- APs in controller based WLAN architectures often do not broadcast data frames over the air.
- For backward compatibility, unique GTKs can be assigned to individual authorized Wi-Fi clients in the network.
- If data frames have to be broadcasted, then transmit as unicast.

The authors indicated the downsides of this approach can be an increase of throughput on the WLAN if broadcast traffic is sent as unicast.

Sending every broadcast/multicast addressed data frame as a unicast data frame would totally destroy the notion of using broadcast/multicast data frames in the network. The very aim of using broadcast or multicast addresses is to achieve less overhead (things like encryption are done only once since the message is supposed to be the same copy at every receiver end) and then sending a once processed copy of the group addressed frame only to the desired receivers instead of doing processing a "frame x" *n* times for *n* number of recipients if it were going to be sent as a unicast data frame since the process of preparing the group addressed data frame for each user is supposed to be unique.

Sohail [11] lists suggested preventions and countermeasures to prevent the Hole 196 vulnerability. There are end point security solutions that are client side software which can be used to detect ARP cache poisoning. But the two limitations of such end point security as listed in the research paper are:

- Varieties of client devices connect to WPA2 secured Wi-Fi networks while such software is available only for either Windows or Linux running devices.
- It is infeasible for a large scale environment as every end-point is supposed to install such client side software.

More emphases on limitations of the end point security we have noticed are summarized as:

- More crucial issue is that new users who get connected to the WLAN are also expected to install these client side solutions which makes it practically infeasible as new clients might only want to hook up to the network only temporarily (especially in places where there are Wi-Fi hotspots such as in cafes and restaurants where random new customers might join or leave the network). So expecting such solution is infeasible in terms of client‟s time, memory and processing consumption of clients‟ devices and can demand client‟s knowledge on identifying and installing the right tools.
- Such solution can‟t prevent malware installation attacks sent through frames encrypted with GTK.

The research paper also discusses various downsides an infrastructure side solution, called Public Secure Data Frame Forwarding (PSPF)/peer-to-peer (P2P) or Client Isolation, to minimize the Hole 196 vulnerability.

Mirzoev and White [25] discuss the benefits of PSPF in more detail, by specifically simulating Cisco‟s PSPF feature. They claim that this feature can be used to stop communication between two Wi-Fi enabled client devices by making the AP not to forward data frames between two clients within the same network. The authors indicated that PSPF is only useful in networks where the data traffic is not encrypted during transmission. They explain the drawback of available security protocols such as WPA is that they implement authentication of users which can be infeasible in places like Wi-Fi hotspots.

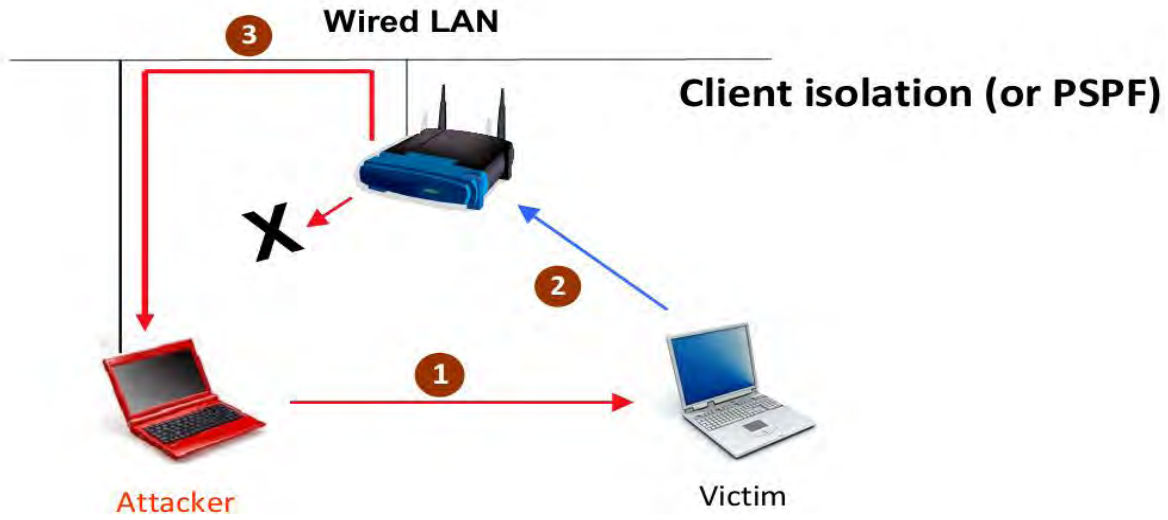

Figure 3.1: Client Isolation

They claim their experiment proved that a network that uses a PSPF does not transfer data traffic and stays invisible to other clients associated to the network (Even though when an ARP attack was attempted, having a previous knowledge of the target‟s MAC and IP address, the target victim experienced brief Denial-of-service) while an access point without the PSPF feature allows its associated clients to communicate with each other which allow malicious clients to perform attacks like the ARP cache poisoning. They indicated the drawback of the PSPF is that it prevents users from connecting to wireless peripherals like wireless printers and storage devices and suggested a solution to connect them to home switch which can hide them from the AP so that the PSPF does not affect them. The authors also finally suggested a TLS encryption of data without authentication even though they put the PSPF as an immediate solution in current scenario.

The downsides of the PSPF features are summarized as follows:

- The PSPF is currently not implemented by many vendors of AP.
- It does not work across APs in standalone mode [11].
- As shown in Figure 3.1 [24] of this solution is that the PSPF feature can be bypassed by connecting to a wired network (a non-Wi-Fi network) and then sending target associated clients of the WLAN the attack, like ARP cache poisoning, from there [11]. This is

possible because the wired network is hidden from the AP so any communication from there is not affected by the PSPF feature at all. Attackers can attach devices to the wired part of the network and victimize targets in the wireless network.

- Even though the PSPF feature blocks frame forwarding between peers within the WLAN it does not block direct (device-to-device) communication between them (without having them to forward data frames through the AP). It only can control frames that pass through it. So malicious authenticated clients can send directly, without using the AP, a valid GTK encrypted group-addressed data frame, which contains the payload, to their victim target client. In order for this attack to work we presumed the attacker spoofs AP‟s MAC and IP address so that the victim won‟t drop it when it checks the source address similarity with the gateway addressed in its ARP cache. Attackers can send malwares, cause denial-of-service by changing victim‟s ARP (example by changing it to a non-existing address) and even if they can‟t capture victim‟s data communication into their device (the Wi-Fi device) they can set-up a non-Wi-Fi device somewhere outside the Wi-Fi network and direct victim‟s data communication to it by poisoning/changing its gateway to the address of the non-Wi-Fi device.
- It does not allow any communication between associated clients of the same WLAN (sharing the same AP). This might not be a desired property in an environment where flexibility is needed (for example in a WLAN where direct group communication between clients is needed); users need to have the full freedom to connect other peers within the WLAN without the possibility of being attacked.
- All other attacks that don‟t involve the AP are possible [11].

So the PSPF or client isolation feature is not a generic solution which can be suitable to every wireless network scenario and so can be only suggested as a temporary prevention mechanism.

Summary of Related Works

Table 3.1: Summary of Related Works

No.	Research	Approach	Drawbacks
1.	Enhancing Security of WI-FI Network	During each new session assign each client a unique randomly generated GTK. Group addressed data frames are supposed to be sent through the AP.	There is no mechanism stated that enforces or checks if a received group addressed data frame is sent through the AP. Still, a forged group addressed data frame can be sent to clients directly and even with a fake GTK this time. There is no reply attack detection mechanism stated.
2.	New approach in wireless intrusion detection system	Learning the client's behavior and assigning score values.	This approach can be infeasible when there are many clients in a WLAN. It can also result in weak learning in WLANs with dynamic environment.

No.	Research	Approach	Drawbacks
3.	Hole196 Vulnerability in WPA2	Deprecate use of GTK and send every group addressed data frame as unicast.	High throughput on the WLAN by as result of having to process group addressed data frames individually.
4.	The Role of Client Isolation in Protecting Wi-Fi Users from ARP Spoofing Attacks	Client isolation/Public Secure Data frame Forwarding	This can be bypassed by connecting to a wired network (a non-Wi-Fi network) which can be used to fire back attacks to target clients through the AP. It does not block direct communication between devices (clients). Thus, all attacks that do not involve AP are possible. It does not allow scenarios which need direct communication between clients of the WLAN.

Until this time there are very few research papers available that aim to solve the Hole 196 vulnerability or the GTK misuse by authenticated clients. Considering the seriousness of the vulnerability, it can be said that it hasn‟t been given enough attention which can be the reason for finding very few resources that proposed solutions.

CHAPTER 4: SECURING THE TRANSMISSION OF GROUP ADDRESSED DATA FRAMES BY ENHANCING IEEE 802.11i

4.1 Proposed Solution Requirement

Key pairs generated through a public key infrastructure will be used in the proposed enhancement of the 802.11i. In the literature review, we have explained that there are two modes of operations in the 802.11i WLANs; namely, personal and enterprise modes. In the personal mode there is no EAP/802.1x authentication involved but in the enterprise mode. In this latter mode, there is a dedicated device or software server that handles authentication of clients in the WLANs. Thus, enterprise mode WLANs directly benefits from our proposed solution. Personal mode WLANs need to have AP with public key infrastructure; because it is with PKI that the proposed solution enforces the basic notion of trust between authenticated clients.

4.2 Overview

The proposed solution is comprised of two sequentially linked architectures that show a PKI generated initial keys distribution and secure session respectively.

The first part of the proposed architecture shown in Figure 4.2 is the one that deals with the distribution of PKI information to clients during initial authentication.

In the current WPA2 protocol, particularly in the Enterprise mode, clients are separately authenticated through the authentication server associated with the WLAN. As already had been explained in the literature review Section of the authentication phases there are various initial phases that include authentication and key sharing which the client who wants to join the WLAN must first pass by providing credentials expected from an authorized. So the proposed architecture of initial keys distribution integrates itself into the initial 802.11i authentication procedure seamlessly.

The second part of the proposed solution designs a secure session which avoids the GTK misuse by authenticated clients. It uses the key information shared during the proposed initial key distribution as an input.

4.3 Proposed Architecture of the Initial Distribution of the AP's Public Key

In the current specification of 802.11i a client on authentication, right after finishing authentication phases successfully, would be ready for the association phase and if successful too would be able to join the WLAN so as to access the WLAN‟s resources.

Our proposed initial key distribution architecture does not merely extend the available authentication phases but it also integrates with it by reusing the PTK shared during the 4-way handshake authentication phase. In the proposed extension the authenticating client would need to have the AP‟s public key before being able to access resources from the WLAN. It has to be noticed that the proposed extension is for sharing the public key, not to authenticate them at all. Because by the time the proposed initial key distribution mechanism starts sharing the public key, the receiver client has already proved its identity and obtained a unique PTK by the 4-way key handshake. **Error! Reference source not found.** lists the client authentication phases in the 802.11i along the proposed key sharing modification which follows after the 4-way handshake step.

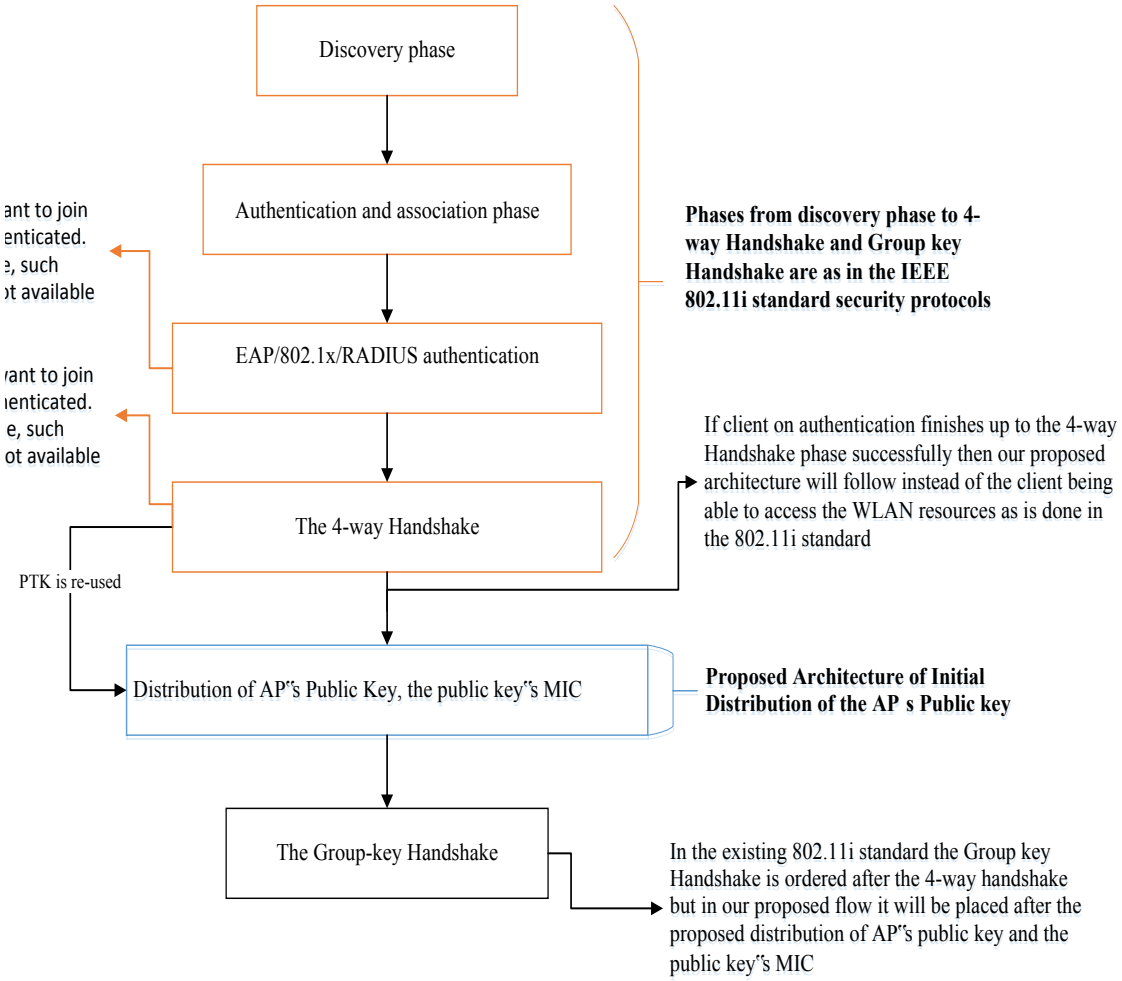

Figure 4.1: A High Level Pictorial Representation of Integration of the Proposed Initial Key Sharing Architecture within the Initial Authentication Phases of the IEEE 802.11i Standard

Core Components of the Proposed Public Key Distribution Mechanism

Public Key Infrastructure (PKI)

Our proposed solution bases on the integration of a PKI technique. A typical PKI would need two mathematically related keys; namely private and public key pair. In our proposed architecture these pair would initially be generated or installed from a trusted source into the

authentication server. This issue is not under the scope of our research. In our proposed initial key sharing architecture an AP would have a unique pair of private and public keys to uniquely identify its communication.

In case of Enterprise mode these pair of keys would be copied or sent to the particular AP from its authentication server. A single authentication server can be associated with many APs while an AP can only be associated with a single authentication server at a time. So the implementation must ensure that a particular pair of public and private key belongs to only one AP as reusing the same key pair with different APs highly increases the chances of the keys‟ vulnerability to cryptanalysis and as a consequence might fell in the hands of unauthorized entities. The process of copying or sending the pair of keys from an authentication server to the particular AP can be done only one time as long as there is no change on the pair or parameters used. This would reduce the cost of bandwidth necessary to send a copy of key pair and parameters from server to an AP every time the WLAN is on.

In case of the pre-shared mode the installation of pair of keys on an AP would mean pre-installation or manual insertion as there is no authentication server to communicate with it.

Message Integrity Code of the AP s Public Key

The proposed distribution of the Public key must have a mechanism that enables the receiver (the client) to make sure that the encrypted public key it received is not corrupted or altered by attackers or by any other reason during the transmission. So to achieve this, we first prepare the MIC value of the AP‟s public key at the AP. Any MIC algorithm can be used; this is mainly an issue of implementation choice. At the receiver side the MIC of AP‟s public key will serve as a validity checking mechanism of the received public key. Receiver client generates MIC of the received public key and compares it with the received MIC and if similar public key will be accepted as valid else drops it.

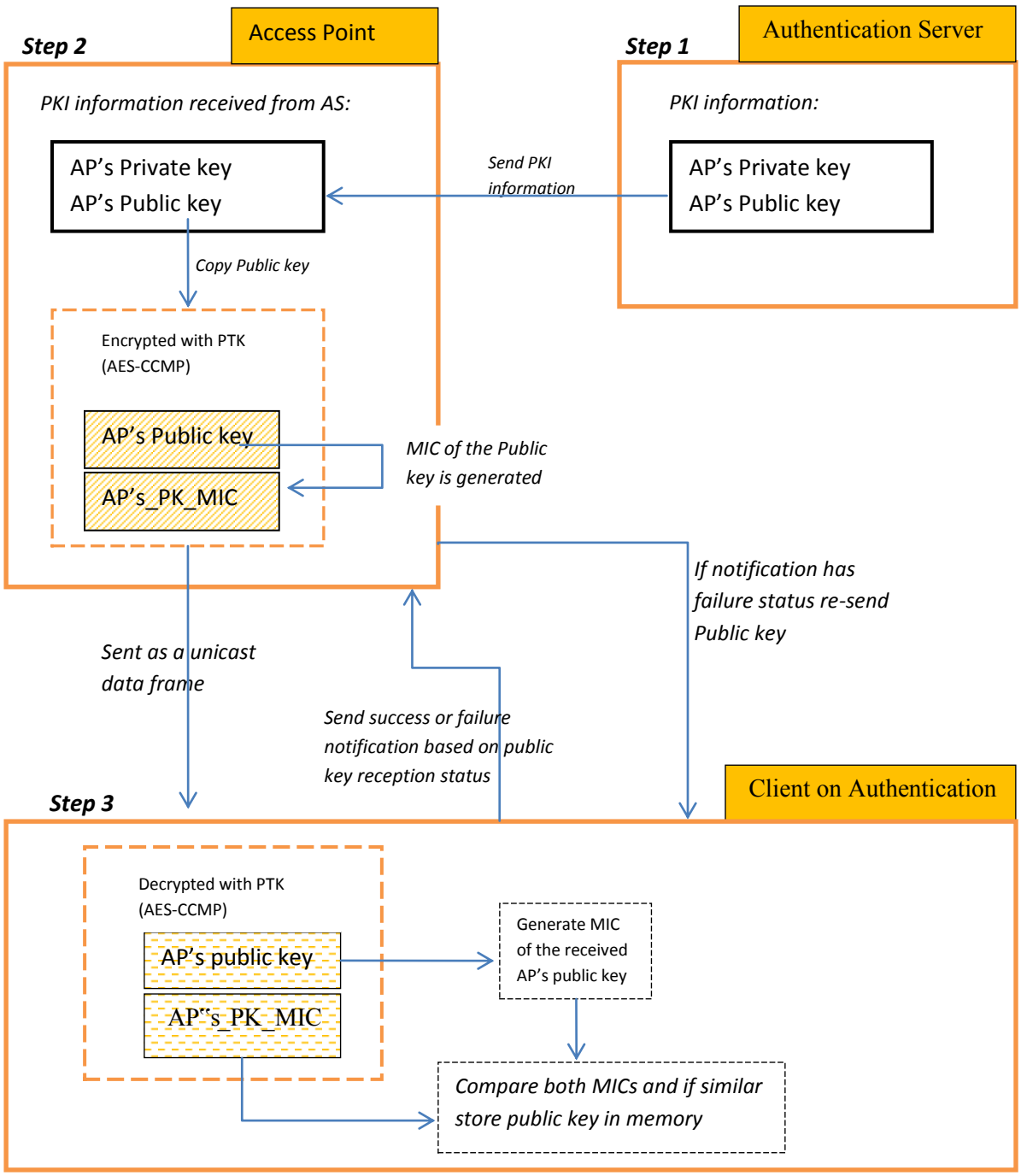

Figure 4.2: Proposed architecture of Initial Distribution of the AP's Public key along its MIC

An authentication server, in case of Enterprise mode of the WPA/WPA2 security protocols of 802.11i standard, is the backend which is responsible for the authentication of clients to join the WLAN. It is responsible to generate the Master Session Key (MSK) (or also called AAA key; this depends on the specific EAP method implemented) and store credentials of clients [47]. In our architecture the authentication server will still have all its previous responsibilities as indicated in the WPA/WPA2 protocols; but also has to share PKI information between the AP and client on authentication as shown in Figure 4.2.

In WPA/WPA2 a successful authentication involves a successful completion of the five authentication phases (which includes the 4-way and Group-key Handshake). So during these authentication phases the authentication server is particularly responsible to generate a Pairwise Master Key (PMK) from MSK which will be used by the AP to generate the PTKs of all authenticated clients and GTK.

As its name indicates the authentication server is only needed to authenticate clients for their credentials and notify the AP of their legitimacy status. It also sends the generated PMK to the AP before the clients are notified of their legitimacy through the AP. Clients who proved their identity to the authentication server would further need to exchange messages with the AP so as to generate keys that will be used, for managements‟ encryption and decryption, during the session, when clients are able to send and receive data traffic from the WLAN.

The proposed authentication server component in Figure 4.2, having all the existing responsibilities, would also have to send a copy of the AP‟s private key and public key to the A P which can be a one-time process in order to reduce processing and bandwidth load.

Access Point: The Access Point component shown in Figure 4.2 is the central coordinator of the network which acts as a bridge to connect peers/associated clients within the network and with the outside network. In the current 802.11i protocol a client is legitimate to access the WLAN‟s resources only after the successful completion of the 4-way and Group Handshakes.

So right after the 802.11i standard‟s successful 4-way Handshake, assuming that the AP has its private and public keys, our proposed distribution of the AP‟s public key to the client on authentication starts on.

Thus in our proposed authentication mechanism after a successful 4-way Handshake, unlike in the current 802.11i standard, a client would not be directed to group key handshake; instead the client must receive the AP‟s public key and verify its authenticity. Only it is after the handshake of the public key information that the AP would initiate the group key handshake and then, if successful, allow the client to access the WLANs resources.

As depicted in Algorithm 4.1 the public key and its MIC data frame is sent to the client encrypted by symmetric algorithm which uses the client‟s PTK, which is shared only between the AP and the owner client, as an encryption key and as decryption key at the client side.

At the client, after decryption of the encrypted public key and MIC, authenticity of the received public key will be verified by computing a new MIC of the received public key and comparing with the received MIC of the public key.

```
INPUT:
AP's public key
AP's public key's MIC
Symmetric encryption algorithm
Message Integrity Code (MIC) generator algorithm
Client's PTK
BEGIN
     Encrypted bundle ← Symmetric encryption of the
Concatenation of AP's public key and the MIC of AP's public key
using Client's PTK
IF Encrypted bundle Is Sent to receiver client
     END IF
ELSE
     Send Encrypted bundle to receiver client
END
```

Algorithm 4.1: Proposed Public key and its MIC Sending Process from AP to Authenticating Client

Client on Authentication: As shown in Figure 4.2, is the client who started an authentication procedure to authenticate itself to join the WLAN through the Authentication server in case of an Enterprise mode or directly with AP during a Pre-shared key mode.

In case of the Enterprise mode of the 802.11i standard a WLAN which uses the authentication server to authenticate the identity of clients has already verified the legitimacy of the client on by the time our architecture starts distribution of public key and the MIC from the AP.

By this time the particular client has already proved its identity to the authentication server; whether it is certificate based or user name and password based which entirely depends on the particular EAP method used and even exchanged confirmation with the AP on the generation of

its own unique PTK for the session and also from the same AP it accepts the group shared key, GTK.

So going back to the flow in Figure 4.2, the client on authentication receives a PTK (the client‟s) encrypted packet of two concatenated values which are the public key of the AP and its MIC.

The client would access from its cache memory its unique PTK it generated earlier to decrypt these encrypted received values.

It would decrypt the whole received value, using its PTK as a key value and by implementing the same symmetric algorithm (AES-CCMP) used during encryption; but this time it will use the decryption mode of the symmetric algorithm.

After decryption the client would find the plain public key of the AP and the MIC of the AP‟s public key in concatenation.

After separating the two values the client will have to use the plain public key to generate a MIC value. The client will compare the generated MIC value with the received MIC of the AP‟s public key. If they prove to be the same then it means the received data is correct or uncorrupted during transmission and store the public key information else it would mean the data have been altered by attackers or corrupted during transmission. So if the comparison is not similar then the client would send to the AP a message that notifies the AP of the failure of the received values so that the AP would retransmit the encrypted concatenation of the public key and AP‟s public key MIC to the client. Algorithm 4.2 shows all processes of the authenticating client‟s for receiving the AP‟s public key explained above.

INPUT:

Encrypted bundle //received from algorithm 4.1

Receiver Client's PTK

MIC generator algorithm

Symmetric decryption algorithm

BEGIN

Decrypted bundle ← **Symmetric Decryption** of *Encrypted bundle* using *Receiver Client's PTK*

Separate *Decrypted bundle* into *Decrypted AP's public key* and *Decrypted MIC of AP's pubic key*

New MIC of AP's Public key ← **Calculate MIC** of the *Received Decrypted AP's public key*

IF *Decrypted MIC of AP's public key* **Is Equal to** *New MIC of AP's Public key*

    Save *Decrypted MIC of AP's public key*

ELSE

    Drop *Decrypted bundle*

END

Algorithm 4.2: Proposed Initial Authentication of Public key Sharing Process, with AP, at Authenticating Client

4.4 Proposed System Architecture of Group Addressed Data Frames Communication between the AP and an Associated Client during a Session

The successful authentication between the client and the authentication server is followed by the exchange of keys and confirmation messages between the AP and the authenticated client to generate and share keys which, in turn, enable them encrypt management and data frames when the session begins. In our initial architecture of Figure 4.2 we have shown that the AP distributes its public key along the MIC to the authenticated client. The GTK along its sequence number is also sent to the authenticated client and generate similar PTK are generated at both sides. Regarding PTK generation the existing 802.11i standard‟s mechanism is assumed.

After the end of the authentication, a client will be legitimate to ask for an association which is the last phase in the 802.11i authentication. The association request phase does not particularly ask for credentials or something similar as is during in the authentication phase but the client will check for the capabilities provided/beaconed by the AP and if match is found then the client will be associated right away else client will be illegible to communicate with the WLAN. An associated client will be legitimate to send/receive data traffic to/from the WLAN it is associated with.

So assuming that a client succeeds in associating with the WLAN, we will start implementing the proposed mechanism to prevent the GTK misuse by authenticated clients during the session by enhancing the group addressed data frames in the current 802.11i standard.

It is during this valid session that an associated client starts sending data traffic to the WLAN. An associated client sends its group addressed traffic to the WLAN in exactly the same way as it does in the WPA2 security protocol of the current 802.11i standard.

In our proposed session architecture, the receiver, AP, will send group addressed data frames to receiver peers and peers use proposed mechanism which enable them discern whether the received group addressed traffic is a potential fraud from another peer or real one that came through the AP.

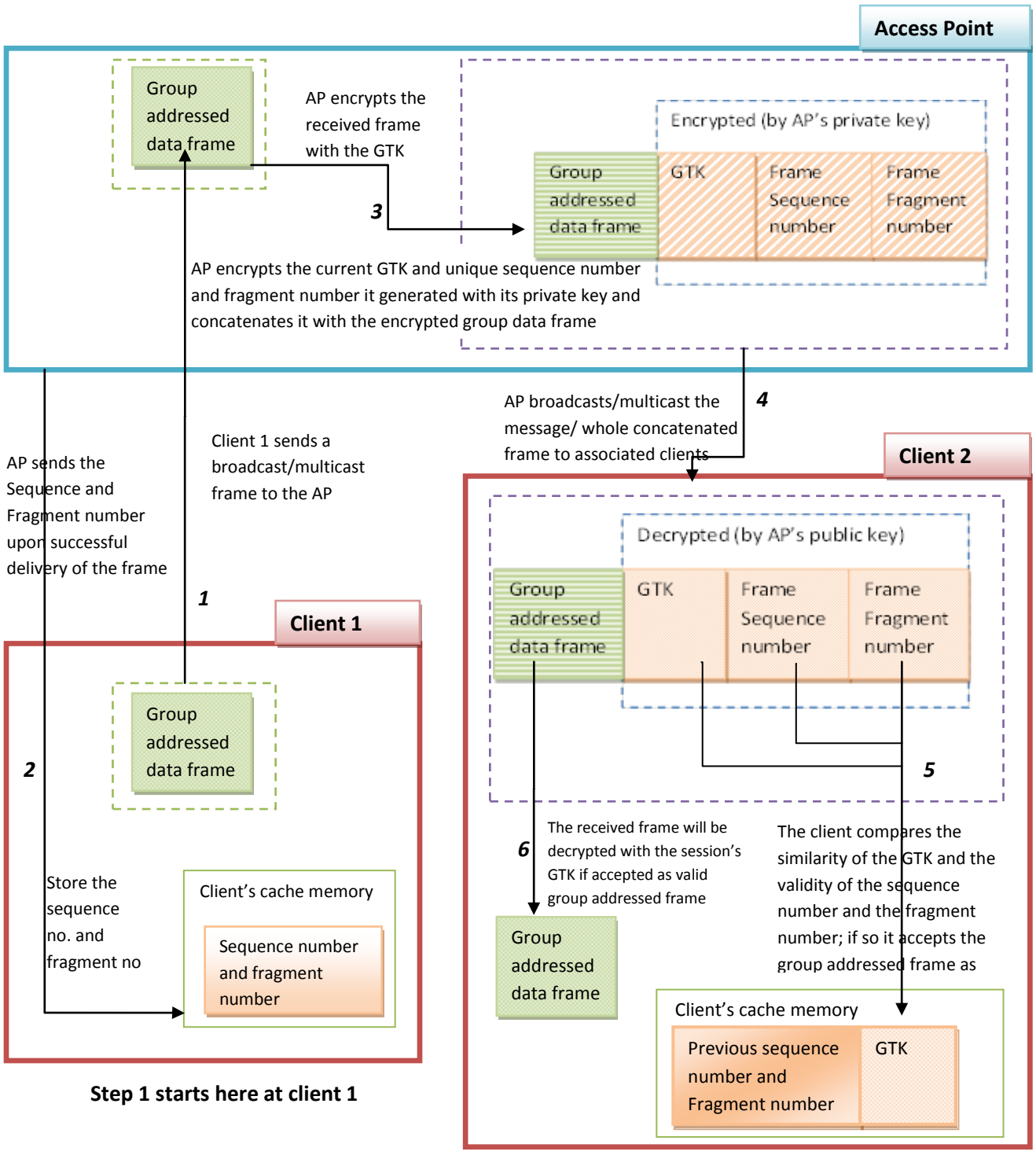

Figure 4.3: Architecture of the Proposed Session between the Associated Clients and the AP

The architecture of the proposed session between associated clients and the AP will be elaborated with a broadcast message transfer within the WLAN. Our architecture implements a mechanism which enables broadcast/multicast message recipients (associated peers) identify if, indeed, a received group addressed data frame/traffic is coming through the AP.

Techniques adapted/reused directly from the IEEE 802.11i standard into the proposed session's architecture in Figure 4.3:

1. The sending mechanism in the sender client (client 1 in Figure 4.3).
2. At AP, the symmetric encryption (using GTK as its encryption key) of a broadcast/multicast data frame which is received from the sender client (at AP in Figure 4.3).
3. The symmetric decryption, using the GTK as a decryption key, of broadcast/multicast data frame at the receiver client (at client 2 in Figure 4.3).

Components of the Proposed Session Architecture

In this thesis the term client refers to a member of the session, an associated client, who joined the network by providing all the necessary credentials. Associated client can send and receive data to/from the network within the session.

Sender Client

Sender client is the associated client who wants to send a broadcast/multicast message to other associated clients. Any communication within the WLAN is done through the AP. We will use the terms „sender" and „client 1" interchangeably.

The sender is expected to store the sequence number and fragment number of the broadcast/multicast frame it is going to send to the AP in its cache memory. The reason for this is, next time a group addressed data frame broadcasted/multicast from other peers arrives at the client 1 through the AP, it would use the sequence number and fragment number in its cache to compare if the received group addressed frame is not a replay attack from other peers. This point will be elaborated more at the receiver client"s section.

Client 1 will encrypt the group-addressed data frame that it wants either to broadcast or multicast with its unicast session key, PTK, using the symmetric algorithm used in WPA2, AES-CCMP, and sends it to the AP.

Access Point

According to the IEEE 802.11 infrastructure WLAN standard specification every communication within the network is supposed to be transmitted through the AP.

The AP will receive any message from its associated clients and transmit that message to a recipient client in case of a unicast message; and transmit a given message to the rest of the associated clients in case of broadcast messages or transmit only to the clients that are in the recipients‟ range stated in the multicast message destination addresses range.

As shown in the proposed session in Figure 4.3 the AP will receive a broadcast data frame from a sender client (client 1). Then the AP will encrypt the received broadcast/multicast data frame using the session‟s GTK. The encryption process will be done in the same way, using the symmetric encryption using the AES-CCMP algorithm, as it is done in the existing WPA2 Protocol. This encryption process is totally adapted from the existing WPA2 protocol. As already indicated in the literature review Section where we explained about the GTK vulnerability, in the current 802.11i protocol there is no mechanism to make sure if a given message is coming through the AP or not which as a consequence made the receiver clients open to attacks of fraud data frames, particularly in case of the GTK misuse (Hole 196 vulnerability), which falsely claim to have come from/through the AP where in reality they are just malicious data frames coming from another associated peer. So in order to prevent such attacks we considered the AP as an ideal place where we can start implementing our solution.

Unicast data frames sent to/received by peers in the WLAN are encrypted and decrypted with the user‟s unique key, PTK. A given PTK is only known by the owner authenticated client and the AP. This forces all associated clients who want to send unicast data frames to other peers to send it first directly encrypted with their own PTK to the AP since it is only the AP which has knowledge of the PTKs‟ of all associated peers would decrypt the received unicast frame and then encrypt it using the recipient client‟s PTK to encrypt it and then send it. Receiver clients can only decrypt unicast data frames encrypted with their own PTKs. So any client, in order to forge

a valid unicast data frame to be sent to other peers, would first need the knowledge of other peer‟s PTKs.

Before explaining the rest of the processes in the AP let us explain separately what a frame sequence number, frame fragment number and GTK are and their importance in relation to our proposed design.

Sequence and Fragment Numbers

A sequence number indicates the sequence/ID number of each frame sent/received across the network so that frames can be uniquely identified among other frames. In IEEE 802.11 MAC frame format the sequence number remains the same for each of the frames sent as fragmented frames; otherwise it is incremented by one until reaching 4095, when it then begins at zero again. One sequence number has a size of 12 bits [33].

A given frame can be fragmented into separate frames or fully sent at once during transmission which depends on the size limit of maximum number of bits allowed to transfer in a single transmission. Similarly a fragment number, in IEEE 802.11 MAC frame format, indicates the assigned number to identify each fragmented frame. The initial value is set to 0 and then incremented by one for each subsequent frame sent of the fragmented frame. The size of a fragment number is 4 bits [33].

The frame structure at MAC or data-link layer have various information attached to it which range from the Frame control field to the payload of the frame. In IEEE 802.11 MAC frame we have a 16 bits field named, *sequence control*, which holds the data frame‟s sequence number and fragment number [49].

There is also a unique field, named *morefrag*, which takes a Boolean value of either 1 or 0 to indicate that whether a given data frame is fragmented or not. If the value of field morefrag=0, then the data frame is not fragmented else if 1 it means the data frame is fragmented.

Going back to the diagram of the AP at Figure 4.3, we have the current session‟s GTK, the sequence number and the fragment number of the received group-addressed data frame concatenated.

The combination of the session‟s GTK, the frame‟s sequence number and fragment number generates a unique number that can only be associated with the particularly received group-addressed frame. This combination is not only unique for every frame within the given session but is also unique when compared with combinations generated for frames within other sessions too as long as a fresh GTK is generated in every session. This helps prevent replay attacks.

Asymmetric Encryption Process

The PKI contains two mathematically related keys called private and public key pairs. These pairs are owned by the AP as sent from the AS. The private key remains secret in the AP but the public key is shared with all the associated clients during the initial authentication phase of our proposed architecture of the initial distribution in Figure 4.2.

In the proposed session scenario in Figure 4.3 the AP encrypts with its private key, using an asymmetric cryptography, the unique combination (the concatenation of the GTK, sequence number and fragment number) generated. This encrypted unique combination packet will be concatenated along the Group-addressed data frame encrypted by the GTK. Then their full concatenation will be broadcasted/ multicast, as a frame, to all the clients in the associated network in case of broadcast frame and to clients who are in the recipients‟ range during multicast communication. Algorithm 4.3 explains these processes AP in a step by step manner.

The asymmetric encryption algorithm can possibly be any standard asymmetric encryption algorithm. Some of such algorithms include RSA, ECC etc. But this is basically a flexible implementation dependent issue.

INPUT:

Group addressed data frame // received from sender client

AP's private key

Symmetric Encryption Algorithm

Asymmetric Encryption Algorithm

Client's PTK

Current GTK, Sequence number, Fragment number

BEGIN

Asymmetric Encrypted bundle ← **Asymmetric Encryption** on the **Concatenation** of *current GTK, sequence number* and *fragment number* using *AP's private key*

Symmetric Encrypted bundle ← **Symmetric Encryption** on the *Group addressed data frame* using *Current GTK*

Group addressed encrypted bundle ← **Concatenate** *Asymmetric Encrypted bundle* and *Symmetric Encrypted bundle*

IF *Group addressed encrypted bundle* **is sent** to receiver client

    END IF

ELSE

    Send *Group addressed encrypted bundle* to receiver client

END

Algorithm 4.3: Process of Proposed Broadcasting or Multicasting of Modified Group Addressed Data Frames from AP to Receiver Clients

Receiver Client

Any broadcast message in the AP will be sent to all the associated clients. Client 2 (Receiver) as depicted in Figure 4.3 represents all associated clients in case of a broadcast message and the recipient group of associated clients in case of a multicast messages. In the proposed session architecture of Figure 4.3 all the proposed mechanisms supposed to be done by a receiver client of a group addressed data frame are implemented in client 2. So we will use the terms „receiver client" and „client 2" interchangeably.

At the receiver client end, the received group addressed frame includes a group-addressed data frame encrypted by the session"s GTK along a concatenation of the session"s GTK, the frame sequence number and its fragment number which are encrypted as a bundle with the AP"s private key using an asymmetric encryption in the AP.

Asymmetric Decryption Process

Client 2 first decrypts the bundle of GTK, sequence number and fragment number which was encrypted with the AP"s private key at the AP, using the same asymmetric algorithm (AES-CCMP) but this time using the decryption flow. After decrypting the plaintext concatenation of GTK, the frame"s sequence number and fragment number of the frame will be available.

Recalling the fact, from Figure 4.2, that every associated client has a copy of the session"s GTK in its memory, client 2 compares the just received and decrypted GTK with the one in its memory and if they are found to be similar it will further go to the next phase of checking the validity of the received group addressed frame as will be explained in the next Section. If the GTKs compared are not similar then the received frame will be dropped. Algorithm 4.4 depicts in detail all processes and mechanisms for checking the authenticity of a received group addressed data frame at the receiver client before it can be accepted as a valid group addressed message.

INPUT:

Group addressed encrypted bundle // received from algorithm 4.3 (from the AP)

AP's public key, Symmetric decryption algorithm, Asymmetric decryption algorithm, PTK

Current GTK, Sequence number, Fragment number

Begin

Separate *Group addressed encrypted bundle* into *Asymmetric Encrypted bundle* and *Symmetric Encrypted bundle*

Decrypted Asymmetric bundle ← **Asymmetric Decryption** on the *Asymmetric Encrypted bundle* using *AP's public key*

Separate the *Decrypted Asymmetric bundle* into *GTK*, *Sequence number* and *Fragment number*

IF *GTK* and *current GTK* **Are Equal** and

    IF *sequence* and *fragment numbers* are in **valid range**

        Current sequence number ← *Sequence number*

        Current Fragment number ← *Fragment number*

        Decrypted Symmetric data ← **Symmetric Decryption** on the *Symmetric Encrypted bundle* using *PTK*

    ELSE

        DROP *Group addressed encrypted bundle*

ELSE

    DROP *Group addressed encrypted bundle*

END

Algorithm 4.4: Proposed Mechanism of Verifying and Accepting Group Addressed Data Frames from AP; at Receiver Clients

Replay Attack Detection through Frame Sequence and Fragment Numbers

Assuming the phase for comparing the similarity of the received GTK with the one in the receiver client‟s memory confirms they are similar, the next phase would be comparing the stored sequence number and fragment number with the just received and decrypted pair of sequence number and fragment number.

The sequence number comparison is useful to check if a group addressed message received is not a replay attack. Every newly received group addressed message is expected to have a new sequence number (or in an incremental increase) which the user would not be able to calculate the cipher without the knowledge of the AP‟s private key.

It is true that the sequence numbers are re-used during every new session but this would not enable attackers to do replay attacks by storing previous sessions‟ group-addressed data frames. The reason is, although the sequence numbers are reused, a new GTK is generated for each session. As a result, the combination of the sequence number and the GTK would always yield a new cipher which would make replay attacks impossible. But this scenario assumes the frames are non-fragmented.

A fragment number comparison mechanism along the sequence number checking mechanism is useful to detect potential replay attack on fragmented data frames. Since fragments of the same frame share same sequence number, the sequence number based replay attack detection alone is not sufficient.

In order to mitigate such possible replay attacks, we use the fragment number of fragmented data frames in MAC frame. Each fragmented frame of a given data frame will have a unique combination of a sequence number and fragment number.

Detail of the data frames sequence and fragment number comparison mechanism at the group addressed data frame s receiver client

The main purpose of comparing the sequence and fragment numbers is for detecting any attempts of replay attacks within the session (more specifically sequence number comparison ensures that data frame level replay attacks within the session and fragment number comparison prevents fragment level replay attacks (within a data frame) within the session).

The comparison of the GTK, sequence and fragment number can be varied depending if the AP is sending a broadcast/multicast message to the clients for the first time in the given session. If the AP is sending a group addressed data frame for the first time, then the comparison of sequence and fragment number will be excluded at the receiver clients.

So if the group address broadcast is for the first time clients will only check the similarity of the session's GTK. (The stored one and the, just, received one). Then if valid, the recipient clients will store in their cache memory the newly received sequence number and fragment number (if available) values to be used for next time when new broadcast/multicast data frames are received.

Here one point to remember is that the sender client always stores the sequence number and fragment number values in its cache memory. If it is not the first sender which means if it had already sequence and fragment number values in its cache, then it just overwrites the old values. This mechanism helps a sender client to always synchronize with the other peers.

But, of course, if the group addressed data frame is not the first for the client then the comparison will include all of them; GTK and sequence (fragment number also during fragmented frames).

One more thing to note here is that a client is its first time to receive a group addressed data frame does not necessarily indicate that a group addressed data frame transmission in the network is for the first time too. Since group addressed data frames can also mean multicast data frame, besides broadcast data frames, and such data frames might be destined only to selected clients within the given network.

So in this case for a group addressed data frame to be accepted as valid:

The GTK must be similar (the one stored and the just received one which came with the group data frame concatenated) and the sequence number must be greater than the one in the receiver clients' cache. But if the data frame has a fragment number then the sequence number used will be the same till the end of the fragments. So in fragmented frames the sequence number is assumed implicitly to be equal so there is not sequence number comparison but only it will be compared and overwrite if valid only during the arrival of the first fragment of the data frame. But the fragment numbers of all the fragment frames of a data frame overwrite the existing one in cache. Only during the arrival of the first fragmented data frame is a sequence number

comparison is done and during this time no fragment number comparison but it will directly overwrite any value in the cache whether fragment number of a previous fragmented data frame or a default value, 0, of non-fragmented data frame.

Then after only fragment comparison and, if valid, overwriting the previous fragment of the same data frame will be done. During an arrival of a new data frame, if non-fragmented, the sequence number will be checked and if valid it will overwrite the older one and the fragment number will be changed to default value for non-fragmented "0". But if the new data frame that arrived is a fragmented one then both the sequence number and the fragment number cache will be overwritten by the new one. Normally the first fragment number of a data frame starts from 1 and then increments by 1. Fragmented frame data frames' sequence numbers are stored only once as the same sequence number is used for all fragments of a given data frame.

The fragment numbers‟ comparison is not initiated unless the data frame has fragments. But if it has fragments, the fragment number comparison will be done until it finishes. When it finishes the old values will be replaced with the values of a new data frame if it is accepted as a valid group addressed data frame. If it is not a fragmented data frame, the value of the fragment number in the cache will be reset to 0, which indicates that the data frame is not fragmented.

In case of fragmented frames, the value of the sequence number is the same so it is not necessary to overwrite the existing one with the just received one. Fragment number will start from 1 and increment for the next ones.

The sequence number will be compared with the older one in the cache only once; during the arrival of the first fragment. The fragment number will be stored during the first fragment arrival and during the next fragments arrival it will be compared with the existing one and if correct it will overwrite the older ones and the group data frame will be received as valid but only assuming that the GTK comparison is succeeded.

The receiver knows whether a received frame is fragmented or not using the Boolean value of the "*More fragments*" in the Frame Control Field within the MAC Header [33].

CHAPTER 5: IMPLEMENTATION AND VALIDATION

5.1 Overview

Dealing with the details of the proposed system designs in the previous chapter, here we discuss the validation of the designed solution in a prototype. Moreover, we will discuss the general design of the prototype, tools and technologies used to develop it. The prototype we developed aimed at providing proof of the concepts presented in the detailed explanation of the architectures of the proposed system design. To achieve this principal goal, the components in the architecture have been implemented as per the theoretical specifications formulated in the proposed system.

5.2 Development Environment

The prototype is implemented using the OMNet++ (version 4.6) simulation software which provides the C++ programming language as its implementation language. We chose the OMNet++ simulation software for its relevance to implement our proposed system. Moreover implementation on the OMNet++ IDE gave us all of the advantages of the C++ programming language to implement the logic of the proposed system.

5.3 Implementation Details

We have developed two simulations; one is a partial simulation of the existing IEEE 802.11i security protocol (which is also vulnerable to the GTK misuse by authenticated clients). The other is the simulation of our proposed system (one which modifies the existing security protocol to prevent the GTK vulnerability).

The simulations only considered important factors which can bring difference between the existing and proposed system‟s simulations. Mechanisms in the proposed systems but that are adapted directly from the existing 802.11i are not simulated. The reason is that re-creating a simulation of the existing IEEE 802.11i security protocol, the WPA2, is a very bulky job; so considering the specific goal of this thesis it is unnecessary to re-implement every detail of the existing protocol in a simulation when only a very small part of the protocol is considered for modification by the proposed system.

Important specifications of the simulation process:

1. Both simulations will contain a similar network topology. This means the number and type of devices, transmission bandwidth and delay will be similar in both simulations.
2. Both simulations will consider all client nodes in the topology are authenticated to the network. The reason to assume only authenticated clients is because the Hole 196 vulnerability (GTK misuse) is an insider attack and can only be caused by malicious authenticated users.
3. Equal number of group addressed frames will be transmitted in both simulations.
4. Equal number of fake group addressed frames will be transmitted in both simulations.
5. Equal simulation or CPU time will be used to test the results of both simulations

Description of topology

- One ***Access Point*** *(*AP*)* – It will have the role of forwarding every valid group addressed data frames to receiver clients that come to it from sender clients.
- ***Client one*** (client 1) – Is sender client; it broadcasts or multicasts messages to other clients in the WLAN through the AP.

PS. *the names inside the brackets are the names of the devices used during the simulation.*

- ***Client two*** (client 2), ***Client three*** (client 3) and ***Client four*** (client 4) - are recipients of the broadcast or multicast messages received through the AP. Here ***client three*** has a role of ***malicious authenticated client***: means at some point during the session it sends fake group addressed data frames which are encrypted with the GTK to ***client two***.

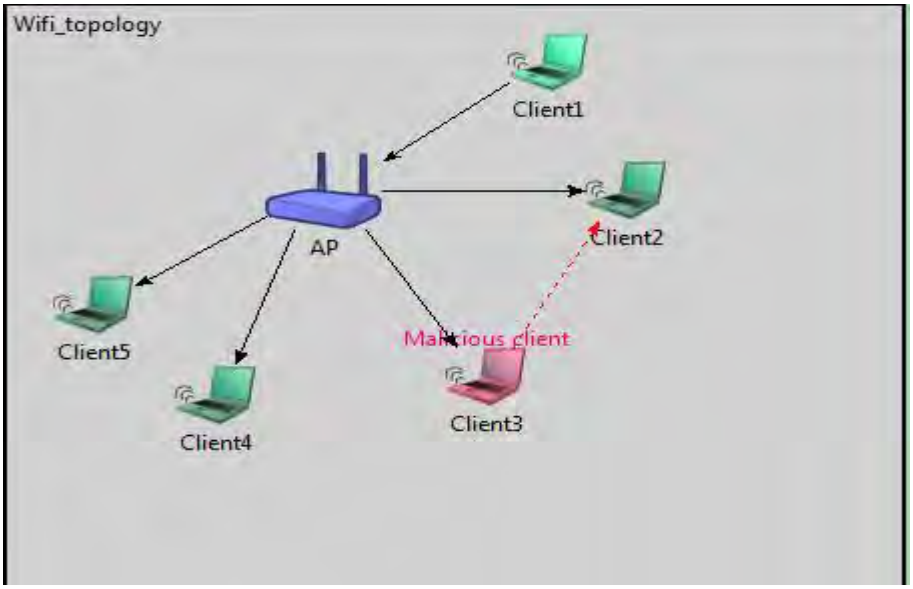

Figure 5.1: Snapshot of the Network Topology in the Simulator (similar for both simulations)

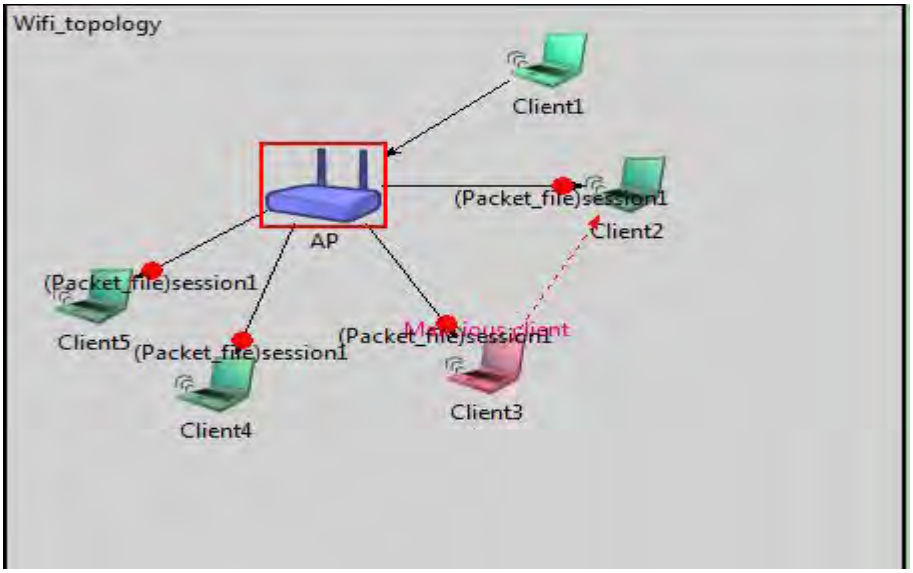

Figure 5.2: Snapshot of an Access Point Broadcasting a Data Frame to Receiver Clients (clients 2, 3, 4, 5)

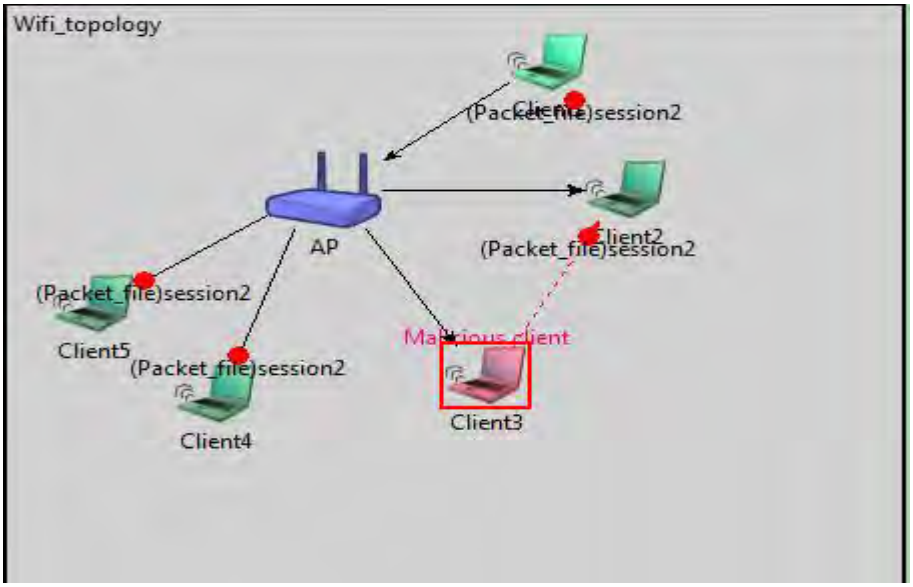

Figure 5.3: Snapshot of Client 3(malicious client) Sending Fake Group Addressed Data Frame to Client 2

5.4 Implementation of the Proposed IEEE 802.11i Enhancement

In our implementation all of the clients in the simulated topologies are considered as authenticated clients and this makes them legitimate to have access to the public key. The main reason for not considering unsuccessful authentication or illegitimate clients is that the prime aim of this research is to avoid the misuse of GTK between authenticated clients. On the other hand Private Key of the AP is solely to be accessed by the AP. So in our implementation the private key is not open for any of the modules except for the AP‟s.

Implementation of the Proposed Session between the Associated Clients and the AP

According to our proposed system design clients are allowed to access the WLAN‟s resources freely only after they are authenticated, received the AP‟s public key and associated.

The session for a client starts when it is allowed to access the WLAN fully. The proposed system‟s algorithm validation is checked by simulating a session where the AP and authenticated clients communicate and inject fake group addressed data frames which would be accepted as valid data frames in the existing security protocols of the IEEE 802.11i standard.

Modules Implemented:

We implemented the proposed solution design within two modules; one in the AP and other in the authenticated client.

Access Point s Module

1. It has knowledge of (stores) values of:
    - The PTKs of all associated clients
    - GTK
    - The Private, Public key pair and the Modulus
2. Implements:
    - Symmetric encryption and decryption algorithm
    - Asymmetric encryption algorithm (the RSA encryption algorithm in our implementation)
3. Functions in their order:
    - Decrypts a group addressed data frame sent to the AP from a sender client using the client‟s PTK with a symmetric decryption algorithm, AES-CCMP. Then it assigned the data frame a Data frame number (Sequence number) and Fragment number (if it is fragmented). But default Fragment number is 0.
    - Then it sends back the sender client the data frame number and fragment number it assigned to the group addressed frame it received from it.
    - It encrypts the group addressed data frame, decrypted earlier, with the symmetric encryption algorithm using the session‟s GTK as an encryption key.
    - It concatenates the session‟s GTK with the newly assigned data frame number and fragment number. It encrypts this bundle with the asymmetric encryption algorithm using the private key as an encryption key.

- It aggregates the encrypted bundle of the GTK, data frame number and fragment number with the encrypted group addressed data frame. It also attaches the necessary frame header files.
- Finally it broadcasts or multicasts the aggregated frame to receiver clients.

Segment of implementation code is shown in annex I.

Authenticated Client s Module

1. It has knowledge of (stores) values of:

- Each client has knowledge of its own PTK
- GTK
- The Public key pair and the modulus

2. Implements algorithms:

- Symmetric encryption and decryption algorithm
- Asymmetric decryption algorithm (the RSA decryption algorithm in our implementation)

This module has two main sub modules; one used during sending a group addressed data frame through the AP and the other when it receives a group addressed data frame.

3. **Sending sub module**: It encrypts the group addressed data frame it wants to broadcast or multicast with its own unique PTK using the symmetric encryption algorithm. Then it sends this data frame to the AP which will prepare the data frame and forward it to receivers. Segment of implementation code is shown at annex II.

4. **Receiving sub module**: It decrypts the received group addressed data frame using the asymmetric decryption algorithm using the public key (and the parameter modulus) it has in its memory as a decryption key. Then upon decryption it compares the GTK, data frame number (sequence number) and fragment number with the ones in the memory. Segment of implementation code is shown in annex III.

If the comparison proves the received data frame as valid then decrypt, the group addressed data frame encrypted with the symmetric algorithm, using the session‟s GTK as a decryption key

using the symmetric decryption algorithm. If not the data frame will be dropped as it can be a fake group addressed data frame sent from malicious peers.

5.5 Validation Results

To validate the proposed system‟s architecture, we sampled multiple group addressed data frames communication scenarios, which differ in size of the data frames, between the clients through the AP during both simulations.

Validation of the proposed solution and failure of the existing approach to detect malicious group addressed frames was confirmed in a scenario where simulations of both the proposed technique and the existing one were set with equal number of frame transmission, equal number of normal and malicious clients and access point with similar network configuration. Validation was done by sending each client valid group addressed frames; some were legitimate (which have been processed by and passed through the AP) and others were fake group addressed data frames sent by a malicious authenticated client directly. The frames sent during the proposed simulation legitimate group addressed frames contain, as already discussed in the proposed system design section, symmetric encryption of the group addressed data frame by the current session‟s GTK concatenated along an asymmetric encryption of the GTK, frame sequence and fragment numbers by the AP‟s private key while during the simulation of the existing implementation of legitimate group addressed frames would contain only symmetric encryption of the group addressed frame by the session‟s GTK.

During both cases malicious group addresses data frames were built in a way that they can fool the existing WPA2 implementation. This means each malicious client would use the correct header information such as GTK, data frame sequence and fragment number (if fragmented) and send a symmetrically encrypted group addressed data frame to victim clients directly. This is the best scenario for attack simulation since all we wanted to know is if the proposed solution can really detect malicious group addressed data frames.

Results from the simulation of the existing implementation showed that they are vulnerable to such attacks by accepting malicious group addressed data frames as valid. Since it only checks for the correctness of GTK to classify it as a valid group addressed frame. But simulation of the proposed solution dropped such data frames as they output an invalid value (invalid GTK, frame

sequence and fragment number) when the receiver client decrypts them using the public key of the AP shared earlier. The result can be attributed to the lack of knowledge AP‟s private key by authenticated clients (including malicious ones).

In the simulations, we have sampled up to 20,000 (in a range of numbers) group addressed data frames communication during both the existing and proposed mechanism simulations where 10 percent of these data frames were forged group addressed data frames sent directly from peer clients to victim clients. The forged data frames contain all the necessary header information found in the valid group addressed data frames found in current WPA2 implementation. This is possible because in the existing WPA2 security protocol authenticated clients have access to all necessary information to create a valid but forged group addressed data frame. Each client knows the session‟s GTK and data frame numbering information. So all needed is to do symmetric encryption on the data frame and attach updated header information to the group addressed data frame before sending it. Results of the existing mechanism showed all of the forged broadcasted data frames were accepted as valid group addressed data frames.

On the other hand, simulation of the proposed mechanism showed 100 percent of detection of forged group addressed data frames sent from peer clients. The forged data frames were processed in the same way as done in the simulation of the existing mechanism while the proposed valid group addressed data frames‟ GTK, sequence and fragment number information are asymmetrically encrypted by the AP‟s private key. Thus, malicious authenticated client need the private key information in order to forge a valid group addressed data frame since a receiver client will apply an asymmetric decryption using the AP‟s public key it has and makes sure if the GTK is the same as the session‟s and checks validity of sequence and fragment number before accepting it as a valid group addressed data frame.

Our proposed system is a complement to the WIDS or WIPS solutions implemented in the AP. The proposed solution cannot prevent attacks that come through the AP as long as they are valid group addressed data frames. Such attacks can include a payload on the data frame or ARP poisoning etc. Such attacks are to be addressed with WIPS or WIDS solutions implemented in the AP or on some device in the pathway. The reason is our solution only provides prevention against fake group addressed data frames which are forged at the client and sent directly but if they are sent through the AP it will mean that they will have a GTK, data frame sequence and

fragment number encrypted with private key of the AP which is the sole mechanism of our solution to eradicate forged group addressed data frames.

Implications of directly sent attacks are vast as described in the literature review‟s section where we explained attacks possible through the Hole 196. We can take ARP cache poisoning attacks, as an example, which initially are sent directly but then, after poisoning client‟s address cache, use the passage through the AP when the AP will participate in forwarding attacks or where it serves as a pathway for MITM attacks.

In case of the existing implementation simulation every group addressed data frames encrypted with the GTK and also accompanied with the valid sequence and fragment number would be accepted as a valid group addressed data frame whether it is coming really through the AP or it is just a fake group addressed data frame sent from another authenticated (possibly malicious) client.

5.6 Computational Cost

The simulator limited us to use maximum keys sizes, only, up to 19 bytes (152 bits) private key exponent and modulus value. Anything larger than this would result in a memory buffer overflow in the simulator. So sizes of PKI, RSA algorithm, keys used during the simulation are: Pubic key 56 bits, Private key = 152 bits, Modulus=152 bits.

We didn‟t consider the memory overhead since the proposed system demands at the AP a storage of a single public key, private key pair along the modulus and at the clients‟ machine only the public key and the modulus. So this logically can‟t incur any significant memory overhead to the devices since modern devices have a memory capability to store or cache much larger data.

We compared the proposed solution with the existing mechanism in terms of Performance (P), Event density (E) and Relative speed (R). Both Performance and Relative speed depend on the performance of the device used. But this doesn‟t affect the quality of our measurements as all we need is a relative comparison between the existing and proposed solutions.

Definition of Measurement Parameters Used in Simulation

Performance (P) represents the number of events processed per second (**ev/sec**). P depends on the performance of the hardware and the computation-intensiveness of processing an event. P is independent of the size of the model [51].

Event density (E) is the number of events that occur per simulated second (**ev/simsec**). E depends on the model only, and not where the model is executed. E is determined by the size, the detail level and also the nature of the simulated system [51].

Relative speed (R) measures the simulation time advancement per second (**simsec/sec**). R strongly depends on both the model and on the software/hardware environment where the model executes. Note that R=P/E [51].

All the experiments here were simulated in a 2.13 GHz laptop. The number of frames sampled in a single simulation in minimum was 3000 and maximum 20,000. Using smaller than 3000 frames makes the outputs insignificant in terms of showing the pattern of results of the simulations.

Table 5.1: Proposed Solution Results

Data frames size	3000	6000	10000	20000
Performance (ev/sec)	1099.35	1087.42	1070.69	1085.77
Event density (ev/simsec)	30.2041	30.1393	30.0529	30.1571
Relative speed (simsec/sec)	36.3974	36.0674	35.6268	36.0038

Table 5.2: Existing Mechanism Results

Data frames size	3000	6000	10000	20000
Performance (ev/sec)	1975.49	1969.03	1960.41	1945.33
Event density (ev/simsec)	20.0574	20.0687	20.0772	20.0583
Relative speed (simsec/sec)	98.492	98.128	97.6437	96.984

Table 5.1 and Table 5.2 show the results of Performance, Event density and Relative speed of both the proposed and existing systems on a simulated environment respectively.

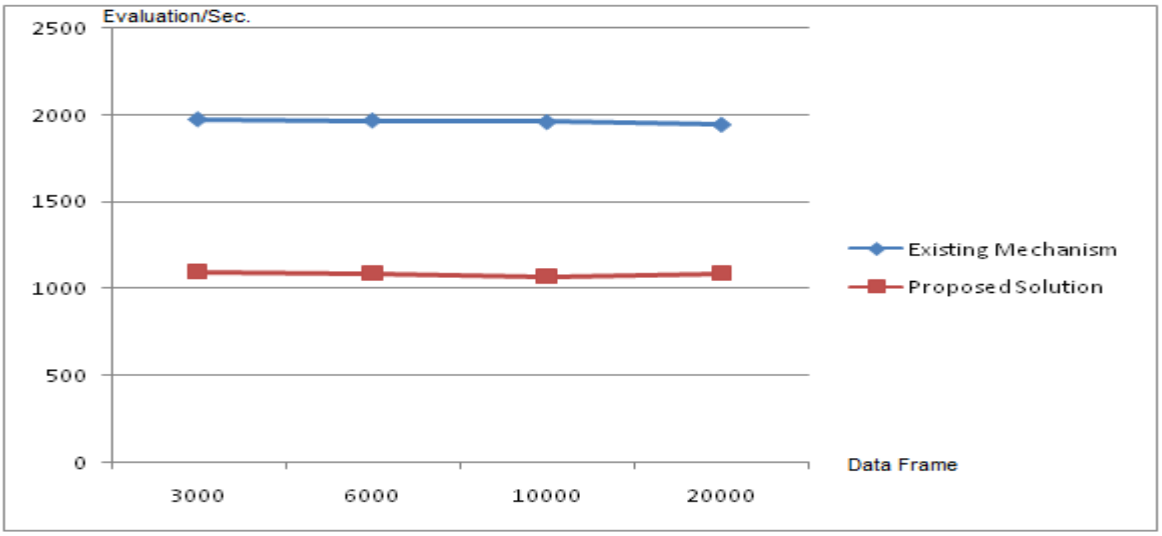

Figure 5.4: Performance Evaluation for Existing Mechanism and Proposed Solution

Figure 5.4 shows performance measurement in terms of events processed per second. The results depend on both the hardware and the computation intensiveness. The existing systems performance is much better and is almost in a parallel relation to the proposed solution but as the number of frames increase performance measurement of both systems start sliding towards each other; makes performance result of the proposed solution improve.

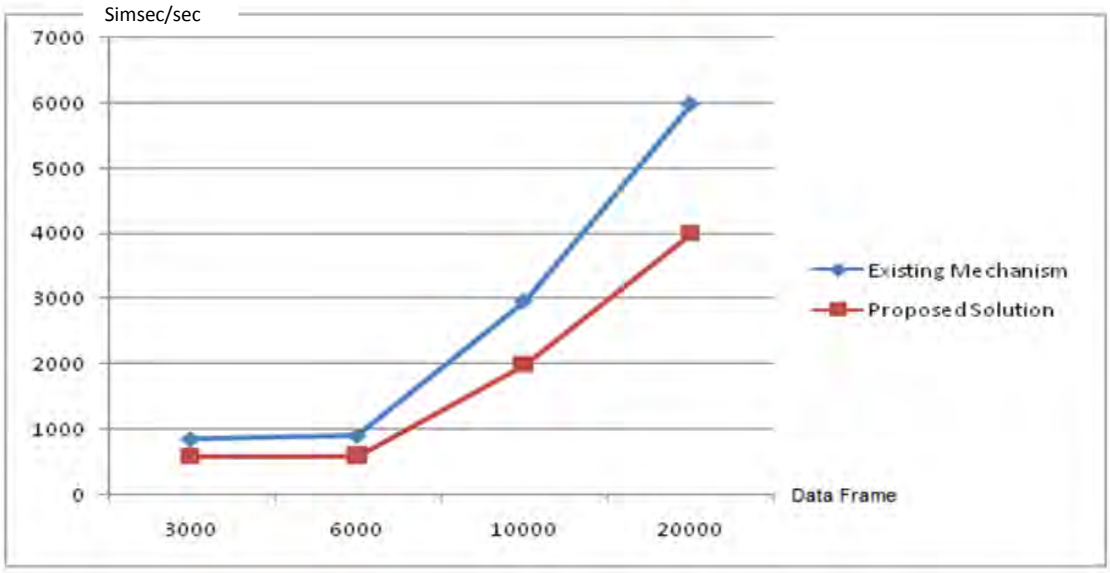

Figure 5.5: Relative Speed Comparison

Figure 5.5 shows the Relative speed measurement in terms of simulated time advancement per second (simsec/sec). Simulations of both systems show an increase in the relative speed as the time goes; with some speed advantage in the existing system.

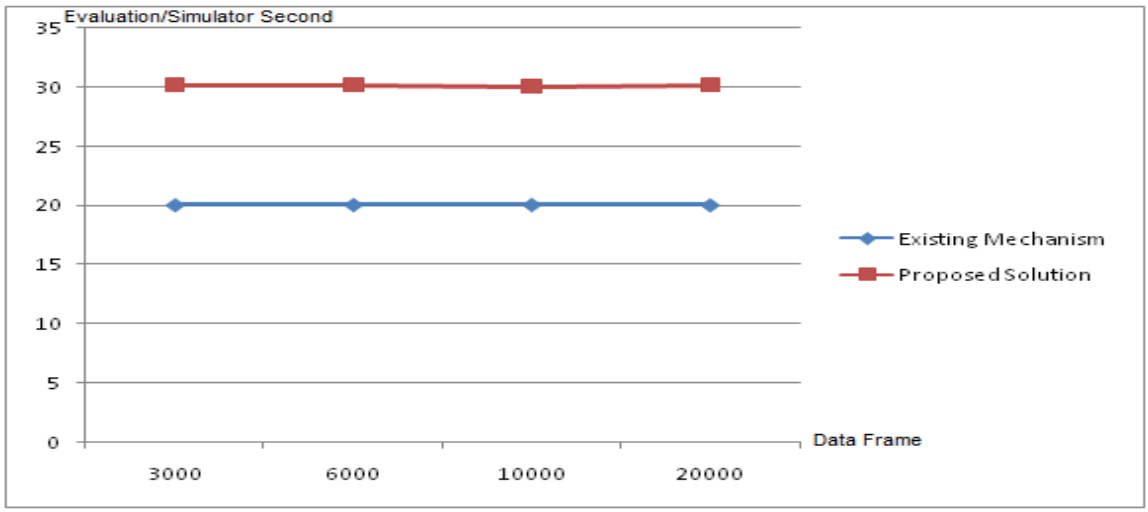

Figure 5.6: Event Density Measurement for the Existing Mechanism and Proposed Solution

Figure 5.6 shows the measurement of the number of events that occurred per simulated second (Event density). The graph shows a relatively parallel relation of the simulations and there are of course fewer events in the existing mechanisms.

Even if there is relatively more event density in the proposed solution‟s simulation (some speed advantage in the existing system), it is also very tolerable considering today's devices processing speed.

The size of the bits needed to be encrypted by the PKI can be highly decreased to much lower than 144 bits in case of CCMP and 278 bits during TKIP. This is possible by cutting the GTK key that gets affected by the PKI. In case of CCMP the 128 bits size GTK can be split in to half or more slices and then encrypting only the half part or any lower than half based on their initial agreement. Then the client too does not have to decrypt the whole GTK and compare fully it with the cached one but after decrypting it will get either half of the GTK used or lower; so it will only use these values and compare it with only part of the cached GTK.

So based on this logic the number of total bits to be encrypted and decrypted by the PKI, in case of WPA2 can get decrease from 144 to 80 bits (12+4+64), 48 bits (12+4+32) or much lower. Similarly, for WPA values can decrease from 278 to 144 bits (12+4+128), 80 bits (12+4+64), 48 bits (12+4+32) or much lower.

The reason why the slicing of the GTK, for PKI, up to some reasonable level is secure is because the PKI encryption done on GTK in concatenation with the sequence and fragment numbers which makes the encrypted output totally unpredictable. Moreover, the GTK is a session based value which changes randomly during every new session or even when a client dissociates or deauthenticates within a session or when AP‟s trigger time for changing GTK is up. So the GTK has a very high probability for a random change which would make the smaller bits encrypted and decrypted through PKI stay safe.

CHAPTER 6: CONCLUSION AND FUTURE WORKS

6.1 Conclusion

This thesis introduced a solution that tackles the issue of the per-frame source authentication in case of broadcast and multicast data frames in WLANs by integrating a PKI mechanism. This avoids the GTK vulnerability in WLANs. The obvious drawback of PKI is a high computational cost. So in order to this cost, PKI has been applied only partially on control information that can be attached to the main data frame. So it leaves out the data/payload part of the frame for symmetric cryptography (as it is done in WPA/WPA2 currently). Thus putting, the parts of a frame that involve PKI, numerically: the GTK (during CCMP the size of GTK is 128 bits), sequence (12 bits) and fragment numbers (4 bits) which together sum up to 144 bits only during the WPA2 and 278 bits during a WPA as TKIP uses a 256 bits GTK.

So the PKI overhead is limited to 144 bits encryption and decryption in case of WPA2, the latest security protocol, which can prove insignificant in terms of computational cost as devices that participate in infrastructure based WLANs are capable of processing much more without the user noticing much overhead.

6.2 Future work

In the future, to lower the computational cost we plan to:

- Implement Elliptic curve cryptography as a PKI technique replacing the RSA implementation. The use of ECC helps in achieving equivalent security level but with much smaller keys.
- Implement the GTK slicing technique (before encrypting it asymmetrically) we indicated in the computational cost section. Slicing of keys can achieve a lesser throughput as a result of reduced length of keys involved in the asymmetric encryption decryption process.

References

[1] Alliance, Wi-Fi, "Wi-Fi Protected Access: Strong, Standards-based, Interoperable Security for Today‟s Wi-Fi Networks," Wi-Fi Alliance, April 2003.

[2] Wagner, David, "Weak Keys in RC4," Posting to Scientific Crypt Usenet Group on Sep. 26, 1995.

[3] Fluhrer Scott, Itsik Mantin, and Adi Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4," In Selected Areas in Cryptography, pp. 1-24. Springer Berlin Heidelberg, 2001.

[4] www.aircrack.org, last accessed on Dec. 7, 2014.

[5] weplab.sourceforge.net, last accessed on Dec. 7, 2014.

[6] Robinson, F., “Examining 802.11i and WPA: The New Standards - Up Close,” Network Computing, April 2004.

[7] Mathews, Moffat and Ray Hunt, "Evolution of Wireless LAN Security Architecture to IEEE 802.11 i (WPA2)," In Proceedings of the fourth IASTED Asian Conference on Communication Systems and Networks, 2007.

[8] Tews, Erik, and Martin Beck, "Practical Attacks Against WEP and WPA," In Proceedings of the second ACM conference on Wireless network security, pp. 79-86. ACM, 2009.

[9] Martin Beck. “Enhanced TKIP Michael Attacks” 2010.

[10] Arana, Paul. "Benefits and Vulnerabilities of Wi-Fi Protected Access 2 (WPA2)." INFS 612 2006: pp. 1-6, 2006.

[11] Ahmad, Sohail, "WPA Too!," Airtight Networks, DEF Conference 18, 2010.

[12] www.airtightnetworks.com, last accessed on Dec. 15, 2014.

[13] Matthieu Caneill and Jean-Loup Gilis, “Attacks against the Wi-Fi Protocols WEP and WPA,” 2010.

[14] AirTight Networks, “WPA2 Hole 196 Vulnerability-FAQ,” AirTight Networks www.airtightnetworks.com, accessed date Dec 15, 2014.

[15] Baker, W., M. Goudie, A. Hutton, C. Hylender, J. Niemantsverdriet, C. Novak, D. Ostertag, "Verizon 2010 Data Breach Investigations Report," Verizon Business 2010.

[16] Baker, Wade, A. Hutton, C. David Hylender, J. Pamula, C. Porter, and M. Spitler, "2011 Data Breach Investigations Report," Verizon RISK Team, Available: www.verizonbusiness.com/resources/reports/rp_databreach-investigations-report 2011_en_xg.pdf, 2011: pp. 1-72. last accessed on Dec. 16, 2014.

[17] "2010 Cyber Security Watch Survey," CSO magazine, U.S. Secret Service. Software Engineering Institute, CERT Program at Carnegie Mellon University and Deloitte, 2010.

[18] 802.11i Authentication and Key Management (AKM) white paper, 2005.

[19] Kacic, Matej, "New Approach in Wireless Intrusion Detection System," 2014.

[20] Xing, Xinyu, Elhadi Shakshuki, Darcy Benoit and Tarek Sheltami, "Security Analysis and Authentication Improvement for IEEE 802.11i Specification," In Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE, pp. 1-5. IEEE, 2008.

[21] Nikita Borisov, Ian Goldberg and David Wagner, "Intercepting Mobile Communications: The Insecurity of IEEE802.11", 7th Annual International Conference on Mobile Computing and Networking, July 2001.

[22] Kacic, Matej, Petr Hanacek, Martin Henzl, and Peter Jurnecka, "Malware Injection in Wireless Networks," In Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2013 IEEE 7th International Conference, Vol. 1, pp. 483-487, IEEE, 2013.

[23] Rajotiya, Pridhi Arora, "Enhancing Security of Wi-Fi Network," International Journal of Computer Applications, Issue 2, Vol. 3, June 2012.

[24] Anthony Paladino, Kaystubh Phanse and Md.Sohail Ahmad, "Hole 196 Vulnerability in WPA2," Airtight Networks, 2010.

[25] Mirzoev and Stacey White, "The Role of Client Isolation in Protecting Wi-Fi Users from ARP Spoofing Attacks," 2014.

[26] IT Security Technical Publication 802.11 Wireless LAN Vulnerability Assessment ITSPSR-21A, May 2009.

[27] Benton, Kevin, "The Evolution of 802.11 Wireless Security," University of Nevada, Las Vegas, Informatics-Spring, 2010.

[28] Wong, Stanley, “The Evolution of Wireless Security in 802.11 Networks: WEP, WPA and 802.11 standards,” SANS, No. 7, 2003.

[29] Day, John D. and Hubert Zimmermann, "The OSI Reference Model," Proceedings of the IEEE 71, No. 12, 1983: pp. 1334-1340.

[30] IEEE LAN/MAN Standards Committee, "IEEE Std 802.1x-2010 (Revision of IEEE Std 802-1x-2004)," IEEE Standard for Local and Metropolitan Area Networks--Port-Based Network Access Control, Feb. 2010.

[31] Williams, Nicolas, "On the Use of Channel Bindings to Secure Channels," 2007.

[32] http://www.quora.com/The-OSI-Open-Systems-Interconnection-model-is-inefficient-each-layer-must-take-the-work-of-higher-layers-add-some-result-and-pass-the-work-to-lower-layers-Surely-this-wrapping-and-unwrapping-is-inefficient-What-is-the-security-advantage-of-the-layered-approach, accessed on 5-10-2015.

[33] https://technet.microsoft.com/en-us/library/cc757419(v=ws.10).aspx last accessed on May 15th, 2015.

[34] Aboba, Bernard, L. Blunk, J. Vollbrecht, James Carlson, and Henrik Levkowetz, "RFC 3748-Extensible Authentication Protocol (EAP)," Network Working Group 2004.

[35] Aboba, Bernard, and P. Calhoun, "RFC 3579-RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)," Internet Society, September, 2003.

[36] Stanley, Dorothey, Jesse Walker, and Bernard Aboba, "Extensible Authentication Protocol (EAP) Method Requirements for Wireless LANs," Request for Comments 4017, 2005.

[37] Baek, Kwang-Hyun, Sean W. Smith, and David Kotz, "A Survey of WPA and 802.11 i RSN Authentication Protocols," Dartmouth Computer Science Technical Report 2004, 2004.

[38] Stallings, William, "The RC4 Stream Encryption Algorithm," 2005.

[39] Kumkar, Vishal, Akhil Tiwari, Pawan Tiwari, Ashish Gupta, and Seema Shrawne, "Vulnerabilities of Wireless Security Protocols (WEP and WPA2)," International

Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 1, No. 2, 2012: pp. 34-38.

[40] Gutjahr, Alexander and A. Ludwigs, "Wired Equivalent Privacy (WEP) Functionality, Weak Points, Attacks," Freiburg University. Available From: http://www.data.ks.unifreiburg.de/download/praxisseminarWS9

[41] Lehembre, Guillaume, "Wi-Fi Security–WEP, WPA and WPA2," Hackin9, January 2005.

[42] Singh, Sukhchain, and Amit Grover, "Study and Analysis of Dictionary attack and Throughput in WEP for CRC-32 and SHA-1," International Journal of Computer Applications 96, No. 17, 2014: pp. 15-18.

[43] Bittau, Andrea, "The Fragmentation Attack in Practice," In IEEE Symposium on Security and Privacy, IEEE Computer Society, 2005.

[44] http://www.airtightnetworks.com/home/resources/knowledge-center/caffe-latte.html last accessed on Feb. 18, 2015.

[45] Scott R. Fluhrer, Itsik Mantin, and Adi Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4," In Selected Areas in Cryptography 2001, Vol. 2259 of Lecture Notes in Computer Science, pp. 1-24, Springer, 2001.

[46] Tews, Erik, Ralf-Philipp Weinmann, and Andrei Pyshkin, "Breaking 104 bit WEP in less than 60 seconds." International Workshop on Information Security Applications. Springer Berlin Heidelberg, 2007.

[47] Al Naamany, Ahmed M., Ali Al Shidhani, and Hadj Bourdoucen. "IEEE 802.11 Wireless LAN Security Overview," IJCSNS 6, no. 5B, 2006: pp. 138.

[48] Ozasa, Yuko, "A Study on the Tews-Weinmann-Pyshkin Attack Against WEP," IEICE Technical Report 2007, 2007: pp. 17-21.

[49] IEEE Computer Society LAN MAN Standards Committee, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Security Enhancements, ANSI," IEEE Std 802.11i, 2004.

[50] Nowicki, Gregory D., "Wireless Security: The Draft IEEE 802.11i Standard," 2004.

[51] Andras, Varge, "OMNet ++ User Manual Version 4.6", 2014.

Annexes

NB. *Code samples shown here are captured from simulation of the proposed solution.*

I. Code segment for sending received data frames module in AP

```
void Wifi_AP::handleMessage(cMessage *msg)
{
 Packet_file *ttmsg = check_and_cast<Packet_file *>(msg);
int gp_from_client1 = ttmsg->getGroup_packet();
   if(ttmsg->getSequence_no() >= 20000 and ttmsg->getSequence_no() %
25 == 0 )
      { morefrag = true;
         if(morefrag)
           {
             { ttmsg->setFragment_no_from_AP(ttmsg-
>getFragment_no_from_AP()+1);
             ttmsg->setFragment_no(ttmsg->getFragment_no()+1);}
               if(ttmsg->getFragment_no()==5)
                 {
                   ttmsg->setFragment_no_from_AP(0);
                   ttmsg->setFragment_no(0);
                   ttmsg->setSequence_no_from_AP(ttmsg-
>getSequence_no_from_AP()+1);
                  ttmsg->setSequence_no(ttmsg->getSequence_no()+1);
                  morefrag = false;
                 }
           }
      }

   else
   {   ttmsg->setSequence_no_from_AP(ttmsg-
>getSequence_no_from_AP()+1);
       ttmsg->setSequence_no(ttmsg->getSequence_no() + 1);}
if(ttmsg->getSequence_no()<=1000){

  gp_from_client1 = gp_from_client1 - PTK1_session1;

  gp_from_client1 = gp_from_client1 + GTK_session1;

 EV <<"Value of group addressed data frame at AP just after symmetric
encryption"<<gp_from_client1<<"\n";


int unencrypted_bundle = GTK_session1 + ttmsg->getSequence_no() +
ttmsg->getFragment_no();
if(unencrypted_bundle>=15){
```

```
    unsigned long bundle1 = 3;
    unsigned long bundle2 = 2;
    unsigned long encrypted_bundle1 =
abs((long)pow(unencrypted_bundle,bundle1)%modl);
    unsigned long encrypted_bundle2 =
abs((long)pow(unencrypted_bundle,bundle2)%modl);

    encrypted_bundle =
(encrypted_bundle1*encrypted_bundle2*encrypted_bundle2)%modl;
    }
else {
     encrypted_bundle=
abs((long)pow(unencrypted_bundle,privatekey)%modl);}
     ttmsg->setPki_encrypted_bundle(encrypted_bundle);
     ttmsg->setGroup_packet(gp_from_client1);

     EV<<"At AP_encrypted Group addressed
frame"<<gp_from_client1<<"\n";
    }

 else if(ttmsg->getSequence_no() > 1000){
      gp_from_client1 = gp_from_client1 - PTK1_session2;
      gp_from_client1 = gp_from_client1 + GTK_session2;
      EV <<"Value of group addressed data frame at AP just after
symmetric encryption"<<gp_from_client1<<"\n";

//RSA encryption on the GTK_session2, sequence number and fragment
number values
      int unencrypted_bundle = GTK_session2 + ttmsg->getSequence_no()
+ ttmsg->getFragment_no();
      ttmsg->setGroup_packet(gp_from_client1);
      encrypted_bundle = abs((long)pow(unencrypted_bundle,privatekey)
% modl);
      ttmsg->setPki_encrypted_bundle(encrypted_bundle);

      EV<<"Encrypted bundle"<<encrypted_bundle<<"\n";
      EV<<"At AP_encrypted Group addressed
frame"<<gp_from_client1<<"\n";
      }

   Packet_file *ttmsgcopy_C2 = ttmsg->dup();
   Packet_file *ttmsgcopy_C3 = ttmsg->dup();
   Packet_file *ttmsgcopy_C4 = ttmsg->dup();
   Packet_file *ttmsgcopy_C5 = ttmsg->dup();

    send(ttmsgcopy_C2, "out", 0);
    send(ttmsgcopy_C3, "out", 1);
    send(ttmsgcopy_C4, "out", 2);
    send(ttmsgcopy_C5, "out", 3);
```

II. Segment code of data frame sending module in client 1

```
void Wifi_C1::handleMessage(cMessage *msg)
{
    Packet_file *ttmsg = check_and_cast<Packet_file *>(msg);
    C1_Cache_Sequence_no = ttmsg->getSequence_no_from_AP()-1;
    C1_Cache_Fragment_no = ttmsg->getFragment_no_from_AP()-1;
    int received_seq = ttmsg->getSequence_no();
    int received_frag = ttmsg->getFragment_no();

 if(received_seq > 0)
   {
    if(received_frag == 0)
     {
      if(received_seq > C1_Cache_Sequence_no and received_frag >=
C1_Cache_Fragment_no)       {

EV << "sequence number at Client One = " <<C1_Cache_Sequence_no<<"
fragment number at one "<<C1_Cache_Fragment_no<<"\n";

   int gp = ttmsg->getGroup_packet();
   if (ttmsg->getSequence_no() <= 1000)
   {
      gp = gp + PTK1_session1;
      ttmsg->setGroup_packet(gp);
      EV << "Group addressed data frame at Client One = " << ttmsg-
>getGroup_packet()<<"\n";

               send(ttmsg, "out", 0);
         }
      else if (ttmsg->getSequence_no() > 1000 and ttmsg-
>getSequence_no() < 20000)
      {
            ttmsg->setName("session2");
              gp = gp + PTK1_session2;
              ttmsg->setGroup_packet(gp);
              EV << "Group addressed data frame at Client One = " <<
ttmsg->getGroup_packet()<<"\n";
                int xx = ttmsg->getGroup_packet();
                ttmsg->setGroup_packet(xx);
                send(ttmsg, "out", 0);
      }
     }
    }

   else if(received_frag > 0)
   {
       if(received_seq == C1_Cache_Sequence_no and received_frag >
C1_Cache_Fragment_no )
              {
```

```
            C1_Cache_Sequence_no = ttmsg->getSequence_no_from_AP();;
            C1_Cache_Fragment_no = ttmsg->getFragment_no_from_AP();;
 EV << "sequence number at Client One = " <<C1_Cache_Sequence_no<<"
fragment number at one "<<C1_Cache_Fragment_no<<"\n";
           int gp = ttmsg->getGroup_packet();
           if (ttmsg->getSequence_no() <= 1000)
           {
              gp = gp + PTK1_session1;
              ttmsg->setGroup_packet(gp);
              EV << "Group addressed data frame at Client One = " <<
ttmsg->getGroup_packet()<<"\n";
                     send(ttmsg, "out", 0);
           }
              else if (ttmsg->getSequence_no() > 1000 and ttmsg-
>getSequence_no() < 20000)
              {
                    ttmsg->setName("session2");
                      gp = gp + PTK1_session2;
                      ttmsg->setGroup_packet(gp);
EV << "Group addressed data frame at Client One = " << ttmsg-
>getGroup_packet()<<"\n";
                        int xx = ttmsg->getGroup_packet();
                        ttmsg->setGroup_packet(xx);
                        send(ttmsg, "out", 0);
              }
           }
     }
   }
   else if (ttmsg->getSequence_no() == 0 and ttmsg->getFragment_no()
== 0 and firsttime==true)
   {
       firsttime=false;
       int gp = ttmsg->getGroup_packet();
       gp = gp + PTK1_session1;
       ttmsg->setGroup_packet(gp);
       send(ttmsg, "out", 0);
   }
   else
       EV << "Group addressed data frame is dropped. It is detected as
a reply attack!";
}
```

III. Segment code of received data frame handler module in client 5

```
void Wifi_C5::handleMessage(cMessage *msg)
{
Packet_file *ttmsg = check_and_cast<Packet_file *>(msg);

C5_Cache_Sequence_no = ttmsg->getSequence_no_from_AP()-1;
C5_Cache_Fragment_no = ttmsg->getFragment_no_from_AP()-1;
int decrypted_tt = ttmsg->getGroup_packet();
int received_seq = ttmsg->getSequence_no();
int received_frag = ttmsg->getFragment_no();

if(received_seq > C5_Cache_Sequence_no and received_frag >=
C5_Cache_Fragment_no)
{
    C5_Cache_Sequence_no = ttmsg->getSequence_no_from_AP();
    C5_Cache_Fragment_no = ttmsg->getFragment_no_from_AP();

    if (ttmsg->getSequence_no() <= 1000)
    {
        decrypted_tt = decrypted_tt - GTK_session1;
        EV << "Group addressed data frame decrypted with GTK_session2
at Client Five = " << decrypted_tt <<"\n";
    }
     else if (ttmsg->getSequence_no() > 1000){
        decrypted_tt = decrypted_tt - GTK_session2;
        EV << "Group addressed data frame decrypted with GTK_session2
at Client Five = " << decrypted_tt <<"\n";
     }
}
else
{
    EV << " Valid Group addressed data frame from AP is dropped. It is
detected as a reply attack!";
    const char *reply = "reply attack";
    bubble(reply);}

}
```

DECLARATION

I, the undersigned, declare that this research is my original work and has not been presented for degree in any other university, and that all sources of materials used for the research have been acknowledged.

Declared by:

Name: **Meareg Abreha Hailemariam**

Signature: _____

Date: _____

Confirmed by advisor:

Name: **Dejene Ejigu (PhD)**

Signature: _____

Date: _____

Place and date of submission: Addis Ababa University, November 2016.