SECURITY OF 802.11i BASED WIRELESS LOCAL AREA NETWORKS

By

Mansoor Ahmed Khan

Submitted to the Faculty of Information Security Department
National University of Sciences and Technology, Rawalpindi in partial fulfillment
for the requirements of an M.S Degree in Information Security
JUNE 2008

ABSTRACT

Even with ratification of 802.11i, WLANs remain vulnerable to Denial of Service (DoS) attacks due to unprotected and unauthenticated Management and Control Frames. These include Deauthentication, Disassociation, Request To Send (RTS), Clear To Send (CTS), Acknowledgement (ACK) and Power Saving Poll (PS-Poll) message attacks. Different defense techniques and protocols have been proposed to counter these threats. These either possess certain deficiencies or have implementation complexities and no solution encompassing all such attacks has yet been proposed. Moreover, a vulnerability related to Advance Encryption Standard (AES) Counter with Cipher Block Chaining Message Authentication Code Protocol (CCMP), used for Confidentiality and Integrity assurance in 802.11i, has also been recently identified. It exploits weak nonce construction mechanism of AES CCMP to calculate initial counter value, lowering effective key length from 128 bits to 85 bits. Hence, Time Memory Trade-OFF (TMTO) attack becomes a possibility. No solution has yet been proposed for AES CCMP vulnerability.

The purpose of this thesis is to devise effective practical countermeasures against DoS attacks based on Management and Control Frames of 802.11 and AES CCMP related vulnerability of 802.11i. The defense mechanism designed for DoS attacks is based on authenticating said messages with a Pseudo Random Number, calculated using Pairwise Transient Key (PTK) that is inaccessable to adversaries. The countermeasure proposed for AES CCMP vulnerability involves strengthening of the nonce construction mechanism of AES CCMP by randomization. Both defense mechanisms have been implemented and tested on actual hardware using a test network. The proposed techniques successfully counter the threats, are simple to implement by a software upgrades and do not require hardware upgradation.

DEDICATION

In the name of Allah, the most beneficent, the most merciful.
Dedicated to my parents, wife and children, their sincere prayers and ever-present support have been the driving forces for my success.

ACKNOWLEDGMENTS

I duly acknowledge all the help and support extended by my committee members, namely Lt Col Attiq Ahmed, Assistant Professor Imtiaz Ahmad Khokhar and Lecturer Ahmad Raza Cheema. Without their guidance, completion of this thesis work would have become a much more uphill task. I would also like to mention the consummate supervision of my advisor Sqn Ldr Aamir Hasan, which made it possible for me to complete my research work.

In the end, I would also exalt the assistance offered by my class mates Mr Sheraz Naseer and Mr Muhammad Zeshan, who were always there to abet me throughout my thesis phase.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

KEY TO SYMBOLS OR ABBREVIATIONS

AA	Authenticator Address
AAA	Authentication, Authorization, and Accounting
AAD	Additional Authentication Data
AID	Association IDentifier
AKM	Authentication and Key Management
AKMP	Authentication and Key Management Protocol
ANonce	Authenticator Nonce
AP	Access Point
ARP	Address Resolution Protocol
AS	Authentication Server
BSA	Basic Service Area
BSS	Basic Service Set
BSSID	Basic Service Set Identification
CBC	Cipher-Block Chaining
CBC-MAC	Cipher-Block Chaining with Message Authentication Code
CCA	Clear Channel Assessment
CCM	CTR with CBC-MAC
CCMP	CTR with CBC-MAC Protocol
CID	Connection Identifier
CRC	Cyclic Redundancy Code
CS	Carrier Sense
CTR	Counter Mode
CTS	Clear To Send
DA	Destination Address
DCE	Data Communication Equipment

DCF	Distributed Coordination Function
DIFS	Distributed (Coordination Function) InterFrame Space
DS	Distribution System
DSAP	Destination Service Access Point
DSM	Distribution System Medium
DSS	Distribution System Service
DSSS	Direct Sequence Spread Spectrum
EAP	Extensible Authentication Protocol (IETF RFC 3748)
EAPOL	Extensible Authentication Protocol over LANs (IEEE P802.1x-REV)
EIFS	Extended Inter Frame Space
ESA	Extended Service Area
ESS	Extended Service Set
FC	Frame Control
FCS	Frame Check Sequence
FHSS	Frequency-Hopping Spread Spectrum
GMK	Group Master Key
GNonce	Group Nonce
GTK	Group Temporal Key
GTKSA	Group Temporal Key Security Association
IBSS	Independent Basic Service Set
ICV	Integrity Check Value
IV	Initialization Vector
KCK	EAPOL-Key Confirmation Key
KDE	Key Data Encapsulation
KEK	EAPOL-Key Encryption Key
LAN	Local Area Network

LLC	Logical Link Control
MAC	Medium Access Control
MIC	Message Integrity Code
MPDU	MAC Protocol Data Unit
MMPDU	MAC Management Protocol Data Unit
MSDU	MAC Service Data Unit
NAV	Network Allocation Vector
PDU	Protocol Data Unit
PHY	Physical (Layer)
PHY-SAP	Physical Layer Service Access Point
PMK	Pairwise Master Key
PMKID	Pairwise Master Key Identifier
PMKSA	Pairwise Master Key Security Association
PN	Packet Number
PS	Power Save (mode)
PRF	Pseudo-Random Function
PRNG	Pseudo-Random Number Generator
PSK	Pre-Shared Key
PTK	Pairwise Transient Key
PTKSA	Pairwise Transient Key Security Association
RA	Receiver Address
RADIUS	Remote Authentication Dial-in User Service (IETF RFC 2865 [B14])
RSN	Robust Security Network
RSNA	Robust Security Network Association
RTS	Request To Send
RX	Receive or Receiver

SA	Source Address
SAP	Service Access Point
SNonce	Supplicant Nonce
SPA	Supplicant Address
SS	Station Service
SSAP	Source Service Access Point
SSID	Service Set Identifier
STA	Station
TA	Transmitter Address
TIM	Traffic Indication Map
TKIP	Temporal Key Integrity Protocol
TMTO	Time Memory Trade Off
TSN	Transition Security Network
TX	Transmit or Transmitter
TXE	Transmit Enable
WEP	Wired Equivalent Privacy
WM	Wireless Medium

Chapter 1

Introduction

Wireless networks are becoming an increasingly popular choice amongst the prevalent network access technologies. Expedient deployment, flexibility and low cost are the prime contributing factors to this widespread popularity. However, the security aspect in Wireless Local Area Networks (WLANs) is an active research area. The inherent security weaknesses of the wireless medium pose a much more stern threat as compared to the wired networks. The most dominant shortcomings are the lack of control on the operation area and the ability of an attacker to passively eavesdrop all network traffic [1-6]. These vulnerabilities may be exploited by adversaries to gain unauthorized access to information over the network and cause disruption or denial of service to legitimate users.

Even with the ratification of 802.11i [7], WLANs based on 802.11 [8] Standard remain vulnerable to Denial of Service (DoS) attacks due to unprotected and unauthenticated Management and Control Frames. The attacks include Deauthentication, Disassociation, Request To Send (RTS), Clear To Send (CTS), Acknowledgement (ACK), and Power Saving Poll (PS-Poll) message based attacks [9], [10]. Different types of defense techniques and protocols have been proposed to counter these threats[2-6], [9-11]. These either possess certain deficiencies or have implementation complexities. Moreover, no solution encompassing all DoS attacks based on Management and Control Frames has yet been proposed.

The IEEE 802.11i Standard offers arguably uncompromised Confidentiality and Integrity Services by utilizing Advance Encryption Standard in Counter with Cipher Block Chaining Message Authentication Code Protocol (AES CCMP).

However the Nonce construction mechanism employed in the standard is weak, leading to Initial Counter prediction. Resultantly, the effective Key Length used for encryption is reduced from 128 to 85 bits and Time Memory Trade Off (TMTO) attack becomes a possibility [12-13].

1.1 Wireless Local Area Networks

Wireless Local Area Networks can be deployed in two configurations Infrastructure mode or Ad Hoc mode. The Infrastructure mode, also known as Basic Service set involves service provisioning to wireless nodes or supplicants through an Access Point, serving as the Authenticator. Use of more than one access points forms what is called an Extended Service Set (ESS). The Ad Hoc mode is when the individual wireless clients are connected to one another directly, without any Access Point and each acts as Authenticator and Supplicant. In WLAN standards, IEEE 802.11 is the De Facto standard for Wireless LANs, 802.11a, b and g are extensions of this standard which differ in operating frequency and bandwidth.

1.1.1 Wired vs. Wireless Network Access

Security of wired networks has been an active research area since their emergence. It has taken decades to finally develop and implement an acceptable level of security mechanisms for the wired network access technologies. Comparatively, it is important to highlight here that wireless networks are comparatively quite new and face more security challenges than wired networks. First, in wired networks, it is possible to secure the connection by concealing the wires inside walls or conduits to restrict access. Conflictingly, wireless medium is accessible to everyone within the coverage area and it is extremely difficult in WLANs to profile and bound the

transmission. As a result, an adversary can access traffic without any trace. For example, an attacker may interfere with the wireless network using high gain antennas, sitting miles away. Secondly, in wired networks, end users have assurance on the authenticity of the networks they connect to. For example, when a user plugs his device into a jack on the wall at his workplace, he knows that network access is provided by his company. In wireless networks, a user is completely sightless to the connected network, as he can only view the network from the associated AP, which might be malicious. Thirdly, the connections between wireless devices are adaptive and flexible according to the user mobility and link quality, which is a desired advantage over the wired networks, but causes a more complicated trust relationship. Finally, the cryptographic operations should adapt to the computation and power restraints of wireless devices. The authentication and key management protocols should be scalable and ubiquitous to support user mobility. Furthermore, due to the inherent vulnerabilities of wireless channels, it is much more difficult to defend against DoS attacks in a wireless environment.

1.1.2 Link Layer and MAC Layer Authentication

The 802.11 Standards deal with the Physical Layer (PHY) and Media Access Control (MAC) that is a sub-layer of the Link Layer. The PHY specifications, including 802.11 FHSS (Frequency Hopping Spread Spectrum), DSSS (Direct Sequence Spread Spectrum), and OFDM (Orthogonal Frequency Division Multiplexing), define the physical signal transmit, receive, and the CCA (Clear Channel Assessment). The MAC specifications define the protocol to access the shared media. These include Frame scheduling, acknowledgement, retransmission,

collision avoidance, etc. The Upper Layers in wireless networks are the same as in common wired networks. In order to satisfy security requirements, authentication mechanism can be implemented in either Link Layer or Upper Layers.

In Wireless Local Area Networks, authentication is implemented at Link Layer due to various reasons. First, the original mechanism in 802.11 Standard aimed to provide Wired Equivalent Privacy (WEP) to protect the wireless link. Thus, other existing mechanisms in wired networks could still work as usual by treating wireless link as a wired one. The more advanced mechanisms developed later also inherited Link Layer authentication for reusing legacy devices. Secondly, it is fast, simple and inexpensive to perform authentication at the Link Layer. Compared with Upper Layer authentication, devices capable of processing Link Layer Frames are fairly cheap due to a much smaller delay in Link Layer Frames exchange. This provides a transparent and independent implementation of protocols running at Upper Layers. Furthermore, since authentication is done at the edge of the system and at the beginning of the session, clients do not need any network access prior to authentication. Allowing network access prior to authentication may introduce vulnerabilities into the system. This is successfully avoided by using edge authentication without prior network access.

1.2 Evolution of Wireless Security

The inherent security weaknesses of wireless media pose a more severe threat as compared to other network access technologies. This was duly recognized by the developers and continuous efforts are in hand to rectify these vulnerabilities. The development of 802.11 series of protocols including WEP, WiFi Protected Access (WPA) and WPA2 form a part of the attempts to address the identified vulnerabilities

of the WLANs. The 802.11i Standard is the latest of the series of protocols ratified for WLAN security. Although the protocol addresses most of the security concerns related to Authentication, Confidentiality and Integrity services, it does not address the Availability of wireless access networks [7].

1.2.1 The CIA Triad

In any network access technology, security has different perspectives depending on differing applications. However, Confidentiality, Integrity and Authentication, also called the CIA triad, are the rudiments for any network. The network must provide strong data Confidentiality and Integrity for all transmitted messages. Data Confidentiality and Integrity aid to develop a secure channel for users to communicate in an insecure wireless environment. In this environment, only the communicating users must able to understand received messages, generate, modify or reply with valid messages. These requirements could be met by implementing well-designed cryptographic functions. The network must also provide mutual authentication in which all the communicating nodes authenticate each other's identity. If required, authentication process should also combine key generation, distribution and management to provide secret keys for the cryptographic functions. Based on the authentication results, flexible authorization and access control policies can be deployed to restrict user privileges.

1.2.2 Wired Equivalent Privacy (WEP)

In order to provide data Confidentiality equivalent to a wired network, IEEE 802.11 Standard originally defines Wired Equivalent Privacy (WEP). This mechanism adopts a stream cipher known as RC4 (Rivest Cipher 4) to encrypt

messages with a shared key. This key is concatenated with a 24-bit Initialization Vector (IV) to construct a per-packet RC4 key. In order to provide data Integrity, WEP calculates an Integrity Check Value (ICV) over the MSDU (MAC Service Data Unit), which is a common Cyclic Redundancy Checksum (CRC). The Frame body and the corresponding ICV are encrypted using the per-packet key. In addition, two authentication mechanisms are defined, Open System Authentication, which is actually a null authentication, and Shared Key Authentication, which is a Challenge-Response handshake based on the shared key.

Numerous studies have shown that data Confidentiality, Integrity, and Authentication are not achieved through WEP defined security mechanisms [1], [14-20]. First, the 40-bit shared key is too short against brute-force attacks. Though some vendors might support a longer key (128 bits, containing a 104-bit key and a 24-bit IV), it is still effortless for an adversary to recover the plaintext as the small IV size and static shared key result in a high possibility of key stream reuse, which defeats any stream cipher. Furthermore, concatenation of the IV and the shared key has inherent weakness for generating the per-packet RC4 key as an adversary can discover this key by eavesdropping several million packets. Moreover, since ICV is a linear and un-keyed function of the message, data Integrity cannot be guaranteed. Even without any knowledge of the key stream, an adversary is able to arbitrarily modify a packet without detection, or forge a packet with a valid ICV. This weak Integrity also enables much easier plaintext recovery. Finally, an adversary can spoof the Shared Key Authentication by observing an authentication process of a legitimate station. In the end, WEP does not implement any mechanism to prevent replay attacks.

1.2.3 WiFi Protected Access

Although WEP fails to satisfy any security requirements, it was not considered practical to expect users to discard their WEP compatible devices completely. Hence, WiFi Alliance proposed an interim solution, called WiFi Protected Access (WPA), to eliminate the vulnerabilities while reusing legacy WEP compliant hardware. WPA adopted a Temporal Key Integrity Protocol (TKIP) for data Confidentiality, still used RC4 for data encryption, but included a key mixing function and an extended IV space to construct disparate and fresh per-packet keys. WPA also introduced Michael algorithm, a weak keyed Message Integrity Code (MIC), for improved data Integrity under the limitation of computation power available in the devices. Furthermore, in order to detect replayed packets, WPA implemented a packet sequencing mechanism by binding a serially increasing Sequence Number to each packet. WPA also provided two improved authentication mechanisms. In one mechanism, possession of a Pre-Shared Key (PSK) authenticates the peers. Furthermore, a 128-bit encryption key and another distinct 64-bit MIC key can be derived from the PSK. Alternatively, IEEE 802.1x and the Extensible Authentication Protocol (EAP) can be adopted to provide a stronger authentication for each association, and generate a fresh common secret key as part of the authentication process.

TKIP was proposed to address all known vulnerabilities in WEP and enhance the security in all aspects. However, there are weaknesses in WPA due to the limitation of reusing legacy hardware. Although TKIP key mixing function has stronger security than the WEP key scheduling algorithm, it is not as strong as expected. It is possible to find the MIC key given one per-packet key. Furthermore, the whole security is broken for the duration of a Temporal Key (TK) given two per-

packet keys with the same IV. Furthermore, Michael algorithm is designed to provide only 20 bits of security in order to minimize impact on the performance, which means an adversary can construct one successful forgery every 219 packets. Thus, countermeasures are necessary to limit the rate of forgery attempts. However, this countermeasure may allow DoS attacks. In addition, 802.1x authentication may be vulnerable to Session Hijacking and Man-in-the-Middle attacks. Though these attacks can be prevented by using mutual authentication and strong encryption, using 802.1x in a shared media WLAN is problematic as it was originally designed for a switched LAN [2].

1.2.4 IEEE 802.11i (WPA2)

As a more robust solution, IEEE 802.11i has been ratified to provide enhanced security at MAC layer. 802.11i provides authentication protocols, key management protocols, and data Confidentiality protocols that may operate concurrently over a network using other protocols as well. The specification defines two types of networks, namely Robust Security Network Association (RSNA) and Pre-RSNA [7]. The Pre-SNA encompasses WEP and 802.11 user authentication. Pre-RSNA has just been included for backward compatibility and is strongly not recommended as it does not provide adequate security for Wireless Access Networks. RSNA includes two data Confidentiality protocols including TKIP and CCMP. TKIP is considered as just a wrapper around WEP, while CCMP is a totally new design with arguably uncompromised security architecture. RSNA provides strong mutual authentication using several components, including an 802.1x authentication phase using TLS over EAP, a 4-Way Handshake to establish a fresh session key, and an optional Group Key Handshake for group communications. CCMP utilizes AES [21] encryption algorithm

in Counter Mode to provide Confidentiality and CBC-MAC for message authentication. Therefore, the default and recommended protocol for reliable security in RSNA is CCMP [22, 23].

Several attacks on Availability of 802.11 based wireless networks have already been identified and demonstrated [1-6]. Since the Management and Control Frames in 802.11 are unprotected, even with the introduction of 802.11i, these can be easily forged to launch DoS attacks. Many solutions have been proposed as countermeasures against these types of attacks [2-6], [10-11]. The IEEE 802.11 Work Group is also presently working on 802.11w protocol, which is intended to address the Availability service of wireless networks and effectively defend against these weaknesses. In addition to the DoS attacks on 802.11 based wireless networks, a vulnerability related to CCMP has also been recently identified , which could be subsequently used to launch TMTO attack against the protocol, thus compromising the complete security of the architecture [12-13].

1.2.5 Problem Statement

The past studies have extensively focused on the data Confidentiality, Integrity and mutual authentication for wireless security. However, Availability has not been considered sufficiently. Many Denial of Service (DoS) attacks have been disclosed on the WLAN systems from the Physical Layer to the Application Layer. Some might think that DoS attacks are inevitable due to the physical characteristics of wireless links. However, since many DoS attacks can be mounted by an adversary with moderate equipment they should be considered to be real threats to a WLAN implementation.

At Physical Layer, a straightforward DoS attack is frequency jamming. An adversary can interfere with the whole frequency band with a strong noise signal, blocking legitimate data transmissions. Fortunately, it is relatively expensive because the adversary needs special equipments and huge power consumption to jam the whole spectrum. Also spread spectrum technology can be adopted in wireless networks to make the frequency jamming more difficult. Additionally, an adversary performing this attack can be easily detected and located by a network administrator. Therefore, it is reasonable to assume that an adversary will not try to launch this attack for common purposes. There exists another easier approach to mount a frequency jamming in a WLAN implementing Direct Sequence Spread Spectrum (DSSS). By exploiting the Clear Channel Assessment (CCA) procedure, an adversary can cause all WLAN nodes within range to consider the channel busy and defer transmissions of any data. Particularly, most vendors do not remove the engineering function PLME-DSSSTESTMODE from their released products, which makes the attack more convenient through the off-the-shelf usage of a common wireless Network Interface Card (NIC). There are no complete solutions for this DoS attack yet. Fortunately, the attack only affects a WLAN system implementing CCA, which is in DSSS and not in OFDM [7].

1.2.5.1 Network Availability and Denial of Service (DoS) Attacks

At MAC layer, an adversary can scramble the channel by MAC preemptive jamming because WLAN is designed to be cooperative. For example, the adversary can send out a short jamming noise in every time interval of SIFS (Short Inter-Frame Space, 10μs in 802.11b networks), which will surely collide with all the legitimate traffic or cause the legitimate traffic to be deferred infinitely. However, this attack is

not considered to be a real threat because the adversary needs to send out about 50,000 packets per second in an 11 Mbps 802.11b network. As another possible attack, an adversary is able to transmit legitimate messages, without obeying the standard. Specifically, the adversary could use a smaller "backoff" time, in order to obtain an unfair allocation of the channel bandwidth. If the adversary adopts no "backoff", he may ultimately cause a DoS attack for legitimate users [7].

More DoS vulnerabilities arise from the unprotected Management and Control Frames. An adversary is able to easily launch a DoS attack on a specific station or the entire Basic Service Set (BSS) by forging the Deauthentication, Disassociation, or Power Saving-Poll (PS-Poll) messages. Furthermore, DoS attacks could be mounted by exploiting the virtual carrier-sense scheme through forging any Frame including Request To Send (RTS), Clear To Send (CTS) and Acknowledgment (ACK) with an extremely large value of NAV (Network Allocation Vector). This can fool devices to consider the channel busy preventing devices from transmitting messages [10].

Additionally, an adversary can perform an ARP (Address Resolution Protocol) cache poisoning to mount a DoS attack. Furthermore, if an IEEE 802.1x Authentication is implemented for stronger authentication, the adversary has more choices to mount a DoS attack through forging EAP-Start, EAP-Logoff, and EAP-Failure messages. The adversary can also exhaust the space of the EAP packet identifier, which is only 8 bits long, by sending more than 255 authentication requests simultaneously [2].

Various defense techniques have been proposed to counter the discussed DoS attacks[2-6], [10-11]. While most types of attacks have been successfully mitigated using appropriate defense mechanisms, attacks based on unprotected Management and Control Frames remain a persistent threat. The countermeasures proposed either

possess certain deficiencies or have implementation complexities. Moreover, no solution encompassing all DoS attacks based on Management and Control Frames has yet been devised.

1.2.5.2 Attack on Confidentiality

RSNA architecture utilizes Advance Encryption Standard (AES) [21] in Counter with Cipher Block Chaining Message Authentication Code Protocol (CCMP) [22, 23]. The Counter (CTR) Mode of AES is used for encrypting data to ensure Confidentiality, while Cipher Block Chaining (CBC) Mode is used to generate the Message Authentication Code, namely Message Integrity Code (MIC) [7]. MIC is utilized to protect Integrity of data and authentication of the sender. The key length used in both modes of AES is 128 bits. Security offered by AES CCMP in Robust Security Networks (RSNs) has been analyzed comprehensively, and it is believed that the protocol offers sufficient level of Confidentiality and Integrity of data[2-6].

As with any modern cipher, security of AES is also dependant on the key used [21]. AES counter mode uses this key (128 bits in length), to encrypt Initial Counter value. The result is exclusive ORed (XORed) with plain text to produce the first cipher text block. Counter is then incremented and operation is repeated to produce the next cipher text block. The procedure runs iteratively until complete plain text is encrypted. Hence, in case of counter mode, security of the architecture is reliant upon key used and the Initial Counter value. Initial Counter is constructed by concatenating the Flags, Nonce and Length of Payload Fields, while the Nonce is obtained by concatenation of Packet Number (PN), Medium Access Control (MAC) Address A2 and Priority Fields.

Prediction of Initial Counter value results into lowering of effective key length for AES from 128 bits to 85 bits, less than the recommended effective key length of 97 bits for block ciphers [13]. Consequently, AES counter mode becomes vulnerable to a Time Memory Trade Off (TMTO) attack, therefore improvement of Initial Counter construction mechanism in AES CCMP is considered imperative [12].

1.3 Objectives

The objective of this thesis is to study the existing DoS attacks, based on Management and Control Frames of 802.11 specification and recommend effective practical countermeasures to defend against this category of attacks. Moreover, the AES CCMP related vulnerability of 802.11i would also be studied and a practical solution would be proposed to counter the threat posed.

1.4 Thesis Organization

Chapter 2 describes the security architecture of IEEE 802.11i. It includes an overview of Authentication, Key Management, Confidentiality and Integrity mechanisms utilized by the standard and a review of attacks on the architecture with already proposed defense mechanisms. Chapter 3 specifically discusses Management and Control Frame based DoS attacks, including Deauthentication, Disassociation, RTS/ CTS/ ACK and PS-Poll message based attacks, and the attack on Confidentiality based on weak Nonce construction mechanism of AES CCMP and discusses the possibility of the TMTO attack on the AES CCM Protocol. Chapter 4 presents the devised defense mechanisms for Management and Control Frame based DoS attacks and the AES CCMP vulnerability. Chapter 5 includes the implementation

and results of the proposed countermeasures, and the thesis is concluded in Chapter 6 with an analysis of the proposed defense methodologies.

Chapter 2

IEEE 802.11i Security Architecture

2.1 Introduction

Ever since flaws were identified in the 802.11 Standard, 802.11i Task Group started designing a new security Framework for eradicating WEP deficiencies. The developed Framework in shape of IEEE 802.11i Standard defines a new Robust Security Network (RSN). It delineates enhanced methods for Authentication, Key Management and data Confidentiality. The 802.11i Standard offers a choice to use either 802.1x or Pre Shared Key for Authentication and Key Management (AKM). It employs AES as the cipher in a newly designed protocol, namely CCMP, as the default protocol for Confidentiality and Integrity. The use of 802.1x / Pre Shared Key (PSK) authentication, together with AES CCMP, forms a Robust Security Network Association (RSNA). For backward compatibility RSN-capable devices support WEP, WPA and the previous 802.11 authentication methods as Pre-RSNA based security architecture. However, their usage is not recommended and operating in the RSN mode prohibits use of these denigrated techniques. CCMP is the default and recommended protocol in 802.11i due to its arguably uncompromised Confidentiality and Integrity services.

2.2 Authentication and Key Management

802.11i Standard includes a Transitional Security Network (TSN) that allows Pre-RSNA based security architecture; however, the recommended Framework to be adopted is RSNA. RSNA utilizes 802.1x port-based authentication that uses

Extensible Authentication Protocol Over LANs (EAPOL). 802.1x does not mandate a specific authentication method, rather defines an architecture and message format to be followed. It is discretionary to the use of certificates, smart cards, passwords or secret keys as optional methods. The fundamental requirement for RSNA is that authentication is mutual, it does not disclose required information to allow impersonation attacks, and implements fresh session keys for each communication session.

The 802.1x authentication is performed between a Supplicant and an Authentication Server (AS) to establish a secure channel between the Supplicant and an Authenticator (Access Point (AP) for WLANs). RSNA structure assumes the channel between Authenticator or AP and AS is secure, and AS is a trusted entity. The AS sends the calculated session key to the authenticator after successful completion of 802.1x authentication. The authenticator and supplicant then perform a 4-way handshake to validate that both share the same security parameters and session keys. In Infrastructure mode, Supplicant may be a laptop, Authenticator an AP, and AS a Remote Authentication Dial-In User Service (RADIUS) server. In ad-hoc mode, each device has to perform the role of both the Supplicant and Authenticator. Additionally, they should have a dedicated AS.

RSNA also allows use of a Pre Shared Key (PSK) installed in the devices. In this scenario, 802.1x authentication is not required and only 4-way handshake is performed. This Framework is intended to simplify the management of home and Ad-hoc WLANs. During the RSN Association establishment, Supplicant and AP transit between several security states, which are Unauthenticated and Unassociated, Open System Authenticated and Unassociated, Open System Authenticated and 802.11

Associated, 802.1x Authenticated and 802.11 Associated, and 802.1x Authenticated and 802.11i Associated.

Step by step RSNA establishment procedure and the above mentioned transitional security states are depicted in Figure 2.1.

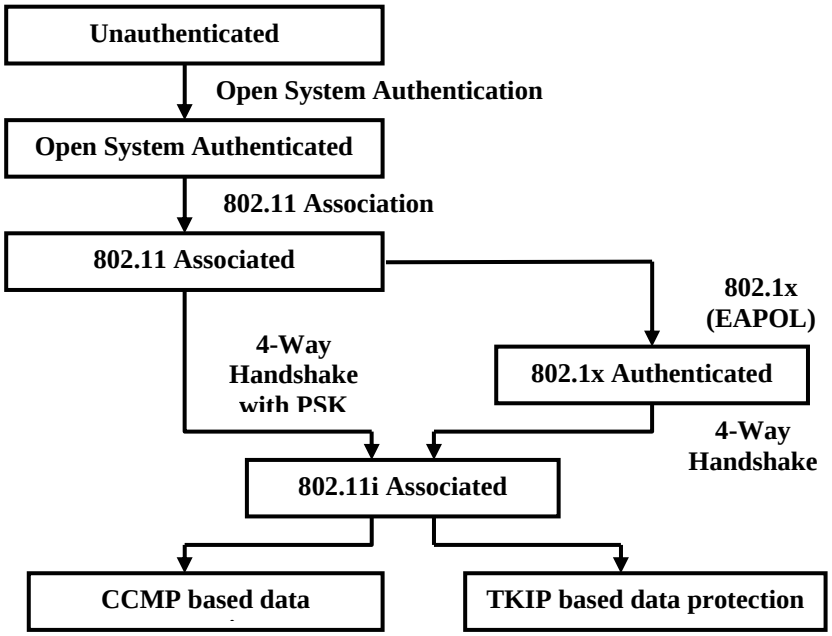

Figure 2.1. Step by step RSNA establishment procedure and transitional security states

As discussed, RSNA relies upon 802.1x and 4 way handshake for Authentication and Key Management (AKM). In case PSK is not used, the AKM proceeds as follows [7]. Before using IEEE 802.1x, IEEE 802.11 presumes that a secure channel has been established between Authenticator and AS, security which is outside the scope of this amendment. The Station determines AP's security policy through passive monitoring of Beacon Frames or through a probe request, following which 802.11 Association is performed (Figure 2.2 [7]). 802.1x authentication begins with EAP authentication process when the AP's Authenticator sends the EAP-Request or the Station's (STA's) Supplicant sends the EAPOL-Start message. EAP

authentication Frames pass between the Supplicant and AS via the Authenticator and Supplicant's Uncontrolled Ports. The Supplicant and AS authenticat each other and generate a Pairwise Master Key (PMK). The PMK is sent from the AS to the Authenticator over the secured channel. The stated procedure is shown in Figure 2.3 [7]. To complete the AKM process, a 4-Way Handshake (depicted in Figure 2.4 [7]) is initiated by the Authenticator using EAPOL-Key Frames to validate that both entities hold the same PMK, ensure that PMK is current, calculate a new Pairwise Transient Key (PTK) or Group Transient Key (GTK) in case of Multicasting.

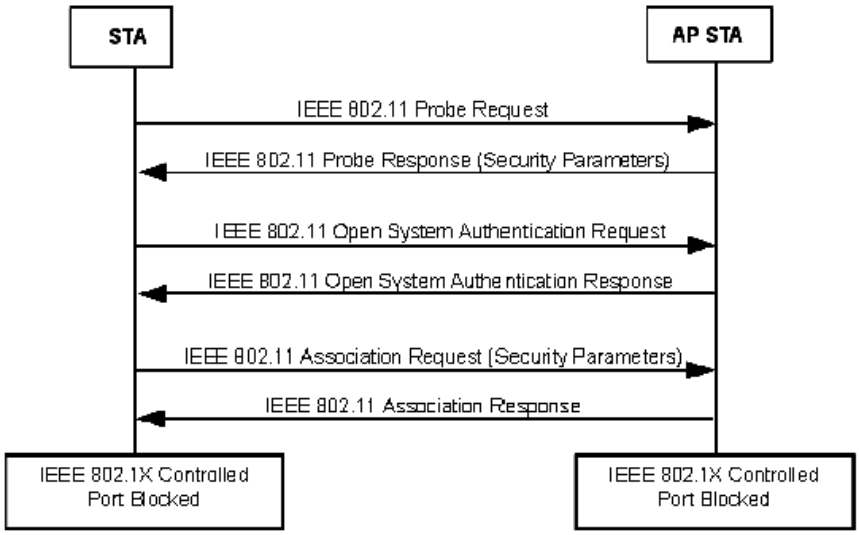

Figure 2.2. 802.1x Association

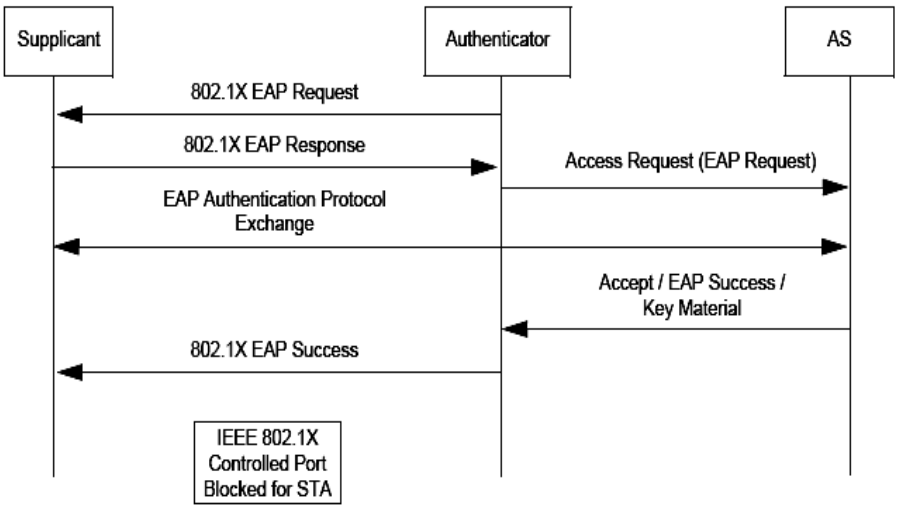

Figure 2.3. 802.1x Authentication using EAPOL.

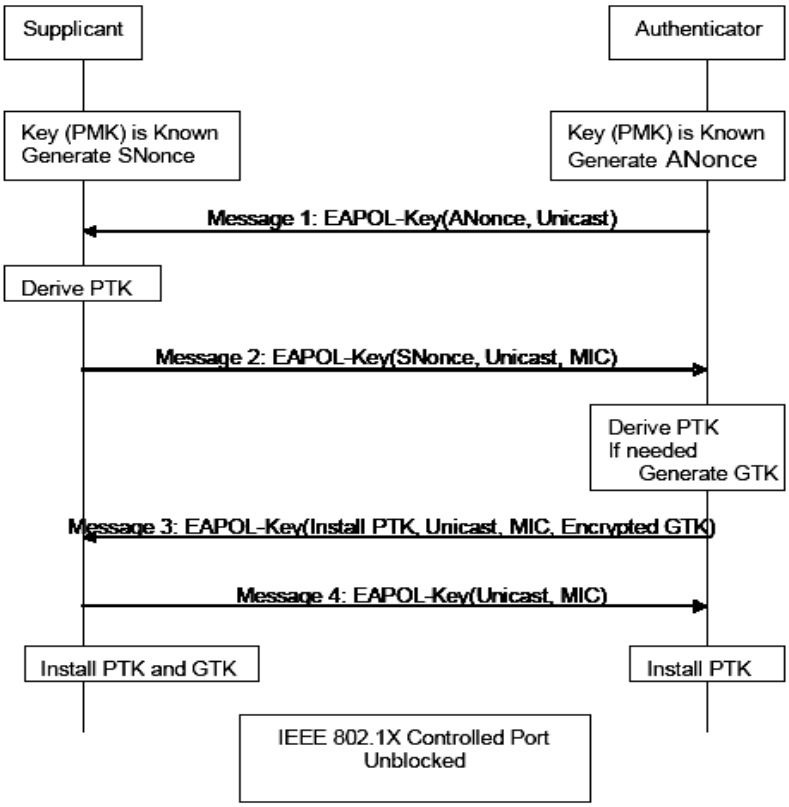

Figure 2.4. Key generation using 4 way handshake.

In case PSK is used, the complete procedure is not required as PSK is utilized as the PMK. First, the Station discovers AP's security policy through passively monitoring Beacon Frames or via a probe request. It then associates with an AP and negotiates a security policy and PMK is used as PSK. Then, 4-Way Handshake using EAPOL-Key Frames is performed as with 802.1x authentication in the previous case where PSK was not used. PTK/ GTK are generated by Authenticator and Supplicant similar to the non PSK case.

2.3 Confidentiality and Integrity using AES CCMP

RSNA architecture utilizes AES CCMP for providing data Confidentiality and Integrity. The Counter (CTR) Mode of AES is used for encrypting data to ensure Confidentiality, while Cipher Block Chaining (CBC) Mode is used to generate the

Message Authentication Code, namely MIC [22-23]. MIC is utilized to protect Integrity of data and authentication of the sender. The key length used in both modes of AES is 128 bits.

2.3.1 AES CCMP Architecture

Like any other block cipher, AES is also required to be used in a particular mode of operation. A mode of operation is an algorithm that employs the cipher to convert plain text into cipher text, or vice versa. While the encryption process provides Confidentiality of data, it does not ensure Integrity. To ensure Integrity of data, a Message Authentication Code is generally attached to the message. The Message Authentication Code utilizes a keyed cryptographic function to generate the Integrity check value. In case of 802.11i, 128 bit AES in CTR Mode is utilized to encrypt data, while CBC-Message Authentication Code (CBC- MAC) Mode is used to generate the MIC to ensure Integrity. Both modes utilize the same PTK derived during 802.1x Authentication for encryption and MIC calculation. However, they differ in Initial Vector (IV) generation mechanism.

2.3.1.1 Key derivation mechanism

The key generation mechanism of 802.11i architecture starts with key generation protocols of 802.1x authentication and authorization. The protocols generate random key material used by Authentication Server (AS) and Supplicant to derive the PMK. PMK is then transmitted to Authenticator by AS on a secured channel. RADIUS Server as AS is recommended to be utilized for copying PMK to the Authenticator. The Supplicant and Authenticator then generate 384 bit PTK that includes a 128 bit Key Conformation Key (KCK), a 128 bit Key Encryption Key

(KEK) and a 128 bit Temporal Key (TK). The key derivation mechanism is illustrated in Figure 2.5 [2].

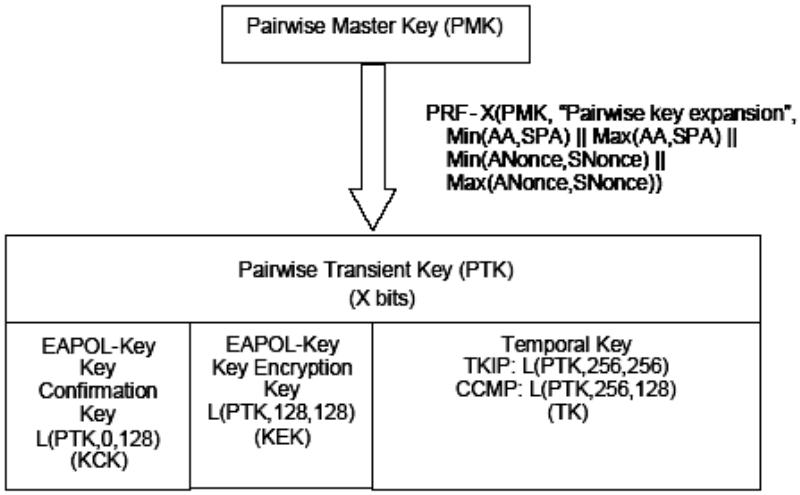

Figure 2.5. Key derivation mechanism

2.3.1.2 MIC Calculation Procedure

The MIC calculation procedure begins with derivation of IV by concatenation of Flags, Nonce and Length of Payload Fields [23]. Nonce is a unique value, never to be repeated for the same TK. This is followed by calculation of CBC-Message Authentication Code over the data part of the packet. The most significant 8 octets (64 bits) of the encrypted 16 octets (128 bits) are utilized as MIC that is appended to the data encrypted by CTR Mode. The MIC calculation procedure is depicted in Figure 2.6 [23].

2.3.1.3 AES Counter Mode Encryption

AES CTR Mode encryption starts with calculation of Nonce value by concatenating Priority, Packet Number (PN) and MAC Address 2 (A2) Fields. This is followed by generation of the Initial Counter by concatenation of Flags, Nonce and

Length of Payload Fields. This Initial Counter value is utilized as the IV or AES CTR Mode that runs iteratively by incrementing the counter value in each round till complete data contained in the MAC Protocol Data Unit (MPDU) is encrypted. The encryption and authentication process is depicted in Figure 2.7 [23].

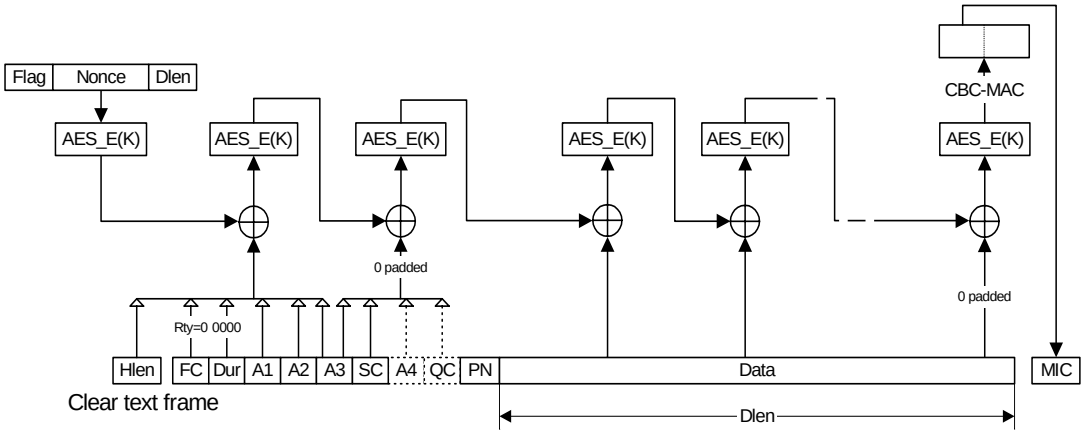

Figure 2.6. MIC Calculation Procedure

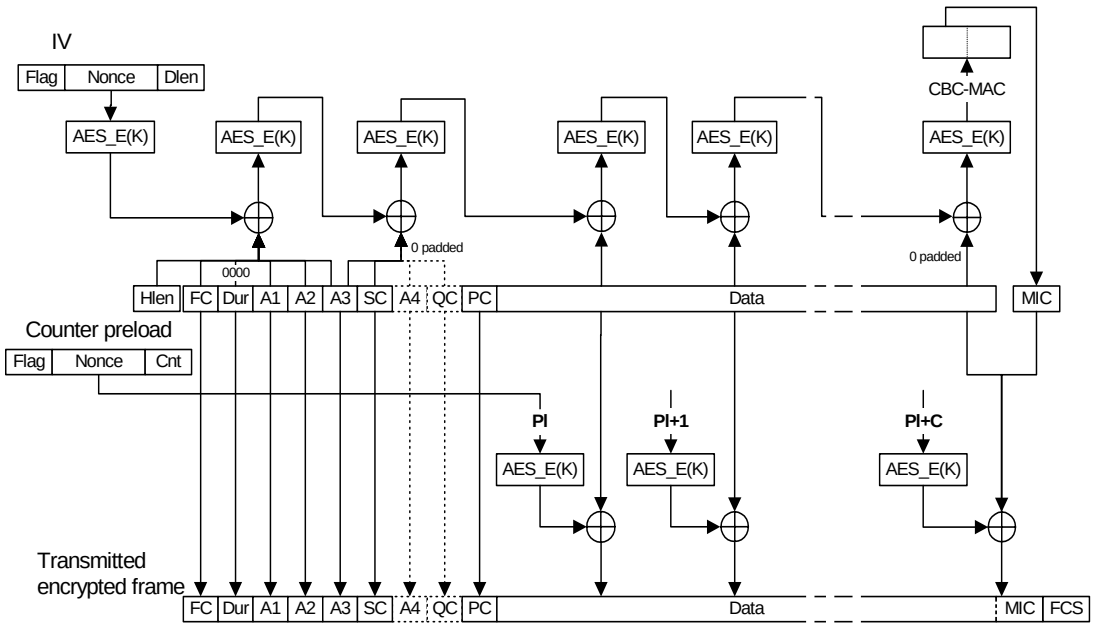

Figure 2.7. AES Counter Mode Encryption

2.3.1.4 Decryption and MIC Validation

During decryption, the complete reverse process is executed. Decryption utilizes AES CTR Mode to recover plaintext from encrypted MPDU. After recovering

plaintext, MIC is calculated using the CBC-MAC Mode. Calculated MIC is then compared with MIC attached with the MPDU to verify that data was not modified during transport, thus ensuring Integrity. Decryption and MIC validation procedure is highlighted in Figure 2.8 [23].

2.4 Attacks on 802.11i Based WLANs and Defense Mechanisms

Security offered by 802.11i RSNs has been analyzed comprehensively and it is believed that the protocol offers sufficient level of Confidentiality and Integrity of data. However, it is important to note that even 802.11i has not been designed to address potential threats to Availability as it was not an original design objective. The Management and Control Frames of 802.11 based WLANs are still unprotected/ unauthenticated. Consequently, WLANs are susceptible to Denial of Service (DoS) attacks, even with the deployment of 802.11i. Apart from these, some more attacks including Security Level Rollback, Reflection, 4-Way Handshake Blocking and RSN IE Poisoning have also been identified and demonstrated.

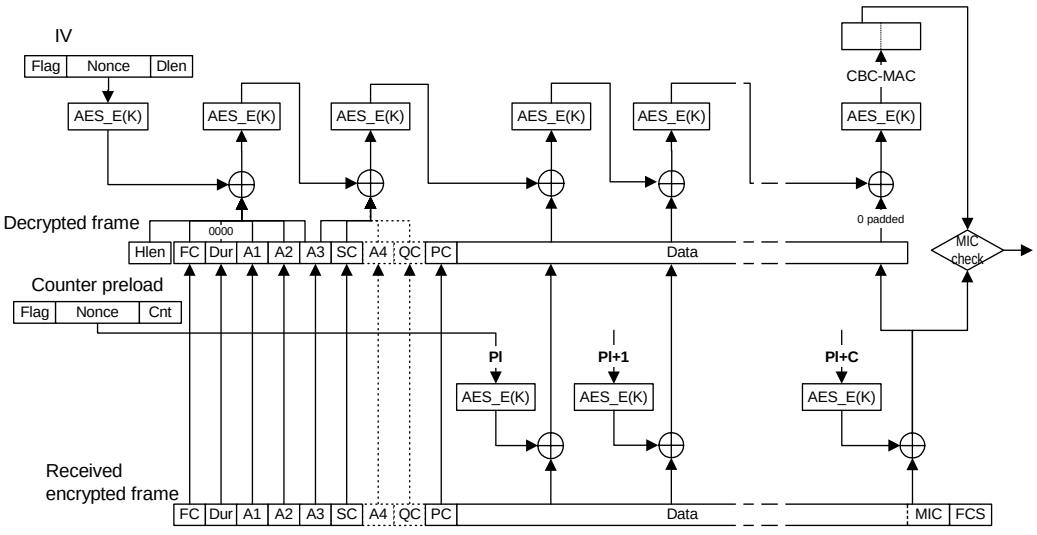

Figure 2.8. Decryption and MIC Validation

2.4.1 Security Level Rollback Attack

802.11i supports both Pre-RSNA and RSNA algorithms in case of Transient Security Network (TSN). Although 802.11i explicitly disallows Pre-RSNA algorithms when RSNA is used, an attacker can launch a Security Level Rollback attack in case of Pre-RSNA.. New WLAN implementations support Pre-RSNA algorithms in order to support migration to RSNA. A supplicant may enable accesses to both RSNA and Pre-RSNA capable networks to ensure Internet access under mobility. Correspondingly, an authenticator might be configured in a similar way to provide services to various supplicants. This hybrid configuration will degrade the security of the entire system to the lowest level as the attacker using attack can succeed in avoiding authentication and disclosing the default keys by launching this type of attack on the network [2].

A Security Level Rollback attack scenario is depicted in Figure 2.9 [2]. In this attack, the adversary impersonates as authenticator and forges Beacon or Probe Response Frames to indicate that only Pre-RSNA WEP is supported. Instead, the attacker can impersonate as supplicant as well, forging the Association Request Frame in a similar manner. As a result, the supplicant and the authenticator will establish a Pre-RSNA connection, even though both of them have RSNA support. Since there is no cipher suite verification in Pre-RSNA, the supplicant and the authenticator will not be able to detect forgery or confirm the used mode. The worst case arises if the adversary is able to discover default keys by exploiting the security weakness of WEP architecture, completely dejecting security. This attack is possible because the adversary could either perform a Man in the Middle attack or forge the starting management Frames timely, that is Beacon or Probe Response Frame to the Supplicant, or Association Request Frame to the authenticator.

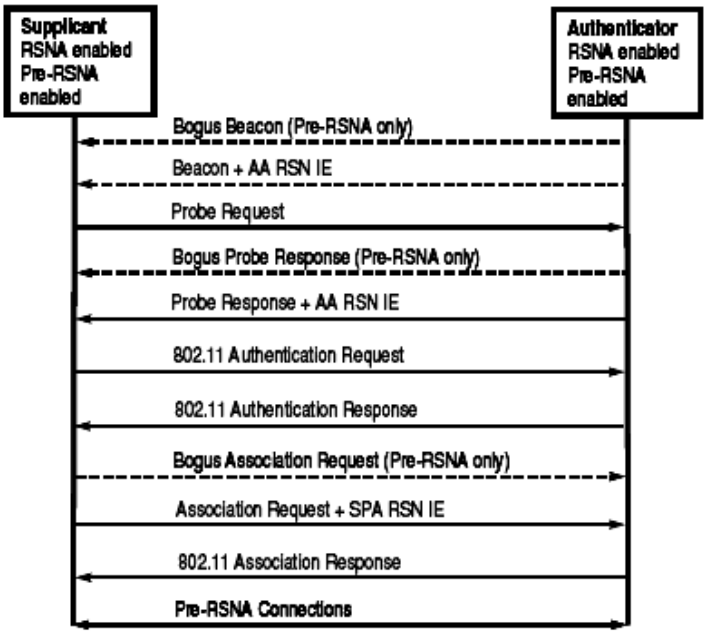

Figure 2.9. Security Level Rollback Attack

The solution to this type of attack is fairly simple. If possible, Pre-RSNA algorithms must be totally disallowed in the network. This would completely circumvent the attack. Obviously, such a solution is possible if all devices in the network are WPA2 compliant. The other option may be to implement a policy based network association Framework, where the information flow with strict security requirement should occur on RSNs and normal data flow may use Pre-RSNs.

2.4.2 Reflection Attack

The 4-Way Handshake uses symmetric cryptography to protect the Integrity of the messages. Since both authenticator and supplicant know the shared PMK, only they are able to calculate correct MICs and create valid messages, ensuring authentication as well. However, if a device plays the role of both the authenticator and the supplicant with the same PMK, an attacker can launch a common reflection attack as illustrated in Figure 2.10 [2].

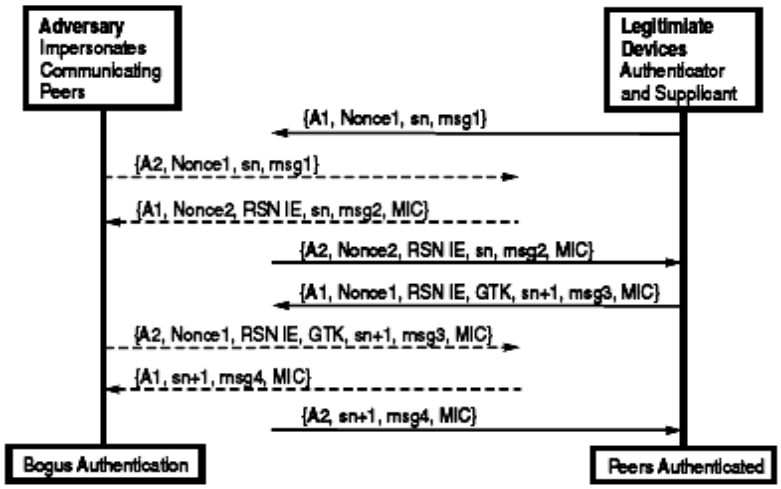

Figure 2.10. Reflection Attack

When the device initiates the 4-Way Handshake as authenticator, the attacker can initialize another 4-Way Handshake, with same parameters, but with victim device acting as the intended supplicant. Once the victim device is deceived to act as a supplicant, the attacker can use these messages as valid responses to the 4-Way Handshake initialized by the victim earlier. Obviously this scenario is not applicable to infrastructure networks, because a legitimate device will never act as both authenticator and supplicant. However, in Ad Hoc networks, 802.11i allows each device to serve both roles to distribute their own GTKs. This makes a reflection attack possible.

This attack can be mitigated by allowing a device to play only one role in the network and requiring separate PMKs if it is acting both as authenticator and supplicant.

2.4.3 DoS Attacks

DoS attacks are aimed at temporarily or permanently disrupting network access and service provisioning by constant flooding of legitimate or attack messages

on the service provider. In case of WLANs, individual clients in Ad Hoc mode or APs in Infrastrucure mode act as service providers. Thus they are targeted in these attacks to halt network access to specific clients or the complete network.

2.4.3.1 Management and Control Frame based Attacks

Management and Control Frames are unprotected and unauthenticated in 802.11i. Hence an adversary can easily forge these Frames to launch a DoS attack [10]. Amongst the Management Frame based attacks, the most prominent are Deauthentication or Disassociation Frame flooding. Amid Control Frames, severe problems exist in the virtual carrier-sense mechanism like RTS/ CTS and ACK messages. Furthermore, forging Power Save Poll (PS-Poll) messages is another method of launching DoS attacks on 802.11i based WLANs. Whilst may defense mechanisms have been proposed to counter these attacks, none of these address the complete range of attacks that can be launched based on the unprotected Management and Control Frames. Moreover, these involve implementation complexities and/ or hardware up gradation.

2.4.3.2 EAP Messages based Attacks

Several DoS attacks exploit the unprotected EAP messages in 802.1x authentication also. Specifically, an adversary can forge EAPOL-Start messages repeatedly to prevent successful 802.1x authentication, forge EAPOL-Success message to maliciously unblock the 802.1x data port of the supplicant without authentication, and forge EAPOL-Failure message and EAPOL-Logoff message to disconnect the supplicant. Fortunately, these vulnerabilities can be eliminated simply ignoring these messages. This does not affect the functionality of the protocol as the

outcome of the subsequent 4-Way Handshake could take the role of EAPOL-Success and EAPOL-Failure to indicate the authentication result. EAPOL-Logoff can be replaced by Deauthentication to disconnect a client and EAPOL-Start is not essential for protocol functionality [2].

2.4.3.3 Association Request Flooding

A DoS attack can also be launched on an AP by flooding forged Association Requests. This exhausts the EAP Identifier space that is 8 bits long (0-255). This weakness can be addressed by careful implementation. As the EAP Identifier is required to be unique only within a single 802.11 association, it is not necessary for the AP to deny new connection requests even when EAP Identifier space is exhausted. Thus, the AP can adopt a separate EAP Identifier counter for each association [2].

2.4.3.4 Michael Countermeasure Attack

In addition to the above DoS attacks, countermeasure associated with Michael algorithm used to compute MIC in TKIP is also vulnerable to DoS attacks. TKIP adopts the Michael algorithm to provide MIC protection for every MSDU (MAC Service Data Unit). Michael algorithm is designed to provide only 20 bits of security due to the limited computation power in legacy devices. Hence it is possible for an adversary to construct a successful forgery after every 2^{19} attempts. As a countermeasure, TKIP implements the following countermeasures to limit the rate of the forgery attempts from an adversary. The first Michael MIC failure is logged. If two successive failures are detected within 60 seconds, transmission and reception ceases for 60 seconds. Furthermore, authenticator can re-key or deauthenticate the supplicant, following which, supplicant should send out a Michael MIC Failure

Report Frame and deauthenticate itself afterwards. In an 802.11b network, an adversary can send out 2^{12} messages per second. Therefore, the adversary is able to make a successful forgery in about 2 minutes if countermeasure is not implemented. However, if countermeasure is deployed to limit the rate to 2 forgery attempts per minute, the attacker is limited to make one successful forgery every 6 months. Unfortunately, this countermeasure leaves an evident DoS vulnerability where an adversary can send out unsuccessful forgery attempts to cause two Michael MIC failures and drop a connection [2].

In order to prevent this DoS attack, the protocol checks the Frame Check Sequence (FCS), Integrity Check Value (ICV), TKIP Sequence Counter (TSC) and MIC sequentially. A MIC failure is only logged when the Frame has been received with correct FCS, ICV, TSC but an invalid MIC. Checking FCS and ICV can detect packet errors caused by noise, while checking TSC can detect replayed packets. Moreover, if the adversary modifies the TSC, the per-packet key will be modified simultaneously, which causes packet decryption to fail before a log of MIC failure. Hence, verifying FCS, ICV, TSC and MIC strictly makes DoS attacks more difficult. It is also worth mentioning that this attack is only applicable on WEP and WPA based security architecture in Pre-RSNs [2].

2.4.3.5 RSN IE Poisoning

RSN IE Poisoning is another possible attack on 802.11i based WLANs. In Message 2 of the 4-Way Handshake, authenticator verifies the MIC before the RSN IE, which is the correct order, but in Message 3, supplicant checks the RSN IE before MIC verification, and aborts if RSN IE is unequaled. An adversary can easily modify the RSN IE in Message 3 to cause handshake failure. However, even if check order is

correct, another fundamental attack exists to cause the RSN IE confirmation process to fail, which is depicted in Figure 2.11 [2]. An adversary can easily eavesdrop Beacon Frames of a legitimate authenticator, modify several insignificant bits in the Frame, modification of which does not affect the validity of the Frame and selection of cipher suites. For example, Reserved bits and the Replay Counter bits in the RSN Capabilities field are insignificant. The adversary then broadcasts this forged Beacon to poison the knowledge of RSN IEs by supplicants. Since this forged Beacon modifies insignificant bits only, supplicant and authenticator are still able to continue authentication and key management using effective security suites. However, the 4-Way Handshake will never succeed because the RSN IE confirmation will fail. Consequently, when the supplicant uses probe request instead, adversary can forge a Probe Response with modified RSN IE, which requires the adversary to interfere with the handshake in a more timely way. The adversary can also forge a Re-Association Request with modified RSN IE to poison knowledge of the authenticator; however, this approach is comparatively less efficient [2].

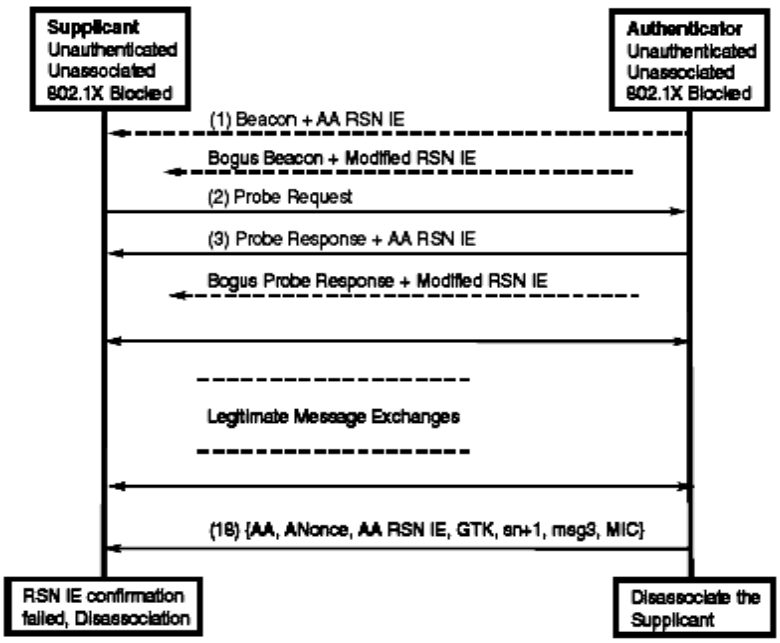

Figure 2.11. RSN IE Poisoning

This category of attacks can be mitigated by either authenticating the Management Frames or by loosening the condition of the RSN IE confirmation. Authenticator and supplicant can ignore differences of the insignificant bits in corresponding RSN IEs, while keeping the session secure. In a RSN IE, only authentication and key management suite selector is essential for subsequent handshakes as authenticator and supplicant are able to negotiate the encryption cipher suites securely after they finish authentication. If an adversary does not change authentication and key management suite selector, RSN IE could be accepted as correct authentication has been accomplished. Authenticator and supplicant can then use authenticated RSN IE in 4-Way Handshake for subsequent encryptions. Conversely, if the adversary modifies authentication and key management suite selector, t can be detected at the beginning of association. The association fails and supplicant quickly retries, without continuing message exchanges. In worst case, modification can be prevented in the 4-Way Handshake itself [2].

2.4.3.6 4-Way Handshake Blocking

Another possible DoS attack is the 4-Way Handshake Blocking. As 4-Way Handshake is the most essential component of the RSNA establishment, it cannot be neglected. In this handshake, the supplicant must accept all Message 1s in order to ensure that handshake can be completed in case of packet loss and retransmission. This allows an attacker to cause PTK contradiction between supplicant and authenticator by sending a forged Message 1 with a different Nonce value in between legitimate Message 1 and Message 3. In order to process forged Message 1s, supplicant has to store all the responding Nonces and derived PTKs. The supplicant can install correct corresponding PTK for data communications, only after a Message

3 with a valid MIC is received and then discard all others. Perceptibly, an adversary is able to launch a memory DoS attack by flooding forged Message 1s (Figure 2.12 [2]). The attack is critical because it is simple for the attacker to perform and a successful attack abandons complete the current authentication process [2]. Three possible approaches can be adopted to address this attack. Firstly, supplicant can implement a random-drop policy based queue to mitigate the vulnerability; however, it does not eliminate it. Secondly, Message 1 can be authenticated to defend against this attack, as authenticator and supplicant have already completed authentication. However, this requires some modifications to the message format. Moreover, authenticator must include a successively increasing Sequence Number in each Message 1 in order to prevent replays.

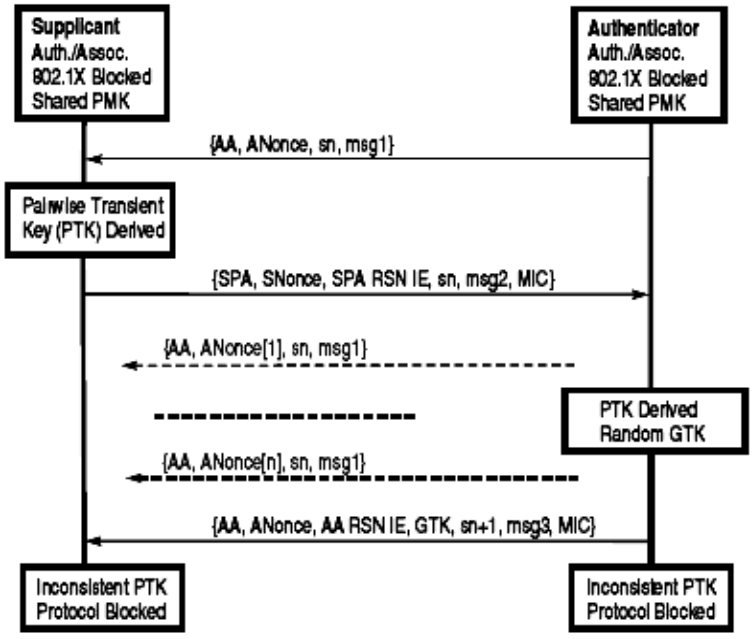

Figure 2.12. 4-Way Handshake Blocking

Thirdly, supplicant can innately eradicate this attack by re-using same Nonce for all received Message 1s until a successful completion of the 4-Way Handshake. In this case, the supplicant only needs to store one Nonce, calculate a PTK based on this

stored Nonce and Nonce in the received message and then verify the MIC. This approach only requires minor modifications in the algorithm and the supplicant need store only one Nonce, avoiding possible memory exhaustion [2].

2.4.4 TMTO Attack

As with any modern cipher, security of AES [21], used in RSNs as the cipher suite, is also dependant on the key used as the algorithm itself is public. AES counter mode uses this key (128 bits in length), to encrypt an Initial Counter Value. The result is exclusive ORed (XORed) with plain text to produce the first cipher text block. Counter is then incremented and operation is repeated to produce the next cipher text block. The procedure runs iteratively until complete plain text is encrypted. Hence, in case of counter mode, the security of the architecture is reliant upon key used and the Initial Counter value. Initial Counter is constructed by concatenating the Flags, Nonce and Length of Payload Fields, while the Nonce is obtained by concatenation of Packet Number (PN), Medium Access Control (MAC) Address A2 and Priority Fields.

Prediction of Initial Counter value results into lowering of effective key length for AES from 128 bits to 85 bits, which is less than the recommended effective key length of 97 bits for block ciphers. Consequently, AES counter mode becomes vulnerable to a Time Memory Trade Off (TMTO) attack [12-13]. Presently, no countermeasure has been proposed against this vulnerability and improvement of Initial Counter construction mechanism in AES CCMP is imperative.

2.5 Conclusion

The discussion in this chapter clearly suggests that efficient and comprehensive defense mechanisms against Management and Control Frame based

DoS Attacks, namely Deauthentication, Disassociation, RTS/ CTS/ ACK and PS-Poll message based attacks have not been proposed yet. Moreover, the AES CCMP vulnerability, leading to a possible TMTO [12] Attack, has not been addressed till date. Thus above two are identified as existing weaknesses in the 802.11i WLANs

Chapter 3

Attacks on Availability and Confidentiality

3.1 Introduction

The attacks on Availability, namely Management and Control Frame based DoS attacks and the possible attack on Confidentiality using AES CCMP vulnerability, clearly require effective and comprehensive countermeasures. Hence, these were identified as area of research for this thesis. In order to design defense mechanism against these, the attack Frameworks were studied in detail. These are briefly discussed in the subsequent subsections.

3.2 Management and Control Frame Based DoS Attacks

This category of attacks includes Deauthentication, Disassociation, RTS/ CTS/ ACK and PS-Poll message based attack. Incessant flooding of these messages causes permanent DoS on the target network/ nodes. Management and Control message based attacks are discussed one by one in the next subsections.

3.2.1 Deauthentication Message Attack

When a Supplicant discovers an AP via a Beacon Frame or a Probe Response to a Probe Request, it proceeds to authenticate itself to the AP. This is achieved by authentication mechanisms discussed in Chapter 2. The authentication mechanism under 802.11 also allows authenticated client or AP to deauthenticate itself with the other entity. The deficiency in this Framework is that Deauthentication message is

neither cryptographically protected nor authenticated, even in the 802.11i Standard [7]. As a result, any attacker may forge this message by either impersonating as the Supplicant or the AP. Consequently, the other entity egresses from the authenticated state and discards all subsequent communication, until the two get reauthenticated (Figure 3.1 [10]). Connection re-establishment time is dependant on a number of factors, including how insistently the client attempts to reauthenticate and higher-level time-outs or back-offs that may suppress the communication demand. Repeated transmission of these forged messages may deny service to the impersonated entity as long as desired.

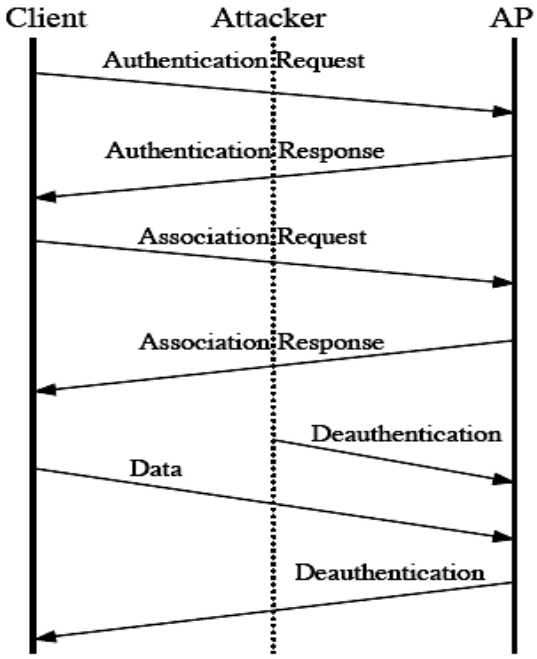

Figure 3.1. Deauthentication Message Attack

The immense strength of this attack lies in its flexibility. An attacker can choose to deny access to particular clients individually or even limit their access, in addition to simply denying service on the entire target channel. However, constant DoS to a particular Supplicant requires the attacker to continuously scan all channels and prevent the victim from authenticating itself to any other AP in the area [10].

3.2.2 Disassociation Message Attack

After a Supplicant authenticates itself to one or more APs in the perimeter, as allowed in 802.11, it has to associate itself with only one of them. This decides which of the Access Points would become responsible for routing traffic to and from the supplicant after successful association. Similar to the Deauthentication requests, Disassociation messages are used by the client to return to unassociated State. Since the Disassociation message is also unprotected and unauthenticated in 802.11i Standard [7], just as an attacker can forge Deauthentication messages to force a Supplicant to Deauthenticate, he can forge Disassociation messages to compel the supplicant into Unassociated state. This type of attack is, however, less effective than the previous one as the victim in this case is only required to Reassociate rather than Reauthenticate to resume communication [10].

3.2.3 RTS/ CTS/ ACK Message Attacks

A Virtual Carrier Sense mechanism is employed in 802.11 for avoiding collisions in the network from hidden terminals. Each Frame in 802.11 specifications uses a Duration Field to indicate the time interval for which a particular channel needs to be reserved. The Duration Field value is used in the Network Allocation Vector (NAV) on each entity. A Station is allowed to transmit only when its NAV value is zero. Same principle is applied in the RTS/ CTS Handshake for channel access synchronization in the event of interference from a hidden node in the network. The RTS/ CTS Handshake is performed as follows [8]. Transmitting client first sends an RTS Frame, specifying the duration necessary to complete the handshake, including response CTS Frame, Data Frame and final ACK Frame. Recipient responds with a CTS Frame that includes the duration field value,

indicating already elapsed time in the handshake. With this updated information, all nearby clients to the transmitting and receiving station update their respective NAVs to avoid transmission during the specified interval.

Unfortunately, the RTS, CTS and ACK messages are also neither protected nor authenticated in 802.11i based WLANs. This may be exploited by an attacker by spoofing any of the RTS/CTS or ACK messages, which control the NAV, and specifying an exaggerated value of Duration field. In this manner, the attacker is able to deny legitimate nearby clients to access the particular channel. However, using the RTS message is comparatively more beneficial to the attacker as this extends the attack perimeter through legitimate clients. The limit of NAV value is up to a maximum of 32767, which translates to approximately 32 milliseconds on the 802.11b network, thereby requiring the attacker to transmit 30 messages per second to block all access to a particular channel [10].

3.2.4 PS-Poll Message Attack

802.11 also includes power management Framework that allows power conservation in the network. Utilizing this feature, stations are allowed to conserve energy by entering into what is called a Sleep State. During this interval, stations are unable to send or receive data to and from the network. However, a station needs to first advertise its intent to enter the sleep state, following which, the AP starts storing all traffic intended for it. When the station exits the sleep state, it sends Poll message to the AP to indicate its liveliness, after which, the AP delivers stored data for the station and then deletes it from the buffer. Like all other Management and Control messages, the PS-Poll message is also unprotected and unauthenticated in 802.11i, allowing any attacker to forge it. A forged PS-Poll message forces the AP to deliver

and subsequently delete all available data for the impersonated station while it is actually in sleep state, denying data delivery to it when it actually awakens.

Similarly, the presence of buffered packets is indicated in a periodically broadcast packet called the Traffic Indication Map (TIM). It is possible for an attacker to deceive a client to believe that there are no buffered packets at the access point, when actually, there are. If the TIM message is spoofed, an attacker may persuade a client that there is no awaiting data for it and the client reverts back to the sleep state immediately [10].

3.3 Attack on Confidentiality

IEEE 802.11i Standard offers arguably uncompromised Confidentiality and Integrity Services by utilizing AES in CCMP mode. However the Nonce construction mechanism employed in the standard is weak, leading to Initial Counter prediction. Resultantly, the effective Key Length used for encryption is reduced from 128 to 85 bits and Time Memory Trade Off (TMTO) attack becomes a possibility [12-13].

3.3.1 Weak Nonce Construction in AES CCMP

In case of AES CCMP, the security of the architecture is reliant upon key used and the Initial Counter value [22-23]. Initial Counter is constructed by concatenating the Flags, Nonce and Length of Payload Fields, while the Nonce is obtained by concatenation of Packet Number (PN), Medium Access Control (MAC) Address A2 and Priority Fields [7]. This construction mechanism has been demonstrated to be weak, as the Fields utilized for calculation of Nonce can be easily sniffed by an attacker using readily available packet sniffing tools like “Ethereal”, “Wireshark” and “Airodump” etc. Thus the Nonce value can be predicted.

Nonce (104 bits) is obtained by simple concatenation of two field values from MAC header, namely the Priority (8 bit) and MAC Address A2 (48 bits), and PN (48 bits) field from the CCMP header. The 8 bit Priority field is currently set to '0' by default, as it is reserved for future use in Frame prioritization. MAC Address is easily obtainable and the dynamically changing PN field is set to '1' every time a TK is re-calculated [8]. Hence, reconstruction of Nonce becomes a fairly simple task. The Nonce reconstruction process is highlighted in Figure 3.2.

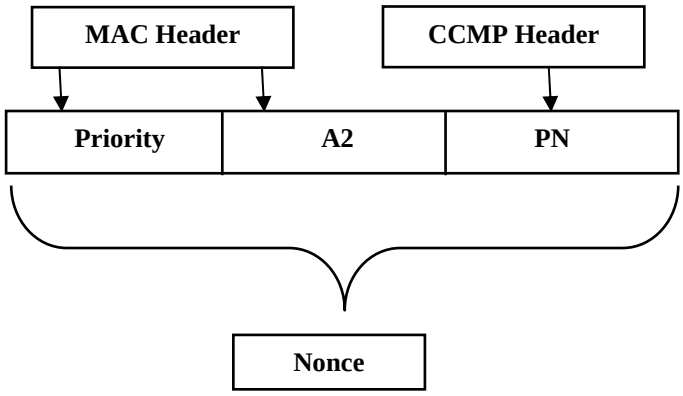

Figure 3.2 Nonce Reconstruction Process

3.3.2 Initial Counter Calculation

Reconstruction of Nonce is followed by calculation of the Initial Counter. This requires finding out values of Flags and Length of Payload Fields. Flags field contains a fixed value that is known. Thus, only Length of Payload field value is required to complete the Initial Counter prediction process. 802.11i Standard [7] specifies the maximum MPDU size as 2312 octets, out of which, 2296 octets are allocated to data while MIC and CCMP header occupy 8 octets each. In case of larger data size, which is the case on most occasions, fragmentation is used. Thus the Length of Payload becomes 2296 octets. Using this information, the bit string representation of Length of Payload field can be computed. Therefore, Initial Counter

prediction can be performed comfortably without completing a legitimate authentication process. The Initial Counter reconstruction methodology is depicted in Figure 3.3.

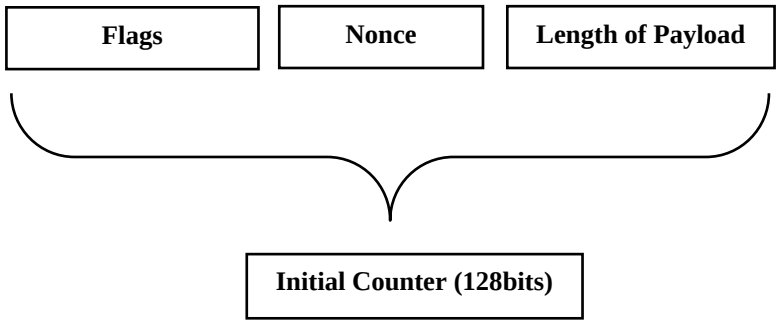

Figure 3.3 Initial Counter Calculation

3.3.3 Effective Key Length Reduction and TMTO attack

Analysis of counter mode suggests that prediction of Initial Counter value results into reduction of effective key length for AES from 128 bits to 85 bits [24]. This value is less than the recommended effective key length of 97 bits for block ciphers according to calculation, using the 1996 ad-hoc report on minimal key lengths and Moore's Laws [13], [26-27]. Consequently, AES counter mode becomes vulnerable to a Time Memory Trade Off (TMTO) attack [12].

The research work highlights that Initial Counter value of AES CCMP can be predicted [13]. The complete procedure utilized for deriving the Initial Counter value is also described. The process involves reconstruction of Nonce value, used to derive the Initial Counter, followed by calculation of the Initial Counter itself. The possibility of a TMTO attack on the protocol, therefore, becomes an imminent threat.

3.4 Conclusion

Management and Control Frame based DoS attack methodologies have been

discussed and it is established that WLANs continue to remain vulnerable to these attacks even after implementing 802.11i Standard [7] for security. A comprehensive, efficient and practicable defense mechanism is essential to counter these vulnerabilities in 802.11i based WLANs. Moreover, Nonce construction mechanism employed in AES CCMP mode is vulnerable to predictive analysis. Consequently, Initial Counter, used to construct the Key in CTR mode encryption employed in 802.11i for Confidentiality, can also be predicted. Hence the reduction of effective key length used in AES from 128 bits to 85 bits exposes the algorithm to Pre-computation attacks (like TMTO [12]). Though a Pre-computation attack on AES CCMP is still theoretical (to the best of knowledge), the threat cannot be overlooked and improvement of Initial Counter construction mechanism in AES CCMP is considered imperative.

Chapter 4

Proposed Defense Mechanisms

4.1 Introduction

Many defense mechanisms have been proposed to counter the Management and Control Frame based DoS attacks. Out of these, two approaches proposed appear to be promising [10, 11]. However, these also have certain deficiencies. In [10], an analysis of Frame Control and Body Fields in the general Management Frame of 802.11 specifications was performed and unused bits were identified. For example, in case of 802.11i, a total of 11 unused bits were detected in the Reason Code field of Deauthentication and Disassociation Frame body. The presented defense mechanism in utilizes the unused bits for inserting Random Bit streams, generated identically by the communicating entities through a pre agreed algorithm, for authenticating Disassociation and Deauthentication messages. It was also recommended that using more than 7 bits for authentication results in effectively defending against Deauthentication/ Disassociation flooding attacks [10]. However, the proposed work does not cover other type of DoS attacks based on Control Frames of 802.11 based networks. It also does not specify the mechanism or algorithm for generation of used Random Bit streams.

The defense mechanism proposed in [11] is another effective solution to Disassociation message DoS attack on 802.11 WLANs. The solution employs a Pseudo Randomized Sequence Number based authentication mechanism to defend against this category of attacks. It replaces the standard sequentially incremental 12 bit Sequence Number of the Disassociation message with a Pseudo Random Number

generated by utilizing the Pseudo Random Function (PRF), already defined in [7]. The PRF12 takes Pairwise Transient Key (PTK) or Group Transient Key (GTK) as input along with MAC Address of the Authenticator and Supplicant, and the previous Sequence Number as inputs to generate 12 bit Pseudo Random Sequence Number [11]. Since the authentication is performed using PTK or GTK, the attacker cannot forge the Disassociation message. Moreover, a pre defined function PRF in [7] is utilized for the Pseudo Randomized Sequence Number generation, making deployment very easy with no additional hardware requirement [11]. However, it works under a fundamental assumption that a Disassociation message is generated by a Supplicant only from 802.11i Associated State [11]. In contrast, a Supplicant can also forward a Disassociation message from 802.11 Associated and 802.1x Authenticated states, when it does not possess the key material namely PTK or GTK. Furthermore, the solution does not address other Management and Control message based attacks including Deauthentication flooding, RTS/ CTS /ACK and PS-Poll message attacks. The PRF defined in [11] has also not been utilized in a most effective manner as the only changing value, namely Previous Sequence Number, has not been included inside PRF, but is XORed with the random number generated by the PRF. This limits change in the overall pseudo randomized Sequence Number to few bits only in the subsequent messages between a particular set of supplicant and authenticator.

As far as the attack on Confidentiality is concerned, the AES CCMP weak Nonce construction vulnerability [13] is still unaddressed. The defense mechanisms against Management and Control Frame based DoS attacks and the attack on Confidentiality proposed in this thesis have been designed to be proficient,

comprehensive and easily deployable. Moreover they do not entail any implementation complexity or hardware up-gradation requirement.

4.2 Defending Management and Control Frame Based DoS Attacks

The defense technique proposed against this category of attacks is a one in all solution. The obvious choices to counter these threats are to either cryptographically protect these messages, or authenticate these Frames to defy an attacker using them to launch DoS attacks. Encrypting all the Management and Control Frames entails lot of overheads and processing requirements. Thus, authentication of messages was considered to be the appropriate choice for designing the defense mechanism.

4.2.1 Defense Mechanism Methodology

The proposed one in all solution is based on a modified Pseudo Random Number authentication mechanism that can be employed to counter all Management and Control Frame based DoS attacks on 802.11i WLANs. In order to identify a common field to be utilized for the proposed authentication mechanism, the format of Management and Control Frames specified in 802.11 Standard is first discussed.

4.2.2 Structure and Analysis of Management and Control Frames

The study of structure and analysis reveals that Frame Control and Frame Check Sequence (FCS) are the only common Fields present amongst all the Management and Control Frames. The Frame Control field contains fundamental information in its subFields and does not contain sufficient number of unused bits that can be utilized. This leaves only the FCS field which contains the IEEE 32-bit Cyclic

Redundancy Code (CRC). The general format of MAC Frames is illustrated in Figure 4.1 [7], Management Frame in Figure.4.2 [8], while that of Control Frames is shown in Figure.4.3 [7]. The FCS field is the only feasible common field which may be utilized for incorporating the Pseudo Random Number based authentication mechanism for all Management and Control messages.

Frame Control	Duration/ ID	Add 1	Add2	Add3	Sequence Control	Add4	Frame Body	FCS
Octets 2	2	6	6	6	2	6	0-2312	4

Figure 4.1 General Frame Format of MAC Frame in 802.11

Frame Control	Duration	DA	SA	BSSID	Sequence Control	Frame Body	FCS
Octets 2	2	6	6	6	2	0-2312	4

Figure 4.2 Frame Format of Management Frames in 802.11

RTS Frame format

Frame Control	Duration	RA	TA	FCS
Octets 2	2	6	6	4

PS-Poll Frame format

Frame Control	AID	BSSID	TA	FCS
Octets 2	2	6	6	4

CTS Frame format

Frame Control	Duration	RA	FCS
Octets 2	2	6	4

CF-End Frame format

Frame Control	Duration	RA	BSSID	FCS
Octets 2	2	6	6	4

ACK Frame format

Frame Control	Duration	RA	FCS
Octets 2	2	6	4

CF-End + CF-ACK Frame format

Frame Control	Duration	RA	BSSID	FCS
Octets 2	2	6	6	4

Fig 4.3 Frame Format of Control Frames in 802.11

4.2.3 Authentication Mechanism

It is proposed that the IEEE 32-bit CRC (CRC32), contained in the FCS field,

be replaced with CRC16. CRC32 has a probability of undetected error at 2.3 x 10^{-10}, which makes it 99.9999% accurate [27]. In comparison CRC16 has a probability of undetected error at 1.5 x 10^{-5}, giving it an accuracy of 99.9984% [27]. The difference in percentage of accuracy is just 0.0015%. Moreover, the bit error rate achieved satisfies the Shannon's Limit of 10^{-5}. Hence, the use of CRC16 instead of CRC32 would not affect the error detection capability in 802.11 MAC Frames by any significant amount. Additionally, it would also save computation time as CRC16 calculation is much faster as compared to CRC32. The replacement of CRC32 with CRC16 would leave 16 bits in the FCS field which will be utilized for inserting the Pseudo Random Number. The modified FCS Field is shown in Figure 4.4. The Pseudo Random Number can be generated using PRFn [7] with n = 16. Thus the PRF16 will output a 16 bit number, which will be inserted in the unused bits of the FCS field for authentication. The Pseudo Random Number generation in terms of PRF16 is expressed in Equation 4.1:-

Pseudo Random Number = PRF(PTK, A, AA || SPA || PSqN, 16) (4.1)

where,

"A" is the unique label used = "Pseudo Random Authentication Number",

"AA" = Authenticator MAC Address,

"SPA" = Supplicant MAC Address, and

"PSqN" = Previous Sequence Number (Including Fragment Number field).

Pseudo Random Number	**CRC16**
Octets **2**	**2**

Figure 4.4. Modified FCS Field

PRF is generated using the algorithm:-

PRF(PTK, A, AA || SPA ||, PSqN, 16)

for i ← 0 to (16+159)/160 do

R ← R || H-SHA-1(K, A, AA || SPA ||, PSqN, i)

return L(R, 0, n)

And H-SHA-1 is computed using Equation 4.2:-

H-SHA-1(TK, A, AA || SPA ||, PSqN, X) = HMAC-SHA-1(PTK, A, AA || SPA ||, PSqN, 16) (4.2)

The generated Pseudo Random Number will be used to authenticate the Deauthentication and Disassociation messages. The messages will be processed only if the number is valid, else they will be discarded. Since the mechanism is based on PTK, an attacker would never be able to forge any of the authenticated messages in absence of the key material. In case of Group Temporal Key Security Association (GTKSA), GTK can be utilized in PRF16 instead of PTK for broadcast messages. It is worth mentioning that the only changing field, namely Previous Sequence Number (PSqN), has been used inside the PRFn function, so that change in input corresponds to maximum change in the output value of Pseudo Random Number after computation of PRFn.

In case of Disassociation and Deauthentication messages initiated by a station prior to reaching 802.11i associated state, where it does not have PTK, these messages could be simply ignored. As a stable client would always attempt to complete the 802.11i Association and transit through states quickly, Deauthentication and Disassociation prior to 802.11i associated state can be accomplished by specifying Time-Out values. In case further authentication and association requests are not received from the respective station after expiry of this Time-Out value, the station can be Deauthenticated or Disassociated as required.

4.3 Countering AES CCMP Vulnerability

An improved Nonce construction mechanism can provide effective defense and is considered feasible to elude Initial Counter prediction. While designing the improved Nonce construction mechanism, analysis and recommendations on CTR mode security were studied and analyzed in detail [21-22], [24], [29-30]. A brief over view of analysis and recommendations on CTR mode security is presented in the next section.

4.3.1 Recommendations on Counter Mode Security

AES CCM Mode for 802.11 WLANs is proposed at [22, 23]. An independent review of CTR Mode security is presented and the possibility of TMTO attacks is also discussed [12, 24]. National Institute of Standards and Technology (NIST) has also presented recommendations regarding employment of CCM mode for Authentication and Confidentiality [28]. Their study reveals some important suggestions regarding defense against TMTO pre-computation attack. These include use of a larger key to increase effective key length, use of an unpredictable Initial Counter value (using a random Initial Counter value), adding 'n' randomized bits to the Initial Counter value to increase effective key length by 'n' (recommended value is 64 bits) or addition of few random bytes in the Nonce value to make pre-computation harder [12, 22, 24].

4.3.2 Improved Nonce Construction for AES CCMP

It has been established that to avoid prediction of Nonce value, randomization needs to be introduced. It is worth mentioning that two other Nonce values, Supplicant Nonce (SNonce) and Authenticator Nonce (ANonce), used in the 4-Way

Handshake of 802.11i standard are random numbers. Keeping the above in view, two approaches to improve the Nonce construction mechanism for AES CCMP are suggested.

First option is using a random Nonce value. An arbitrary Nonce would contribute 104 unpredictable bits to the Initial Counter value, making it computationally infeasible for the attacker to predict the Initial Counter [22]. It is suggested that Pseudo Random Function PRFn, already described in 802.11i Standard be utilized to generate 104 bit Nonce value. PRF would utilize the same field values presently being utilized by the Nonce construction mechanism. The Nonce generation in terms of PRFn is expressed in Equation 4.3:-

Nonce = PRF(TK, A, Priority || A2 || PN, 104) (4.3)

where,

"TK" = Temporal Key,

"A" is the unique label used = "Nonce",

"Priority" = Priority field of MAC header,

"A2" = MAC Address A2, and

"PN" = Packet Number value of CCMP header.

PRF is generated using the algorithm:-

PRF(TK, A, Priority || A2 || PN, 104)

for i ← 0 to (104+159)/160 do

R ← R || H-SHA-1(K, A, Priority || A2 || PN, i)

return L(R, 0, n)

And H-SHA-1 is computed using expression in Equation 4.4:-

H-SHA-1(TK, A, Priority||A2||PN, X) = HMAC-SHA-1(TK, A, Priority || A2 || PN, X) (4.4)

Alternatively, any of the NIST certified Pseudo Random Number Generator (PRNG), may also be utilized to generate the random Nonce [21]. Use of PRFn, however, has the advantage that it is already a part of 802.11i specification [7].

The second option is to use a random value for 8 bit Priority field. This would require an attacker to pre-compute a table for each of the 256 different possible values of Priority field, making pre-computation attack harder to execute [24]. Again, PRFn described above may be utilized to generate the 8 bit Priority field value with n = 8, or an alternative NIST approved PRNG may also be utilized for random Priority field value calculation [29].

4.4 Conclusion

A robust solution has been proposed to effectively counter all Management and Control Frame based DoS attacks by using Pseudo Random Number Based authentication. The mechanism involves replacement of Cyclic Redundancy Checksum 32 (CRC32) in the Frame Check Sequence Field (FCS) with CRC16 and using the spared 16 bits for authentication. Moreover, an improved Nonce construction scheme has been devised for AES CCMP to effectively prevent Initial Counter Prediction and the possibility of a subsequent TMTO attack [12]. The proposed technique involves randomization of the Nonce value to make it unpredictable.

Chapter 5

Implementation of Proposed Defense Mechanisms

5.1 Introduction

Implementation of any proposed architecture is an indispensable component that also aids to validate the efficacy of the devised approach. Two different approaches are primarily adopted for implementation, namely simulation and experimentation on actual hardware. For network architectures including WLANs, many simulators like “ns2”, “OPNET” and “OMNET” etc. are available and widely utilized. The results achieved through these simulators are also globally accepted by research community. However, it is worth mentioning that results of simulations can never be completely reliable, as opposed to testing on actual hardware. For the same reason, implementation on actual hardware was chosen to authenticate the performance and efficiency of the proposed defense mechanisms.

5.2 Selection of Platform

“wpa_supplicant” was chosen as the implementation package. It is an application for Linux, BSD, and Windows with support for WPA and IEEE 802.11i RSN (WPA2). It is suitable for both desktop/ laptop computers and embedded systems. Supplicant is the IEEE 802.1x/WPA component that is used in the client stations. It implements key negotiation with a WPA Authenticator and it controls roaming and IEEE 802.11 authentication/ association of WLAN device driver. “wpa_supplicant” is written in C and designed to be a "daemon" program that runs in background and acts as the backend component for controlling wireless connections.

It supports separate front-end programs; a text-based front-end (wpa_cli) and a GUI (wpa_gui) are also included. It is an open source program that is freely available with community support through a mailing list. It uses a flexible build configuration that can be used to select features to be installed. This allows minimal code size (from 50 KB binary for WPA/WPA2-PSK and 130 KB binary for WPA/WPA2-Enterprise without debugging code to 450 KB with most features and full debugging support). "wpa_supplicant" can also operate in Ad Hoc Mode.

As mentioned "wpa_supplicant" is available in Linux, BSD and Windows versions. However, the Windows version is in form of binaries constructed to operate from the windows command prompt environment. Therefore, Linux version was selected as the platform because of its flexibility and control over hardware and the application itself. Under Linux, wpa_supplicant can be easily modified and recompiled. Hence any modification in the WLAN standard architecture can be easily incorporated in the application and tested on actual hardware.

5.3 Authentication of Management and Control Frames

The Pseudo Random Number Authentication based defense mechanism proposed to counter DoS attacks on WLANs required modification of Management and Control Frames including Deauthentication, Disassociation, RTS/ CTS/ ACK and PS-Poll messages. Specifically, the FCS field of these messages required alteration. The CRC32 contained in the 8 Octet (32 bit) FCS field was to be replaced with CRC16. The spared 16 bits were to be used to insert the Pseudo Random Number generated using PRFn [7] with n=16.

First, the original "wpa_supplicant" package was used to create a wireless Ad Hoc network. Test network was setup using a total of 4 nodes. Two nodes operated

with Linux (Fedora Core 6) and utilized “wpa_supplicant” for wireless connection management. A Wireless Ad Hoc network was setup between the two nodes using WPA2. The third node was used as a monitor node utilized for network traffic monitoring and capturing. The fourth node was the attack node utilized to launch the DoS attacks. Deauthentication and Disassociation message DoS attacks were utilized for testing and validation, since they are the most widely used attacks. Successful defense against these two attacks would validate the efficacy of the proposed mechanism on all the Management and Control message based DoS attacks, since the proposed authentication mechanism is same for all Frames. The test network layout is depicted in Figure 5.1 and details of test network hardware are listed in Table 5.1.

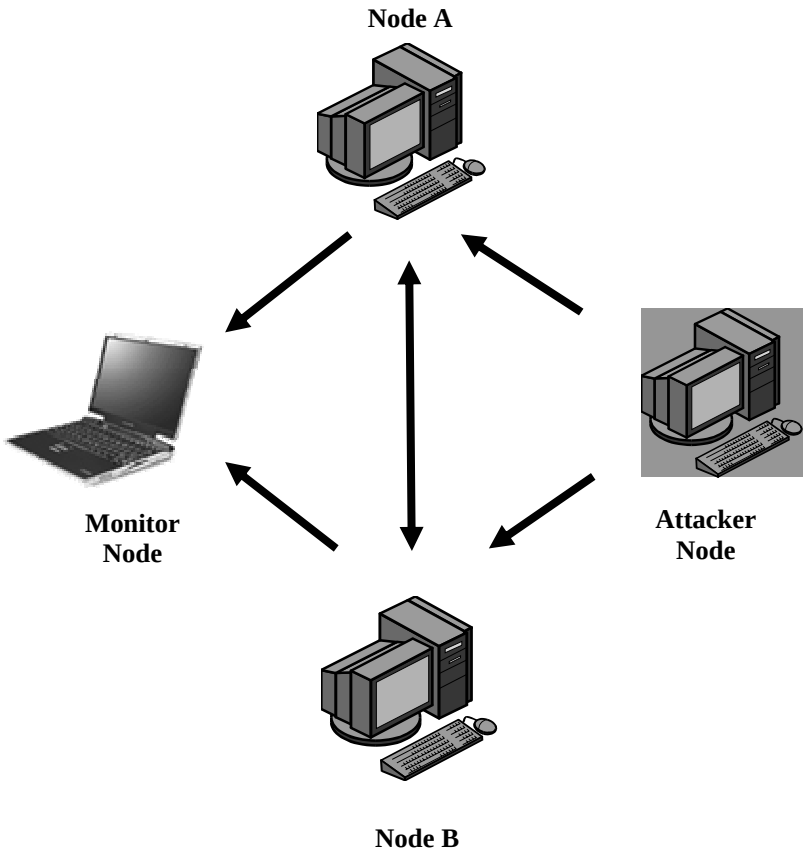

Figure 5.1. Test Network Layout

FTP sessions were then initiated for file transfer between Node A and Node B.

The Monitor Node continuously monitored and captured network traffic. The Attacker Node was then utilized to launch Deauthentication and Disassociation message based DoS attacks one by one. The effects of these attacks on network traffic were also recorded by the Monitor Node. Results are shown in Figures 5.2 and 5.3 respectively. X- axis shows time in seconds, Y- axis has number of packets exchanged. In Figure 5.2, the attack started at 300 and ended at 580 seconds, while in Figure 5.3, the attack window is between 290s to 560 seconds. The traffic within the attack window is from the attacker node.

Table 5.1 Details of Test Network Hardware

Node	Role	Hardware	WLAN card	OS	Application
Node A	Authenticator/ Supplicant	Desktop PC (Intel P-IV 2.4 GHz, 512 MB RAM)	D-Link DWL-G122 USB (Ralink rt73)	Linux (Fedora Core 6.0)	wpa_supplicant (Ad Hoc Mode)
Node B	Authenticator/ Supplicant	Desktop PC (Intel P-IV 2.4 GHz, 512 MB RAM)	D-Link DWL-G122 USB (Ralink rt73)	Linux (Fedora Core 6.0)	wpa_supplicant (Ad Hoc Mode)
Monitor Node	Traffic Capturing	Laptop (Dell Vostro1500)	Dell Wireless™ 1395 (Linksys WMP300N)	Windows Vista	Wireshark
Attacker Node	Launch DoS attacks	Desktop PC (Intel P-IV 2.4 GHz, 512 MB RAM)	D-Link DWL-G122 USB (Ralink rt73)	Backtrack 2.0 (Live CD)	Aireplay, Packetforge

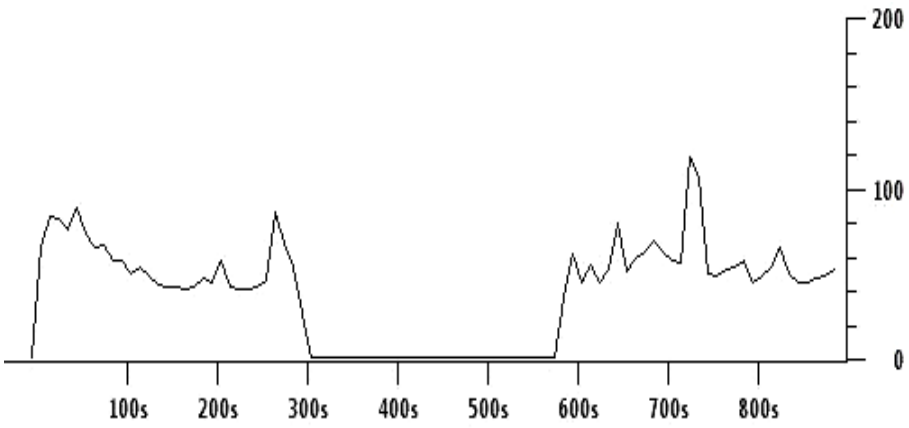

Figure 5.2. Network Traffic during Deauthentication Attack

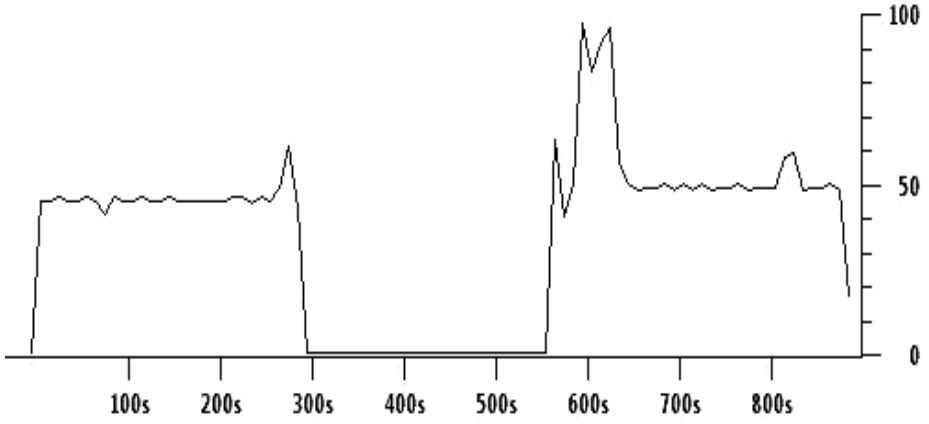

Figure 5.3. Network Traffic during Disassociation Attack

After obtaining results on the test network without defense implementation, wpa_supplicant package was modified to incorporate the proposed countermeasure. Modules related to Deauthentication and Disassociation Frame construction were amended to integrate the Pseudo Random Number based authentication mechanism. Same test network was then subjected to similar attack scenarios and results were recorded. The proposed defense mechanism successfully detected and stopped Deauthentication and Disassociation Frame attacks. Recorded results are highlighted in Figures 5.4 and 5.5 respectively. X- axis shows time in seconds, Y- axis has number of packets exchanged. In Figure 5.4, the attack started at 280 and ended at 440 seconds, while in Figure 5.5, the attack window is from 300 to 420 seconds. The altered modules of wpa_supplicant are included in Appendix "A".

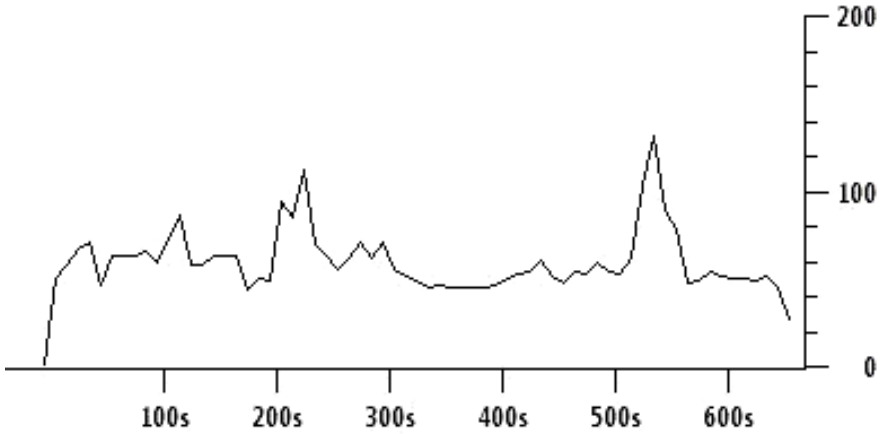

Figure 5.4. Network Traffic during defended Deauthentication Attack

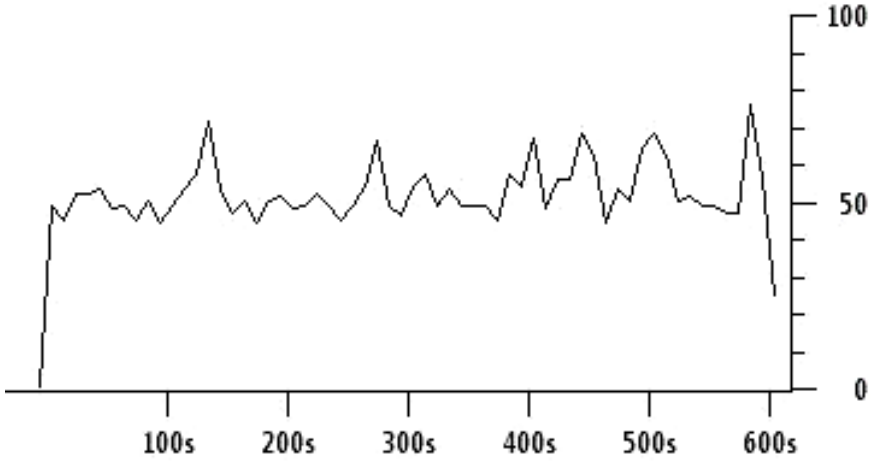

Figure 5.5. Network Traffic during defended Disassociation Attack

5.4 Improved Nonce Construction Scheme

Nonce construction mechanism used for AES CTR mode encryption in 802.11i RSNs was required to be modified. The two proposed construction methods were implemented in wpa_supplicant encryption modules one by one on test network nodes A and B. In the first case, Random Nonce was computed and utilized for Initial Counter calculation and subsequent encryption. PRFn [7] with n=104 was employed to derive the 13 octet Nonce. Since Nonce is computed only at the beginning of CTR mode encryption cycle, no evident overheads were recorded between the encryption time with original AES CTR mode Nonce construction and proposed Nonce derivation scheme for the same data. The average time for encryption with actual AES CTR mode encryption came out to be 0.01684 μsec, while with random nonce computation, it came out to be 0.01687 μsec. The measured processing time per encryption cycle for 50 instances is depicted in Figure 5.6. The amended encryption modules of wpa_supplicant package with Random Nonce construction are included in Appendix “B”.

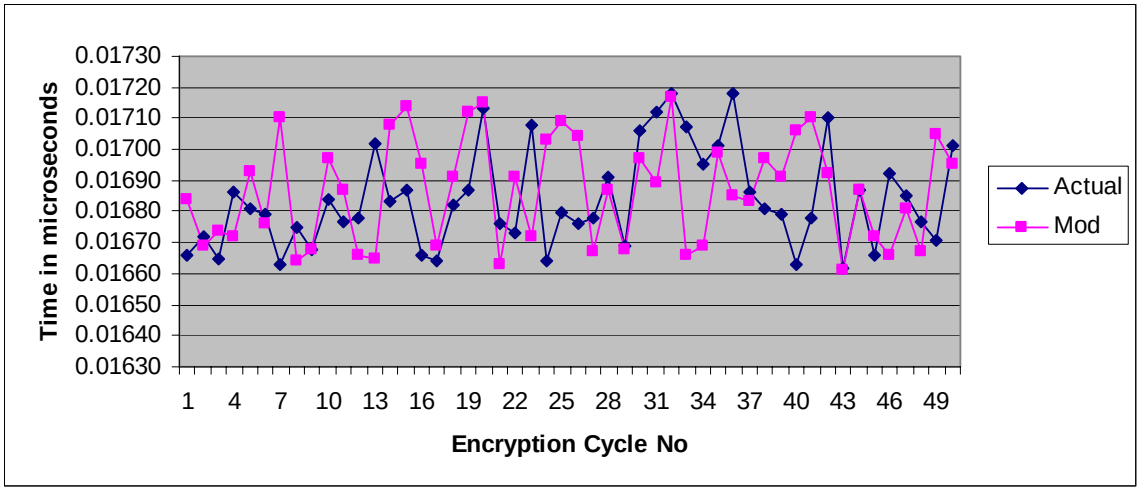

Figure 5.6. Measured processing time per AES CTR Mode encryption cycle

In case of the second proposed Nonce construction scheme, the 8 bit Priority field was computed using PRFn with n=8. This yielded in a Random number that was utilized to as Priority field value to construct the Nonce. The modifications were again incorporated in the encryption modules of wpa_supplicant successfully. Again, no apparent overheads were recorded between the encryption time with original AES CTR mode Nonce construction and proposed Nonce derivation scheme for identical data. The amended encryption modules of wpa_supplicant package with Random Priority field construction are included in Appendix "C".

5.5 Analysis of Proposed Solutions

The proposed solution for Management and Control Frame base DoS attacks is based on a modified Pseudo Random Number authentication mechanism to counter all discussed DoS attacks on 802.11 based WLANs. It employs the 16 bits, spared by replacing CRC32 in the FCS Field with CRC16, for authentication of these Frames. The replacement does not degrade the error detection capability by any significant amount and at the same time, mitigates the Management and Control Frame based

DoS attacks. It is an effective technique to resist Deauthentication, Disassociation and PS-Poll message based attacks. It also forces Non-Repudiation in case of RTS/ CTS and ACK message based attacks that would create sufficient deterrence for an attacker to launch these attacks. The Pseudo Random Number used for authentication is computed using PRFn, already described and utilized in 802.11i standard [7]. Since 16 bit authentication mechanism is employed with Pseudo Random Authentication Number, the probability of a successful forgery comes out to be $(0.5)^{16} = 1.5 \times 10^{-5}$.

In case of weak Nonce construction used in AES CCMP, Randomization of either the Initial Counter or the Nonce are two possible approaches to prevent Initial Counter prediction. Increase in key length may be another method, but is not considered practical since it involves addition of huge processing overheads. Randomization of Initial Counter can be achieved either by addition of random bits in the Nonce value, or by generating the complete Nonce randomly. Both approaches have been explored and proposed as potential solutions. In the first case, use of a Random Nonce value (104 bits or 13 octets) has been proposed. The Random value is computed using the same PRFn that is also utilized in 802.11i standard for PTK derivation. As a second possible solution, use of a Random Priority field (8 bits) has been suggested to add randomization into Nonce value. Again PRFn is used to calculate the 8 bit Priority field Random value. This would require an attacker to pre-compute a table for each of the 256 different possible values of Priority field, making pre-computation attack harder to execute. However, it is iterated that use of a 104 bit Random Nonce is a more robust countermeasure against TMTO attacks. Use of more random bits in Initial Counter increases the effective key length of the cipher, making it more resilient against pre-computation attacks. It is worth mentioning that the proposed improvement methodologies are in line with the recommendations on AES

CCMP and Counter Mode Security as well [3, 4, 21, 30].

5.6 Conclusion

Both defense mechanism were implemented and tested using wpa_supplicant package. In case of DoS attacks, the proposed authentication mechanism successfully identified and mitigated launched DoS attacks. Moreover, the improved Nonce construction scheme was also implemented and successfully tested. The scheme did not induce any noticeable overheads in AES CTR mode encryption used in 802.11i RSNs. Thus the proposed countermeasures were successfully tested on actual hardware using test network.

Chapter 6

Conclusion

6.1 Overview

Security analysis of 802.11i based WLANs revealed two distinct areas vulnerable to attacks. First were the unprotected and unauthenticated Management and Control Frames, which could be exploited to launch DoS attacks. Second was the weak Nonce construction mechanism of AES CCMP architecture utilized in 802.11i RSNs that could lead to Initial Counter prediction, and a subsequent TMTO attack to undermine complete Confidentiality architecture offered by the standard. The analysis of DoS attacks based on Management and Control messages in 802.11 WLANs and the proposed defense techniques revealed that none of them addressed the complete range of such attacks comprehensively. Moreover, they were either complex to implement or possessed deficiencies. Additionally, the weak Nonce construction vulnerability of AES CCMP was found unaddressed (to the best of knowledge).

The vulnerabilities were studied in detail and countermeasures have been devised against the posed threats. A robust solution to effectively counter all Management and Control Frame based DoS attacks by using Pseudo Random Number Based authentication has been proposed. An improved Nonce construction scheme has also been proposed for AES CCMP to effectively prevent Initial Counter Prediction and the possibility of a subsequent TMTO attack. Both countermeasures have also been implemented, tested and validated on actual hardware using a test network. In the end, an analysis of these proposed defense methodologies has also been presented, alongwith the results.

6.2 Achievements

Effective and practicable countermeasures have been proposed in this thesis to defend Management and Control Frame based DoS attacks and evade Initial Counter prediction in AES CCMP architecture. In case of DoS attacks, authentication of said messages has been proposed with a Pseudo Random Number, calculated using Pairwise Transient Key (PTK) that is inaccessable to adversaries. The defense proposed for AES CCMP vulnerability involves random calculation of nonce, strengthening the nonce construction mechanism of AES CCMP. Both defense mechanisms have been implemented and tested on actual hardware using a test network. The proposed techniques successfully counter the threats, are simple to implement by a software upgrades and do not require hardware upgradation. Proposed defense methodologies utilize a Pseudo Random Number generation mechanism that is already a part of 802.11i specifications [7], further facilitating employment. Same has been demonstrated by implementing the proposed techniques in wpa_supplicant package on Linux (Fedora Core 6.0) platform. The techniques do not compromise or affect the functionality of 802.11 WLAN architecture in any manner, making them robust and easily deployable solutions against the identified threats. Certain amendments are required in the 802.11i standard that are also considered practical as compared to the value of protection offered. The proposed solutions have also been accepted for publication [31, 32].

6.3 Limitations

The proposed countermeasures have been implemented on a WLAN test network in Ad Hoc mode, due to software inaccessibility on the available APs.

However, the defense mechanisms can be easily implemented on APs with open software control as well and testing can be subsequently performed for Infrastructure mode as well. Since the proposed techniques have been integrated and tested in Ad Hoc mode, no complications are anticipated in case of deployment on Infrastructure WLANs also.

6.4 Future Work

Deployment of proposed defense techniques on WLANs in Infrastructure mode may be undertaken as an implementation based research project. The project would involve implementation on wireless nodes as well as APs. Since implementation on wireless nodes has already been performed, the project would require accessing and understanding software of an AP, followed by necessary modification to incorporate the countermeasures. In the end, testing could be conducted in different attack scenarios to validate the efficacy of proposed defense mechanisms on Infrastructure mode WLANs.

Appendix A

Modified wpa_supplicant module for Pseudo Random Number Based Authentication

```
/*
 * WPA Supplicant - Client mode MLME
 * Copyright (c) 2003-2006, Jouni Malinen <j@w1.fi>
 * Copyright (c) 2004, Instant802 Networks, Inc.
 * Copyright (c) 2005-2006, Devicescape Software, Inc.
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU General Public License
 * version 2 as published by the Free Software Foundation.
 *
 * Alternatively, this software may be distributed under the terms
 * of BSD license.
 *
 * See README and COPYING for more details.
 *
 * Note :- Only the components modified are included to avoid
 * unnecessary details. Modified portions of code is highlighted as
 * bold text.
 */

#include "includes.h"
#include "sha1.h"
#include "aes_wrap.h"
#include "wpa_i.h"
#include "common.h"
#include "eloop.h"
#include "config.h"
#include "wpa_supplicant.h"
#include "wpa_supplicant_i.h"
#include "wpa.h"
#include "os.h"
#include "l2_packet.h"
#include "driver.h"
#include "mlme.h"
#include "eapol_sm.h"
#include "preauth.h"
#include "pmksa_cache.h"

/* Timeouts and intervals in milliseconds */
#define IEEE80211_AUTH_TIMEOUT (200)
#define IEEE80211_AUTH_MAX_TRIES 3
#define IEEE80211_ASSOC_TIMEOUT (200)
#define IEEE80211_ASSOC_MAX_TRIES 3
#define IEEE80211_MONITORING_INTERVAL (2000)
#define IEEE80211_PROBE_INTERVAL (60000)
#define IEEE80211_RETRY_AUTH_INTERVAL (1000)
#define IEEE80211_SCAN_INTERVAL (2000)
#define IEEE80211_SCAN_INTERVAL_SLOW (15000)
#define IEEE80211_IBSS_JOIN_TIMEOUT (20000)

#define IEEE80211_PROBE_DELAY (33)
#define IEEE80211_CHANNEL_TIME (33)
#define IEEE80211_PASSIVE_CHANNEL_TIME (200)
```

```
#define IEEE80211_SCAN_RESULT_EXPIRE (10000)
#define IEEE80211_IBSS_MERGE_INTERVAL (30000)
#define IEEE80211_IBSS_INACTIVITY_LIMIT (60000)

#define IEEE80211_IBSS_MAX_STA_ENTRIES 128


/* Information Element IDs */
#define WLAN_EID_SSID 0
#define WLAN_EID_SUPP_RATES 1
#define WLAN_EID_FH_PARAMS 2
#define WLAN_EID_DS_PARAMS 3
#define WLAN_EID_CF_PARAMS 4
#define WLAN_EID_TIM 5
#define WLAN_EID_IBSS_PARAMS 6
#define WLAN_EID_COUNTRY 7
#define WLAN_EID_CHALLENGE 16
/* EIDs defined as part fo 11h - starts */
#define WLAN_EID_PWR_CONSTRAINT     32
#define WLAN_EID_PWR_CAPABILITY     33
#define WLAN_EID_TPC_REQUEST  34
#define WLAN_EID_TPC_REPORT   35
#define WLAN_EID_SUPPORTED_CHANNELS 36
#define WLAN_EID_CHANNEL_SWITCH     37
#define WLAN_EID_MEASURE_REQUEST    38
#define WLAN_EID_MEASURE_REPORT     39
#define WLAN_EID_QUITE  40
#define WLAN_EID_IBSS_DFS     41
/* EIDs defined as part fo 11h - ends */
#define WLAN_EID_ERP_INFO 42
#define WLAN_EID_RSN 48
#define WLAN_EID_EXT_SUPP_RATES 50
#define WLAN_EID_VENDOR_SPECIFIC 221


#ifdef _MSC_VER
#pragma pack(push, 1)
#endif /* _MSC_VER */

struct ieee80211_mgmt {
      u16 frame_control;
      u16 duration;
      u8 da[6];
      u8 sa[6];
      u8 bssid[6];
      u16 seq_ctrl;
      u16 auth_numb;
      union {
            struct {
                  u16 auth_alg;
                  u16 auth_transaction;
                  u16 status_code;
                  /* possibly followed by Challenge text */
                  u8 variable[0];
            } STRUCT_PACKED auth;
            struct {
                  u16 reason_code;
            } STRUCT_PACKED deauth;
            struct {
                  u16 capab_info;
                  u16 listen_interval;
```

```
                        /* followed by SSID and Supported rates */
                        u8 variable[0];
                } STRUCT_PACKED assoc_req;
                struct {
                        u16 capab_info;
                        u16 status_code;
                        u16 aid;
                        /* followed by Supported rates */
                        u8 variable[0];
                } STRUCT_PACKED assoc_resp, reassoc_resp;
                struct {
                        u16 capab_info;
                        u16 listen_interval;
                        u8 current_ap[6];
                        /* followed by SSID and Supported rates */
                        u8 variable[0];
                } STRUCT_PACKED reassoc_req;
                struct {
                        u16 reason_code;
                } STRUCT_PACKED disassoc;
                struct {
                        u8 timestamp[8];
                        u16 beacon_int;
                        u16 capab_info;
                        /* followed by some of SSID, Supported rates,
                         * FH Params, DS Params, CF Params, IBSS Params,
TIM */
                        u8 variable[0];
                } STRUCT_PACKED beacon;
                struct {
                        /* only variable items: SSID, Supported rates */
                        u8 variable[0];
                } STRUCT_PACKED probe_req;
                struct {
                        u8 timestamp[8];
                        u16 beacon_int;
                        u16 capab_info;
                        /* followed by some of SSID, Supported rates,
                         * FH Params, DS Params, CF Params, IBSS Params */
                        u8 variable[0];
                } STRUCT_PACKED probe_resp;
                struct {
                        u8 category;
                        union {
                                struct {
                                        u8 action_code;
                                        u8 dialog_token;
                                        u8 status_code;
                                        u8 variable[0];
                                } STRUCT_PACKED wme_action;
                                struct{
                                        u8 action_code;
                                        u8 element_id;
                                        u8 length;
                                        u8 switch_mode;
                                        u8 new_chan;
                                        u8 switch_count;
                                } __attribute__((packed)) chan_switch;
                        } u;
                } STRUCT_PACKED action;
        } u;
```

```
} STRUCT_PACKED;

#ifdef _MSC_VER
#pragma pack(pop)
#endif /* _MSC_VER */


static void ieee80211_send_deauth(struct wpa_supplicant *wpa_s, u16
reason)
{
      u8 *buf;
      u8 exdata[2*ETH_ALEN + 16];
      size_t len;
      struct ieee80211_mgmt *mgmt;

      os_memcpy(exdata, addr1, ETH_ALEN);
      os_memcpy(exdata + ETH_ALEN, addr2, ETH_ALEN);
      os_memcpy(exdata + 2*ETH_ALEN, seq_ctrl , 16);
      seq_ctrl += 1;

      buf = os_zalloc(sizeof(*mgmt));
      if (buf == NULL) {
            wpa_printf(MSG_DEBUG, "MLME: failed to allocate buffer
      for "deauth frame");
            return;
      }

      mgmt = (struct ieee80211_mgmt *) buf;
      len = 24;
      os_memcpy(mgmt->da, wpa_s->bssid, ETH_ALEN);
      os_memcpy(mgmt->sa, wpa_s->own_addr, ETH_ALEN);
      os_memcpy(mgmt->bssid, wpa_s->bssid, ETH_ALEN);
      mgmt->frame_control = IEEE80211_FC(WLAN_FC_TYPE_MGMT,
                                  WLAN_FC_STYPE_DEAUTH);
      len += 2;
      mgmt->u.deauth.reason_code = host_to_le16(reason);

      len += 16;
      mgmt->auth_numb = sha1_prf(ptk, ptk_len, authnumb, exdata,
      sizeof(exdata), auth_numb, 2);

      ieee80211_sta_tx(wpa_s, buf, len);
      os_free(buf);
}


static void ieee80211_send_disassoc(struct wpa_supplicant *wpa_s,
u16 reason)
{
      u8 *buf;
      u8 exdata[2*ETH_ALEN + 16];
      size_t len;
      struct ieee80211_mgmt *mgmt;

      os_memcpy(exdata, addr1, ETH_ALEN);
      os_memcpy(exdata + ETH_ALEN, addr2, ETH_ALEN);
      os_memcpy(exdata + 2*ETH_ALEN, seq_ctrl , 16);
      seq_ctrl += 1;

      buf = os_zalloc(sizeof(*mgmt));
      if (buf == NULL) {
```

```
            wpa_printf(MSG_DEBUG, "MLME: failed to allocate buffer
            for disassoc frame");
            return;
      }

      mgmt = (struct ieee80211_mgmt *) buf;
      len = 24;
      os_memcpy(mgmt->da, wpa_s->bssid, ETH_ALEN);
      os_memcpy(mgmt->sa, wpa_s->own_addr, ETH_ALEN);
      os_memcpy(mgmt->bssid, wpa_s->bssid, ETH_ALEN);
      mgmt->frame_control = IEEE80211_FC(WLAN_FC_TYPE_MGMT,
                                  WLAN_FC_STYPE_DISASSOC);
      len += 2;
      mgmt->u.disassoc.reason_code = host_to_le16(reason);

      len += 16;
      mgmt->auth_numb = sha1_prf(ptk, ptk_len, authnumb, exdata,
      sizeof(exdata), auth_numb, 2);

      ieee80211_sta_tx(wpa_s, buf, len);
      os_free(buf);
}

static void ieee80211_rx_mgmt_deauth(struct wpa_supplicant *wpa_s,
                              struct ieee80211_mgmt *mgmt,
                              size_t len,
                              struct ieee80211_rx_status *rx_status)
{
      u16 reason_code;
      u16 auth_numb_rx;
      u8 exdatarx[2*ETH_ALEN + 16];

      os_memcpy(exdatarx, addr1, ETH_ALEN);
      os_memcpy(exdatarx + ETH_ALEN, addr2, ETH_ALEN);
      os_memcpy(exdatarx + 2*ETH_ALEN, seq_ctrl , 16);
      seq_ctrl += 1;

      if (len < 24 + 18) {
            wpa_printf(MSG_DEBUG, "MLME: too short (%lu)
deauthentication "
                       "frame received from " MACSTR " - ignored",
                       (unsigned long) len, MAC2STR(mgmt->sa));
            return;
      }

      if (os_memcmp(wpa_s->bssid, mgmt->sa, ETH_ALEN) != 0) {
            wpa_printf(MSG_DEBUG, "MLME: deauthentication frame
      received from unknown AP (SA=" MACSTR " BSSID=" MACSTR
                       ") - ignored",
                       MAC2STR(mgmt->sa), MAC2STR(mgmt->bssid));
            return;
      }


      auth_numb_rx = sha1_prf(ptk, ptk_len, authnumbrx, exdatarx,
      sizeof(exdatarx), auth_numb_rx, 2);

      if (auth_numb_rx == mgmt->auth_numb) {

            reason_code = le_to_host16(mgmt->u.deauth.reason_code);
```

```
            wpa_printf(MSG_DEBUG, "MLME: RX deauthentication from "
MACSTR
               " (reason=%d)", MAC2STR(mgmt->sa), reason_code);

            if (wpa_s->mlme.authenticated)
                  wpa_printf(MSG_DEBUG, "MLME: deauthenticated");

            if (wpa_s->mlme.state == IEEE80211_AUTHENTICATE ||
                  wpa_s->mlme.state == IEEE80211_ASSOCIATE ||
                  wpa_s->mlme.state == IEEE80211_ASSOCIATED) {
                  wpa_s->mlme.state = IEEE80211_AUTHENTICATE;
                  ieee80211_reschedule_timer(wpa_s,
                                    IEEE80211_RETRY_AUTH_INTERVAL);
            }

            ieee80211_set_associated(wpa_s, 0);
            wpa_s->mlme.authenticated = 0;

      }

      else {
            wpa_printf(MSG_DEBUG, "MLME: Deauthentication attack
message detected - ignored");
            return;
      }
}


static void ieee80211_rx_mgmt_disassoc(struct wpa_supplicant *wpa_s,
                                 struct ieee80211_mgmt *mgmt,
                                 size_t len,
                                 struct ieee80211_rx_status
*rx_status)
{
      u16 reason_code;
      u16 auth_numb_rx;
      u8 exdatarx[2*ETH_ALEN + 16];

      os_memcpy(exdatarx, addr1, ETH_ALEN);
      os_memcpy(exdatarx + ETH_ALEN, addr2, ETH_ALEN);
      os_memcpy(exdatarx + 2*ETH_ALEN, seq_ctrl , 16);
      seq_ctrl += 1;

      if (len < 24 + 18) {
            wpa_printf(MSG_DEBUG, "MLME: too short (%lu)
            disassociation frame received from " MACSTR " –
            ignored",
                     (unsigned long) len, MAC2STR(mgmt->sa));
            return;
      }

      if (os_memcmp(wpa_s->bssid, mgmt->sa, ETH_ALEN) != 0) {
            wpa_printf(MSG_DEBUG, "MLME: disassociation frame
            received from unknown AP (SA=" MACSTR " BSSID=" MACSTR
            ") - ignored",
                     MAC2STR(mgmt->sa), MAC2STR(mgmt->bssid));
            return;
      }

      auth_numb_rx = sha1_prf(ptk, ptk_len, authnumbrx, exdatarx,
      sizeof(exdatarx), auth_numb_rx, 2);
```

```
      if (auth_numb_rx == mgmt->auth_numb) {

            reason_code = le_to_host16(mgmt-
>u.disassoc.reason_code);

            wpa_printf(MSG_DEBUG, "MLME: RX disassociation from "
MACSTR
               " (reason=%d)", MAC2STR(mgmt->sa), reason_code);

            if (wpa_s->mlme.associated)
                  wpa_printf(MSG_DEBUG, "MLME: disassociated");

            if (wpa_s->mlme.state == IEEE80211_ASSOCIATED) {
                  wpa_s->mlme.state = IEEE80211_ASSOCIATE;
                  ieee80211_reschedule_timer(wpa_s,
                                 IEEE80211_RETRY_AUTH_INTERVAL);
            }

            ieee80211_set_associated(wpa_s, 0);
            }

      else {
            wpa_printf(MSG_DEBUG, "MLME: Disassociation attack
message detected - ignored");
            return;
      }
}
```

Appendix B

Modified wpa_supplicant module for Random Nonce construction

```
/*
* AES-based functions
*
* - AES Key Wrap Algorithm (128-bit KEK) (RFC3394)
* - One-Key CBC MAC (OMAC1) hash with AES-128
* - AES-128 CTR mode encryption
* - AES-128 EAX mode encryption/decryption
* - AES-128 CBC
*
* Copyright (c) 2003-2007, Jouni Malinen <j@w1.fi>
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU General Public License
* version 2 as published by the Free Software Foundation.
*
* Alternatively, this software may be distributed under the terms
* of BSD license.
*
* See README and COPYING for more details.
*
* Note :- Only the components modified are included to avoid
* unnecessary details. Modified portions of code is highlighted as
* bold text.
*/

/**
* aes_128_ctr_encrypt - AES-128 CTR mode encryption
* @key: Key for encryption (16 bytes)
* @nonce: Nonce for counter mode (16 bytes)
* @data: Data to encrypt in-place
* @data_len: Length of data in bytes
* Returns: 0 on success, -1 on failure
*/

#include "includes.h"
#include "common.h"
#include "aes_wrap.h"
#include "crypto.h"
#include "sha1.h"
#include "wpa_i.h"
#include "eloop.h"
#include "config.h"
#include "wpa_supplicant.h"
#include "wpa_supplicant_i.h"
#include "wpa.h"
#include "os.h"
#include "l2_packet.h"
#include "driver.h"
#include "eapol_sm.h"
#include "preauth.h"
#include "pmksa_cache.h"
#include "wpa_i.h"
```

```
int aes_128_ctr_encrypt(const u8 *key, const u8 *nonce,
                  u8 *data, size_t data_len)
{
      void *ctx;
      size_t j, len, left = data_len;
      int i;
      u8 *pos = data;
      u8 counter[BLOCK_SIZE], buf[BLOCK_SIZE];

      ctx = aes_encrypt_init(key, 16);
      if (ctx == NULL)
            return -1;

      nonce = sha1_prf(ptk, ptk_len, mnonce, nonce, sizeof(nonce),
      mnonce, 16);

      os_memcpy(counter, nonce, BLOCK_SIZE);

      while (left > 0) {
            aes_encrypt(ctx, counter, buf);

            len = (left < BLOCK_SIZE) ? left : BLOCK_SIZE;
            for (j = 0; j < len; j++)
                  pos[j] ^= buf[j];
            pos += len;
            left -= len;

            for (i = BLOCK_SIZE - 1; i >= 0; i--) {
                  counter[i]++;
                  if (counter[i])
                        break;
            }
      }
      aes_encrypt_deinit(ctx);
      return 0;
}

#endif /* CONFIG_NO_AES_CTR */
```

Appendix C

Modified wpa_supplicant module for Random Priority Field construction

```
/*
* AES-based functions
*
* - AES Key Wrap Algorithm (128-bit KEK) (RFC3394)
* - One-Key CBC MAC (OMAC1) hash with AES-128
* - AES-128 CTR mode encryption
* - AES-128 EAX mode encryption/decryption
* - AES-128 CBC
*
* Copyright (c) 2003-2007, Jouni Malinen <j@w1.fi>
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU General Public License
* version 2 as published by the Free Software Foundation.
*
* Alternatively, this software may be distributed under the terms
* of BSD license.
*
* See README and COPYING for more details.
*
* Note :- Only the components modified are included to avoid
* unnecessary details. Modified portions of code is highlighted as
* bold text.
*/

/**
* aes_128_ctr_encrypt - AES-128 CTR mode encryption
* @key: Key for encryption (16 bytes)
* @nonce: Nonce for counter mode (16 bytes)
* @data: Data to encrypt in-place
* @data_len: Length of data in bytes
* Returns: 0 on success, -1 on failure
*/

#include "includes.h"
#include "common.h"
#include "aes_wrap.h"
#include "crypto.h"
#include "sha1.h"
#include "wpa_i.h"
#include "eloop.h"
#include "config.h"
#include "wpa_supplicant.h"
#include "wpa_supplicant_i.h"
#include "wpa.h"
#include "os.h"
#include "l2_packet.h"
#include "driver.h"
#include "eapol_sm.h"
#include "preauth.h"
#include "pmksa_cache.h"
#include "wpa_i.h"
```

```
int aes_128_ctr_encrypt(const u8 *key, const u8 *nonce,
                  u8 *data, size_t data_len)
{
      void *ctx;
      size_t j, len, left = data_len;
      int i;
      u8 *pos = data;
      u8 counter[BLOCK_SIZE], buf[BLOCK_SIZE];
      u8 pty;
      ctx = aes_encrypt_init(key, 16);
      if (ctx == NULL)
            return -1;

      pty = sha1_prf(ptk, ptk_len, priority, nonce, sizeof(nonce),
      priority, 1);

      os_memcpy(nonce, pty, 8);

      os_memcpy(counter, nonce, BLOCK_SIZE);

      while (left > 0) {
            aes_encrypt(ctx, counter, buf);

            len = (left < BLOCK_SIZE) ? left : BLOCK_SIZE;
            for (j = 0; j < len; j++)
                  pos[j] ^= buf[j];
            pos += len;
            left -= len;

            for (i = BLOCK_SIZE - 1; i >= 0; i--) {
                  counter[i]++;
                  if (counter[i])
                        break;
            }
      }
      aes_encrypt_deinit(ctx);
      return 0;
}

#endif /* CONFIG_NO_AES_CTR */
```

Bibliography

[1] W. A. Arbaugh, N. Shankar, and J. Wang. "Your 802.11 Network has no Clothes". In *Proc. First IEEE International Conf on Wireless LANs and Home Networks*, pp. 131- 144, December, 2001.

[2] Changhua He and John C Mitchell, "Security Analysis and Improvements for IEEE 802.11i", in *Proc. 12th Annual Network and Distributed System Security Symposium (NDSS'05)*, 2005.

[3] J. Bellardo, and S. Savage. "802.11 Denial of Service Attacks: Real Vulnerabilities and Practical Solutions", in *Proc. USENIX Security Symposium*, pp. 15-28, August, 2003.

[4] Floriano De Rango, Dionigi Cristian Lentini and Salvatore Marano, "Static and Dynamic 4-Way Handshake Solutions to Avoid Denial of Service Attack in Wi-Fi Protected Access and IEEE 802.11i", in *Proc. EURASIP Journal on Wireless Communications and Networking*, Vol 2006, Article ID 47453, pp. 1–19.

[5] Ahmed M. Al Naamany , Ali Al Shidhani and Hadj Bourdoucen, "IEEE 802.11 Wireless LAN Security Overview", in *Proc. IJCSNS International Journal of Computer Science and Network Security*, Vol. 6 No.5B, May 2006.

[6] Alexandros Tsakountakis, Georgios Kambourakis and Stefanos Gritzalis, "Towards effective Wireless Intrusion Detection in IEEE 802.11i", in *Proc. Third International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, 2007. SECPerU 2007. 19 July 2007, pp. 37-42.

[7] IEEE Standard 802.11i. "Medium Access Control (MAC) security enhancements, amendment 6 to IEEE standard for Information technology - Telecommunications and information exchange between systems - local and

metropolitan area networks - specific requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications", April 2004.

[8] IEEE Standard 802.11-1999. "Information technology-Telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements-Part 11:Wireless LAN Medium Access Control and Physical Layer Specifications", 1999.

[9] Baber Aslam, M. Hasan Islam and Shoab A. Khan, "802.11 Disassociation DoS Attack and Its Solutions: A Survey", in *Proc. First Mobile Computing and Wireless Communication International Conference, MCWC 2007*. 17-20 Sept. 2006, pp. 221-226.

[10] Ying-Sung Lee, Hsien-Te Chien and Wen-Nung Tsai, "Using Random Bit Authentication to Defend IEEE 802.11 DoS Attacks", *http://hdl.handle.net/2377/3584*.

[11] Baber Aslam, M Hasan Islam and Shoaib A Khan, "Pseudo Randomized Sequence Number Based Solution to 802.11 Disassociation Denial of Service Attack", in *Proc. First Mobile Computing and Wireless Communication International Conference*, 2006. MCWC 2007.17-20 Sept. 2006, pp. 215 – 220.

[12] J. Hong and P. Sarkar, "Rediscovery of Time Memory Tradeoffs", 2005. *http://crypto/2005-590/hong.pdf*.

[13] M. Junaid , Dr Muid Mufti and M.Umar Ilyas, "Vulnerabilities of IEEE 802.11i Wireless LAN CCMP Protocol", in *Proc. World Academy Of Science, Engineering And Technology*, Volume 11, February 2006.

[14] R. Jueneman, S. Matyas, and C. Meyer. "Message Authentication", *IEEE Comm. Magazine*, 23(9), pp. 29-40, Sept. 1985.

[15] Stubblebine and Gligor. “On Message Integrity in Cryptographic Protocols”. in *Proc. IEEE Symposium on Research in Security and Privacy*, pp. 85-105, 1992.

[16] Core SDI. “CRC32 Compensation Attack against ssh-1.5”, July 1998: *http://www.coresdi.com*.

[17] A.Stubblefield, J.Ioannidis and A.Rubin, “Using the Fluhrer, Mantin, and Shamir Attack to Break WEP”. *Tech. Report TD-4ZCPZZ*, AT&T Labs.

[18] J. Walker, “Unsafe at Any Key Size: An Analysis of the WEP Encapsulation”, *Tech. report 03628E, IEEE 802.11 Committee*, March 2000.

[19] N. Borisov, I. Goldberg, and D. Wagner, Intercepting mobile communications: “The Insecurity of 802.11”, in *Proc. 7th Annual ACMIIEEE International Conf on Mobile Computing and Networking - Mobicom'01*, Rome, Italy, pp. 180-189, July 2001.

[20] S. Fluhrer, I. Mantin, and A. Shamir, “Weaknesses in the Key Scheduling Algorithm of RC4”. in *Proc. 8th Annual International Workshop on Selected Areas in Cryptography*, pp. 1-24, 2001.

[21] Specification for the Advanced Encryption Standard (AES), FIPS 197, U.S. National Institute of Standards and Technology. November 26, 2001. *http://www.nist.gov/aes*.

[22] D . Whiting , R . Housley , and N. Ferguson , “ AES Encryption & Authentication Using CTR Mode & CBC – MAC ” , *IEEE Doc . 802.11-02/144r2*, Mar 2002.

[23] O. Letanche, and D. Stanley, “Proposed TGi D1.9 Clause 8 AES - CTR CBC - MAC (CCM) text”, *IEEE Doc. 802.11-02/144r0* , Feb 2002.

[24] David A. McGrew, “Counter Mode Security : Analysis and Recommendations”, Cisco Systems, November 2002.

[25] M. Blaze, W. Die, R. Rivest, B. Schneier, T. Shimomura, E. Thompson, and M. Weiner, "Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security", January 1996. *http://www.counterpane.com/keylength.html*.

[26] "Moore's law". *http://www.Webopedia.com/TERM/M/Moores_Law.html*.

[27] Stone, J. Greenwald, M. Partridge and C. Hughes, J, "Performance of Checksums and CRCs over real data", in *IEEE/ACM Transactions on Networking*, Volume 6, Issue 5 Oct 1998, pp. 529 – 543.

[28] NIST Special Publication 800-38C, "Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality". May 2004. *http://csrc.nist.gov/publications*.

[29] NIST Special Publication 800-90, "Recommendation for Random Number Generation Using Deterministic Random Bit Generators (Revised)", Mar 2007. *http://csrc.nist.gov/publications*.

[30] Elio Perez, "802.11i (How we got here and where are we headed)", *http://www.sans.org/rr/whitepapers/wireless/1467.php*.

[31] Mansoor Ahmed Khan and Aamir Hasan, "Pseudo Random Number Based Authentication To Counter Denial of Service Attacks on 802.11" in *Proc. Fifth IEEE and IFIP International Conference on Wireless and Optical Communication Networks (WOCN2008)*, Indonesia, May 2008.

[32] Mansoor Ahmed Khan, Ahmad Raza Cheema and Aamir Hasan, "Improved Nonce Construction Scheme for AES CCMP To Evade Initial Counter Prediction", in *Proc. Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD2008)*, Thailand, August 2008.