ROBUST AND LOW LATENCY SECURITY FRAMEWORK FOR IEEE 802.11 WIRELESS NETWORKS

BY

MUHAMMAD JUNAID

A Dissertation Submitted in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy (Information Security)
to the Faculty of Information Security, College of Signals,
National University of Sciences and Technology

September 2009

ROBUST AND LOW LATENCY SECURITY FRAMEWORK FOR IEEE 802.11 WIRELESS NETWORKS

BY

MUHAMMAD JUNAID

A Dissertation Submitted in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy (Information Security)
to the Faculty of Information Security, College of Signals,
National University of Sciences and Technology

September 2009

Abstract

Wireless Networks call for enhanced confidentiality, integrity and authentication services because of their inherent weakness. 'Counter Mode Cipher Block Chaining Message Authentication Code Protocol' (CCMP) has recently been employed to replace flawed 'Wired Equivalent Privacy' (WEP) Protocol for the provision of security to IEEE 802.11 wireless local area networks (WLAN). Meanwhile, IEEE 802.11s – draft standard for wireless mesh networks (WMN) – has also proposed to use CCMP. CCMP, a two pass process, introduces considerable latency in multi-hop wireless networks, such as WMN. Increase in latency leads to a decrease in the quality of service for delay sensitive real-time multimedia applications.

This research exposes the vulnerability of CCMP against pre-computation time memory trade-off (TMTO) attack and proposes a framework to strengthen the security of WLAN packets using Per-Packet security mechanism. Furthermore, a novel, robust and low latency framework for WMN is also proposed. The architecture of security framework involves introduction of piggyback challenge response protocol for providing data confidentiality and data integrity. Piggyback challenge response protocol offers fresh encryption key for every packet, per-packet authentication and use of secret nonce. Authentication of every packet offers prompt defense against unauthorized access. It is also demonstrated that the security framework is robust against a variety of security attacks. Encrypted and unique nonce provides unpredictability and freshness. Unpredictability prevents pre-computation attack and freshness ensures successful defense against replay attacks. Proposed framework is simulated and its

performance is compared with IEEE 802.11i in terms of latency introduced by the security components. For single hop, latency due to the proposed protocol is less than half as compared to CCMP. The improvement in latency becomes more pronounced as the number of hops increase. This novel framework addresses the CCMP deficiencies of high latency and vulnerability against TMTO attack, without compromising any of the security measures implemented in the standard.

I dedicate this thesis to my parents and family

Acknowledgements

I am extremely thankful to Almighty Allah for his blessings upon me and giving me the direction at each and every step.

I am heartily indebted to my parents and family for their immeasurable faith and confidence in me, which had always been a source of inspiration and encouragement.

I wish to express my deepest appreciation to my thesis supervisor, Dr. Muid Mufti for his encouragement, logical understanding of my problems, invaluable guidance and ready assistance at the most of his precious time.

My sincere gratitude goes to Dr Muhammad Akbar, College of Signals and National University of Sciences and Technology for the provision of facilities for this research.

I would like to thank Dr Ejaz Muhammad, Dr. Noman Jafri and Dr. Saeed Murtaza for agreeing to serve on my Ph.D. as GEC member. Thank you also for your willingness to accommodate my defense and dissertation filing dates into your respective schedules.

I thank Higher Education Commission (HEC) of Pakistan for financially supporting my visit to University of New South Wales (UNSW), Sydney.

The Computer Science Department of UNSW has been extremely cooperative in carrying out the simulation work of our proposed security framework and extending assistance by providing Licensed Qualnet Simulation Software.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Introduction

Wireless networks are being widely deployed all over the world. This adoption of wireless networks is due to its simple and quick installation, inexpensive equipment, scalable network, less alterations and additions in buildings, and fascination of being wirelessly connected.

Unlike wired networks, signals of wireless networks are present in the air at ranges corresponding to their frequencies and can be received by everyone present in the vicinity. The wireless frequency is also a limiting factor for achieving significant throughput at required ranges. Likewise, quality of signals is badly affected due to interference of signals in the air. Accordingly, security and quality of service have been the challenging areas, over which much of research work for wireless networks has been concentrated in the recent past.

The technologies, which are emerging in the fields of computers and communications, are playing a vital role in the field of information security. These technologies make it possible to develop systems, which are cheaper in cost, faster and more efficient. The use of these systems increases the probability of penetration into the computer networks. Moreover, wireless networks, having ubiquitous signals, are more vulnerable to attacks than wired networks. Consequently, the task of the cryptographers, which is to make sure that information is seen by only those persons who are authorized

to see that information and the information reaches its destination unaltered, is becoming more difficult day by day. Therefore, the need to continuously search new ideas to strengthen the security and to explore new mechanisms by employing cryptographic tools is becoming more and more vital.

This Chapter introduces the area of research and reviews the wireless networks' security background and trends. The goal of the thesis, its general background and organization of the thesis is also presented in this Chapter.

1.2 Area of Research

Today, organizations are increasingly relying upon data networks to interconnect employees, partners and international markets. While the resulting connectivity has brought innumerable benefits, it has also increased the risk of becoming victims of digital attacks, resulting in serious business disruptions. Among other national security risks, cyber-attacks are also included as a potential threat and USA spent more than $100 million in the last six months repairing the damages and shielding the digital assets against cyber-threats and other computer network problems [1].

A variety of wireless network technologies are being employed to provide digital connectivity. Among them, IEEE 802.11 [2] based WLANs are being widely adopted. Laptops, PDAs, smart mobile phones, security cameras, parking meters, home entertainment devices, printers and peripherals are few of common platforms that are using WLAN devices. Moreover, Task Group (TG) IEEE 802.11s [3] is working on the extension of WLAN from single-hop to multi-hop WMN. WMN standard is enlarging the range of markets and applications for the WLAN. Applications of mesh

networks include unwired campuses and community area networks (hotzones). With a large, diverse and rapidly growing usage of WLAN, R&D efforts in this area remain very high. As a result, during the next few years WLAN will continue to become more secure, faster and value added. These advances will encourage continued adoption of WLAN, which will in turn drive even more R&D efforts.

The ingrained weakness of wireless LAN is its wireless medium. The wireless security protocols were mainly employed to make wireless networks secure, but they didn't prove to the task. They are not only vulnerable; they are robustly vulnerable. The vulnerability is deep-seated into the wireless network protocol, which makes it much harder to patch up [4]. A plausible problem with wireless medium is possibility of eavesdropping through any compliant receiver. This can result in breach of information, if strong encryption mechanism is not present. Moreover, as denial of service (DoS) attack, viruses, trojan horses etc. have become rampant and sophisticated, WLAN is no exception to be a victim of these attacks.

At the outset, some flaws were noticed in the encryption mechanism of IEEE 802.11 WLAN. The WEP protocol was hence developed. Again, some deficiencies were highlighted in WEP, both due to weak Initialization Vector (IV) and poor configurations in normal deployment [5]. Accordingly, 802.11i [6] replaced WEP by offering improved security through better confidentiality, integrity and authentication services in 802.11 WLANs. 802.11i offers access control mechanism through IEEE 802.1X [6] and data confidentiality and integrity through any of WEP, Temporal Key Integrity Protocol (TKIP) or CCMP.

IEEE 802.11s – the draft standard for WMN – also proposes to use CCMP of IEEE 802.11i for security in WMN. CCMP is a two pass process. In CCMP, the message is decrypted using counter mode to obtain the MAC header and the corresponding payload. Then the integrity check is carried out and if successful, the packet is decrypted and handed over to the next layer (i.e. network layer). While this two pass processing is suitable for single-hop WLAN, it induces considerable latency in multi-hop wireless networks, such as WMN, where the complete process of encryption and decryption needs to be performed at every intermediate hop. The subsequent increase in latency can lead to the decrease in the quality of service for delay sensitive services like voice over IP and video on demand. The breach in the WEP based WLAN security and the introduction of CCMP, having considerable latency due to two pass processing, motivated us to evaluate the IEEE 802.11i standard for possible vulnerabilities in authentication and access control mechanisms and to find ways to provide robust and low latency security mechanism.

1.3 Research Milestones

Aim of this research work is to critically analyze the existing security features of wireless networks in general and WLAN and WMN in particular with a view to propose a robust and low latency security mechanism. The work commences with reviewing different security mechanisms being used in WLAN, Wireless Metropolitan Area Network (WMAN) and Wireless Personal Area Network (WPAN). IEEE standards 802.11b [2], 802.11i [6], 802.11s [3], 802.15 [8], 802.16 [9] and 802.1X [7] are also extensively

surveyed to comprehend the cutting edge technologies in the field of wireless networks.

While surveying IEEE 802.11i (Standard for the security enhancement of WLAN) [6], vulnerability in the security mechanism is discovered [10]. This vulnerability is due to weak implementation of counter mode for block cipher, which renders the 802.11 WLAN exposed to TMTO attack. Our work [10] is also cited in the book, "On the Move to Meaningful Internet Systems", published by 'Springer Verlag'.

The next step is to find remedial measures to avoid TMTO attack on 802.11 based WLAN. In this endeavor, this research proposes a per packet authentication mechanism having capability to successfully defend TMTO attack [11].

In continuation to above, to strengthen the security of WMN, a novel, reliable and low latency security framework, suggesting piggy back authentication system has been proposed [12]. It is worth mentioning here that this work on piggy back authentication system, as a stand alone security mechanism, has also been published as a 'Technical Report' at UNSW, Sydney [13]. For the security framework of WMN, we provide the services of authentication and trust establishment using an extension of the 802.1X protocol over extensible authentication protocol (EAP). Proposed framework uses a novel piggybacked challenge response protocol for providing link level data confidentiality and data integrity at the MAC layer. The challenge-response mechanism also provides for per packet authentication. In addition, end-to-end data confidentiality and data integrity is realized at the network layer. It is also demonstrated that our framework

is robust against a variety of security attacks [12]. This security framework is simulated on Qualnet software during my visit to University of New South Wales (UNSW), Sydney. Finally, proposed framework is evaluated through simulation based experiments and its performance is compared against IEEE 802.11i in terms of the latency induced by the security components. For single hop, the latency due to our proposed protocol is less than half as compared to CCMP. The improvement in latency is more pronounced as the number of intermediate hops increases [12].

1.4 Organization of Thesis

This thesis is comprised of six chapters. This chapter presents the overview of the area of research, the goal of the thesis, major objectives of the research and a brief account of milestones covered during the research work. The rest of the thesis is organized as follows:

Second chapter is devoted to the literature review and description of existing vulnerabilities of Wired Equivalent Privacy (WEP) protocol and CCMP protocol.

Third chapter presents vulnerabilities in IEEE 802.11 WLAN CCMP Protocol inferred as a consequence of our research [10]. In this chapter it is shown that 'Nonce' can be reconstructed which in turn helps in reconstruction of initial counter value. Accordingly, it is established that CCMP is vulnerable to TMTO precomputation attack [10].

Fourth chapter presents proposed solution [11] to strengthen WLAN against the TMTO precomputation attack. In this chapter, the mechanism of

proposed Per Packet Authentication Protocol and its efficacy is discussed [11].

Fifth chapter explains proposed security framework for WMN [12]. Firstly, the general overview and the proposed architecture of the piggyback authentication mechanism are presented. Then, the efficacy of security mechanism to effectively thwart different attacks is discussed. Finally, this chapter presents the simulation results of proposed security mechanism which verifies the low latency characteristics of proposed security mechanism as compared to CCMP [12].

Last chapter concludes the entire research work. It first presents the achievements, contributions and summary of thesis. Suggestions for future work, arising out of this research work, are presented in the end.

Chapter 2 Vulnerabilities of IEEE 802.11 WLAN

2.1 Introduction

This Chapter is devoted to an overview of existing well known research work conducted to solve the security issues of WLAN and WMN. Existing vulnerabilities of IEEE 802.11 WLAN and methods that have been tried to enhance security of WLAN and WMN have been covered in detail in this Chapter.

2.2 Related Work

Wireless network security is being actively pursued by research communities. Various cryptographic mechanisms and frameworks have been recommended for different types of wireless networks [15 - 23]. For example, IEEE 802.11i has been recommended as a standard for MAC layer security of wireless LAN (WLAN) [6]. IEEE 802.11s – the draft standard for WMN – has also recommended to use IEEE 802.11i for security in WMN [3]. Improved cryptographic services have been incorporated into IEEE 802.11i. It uses IEEE 802.1X over Extensible Authentication Protocol (EAP) for authentication and authorization using a centralized authentication server such as RADIUS [7,24,25]. The centralized authentication server is usually placed on the wired network. AES is employed by the CCMP for providing data confidentiality and integrity [26]. CCMP is a two pass process. In CCMP, the message is encrypted by using the AES algorithm in counter mode. On the receiving side, the

message is decrypted to acquire the MAC and the corresponding payload. The integrity check is then carried out. Upon successful integrity check, the packet is decrypted and delivered to the upper layer (i.e. network layer). While this two pass processing is suitable for single-hop WLAN, it induces undesirable latency in multi-hop wireless networks where the complete process of encryption and decryption needs to be carried out at every intermediate hop. The increase in latency leads to the decrease in the quality of service for delay sensitive services like voice over IP and video on demand.

There is no explicit security framework proposed for WMN. However, various security frameworks have been proposed for multi-hop wireless networks like MANET [16 - 19] and WSN [20 - 22]. Kong et. al. [15] have proposed a distributed certification authority using threshold secret sharing cryptography. The resulting solution is highly scalable. Ren et. al. [20] have proposed the security framework for wireless sensor networks. Besides other features like distributed key establishment, authentication and confidentiality, the framework is explicitly tailored for WSN, where the traffic flows from sensors towards the base station using intermediate hops. However, the solution is extremely expensive in terms of communication and computation resources. The lack of a single administrative domain in case of WMN makes selfish behavior of nodes a major design challenge, rendering distributed trust establishment solutions like [16], [20] prone to service provisioning DoS attacks. The work from Soliman and Omari [23] is most relevant to our work. The authors have proposed the use of stream ciphers and an algorithm to produce permutations at random for multi-hop ad-hoc networks. These permutations work as a seed for the stream cipher. However, the randomness of their

algorithm can not be assured, since their scheme has not been tested using standard tests such as the NIST battery of tests [27]. They have proposed the elimination of the frame check sequence (FCS) field (required for error checking in 802.11 MAC frame) from the MAC header and the use of the key synchronization to detect corrupt packets. However, this requires the decryption of the complete packet before an error can be detected, which may not only be resource demanding but also adds to the delay.

Use of multiple schemes based on the available resources at the MANET nodes was proposed in [17]. Light weight security protocols for wireless sensor networks have been proposed in [21] consisting of two secure building blocks; SNEP and TESLA. SNEP deals with confidentiality and authentication, while TESLA provide authenticated broadcast for resource constrained environments like WSN. Park and Shin [22] have proposed the "light weight security protocol" for key management with limited resources as a design constraint. The major design constraint in these proposals is the resource limitation and lack of infrastructure for these networks. The limited resources lead to the tradeoff between security provisioning and the available resources. However, given the adequate level of resources (battery power, memory, computation and communication) that are available in case of WMN router nodes, a more robust security solution can be implemented.

2.3 Vulnerabilities in IEEE 802.11 WLAN

In this section working mechanism vis-à-vis vulnerabilities in the procedures / architectures of WEP and CCMP are explained.

2.3.1 Wired Equivalent Privacy (WEP)

WEP is the first security mechanism incorporated in IEEE 802.11 standard to minimize the risk of exchanging data in the open air over a medium having broadcasting nature. It is self-synchronizing, which allows for the loss of individual data frames without requiring re-initialization [28].

The main problem with WEP, identified by Jesse Walker, Network Security Architect for Intel's Platform Networking Group [29], is that it reuses the 24-bit Initialization Vector (IV) that is combined with a pseudo random number to construct the secret key. Because the IV is relatively short, and is transmitted in the clear as part of each data frame's MAC layer protocol, it is repeated with sufficient frequency that rest of the cipher can be relatively easily cracked [30]. By collecting a grouping of similar frames (such as TCP exchanges, which utilize identical formatting fields for every frame) that have used the same secret key and IV, enough correlating data can be compared to reveal the secret key [28]. Most of the WEP cracking programs, such as AirSnort (available at www.snort.org), depends on this approach.

More technical deficiencies in the IEEE 802.11 implementation of WEP also continue to be brought out in literature. For example, state table used to generate the first 256 bytes of WEP cipher stream is flawed. WEP can not defend against cryptanalytic attack based on comparison of the encrypted version of a known message (intercepted along with the WEP IV through passive sniffing) to repetitive IV based encryption combinations of the known text. Data modification is possible in transit by manipulating the

cipher text in special ways that do not change its cyclic redundancy checks [28].

Use of WEP is optional to user and this is considered as the most glaring flaw because it relies on users for availing the facility of WEP while using WLANs. When security depends on the novice user or hurried network administrator to undertake extra steps other than routine tasks, it often fails. Moreover, a good number of users that do utilize WEP fail to change the default passwords. This provides an opportunity to the hackers (war drivers) to intrude into WLAN without even needing to crack the WEP key.

In short, WEP can minimize the risk of being hacked by casual war drivers, but it may not thwart a determined intruder.

2.3.2 Issues with Service Set Identifiers (SSID) and Beacon Frames

WLAN Access Point (AP) and Stations (STA) transmit periodic keep alive frames to establish and maintain connections within their Basic Service Set (BSS). Generally, it is a misconception that an open system can also offer some level of security by deleting the network SSID from the beacon frames of an AP, ceasing to broadcast its beacon frame altogether, or even setting the AP to ignore all STA's probe frames not specifically addressed to its SSID.

There are two problems with any of these approaches. The first is that these actions violate the Wireless Ethernet Compatibility Alliance (WECA) standard known as "Wi-Fi." The standard ensures that the devices not employing active scanning are still able to make network connections. More

important (from a security perspective) however, is that the WLAN's SSID is broadcast in the clear as part of the association process, so potential intruders sniffing traffic in the service area are able to obtain the network's SSID despite the administrator's efforts to withhold it.

2.3.3 Issues with MAC Access Control List (ACL)

Just like wired LANs, WLANs can employ ACLs to define a group of users that are authorized to access the network. If a STA whose unique MAC address is not on the ACL of the particular AP with whom the STA is attempting to establish an association, the connection will be denied. Unlike wired LANs, however, the ACL for an AP must include both the SSID (which has no equivalent in wired LANs) as well as the client MAC address. WLAN MAC ACLs are particularly vulnerable to MAC spoofing because their two components are passed in the clear. The AP's SSID can be easily sniffed as explained above, and the MAC address of legitimate users may be similarly obtained from each frame that is passed between AP and STA.

It is not much difficult to modify the WLAN adapter's MAC address of the STA to one that is accepted by the target AP's ACL. A further disadvantage of WLAN ACLs is the administrative expense to maintain them, particularly if the WLAN is in conference room or coffee bar. Hence, ACLs are of very trivial importance from security stand point.

2.3.4 Weaknesses in Authentication Schemes

IEEE has developed 802.1X standard for authentication of wired and wireless LAN installations [6]. The authentication process is illustrated in Fig. 2.1. In step one, Mobile Unit (MU) requests authentication through the AP. The AP responds to probe requests and executes synchronization but holds connection authentication in abeyance until server authentication is complete. In step two, the AP forwards the MU's encrypted credentials to the Authentication Server (AS) such as RADIUS, which allows multiple MUs to share the same authentication database. In third step, the AS validates the user's password against its access database and access clearance is sent back to the AP. If the validation fails, the connection is terminated by the AP.

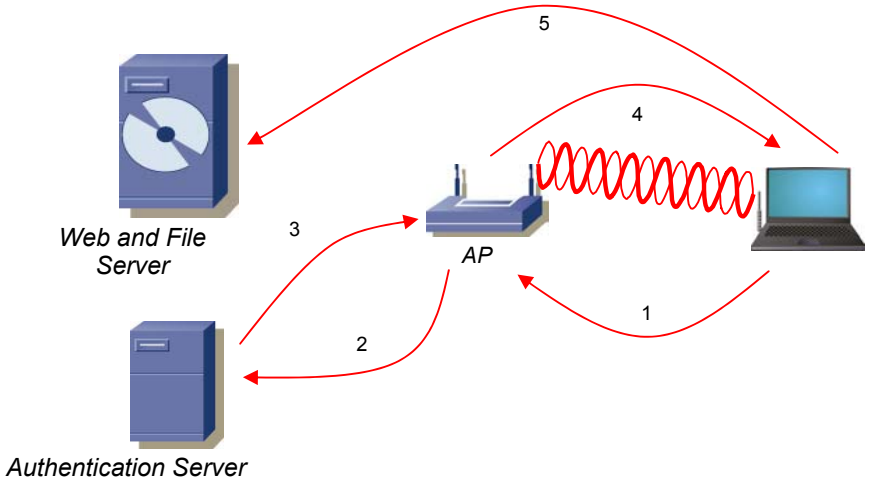


Fig. 2.1. IEEE 802.1X Authentication

Step four involves the activation of the AP port, the exchange of encrypted WEP keys, and full association with the AP. Finally in step five the MU is permitted access to general network and file servers.

There are two main drawbacks in the 802.1X standard. The first is that the authentication database is in a single location that, if compromised, would leave the WLAN exposed. The second is that it does not provide complete network protection because it only addresses the need for station authentication [31]. If used alone, the 802.11x WLAN will suffer the same deficiencies in confidentiality as an ordinary WLAN because it must rely on WEP to encrypt the data frames being exchanged.

2.4 CCMP Security Mechanism

To address the security deficiencies of WEP (discussed in section 2.3), IEEE has introduced 802.11i and incorporated it into the IEEE 802.11-2007 Standard. 802.11i offers improved security by employing confidentiality, data integrity and authentication services for WLANs. 802.11i employs IEEE 802.1X for node authentication [7]. Data security is provided through any of WEP, Temporal Key Integrity Protocol (TKIP) or CCMP.

CCMP employs Advanced Encryption Standard (AES) block cipher algorithm, in a counter mode [32]. Counter mode is used to employ block ciphers as a component of stream generators to provide data confidentiality. In the Counter mode, the initial counter is first encrypted and then X-ored with the plaintext to generate ciphertext [33]. Whereas, CBC-MAC provides message integrity.

In CCMP, fresh temporal key is recommended to be used for every session. Moreover, use of unique nonce value is recommended for each frame and 48-bit packet number (PN) is used to produce unique nonce value [6]. MAC layer Packet Data Unit (MPDU) comprises MAC header, CCMP header, FCS field, encrypted payload and encrypted MIC. CCMP MPDU structure is shown in Fig. 2.2 [6].

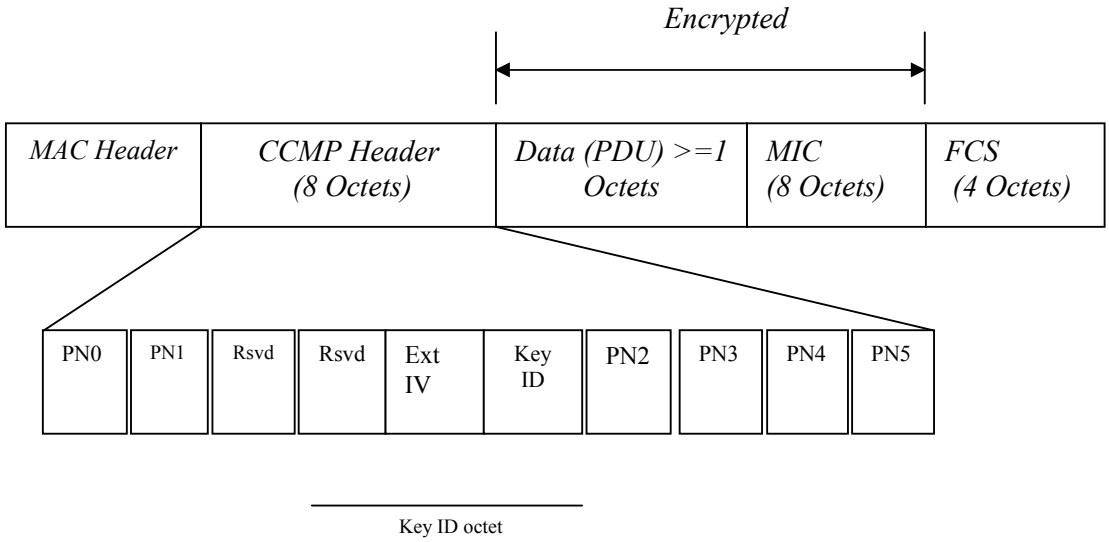


Fig.2.2. CCMP MAC Packet Data Unit

The CCMP encryption process is depicted in Fig. 2.3 [6]. Encrypted MPDU is produced in steps [6]. In the first step, Packet Number (PN) is incremented for each MPDU. When retransmitted, MPDUs are not modified. The second step is to form additional authentication data (AAD) by using fields in the MPDU header. Integrity services are provided to the

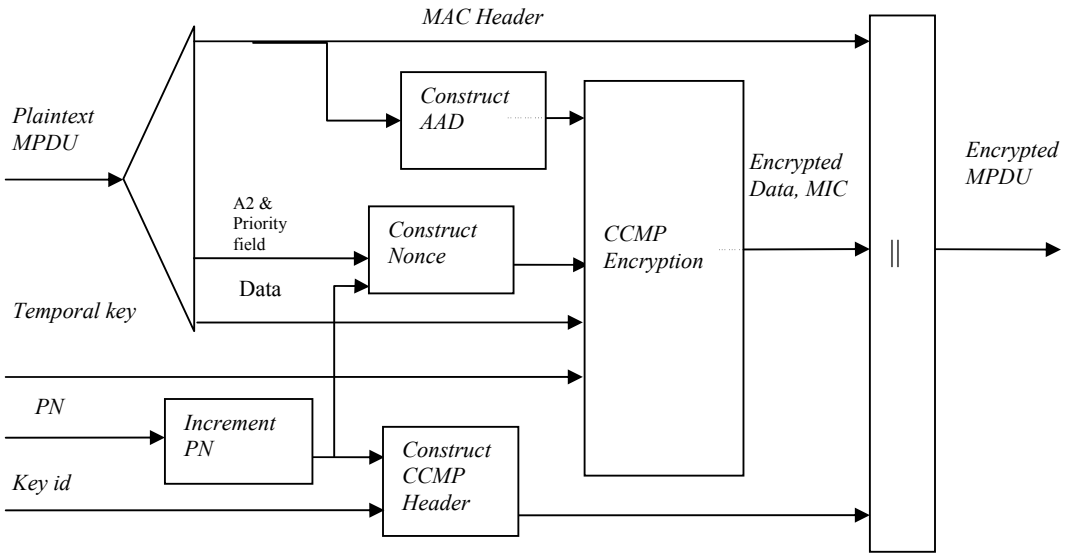


Fig. 2.3. CCMP Encapsulation Block Diagram

fields included in the AAD. For retransmission, when calculating the Additional authentication data and packet header fields that may change on retransmission are masked to 0. In the third step, Nonce block is formed from the packet number, source address and priority field. Priority field value is set to 0. Fourth step put fresh PN and the key identifier into the 8-octet CCMP header. In the fifth step Cipher text and MIC is produced. Finally, encrypted MPDU is produced.

On receiving the encrypted MPDU, CCMP undertakes procedure depicted in Fig. 2.4 [6]. In the first step, AAD and nonce values are extracted from the encrypted MPDU. These values are extracted by parsing the encrypted MPDU. AAD is obtained from the MPDU header. Priority field, A2 and PN form the nonce. For integrity checking, MIC is also obtained. The CCMP decryption process recovers the MPDU plaintext data and checks the integrity of additional authentication data and packet's plaintext data by using the TK, MIC, additional authentication data, nonce and packet's

cipher text data. Finally, plaintext is formed by concatenating the received MPDU header and MPDU plaintext data.

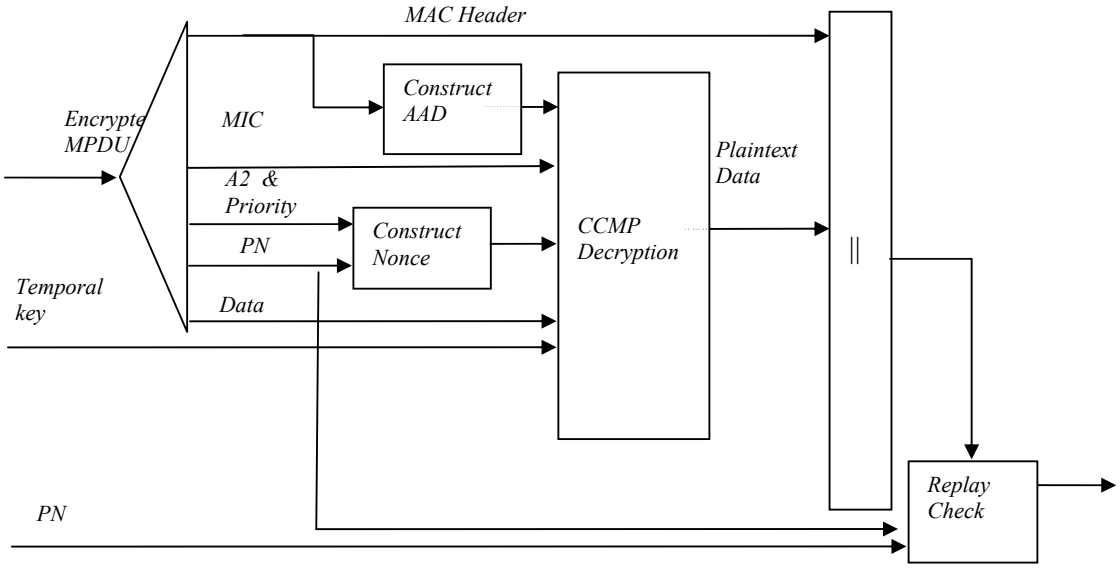


Fig. 2.4. CCMP Decapsulation Block Diagram

In our work [10], it is shown that CCMP is vulnerable to TMTO attack due to predictable initial counter value. Consequently, the security level of AES algorithm (128 bits key length) in counter mode recedes below the recommended strength for block ciphers [34]. The description of this proof is given in the Chapter 3 [10].

Chapter 3 Proposed Vulnerabilities in CCMP Protocol of IEEE 802.11 WLAN

3.1 Introduction

In this chapter, it is shown that the initial counter value of CCMP (explained in Chapter 2) can be predicted, which subsequently renders the WLAN vulnerable to TMTO attack [10]. Section 3.2 and 3.3 present the method that may be adopted by adversary, to pre-compute the nonce and counter block value. Section 3.4 explains TMTO attack on CCMP and Section 3.5 concludes the chapter.

3.2 Reconstruction of Nonce

The CCMP encapsulation and de-capsulation processes along with description of its packet header have been explained in Chapter 2. The nonce block used in the CCMP constitutes three fields. These fields are priority field (set to '0' by default), address field (MAC header A2 field) and Packet Number (PN) field as given below in Eq. 3-1:

Priority Field || Address (A2) || Packet Number (PN) = Nonce (3-1)

The construction of nonce has been devised in such a manner that it is possible for an adversary to rebuild it. The priority field occupies first 8 bits of the nonce. These 8 bits are kept as '0' value. However, in future these bits may be utilized for 802.11 frame prioritization. The next 48 bits in the

nonce are A2 address field. These 48 bits are taken from the A2 address of MAC header field. The remaining bits of the nonce are occupied by the Packet Number (PN) field. This PN field is the only field of nonce which is dynamic in nature. But, even this field is increasing sequentially for every MPDU. Moreover, [34] recommends in sub-clause 8.3.3.4.3 that whenever fresh/initialized temporal key is activated, PN should also be initialized to Value '1'.

In case of wireless signals, being boundless, 802.11 MPDUs can be easily sniffed. Once the MPDU is sniffed, the MAC and CCMP headers can be obtained because these are traversing in plaintext and at a fixed location within the MPDU. Subsequently, a hacker could verify pre-computed nonce by extracting A2 address and priority fields from the MAC header. Now, to reconstruct the nonce, only the knowledge of PN field is required. The CCMP header is already sniffed and is available in plaintext. As PN field is monotonically increasing for every MPDU and is part of CCMP header, its current and future value can be easily identified. Accordingly, nonce can be calculated in advance and verified as shown in Fig. 3.1[10].

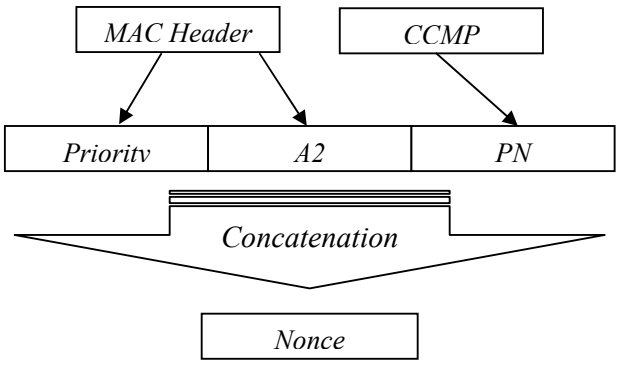


Fig. 3.1. Nonce Reconstruction Scheme

3.3 Reconstruction of Initial Counter

AES block cipher in counter mode is used for encrypting the payload and message integrity code (MIC). In the first step, counter value (Ctr_i) is encrypted to produce output in the form of key-stream blocks (S_i) as shown in Eq. 3-2:

$$S_i = e_K(Ctr_i) \tag{3-2}$$

where,

$Ctr_i = (Ctr1 + i - 1) \bmod 2^n \; (1 \le i \le b)$

Ctr_i = counter block value of the i^{th} iteration

e_K = Encryption with 128 bit AES Key(k)

n = number of bits in a block.

b = number of key stream blocks to be exclusive-OR with Plaintext block.

These keystream blocks (S_i) are X-ored with the plaintext (P) to get ciphertext (C) as shown in Eq. 3-3:

$$C = P \oplus (S_1 \,||\, \ldots\ldots \,||\, S_b) \tag{3-3}$$

On reception, the cipher-text is X-ored with the already computed key-stream blocks to obtain the plain-text as shown in Eq. 3-4:

$$P = C \oplus (S_1 \,||\, \ldots\ldots \,||\, S_b) \tag{3-4}$$

The structure of counter block values is based upon length of length of Payload, nonce and flag field.

Counter blocks (Ctr_i) having counter index ‘i’ are formatted as illustrated in Table 3.1 [6]. There are 8 bits in the flags field. First 2 bits are kept reserved for future utility. Each of next 3 bits have been assigned a ‘0’ value. The length (in octets) of length (in octets) of payload (binary representation) , q, are encoded in the last three bits of the flag field and calculated as $[q-1]_3$.

TABLE 3.1.

Formatting of Counter Blocks

Octet number	*0*	*1…..15-q*	*16-q…….15*
Contents	*Flags*	*Nonce*	$[i]_{8q}$

Next field in the counter block is the nonce field. Refer to Section 3.2 for composition and reconstruction of nonce field. Each input string (nonce & Payload) has a bit length of size of an exact multiple of 8 bits [35]. The lengths (in octets) of nonce and payload are represented by integer values of ‘n’ and ‘p’ respectively. The length (in octets) of Payload is represented by ‘Q’. ‘Q’ appears as an octet string in the first block of data. The length (in octets) of ‘Q’ is represented by ‘q’. Therefore, ‘Q’ is equivalent to $[p]_{8q.}$

After having discussed the reconstruction of nonce and the status of flag field as a constant known value, to obtain counter block value, only the length of the payload is needed. The maximum payload length of IEEE 802.11 MPDUs is 2312 bytes.

For MSDUs having larger data than 2296 bytes, IEEE 802.11 recommends fragmentation of MSDU into MPDUs. Consequently, the requirement of fragmentation exists in almost all MSDUs because of the large length of the payload of the MPDU due to the presence of TCP Header, IP Header and SNAP Header. Accordingly, the first packet is usually of maximum size. Therefore, the length of payload length is known in advance. This leads to the successful pre-computation of the initial counter value and the same is true for the pre-computation of successive counter values. The computation of payload is presented in Eq. 3-5 [6]:

$$
\begin{aligned}
&P = 2296 \ \ octets \\
&\quad \text{if} \quad q = 2, then \\
&Q = [p]8_q \\
&\quad = [2296 \ ]_{8\times 2} \\
&Q = 000010000 \quad 11111000
\end{aligned}
\tag{3-5}
$$

Where, p = octet length of Payload.

q = The octet length of the binary representation of octet length of payload.

Q = A bit string representation of the octet length of P.

Extraction of fields to pre-compute the initial counter value is illustrated in Fig. 3.2 [10]. A2, priority field, PN, and length of length of payload can be acquired by unauthorized user. Accordingly, initial counter can be reproduced irrespective of undergoing the successful authentication process.

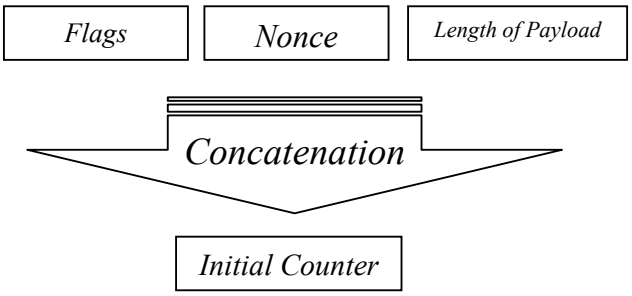


Fig. .3.2. Reconstruction of Initial Counter

3.4 TMTO Precomputation Attack

Sections 3.2 and 3.3 described the pre-computation of A2 Address field of MAC header, priority field, Packet Number field and length of length of payload field. The initial counter can be computed in advance by using these values. This acquired counter value makes it possible for hacker to launch TMTO attack [10].

TMTO attack is a shortcut over brute force / exhaustive key search attack [36]. TMTO is a trade-off of increased storage requirements against decreased computational power. In TMTO attack, a hacker forms a large table of pre-computed data before launching an actual attack on the cipher. Then during the actual phase, using the pre-computed data tables, an offensive is launched on many different cryptographic keys. A very important characteristic of TMTO attack is that it needs not have a priori knowledge of the plain-text during the preparation of pre-computed data tables. Furthermore, TMTO also takes hint from the error-correcting codes techniques and is effective even if there is uncertainty in the plain-text [37]. The significance of TMTO attack can be judged from the important role it has played in the break of A5/1 cryptographic algorithm [38].

At times, a cryptographic system is considered to be broken even if partial information about the keys is obtained by hacker [34]. The significance of pre-computation attacks is much more pronounced in these cases. The more data is available, the more are the chances of successful pre-computation attack. So, creating a suitable attack scenario, in which ample amount of data is available, is extremely important. In 802.11, the counter for CCMP increments monotonically during the same session. Also noteworthy is the fact that there is no limit on the maximum number of MPDUs per session. Hence, enough data is available to launch successful TMTO pre-computation attack.

[39] maintains that if counter update is predictable then counter mode is vulnerable to TMTO pre-computation attack. Previous sections of this chapter described that initial counter value along with its increments are predictable, hence, makes CCMP vulnerable to TMTO attack. The effective key size for a TMTO attack is 2n/3 [36]. Where 'n' is the size of the key. In 802.11, key size of AES counter mode is 128 bits. As per [36], after TMTO attack the effective key size will be 85 bits as shown in Eq. 3-6 [10]:

$$\text{Effective key size} = 2n/3 \text{ bits} \tag{3-6}$$

where, n = 128 bits

$$\text{Effective key size} = (2 \times 128)/3 \text{ bits}$$

$$\text{Effective key size} \approx 85 \text{ bits}$$

To provide adequate security to symmetric ciphers, 75 bits key length is recommended [40]. For the sake of maintaining secrecy at least up to next

20 years, [40] also recommends adding 14 bits. Additional key bits are required to be added for 13 years (1996 to 2013). Furthermore, additional key bits for 5 more years for the validity of [2] are also required. Accordingly, as per Moore's laws [41], recommended current strength for the cipher becomes 101 bits. In the context of TMTO, the effective key size of CCMP (AES counter mode) is 85 bits, whereas it should be at least 101 bits to provide adequate security. The fact that effective key size of CCMP is far less than the recommended minimal key length exposes the vulnerability of 802.11 WLAN to TMTO pre-computation attack.

Moreover, to have effective defense against TMTO attack, [34] suggests adoption of at least one of the following remedial measures:

i Initial counter should have at least 64 bits of unpredictable value, which is taken as part of the AES Counter Mode key, OR

ii Initial Counter should have uniformly distributed component, OR

iii The length of the AES key should be more than 128 bits.

During this research [10], it is observed that none of above mentioned remedial measures has been taken in the 802.11 standard. Consequently, 802.11 WLANs are exposed to TMTO attack [10]. Solely relying on the cryptographic algorithm (AES) and ignoring appropriate implementation of modes of operation/ associated protocols leave hidden weak links in the security framework. Exploitation of these weak links may subsequently break entire cryptographic framework as observed in the case of WEP.

3.5 Conclusion

After the collapse of the WEP, CCMP protocol has been employed to provide data confidentiality, message integrity and authentication services to IEEE WLANs. AES block cipher in counter mode encrypts the data. The counter value is formed by Packet Number field, Address A2 field, priority field and length of payload length field. This research exposes the vulnerability of CCMP protocol to TMTO pre-computation attack [10]. Sequel to this work, a robust security framework is proposed to guard IEEE WLANs against possible TMTO pre-computation attack [11]. The same is described in the Chapter 4.

Chapter 4 Proposed Security Mechanism to Defend TMTO Attack

4.1 Introduction

It has been shown in our earlier work [10], described in Chapter 3, that CCMP is vulnerable to Time Memory Trade off (TMTO) attack. To overcome the said vulnerability, we continued the work and proposed a robust per-packet security mechanism for future Wireless LAN implementations [11]. The architecture of Per-Packet security mechanism involves Per-Packet Authentication and Secret Nonce. The proposed Per-Packet Authentication protocol is a continuous challenge-response process operating throughout the session. The Per-Packet challenge response mechanism secures the connection against Denial of service attack by immediately discarding the packet if Per-Packet Authentication fails. It is suggested to derive the Nonce from the session key and keep it secret. Since the nonce is unique and secret, it provides freshness and unpredictability. Freshness provides protection against replay attacks and unpredictability of Nonce prevents pre-computation attack.

Rest of the chapter explains proposed per packet authentication mechanism, shows how proposed security mechanism obviates the requirement of MIC and discusses the advantages of proposed security mechanism.

4.2 Proposed Per-Packet Authentication Mechanism

Pairwise key hierarchy utilizes pseudorandom functions (PRF) to derive session-specific keys from a pairwise master key (PMK) [6]. The PMK is available as a result of successful IEEE 802.1X exchange, pre-shared key (PSK) or PMK cached via some other mechanism. The PMK is 256 bits. The pairwise key hierarchy takes the PMK and generates a pairwise transient key (PTK). The PTK further generates temporal key (TK) . This temporal key is the shared encryption key used in the AES counter mode to encrypt the Data and MIC. We propose that the initial counter value should be derived from the temporal key using the PRF-128. The PRF-128 is a pseudo-random function which outputs 128 bits and is defined in subclause 8.5.1.1 of [6]. The proposed method for generation of initial counter value is illustrated in Fig. 4.1 [11].

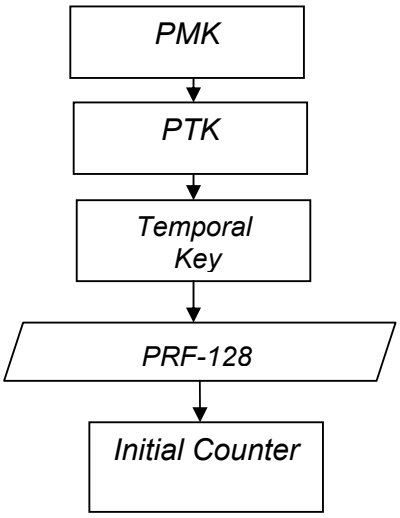


Fig. 4.1. Derivation of Initial Counter from Temporal Key

Initial counter value along with temporal key will be used to encrypt the first packet from the authenticator to supplicant. The authenticator will encrypt Data, MIC and N_0 (nonce value) as shown in Eq. 4-1 [11]. The N_0 will be a 48 bit value generated by using the PRFs.

$$E_{TK}(\mathrm{N}_0 \,||\, \mathrm{Data} \,||\, \mathrm{MIC}) \tag{4-1}$$

Where,

$E_{TK}(A)$ = Encryption of A with TK
No = Nonce
MIC = Message integrity Code

The supplicant will decrypt the packet using the temporal key and initial counter value as shown in Eq. 4-2 [11]. If the temporal key and the initial counter value are correct, then supplicant will obtain the correct N_0.

$$D_{TK}(e_{TK}(No \,||\, Data \,||\, MIC) = No \,||\, Data \,||\, MIC \tag{4-2}$$

Where, $\mathbf{D}_{\mathrm{TK}}$ (B) = Decryption of B with TK
N_0 = Nonce
MIC = Message integrity Code

The supplicant will encrypt the N_0 , Data and MIC and send it to authenticator. Upon decryption, if the authenticator gets the correct N_0 value, then this means that the supplicant is an authorized entity. The authenticator will generate N_1 (nonce value) and then send it with the second packet. The supplicant will do the same exercise as it did with the first packet. This challenge response mechanism will continue through out the session providing per packet authentication as illustrated in Fig. 4.2 [11].

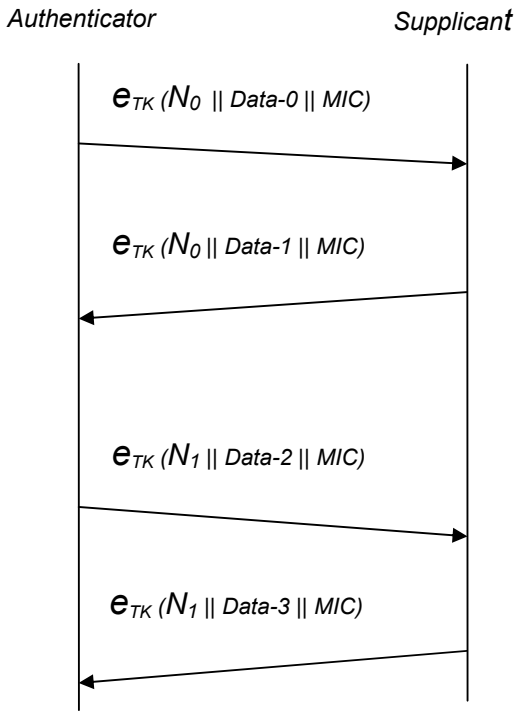


Fig. 4.2. Per-Packet Authentication Mechanism

4.3 Per-Packet Authentication Mechanism without MIC

We analyzed the per packet authentication protocol by eliminating the CBC-MAC part of the CCMP [11]. Although this elimination apparently appears harmful as there will be no MIC and hence the message integrity part is missing. But upon closely observing the effect of each input to the CBC –MAC we found out that most of the fields are not at all required in our protocol and only three fields that are required consume so less bandwidth that sending their MIC is rather uneconomical and instead we can append these fields to the payload. These fields are 'More fragment field' (1bit), length of payload field (11 bits), length of length of payload field (4 bits) and retry bit (1 bit). In total 2 octets of these fields are to be appended. Whereas, we are saving 6 Octets of bandwidth per MPDU by not

sending MIC (8 Octets). Upon evaluating the protocol against MAC spoofing, precomputation attack, replay attack and Denial of Service attack, we found out that our protocol works well in protecting the communication against these attacks. We found out that eliminating MIC process saves the computational resources and the bandwidth. One of the disadvantages with the CCMP protocol is the requirement of collection of entire ciphertext before the verification-decryption can begin. Due to this reason the CCMP processes the online data with considerable latency. In proposed per packet authentication mechanism without MIC even the online processing would be possible with good quality of service. The per packet authentication protocol without MIC is illustrated in Fig. 4.3 [11].

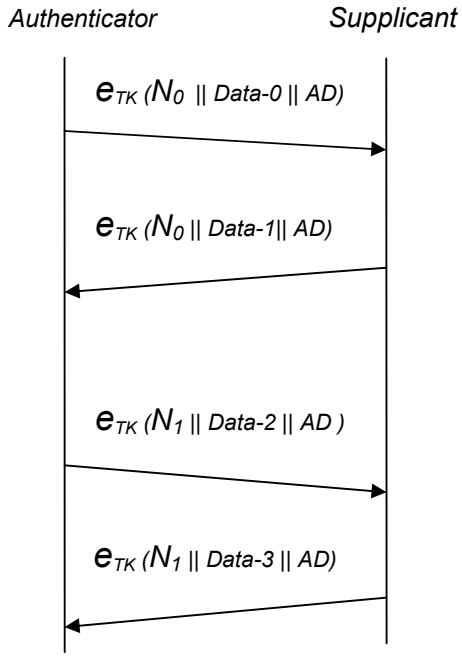


Fig. 4.3. Per-Packet Authentication without MIC

4.4 Robustness Against Attacks

In this section strength of proposed per packet authentication mechanism against MAC spoofing, replay attack, precomputation attack and denial of service attack is discussed [11]. Performance of proposed protocol in different scenarios of attacks is established.

4.4.1 MAC Spoofing

We assume that there is a rogue station. We also assume that the rogue station is capable to successfully sniff and parse the wireless packet and get the MAC address of the legitimate station. We also assume that the rogue station has generated a malicious packet and attached the MAC header of the legitimate station. Now upon reception, since there is no CBC-MAC process, the AP directly decrypts the received packet. The packet is decrypted with the transient key. The packet contains the nonce, N_0, which the AP sent to the STA. The AP is maintaining the table which attaches each nonce to the corresponding STA MAC Address. Upon examining the decrypted packet, if the AP does not get the nonce, $N_{0,}$ which it sent earlier to the STA having the MAC address in the MPDU then it discards the packet. Hence, the AP prevents the MAC spoofing attack instantly.

4.4.2 Replay Attack

In case of replay attack by a rogue station, the decryption at the reception side by the AP would present the older nonce and not the current one. Since the nonce is changing for every packet , the AP would expect the current

nonce and on receiving the invalid nonce it will discard the packet. Therefore, the continuous challenge response mechanism successfully prevents the replay attack. We have proposed to add the retry bit in the Associated Data field which is concatenated with the payload to differentiate between legitimate retry packet and illegitimate replay packet.

4.4.3 Denial of Service Attack

Any attack to degrade the resources would be in the form of either sending excessive invalid packets or try to have illegitimate access to the services. In the case of CBC-MAC processing, the data is first collected, MIC is generated and verified and then the decryption of ciphertext is carried out. If the attacker is sending packets by changing the packet number to future packet numbers, the CBC-MAC would carry out the whole MIC verification process but in our protocol none of the procedures are carried out, instead the decryption is done and if the nonce is not the current one it will discard the packet. Although, we have to decrypt every packet to check the currency of the nonce, but on the other hand we are saving computational power by not calculating the MIC for every packet and saving bandwidth by not concatenating 48 bits packet number in plaintext with every packet as in the case of CCMP.

4.4.4 Pre-Computation Attack

As we have shown in the Chapter 3 that time memory trade off attack is possible and pre-computation by a potential adversary can reduce the key strength of the CCMP encryption key [10]. This precomputation attack is possible since the packet number was traversing in the air in plaintext and

consequently the counter value is predictable by the adversary [10]. In the proposed mechanism [11], nonce is transmitted in encrypted form and none of the parts of counter value are available in plaintext to the unauthorized station. Accordingly, precomputation by the adversary becomes impossible.

4.5 Per-Packet Authentication Mechanism – Benefits

Nonce in the existing CCMP provides freshness to every packet, but it is predictable. This predictability of nonce renders the protocol vulnerable to pre-computation attack. The proposed per packet security mechanism provides per-packet authentication mechanism using the secret nonce. It is shown that the nonce is derived from the session key and is kept secret. The same nonce is used as a challenge text from authenticator to supplicant. This per-packet authentication protocol is a continuous process, thus provides freshness and unpredictability beside per-packet challenge response mechanism. The freshness provides protection against replay attacks, the unpredictability of nonce prevents pre-computation attack. The per-packet challenge response mechanism, additionally, secures the connection against denial of service attack by immediately discarding the packet if Per-Packet Authentication fails. Comparison of existing and proposed security mechanism is given in Table-4.1 [11]. Besides prevention of attacks, the per packet authentication mechanism without MIC saves computational resources in calculating the MIC at receiving side, reduces latency to half, reduces 64 bits overhead of CCMP header and 64 bits overhead of MIC field per MPDU and makes online processing of data possible.

TABLE 4.1.

Comparison of Security Mechanisms

	Pre-Computation	*DoS*	*Replay*	*MAC Spoofing*	*Latency*
Existing Security Mechanism	*Yes*	*Yes*	*No*	*No*	*High*
Proposed Per-Packet Authentication Mechanism	*No*	*No*	*No*	*No*	*Significant Reduction*

4.6 Conclusion

CCMP has been formally incorporated in IEEE 802.11- 2007 Standard. This research work [10], described in Chapter 3, has established that CCMP is vulnerable to TMTO attacks. To strengthen the weak links of CCMP, this chapter has presented proposed per packet authentication mechanism [11]. It has been observed that per packet authentication mechanism obviates the requirement of MIC, thus improves latency and computational overheads. The architecture of Per-Packet security mechanism involves Per-Packet Authentication and use of secret Nonce. The unpredictable nonce guards against pre-computation and replay attack. The proposed Per-Packet Authentication protocol is a continuous challenge-response process operating throughout the session and secures the connection against pre-computation attack and Denial of Service attack.

Chapter 5 Security Framework for Wireless Mesh Networks

5.1 Introduction

This chapter presents the proposed security framework for WMN [12]. Chapter begins with the background information, then the problem is formulated and robustness of proposed framework against well-known attacks is discussed [12]. Finally, comparative results in terms of security robustness index and simulation results are presented.

5.2 Background

WMN are evolving as a promising technology for providing scalable, efficient and robust wireless broadband access. However, the true potential of WMN cannot be utilized without adequately addressing the security issues. Wireless networks are insecure because of the unrestricted nature of the transmission medium. In multi-hop wireless networks, which include WMN, mobile ad hoc networks (MANET) and wireless sensor networks (WSN), the inherent insecurity is further complemented by the traversal of data through non-trusted intermediate hops. Various security solutions [15-23] have been proposed for MANET and WSN. Although both MANET and WSN are multi-hop in nature similar to WMN, the design goals of the security solution for these networks are significantly different from WMN. For example, limited resources (battery power, computational, bandwidth and storage resources) is a major design constraint in MANET and WSN and a number of security solutions have been proposed keeping in view the trade off between robust security services and the overhead introduced.

However, WMN router nodes are equipped with reasonable battery power, memory and computational resources while the bandwidth could be increased using multiple radios per node, each operating on an orthogonal channel. The adequate level of resources can support a more robust security solution although a security mechanism with low communication and computation overheads is appreciated.

Similarly, keeping in view the high mobility, lack of infrastructure and lack of connectivity with the Internet, the authentication and authorization mechanisms for MANET and WSN focus on discovery and distribution of a Certification Authority, which is needed for providing authentication [15, 16]. Given the high mobility and the possibility of physical node compromise, it is impractical to have a single certification authority placed at a fixed location. Consequently extra communication and computation overhead is induced because of distributing the certification authority among multiple nodes. However, in case of WMN, router nodes are generally static while clients exhibit limited mobility. Therefore, the extra overhead induced by discovery and distribution of certification authority can be eliminated. In addition, the fact that a WMN is connected to the Internet via gateway nodes can be leveraged to develop a more efficient solution. Furthermore, WSN and MANET are usually deployed under one administrative domain, which may not be the case with WMN. For example, in case of a community deployment, almost every node is owned by a different user. This makes it inappropriate to distribute the certification authority because such a scheme will be prone to service provisioning DoS attack based on the selfish nature of the WMN nodes.

The difference between design constraints of WMN from other multi-hop wireless networks (MANET and WSN) leads us to propose a security

framework for WMN with explicit consideration of the underlying network characteristics. The proposed framework provides the services of authentication, trust establishment, link level data confidentiality and data integrity at MAC layer. The security is further strengthened by providing end-to-end data confidentiality and data integrity at the network layer using a novel piggybacked challenge-response protocol. The challenge response mechanism also provides for per packet authentication. The proposed security framework explicitly addresses the security issues originating from the existence of the selfish and the malicious nodes in the network [12].

5.3 Problem Formulation

5.3.1 Network Architecture

Fig. 5.1. illustrates the architecture of a typical WMN, which shall be used as the network model for this work [12]. The architecture is similar to the one used in [42] where the network consists of WMN router nodes which act as access points for the clients as well as forward the traffic for neighboring nodes. The bandwidth capacity of the network could be increased by equipping each node (WMN router) with multiple interfaces (NICs) operating on different radio frequencies (channels). However, it should be noted that our security framework is equally applicable to WMN that use single and multiple radio interfaces and/or single and multiple channels. Few of the nodes such as G1, G2 in Fig. 5.1 are directly connected to the Internet and are referred to as wired gateways.

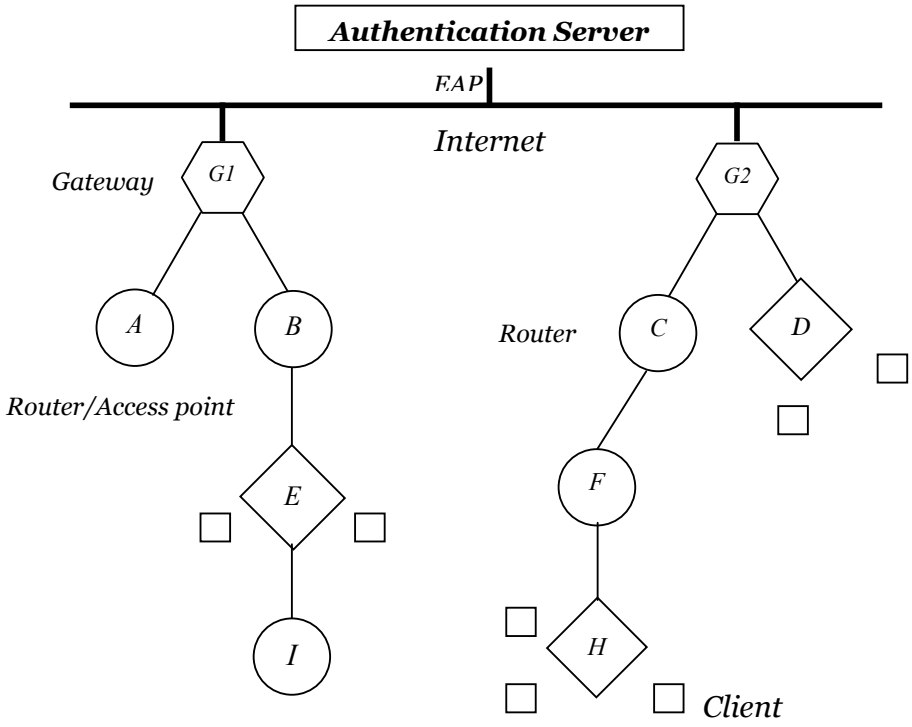


Fig.5.1 Wireless Mesh Network Architecture

The router nodes are static, resulting in a more or less fixed topology with infrequent changes due to departing or joining nodes. Some of the routers also serve as access points (referred to as router/access points, e.g.: E, H in Fig. 5.1) and allow client nodes to connect to the mesh network. The client nodes may have limited mobility. The nodes in the WMN may not belong to the same administrative domain, specifically in case of community deployment. Unlike [42] where the authors assume that data only flows to and from Internet, we also consider the possibility of data communication between two nodes within the WMN. Such type of traffic can originate from applications like network gaming and video on demand.

5.3.2 Assumptions

It is assumed that the wired network is secure. The proposed framework only addresses the security issues within the WMN [12]. It is also assumed that the node can differentiate between traffic to and from Internet (going through the wired gateway) and the traffic that is local to the WMN (i.e. two WMN nodes communicating with each other). Note that, this can easily be accomplished using IP based network domain information.

The keys required for EAP communication are pre-configured in all the nodes. This assumption is basis of 802.1X [7] which requires secure EAP communication for exchange of messages.

The term ’node’ is used for WMN routers and router/access points and the term ’client’ to refer to the end user devices, throughout the rest of the Chapter.

5.3.3 Security Requirements for Wireless Mesh Networks

ITU-T Recommendation X.800 [43] -Security Architecture for OSI - defines the required security services for data communication networks. The security services have been broadly categorized into five groups, namely, authentication, access control or authorization, confidentiality, integrity and non-repudiation. WMN are subject to same security requirements although the means to accomplish these requirements may vary significantly as compared to traditional wired networks. At the time of joining the network, a WMN node needs to show its credentials and get authenticated (Initial Trust Establishment). This trust establishment needs to be mutual to ensure

that neither does a legitimate node join a network of malicious nodes nor does a malicious node join a trusted network. Whenever two nodes need to communicate, both nodes need to authenticate each other to eliminate the possibility of compromised nodes. WMN are inherently insecure like any other wireless network because of the broadcast nature of wireless communication. Hence, it is imperative that data confidentiality is provided for. The problem of insecurity worsens when the data may traverse multiple intermediate insecure WMN nodes. Multi-hop communication necessitates strong data integrity.

The link level (between two adjacent communicating nodes) authentication, confidentiality and integrity can be realized at the MAC layer of wireless networks, while end-to-end data confidentiality and integrity can only be realized at the network or higher layers as recommended by [43]. The recommendation is based on the following fact. If the end-to-end data confidentiality is realized at the MAC layer, then the IP header must be encrypted by the source node (recall that the entire IP datagram including the header is the payload for the MAC frame) and should be forwarded in its encrypted form all the way to the destination node. This would prevent intermediate nodes from routing the packet, since they would not be able to access the IP address of the destination header, which is carried in the IP header. Note that, end-to-end data confidentiality and integrity are mandatory services for WMN to counteract the passive eavesdropping by the intermediate nodes, which can readily copy the data for off-line analysis before forwarding it to the next hop. The passive eavesdropping by intermediate nodes can be a serious security compromise, especially given the high likelihood that all the nodes in a WMN are not owned and managed by a single administrator. The problem is further

worsened due to the possibility of compromised nodes. Therefore, the proposed framework provides link level data confidentiality and data integrity at the MAC layer, while end to-end data confidentiality and data integrity are realized at network layer of the protocol stack.

The above mentioned security services may not eliminate the possibility of compromised node or counteract selfish behavior. However, one of the design goals for an effective security framework would be to reduce the dependency of security services on non-trusted intermediate hops of a data flow. Furthermore, the computation and communication overheads introduced by the security services should be relatively low in an effort to reduce the latency induced at each hop. Note that, link level data confidentiality and integrity are necessary to counteract the MAC layer attacks like replay attack and MAC spoofing attacks, while end-to-end confidentiality and integrity are necessary to mitigate the effect of malicious and selfish behavior of intermediate hops.

5.4 Security Framework

5.4.1 Node Authentication -Initial Trust Establishment

In our framework, node authentication, initial trust establishment and cryptographic key distribution are realized by using 802.1X over Extensible Authentication Protocol (EAP) and the authentication server (e.g. RADIUS or DIAMETER), which is placed on the wired Internet [7, 24, 25, 44]. Although 802.1X has been used for key distribution and trust establishment in many other proposals including 802.11i, the actual key distribution mechanism used in our framework needs further elaboration. The protocol

used for 802.1X authentication messages is EAP, which uses pre-configured keys to secure the EAP messages.

As a general procedure for initial trust establishment and the key distribution, the nodes that are already connected to the WMN (initially only the wired gateways) broadcast EAP request messages at regular intervals. Any unauthenticated node (call it joining node) that is within the transmission range of the broadcasting node (call it connected node) can request to join the WMN by replying to the EAP request message. The joining node presents its credentials[1] for authentication from the authentication server. The connected node forwards the credentials of the joining node along with its own credentials to the RADIUS server through a wired gateway.

The RADIUS server mutually authenticates the joining node and the gateway as well as the joining node and the connected node. Upon successful authentication, the server issues a unique Pair-wise Master Key (PMK) for the joining node and the connected node. Another unique PMK is issued for the joining node and the wired gateway through which the request was forwarded to the server. Therefore, all nodes share one PMK with the parent node and another PMK with the gateway through which it is connected to the Internet. The authentication is followed by the connection. The destination node will send the EAP request message to the originating node (not a broadcast). The two nodes will then go through the mutual authentication using the authentication server.

(footnote) 1: The credentials can be digitally signed message or the challenge text. They are based on the specification of the authentication server. Further details of the credentials used are beyond the scope of this research.

Upon successful authentication, the server will issue a unique PMK for communication between the two nodes. These authentication messages are also transmitted using EAP as the underlying protocol.

Note that, a non-connected node could be within the transmission range of multiple connected nodes, in which case it will receive multiple EAP request messages from the connected nodes (one from each connected node). The non-connected node will only respond to the first message that it will receive, ignoring the subsequent messages. Similarly, a connected node could receive the EAP request message from its neighboring connected nodes but it will ignore the messages unless the node requires changing its connectivity because of the failure of its parent node. The connected nodes will continue to broadcast the EAP request message at regular interval even after the initial trust establishment. This is necessary to take into account the possibility of new nodes joining, existing nodes departing and node failures. Note that, a malicious node cannot broadcast the EAP request message as it lacks the pre-configured keying material. Further, keys expire after a fixed interval, thus requiring nodes to refresh the keys before they expire. In presence of an intrusion detection mechanism that can detect the malicious and selfish behavior of the nodes, the authentication server can be informed of misbehaving nodes, which can then be denied key renewal. The child nodes connected to the malicious node can use the recovery mechanism as in [42] to connect to a different parent node.

Consider Fig. 5.1. as an example to elaborate the procedure. The wired gateway nodes (G1 and G2) are the first to be authenticated. Since gateways are directly connected to the Internet, they do not need any intermediate node within the WMN to forward their request to the

authentication server. Once authenticated, these nodes (G1 and G2) will broadcast the EAP request messages periodically. Any unauthenticated node within the transmission range of the gateways can reply to the EAP request message, sending its credentials for authentication (nodes A and B in case of gateway G1 in Fig. 5.1).

The gateway will act as authenticator and forward the authentication request to the RADIUS server along with its own credentials. The server will first authenticate the gateway to ensure that a compromised node is not maliciously acting as a fake gateway. It will then authenticate the nodes A and B. This will result in strong joint authentication between the gateway and the requesting nodes. On successful authentication, the server will issue a pair-wise master key (PMK) for each node-gateway pair (in this case, node-gateway pair is same as node-parent pair). Network layer connectivity will then be established between the gateway and the authenticated nodes as per the routing protocol in use.

The successfully authenticated nodes (A and B) can then broadcast the EAP request message so that any unauthenticated nodes within the transmission range can request for authentication and connectivity by replying the EAP request message. In our example, node E (joining node) will reply to the EAP request message from node B (connected node). The intermediate node (node B) only forwards the request to the gateway G1. The gateway then repeats the same process for node E and after successful authentication, the RADIUS server provides the PMK for the node-gateway pair (node E and node G1) as well as another PMK for the joining node-connected node pair (Node E and node B). This will be followed by network layer connectivity between the node E and node B. After successful mutual

authentication and distribution of keys across the network, the keys can be used to provide data confidentiality and data integrity through cryptographic components as explained in the subsequent sections.

5.4.2 Data Confidentiality -Piggyback Challenge-Response Protocol

This section, first describes the proposed piggyback challenge-response protocol that is being used for data confidentiality between two communicating nodes (at the link layer) [12]. Subsequently, achievement of end-to-end data confidentiality using same protocol is described.

5.4.2.1 Link Based Data Confidentiality. The proposed piggyback challenge-response protocol relies on symmetric key block cipher having block size of 128 bits or more, such as Advanced Encryption Standard (AES), in Counter Mode for providing data confidentiality [45]. The message is divided into blocks of 128 bits. AES in counter mode functions by encrypting the unique counter value using the encryption key and the resulting block is X-ored with the message block to produce the cipher text block (see [45] for details). To understand the generation of the encryption key and the initial counter block used in AES encryption algorithm, we first elaborate on the key generation mechanism used in IEEE 802.11i [6]. The CCMP proposed in IEEE 802.11i uses PMK (Section 5.3.1 described how the PMK is established) as the seed for the pseudo-random function (PRF) to generate a pairwise transient key (PTK) through a 4-way handshake. PTK is then used to generate a temporal key (TK), which is the shared encryption key that is used in 'AES counter mode with CBC-MAC (CCM)' [26] to encrypt the data. However, this mechanism employed in IEEE 802.11i is vulnerable to pre-computation attacks (discussed in Section 5.4.4), mainly

because the same encryption key is used for all the messages within one session and also because initial counter value is predictable.

Our proposed protocol [12] uses a similar procedure to generate the temporal key (TK), which is then used as seed for the pseudo-random function (PRF-128) to generate the nonce N_0. The primary difference is that the nonce is subsequently used as the AES encryption key. The first nonce N_0, is transmitted with the first message. The key generation process is shown in Fig. 5.2 [12].

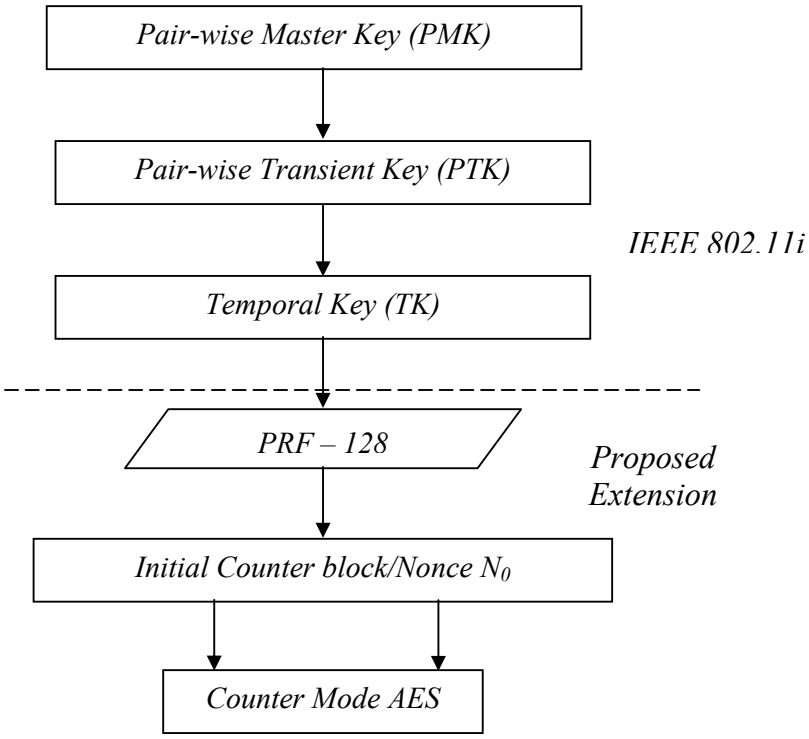


Fig. 5.2. Key Generation Mechanism

Suppose node A and node B share a PMK and wish to communicate. Assume that node A initiates the communication by sending an initial message to node B. Node A will use TK as the encryption key for this message. It will encrypt the first message along with the nonce N_0 (generated using Fig. 5.2) and the Meta Data using Eq. 5-1 [12].

$$E_{TK}(initial-counter) \oplus (N_o \,||\, Data \,||\, Meta\ Data) \qquad (5\text{-}1)$$

The field Meta Data and its use is explained in Section 5.3.2.2. The intended recipient (node B), upon receiving the message will also generate the nonce N_0 using the procedure shown in Fig. 5.2. It will decrypt the message using Eq. 5-2 [12], TK being the decryption key. After decryption, node B will compare its own generated nonce value with the received nonce. Since both nodes A and node B share the PMK, the N_0 generated should be same as the N_0 which was transmitted as part of message by node A. The nonce will act as challenge text to authenticate the source of the message.

$$D_{TK}(E_{TK}(initial-counter) \oplus (N_o \,||\, Data \,||\, MetaData)) = N_o \,||\, Data \,||\, Metadata \qquad (5\text{-}2)$$

Node B will then use N_0 as the encryption key for the reply, rather than the TK. PRF-128 will be used to generate a new nonce N_1, which will be concatenated with the data and Meta Data, encrypted using N_0 and transmitted back to Node A. Thus, a new nonce is generated iteratively for each subsequent message, which enhances the robustness of the security solution. Node A will employ the aforementioned decryption process to retrieve the message and authenticate (using the response nonce) the sender. The communication between nodes A and B is shown in Fig. 5.3. In general, the i-th message exchanged between nodes A and node B is encrypted using

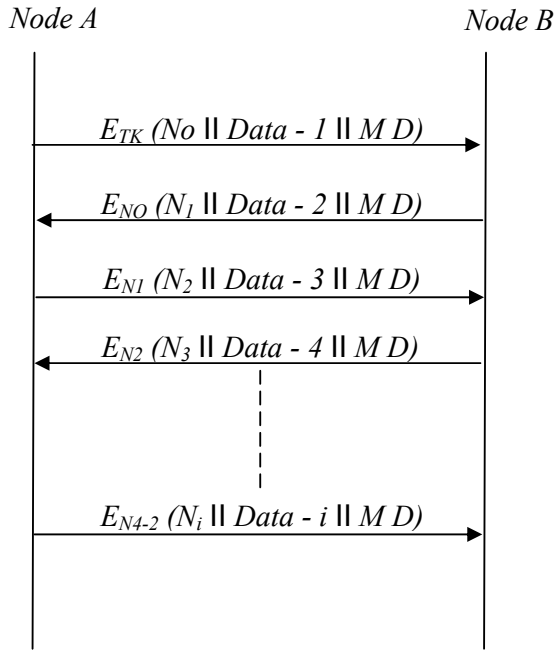


Fig.5.3. Per Frame Authentication Mechanism

Eq. 5-3 [12] and the corresponding decryption process uses Eq. 5-4 [12]. The exchange of the nonce results in a continuous challenge-response protocol, which provides data confidentiality as well as per packet authentication. The per packet authentication protects against MAC spoofing attacks as well as replay attacks as discussed in Section 5.4.

$$E_{Ni-2} \oplus (N_{i-1} \,||\, Data \,||\, Meta \,||\, Data) \qquad (5\text{-}3)$$

$$D_{Ni-2}(E_{Ni-2} \oplus (N_{i-1} || \, Data \,||\, Meta\ Data)) = N_{i-1} || \, Data \,||\, Metadata \qquad (5\text{-}4)$$

Note that, no other node in the network will be able to generate same nonce because the underlying PMK is unique and only known to the two communicating nodes. Further, even if a malicious node obtains the PMK (although it is practically hard to do so), it will still be unable to decrypt the

messages. This is because by obtaining the PMK, the malicious node can only infer TK (see Fig. 5.2. which is only used to encrypt the first message. All subsequent messages are encrypted using different nonces, which are generated sequentially using PRF128. Hence, to be able to decrypt an arbitrary message, the malicious node (in addition to the PMK) will need to know the sequence number of the packet in order to generate the appropriate nonce. (refer to Eq. 3 where the i-th packet is encrypted using i-2 nonce). Further, the subsequent nonce value is transmitted in an encrypted form within the message, which ensures its confidentiality.

5.4.2.2 End-to-End Data Confidentiality. The piggyback challenge-response protocol proposed in the previous section successfully provides the data confidentiality for a link between two adjacent nodes.

However, link based data confidentiality is not sufficient in case of multi-hop wireless mesh networks. This is because a node could be compromised at a later point in time, even if initial trust was established between this node and its neighbors. Since the message is decrypted and re-encrypted at every intermediate hop, a compromised intermediate node will get access to the plain text information, leading to a compromise in the data confidentiality. To solve this problem, our framework provides end-to-end data confidentiality at network layer. Note that, end-to-end data confidentiality does not eliminate the need for link level data confidentiality, which is necessary to protect the network from MAC layer attacks such as replay attacks and MAC spoofing attacks.

We assume that the client nodes will be connected to the trusted router/AP only. The assumption is valid without loss of generality, because

in a real-world scenario, the clients either connect to the router/AP nodes that are deployed by service provider or the router/AP within the user premises of the owner of the client devices. As an example, consider a community deployment scenario where a user has multiple wireless clients connected to a router/AP, which is owned by the user himself. Based on this assumption, the end-to-end data confidentiality can be realized by splitting the end-to-end semantics into two distinct parts. The first part ensures data confidentiality between the client and the router/AP with which the client is associated. The second part ensures the data confidentiality between the two routers/APs of the communicating clients (or one router/AP and one gateway in case the client is communicating with the wired Internet). Note that, intermediate hops between routers cannot be trusted. To better explain this, we make use of the example illustrated in Fig. 5.4. Let us suppose that the client c3 wants to communicate with the client c2. The data confidentiality for the communication between the client c3 and the router node/AP C (Similarly c2 and F) is the first part which consists of a single-hop. The data confidentiality for this part can easily be realized using protocol proposed in the previous section. Fig. 5.4 shows the format of the message exchanged between client c3 and router node/AP, C, where the data is encrypted at the MAC layer only.

The second part aims to provide data confidentiality on an end-to-end basis between nodes F and C. Here, data confidentiality will only be ensured if the intermediate hops (D and E along the end-to-end path between C and F receive the client data in encrypted form. This is achieved by encrypting the payload of the network layer packet between the two end router nodes

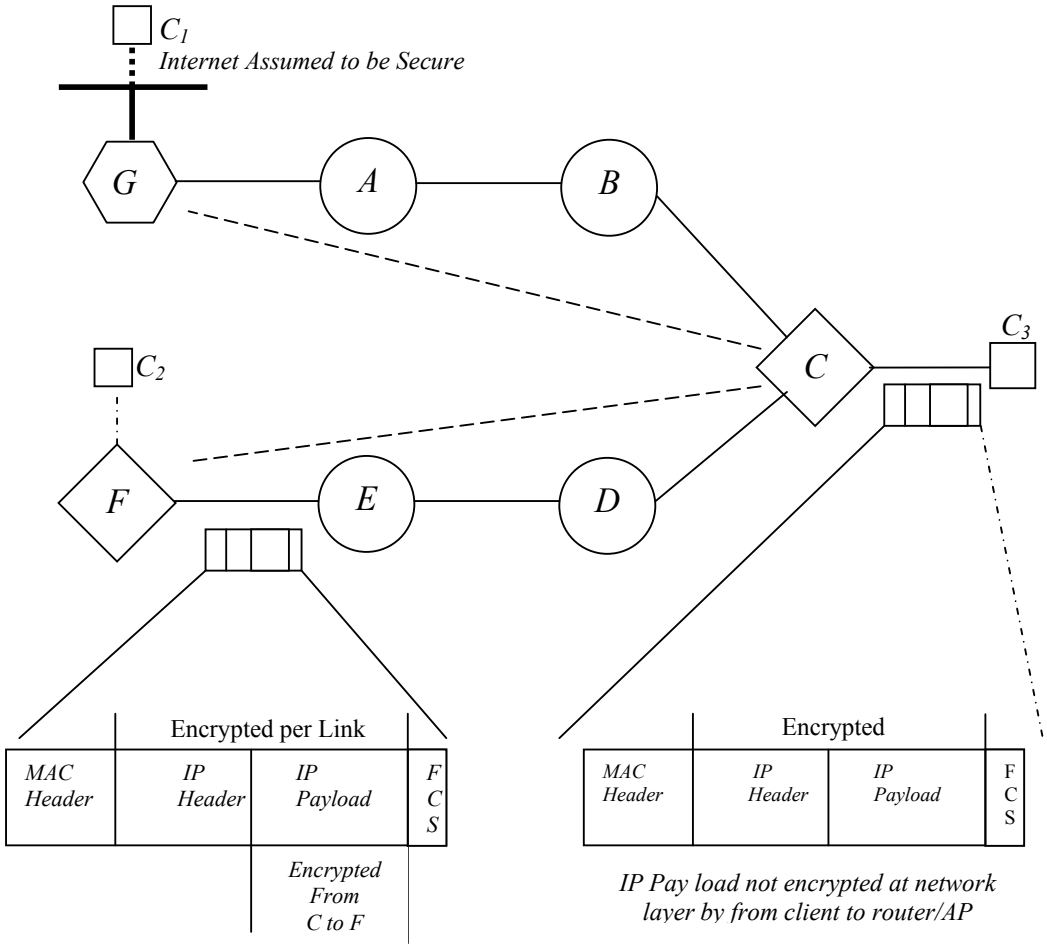


Fig. 5.4. End-to-end Data Confidentiality.

(C and F) using the protocol proposed in previous section. Note that if the second client is on the Internet (c1) then the network layer payload will be encrypted between the router node (C) and the wired gateway (G) as shown in the Fig. 5.4. Recall from Section 5.3.1 that every node shares a PMK with the gateway (or the node with which it needs to communicate), which is used for the aforementioned encryption. Note that, we only encrypt the payload, while the header is left un-encrypted. This is necessary to allow the intermediate hops to route the packet towards the destination based on the header information. The format of the message exchanged between nodes E

and node F is shown in Fig. 5.4. Note that, even though the network layer payload is encrypted, the MAC payload is also encrypted for every link (In case of Fig 5.4, the link EF) to prevent the possibility of MAC layer attacks.

Consequently, the end-to-end data confidentiality between two clients is ensured by the link level data confidentiality for the communication between the client and the access point AND the end-to-end data confidentiality between the two router nodes (Source access point and destination access point or the gateway). The un-encrypted data is only available at the trusted routers/APs to which the clients are connected to or at the gateway nodes from where the data enters or leaves the WMN.

5.4.2.3 Data Integrity. We use the Meta Data field in the MAC frame and the encrypted nonce to provide Data Integrity. Meta Data consists of two bytes and contains the length of the payload of the original message, the length of length of payload, more fragments bit and the retry bit as shown in Fig. 5.5.

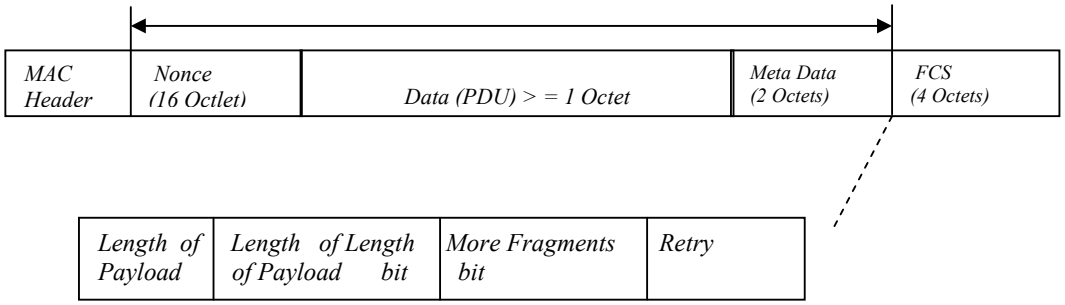


Fig.5.5. IEEE 802..11 Frame and Meta Data

Proposed framework replaces Cyclic Block Chaining (CBC) process. CCMP is using block ciphers in counter mode for data confidentiality. The integrity of data in CCMP is ensured by using CBC process. The Cyclic Block Chaining Process is computational intensive. The CCMP finally concatenates Message Integrity Code (MIC) with the message and this process introduces high overheads. In our frame work, introduction of per packet authentication obviates MIC concatenation [12].

To show the effectiveness of our mechanism, let us consider the possible ways in which the data integrity can be compromised. We then show how the proposed protocol prevents these compromises.

The message can be tampered in two possible ways. Firstly, a compromised intermediate node may alter a few bits or the entire message. Secondly, a compromised node may remove a complete block from the message (recall that counter mode AES divides message into blocks during encryption). In the first case, due to the use of a nonce in our scheme, when the receiving node compares the received nonce with the nonce generated by itself, a mismatch results, thus allowing it to detect the tampering of the message. In the second case, the message can be decrypted successfully and the nonce may match as well. However, when the receiving node compares the length of the payload with the value of length of payload extracted from the Meta Data, there is a mismatch. Since one or more blocks are missing from the message, the length of the received message will be less than the original payload length carried in the Meta Data. As a result, data integrity is maintained. The Meta Data also helps to counteract replay and MAC spoofing attacks as explained in earlier Section.

5.5 Robustness Against Attacks

In this section, it is demonstrated that how proposed security framework successfully protects the network from passive traffic analysis and prevents malicious nodes from launching a host of attacks including MAC spoofing attacks, replay attacks, pre-computation attacks and partial matching attacks [12]. Finally, robustness index for the security framework has been defined.

5.5.1 Robustness against Passive Eavesdropping

In multi-hop wireless networks, passive eavesdropping can be launched in two ways. Firstly, a malicious node within the transmission range of a sender can overhear all the transmission due to the broadcast nature of wireless transmission. Secondly, the intermediate nodes along the route from the sender to the receiver will readily have access to all the data that is relayed by them, which could potentially be copied into memory. Our proposed protocol employs link level data confidentiality, wherein each MAC frame is encrypted prior to transmission. Consequently, even if a malicious node eavesdrops and acquires a copy of the encrypted message, it cannot decrypt it as it does not have access to the particular nonce (nonce generated in our proposed piggyback challenge response protocol acting as encryption key) at that instance. It would need to perform a brute force analysis to retrieve the plain text. Further, in our proposed protocol, each packet is encrypted using a different randomly generated nonce, which renders brute force cryptanalysis to be ineffective. On a similar note, end-to-end data confidentiality at the network layer eliminates the possibility of eavesdropping by malicious intermediate nodes.

5.5.2 Robustness against MAC Spoofing Attack

A malicious node launches a MAC spoofing attack by changing the source MAC address in the transmitted frames to match the address of a node that has legitimate access to the WMN. The node can send a large number of such bogus frames to deplete the resources in the network (in particular bandwidth and energy). Let us assume that a malicious node is aware that two legitimate nodes A and B (B being a router/access point) are communicating. It can then spoof the MAC address of node A, construct a malicious frame and transmit the frame to node B. Our framework is resistant to such attacks as illustrated in the following discussion.

As shown in Fig. 5.6 [12], based on the proposed challenge-response protocol and per frame authentication mechanism, node B will decrypt the frame using a specific nonce Ni, which it expects node A to encrypt the frame with. Upon decryption of the frame, node B will compare the first 128 bits of retrieved text with the generated nonce Ni +1.

Recall that the next generated nonce is concatenated with the payload and is transmitted by the sender of the message in encrypted form. Since the malicious node did not have the knowledge of the nonce with which the frame was encrypted or the nonce which was sent in the encrypted text, the match will fail, therefore, the node B will discard the frame as malicious frame. The proposed per frame authentication mechanism will lead to a successful detection of malicious frames and MAC spoofing attacks will be thwarted spontaneously.

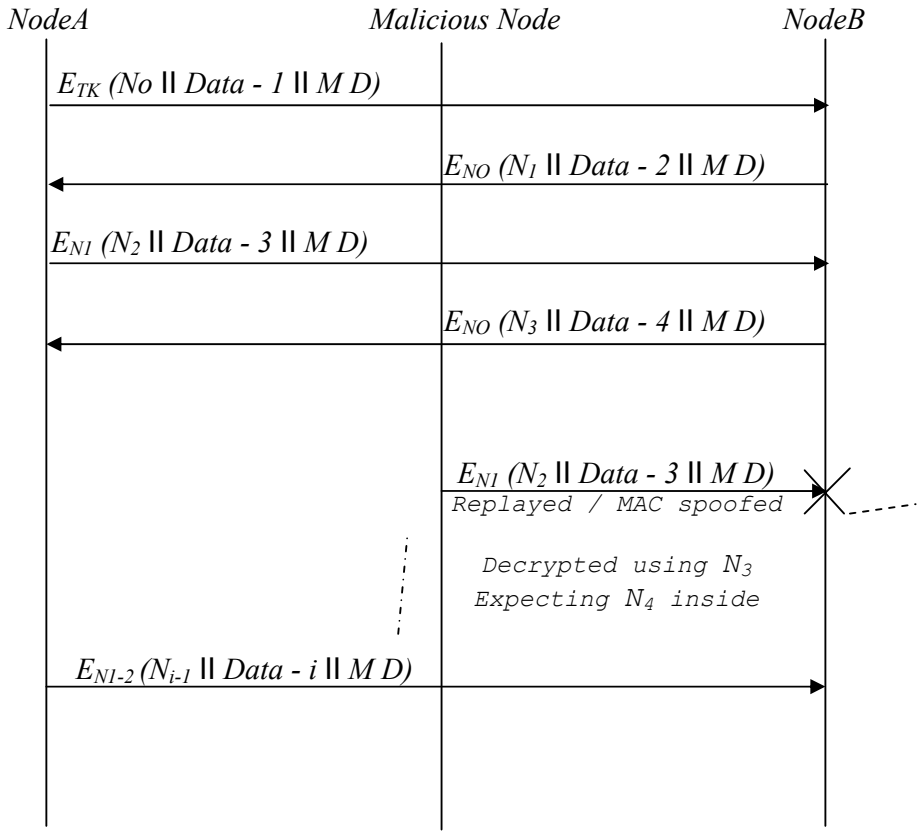


Fig.5.6. Robustness against MAC Spoofing and Replay Attacks

5.5.3 Robustness against Replay Attack

In a replay attack, the malicious node copies the legitimate message and transmits it at a later time to gain access to the resources. As mentioned in the previous section, in the proposed protocol, each frame is encrypted using a different nonce and the next generated nonce is transmitted in the encrypted form and concatenated with the frame payload. The freshness of the nonce for each frame and the retry and more fragment bits in the Meta Data ensure protection against replay attacks as explained in the following discussion.

Let us assume that a malicious node is in knowledge that two legitimate nodes A and B (B being a router/access point) are communicating. Suppose a malicious node overhears a legitimate frame transmitted by node A which is encrypted using nonce N_i and replays it later in time while the session between A and B is still active as shown in Fig. 5.6.

Let us further assume that node A and node B have exchanged t frames before node B receives the replayed frame. Two possibilities arise depending on the value of t. If t is greater than 0, node B will use the nonce N_{i+t} to decrypt the frame and expect the nonce N_{i+t+1} inside the frame which is not the case. Hence, node B will interpret the replayed frame to be malicious and discard it. If t is equal to 0, the malicious node can change the retry bit or the more fragments bit in the MAC header in the replayed frame.

Let us assume that the original frame delivery failed. In this case, node B is waiting for retransmission of the same frame or it is waiting for more fragments of the same frame. In both cases, node B will successfully decrypt the frame. However, the retry bit (or the more fragments bit, whichever is applicable) in the Meta Data will have a different value as compared to the retry bit value (more fragments bit) in the header. This will result in successful detection of the malicious frame by the receiving node.

5.5.4 Robustness against Pre-computation and Partial Matching Attacks

In a pre-computation attack or TMTO, the attacker computes a large amount of information and stores that information before launching the

attack. When the actual transmission starts, the attacker uses the pre-computed information to speed up the cryptanalysis process. The IEEE 802.11i [6] standard has been shown to be vulnerable to TMTO attacks [20]. However, in our proposed solution, the freshness of the encryption key (nonce) for every message makes the pre-computation attack ineffective. The proposed solution will thus successfully protect the network from these attacks.

5.6 Security Robustness Index

In this section, the degree of robustness is measured for proposed security framework and the CCMP employed in IEEE 802.11i. To measure the degree of robustness, this research introduces an index, named as 'Security Robustness Index' (SRI). SRI provides a novel method of quantitative presentation of effectiveness of security framework. SRI depicts the cryptographic strength of following underlying security mechanisms and cryptographic primitives:

i. Cryptographic Algorithm (CA)
ii. Key Length (KL)
iii. Modes of Operation of Block Ciphers (MO)
iv. Message Authentication Codes (MAC)
v. Node Authentication Protocols (NAP)
vi. Key Management Frame Work (KMF)

The relationship between SRI and said cryptographic primitives/mechanisms is defined in Eq. 5-5 as:

$$SRI = \frac{CA(KL + MO + MAC + NAP + KMF)}{(MLTK \times 10^3) + \sum_{1}^{n} P} \qquad (5\text{-}5)$$

Where,

MLTK = Mean Life Time of a Key

P = Number of Packets with Same Key

To measure the SRI of a security mechanism, different weights are assigned to the Cryptographic primitives/mechanisms corresponding to the level of security offered by them as per criteria at Table. 5.1. The weights assigned to Cryptographic Primitives/Mechanisms are shown in Table. 5.2.

TABLE 5.1.

Criteria for the Assignment of Weights

Categories of Break in Cryptographic Primitives	*Weights*
Total Break	*0*
Global Deduction	*2*
Instance Deduction	*5*
Information Deduction	*8*

The cryptographic algorithm and key length used in the proposed framework is same as of CCMP i.e. NIST approved AES algorithm [32] with 128 bits key length (key length may be increased to 192 and 256 bits).

.

TABLE 5.2.

Weights Assigned to Cryptographic Primitives/ Mechanisms

Cryptographic primitives and mechanisms	*Weights*			
	Min	*Max*	*CCMP*	*Proposed Security Framework*
CA	*0*	*10*	*10*	*10*
KL	*1*	*10*	*8*	*8*
MO	*1*	*10*	*5*	*8*
MAC	*1*	*10*	*8*	*8*
NAP	*1*	*10*	*5*	*10*
KMF	*1*	*10*	*5*	*10*

Therefore, CA and KL in both the schemes carry equal weights. The modes of operation of block cipher used in proposed security framework and CCMP is Counter Mode [2] but the difference is in its implementation

As a result of flawed implementation, CCMP is vulnerable to precomputation attack [10]. Therefore the weight of MO is kept significantly less in CCMP than in proposed security framework. For message integrity, CBC MAC [33] is being used by CCMP, whereas, proposed security mechanism transmits encrypted Meta Data (MD). Therefore, the weight of CBC MAC is kept same as of encrypted MD. Port based network access control protocol [7] is same in both the schemes, however due to the introduction of underlying piggyback authentication protocol for every packet [12], the weight of NAP is higher in proposed framework than CCMP. The proposed security framework offers unique key for every packet, therefore the KMF carries more weight in proposed security mechanism than in CCMP. Moreover, the mean life time of the key is extremely less in the proposed security mechanism as compared to

CCMP. While taking into account the discussed parameters in the context of security robustness, the SRI is calculated and the effect of increasing the amount of traffic on the SRI in both the schemes is observed. In the beginning, the SRI for transmission of only one packet is calculated. Then, the SRI for 10 packets is recorded and subsequently the amount of traffic in both the schemes is increased. The results are summarized in Fig. 5.7.

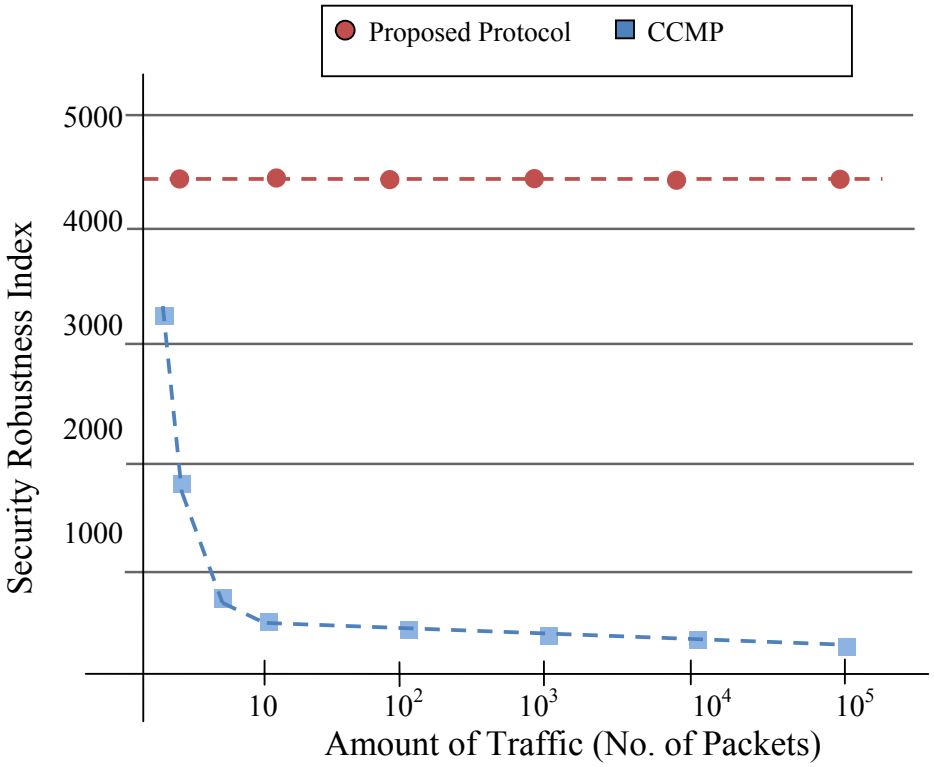


Fig. 5.7. Security Robustness Index as a Function of Amount of Traffic

The results indicate that the proposed piggyback challenge-response protocol significantly outperforms CCMP in terms of the SRI. In the beginning, the SRI due to our proposed protocol is slightly higher as compared to CCMP. As the amount of traffic increases, the reduction in SRI of CCMP is more pronounced. The constant SRI, exhibited by the proposed

piggyback challenge-response protocol is attributed to the unique combination of proposed piggy back challenge response mechanism for every packet and AES in counter mode [45], which is capable of providing enhanced security as compared to CCMP.

5.7 Simulation Results

In this section, simulation results of latency induced by the security mechanism used in proposed piggyback challenge-response protocol and the CCMP employed in IEEE 802.11i are compared. The simulations were performed using the Qualnet simulator (source code is listed at Appendix I to this dissertation). Note that, latency is particularly important for delay-sensitive applications such as network gaming, IP telephony and video conferencing. To test the effect of security provisioning on the latency, we used a simple chain topology, with the source and destination nodes located at the two end-points of the chain. The reason for choosing this model of WMN architecture in our simulation is to observe the results while data is traversed through multiple nodes after having successful mutual authentication and confidentiality between source and destination nodes. In our model source and destination nodes are individual devices using mesh services to communicate with other devices in the network. Our model consists of WMN router nodes which act as access points for the clients as well as forward the traffic for neighboring nodes. Wired gateways (G in Fig. 5.4) are directly connected to the Internet. The router nodes are static, resulting in a more or less fixed topology with infrequent changes due to departing or joining nodes. Some of the routers also serve as access points and allow client nodes to connect to the mesh network.

We observed the effect of increasing the number of intermediate hops on the end-to-end delay in both the schemes. In the beginning, we examined the latency induced between two nodes placed at single hop distance. Then, we recorded the latency for two hops distance and subsequently we increased the number of intermediate hops in both the schemes. The simulation results are shown in Fig. 5.8.

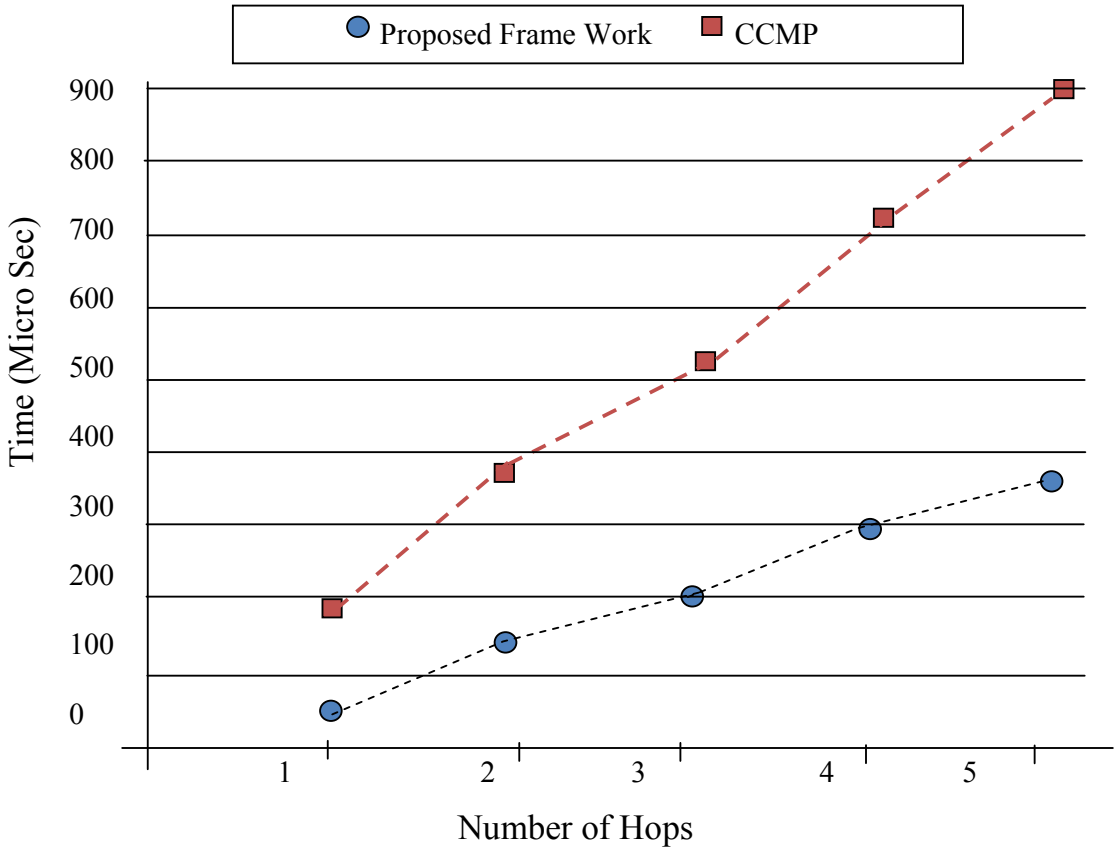


Fig. 5.8. End-to-end delay as a Function of the Number of Nodes in the Chain

The results indicate that the proposed piggyback challenge-response protocol significantly out performs CCMP in terms of the induced latency. For single hop, the latency due to the proposed protocol is less than half as compared to CCMP. The improvement in latency is more pronounced as the

number of intermediate hops increase. The improved performance, in the presence of robust security, exhibited by the proposed piggyback challenge-response protocol is attributed to the unique combination of proposed security mechanism for every packet and AES in counter mode [45], which is capable of faster message processing as compared to AES in 'counter mode with cipher block chaining -message authentication code' (CCM) [26] used in CCMP. Note that, the performance can be further improved if the AES cipher block stream pre-computation is done prior to the receipt of the message. This is possible in the proposed scheme, since the cipher stream is not dependent on the received message. On the contrary, this pre-computation is not possible in CCMP, as the message authentication code is dependent on the received data.

5.8 Summary of Results – Proposed Security Mechanism

In the preceding sections, it is shown that proposed security framework successfully protects the network from passive traffic analysis and prevents malicious nodes from launching a host of attacks including MAC spoofing attacks, replay attacks, pre-computation attacks and partial matching attacks. Besides prevention of attacks, proposed security mechanism saves computational resources being utilized in the calculation of MIC on receiving side, reduces the 64 bits overhead of CCMP header, 64 bits overhead of MIC field per MPDU and improves quality of service for delay sensitive real-time multimedia applications like voice over IP and video on demand. The benefits of proposed security mechanism over existing scheme are listed in Table 5.3.

TABLE 5.3.

Benefits - Proposed Security Mechanism

	Passive Eavesdropping	*MAC Spoofing*	*Replay*	*Pre-Computation*	*Latency*
CCMP	*Yes*	*Yes*	*No*	*No*	*High*
Proposed Security Framework	*No*	*No*	*No*	*No*	*Less than 50%*

5.9 CONCLUSION

This chapter proposes a comprehensive framework for wireless mesh networks having a blend of improved security and low latency. The framework uses a novel hop-by-hop piggybacked challenge response protocol, which provides data confidentiality and integrity at the MAC layer. Initial trust establishment, key distribution and node authentication is carried out using 802.1X over EAP. It is also identified that hop-by-hop security is not sufficient to protect against selfish and compromised node. The scheme also implements end-to-end data confidentiality and data integrity at the network layer. It is also demonstrated that our framework is robust against a wide variety of security breaches including passive eavesdropping, MAC spoofing, replay and pre-computation attacks. Finally, simulation results have shown that the latency induced by the security services in our framework is significantly lower than that observed in CCMP.

Chapter 6 Conclusion

6.1 Introduction

This Chapter summarizes the entire research work. An overview of thesis is presented at second section. The third and fourth sections present the achievements and contributions respectively. The last section recommends the future work that could be undertaken to extend this research work.

6.2 Thesis Overview

Wireless networks are being widely deployed all over the world. This adoption of wireless networks is due to its simple and quick installation, inexpensive equipment, scalable network, less alterations and additions in buildings, and fascination of being wirelessly connected.

Like all the wireless network technologies, IEEE 802.11 based WLAN are also being extensively adopted. Laptops, PDAs, smart mobile phones, security cameras, parking meters, home entertainment devices, printers and peripherals are a few of common platforms that are using WLAN devices. Moreover, Task Group IEEE 802.11s is working on the extension of WLAN from single-hop to multi-hop WMN. WMN standard is enlarging the range of markets and applications for the WLAN. Applications of WMN include unwired campuses and community area networks (hotzones).

Unlike wired networks, signals of wireless networks are present in the air at ranges corresponding to their frequencies and power and can be received by everyone present in the vicinity. The wireless frequency is also a limiting factor for achieving significant throughput at required ranges. Likewise, quality of signals is badly affected due to interference in the air. Accordingly, security and quality of service have been the challenging areas, over which much of research work for wireless networks have been focused.

Initially, some weaknesses were highlighted in the encryption mechanism of IEEE 802.11 wireless networks. The WEP mechanism was hence developed to create the level of privacy experienced on wired LANs. Again, some flaws in the WEP were highlighted. Subsequently, 802.11i - amendment for WLAN security mechanism- replaced WEP by providing enhanced security. 802.11i offers port based access control mechanism for node authentication. Data confidentiality and integrity is ensured through any of WEP, TKIP or CCMP.

IEEE 802.11s -the draft standard for WMN –has also proposed to use CCMP of IEEE 802.11i for security in WMN.CCMP is a two pass process. While this two pass processing is suitable for single-hop WLAN, it induces considerable latency in multi-hop wireless networks, such as WMN. The increase in latency can lead to the decrease in the quality of service for delay sensitive applications like voice over IP and video on demand.

The breach in the WEP based WLAN security and the introduction of CCMP, having considerable latency due to two pass processing, motivated this research to evaluate the IEEE 802.11i standard for possible

vulnerabilities and to find ways to provide robust and low latency security mechanism.

This research exposes the vulnerability in the security mechanism of IEEE 802.11i WLAN. The vulnerability is due to weak implementation of counter mode for block cipher, which renders the 802.11 WLAN exposed to 'Time Memory Trade off' (TMTO) attack [10]. This work [10] has been cited in book [13], published by 'Springer Verlag'.

Sequel to above, remedial measures to avoid TMTO attack on 802.11 based WLAN are suggested. In this endeavor, a per packet authentication mechanism having capability to successfully defend TMTO attack has been proposed [11]. Furthermore, to provide low latency and robust security to WMN, a very reliable and low latency security framework, suggesting piggy back authentication system, is proposed [12]. This security framework was simulated on Qualnet software, during my visit to University of New South Wales (UNSW), Sydney. The simulation results verified the low latency and high reliability characteristics of proposed framework.

6.3 Achievements

IEEE has enhanced the security mechanisms for WLAN and WMN with 802.11i, but the standard still suffers from the similar problems that previous standards did. The CCMP introduced in the 802.11i improved the security but the vulnerabilities and latency due to two pass process still exist. The resulting latency limits the performances for multimedia application in WMN (multi-hop wireless LANs). This research exposes the vulnerability of CCMP against pre-computation TMTO attack and proposes a low latency security framework which addresses the above mentioned

problems without compromising any of the security measures implemented in the standard. Details of achievements are elaborated in the ensuing paragraphs.

It has been shown during the course of research that the nonce, used in the CCMP, can be reconstructed by an unauthorized user. Consequently, initial counter value can also be reconstructed. Thus, TMTO pre-computation attack becomes possible on CCMP, which renders entire security framework of 802.11WLAN and WMN ineffective. To strengthen the weak links of CCMP, we propose per packet authentication mechanism. It is shown that per packet authentication mechanism obviates the requirement of MIC, thus improves latency and computational overheads. The Per-Packet authentication promptly secures the connection against unauthorized access by immediately discarding the packet if Per-Packet Authentication fails. It is proposed to derive the Nonce from the session key and is kept secret. Encrypted and unique nonce provides unpredictability and freshness. Unpredictability prevents pre-computation attack and freshness ensures defense against replay attacks.

A comprehensive framework for wireless mesh networks with a blend of improved security and low latency is also proposed. The framework uses a novel hop-by-hop piggybacked challenge response protocol, which provides data confidentiality and integrity at the MAC layer. Initial trust establishment, key distribution and node authentication is carried out using 802.1X over EAP. It is also identified that hop-by-hop security is not sufficient to protect against a selfish and compromised node. Therefore, end-to-end data confidentiality and data integrity at the network layer is also offered. It is also shown that proposed framework is robust against a wide

variety of cyber attacks including passive eavesdropping, MAC spoofing, replay and pre-computation attacks. Finally, simulation results verify that the latency induced by the security services in our framework is reduced to half as compared to CCMP.

6.4 Contributions

This research is an endeavor to analyze the existing cutting edge security mechanism of WLAN and WMN for the benefit of the exponentially increasing wireless users around the world. Accordingly, this research successfully discovers the vulnerability in the CCMP [10]. It is also demonstrated that CCMP, being two pass process, will introduce latency and is not suitable for multi-hop WMN. The increase in latency leads to the decrease in the quality of service for delay sensitive real-time multimedia applications like voice over IP and video on demand. As a remedial action, this research contributes a novel, secure and low latency security mechanism for WLAN and WMN. This research introduces a novel index to measure the degree of robustness. This index is named as 'Security Robustness Index' (SRI). SRI provides a method to judge the cryptographic strength of security framework. The simulation results verified the low latency characteristics of proposed security framework vis-à-vis CCMP.

6.5 Suggestion for Future Research

It is highly desirable to assess the performance improvement achieved by proposed security framework when applied to real 802.11 platforms. To this end, there is a need to decide on a suitable vendor who could implement proposed security mechanism while using rest of the paraphernalia of 802.11.

Further suggestion for the future work, arising out of this research, is to employ algorithms other than AES. This research employs default AES algorithm in proposed security mechanism. AES can be replaced with other block ciphers in counter mode. The performance of proposed security mechanism with different algorithms can then be measured in terms of efficacy against the cyber-attacks and the effect on latency of the security mechanism.

An alternative mode to CCM mode is EAX mode – an Authenticated Encryption with Associated Data mode. EAX mode combines the use of Counter mode and One Key MAC algorithm (OMAC). EAX has advantage over CCM because EAX mode can begin to process data as it arrives. Moreover, EAX can preprocess fixed associated data. A comparative analysis of CCM versus EAX in the context of WLAN and WMN security would be a significant contribution in the field of information security.

Other authentication encryption modes, such as offset codebook mode (OCB) and AES Key wrap need to be evaluated against CCMP for the provision of security services to WLAN and WMN.

Appendix:

I. Source Code for the Simulation Program of Security Framework on Qualnet Simulation Software.

```
//
//=========================================================
=======
// Secure,Reliable and Low Latency Frame work for WMN//
//=========================================================
=======
//
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <assert.h>
#include "aes_defs.h"
//#include "aes_vect.h"
#include "mac-802_11i.h"

struct prng_t prng;
block new_tmp_nounce;
/*
Mac802_11i::Mac802_11i()
{
    this((int) time(NULL));
}
*/
Mac802_11i::Mac802_11i(int seed)
{
  InitRand(seed);
}
void
Mac802_11i::InitRand(u32b seed)
```

```
{
    memset(prng.ptCntr.b,0,BLK_SIZE);
    prng.ptCntr.x[(BLK_SIZE/4)-1]=bswap(seed*17);
    prng.cnt=0;
u32b
Mac802_11i::Random32(void)
{
    int  i;
    u32b x;
    if (prng.cnt == 0)
    {
        prng.cnt=BLK_SIZE/4;
        for (i=0;i<16;i++)
        {
            prng.ptCntr.b[i]++;
            if (prng.ptCntr.b[i])
                break;
        }
        AES_Encrypt(prng.ptCntr.x,prng.ct.x);
    }
    --prng.cnt;
    for (i=x=0;i<4;i++)
        x = (x << 8) + prng.ct.b[4*prng.cnt+i];
    return x;
}
void
Mac802_11i::ShowBlock(const block *blk,const char
*prefix,const char *suffix,int
a)
{
    int i,blkSize = BLK_SIZE;
    printf(prefix,a);
    if (suffix == NULL) { suffix = "\n"; blkSize = a; }
```

```
    for (i=0;i<blkSize;i++)
        printf("%02X%s",blk->b[i],((i&3)==3)?"  ":" ");
    printf(suffix);
}
void
Mac802_11i::ShowAddr(const packet *p)
{
    int i;
    printf("      TA = ");
    for (i=0;i<6;i++) printf("%02X%s",p->TA[i],(i==3)?"
":" ");
    printf("  48-bit pktNum = %04X.%08X\n",p->pktNum[1],p-
>pktNum[0]);
}
void
Mac802_11i::ShowPacket(const packet *p,const char
*pComment,int a)
{
    int i;
    printf("Total packet length = %4d. ",p->length);
    printf(pComment,a);
    if (p->encrypted) printf("[Encrypted]");
    for (i=0;i<p->length;i++)
    {
        if ((i & 15) == 0) printf("\n%11s","");
        printf("%02X%s",p->data[i],((i&3)==3)?"  ":" ");
    }
    printf("\n");
}
void
Mac802_11i::set_key(const u32b in_key[], const u32b
key_len)
{
```

```
    AES_SetKey(in_key,key_len);
}
void
Mac802_11i::encrypt(packet *p,int verbose)
{
    int     i,j,len,needPad,blkNum;
    block   m,x,T;
    assert(p->length    >= p->clrCount && p->length    <=
MAX_PACKET);
    assert(p->micLength >  0           && p->micLength <=
BLK_SIZE);
    len = p->length - p->clrCount;

    //ShowPacket(p,"[Input (%d cleartext header
octets)]",p->clrCount);
    m.b[ 0] =(u08b) ((L_SIZE-1) << L_SHIFT) +      // flags
octet
        ((p->clrCount)?A_DATA:0) + (((p->micLength-2)/2 <<
M_SHIFT));
    m.b[ 1] = N_RESERVED;               // reserved nonce
octet
    m.b[ 2] = Lo8(p->pktNum[1] >> 8);// 48 bits of packet
number ("IV")
    m.b[ 3] = Lo8(p->pktNum[1]);
    m.b[ 4] = Lo8(p->pktNum[0] >>24);
    m.b[ 5] = Lo8(p->pktNum[0] >>16);
    m.b[ 6] = Lo8(p->pktNum[0] >> 8);
    m.b[ 7] = Lo8(p->pktNum[0]);
    m.b[ 8] = p->TA[0];
    m.b[ 9] = p->TA[1];
    m.b[10] = p->TA[2];
    m.b[11] = p->TA[3];
    m.b[12] = p->TA[4];
```

```
    m.b[13] = p->TA[5];
    m.b[14] = Lo8(len >> 8);
    m.b[15] = Lo8(len);
    //---- compute the CBC-MAC tag (MIC)
    AES_Encrypt(m.x,x.x);
    //ShowBlock(&m,"CBC IV in: ","\n",0);  //priting out
     //the unencrypted IV
(NOUNCE)
    if (verbose) ShowBlock(&x,"CBC IV out:","\n",0);
    j=0;
block
    if (p->clrCount)
    {
        //printf("Does this ever enter here"); // if so,
"insert" length field:
l(a)
        assert(p->clrCount < 0xFFF0);   // [don't handle
larger cases (yet)]
        x.b[j++]^=(p->clrCount >> 8) & 0xFF;
        x.b[j++]^= p->clrCount       & 0xFF;
    }
    for (i=blkNum=0;i<p->length;i++)    // do the CBC-MAC
processing
    {
        x.b[j++] ^= p->data[i];
        needPad = (i == p->clrCount-1) || (i == p->length-
1);
        if ((j == BLK_SIZE) || needPad)
        {
            if (verbose) ShowBlock(&x,"After xor: ",
                (i >= p->clrCount) ? " [msg]\n" : "
[hdr]\n",blkNum);
            AES_Encrypt(x.x,x.x);
```

```
            if (verbose) ShowBlock(&x,"After AES:
","\n",blkNum);
            blkNum++;
            j = 0;
        }
    }
    memcpy(T.b,x.b,p->micLength);
    //ShowBlock(&T,"MIC tag  : ",NULL,p->micLength);
   //---- encrypt the data packet using CTR mode
    m.b[0] &= ~ (A_DATA | (7<<M_SHIFT));
    for (i=blkNum=0;i+p->clrCount < p->length;i++)
    {
        if ((i % BLK_SIZE) == 0)
        {                           // generate new
keystream block
            blkNum++;                   // start data with
block #1
            m.b[14] = blkNum/256;
            m.b[15] = blkNum%256;
            AES_Encrypt(m.x,x.x);       // then encrypt the
counter
            if (verbose && i==0) ShowBlock(&m,"CTR Start:
","\n",0);
            if (verbose) ShowBlock(&x,"CTR[%04X]: "
,"\n",blkNum);
        }
        p->data[i+p->clrCount] ^= x.b[i % BLK_SIZE];    //
merge in the
keystream
    }
    m.b[14] = m.b[15] = 0;
    AES_Encrypt(m.x,x.x);
```

```
    if (verbose) ShowBlock(&x,"CTR[MIC ]: " ,NULL,p-
>micLength);
    for (i=0;i<p->micLength;i++)
    {
        p->data[p->length+i]=T.b[i] ^ x.b[i];
        //printf("current length value is:%d ", p-
>length+i);
        //printf("encrypted mic:%02X%s \n", p->data[p-
>length+i] );
        //printf("T.b value:%02X%s \n", T.b[i] );
        //printf("x.b value:%02X%s \n", x.b[i] );
    }
    //printf("the length here is %d\n",p->length);
    p->length+=p->micLength;
    p->encrypted = 1;
    //printf("Outputting the packet now\n");
    //ShowPacket(p,"",0);                 // show the final
encrypted packet
}
block*
Mac802_11i::Random128(block seed)
{
    block* random = new block;
    for (int i = 0; i < 4; i++)
    {
        InitRand(seed.x[i]);
        random->x[i] = Random32();
        //printf("the random value generated is: %d" +
random[i]);
        //ShowBlock(&seed,"old seed: "," \n",0);
    }
    return random;
}
```

```
void
Mac802_11i::encryptCTR(packet *p, block old_nounce)
{
    int verbose = 0;
    block T,m,x,e;
    int i,blkNum;
    bool needPad;
    //AES_setKey(old_nounce);
    //block* new_nounce = Random128(old_nounce);
    for ( i = 0; i < 4; i++) {
        new_tmp_nounce.x[i] = i;
    }
    block* new_nounce = &new_tmp_nounce;
    //block* try_again = Random128(old_nounce);
    u08b* encrypted_data = new u08b[p->length];
    AES_Encrypt(new_nounce->x, T.x);    //encrypt the nonce
    for (i = 0; i < BLK_SIZE; i++) {
        //encrypted_data[i] = T.b[i];
        encrypted_data[i] = p->data[i]; //copy the first
block somewhere else
        p->data[i] = T.b[i];            //to avoid being
overwritten by
        //m.b[i] = new_nounce[i];       //encrypted nonce
        //e.b[i] = try_again[i];
    }
    //ShowPacket(p,"[Input (%d cleartext header
octets)]",p->clrCount);
    //ShowBlock(new_nounce,"new nounce: "," \n",blkNum);
    //ShowBlock(try_again,"try again: "," \n",blkNum);
    //ShowBlock(&T,"new encrypted nounce: "," \n",blkNum);

    int j = 0;
    for (i=blkNum=0;i < p->length;i++)
```

```
    {
        if ((i % BLK_SIZE) == 0)
        {                           // generate new
keystream block
            blkNum++;                   // start data with
block #1
            new_nounce->b[14] = blkNum/256;
            new_nounce->b[15] = blkNum%256;
            AES_Encrypt(new_nounce->x,x.x);       // then
encrypt the counter
            if (verbose) ShowBlock(new_nounce,"CTR Start:
","\n",0);
            if (verbose) ShowBlock(&x,"CTR[%04X]: "
,"\n",blkNum);
        }
        if (i < p->length-BLK_SIZE)
            encrypted_data[i+BLK_SIZE] = p-
>data[i+BLK_SIZE];

        p->data[i+BLK_SIZE] = encrypted_data[i] ^ x.b[i %
BLK_SIZE];    // merge
in the keystream
    }
    p->length = p->length + BLK_SIZE;
    // for (int i = 0; i < p->length; i++)
    //    p->data[i] = encrypted_data[i];
    //p->data = encrypted_data;
    //ShowPacket(p,"Encrypted",0);
}
void
Mac802_11i::decryptCTR(packet *p, block old_nounce)
{
    int verbose = 0;
```

```
    block T,T_dec,d,x;
    int i,blkNum = 0;
    bool needPad;
    int k=0, j=0;
    for ( i = 0; i < 4; i++) {
        new_tmp_nounce.x[i] = i;
    }
        block* new_nounce = &new_tmp_nounce;
    //block* new_nounce = Random128(old_nounce);
    u08b* decrypted_data = new u08b[p->length - BLK_SIZE];
    //AES_Encrypt(new_nounce->x, T.x);

    for (int i = 0; i < BLK_SIZE; i++) {
        //d.b[i] = new_nounce[i];
        T.b[i] = p->data[i];
    }
    AES_Decrypt(T.x,T_dec.x);   //decrypt the encrypted
nonce
    if (memcmp(T_dec.b,new_nounce->b,BLK_SIZE) != 0)
     printf("NOUNCE ISN'T THE SAME\n");

    for (int i = BLK_SIZE; i < p->length; i++)
    {
        T.b[i % BLK_SIZE] = p->data[i];
        j++;
        needPad = (i == p->length-1);
        if ((i % BLK_SIZE) == 0)
        {                           // generate new
keystream block
            blkNum++;                   // start data with
block #1
            new_nounce->b[14] = blkNum/256;
            new_nounce->b[15] = blkNum%256;
```

```
            AES_Encrypt(new_nounce->x,x.x);       // then
encrypt the counter
            if (verbose) ShowBlock(new_nounce,"CTR Start:
","\n",0);
            if (verbose) ShowBlock(&x,"CTR[%04X]: "
,"\n",blkNum);
        }
        p->data[i-BLK_SIZE] =  x.b[i % BLK_SIZE] ^ p-
>data[i];
    }
    p->length = p->length - BLK_SIZE;   //adjust the length
of the data
//    for (int i = 0; i < p->length; i++)
//        p->data[i] = decrypted_data[i];
    p->encrypted = 0;
    //ShowPacket(p,"Decrypted",0);                  // show
the final encrypted
packet
}
void
Mac802_11i::decrypt(packet *p,int verbose)
{
    int     i,j,len,needPad,blkNum;
    block   m,x,T2,T;
    assert(p->length    >= p->clrCount && p->length    <=
MAX_PACKET);
    assert(p->micLength >  0           && p->micLength <=
BLK_SIZE);
    //printf("the length here is %d\n",p->length);
    //ShowPacket(p,"",0);
     p->length-=p->micLength;
    len = p->length - p->clrCount;      // l(m)
```

```
    m.b[ 0] =(u08b) ((L_SIZE-1) << L_SHIFT) +      // flags
octet
        ((p->clrCount)?A_DATA:0) + (((p->micLength-2)/2 <<
M_SHIFT));
    m.b[ 1] = N_RESERVED;               // reserved nonce
octet
    m.b[ 2] = Lo8(p->pktNum[1] >> 8);   // 48 bits of
packet number ("IV")
    m.b[ 3] = Lo8(p->pktNum[1]);
    m.b[ 4] = Lo8(p->pktNum[0] >>24);
    m.b[ 5] = Lo8(p->pktNum[0] >>16);
    m.b[ 6] = Lo8(p->pktNum[0] >> 8);
    m.b[ 7] = Lo8(p->pktNum[0]);
    m.b[ 8] = p->TA[0];                 // 48 bits of
transmitter address
    m.b[ 9] = p->TA[1];
    m.b[10] = p->TA[2];
    m.b[11] = p->TA[3];
    m.b[12] = p->TA[4];
    m.b[13] = p->TA[5];
    m.b[14] = Lo8(len >> 8);            // l(m) field
    m.b[15] = Lo8(len);
    m.b[0] &= ~ (A_DATA | (7<<M_SHIFT));// clear flag
fields for counter mode

    for (i=blkNum=0;i+p->clrCount < p->length;i++)
    {
        if ((i % BLK_SIZE) == 0)
        {                           // generate new
keystream block
            blkNum++;                   // start data with
block #1
            m.b[14] = blkNum/256;
```

```
            m.b[15] = blkNum%256;
            AES_Encrypt(m.x,x.x);       // then encrypt the
counter
            if (verbose && i==0) ShowBlock(&m,"CTR Start:
","\n",0);
            if (verbose) ShowBlock(&x,"CTR[%04X]: "
,"\n",blkNum);
        }
        p->data[i+p->clrCount] ^= x.b[i % BLK_SIZE];    //
merge in the
keystream
    }
    //ShowPacket(p,"[Input (%d cleartext packet data)]",p-
>clrCount);

    //---- truncate, encrypt, and append MIC to packet
    m.b[14] = m.b[15] = 0;              // use block
counter value zero for tag
    AES_Encrypt(m.x,x.x);               // encrypt the
counter
    if (verbose) ShowBlock(&x,"CTR[MIC ]: " ,NULL,p-
>micLength);
    //memcpy(T.b,&p->data[p->length],p->micLength);
    //printf("the length here is %d\n",p->length);

    for (i=0;i<p->micLength;i++) {
        T2.b[i] = p->data[p->length+i] ^ x.b[i];
        //printf("Outputting each of the mic stuff\n");
        //ShowBlock(&p->data[p->length+i],NULL,1);
        //printf("current length value is:%d ", p-
>length+i);
        //printf("encrypted mic:%02X%s \n", p->data[p-
>length+i] );
```

```
        //printf("T2.b value:%02X%s \n", T2.b[i] );
        //printf("x.b value:%02X%s \n", x.b[i] );
        //printf("encrypted mic:%02X%s and the p.length
value is:%d \n", p-
>data[p->length+i], p->length+i);

        //p->data[p->length+i]=T.b[i] ^ x.b[i];
    }
    //ShowBlock(&T,"Decrypted MIC tag  : ",NULL,p-
>micLength);
    //memcpy(T2.b,T.b,p->micLength);
    //ShowBlock(&T2,"Decrypted MIC tag : ",NULL,p-
>micLength);

    //calculating the cbc-mic from the header and the
packet now
    assert(p->length    >= p->clrCount && p->length    <=
MAX_PACKET);
    assert(p->micLength >  0           && p->micLength <=
BLK_SIZE);

    //ShowPacket(p,"[Input (%d cleartext header
octets)]",p->clrCount);

    //---- generate the first AES block for CBC-MAC
    m.b[ 0] =(u08b) ((L_SIZE-1) << L_SHIFT) +      // flags
octet
                    ((p->clrCount)?A_DATA:0) + (((p-
>micLength-2)/2 <<
M_SHIFT));
    m.b[ 1] = N_RESERVED;               // reserved nonce
octet
```

```
    m.b[ 2] = Lo8(p->pktNum[1] >> 8);   // 48 bits of
packet number ("IV")
    m.b[ 3] = Lo8(p->pktNum[1]);
    m.b[ 4] = Lo8(p->pktNum[0] >>24);
    m.b[ 5] = Lo8(p->pktNum[0] >>16);
    m.b[ 6] = Lo8(p->pktNum[0] >> 8);
    m.b[ 7] = Lo8(p->pktNum[0]);
    m.b[ 8] = p->TA[0];                 // 48 bits of
transmitter address
    m.b[ 9] = p->TA[1];
    m.b[10] = p->TA[2];
    m.b[11] = p->TA[3];
    m.b[12] = p->TA[4];
    m.b[13] = p->TA[5];
    m.b[14] = Lo8(len >> 8);            // l(m) field
    m.b[15] = Lo8(len);

    //---- compute the CBC-MAC tag (MIC)
    AES_Encrypt(m.x,x.x);               // produce the CBC
IV
    //ShowBlock(&m,"CBC IV in: ","\n",0);
    if (verbose) ShowBlock(&x,"CBC IV out:","\n",0);
    j=0;                                // j = octet
counter inside the AES
block
    if (p->clrCount)                    // is there a
header?
    {                                 // if so, "insert"
length field: l(a)
        assert(p->clrCount < 0xFFF0);   // [don't handle
larger cases (yet)]
        x.b[j++]^=(p->clrCount >> 8) & 0xFF;
        x.b[j++]^= p->clrCount       & 0xFF;
```

```
    }
    for (i=blkNum=0;i<p->length;i++)    // do the CBC-MAC
processing
    {
        x.b[j++] ^= p->data[i];         // perform the CBC
xor
        needPad = (i == p->clrCount-1) || (i == p->length-
1);
        if ((j == BLK_SIZE) || needPad) // full block, or
hit pad boundary
        {
            if (verbose) ShowBlock(&x,"After xor: ",
                                   (i >= p->clrCount) ? "
[msg]\n" : "
[hdr]\n",blkNum);
            AES_Encrypt(x.x,x.x);       // encrypt the CBC-
MAC block, in place
            if (verbose) ShowBlock(&x,"After AES:
","\n",blkNum);
            blkNum++;                   // count the blocks
            j = 0;                      // the block is now
empty
        }
    }
    memcpy(T.b,x.b,p->micLength);       // save the MIC tag
    //ShowBlock(&T,"Calculated MIC tag  : ",NULL,p-
>micLength);
    if (memcmp(T.b,T2.b,p->micLength) == 0) {
        p->encrypted = 0;
        //printf("Outputting the packet now\n");
        //ShowPacket(p,"[Input (%d cleartext header
octets)]",p->clrCount);
    }
```

```
    else
    {
        printf("Calculated and Decrypted MIC tags are not
same");
    }
    //p->length+=p->micLength;            // adjust packet
length accordingly
}

/*int main(int argc,char *argv[])
{
    int     i,j,k,len,pktNum,seed;
    packet  p;

    seed = (argc > 1) ? atoi(argv[1]) : (int) time(NULL);
    Mac802_11i ccmp(seed);

     for (k=pktNum=0;k<2;k++)
        {   // k==1 --> random vectors. k==0 --> "visually
simple" vectors
        for (i=0;i<BLK_SIZE  ;i++)
            p.key.b[i] = (k) ? (u08b) ccmp.Random32() &
0xFF : i + 0xC0;
        for (i=0;i<6;i++)
            p.TA[i]    = (k) ? (u08b) ccmp.Random32() &
0xFF : i + 0xA0;
        //AES_SetKey(p.key.x,BLK_SIZE*8);     // run the
key schedule
        ccmp.set_key(p.key.x, BLK_SIZE*8);

        // now generate the vectors
        //for (p.micLength  = 8;p.micLength
<12;p.micLength+=2)
```

```
        //for (p.clrCount   = 8;p.clrCount
<16;p.clrCount+=4)
        //for (len          =32;len          <64;len*=2)
        //for (i            =-1;i            < 2;i++)
         //   {
            p.micLength = 8;
            p.clrCount = 8;
            len = 40;
            p.pktNum[0] = (k) ? ccmp.Random32()          :
pktNum*0x01010101 +
0x03020100;
            p.pktNum[1] = (k) ? ccmp.Random32() & 0xFFFF :
0;    // 48-bit IV
            p.length    = len;           // len+i is packet
length
            p.encrypted = 0;
            assert(p.length <= MAX_PACKET);
            for (j=0;j<p.length;j++)        // generate
random packet contents
                p.data[j]=(k) ? (u08b  ) ccmp.Random32() &
0xFF : j;
            pktNum++;
            printf("=============== Packet Vector #%d
==================\n",pktNum);
            ccmp.ShowBlock(&p.key ,"AES Key:   ","\n",0);
            printf("================Block
Shown=========================\n");
            ccmp.ShowAddr (&p);
            printf("==================Address
Shown======================\n");
            ccmp.ShowPacket(&p,"",0);
            //ccmp.encrypt(&p,1);
            //ccmp.decrypt(&p,1);
```

```
            //ccmp.encrypt(&p,p.key.x);
            //ccmp.decrypt(&p,p.key.x);
            ccmp.encryptCTR(&p,p.key);
            ccmp.decryptCTR(&p,p.key);
         //   }
        }
    return 0;
    }
*/
/*
void
Mac802_11i::encrypt(packet *p, block old_nounce)
{
    block T,T_enc,m;
    int j = 0,blkNum = 0;
    bool needPad;
    u32b* new_nounce = Random128(old_nounce);
    u08b* encrypted_data = new u08b[BLK_SIZE + p->length];
    AES_Encrypt(new_nounce, T_enc.x);
    for (int i = 0; i < BLK_SIZE; i++) {
        encrypted_data[i] = T_enc.b[i];
        m.b[i] = new_nounce[i];
    }
    ShowBlock(&m,"new nounce: "," [msg]\n",blkNum);
    int k = BLK_SIZE;
    for (int i = 0; i < p->length; i++)
    {
        T.b[i % BLK_SIZE] = p->data[i];
        j++;
        needPad = (i == p->length-1);
//        printf("the value of i is %d\n", i);
        if ((j == BLK_SIZE) || needPad) // full block, or
hit pad boundary
```

```
        {
            ShowBlock(&T,"Unencrypted xor: "," 
[msg]\n",blkNum);
            AES_Encrypt(T.x,T_enc.x);   // encrypt the CBC-
MAC block, in place
            ShowBlock(&T_enc,"After AES: ","\n",blkNum);
            // here: need to think about how to add the 
encrypted packet onto
            //use needPad to solve the existing problem
            for (; k <= i+BLK_SIZE && k < p->length + 
BLK_SIZE; k++) {
//                printf("the value of k is %d \n", k);
                encrypted_data[k] = T_enc.b[k % BLK_SIZE];
                T.b[k%BLK_SIZE] = T_enc.b[k%BLK_SIZE];
                //T.b[k%BLK_SIZE] = 0x00;
            }
//            ShowBlock(encrypted_data,"something is 
wrong", "\n", blkNum);
            blkNum++;                   // count the blocks
            j = 0;                      // the block is now 
empty
        }
    }

    p->length = p->length + BLK_SIZE;
    for (int i = 0; i < p->length; i++)
        p->data[i] = encrypted_data[i];
    ShowPacket(p,"",0);                 // show the final 
encrypted packet
}
void
Mac802_11i::decrypt(packet *p, block old_nounce)
{
```

```
    block T,T_dec;
    int j = 0,blkNum = 0;
    bool needPad;
    u32b* new_nounce = Random128(old_nounce);
    u08b* decrypted_data = new u08b[p->length - BLK_SIZE];
//    AES_Decrypt(new_nounce, T_enc.x);
//   for (int i = 0; i < BLK_SIZE; i++) {
//        encrypted_data[i] = T_enc.b[i];
//    }
    int k = 0;
    for (int i = 0; i < p->length; i++)
    {
        T.b[i % BLK_SIZE] = p->data[i];
        j++;
        needPad = (i == p->length-1);

        if ((j == BLK_SIZE) || needPad) // full block, or
hit pad boundary
        {
            ShowBlock(&T,"Encrypted xor: ","
[msg]\n",blkNum);
            AES_Decrypt(T.x,T_dec.x);   // encrypt the CBC-
MAC block, in place
            ShowBlock(&T_dec,"After DECRYPT:
","\n",blkNum);
            if (blkNum == 0) {
                if (memcmp(T_dec.b,new_nounce,BLK_SIZE) !=
0)
                    printf("NOUNCE ISN'T THE SAME\n");
            }
            else
```

```
                // here: need to think about how to add the
encrypted packet
onto
                for (; k <= i-BLK_SIZE && k < p->length -
BLK_SIZE; k++) {
                    decrypted_data[k] = T_dec.b[k %
BLK_SIZE];
//                    T.b[k%BLK_SIZE] =
T_dec.b[k%BLK_SIZE];
                }
//            ShowBlock(encrypted_data,"something is
wrong", "\n", blkNum);
            blkNum++;                   // count the blocks
            j = 0;                      // the block is now
empty
        }
    }
    p->length = p->length - BLK_SIZE;
    for (int i = 0; i < p->length; i++)
        p->data[i] = decrypted_data[i];
    ShowPacket(p,"",0);                 // show the final
encrypted packet
}
*/
```

II. Transcript for Exchange of Questions / Answers with the Evaluator

pp.4, 9

"the message is decrypted using counter mode to obtain the MAC and the corresponding payload. Then the integrity check is performed and if successful, the packet is decrypted and delivered to the higher layer (i.e. network layer)"

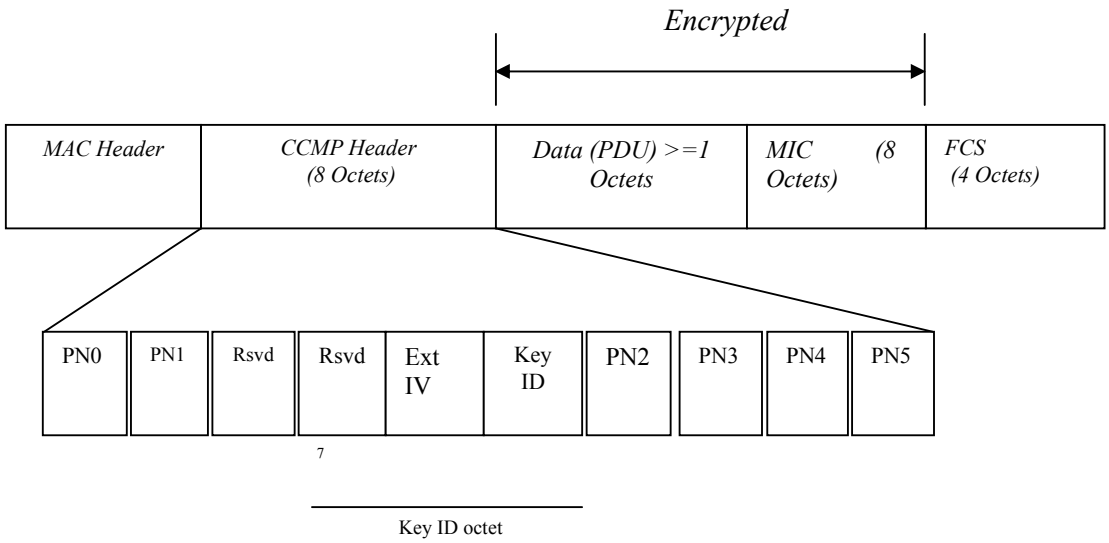
Q. Decryption by the receiver twice requires using keys twice. Are these two public keys or same key used twice? How does receiver get these keys at the outset of the transmission? Please explain.

A. *IEEE* 802.11i requires authentication in two phases; the first is an open system authentication and the second uses IEEE 802.1X along with Extensible Authentication Protocol (EAP) authentication method. For networks without a Remote Authentication Dial-In User Service (RADIUS) infrastructure such as small offices and home networks, 802.11i supports the use of a pre-shared key (PSK).

The key management feature of IEEE 802.11i requires the determination of a mutual pairwise master key (PMK) based on the Extensible Authentication Protocol (EAP) or Pre shared Key (PSK) authentication processes and the calculation of pairwise transient keys through a 4-way handshake. Subsequently, pairwise key hierarchy utilizes pseudorandom functions (PRF) to derive session-specific keys from a pairwise master key (PMK). The PMK is available as a result of successful IEEE 802.1X exchange, pre-shared key (PSK) or PMK cached via some other mechanism. The PMK is 256 bits. The pairwise key hierarchy takes the PMK and generates a pairwise transient key (PTK). The PTK further generates temporal key (TK) . This temporal key is the shared encryption key used in the AES counter mode to encrypt the Data and MIC.

***Q. The IV and ciphertext generated should be authenticated with a secure MAC by the receiver prior to decryption is not a new idea. How is this integrity* check performed? If it is done by secure MAC how does the receiver confirm the authentication?**

A. Sir, I may first elaborate here that in 802.11 standard, terminology of MAC and MIC need slight clarification. MAC is Media Access Control sub layer and MIC is message integrity code**.** The format of MPDU is illustrated below:



On receiving the encrypted MPDU, the MPDU is first decrypted by using the temporal key (explained in previous answer). As a result of successful decryption, the MIC is obtained. Also the additional authentication data (AAD) and nonce values are extracted from the encrypted MPDU. AAD is extracted from the MPDU header and Nonce value is formed from the A2, PN, and Priority Octet fields. Subsequently, the cyclic Block Chaining (CBC) encryption is performed on the received data to obtain another MIC. The resulting MIC and the MIC received through incoming packet is then compared for integrity check.

pp.24 - Fig. .3.2. Reconstruction of Initial counter

"The extraction of fields to pre-compute the initial counter value is illustrated in Fig. 3.2. Any unauthorized user may calculate the counter value irrespective of undergoing the successful authentication process."

Q. Even if unauthorized user may calculate the initial counter value how is this a security threat if the remaining numerous counter values can not be known (without timing out the sessions – rendering it useless), specially if the counter is a

computed function that does not increment monotonically during the same session rather than a simple counter?

A. *Yes Sir, this is the vulnerability highlighted in our published paper and cited in the book by springer verlag. The IEEE 802.11i specify that counter is to increment monotonically.*

It is pertinent to refer a paper , "Counter Mode Security: Analysis and commendations" by David A. McGrew,, Cisco Systems, November, 2002 , which .recommends that at least one of the following points for effective defense against TMTO precomputation attack:

a. There must be 64 bits unpredictable value to the initial counter, which is considered as part of the AES CM key, or

b. Use a predictable but uniformly distributed component in the initial counter, or

c. The key length should be larger than 128 bits.

We have observed that none of these recommendations has been incorporated in the IEEE 802.11i standard, resulting in exposure to TMTO pre-computation attack.

pp.27

"We have proved the vulnerability that these values can be pre-computed by an unauthorized user leading to TMTO pre-computation attack."

Q. Sounds very strong! Have you actually proved empirically or through simulation? Or it is just your conviction and hypothesis based upon the literature survey? Have you referred "Making a Faster Cryptanalytic Time-Memory Trade-Off", Philippe Oechslin Laboratoire de Securit´e et de Cryptographie (LASEC) Ecole Polytechnique F´ed´erale de Lausanne Facult´e I&C, 1015 Lausanne, Switzerland philippe.oechslin@epfl.ch ?

A. *Fabricating a hardware to prove TMTO is a huge task as also endorsed by the book* by Robert, M., Zahir,T, titled "On the Move to Meaningful Internet Systems", Springer-Verlag Berlin Heidelberg, Germany, November, 2007. However, we have systematically showed that the nonce can be reconstructed

, which in turn make it possible to reconstruct the initial counter. Furthermore, by referring to findings and recommendations of different paper (referred in the thesis) we established that effective key size of the algorithm reduces below the recommended minimal key length.

pp.45

"As a general procedure for initial trust establishment and the key distribution, the nodes that are already connected to the WMN (initially only the wired gateways) broadcast EAP request messages at regular intervals. Any unauthenticated node (call it joining node) that is within the transmission range of the broadcasting node (call it connected node) can request to join the WMN by replying to the EAP request message."

Q. Shouldn't "request messages" (sounds like messages from nodes requesting to join) be termed as "Invitation to Join (IJ) messages"?

A. *Yes Sir, this sounds more appropriate, but IEEE 802.11 s - standard for WMN- calls it 'EAP request message'.*

pp.62 - TABLE 5.1 & 5.2.

"To measure the SRI of a security mechanism, different weights are assigned.."

Q. What criteria is used to assign weights?

A. *The criteria Is the established/ known security strength of cryptographic elements against different levels of breaks i.e. total break , global deduction, instance deduction and information deduction. For example, DES is already broken i.e. total break, therefore this cryptographic algorithm will carry no weight. Other algorithms such as , MARS, SERPENT and TWO FISH were among the first five finalist of the AES competition. Since these algorithms along with the*

RIJNDAEL (AES) successfully reached to the top five of the AES competition by NIST , these Algorithms are awarded highest weights.

pp.68

"The improved performance, in the presence of robust security, exhibited by the proposed piggyback challenge response protocol is attributed to the unique combination of our proposed security mechanism for every packet and AES in counter mode [45], which is capable of faster message processing as compared to AES in 'counter mode with cipher block chaining -message authentication code' (CCM) [26] used in CCMP."

Q. Using a simple deterministic input function deliberately exposes a cryptosystem to a known systematic input. Is this is not an unnecessary risk?

A*. Although the standard recommends simple deterministic function, we have recommended not to use simple deterministic input function. The pseudorandom number generators (PRNG)for generating PRF (128) and PRF (256) have been specified in the IEEE 802.11 standard. We have recommended to create Nonce and IV using these PRNG. In our proposed solution, temporal key is the input to the PRNG and the resulting random numbers are recommended to be utilized for Nonce and IV. Hope I could answer this question. Please do let me know if you want me to have amplifying reply.*

Q. While CTR mode is widely accepted, it is known to be well suited to operation on a multi-processor machine where blocks can be encrypted in parallel.Hence, my reservation is that you may have shown improved performance under simulation, it will be hard to demonstrate same success without having technologically superior machines than what we have now. Your comments?

A. The performance has increased because our proposed protocol is a single pass process instead of two pass process. We are also saving computational power by not processing 48 bits packet number with every MPDU. Notwithstanding, It is highly desirable to assess the performance improvement

achieved by proposed security framework when applied to real 802.11 platforms. To this end, there is a need to decide on a suitable vendor who could implement proposed security mechanism while using rest of the paraphernalia of 802.11.

References

[1] Admiral Mike Mullen, Chairman of the Joint Chiefs of Staff Committee, USA, "New Digital Defense Policies", US Homeland Security Department Report, April, 2009. Available at http://www.startribune.com/science/43220562.html

[2] IEEE Std. 802.11 - 2007, Standard for Telecommunications and Information Exchange Between Systems – LAN/MAN Specific Requirements – Part 11: "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 2007.

[3] IEEE Std. 802.11s, 2009 Draft Amendment to Standard for Telecommunications and Information Exchange Between Systems – LAN/MAN Specific Requirements – Part 11: "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Wireless Local Area Mesh Networking Enhancements"2009.

[4] Wireless Security: When Will It Arrive?, CIO Insight, http://www.cioinsight.com/article2/0,3959,394702,00.asp

[5] Brisov, N. Goldberg, I. Wagner, D., "Intercepting Mobile Communications: The Insecurity of 802.11", http://www.isaac.cs.berkley.edu/isaac/wep-draft.pdf

[6] IEEE Std. 802.11i-2004, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Security Enhancements. July, 2004. http://standards.ieee.org/getieee802/download/802.11i-2004.pdf.

[7] IEEE Std. 802.1X-2004, IEEE Standard for Local and metropolitan area networks -Port-Based Network Access Control. June, 2001. http://standards.ieee.org/getieee802/download/802.1X-2004.pdf

[8] IEEE Std. 802.15.1-2005, IEEE Standard for Information technology--Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks--Specific Requirements. Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs),2005.

[9] IEEE Std. 802.16e-2005, IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands (WMAN), 2005.

[10] Junaid, M., Mufti, M., Ilyas, M.U., "Vulnerabilities of IEEE 802.11i Wireless LAN CCMP Protocol", Proceedings of World Academy of Science, Engineering and Technology, Volume 11, pp. 228-233, Czech Republic, February 2006.

[11] Junaid, M., Akbar, M., Mufti, M., "Per Packet Authentication for IEEE 802.11 Wireless LAN", 12th IEEE International Multitopic Conference, pp. 207-212, Karachi, Pakistan, December, 2008.

[12] Junaid, M., Anjum, N., Kanhere, S., "A Novel Reliable and Low Latency Framework for Wireless Mesh Networks", Mehran University Research Journal of Engineering and Technology, Volume 28, No. 3, pp. 263-276, Jamshoro, Pakistan, July 2009.

[13] Junaid, M., Akbar, M., Mufti, M., Kanhere, S., "Piggy Back Challenge Based Security Mechanism for IEEE 802.11i Wireless LAN CCMP Protocol", Technical Report, University of New South Wales, UNSW-CSE-TR-0606, Sydney, Australia, May, 2006.

[14] Umar, M., Mufti, M., Junaid, M., Iqbal, R., Kanhere, S., "INDICT: Intruder Detection, Identification, Containment and Termination", IEEE International Conference on Computing and informatics, Kuala Lumpur, Malaysia, June, 2006.

[15] Keoh, S.L. and Lupu, E., "Towards Flexible Credential Verification in Mobile Ad-hoc Networks", Proceedings of the Second ACM international Workshop on Principles of Mobile Computing, pp. 58-65, Toulouse, France, October 2002.

[16] Kong, J., Zerfos, P., Luo, H., Lu, S., and Zhang, L., "Providing Robust and Ubiquitous Security Support for MANET", Proceedings of IEEE ICNP, pp. 251-260, California, USA, 2001.

[17] Chigan, C., Li, L., Ye Y., "Resource-aware Self-adaptive Security Provisioning in Mobile Adhoc Networks", Wireless Communications

and Networking Conference, Volume 4, pp. 2118-2124, New Orleans, USA, March 2005.

[18] Messerges, T.S., Cukier, J., Kevenaar, T.A.M., Puhl, L., Struik, R., Callaway, E., "A Security Design for a General Purpose, Self-Organizing, Multihop Ad Hoc Wireless Network", Proceedings of the 1st ACM workshop on Security of Adhoc and Sensor Networks, pp.1-11, Virginia, USA, October 2003.

[19] Allen, J., Wilson, J., "Securing a Wireless Network", Proceedings of the 30th annual ACM SIGUCCS conference on User Services, pp. 213-215, Hawaii, USA, November 2002.

[20] Ren, K., Lou, W., Zhang, Y., "LEDS: Providing Location-aware End-to-End Data Security in Wireless Sensor Networks", Proceedings of IEEE International Conference on Computer Communication (INFOCOM'06), Volume 7, issue 5, pp. 585-598, Barcelona, Spain, April 2006.

[21] Perrig, A., Szewczyk, R., Tygar, J. D., Wen V., Culler D.E., "SPINS: Security Protocols for Sensor Networks, Wireless Networks", Wireless Networks, Springer Netherlands, volume 8 , No. 5, pp. 521-534, Netherlands, September 2002.

[22] Park, T., Shin, K.G., "LiSP: A Lightweight Security Protocol For Wireless Sensor Networks", ACM Transactions on Embedded Computing Systems (TECS), Volume 3 , issue 3, pp. 634-660, USA, August 2004.

[23] Soliman, H.S., Mohammed, O., "Application of Synchronous Dynamic Encryption System In Mobile Wireless Domains", Proceedings of the 1st ACM international workshop on Quality of Service and Security in Wireless and Mobile Networks (Q2SWinet '05), Wireless Network Security Session, pp. 24-30, Montreal, Canada, October 2005.

[24] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., Levkowetz, H., "Extensible Authentication Protocol (EAP)", RFC 3748, IETF, USA, June 2004.

[25] Rigney, C., Willens, S., Rubens, A., Simpson, W., "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, IETF, USA, June 2000.

[26] Whiting, D., Housley, R., Ferguson, N., "Counter with CBC-MAC (CCM)", RFC 3610, IETF, USA, September 2003.

[27] U.S. National Institute of Standards and Technology, Random Number Generation and Testing. http://csrc.nist.gov/rng

[28] J.D. Morrison, IEEE WLAN Security Through Location Authentication, MS Thesis, Naval Postgraduate School, California, USA

[29] Jesse Walker, Part I, 802.11 Security Series, Intel's Platform Networking Group.

[30] Brisov, N. Goldberg, I. Wagner, D., "Intercepting Mobile Communications: The Insecurity of 802.11", http://www.isaac.cs.berkley.edu/isaac/wep-draft.pdf

[31] D. A. McGrew and S. R. Fluhrer, "Attacks on Additive Encryption of Redundant Plaintext and Implications on Internet Security", The Proceedings of the Seventh Annual Workshop on Selected Areas in Cryptography (SAC 2000), Springer-Verlag, August, 2000. [Online] Available: http://www.mindspring.com/~dmcgrew/dam-srf-sac00.pdf

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

[33] D. Whiting, R. Housley, and N. Ferguson. "Counter with CBC-MAC (CCM)". RFC 3610, September 2003.

[34] David A. McGrew, "Counter Mode Security: Analysis and Recommendations", Cisco Systems, November, 2002.

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

[36] M.E. Hellman, “A cryptanalytic time-memory trade-off”, IEEE Transactions on Information Theory, July, 1980, pp. 401-406.

[37] D. A. McGrew and S. R. Fluhrer, “Attacks on Additive Encryption of Redundant Plaintext and Implications on Internet Security”, The Proceedings of the Seventh Annual Workshop on Selected Areas in Cryptography (SAC 2000), Springer-Verlag, August, 2000. [Online] Available: http://www.mindspring.com/~dmcgrew/dam-srf-sac00.pdf

[38] A. Biryukov, A. Shamir, D. Wagner , “Real Time Cryptanalysis of A5/1 on a PC”, Proceedings of the Fast Software Encryption Workshop 2000, Springer-Verlag, Lecture Notes in Computer Science, 2000.

[39] Jin Hong, Palash Sarkar, “Rediscovery of Time Memory Tradeoffs”, 2005. [Online] Available: http://cr.yp.to/2005-590/hong.pdf

[40] 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. [Online] Available: http://www.counterpane.com/keylength.html

[41] Moore’s law [Online] Available: http://www.Webopedia.com/TERM/M/Moores_Law.html

[42] Raniwala, A. and Chiueh, T., "Architecture and Algorithms for an IEEE 802.11-based Multi-channel Wireless Mesh Network", proceedings of IEEE InfoCom, Volume 3, pp. 2223-2234, Miami, USA, March 2005.

[43] "Security Architecture for Open Systems Interconnection for CCITT Applications", Recommendation X.800, ITU-T, USA, March 1991.

[44] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., Arkko J., "Diameter Base Protocol", RFC 3588, IETF, USA, September 2003.

[45] Housley, R., "Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)", RFC 3686, IETF, USA, January 2004.