國立交通大學

理學院網路學習學程

碩士論文

以隨機位元認證機制抵禦 802.11 無線網路阻絕式攻擊

Using Random Bit Authentication to Defend IEEE 802.11 DoS Attacks

研究生：簡先得

指導教授：蔡文能

中華民國九十五年五月

以隨機位元認證機制抵禦 802.11 無線網路阻絕式攻擊

Using Random Bit Authentication to Defend IEEE 802.11 DoS Attacks

研 究 生：簡先得　　Student：Hsien-Te Chien

指導教授：蔡文能　　Advisor：Wen-Nung Tsai

國 立 交 通 大 學

理學院網路學習學程

碩 士 論 文

A Thesis

Submitted to Degree Program of E-Learning

College of Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Degree Program of E-Learning

May 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年五月

授權書

(博碩士論文)

本授權書所授權之論文為本人在國立交通大學(學院)網路學習研究所九十四學年度第二學期取得碩士學位之論文。

論文名稱：以隨機位元認證機制抵禦 802.11 無線網路阻絕式攻擊

1.□同意　　□不同意

本人具有著作財產權之論文全文資料,授予行政院國家科學委員會科學技術資料中心、國家圖書館及本人畢業學校圖書館，得不限地域、時間與次數以微縮、光碟或數位化等各種方式重製後散布發行或上載網路。

本論文為本人向經濟部智慧財產局申請專利的附件之一,請將全文資料延後兩年後再公開。(請註明文號:　　)

2.□同意　　□不同意

本人具有著作財產權之論文全文資料,授予教育部指定送繳之圖書館及本人畢業學校圖書館，為學術研究之目的以各種方法重製，或為上述目的再授權他人以各種方法重製，不限地域與時間，惟每人以一份為限。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未鈎選，本人同意視同授權。

指導教授姓名：

研究生簽名：　　學號：

（親筆正楷）　　（務必填寫）

日期：民國　　年　　月　　日

1.本授權書請以黑筆撰寫並影印裝訂於書名頁之次頁。

2.授權第一項者,所繳的論文本將由註冊組彙總寄交國科會科學技術資料中心。

3.本授權書已於民國 85 年 4 月 10 日送請內政部著作權委員會(現為經濟部智慧財產局）修正定稿。

4.本案依據教育部國家圖書館 85.4.19 台(85)圖編字第 712 號函辦理。

國 立 交 通 大 學

論文口試委員會審定書

本校 理學院網路學習學程 碩士班 簡先得 君

所提論文 以隨機位元認證機制抵禦 802.11 無線網路阻絕式攻擊

Using Random Bit Authentication to Defend IEEE 802.11 DoS Attacks

合於碩士資格標準，業經本委員會評審認可。

口試委員: 周勝鄰

莊祚敏

蔡文能

指導教授: 蔡文能

班 主 任: 莊祚敏

中 華 民 國 九十五 年 五 月 三十 日

以隨機位元認證機制抵禦 802.11 無線網路阻絕式攻擊

學生：簡先得　　指導教授：蔡文能

國立交通大學網路學習學程碩士在職專班

摘　　要

IEEE802.11(a,b,g)無線網路方便佈建的方式與便宜的價格，已使 802.11(a,b,g)無線網路普遍的建置在家庭、學校、民間企業、政府機構及公共場所。然而無線電波的特性，使無線網路與傳統的有線網路，多了許多安全性的考量。

WEP 是 802.11(a,b,g)無線網路的安全性機制，早被證實存在許多弱點且容易被破解，WPA 及 802.11i 是 802.11(a,b,g)無線網路安全性的加強版。802.11i 改善了 802.11(a,b,g)無線網路資料傳送的完整性(integraty)及可信性(confidentiality)，但在可用性(availability)卻沒有嚴謹的考量與設計，因此使得 802.11 系列的無線網路，容易遭受阻絕性攻擊(Denial of Service attack)。

本研究即利用 802.11 在 MAC 層的封包標頭結構，以共有金鑰的假設下，在認證（(de)authentication）及連結（(dis)association）封包中，以隨機方式加入 3 到 4 個位元，作為無線網路存取點（AP）和工作端（STA）的雙方溝通的認證機制，配合 MAC 層封包標頭中的 Sequence Counter 欄位連續數質的特性，設計有效過濾偽造的阻絕式攻擊封包的機制。

本研究設計的抵禦無線網路阻絕式攻擊機制，經實作與模擬實驗後證明我們所設計的隨機位元認證機制，能有效的抵禦 802.11 無線網路阻絕式攻擊。

Using Random Bit Authentication to Defend IEEE 802.11 DoS Attacks

Student: Hsien-Te hien Advisor: Wen-Nung Tsai

Degree Program of E-Learning College of Science

National Chiao-Tung University

ABSTRACT

IEEE 802.11 network is prevailing, but the security issue is an important concern.

WEP is the security mechanism in 802.11 specification. It has been proved that WEP is vulnerable and easy to be cracked. 802.11i is the enhanced version of security for 802.11 networks. The 802.11i focuses on integrity and confidentiality of transmitting data. The availability of 802.11 network is not considered properly. The management frames of 802.11 are not protected by any key based authentication. It causes the 802.11 network vulnerable to Denial of Service attacks.

We designed a so called random bit authentication mechanism to defend Denial of Service attacks against 802.11 networks. We replace some unused bits in the MAC header of the 802.11 management frames with some authentication bits. The AP and STA can authenticate each other according to these authentication bits. We also exploited the characteristic of Sequence Number field in MAC header of the 802.11 frames to design an effective mechanism to filter out attacking frames.

In our implementation and experiments, it shows that our two-phase filtering mechanism is effective and lightweight to defend IEEE 802.11 Denial of Service attacks.

誌謝

本研究論文能夠完成，最先要感謝的是我的指導教授－蔡文能老師。常常在研究或實驗中，腸枯思竭的時候，老師都能於關鍵問題上，猶如畫龍點睛般的指引。在學習過程中，體會老師嚴謹的治學精神與態度。實驗完成撰寫論文時，老師時時的鼓勵，使我不敢懈怠的努力寫作，才使本論文得按步就班的付梓。以學生觀察，老師幾乎就是以校為家，致力於教學與研究工作。在小組論文研討中，常常和老師研討到深夜，老師的專業與敬業的精神，令人佩服。

工研院電通所副所長周勝鄰博士與網路學習專班主任莊祚敏博士，在口試時的指導，也啟迪學生不少重要的觀念，在此一併致謝。

也感謝同研討室的明傑兄、威德兄、文彬兄與典龍學弟，大家雖各有家庭及事業，還能共聚一堂，彼此學習與提攜，是不可多得的經驗。

當然最當感謝的是靜怡，使我在家庭與二個 Baby 的照顧上無後顧之憂。每週或每隔週至少一次的往返台中與新竹間的奔波，如此歷經四年，如果沒有內人的鼓勵與支持，這學業難以完成。

對於家中的二個小 Baby 也有些歉意，為了完成論文寫作，假期間常常無法陪她們一同出遊，這小小的遺憾也只待日後慢慢的來彌補。

Contents

List of Tables

List of Figures

Chapter 1 Introduction

Wireless Local Area Networks (WLANs) are popularly and widely deployed in public places, companies, and homes. Many goverments also deployed WLANs in the metropolitan areas to offer citizens to access information via the mobile devices. While WLANs are prevailing, wireless security has become an importment issuse. If one carelessly deploys the WLANs, the important data of the corporations or individuals may be leaked.

Since 802.11i standard was ratified on June 24, 2004, many of the vulnerabilities of Wired Equivalent Privacy (WEP) were fixed. However, security problems still remain. Wireless network are inherently diffirent from wired network. Evil nodes within the range of a certain wireless network are as capable as the legal nodes in receiving radio signals. Hence, the frames of WLANs are easily sniffed, intercepted, forged and blocked by the attacker. It is thus an important issue to defend the various attacks when designing the security mechanisms for WLANs.

1.1 Research Motivation

The vulnerabilities of WEP are well known. WEP is the security scheme defined in IEEE 802.11. The security scheme of 802.11 has been modified for many times, either by the IEEE institute or by others. Wi-Fi Protected Access (WPA) was proposed by Wi-Fi alliance to solve the vulnerability problem of WEP. Being compatible to prevailing 802.11 devices, WPA still contains some security weakness. IEEE 802.11 task group I amended 802.11 standard with 8002.11i specification to enhance the security.

WLANs basically focused on implementing three security services:

confidentiality, integrity and availability. Some researchers found that 802.11i only concentrated on addressing the issues of confidentiality and integraty, but not on the availability. 802.11i appears vulnerable to DoS attacks even when RSNA is implemented [6].

Launching DoS attacks against 802.11(i) is easy. Attacking tools, such as void11 are readily obtainable from the Internet. The tools are easily installed on common devices equipped with 802.11 wireless cards. An attacker with only median skills can therefore can launch DoS attacks to block or slow down the WLAN network. One may argue that the DoS attacks are inevitable, because of the inheretant characteristics of wireless network. The idea in solving such an issue is to impose relatively higher cost for an adversary to mount these attacks.

Wireless devices usually have only limited computing power and bandwidth. A malicious attacker launching a great number of attacking frames can easily exhaust the limited resources. Probe, authentication and association request frames can be used to flood the network to cause a traffic jamming DoS attack.. On the other hand, some complex encryptional and decryptional algorithms may lead the computing resource-exhausted vulnerabilities which are easily exploited by the attacker to launch DoS attacks.

Certain vulnerabilities are inherent in the design of protocol. The most common exploition are the deauthentication and disassociation flooding attacks. 802.11 network use management frames to establish transmitting sessions. However these frames are not protected by any key-based authentication. Also they are transmitted in the clear. They are easily captured, spoofed and blocked by the attackers.

Our research has three purposes:

First of all, we want to design a mechanism to defend deauthentication and disassociation flooding based DoS attacks against 802.11 WLANs. Secondly, we want

the mechanism to be backward compatible. Finally, we want the mechanism to be efficient and lightweight.

1.2 Thesis Organization

This thesis is organized as follows. Chapter 2 describes background materials, such as the diffirence between the wireless and wired networks, the evolution of the 802.11 security issues and the concepts of DoS attacks. Chapter 3 briefly discusses related works. In Chapter 4 we present our proposed protocol that defends deauthentication and association flooding attacks. Chapter 5 shows our experimental results in detail. Chapter 6 presents the conclusion and future work.

Chapter 2 Background

Wireless networks have characteristics that are different from the wired networks. These differences introduce many security issues that must be considered, or the wireless network can be exposed to security risks

IEEE 802.11 standard defined wired equivalent privacy (WEP) algorithm to protect the data transmitted over wireless networks. It had been proven that WEP has insufficient security. To replace WEP, Wi-Fi alliance proposed Wi-Fi Protected Access (WPA). As a long term solution, IEEE 802.11 TGi approved 802.11i specification to address 802.11 security issues.

802.11 networks are vulnerable to Denial of Service (DoS) attacks. A DoS attack is an attack against a computer system or network that results in a loss of services to legitimate users. Denial of Service attack not only disables your computer system or networks, but more seriously, it effectively disrupts the normal operation of your organization.

2.1 Overview of Wireless Local Area Network (WLAN)

2.1.1 Characteristics of Wireless LAN

Wireless networks have inherent characteristics that are significantly different from traditional wired LANs [2].

The most distinguishing characteristic is that radio transmission has no definite boundaries. When we set up wireless equipments, we cannot predict with any certainty where the message transmitted may eventual arrive. A transceiver placed near a wireless network by an evil adversary can potentially receive messages from or insert

messages into the network.

Furthermore, the signals transmitted wirelessly are significantly less reliable than the signals transmitted through wired networks. Frames are much more likely to be lost or be distorted by noise interference.

Besides, IEEE 802.11 PHYs (physical layer) lack full connectivity. Therefore, we cannot make the normal assumption that every STA (station) can hear every other STA. For example, some STAs may be hidden from others.

Last but not the least, IEEE 802.11 is designed to handle mobile as well as portable stations. Portable stations can be moved from location to location, yet can only be used in fixed locations. On the other hand, mobile stations can access the WLAN while in motion, and are often battery powered. Hence, power management is an important consideration for mobile stations [2].

We must understand the inherent characteristics of a wireless network when considering its security issues. We must design the security measures to protect the messages from being intercepted, forged, interfered, or blocked.

The system availability is threatened by the peculiar features of 802.11 standard, since they expose the wireless networks to more DoS attacks than wired networks. The wireless medium allows a malicious station or a directive antenna inside the range of a wireless network to launch an attack that blocks any legitimate communications [13].

2.1.2 Evolution of 802.11 network

Originally, the 802.11 standard was ratified by IEEE-SA in 1997. Now, 802.11 is a set of IEEE standards that govern wireless networking transmission methods. Today, versions like 802.11a, 802.11b, and 802.11g are commonly used to provide wireless connectivity at homes, offices, and commercial establishments. The draft of 802.11n is underway. When accepted, it will provide a new amendment to the 802.11 standard

that boost the throughput of wireless LANs.

a. 802.11

The original version of the IEEE 802.11 standard was released in 1997. The standard specifies that the data can be transmitted via infrared signals in raw data rates of 1 or 2 Mbps; or the data can be transmitted using the Industrial Scientific Medical frequency of 2.4 GHz. It also defines Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) as the media access method.

The original 802.11 standard is really a “meta-specification” than a rigid specification. Individual product vendors are given the flexibility to add innovations that could enrich their products. Legacy 802.11 devices are rapidly supplemented (and popularized) by 802.11b modifications. Widespread adoption of 802.11 networks only began after 802.11b was published, and as a result, few networks follow the 802.11-1997 standard [27]. However, 802.11-1997 standard acts as the base of the following 802.11 standard family.

b. 802.11b

The 802.11b was amended to the original 802.11 standard in 1999. The maximum raw data rate is increased to 11 Mbps and also uses the same CSMA/CA media access method as the original standard.

802.11b products appeared on the market very quickly. The dramatic increase in data throughput (compared to the original standard) along with the substantial price reduction both contribute to the rapid acceptance of 802.11b as the definitive wireless LAN technology.

A point-to-multipoint configuration is usually used by an 802.11b network. In this configuration, a single AP communicates with one or more clients (STAs) located inside the coverage area of the AP. The throughput rate varies inversely with the distance between the AP and the STA [2].

Even though 802.11b network cards can operate at the rate of 11 Mbps, they will scale back to the rate of 5.5, 2, or 1 Mbps. Extensions have been proposed to increase the speed to 22, 33, and 44 Mbps, but the extensions are proprietary and not yet endorsed by IEEE. Many companies call their enhanced versions as “802.11b+”.

c. 802.11a

The 802.11a was amended to the 802.11 family in 1999. The 802.11a standard uses the same core protocol as the original standard, but the operating frequency is changed to 5 GHz. It also employs the orthogonal frequency division multiplexing (OFDM) with a maximum raw data rate of 54 Mbps.

If required, the data rate can be reduced to 48, 36, 24, 18, 12, 9 or even 6 Mbps. In 802.11a standard, 12 non-overlapping channels are utilized. Eight of the channels are dedicated to indoor communication, and the other four channels are reserved for point-to-point communication. In this respect, 802.11a is not interoperable with 802.11b.

Since the 2.4 GHz band is heavily used, using the 5 GHz band gives 802.11a the advantage of having less interference. However, the higher carrier frequency also bears a disadvantage. It restricts the use of 802.11a to almost line of sight. This means that 802.11a cannot penetrate as far as 802.11b [27].

d. 802.11g

The 802.11g was the new addition to the 802.11 family in June 2003. It has data rates up to 54 Mbps, and works in the 2.4 GHz band like 802.11b. 802.11g hardware is compatible with 802.11b hardware. However, the presence of an 802.11b STA significantly reduces the speed of an 802.11g network.

Identical to 802.11a standard, the modulation scheme used in 802.11g is orthogonal frequency division multiplexing (OFDM) with data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mbps [27].

Since the crowded 2.4 GHz frequency band is also used, 802.11g suffers the same interference problems present in a 802.11b network. Devices operating in this frequency range include microwave ovens, Bluetooth® devices, and cordless telephones. Those devices can emit signals that would interfere with the 802.11g transmission.

e. 802.11n

The IEEE 802.11n standard development project began in 2003. The standard committee expects to complete the draft in late 2006, and plans to publish the 802.11n amendment to the 802.11 standard in 2007 [28].

802.11 Task Group n (TGn) was formed by IEEE to be responsible in developing a new amendment to the 802.11 standard. The actual data throughput attained is estimated to reach a theoretical 600 Mbps [28]. It is projected that 802.11n will also offer a better operating distance than current networks.

802.11n builds upon previous 802.11 standards and adds MIMO (multiple-input multiple-output) feature. MIMO uses multiple transmitter and receiver antennas to increase data throughput by incorporating spatial multiplexing and to increase range by exploiting the spatial diversity.

2.2 IEEE 802.11 security issues

The IEEE 802.11 security issues were discussed fierily and criticized seriously. After some papers presented the weakness of 802.11 security and shoed the way to crack WEP. Some solutions, including short term and long term, were published. The Wi-Fi Alliance proposed WPA security solution for the vulnerabilities of 802.11 before 802.11i standard was approved. To enhance the security of 802.11, the IEEE 802.11i task group was formed, and approved the 802.11i standard on June 24, 2004. Though 802.11i specifies the rigorous organism to protect the propagating packets,

some secure issues like DoS attacks, are not solved.

2.2.1 WEP

According to the 802.11 standard, authentication and privacy services are provided to bring the IEEE 802.11 functionality in line with wired local area networks. Authentication is used instead of the wired media physical connection. Privacy is used to provide the confidential aspects of closed wired media. Access control and confidentiality services of 802.11 standard are also important security services [2].

To control access to the unrestricted radio medium, IEEE 802.11 provides two modes of authentication services: Open System and Shared Key.

Open System authentication is a null authentication algorithm. Any STA requesting authentication using this method may become authenticated. Shared Key authentication requires STAs to agree on a common shared key before the authentication service can be used. In infrastructure architecture, the AP is the authenticator, and the other STAs are authenticated by the AP. This authentication scheme is only available if the WEP option is implemented.

Shared Key authentication accomplishes its task without transmitting the secret key over the air. Upon receiving an authentication request, the AP sends a challenge text for the STA to encrypt with the shared secret key using WEP. Then, the AP decrypts the encrypted response packet to match the original challenge text. If there is a match, the authentication succeeds [2]. Please refer to Figure 1 for detail.

Figure 1 Shared Key authentication scheme

No matter which authentication method is used, either sides of the communicating parties can send deauthentication notification to cease the session. For an AP, the deauthentication notifications are broadcasted to all authenticated STAs to stop communicating with them. The deauthentication frame is send in clear, and no key-based authentication is used. This makes forging the deauthentication frame easy.

If a STA is authenticated by some AP, the STA then has to become associated with the authenticating AP to be allowed to send data messages through the AP. The association service provides the STA-to-AP mapping, and this mapping is delivered to the distribution system (DS). The DS will use this information to accomplish message distribution services.

The disassociation service is invoked whenever an existing association is to be terminated. Either the STA or AP may initiate the disassociation process. To do so, one sends a disassociation message to the other, then the disassociation succeeds. The disassociation frame is transmitted in the clear and has similar problems as the deauthentication frame.

The (de)authentication and (dis)association processes are illustrated in Figure 2 as a state transition diagram.

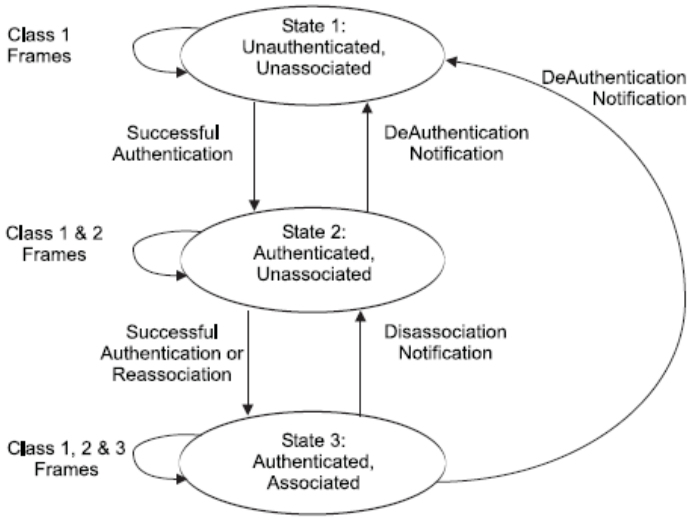


Figure 2 Relationship between state variables and services [2].

As shown in Figure 2, any STA and AP must follow the state machine specified in the IEEE 802.11 standard. A successfully associated STA stays in State 3 in order to continue wireless communication. In State 1 or 2, a STA cannot participate in the WLAN data communication process until it is again authenticated and associated.

IEEE 802.11 specifies a data confidentiality algorithm that hopefully provides the level of secrecy comparable to that provided in a wired LAN. The algorithm, named WEP, protects authorized users of a wireless LAN from casual eavesdropping. This service is intended to provide functionality for the wireless LAN equivalent to that provided by the wired medium [2].

The standard document claims that the WEP algorithm is “reasonably strong”. The security of the algorithm, as stated in the standard, relies of the difficulty of discovering the secret key through a brute-force attack. Therefore, the security can be enhanced by increasing the length of the secret key and the frequency for changing the IV. Please refer to [2]. for detailed WEP algorithm description.

WEP does not include any key management protocols; the pre-shared key must be fed into devices manually. The AP and the associated STAs share the same key. If the shared key is leaked, the WEP security mechanism is cracked. Since WEP keys are

changed manually, they would be changed infrequently. However, this increases the danger of being sniffed and cracked.

Jesse R. Walker noted that it is infeasible to achieve privacy using WEP encapsulation, even if the key size is expanded from 40 bits to 104 bits. He presented an attack against WEP and demonstrated that the attack will succeed regardless of the key size or the cipher used. Also, the attack can be implemented easily [4].

Fluhrer, Mantin, and Shamir described several attacks on RC4 algorithm used in WEP [10]. They found that an eavesdropper, who can obtain several million encrypted packets with a known first byte of plaintext, would be able to deduce the RC4 key by exploiting properties of the RC4 key schedule. Their passive ciphertext-only attack can recover an arbitrary long key in a short amount of time for any key lengths, even when 24 and 128 bit IV modifiers are added.

Stubblefield, Ioannidis, and Rubin experimentally implemented the “F.M.S.” attack using off-the-shelf devices, and demonstrated that real systems could be cracked in several hours [5]. They also improved the RC4 attack implementation with some optimizations. Fluhrer et al. speculated that around 4,000,000 to 6,000,000 packets would be sufficient to successfully attack RC4, but Stubblefield’s improvement dropped the number to around 1,000,000 packets. They concluded that 802.11 WEP was totally insecure.

Once the WEP vulnerabilities were publicized, tools like Airsnort [16]. and WEPCrack [17]. emerged and they enabled anyone having popular 802.11 devices to sniff 802.11 packets and discover the key in a short time.

2.2.2 WPA

Due to the poor access control and weak WEP privacy in the 802.11 standard, the Wi-Fi Alliance devised Wi-Fi Protected Access (WPA) to enhance the security of a

802.11 network.

WPA is an intermediate solution. It replaces WEP with Temporal Key Integrity Protocol (TKIP). TKIP is a compromise on achieving strong security while still using existing hardware. It continues the use of RC4 as the encryption algorithm. However, a keyed packet authentication mechanism (called Michael) is implemented to guard against replay attacks.

To provide access control and key management, WPA can either use an external authentication server (e.g., RADIUS) and EAP similar to that used in IEEE 802.1x, or it can use pre-shared keys without additional servers. WPA implements a new key handshake process (4-Way Handshake) for generating and exchanging data encryption keys between the Authenticator and the Supplicant. This handshake is also used to verify that both the Authenticator and the Supplicant know the master session key.

When a station would like to use the services of an AP, the station will first perform an IEEE 802.11 authentication. Open system authentication is used in this case, so there is no security. After this, IEEE 802.11 association is performed. If 802.11x is configured to be used, the virtual port of the station is set in the unauthorized state and this port can now accept only IEEE 802.11x frames. The Authenticator will then ask the Supplicant to authenticate itself with the Authentication Server. After the Supplicant is authenticated successfully, the virtual port is set to the authorized state, and any frames from and to the station are accepted.

TKIP is proposed to address known vulnerabilities of WEP, and it does enhance the security. However, weakness is predestined since the appearance of WPA due to the limitation imposed by its re-use of legacy hardware [6].

2.2.3 IEEE 802.11i

IEEE 802.11i [1]., ratified on June 24, 2004, is designed to provide enhanced

security in the Medium Access Control layer for 802.11 networks. The 802.11i specification defines two classes of security algorithms: Robust Security Network Association (RSNA), and Pre-RSNA. Pre-RSNA is the old security mechanism discussed in section 2.2.1.

RSNA provides two data confidentiality protocols: the Temporal Key Integrity Protocol (TKIP) and the Counter-mode/CBC-MAC Protocol (CCMP). The TKIP protocol has already been discussed in section 2.2.2. It is designed to be backward compatible to older 802.11 devices.

802.11i RSNA establishment procedure makes use of 802.1x authentication and key management protocols. The complete handshakes of establishing a RSNA are shown in Figure 3. These steps can be divided into 6 stages as follows [6].

Stage 1. Network and Security Capability Discovery

Stage 2. 802.11 Authentication and Association

Stage 3. EAP/802.1x/RADIUS Authentication

Stage 4. 4-Way Handshake

Stage 5. Group Key Handshake

Stage 6. Secure Data Communication

For our research purposes, we must keep an eye on Stage 2. The deauthentication service is invoked when an existing Open System authentication is to be terminated. In an ESS, because authentication is a prerequisite for association, deauthentication also causes the station to be disassociated. In an RSN ESS, Open System authentication is required. Deauthentication results in the termination of any association for the deauthenticated station. It also results in the disabling of the IEEE 802.1x Controlled Port for that STA and the deletion of the pairwise transient key security association (PTKSA). It also destroys the pairwise master key security association (PMKSA) from which the deleted PTKSA was derived [1].

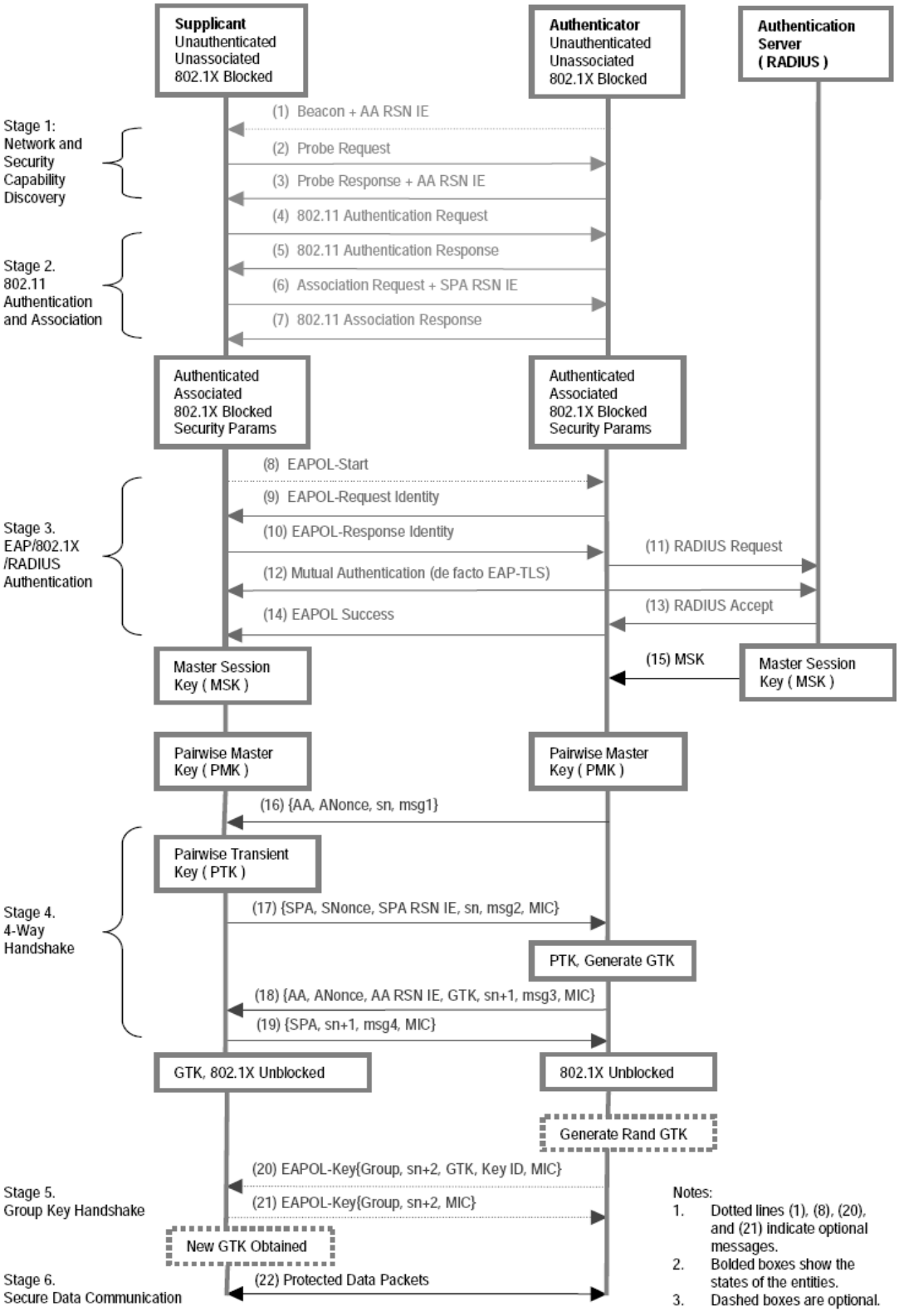


Figure 3 RSNA Establishment Procedures [6]

2.2.4 802.11w

Although 802.11i addresses the security of data frames, the unprotected management frames still leave wireless networks vulnerable to malicious attacks.

The IEEE-SA Standards Board had approved the 802.11w project on March 20, 2005 [25]. The timeline of the project is expected to end in December, 2009. The goal of 802.11w is to strengthen the IEEE 802.11 Medium Access Control layer to provide appropriate mechanisms that enable data integrity, data origin authenticity, replay protection, and data confidentiality for selected IEEE 802.11 management frames. The Task Group responsible for developing IEEE 802.11w focuses on increasing the security of IEEE 802.11 management frames.

The 802.11w project is still in progress. As of March 2006, P802.11w D1.0 was send to Working Group letter ballot [26].

2.3 Denial of Service attacks

Denial of service attacks can paralyze your computer systems or networks. More seriously, they can easily disconnect your organization from the rest of the world. Some denial of service attacks can be executed against a large web site, a network device or a communication session using only limited resources.

2.3.1 Definition of Denial of Service attacks

A denial of service (DoS) attack is characterized as an explicit attempt by attackers to prevent legitimate users of a service from using that service [29]. For example, it includes:

a. One attempts to “flood” a network, and thereby blocks legitimate network traffic.

b. One attempts to disrupt connections between two machines, and thereby prevents one machine to access a service on the other.

c. One attempts to prevent a particular individual from accessing a service.

d. One attempts to disrupt a service to a specific system or a person [29].

In other words, a denial of service attack is an attack against a computer or a network that causes the loss of services to users. Typically, the DoS attacks disrupt network connectivity and services by consuming the bandwidth of the victim network or overloading the computing or memory resources of the victim system. When DoS attacks are mounted, the victim system cannot operate normally to provide its services.

2.3.2 Types of Denial of Service attacks

Denial of Service attack comes in a variety of forms and aims at a variety of services. We simply describe some typical types of DoS attacks here.

a. Consumption of system resources

Computers and networks need certain resources to work normally. These resources include CPU time, memory and disk space, etc.

Denial of Service attack is most frequently executed against network connectivity. The goal is to prevent hosts or networks from communicating on the network. An example of this type of attacks is the "SYN flood" attack.

In SYN flood attack, the attacker attempt to establish a connection with the victim machine, but it does not really complete the connection. By flooding SYN frames, the attackers can exhaust the resources of the victum node.

We should note that this type of attacks does not rely on the attacker being able to consume the network bandwidth. In 802.11 networks, the probe request flooding, authentication request flooding and association flooding attacks are such kinds of DoS attacks.

b. Bandwidth Consumption

An attacker may also be able to consume all the available bandwidth on your network by generating a large number of packets directed to the victim network. Typically, the easist way is to use ICMP ECHO packets. The “Ping flooding” is one such attack. The idea is to simply flood the victim with so much ping traffic that normal traffic will fail to reach the aimed node.

Further, the attackers need not be operating from a single machine. He may be able to coordinate several machines on the same or different networks to achieve the DoS attack. We usually call this form of DoS attack Distributed Denial of Service attack (DDoS) [30].

c. Exploit of the unsophisticated protocol

Some protocols contain vulnerabilities that can be exploited by the attackers to launch DoS attacks. The networks based on IEEE 802.11 contain many inheret vulnerabilities to be exploited to launch DoS attacks. The deauthentication and deauthentication flooding attacks both belong to this category. The unbound radio signals are easily captured and spoofed by the attacker. Many management and control frames of 802.11 are not protected by any key-based authentication.

In addition to the deauthentication and deauthentication flooding attacks, the power saving mode attack is a similar attack. Since the control messages in 802.11 are not protected or authenticated, an attacker spoofs the polling message on behalf of the STA, and cause the access point to discard the STA packets while the STA is asleep in power saving mode. On the other hand, it is possible to trick a STA into believing that there are no buffered packets at the AP [13].

Virtual carrier sense DoS attack is another vulnerability of the 802.11 network. It is also called the duration attack. WLAN devices perform virtual carrier sensing prior to accessing the medium. This mechanism is designed to reduce frame collisions and

prevent the hidden node problem. The virtual carrier-sense function is based on the Network Allocation Vector (NAV). IEEE 802.11 MAC frames carry a duration field, which is used to reserve the medium for a certain period of time. The NAV is a timer that indicates the time for which the medium has been reserved. Transmitting nodes set the NAV to the time for which they expect to use the medium. Other nodes set up a mechanism to count down the value of NAV. When the NAV is greater than zero, the virtual carrier-sense function indicates that the medium is busy.

However, an attacker can send frames with huge duration values. This would force other nodes in the range to wait till the NAV value reaches zero. If the attacker successfully sends such continuous packets, then it prevents other nodes from accessing the media, and therefore results in DoS attacks [3][13].

802.11i is the enhanced security amendment of 802.11. 802.11i contains some unique vulnerabilities of DoS attacks. The 4-Way Handshake block and RSN IE Poisoning attacks etc., are detailed in [6][15].

The weakness of 802.11 is well known. The 802.11 specification about the security is modified many times as we described in section 2.2.

Chapter 3 Related work

In order to analyze the 802.11 protocol and its security issues, it is important to characterize the possible capabilities of any adversaries. For the Link Layer of a WLAN, there are three possible types of frames: Management Frame, Control Frame, and Data Frame. Any illegal manipulation of these frames may be exploited by the adversary who creates a security threat.

WLAN systems are quite vulnerable to DoS attacks. Because of the inherent unshielded characteristics of wireless network, an adversary may launch DoS attacks in several ways. For example, by forging the unprotected management frames, exploiting some protocol weaknesses, or jamming the frequency bandwith, the services to legitimate clients will be denied.

However, we only consider DoS attacks that require reasonable effort on the part of an adversary. For instance, the commodity devices like prevailing Wi-Fi cards, PDAs and laptops consume little energy and can easily be bought from the markets. Such devices can be exploited by commom users. Some DoS attacks using special devices and consuming considerable resources would not be discussed in our thesis.

3.1 DoS attacks against 802.11 network

The 802.11 MAC layer incorporates functionalities uniquely designed to be used in the specific wireless networks. In particular, these include the ability to discover radio networks, to join and leave networks, and to coordinate access to the radio medium.

Many of the vulnerabilities are resulted from the unprotected management and control frames. Some adversaries exploit the 802.11 specific mechanisms like

CSMA/CA to launch DoS attacks.

Among the 802.11 DoS attacks, the most efficient way is to forge and send deauthentication or disassociation frames repeatedly to the victims. It is not difficult to setup such kinds of DoS attacks. Many commodity 802.11 devices can be exploited. More seriously, it is easy to download the attacking softwares from the Internet and to install on devices.

3.1.1 802.11 Deauthentication and Disassociation flooding attacks

After an 802.11 STA has selected an AP to use for communication, it must first authenticate then associate itself to the AP. When the STA or the AP wants to stop the communication, the STA or AP can send a deauthentication message to the other side and disconnect the communication. Unfortunately, this deauthentication message itself is not protected by using any key material. Consequently the attacker may spoof this message, by pretending to be the AP or the STA, and then sends the spoofed message to the other node. Figure 4 illustrates the scenario of the deauthentication and disassociation attacks.

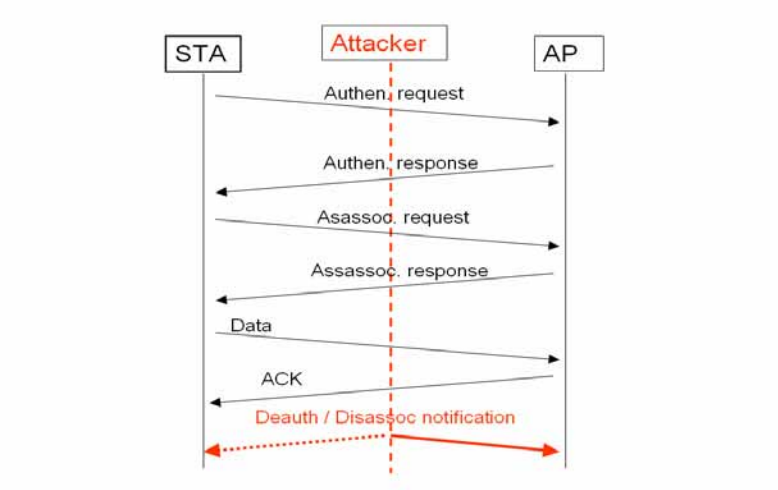


Figure 4 Graphical depiction of the Deauth and Disassoc attacks [13]

When the AP or STA receives the deauthentication notification, it will leave the

authenticated state until authentication is reestablished. If the attacker keeps sending deauthentications frame to the victim, the victim may be prevented from transmitting or receiving data until the attack stops.

The attacker can pretend to be an AP to broadcast deauthentication or disassociation frames to all the STAs that were authenticated by the legal AP, and forces them to stop the communication. Alternatively, the attacker can pretend to be a legal STA to send deauthentication or disassociation frames to the AP and disconnect the communication between the AP and the legal STA.

J. Bellardo and S. Savage implemented successfully the attacks described above. They suggested that the vulnerabilities can be solved directly by explicitly authenticating management frames. However, the 802.11 standard is still some ways off and it is clear that legacy 802.11 devices do not have sufficient CPU capacity to implement this functionality as a software upgrade [13]. Therefore, to defend such attacks with low-overhead designs can still offer significant value.

To defend the deauthentication and disassociation attacks, J. Bellardo and S. Savage designed one mechanism that delays the effects of deauthentication or disassociation requests (e.g., by queuing such requests for 5-10 seconds). The AP or STA has the opportunity to observe subsequent packets from the the other corresponding node. If a data packet arrives after a deauthentication or disassociation request is queued, that request is discarded, since a legitimate node would never generate packets in that order [13]. They implemented their mechanism to defend deauthentication attacks successfully when there are the FTP sessions between the communicating nodes.

There are some drawbacks, as J. Bellardo and S. Savage metioned. Such defending mechanism delays the handoff processes when the mobile STA roams between APs and opens up a new vulnerability when mobile clients roam between

access points [13].

If the STA sends the deauthentication or disassociation frame to the AP, but the frame is queued and delayed for 5-10 seconds, the session hijacking attacker may exploit the delaying period to send forged data frames to the AP and void the real deauthentication frame. The hijacking attacker will successfully connect to the AP without being detected.

On the other hand, FTP applications are not always used, and other applications like the web browser and email are used more frequently. When the user browses the web, the browser does not always downloand the webpage during the connecting phase. When the user reads the webpage, the network usually idles more than 10 seconds. If the deauthentication attack happened at this time, it would succeed disconnecting the network. Then the STA will reauthenticate itself with the AP. This situation may happen frequently, and bothers the users who browse a certain website. It is similar to the DoS attacks, though the network probabily is not blocked thoroughly.

3.1.2 Traffic Jamming DoS attack

In our thesis, the traffic jamming DoS attack means the attack that exhausts resources of the devices (e.g. AP), and hinders the devices to communicate with other legal nodes. With a request-respond model, the management frames of the 802.11 seem the most suitable to be exploited for this type of attack. Any management frames sent to a certain node (e.g. AP) consumes some computing or memory resources.

We discuss probe, authentication and association request flooding attacks in this section. We do not discuss the bandwidth consuming traffic jams caused by common abundant data transmissions.

An AP can easily become a bottleneck for the entire network in infrastructured network. If an AP failed, the entire network blocks on the condition that no other APs

are deployed besides the failed AP.

Ferreri, F. et al., implemented probe, authentication and association request flooding attacks againt 4 models of APs: Enterasys RoamAbout R2, Netgear ME102, 3Com AP 8000 and Host AP. They developed a simple application, named wfit (wireless frame injection tool), based on the Radiate library which is built on top of an old version of the HostAP driver [14]. They exploited this attack tool to launch probe, authentication and association request flooding attacks by spoofing MAC addresses of attacking frames.

We summarize their experimental results as shown in Table 1.

Attack Type / AP Model	Probe Request Flood	Authentication Request Flood(open system)	Authentication Request Flood(WEP enable)	Association request flood
Enterasys RoamAbout R2	DoS	performance degradation	performance degradation	performance degradation
Netgear ME102	No effect on performance	Can withstand the attacks	AP crashed each time	DoS
3Com AP 8000	DoS	DoS	DoS	No relevant effects
Host AP	DoS	Similar to Enterasys RoamAbout R2 AP		

Table 1 probe, authentication and association request flooding attacks against APs [14].

Note: DoS attacks are to exhaust the AP resources, and the communication among legitimate clients becomes impossible.

According to their experiments, they concluded:

a. Such attacks can be executed by a malicious station without being neither associated nor authenticated with the AP.

b. AP's main vulnerability to these flooding attacks seems to reside in unacked frame retransmissions, which cause memory buffer exhaustion and freeze up the AP.

c. Weak implementations of the 802.11 protocol in APs can bring about further vulnerabilities that allow malicious stations to crash an AP [14].

Although they had attempted to design a detection and defense mechanism and tried to embed it in the Linux HostAP, the vulnerabilities of probe flooding request flooding attack could not be mitigated at the software driver level. The AP vulnerabilities to those DoS attacks were at the firmware level [14]. Ferreri, F. et al., did not suggest better defensive solutions in their paper.

There are other Denial of Service attacks such as virtual carrier sense DoS attack [3], power saving mode attack, and more [14]. We have discussed them in section 2.3.2, amd do not detail them here.

3.2 DoS attacks against 802.11i network

IEEE institute claims that it deploys the strongest data confidentiality and authentication protocol with 802.11i, but it seems not to emphasize the availability as a primary objective. The result makes 802.11i vulnerable to DoS attacks [6].

Changhua He et al. found that DoS vulnerabilities in 802.11i appear to be more severe than 802.11. An adversary can launch an 802.11i attack much more easily with moderate equipment only. The attack is not easily detectable. A more robust 802.11i specification is needed to strengthen 802.11i against DoS attacks [6].

As shown in Figure 5, the STA, AP and Authentication Server (AS) negotiate with each other. These negotiations are complex processes. The red rectangles mark the messages that are not authenticated by using any keys. These messages provide possible vulnerabilities to be exploited by the attacker by spoofing the same messges.

Since the DoS vulnerabilities is not solved in the 802.11i standard, the 802.11i network inherits the vulnerabilities of the 802.11 network. The DoS vulnerabilities that were decribed in section 3.1 also happened in 802.11i network.

We first discuss the related countermeasures of the disassociation attack in 802.11i network, and then describe the DoS attacks that are unique in 802.11i network.

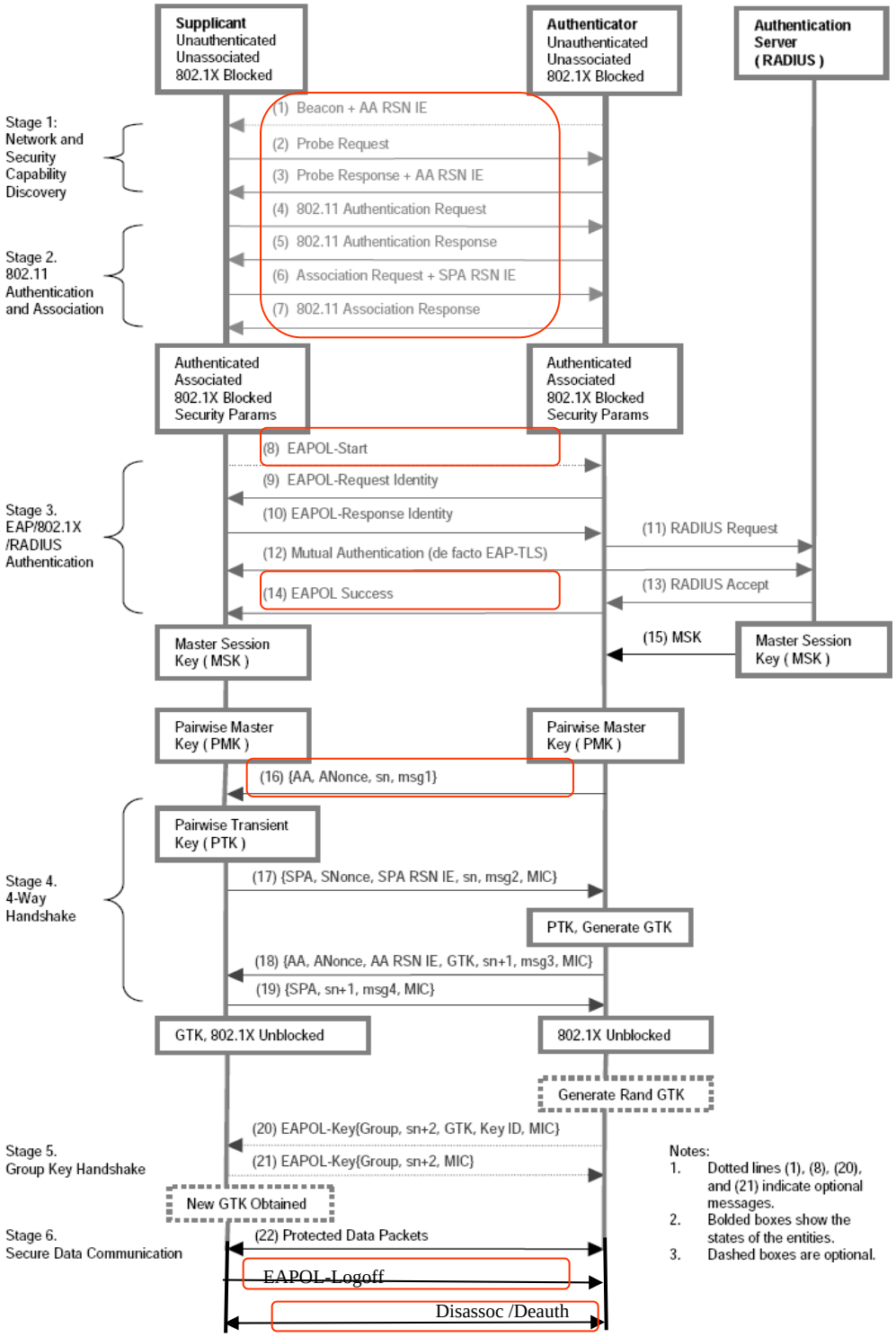


Figure 5 RSNA Establishment Procedures [6]

3.2.1 Deauthentication and disassociation attacks against 802.11i

network

As shown in Figure 6, the 802.1X state machine was applied to 802.11i. 802.11i did not modify the 802.11 (de)authentication and (dis)association mechanism. The deauthentication and disassociation attacks happened as usual.

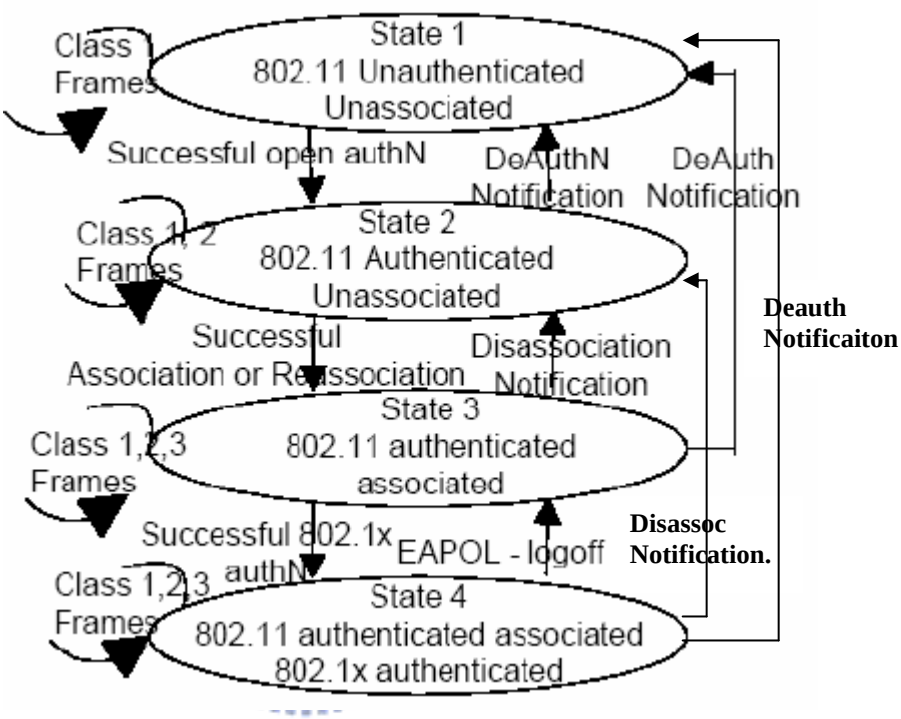


Figure 6 802.11 / 802.1X state machine. Amended from [21][24]

Ping Ding, JoAnne Holliday and Aslihan Celik devised a so called Central Manager (CM) to defend DoS attacks when the 802.1X was applied to 802.11i network. A CM is designed to manage a large number of APs and their STAs. The CM is a back-end server that takes the role of the authentication server (AS) defined by 802.1X. It not only takes the responsibilities of the AS, but also tracks STAs to avoid the DoS attacks [21].

After an AP receives a disassociation frame from a STA, the AP will forward the frame to the CM. The CM sends a request frame to that STA and asks if it really wants to be disassociated. After the STA receives the request packet, it needs to give a confirmation or denial response. If the CM receives a confirmation message, it will

send a disassociation-continue message to the AP. Then the AP really disassociates the STA. If the CM receives a denial message or does not get a message from the STA, the CM will send a disassociation-ignore message to the AP. The AP will ignore the disassociate request and keep the STA's current status [21].

There are some disadvantages in their design. If an attacker pretends to be an AP and sends disassociation frames to a certain STA or broadcasts them to all STAs, the attack will succeed, since the AP and the CM server do not receive any disassociation messages. Furthermore, the current authentication server needs to be modified. The server will inevitably suffer from heavier load. Also, the 802.11i standard must be altered and will not be backward compatible with 802.11 networks that do not implement 802.1X architecture.

3.2.2 EAPOL-Failure and EAPOL-Logoff message attacks

Besides the disassociation attack, there are several DoS attacks that exploit the unprotected EAP messages in 802.1X authentication. As shown in Figure 5, an adversary can forge the EAPOL-Failure message and the EAPOL-Logoff message to disconnect the supplicant [14][15][22].

By disguising as an AP, an attacker can send EAPOL-Failure message to force the STA to stop the negociation among the AP, AS and STA. Then the STA must restart the process of authentication and association. If the attacker sends EAPOL-Failure message repeadly, the DoS attacks happen.

On the other hand, if an attacker pretends to be a certain STA and sends EAPOL-Logoff message to the AP, the AP would log off the STA. If the attacker sends EAPOL-Logoff messages continually, the communication between the AP and STA would be blocked.

Ping Ding et al. also devised Central Manager to defend EAPOL-Failure and

EAPOL-Logoff message DoS attacks in 802.1X applied to 802.11i network [14]. We have mentioned the deficits of the CM approach in the previous section. We do not discuss them here again.

3.3 Lightweight authentication on 802.11

Since the 802.11 management frames are easily forged and exploited by the attacker to launch DoS attacks, it is important to protect the critical frames such as deauthentication frames and disassociation frames. However, we must keep in mind that the computing and memory resources of mobile devices are limited. The use of sophisticated encryption and decryption processes would dominate other kinds of DoS attacks. The lightweight authentication on 802.11 privides another approach to solve the DoS attacks against 802.11 networks.

3.3.1 One-bit lightweight authentication

SOLA, Statistical One-bit Lightweight Authentication, is a new identity authentication protocol proposed to detect unauthorized access in 802.11 network.

It assumes that no encryption will be used at the link layer and that IPSec is used for end-to-end security at the network layer. The main idea is to compute an identical random authentication stream in the STA and the AP, and then add one bit from this stream into the MAC layer header for identity authentication. The goals of SOLA are secure and useful, cheap and robust [11].

We briefly decribe SOLA protocol with the following key words: ASG, packet format, and synchronization algorithm [11].

ASG (Authentication Stream Generator): The purpose is to generate an authentication stream that cannot be guessed by an attacker. It is assumed that the STA and the AP will share a session key. Based on this session key the random

authentication stream is generated from the ASG.

Packet Format: Inserting this new identification bit in the packet is an important issue. The authentication bit would be inserted into IEEE 802.11 MAC header of the data packet, and the “failed” or “succeeded” bit, from the AP to the STA, will be inserted in the response ACK packet. In the simulations, the most significant bit in the sequence control field and the most significant bit in the duration field are used for the data packet and for the ACK packet, respectively..

Synchronization Algorithm: Due to packet loss and other reasons for failures, the bit stream will not be synchronized. So, SOLA designed a synchronization mechanism to mitigate the problem.

The major purpose of SOLA protocol is to detect an attack. SOLA protocol offers a statistical way to identify the origin of the packets for the purpose of access control. The authors claim that the SOLA protocol is well suited in a wireless resource-constrained environment. Furthermore, it is possible to develop a framework to detect Denial of Service attacks or an adversary who tries to attack the network by guessing the identity authentication bit [11].

H. Wang et al. followed the lightweight authentication ideas, but criticized that a severe problem exists in the synchronization algorithm of [11]. They developed a workable synchronization algorithm [12]. H. Wang et al. incorporated the synchronization mechanism into the current IEEE 802.11 network. They concluded that their lightweight authentication for access control contains the following feature [12]:

Lightweight: Only one bit is added into each frame in the proposed scheme and is easily processed.

Simplicity: No encryption or decryption is needed for the proposed scheme.

Continuous authentication: The system is always authenticating hosts. Continuous

authentication is suitable for wireless networks since lower overhead is needed in authentication process.

High efficiency: When non-synchronization is detected, the synchronization algorithm can resynchronize in a short time.

Fault tolerance: When the BER (Bit Error Rate) is high, the system can tell there are malicious attackers and wireless errors [12].

They claimed that they would “show some evaluation results later to approve the high efficiency and fault tolerance”, but we cannot find the results they promised.

Both of the researchers focused on sycronization algorithm and the statistical analysis. Implementation on real wireless environment is not sufficiently considered.

Based on the above analysis, lightweight, mutual and per-packet authentication are feasible approaches for enhancing the security of 802.11 networks.

3.3.2 Enhanced lightweight authentication

Kui Ren et al., found that the sever synchronization problem exists in Johnson’s work on the lightweight authentication due to the frame loss problem in the wireless networks [20]. They also criticized that the researches of Wang et al., were still not efficient.The loss of frames happen frequently in wireless networks, and non-synchronization between communication parties occurs frequently too. It results in additional communication delay, which could be critical to many realtime applications.

Kui Ren et al. proposed an enhanced lightweight authentication protocol for access control at the MAC layer in wireless LAN. They examined the redundancy existed in the MAC header, and adopted an enhanced 3-bit authentication mechanism [20]. (see Figure 7, the part of data frame control field)

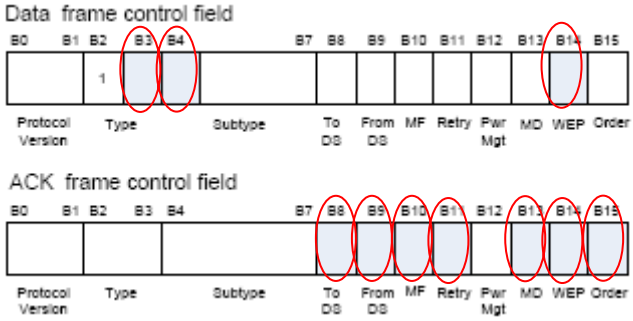


Figure 7 Adaption of frame format to the Kui's proposed protocol [20]

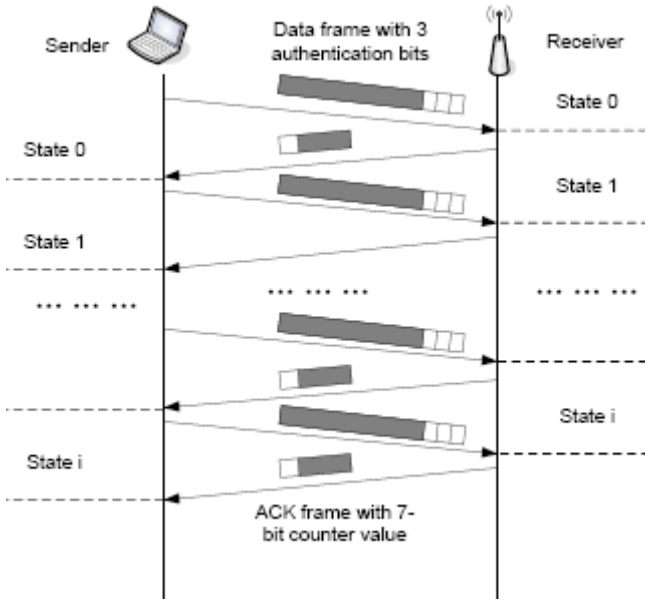


Figure 8 Overview of Kui's proposed protocol [20]

As shown in Figure 8, Kui's proposed protocol works as follows [20]. At the beginning, the sender and the receiver establish a common random bit stream generator by sharing a seed value. The random bit stream generator continuously outputs 3 bits as a unit which was then inserted into frame control field of the sending data frame. The receiver will generate the same authentication bit stream as that of the sender. Upon receiving a frame, the receiver first checks the 3-bit authentication value. If the value matches that of the receiver's, the frame is authenticated and is processed further.

On the other hand, when the ACK-failure frame was sent, 7-bit counter value is inserted into the MAC frame header. As shown in Figure 7 (see the part of ACK frame

control field), seven corresponding bits are chosen based on the structure of the frame control field of the ACK frame. It is known that the 7 bits in the control frame are simply set to 0. The 7-bit counter contains the synchronization information between the communicating parties [20]. The details are, however, beyond the scope of our research.

The authors also offered a statistical way to identify the origin of the data frame for the purpose of detecting an attack. They asserted that the protocol is fully compatible with current IEEE 802.11 frame structure and provides a highly efficient identity authentication scheme [20].

Howerever, all of the researchers described in sections 3.3.1 and 3.3.2 were certain that the lightweight authentication utilized in the wireless network is a feasible approach. In summary, the lightweight authentication mechanism, if it is applyed to the 802.11 network, contains some benefits: lightweight, mutual authentication per frame, high efficiency, and backward compatibility.

Chapter 4 Proposed protocol to defend 802.11 DoS attacks

We designed a random bit authentication mechanism to defend 802.11 deauthentication and disassociation flooding attacks. In this chapter, for backward compatibility reason, we first analyze the unused bits of 802.11 management frames to determine the unused bits in the header and body of the management frame we may insert our random authentication bits.

Secondly, we examine the characteristics of the sequence number subfield in the sequence control field of the management frame to devise our defending methods.

Finally, we design a two-phase filtering mechanism to defend deauthentication and disassociation flooding attacks.

4.1 Unused bits of 802.11 management frame

We analyze the frame control field and frame body field of the management frame to find out the unused bits. The other fields are not suitable to be modified in 802.11. The general management frame format is shown in Figure 9.

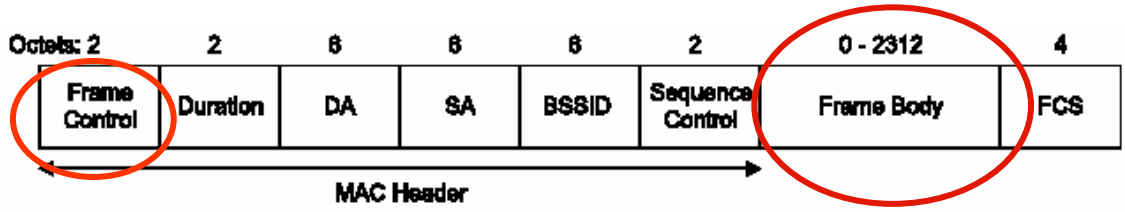


Figure 9 General management frame format and the fields to be used to insert random authen bits [2]

4.1.1 Management frame control field analysis

Figure 10 illustrates the frame control field of the MAC header in the management frame. The frame type can be determined by the type field and the subtype field. It is not nessary to examine the unused bits shown in Figure 10 for a management frame.

In Figure 10, the “To DS” field is set to 1 in frames destined for the distribution system (DS). The “From DS” field is set to 1 in frames exiting the DS. The “More Fragments” bit is set to 1 to indicate that the frame is a fragmental frame. The management frame is always limited bythe maximal length of 2312 bytes for the frame body. There are no framgmental frames in any subtypes of the management frames. The “Power Management” bit and the “More Data” are used only in the control frame to indicate the power management mode of a STA. The “Order bit” is used in the data frame which is being transferred using the Strictly Ordered service class [2].

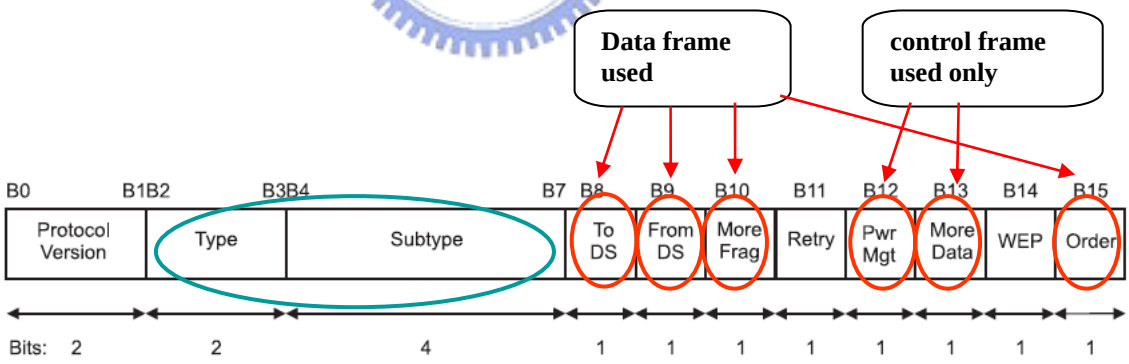


Figure 10 Unused bits of the frame control field in the management frame [2]

4.1.2 Management frame body analysis

Table 2 indicates the contents of the frame body in the authentication frame. The value of authentication algorithm number is 1 or 0. There is only 1 bit that is used, but it contains 16 bits in the 802.11 standard as shown in the Figure 11. It leaves 15 unused

bits. Similarly the authentication transaction sequence number uses 3 bits, and others are unused, as showen in the Figure 12.

Order	Information	Size
1	Authentication algorithm number	16 bits
2	Authentication transaction sequence number	16 bits
3	Status code	16 bits
4	Challenge text	Max 255 octets only present in shared key Authentication frames

Table 2 Athentication frame body [2]

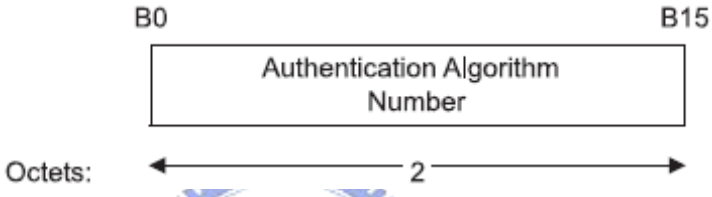


Figure 11 Authentication Algorithm Number fixed field [2]

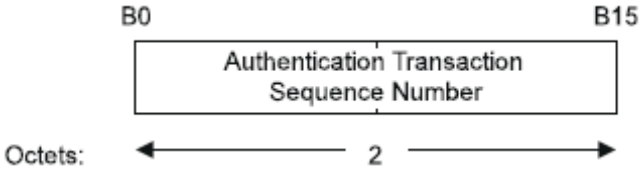


Figure 12 Authentication Transaction Sequence Number fixed field [2]

Order	Information
1	Reason code

Table 3 Deauthentication and disassociation frame body [2]

Reason code	Meaning
0	Reserved
1	Unspecified reason
2	Previous authentication no longer valid
3	Deauthenticated because sending station is leaving (or has left) IBSS or ESS
4	Disassociated due to inactivity
5	Disassociated because AP is unable to handle all currently associated stations
6	Class 2 frame received from nonauthenticated station
7	Class 3 frame received from nonassociated station
8	Disassociated because sending station is leaving (or has left) BSS
9	Station requesting (re)association is not authenticated with responding station
10–65 535	Reserved

Table 4 Deauthentication and disassociation reason code [2]

The reason codes of the deauthentication and disassociation frame bodies are shown in the Table 3 and Table 4. In 802.11 standard, only 4 bits are used. There are 12 unused bits that can be used for random authentication bits. In 802.11i only 5 bits are used, and other bits are unused.

The (re)association request and response frame body as shown in Table 5 and Table 6 has the same 16 bits “Capability information” field. There are 11 (B5 to B15) bits reserved and there can be used as random authentication bits as shown in Figure 13.

Association request			Association response		
Order	Infromation	Size	Order	Infromation	Size
1	Capability information	16 bits	1	Capability information	16 bits
2	Listen interval	16 bits	2	Status code	16 bits
3	SSID	Max 34 octets	3	Association ID (AID)	16 bits
4	Supported rates	Max 10 octets	4	Supported rates	Max 10 octets

Table 5 Association request and response frame body [2]

Reassociation request			Reassociation response		
Order	Infromation	Size	Order	Infromation	Size
1	Capability information	16 bits	1	Capability information	16 bits
2	Listen interval	16 bits	2	Status code	16 bits
3	Current AP address	48 bits	3	Association ID (AID)	16 bits
4	SSID	Max 34 octets	4	Supported rates	Max 10 octets
	Supported rates	Max 10 octets			

Table 6 Reassociation request and response frame body [2]

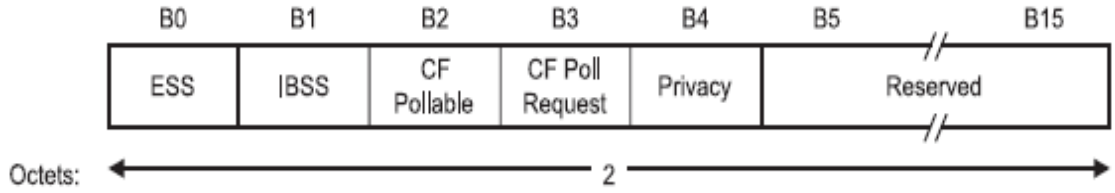


Figure 13 Capability Information fixed field [2]

Note: 1. 11(B5 - B15) bits are reserved and can be inserted random authen bits.

2. In 802.11b only 8(B8-B15) bits are reserved, others are defined.

The above analysis in sections 4.1.1 and 4.1.2, has determined the unused bits in the header and body of 802.11 management frames. The number of unused bits is enough to be exploited to implement our DoS defense.

4.2 Applying Sequence Number field to detect DoS attack

4.2.1 Sequence Number field characteristics and function

Figure 9 in section 4.1 exhibits the format of 802.11 management frame. There is a sequence control field. The sequence control field is 16 bits in length and consists of two subfields, the Sequence Number (SN) and the fragment number. The format of the sequence control field is illustrated in Figure 14.

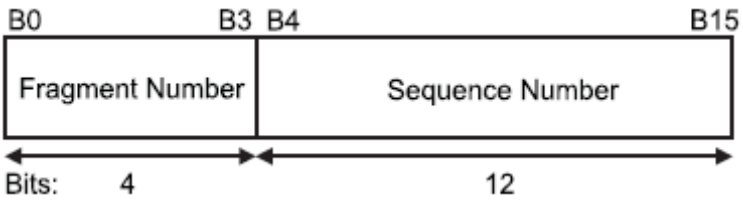


Figure 14 Sequence control field [2]

The sequence number field contains 12 bits to indicate the sequence number of a frame. Sequence numbers are assigned from a single modulo 4096 counter, which starts from 0 and is incremented by 1 for each frame. The sequence number remains unchanged in all retransmission frames, or fragments thereof [2].

4.2.2 Filtering sequential Sequence Number to detect illegal frames

Under normal conditions, a legal STA or AP transmits the deauthentication or disassociation frame once to disconnect the session. If the frame is lost, the sender will not receive the ACK frame and retry to transmit the frame with the same sequence number.

In the situation of DoS, the attacker will send the deauthentication or disassociation to the victim continually. Mostly the attacker will flood the deauthentication or disassociation frames to the victims. The sequence number of the deauthentication or disassociation frames will increase by 1 for each attacking frame. The sequence numbers of attacking frames will be sequential. We can utilize this characteristic to detect and defend the flooding attack frames [23]. If the STA or AP sends deauthentication or disassociation frames with sequential sequence numbers, we can treat them as forged frames and drop them directly.

4.3 Random bit authentication for management frame

4.3.1 Some assumptions and random bit stream generation

We assumed that the communicating nodes had shared the same key, and one session key will be generated for each communication based on the shared key. We do not discuss the key generation and exchange issues in our thesis. Furthermore, we assumed that the communicating nodes that implement the same algorithm use the

shared session key to generate the same bit stream. The algorithm is not secret, but the shared key is.

The nodes in the same basic service set utilize the shared session key and algorithm to generate the same bit stream independently, and divide the bit stream into 8 units, each having “N” authentication bits. We call it “N random bits”. As shown in the Figure 15, the node generates a bit stream, and divides each 3 bits as a unit for authentication. We give each unit a certain index number sequentially. We need only 8 units in our design in 802.11 (b, g).

1	2	3	4	5	6	7	8
100	110	101	010	110	101	010	011

Figure 15 Example of the bit stream shard by both of the communicating nodes while N = 3

4.3.2 Random bit authentication for management frames

The scenario of the random bit authentication mechanism is shown in Figure 16. Both of the AP and STA generate the same bit steam independently. When the node (AP or STA) sends (de)authentication or (dis)association frames, it inserts sequentially 3 bits, as a unit, into the unused bit positions of each frame. When the receiving node receives the frame, it first checks the random authentication bits in the MAC header of the incomming frame. If the random authentication bits match the corresponding bits of the receiver, the frame is processed as a legal frame.

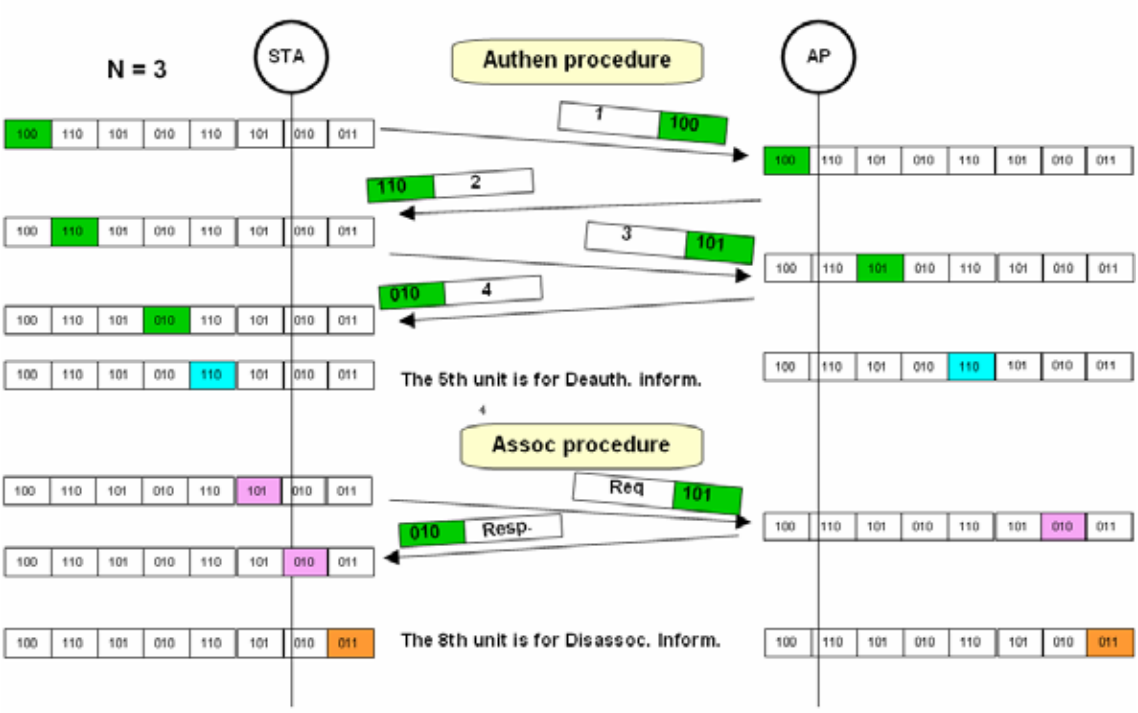


Figure 16 Scenario of random bit authentication for Authen & Assoc procedures

Note: The number of random authentication bits is 3.

Figure 17 illustrates the scenario of random bit authentication under deauthentication or disassociation flooding attacks. We can imagine that since the attaker does not know the value of the 5^{th} and 8^{th} units of random authentication bits, he must keep sending the forged frames containing guessed authentication bits until the guessed bits match the ones on the target victim. The attacker can alternatively use "brute force" to guess the random bits. As shown in Figure 17, the attacker can circularly generate the values from 0 to 7 and insert one as a 3-bit unit to the sending frame for authentication. One of the 8 forged random authentication bits will match the receiver's. The success rate of an attacker to disconnect the session between the AP and STA is 1/8 per cycle. If we increase the number of authentication bits, the rate of successful DoS attacks will decrease exponentially.

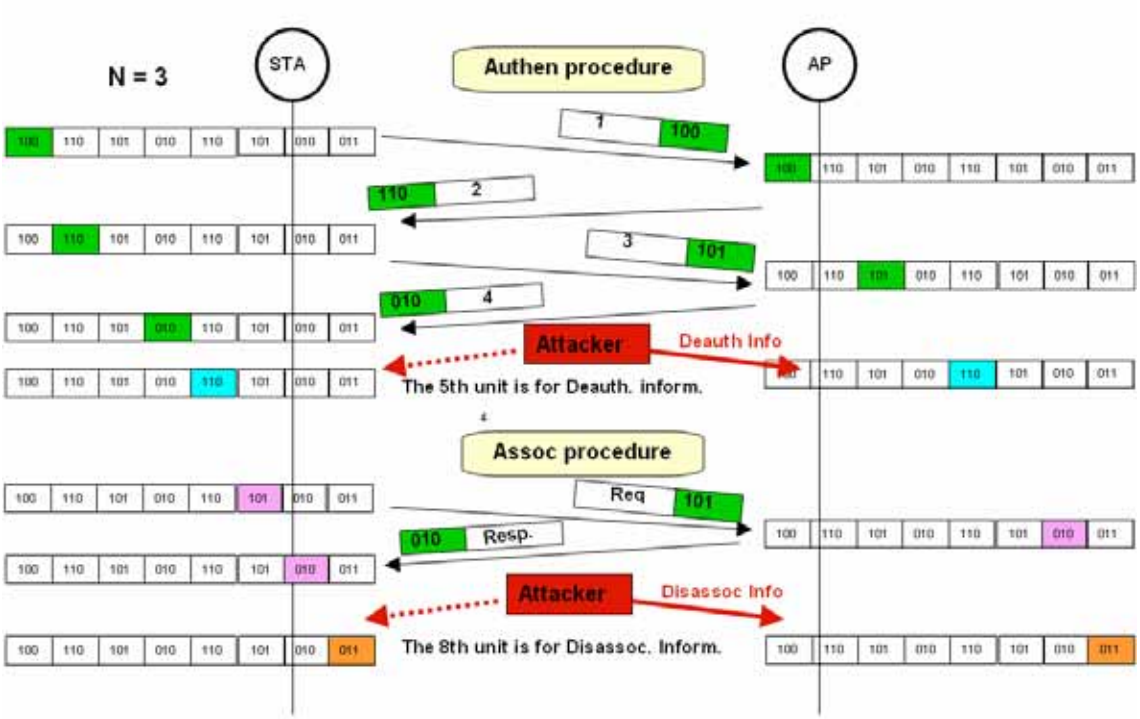


Figure 17 Scenario of attacks of using Random bit authentication for Authen. & Assoc procedures

When we designed and changed the system to insert random authentication bits to the unused bits of the (de)authentication and (dis)association frames, we confronted some problems. In our Linux based platform decribed in the next chapter, we could modify the Host AP driver to insert a maximum of 4 random authentication bits into the frame control field in the MAC header of the authentication and association frames. As shown in Figure 18, we successfully set B9, B12, B13 and B15 to 1 in the frame control field of the (de)authentication and (dis)association frames in master mode. The communication between the AP and STA and the original functions of 802.11 are not disturbed. However we were only successful in changing the bits in **Master mode (Host AP mode).** When we configured the Host AP driver into managed mode (STA mode), we could not modify the driver to insert random authentication bits to the frame control field of the authentication and association frames. We asked Jouni Malinen, the auther of the Host AP driver, for help, and he told us that we could not modify the frame control field through the Host AP driver, because the frame control field is

processed in the firmware of the prism2/2.5/3 based 802.11b card in **Managed mode**.

We searched for other drivers, for example, the Linux-wlan-ng driver, but our efforts went in vain. Consequently, we have to devise an alternative method to implement our design.

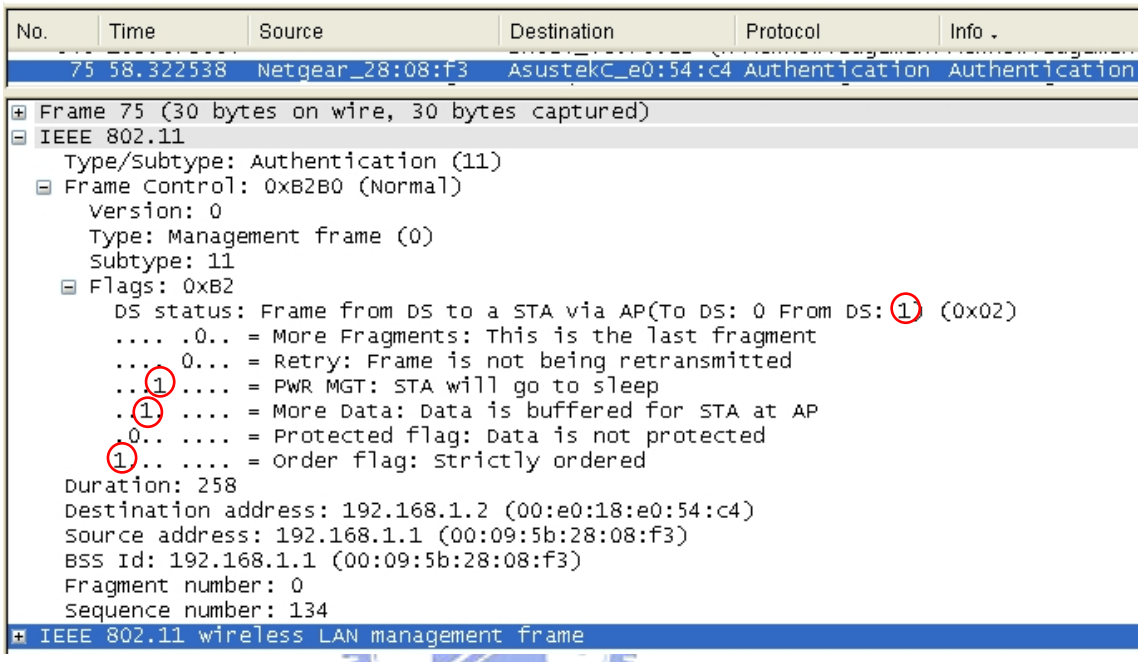


Figure 18 Bit9, Bit12, Bit13 and Bit15 were set to 1 successfully.

The frame control field can be modified to insert random bits in master mode. The attacker (void11), based on Host AP driver in master mode, could transmit arbitrary deauthentication and disassociation frames while the frame control field was modified. We inserted the random authentication bits into the frame control field of the deauthenticaiton and disassociation frames, and send them to the Host AP with forged MAC address of the STA. The attacks succeeded as the forged random authentication bits match the ones of the Host AP.

When the deauthentication and disassociation frames arrive, the Host AP node examine the random authentication bits portion, and drops the forged frames in the case of a mismatch.

To attackour random bit authentication design, it is conseivable that the attacker

will emit the frames with random bits in “brute force”. The attacker sends forged frames with random authentication bit value circularly from 0 to 7 (if N = 3). So, the probability that the attacker succeeds is 1/8 when the number of random authentication bits is 3, and the probability becomes 1/16 when the number of random authentication bits is 4, and so on.

Alternatively, we configured the attacking node (void11) to send one deauthentication or disassociation frame with one of the unused bit (random authentication bit) set to 1 (Bit15 in our implementation) 8 times in a row if the number of random authentication bits is 3. On the receiver (Host AP) side, the Host AP examines only Bit15, and drops frames with Bit15 set to 0.

This alternative implementation is similar to the random bit authentication mechanism that we devised in Figure 16. We exploited this alternative designs to test our random bit authentication mechanism.

Chapter 5 Experimental results

In this chapter, we will present our experimental results and discuss about them. First, we describe our implementation scenario, utilized tools and procedures. Second, we present the experimental results with figures and explanations. Third, we discuss the experimental results and the limitations in our experiments.

5.1 Implementation environment and issues

5.1.1 Scenario of implementation

Figure 19 illustrates our implementation scenario. All the hardware devuces we used are bought from electronic stores. They are popularly used all over the world. We do not design or buy special hardware equipments. We want to examine the vulnerability of the 802.11 network to use commodity devices. We demonstrate that common devices can be easily used to launch DoS attacks.

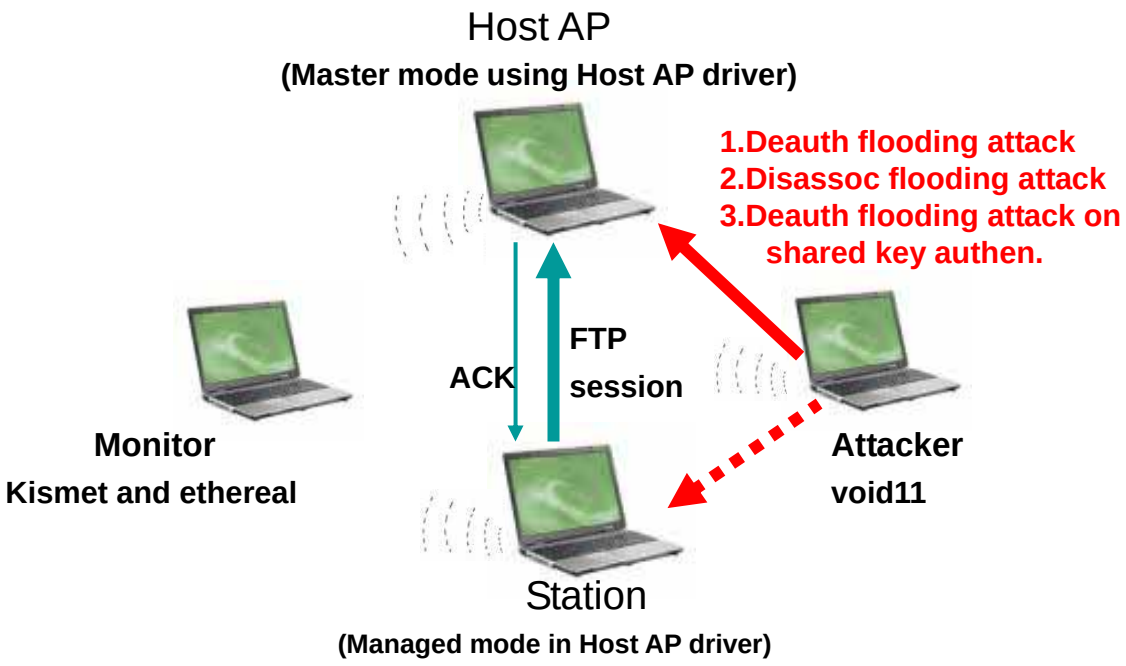


Figure 19 Implementation Scenario

We describe the hardware and software devices in .Table 7. We utilized 4 laptop PCs equipped with Intersil's Prism2/2.5/3 802.11b PCMCIA cards. All the platforms are Linux FC3 based, and all of the wireless network card drivers and necessary softwares are open source. We modified the relevant codes to suit our experiments. The next section would briefly describe the functions of the Host AP drivers and other application programs.

Fuction	Laptop Model	CPU	RAM	802.11 PC Card Model	O.S.	PC Card Driver & Software
Host AP	HP Compaq nc6230	Intel P.M 1.73GHz	1.00GB	Netgear 802.11b MA401 (Chipset: Prism2)	Linux FC3 Kernel: 2.6.9-1.667	Host AP driver with Master mode
Station (STA)	Asus A2500H	Intel P4 2.8 GHz	224MB	Intersil Prism2.5 802.11b PC card	Linux FC3 Kernel: 2.6.9-1.667	Host AP driver with Managed mode
Attacker	HP Compaq nc6230	Intel P.M 1.73 GHz	1.00GB	Netgear 802.11b MA401 (Chipset: Prism2)	Linux FC3 Kernel: 2.6.9-1.667	Host AP driver void11
Monitor	Toshiba TE2100	Intel P4-M 1.80GHz	256MB	Asus WL-100 (Chipset: Prism2)	Linux FC3 Kernel: 2.6.9-1.667	Host AP driver Kismet Ethereal

Table 7 Implemental equipment hardware model and software

5.1.2 Tools and utilities

a. Host AP

Host AP is the driver for the 802.11b cards with Intersil Prism2/2.5/3 chip.

The IEEE 802.11b network cards based on Intersil's Prism2/2.5/3 chipset support a so called Host AP mode. The firmware of such cards takes care of time critical tasks like beacon sending and frame acknowledging, but leaves other management tasks to the host computer driver. In addition to implementing common wireless interface functions, the driver implements the following IEEE 802.11 functions: authentication (and deauthentication), association (reassociation, and disassociation), and data transmission between two wireless stations. The driver also contains various features for exploring IEEE 802.11 environments [8].

In our implementation on Linux platform, we installed the Host AP drivers on all four experimental nodes to drive the 802.11b PCMCIA cards. However, we configured each node to play different roles using “iwconfig” command. The Host AP node was configured into the master mode, the STA node was configured into the managed mode, and the other two nodes were configured into the master mode which was necessary for Kismet (monitor) and void11 (attacker).

b. Kismet

Kismet is an 802.11 layer 2 wireless network detector, sniffer, and intrusion detection system. Kismet works with wireless cards which support raw monitoring mode, and can sniff 802.11b, 802.11a, and 802.11g traffic. It identifies networks by passively collecting packets, detecting standard named networks, detecting hidden networks, and inferring the presence of no beaconing networks via data traffic [6].

We installed Kismet, not ethereal, on the monitor node to gather 802.11 frames. The ethereal is a popular tool, but it cannot capture the layer 2 802.11 frames. We configured the Kismet to disable channel hopping in our implementation. The capturing work did focus on the channel 3 singal by which the Host AP and STA communicated with each other.

c. Ethereal

Ethereal is popularly used for troubleshooting, analysis, software and protocol development, and education. It's open source, and runs on all popular computing platforms, including Unix, Linux, and Windows [18]. In our implementation we install ethereal on both Linux and Windows plantforms.

Ethereal is a GUI network protocol analyzer. We used it to display and analyze the packets captured by Kismet. In addition to analyzing the captured data, we utilize the "IO Graphs" tool to construct the graphs of the FTP sessions.

d. void11

Void11 is a free implementation of some basic 802.11b attacks, and also some original features. To implement deauthentication DoS flood, void11 floods wireless networks with deauthentication packets and spoofed BSSID. The authenticated stations will drop their network connections. To implement authentication and association request flooding, void11 floods APs with authentication or association request packets with spoofed random station MAC addresses. Some APs will deny any service after the request flooding launched. Some APs or 802.11 cards died for a period after void11 launched DoS flooding attacks [7]. Gvoid11 is the new graphical user interface of void11.

We installed void11 version 0.2.0. For our specific experimental design, we used the command mode, the "void11_penetration -D -s 00:30:B4:01:00:06 -B 00:09:5B:28:08:F3 wlan0" command to start attacking the target Host AP and STA.

Next, we modified void11 to launch deauthentication and disassociation flooding attacks with skipping several sequence numbers, so that the deauthentication or disassociation frames would not be filtered out by the receiver (Host AP).

5.1.3 Testing procedures

To monitor the behavior of the 802.11 network, we let the STA send a file through FTP protocol to the Host AP. The file size is 21,872,640 bytes, and it takes about 35 seconds to be transmitted from STA to AP in 802.11b environment. After the STA starts to send the files for 15 seconds, which we later feferred to as the “waiting time”, we launch deauthentication or disassociation flooding attacks for about 10 seconds from the attacker node (void11). The 15 second “waiting time” plus the 10 second attack time is 25 seconds which is under 35 seconds. We recorded the duration of the FTP sessions between the STA and Host AP to measure the effect of the attack. The behaviors of the whole FTP sessions are monitored and saved by the monitor node (Kismet).

In the following experiments, we tested the scenario for at least 10 times, and averaged the duration. To be concrete, we chose one of the 10 experiment trials as demonstration graph. We decribe the procedure below:

a. We tested the normal FTP sessions as our baseline model.

b. The deauthentication and the disassociation flooding attacks were launched knowing that all of the forged frames were dropped directly by the Host AP. Although all the attacking frames failed, the traffic also consumed the wireless bandwidth. We measured the duration of such deauthentication, disassociation flooding attacks, and association request flood.

c. We launched the deauthentication and the disassociation flooding attacks with the random bit authentication mechanism enabled. The number of random bits for authentication ranged from 0 to 9. We tested each number of the random bit authentication separately.

d. We launched deauthentication and the disassociation flooding attacks with

sequential sequence number filtering mechanism enabled. The Host AP filtered out the subsequent frames of which the value of sequence number deviation (SND) was positive under or equal to 1, 4, 8 and 16.. We tested each value of SND separately.

e. The testing condition was similar to that in step “d”, but the SND was set equal to 8, 16, 24, 32, and 64. We further modified void11 to emit deauthentication or disassociation flooding frames after sending SND association request frames. The value of the sequence number in the deauthentication or disassociation frames would increment in steps of SND, and the attacking frames would be accepted by the Host AP. The object of this test was to examine how effective it is to use only the sequence number filtering mechanism to defend the deauthentication and the disassociation flooding attacks without the use of the random bit authentication mechanism.

f. We designed a two-phase filter mechanism to defend the deauthentication and disassociation flooding attacks. We combined the above “c” and “d” steps to test how it worked efficiently to defend deauthentication and disassociation flooding attacks.

g. We utilize the above “f” step to the Host AP and the Windows STA which were configured as shared key authentication. The WEP was enabled between the Host AP and the STA when they transmted data to each other.

5.2 Results

We present the experimental results according to section 5.1.3 implemental procedures. In section 5.2.1, we present the normal FTP session and some bandwidth consuming considerations under deauthentication, disassociation and association request floods. In section 5.2.2, we present the experimental results of random bit authentication mechanism. In section 5.2.3, we present the results of the sequence

number filtering mechanism. In section 5.2.4 we present the two-phase filtering mechanism to defend deauthentication and disassociation flooding attacks. In the final section we present the results of applying two-phase filter mechanism to the shared key authentication.

5.2.1 Normal FTP session and bandwidth comsuming consideration

To establish the normal FTP session between the STA and Host AP as a reference model, we recorded the duration of the FTP sessions 10 times. We averaged these 10 durations. The average duration of normal FTP sessions is 35.5 seconds as shown in Table 8.

Session No.	1	2	3	4	5	6	7	8	9	10	Average
Duration (sec)	35	35	35	37	36	36	35	35	36	35	35.5

Table 8 Average duration of normal FTP sessions

For concreteness, we choose one of the 10 captured experimental data to represent normal FTP behavior in Figure 20. The graph was taken from the “IO Graphs” of the ethereal.

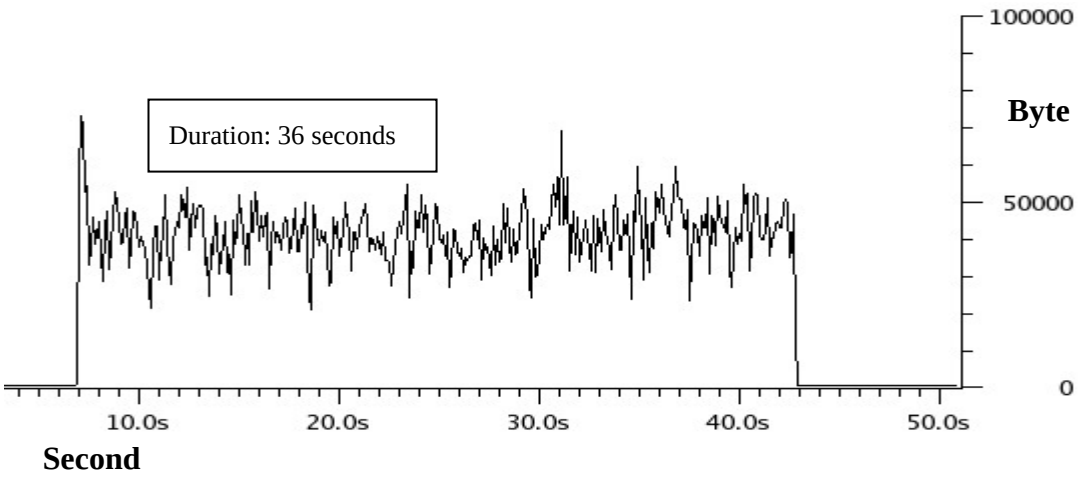


Figure 20 Graph of normal FTP session

In our experiments, we measure the duration of uploads to explore the effect of

the deauthentication and disassociation flooding attacks. We must consider the bandwidth consumption of deauthentication or disassociation flooding attacks. Since the wireless bandwidth is consumed by attacking frames, whether the attacking frames were successful or not.

In Table 9, we recorded the durations under failed deauthentication and disassociation flooding attacks. In other words, we launched deauthentication and disassociation flooding attacks, but all attacking frames were dropped by the Host AP. The average duration is 38.4 seconds for deauthentication flooding attacks, and 38.1 seconds for disassociation flooding attacks, respectively.

Session No.	1	2	3	4	5	6	7	8	9	10	Average
Duration of FTP session under failed Deauth flooding attacks	39	38	38	38	38	38	38	39	39	39	38.4
Duration of FTP session under failed Disassoc flooding attacks	38	38	38	38	38	38	38	39	38	38	38.1

Table 9 Duration under consideration of bandwidth consumption of Deauth & Disassoc flooding attacks

To test the sequence number filtering mechanism decribed in section 5.2.3, we must measure the bandwidth consumption of association request flooding. The result is 38.5 seconds as shown in Table 10.

Session No.	1	2	3	4	5	6	7	8	9	10	Average
Duration (sec)	38	38	37	42	37	38	38	38	37	42	38.5

Table 10 Duration under consideration of bandwidth consumption in association flooding attacks

5.2.2 Random bit authentication defending mechanism

Random bit No.	0	1	2	3	4	5	6	7	8	9
Duration under Deauth attacks	59.2	58.6	59.1	59.3	57	58	50.2	39.7	37.9	38.2

Duration under Disassoc attacks	60.3	58.6	59.6	58.3	59.1	54.4	44.8	38.6	37.1	36.9
Delay (1)	23.7	23.1	23.6	23.8	21.5	22.5	14.7	4.2	2.4	2.7
Delay (2)	20.8	20.2	20.7	20.9	18.6	19.6	11.8	1.3	-0.5	-0.2

Table 11 Relation of random bit authentication number and Deauth & Disassoc flooding attacks

Note: 1.Delay (1) is to compare the duration of Deauth flooding attacks with the normal FTP Session duration.

2. Delay (2) is to compare the duration of Deauth flooding attacks with duration under bandwidth consumption of Deauth flooding attacks.

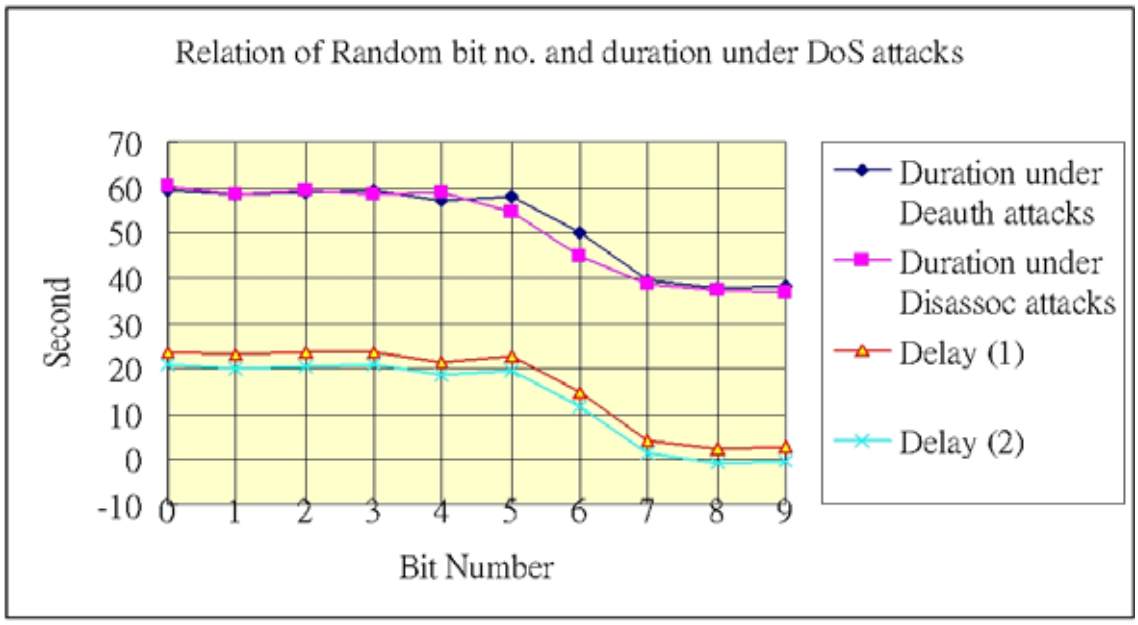


Figure 21 Figure of relation of random bit authentication number and Deauth & Disassoc flooding attacks

As shown in Table 11 and Figure 21, the more random authentication bits we used to defend deauthentication and disassociation flooding attacks, the more difficult it is for the attacker to succeed. As shown in Figure 22, we do not use the random bit authentication mechanism (or the number of random authentication bit is 0), and the FTP session was blocked when the attacker launched deauthentication flooding attacks. We examined the snapshot of this FTP session in the monitor node and found that there were few FTP data frames flowing between about the 31st second and the 33rd second, just as shown in Figure 22. Futhermore, after the deauthentication

flooding attacks, the FTP session delayed an additional 13 seconds before recovery. We called this phenomenon “FTP delay”. We would discuss the FTP delay later.

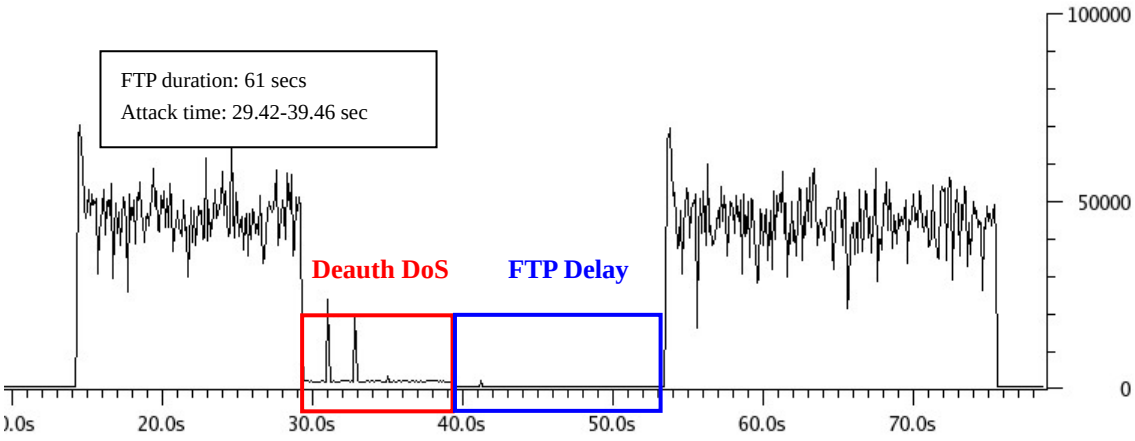


Figure 22 Attacker launched Deauth flooding attack with no (or 0) Random bit authentication defense.

As shown in Table 11 and Figure 21, using 5 random authentication bits to defend DoS attack is not enough. The deauthentication flooding attacks succeeded as though there is no defense. Void11, the attack tool, launched deauthentication flooding attacks intensely, and the effect of such flooding attacks was serious. The function in void11 for launching deauthentication flooding attack is about 20 lines. It was originally implemented to forge the MAC address of AP and BSSID to send deauthentication messages to all STAs (broadcast) or to a single victim. In our implementation we modified void11 to spoof the MAC address of STA, and send deauthentication messages to AP. Void11 sends the forged frames with a “for” loop, so the attacking frames are send very frequently. For instance, in our experiement, we found that the attacker node could send about 80 forged deauthentication or disassociation frames per second.

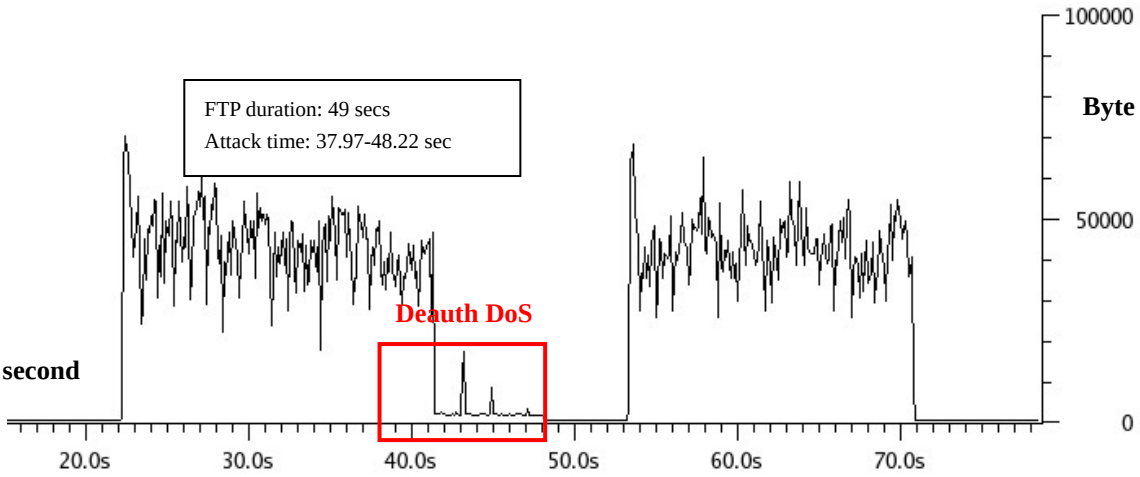


Figure 23 Using 6 random bits for authentication to defend Deauth flooding attacks

We also tested using 6, 7 random authentication bits. We found that the effects of the attacks were alleviated. Figure 23 illustrates the result of using 6 random bits for authentication. After we increased the number of random authentication bits up to 8 and 9, we found that the serious deauthentication flooding attacks were defended successfully. As shown in Figure 24, there were no noticeable DoS attacking traits, and the FTP session continued smoothly. However, the wireless bandwidth consumption is inevitable, but it is not heavy. We compare the bandwidth consumption of using 8 random authentication bits (2.4 seconds) with that of failed deauthentication flooding attacks (2.9 seconds). They are close, and the deviation is slight.

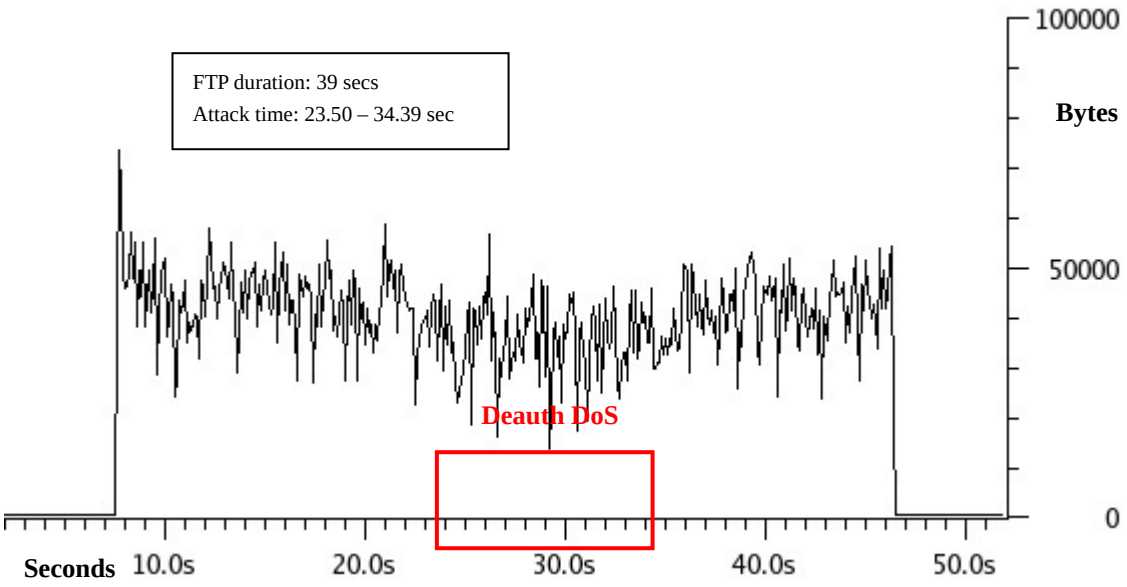


Figure 24 Using 8 random bits for authentication to defend Deauth flooding attack

As for the afore mentioned FTP delay, we founded that the delay happened after deauthenticaiton flooding attacks as shown in Figure 22 and 23. Someone may question the reason why the FTP session delayed for a long time after deauthentication flooding attacks stopped. We have some evidents to explain this phenomenon.

After the attacker stopped launching deauthentication flooding attacks, the MAC layer connection between the Host AP and the STA recovered immediately, but the FTP session did not, since the delay is caused by the FTP program.

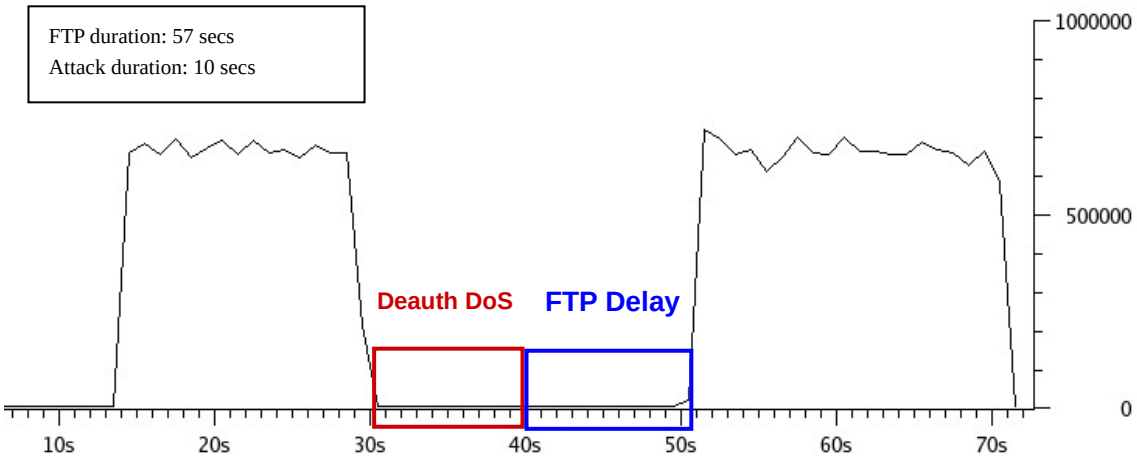


Figure 25 FTP session delay after Deauth flooding attacks

First, the raw data of Figure 22 is captured by kismet installed on the monitor

node. On the other hand, the raw data of Figure 25 is captured by ethereal installed on the STA node. As shown in Figure 25, we captured the TCP frames corresponding to FTP data frames in our experiment, flowing through the 802.11b card directly. We examined the behavior of the data flow. It shows that the FTP session delay happened like that in Figure 22.

Second, we sended ICMP frames continually with “ping” command from the STA to Host AP before starting FTP session between the STA and Host AP. Figure 26 illustrates the result.

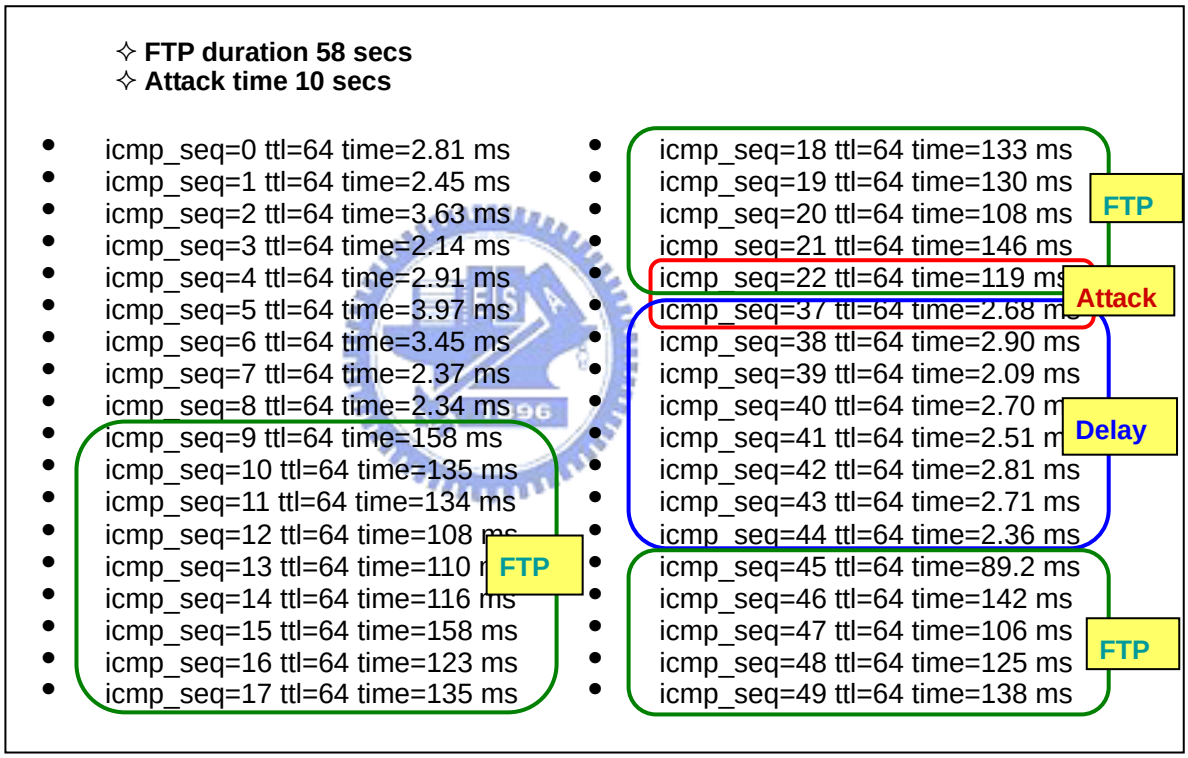


Figure 26 Changes of ping echo time during the FTP session delayed after Deauth flooding attacks

We found that the echo time of the 9th to 22nd and 45th to 49th frames are longer than that of 0th to 8th and 37th to 44th frames. The load of the network was heavy, because the STA was sending frames to Host AP by FTP. In Figure26, we marked the lost frames between the 22nd to 37th frames with red rectangle, beause we launched deauthentication flooding attacks for 10 seconds after 22nd ICMP frame. All of the communications between the STA and Host AP were blocked, and it caused the ICMP

session to lose 14 frames. After 10 second deauthentication flooding attacks, the MAC layer between the STA and Host AP connected again, but the FTP session is slow to recover. It is clearly shown in Figure 26 that the echo time of 37th to 44th ICMP frames was lightweight just as 0th to 8th frames.

The phenomenon of the FTP delay does not only happen on our Linux-based DoS implementation, but also happened on Windows-based system. We found the similar result as we implemented the STA node on Windows XP. As shown in Figure 27, we launched deauthentication flooding attacks during the FTP session between Windows XP STA and Host AP. The red rectangle marked the attacking duration, and the yellow rectangle marked the FTP delay in Figure 27.

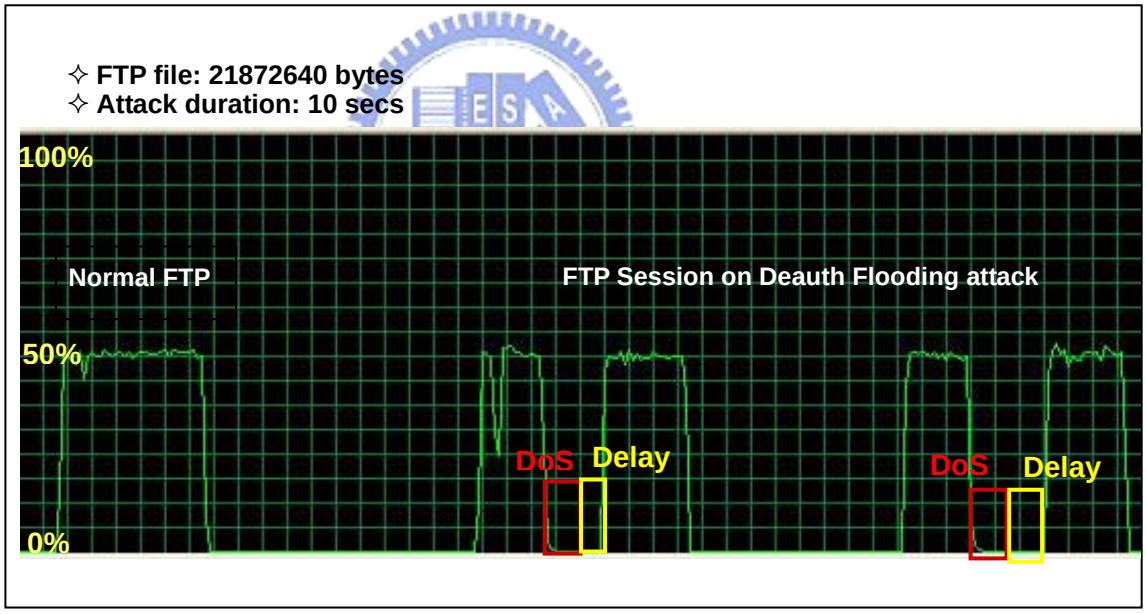


Figure 27 FTP session delay after deauth flooding attacks between Windows XP STA and Host AP.

Why did the FTP delay happen? We did not examine the FTP program. That's beyond our research scope in this thesis. We do focus on our 802.11 DoS defending issues.

5.2.3 Sequence Number filtering mechanism

SND ($SN_i - SN_{i-1}$)	1	4	8	16
FTP Duration	59.7	42.4	39	39.3
Duration under bandwith consumption of failed Deauth flooding attacks	38.4	38.4	38.4	38.4
Delay (1)	24.2	6.9	3.5	3.8
Delay (2)	21.3	4	0.6	0.9

Table 12 Relation of filtering out the SND of subsequent SN and the FTP duration under Deauth flooding attacks.

Note: 1. Delay (1) is to compare the duration with the normal FTP Session duration.

2. Delay (2) is to compare the duration with duration under bandwidth consumption of failed Deauth flooding attacks.

3. The acronym SND stands for Sequence Number Deviation. SND = ($SN_i - SN_{i-1}$)

Table 12 displays the result of filtering out the subsequent deauthentication frames of which the sequence numbers are under or equal to the SND. To our astonishment, when SND is equal to 1, we cannot find any defense to the deauthentication flooding attacks as we analyzed 802.11 specificaiton in section 4.2. The STA sending FTP data was blocked as similar as the situation between the STA and the Host AP with no defensive mechanisms. However, if we set SND equal to 4, the Host AP can defend deauthentication flooding attacks to a certain extent. Moreover, when we set SND equal to or more than 8, we can defend the deauthentication flooding attacks successfully.

Time	SN	SND
12:13:09	0	0
12:13:09	1	1
12:13:09	2	1
12:13:09	4	2
12:13:09	5	1
12:13:09	6	1
12:13:09	7	1
12:13:09	8	1
12:13:09	9	1
12:13:09	10	1
12:13:09	11	1
12:13:09	12	1
12:13:09	14	2
12:13:09	15	1
12:13:09	16	1
12:13:09	17	1
12:13:09	18	1
12:13:09	19	1
12:13:09	20	1
12:13:09	22	2
12:13:09	23	1
12:13:09	24	1
12:13:09	25	1
12:13:09	26	1
12:13:09	27	1

Time	SN	SND
12:13:09	28	1
12:13:09	29	1
12:13:09	30	1
12:13:09	32	2
12:13:09	33	1
12:13:09	34	1
12:13:09	35	1
12:13:09	36	1
12:13:09	37	1
12:13:09	38	1
12:13:09	39	1
12:13:09	41	2
12:13:09	42	1
12:13:09	43	1
12:13:09	44	1
12:13:09	46	2
12:13:09	47	1
12:13:09	49	2
12:13:09	50	1
12:13:09	51	1
12:13:09	52	1
12:13:09	53	1
12:13:09	54	1
12:13:09	55	1
12:13:09	56	1

Figure 28 The SNs of the sequential frames captured by Host AP

Why did it failed to defend deauthentication flooding attacks to set the SND equal to 1? We launched deauthentication flooding attacks for 100 seconds during the FTP session between the STA and Host AP, recorded the sequence number of the captured frames on the Host AP, and analyzed the variabilty of the sequence numbers. Figure 28 shows a period of the captured frames with sequential sequence numbers. We found that the sequence numbers did not increase by 1 for every subsequent frames; instead, sometimes it is increased by (see the shadowed records in Figure 28).

In Figure 29, we reordered the same data by the sequence of SND, Time and SN column. We found the sequence number even was increased by 3 for the subsequent frame. We can neglect the “Jump” value where the SND is under 0, because the value

of SN must be modulo 4096. The “Jump” column is computed as the SN of a certaim record minus the SN of the previous one. It means how often the “SND=2” or “SND=3” happened.

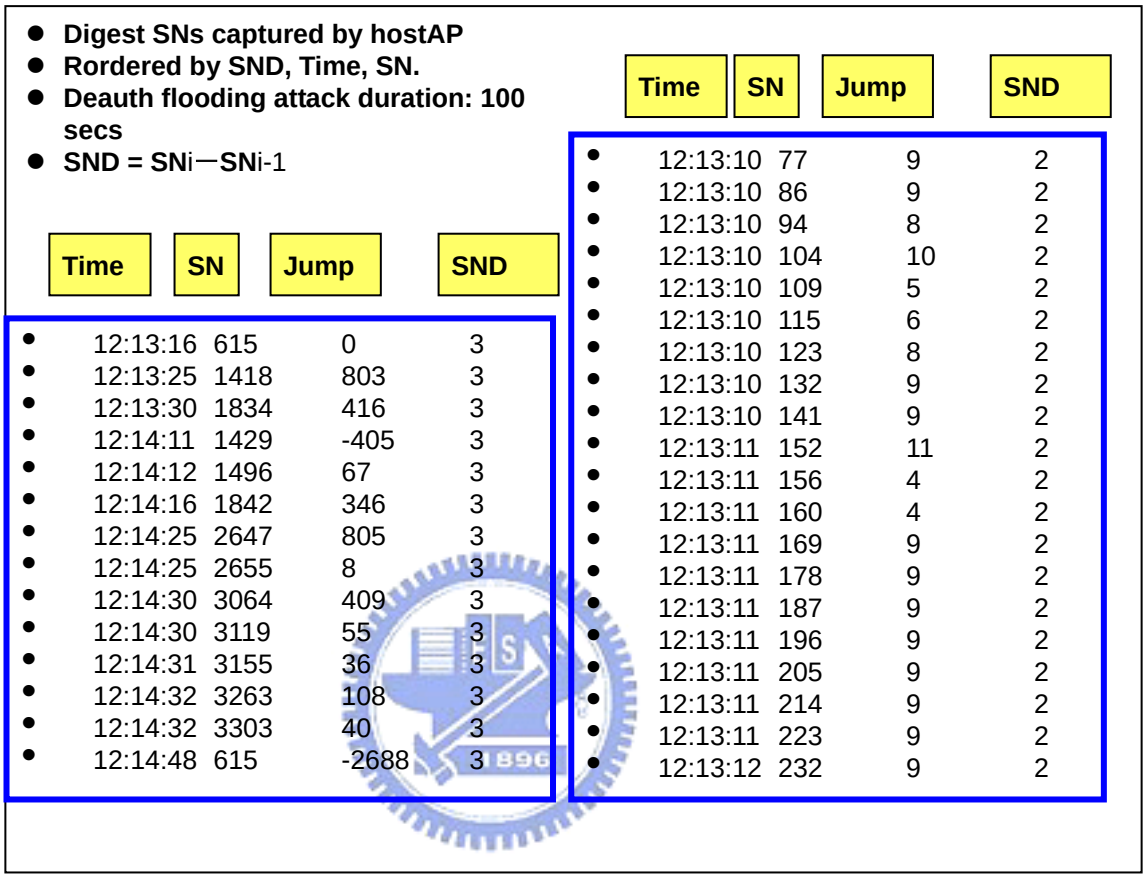


Figure 29 Reorded SND record

We do not know whether the value of SND that is more than 3 would happen or not. We focus on whether we can defend deauthentication flooding attacks by fitering out the frames with the squential SN only. Normally, the STA or AP will send only one deauthentication notification to inform the other connected node to disconnect the session. Even if the deauthentication frame is lost, the sending node would retransmit the frame with the same SN, because it did not receive the ACK frame from the receiving node. Why does the sending node transmit deauthenticaiton frames with sequential SN using the same MAC address? We highly suspect that these frames are forged frames. If we detect the sequential deauthentication frames, we can process

them as the forged frames and drop them directly.

According to the above analysis and the records in Table 12, would the deauthentation flooding attacks be defended effectively if we filtered out the sequential frames while the SND is equal to or more than 8? The answer is no. The attacking tool, void11, emits forged frames frequently as we described before. We can easily modify the programs of void11 to send forged frames with SNs not fitting the filter criterion.. In actuality, we modified void11 to send the number of SND association request frames, even though the association request frames were malformed packets, and then to send one deauthentication or disassociation frame. The SN of the deauthentication or disassociation frame will skip over the value of SND, and the frame would not be filtered by the victim node.

We implemented the filtering mechanism described above and launched attacks with the modified void11. We configured Host AP to filter out the subsequential forged deauthentication or disassociation frames under and equal to the value of SND. On the other hand, the attacker (void11) circularly emitted one deauthenticaiton or disassociation frame after sending the number of SND association request frames. Then, the forged frames would not be filtered out and is processed as legal frames. We exploited the method to implement the deauthenticaiton and disassociation flooding attacks against the Host AP. On the Host AP node we implement SN filtering mechanism. We got the result in Table 13.

SND	8	16	24	32	64
FTP duration under Deauth flooding attack (Avg of 10 times)	60.8	58.5	58.7	54	44.7
FTP duration under Disassoc flooding attack (Avg of 10 times)	60.6	58.5	59.3	54	46.1
Delay (1)	22.3	20.0	20.2	15.5	6.2
Delay (2)	22.1	20.0	20.8	15.5	7.6

Table 13 Filter out sequential SN to defend Deauth / Disassoc attack

Note: 1.Delay (1) is to compare the FTP duration under Deauth attacks with duration under bandwidth consumption of malformed Assoc request flood.

2.Delay (2) is to compare the FTP duration under Disassoc attacks with duration under bandwidth consumption of malformed Assoc request flood.

In Table 13, we found that even if SND was equal to 64, the Host AP cannot defend the deauthenticaiton and disassociation flooding attacks launched by the modified void11. The value of Delay (2) is 7.6 seconds under disassociation flooding attacks while SND is equal to 64. The results were illuminated in Figure 30.

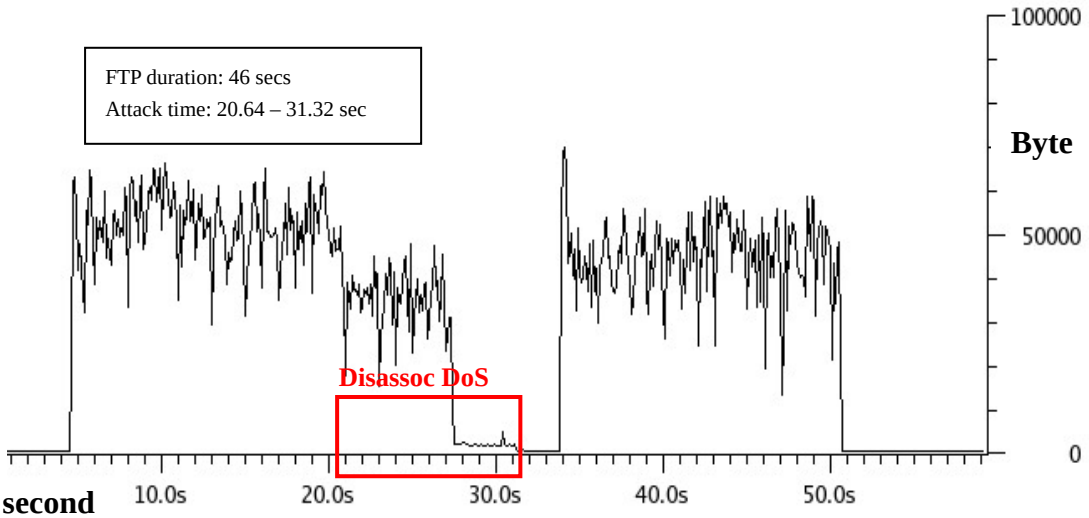


Figure 30 Filter out the Disassoc flooding frames of which SND was under or equal to 64

As shown in Figure 30, filtering out the forged frames with SND under 64 is not enough to defend disassociation flooding attacks. We do not repeat the same

experiment with the value of SND more than 64, because it will affect the normal node to send genuine deauthentication or disassociation frame to disconnect the session.

5.2.4 Two-phase filtering mechanism

We analyzed random bit authentication in section 5.2.2. As the result, we must insert at least 8 bits to 802.11 frames to effectively defend the deauthentication and disassociation flooding attacks with random bit authentication mechanism. However, as explained in section 4.3.2, we can only insert a maximum of 4 bits to the header of the deauthenticaiton and disassociation frames. If we want to insert more than 4 bits to the 802.11 frames, we must use the frame body fields, and the 802.11 standard must be modified. For the minimally modified and backward compatible considerations, we exploited the unused bits in the header of the 802.11 authentication and disassociation frames.

Since we can only insert 1 to 4 random bits for authentication to the frame control field of the 802.11 MAC header, and we cannot defend deauthentication and disassociation flooding attacks effectively by utilizing the sequence number filtering mechanism only. We can combine these two mechanisms to defend 802.11 DoS attacks.

First, we filtered out the forged frames of which SNs were within the value of SND, and second, we examined the random bits and dropped the frames of which random bits did not match the ones of the receiver (Host AP). On the attacking side (void11), the attacking node circularly transmitted one deauthentication or disassociation frame after sending SND association request frames, and all the sending frames including deauthentication, disassociation or association request frames has the Bit15 in the frame control field set to 1 every 2^{RBN} times. RBN is the number of Random Bit Number.

RBN \ SND / Duration		4	8	12	16	20	24	28	32
3	**Deauth flooding attack**	58.3	49	43.8	42.8	41.8	39.1	39.1	38.9
	Disassoc flooding attack	58.1	47.7	43.6	43.2	41.1	39.7	39.2	38.8
	Delay (1)	19.8	10.6	5.4	4.4	3.4	0.7	0.7	0.4
	Delay (2)	19.6	9.2	5.1	4.7	2.6	1.2	0.7	0.3
4	**Deauth flooding attack**	47.7	43.2	39.1	37.7	38.0	37.8	37.7	37.9
	Disassoc flooding attack	44.9	44.3	39.4	37.9	38.1	37.9	38.0	37.7
	Delay (3)	9.3	4.8	0.7	-0.8	-0.5	-0.7	-0.8	-0.6
	Delay (4)	6.4	5.8	0.9	-0.6	-0.4	-0.6	-0.5	-0.8

Table 14 Two-phase filter to defend Deauth / Disassoc attack

Note:1.Delay (1) is to compare the FTP duration under Deauth attacks with duration under bandwidth consumption of Assoc request flood.

2.Delay (2) is to compare the FTP duration under Disassoc attacks with duration under bandwidth consumption of Assoc request flood.

3.Delay (3) is to compare the FTP duration under Deauth attacks with duration under bandwidth consumption of Assoc request flood.

4.Delay (4) is to compare the FTP duration under Disassoc attacks with duration under bandwidth consumption of Assoc request flood.

5.Acronym RBN is Random bit number.

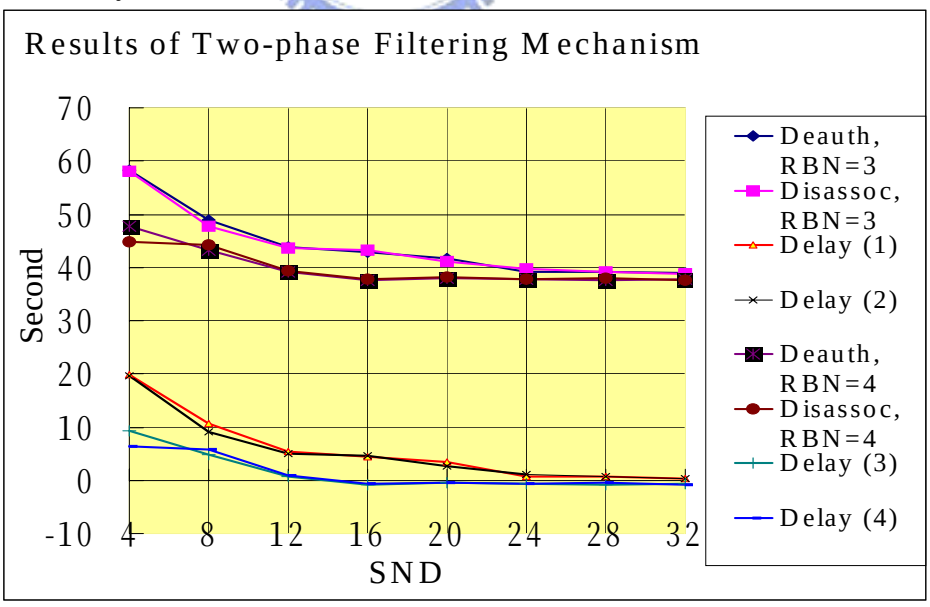


Figure 31 Graph of results according to table 14

We tested the two-phase filter mechanisms where the value of RBN ranged from 3

to 4 and the SND ranged from 4 to 32. We got the result as showed in Table 14 and Figure 31.

As shown in Table 14 and Figure 31, if RBN is equal to 3 and SND is equal to 24, or RBN is equal to 4 and SND is equal to 12, we can effectively defend deauthentication and disassociation flooding attacks. Figure 32 is the graph of FTP duration under disassociation flooding attacks when RBN is equal to 3 and SND equal to 24. Figure 33 is the graph of FTP duration under disassociation flooding attacks when RBN is equal to 4 and SND equal to 12.

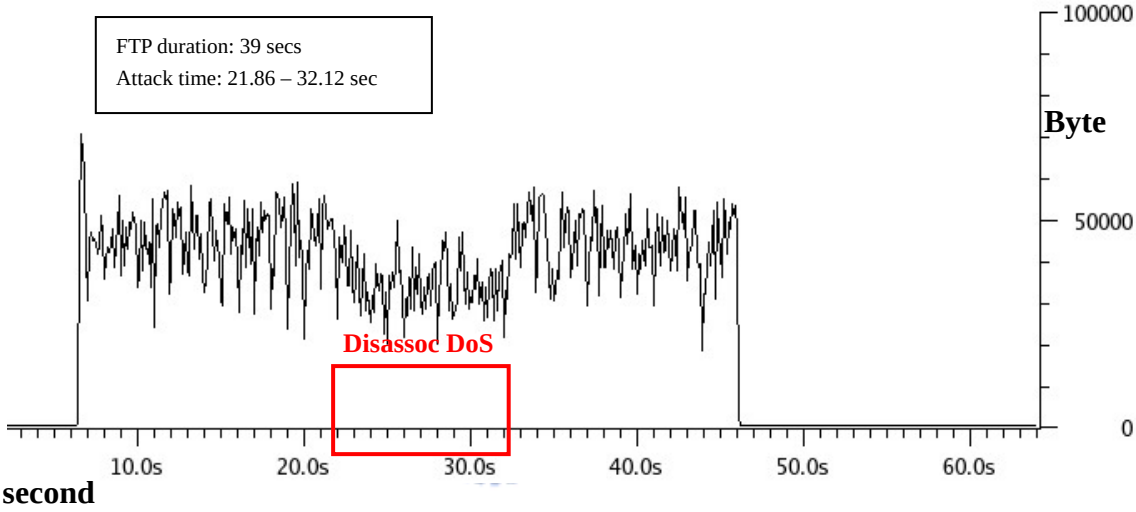


Figure 32 FTP duration under Disassoc flooding attacks on condition of RBN=3 & SND=24

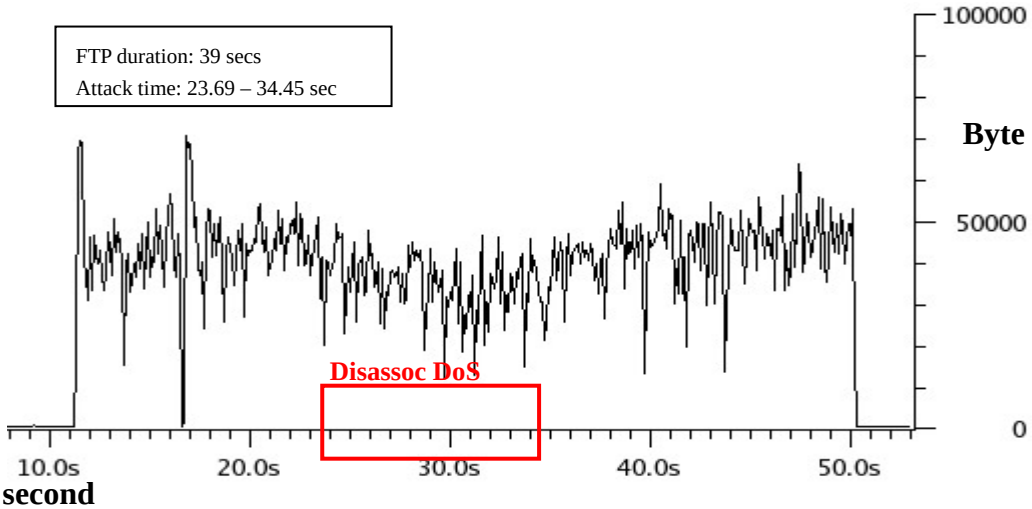


Figure 33 FTP duration under Disassoc flooding attacks on condition of RBN=4 & SND=12

The two-phase filter mechanism can effectively defend modified void11

deauthentication and disassociation flooding attacks. Both Figure 32 and Figure 33 demonstrate that deauthentication and disassociation flooding attacks are defended effectively in our implement.

5.2.5 Applying two-phase filter mechanism to the shared key authentication

We implement the previous two-phase filter mechanism on the open system authentication so far. The 802.11 specification defines open system and shared key authentication. We would like to apply the mechanism to the shared key authentication. We were confronted with some limitations on our Linux-based systems.

We configured the 802.11b cards with the command “iwconfig” to become restricted mode (shared key authentication). But, when we monitored the negotiation of the authentication frames captured by kismet between Host AP and STA, we found the STA was authenticated by Host AP in open system authentication, and they communicated with each othe well, even if we set both STA and Host AP in shared key authentication mode. If we set STA in open system mode, but Host AP in shared key authentication mode, both of them communicated with each other well also. We found the authentication process also in open system mode.

To launch deauthentication flooding attacks in shared key authentication mode, we alternatively installed the 802.11b card of STA on Windows XP system, but Host AP is still Linux based. We found the STA was successfully authenticated by Host AP in shared key authentication mode. Then we applied the two-phrase filter mechanism and deauthentication flooding attacks to this Linux-Windows architecture.

RBN & SND / FTP duration	DoS attacks with no defense	RBN = 3 SND = 24	RBN = 4 SND = 12

Deauth flooding attack	53.42	42.36	43.02
Disassoc flooding attack	53.06	40.48	42.91
Duration under bandwidth consumption of malformed Assoc request flood	41.19	41.19	41.19
Delay (1)	12.23	1.15	1.83
Delay (2)	11.87	-0.71	1.72

Table 15 FTP duration under Deauth and Disassoc flooding attacks on shared key authentication mode

Note:1. Normal FTP duration is 36.16 seconds

2. The delay (1) is comparing the FTP duration under Deauth attacks with duration under bandwidth consumption of malformed Assoc request flood.

3. The delay (2) is comparing the FTP duration under Disassoc attacks with duration under bandwidth consumption of malformed Assoc request flood.

As shown in Talbe 15, our two-phase filter mechanism effectively defended the deauthentication and disassociation flooding attacks. The results were similar to the results in section 5.2.4. Figure 34 illustates the results.

We circle the 13th to 15th second area where no data is transmitted in Figure 34. We must mention that it was not caused by deauthetication flooding attacks which happened during the 24th to 34th second area marked by rectangle.

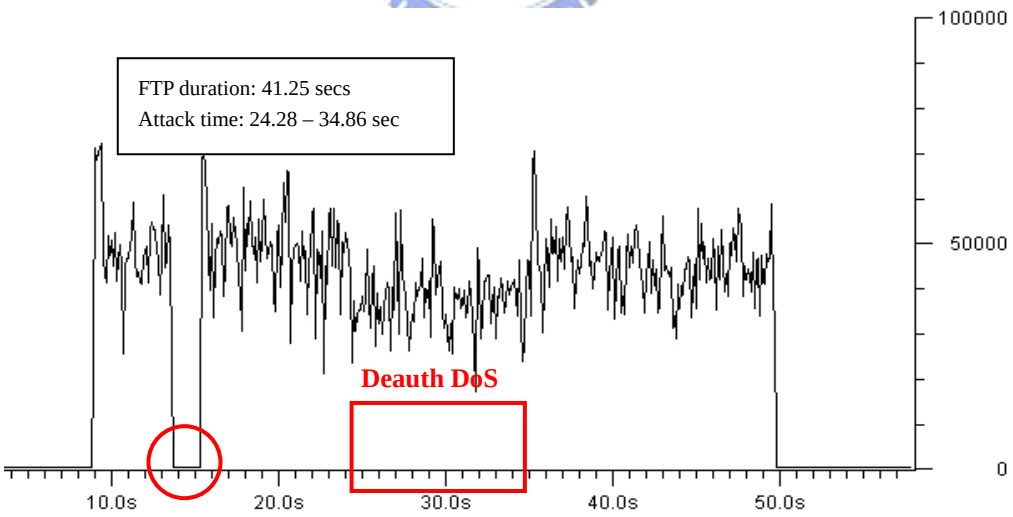
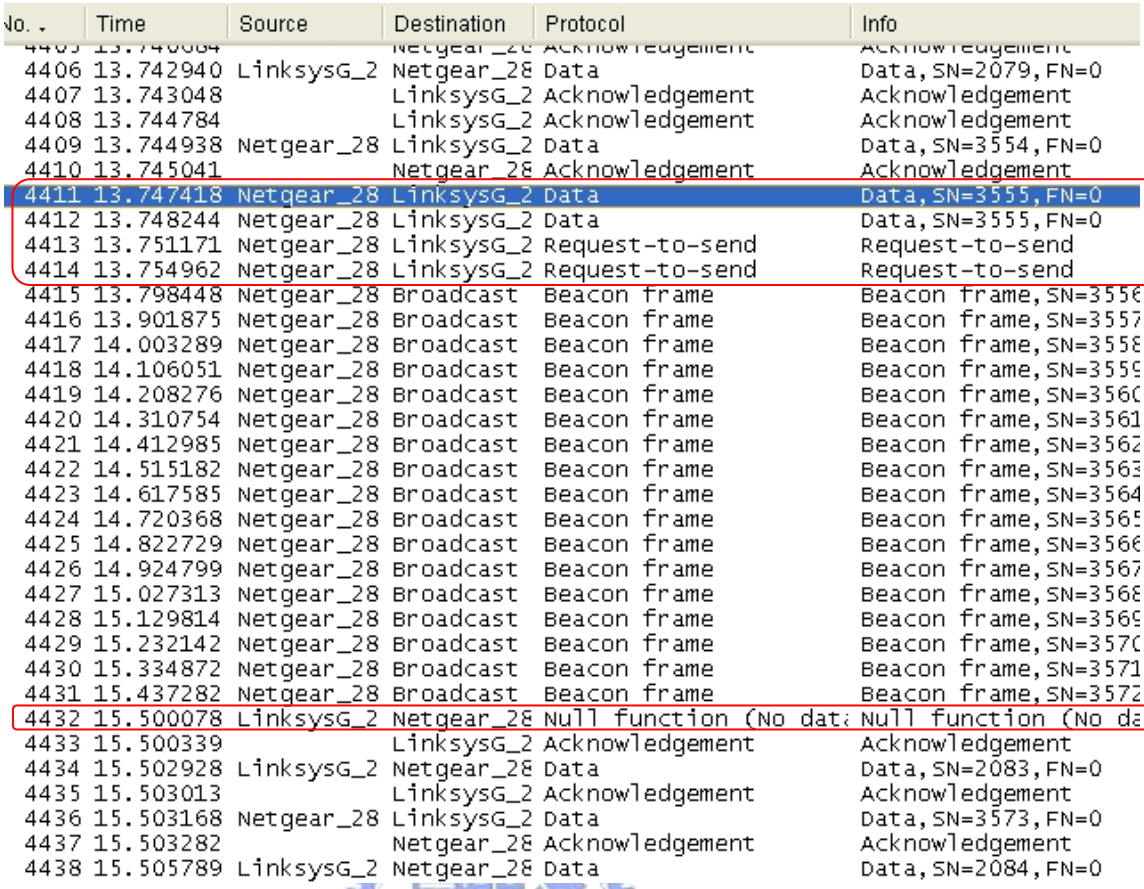


Figure 34 FTP duration under Deauth flooding attacks in windows STA on condition of RBN=3 & SND=24



No.	Time	Source	Destination	Protocol	Info
4406	13.742940	LinksysG_2	Netgear_28	Data	Data, SN=2079, FN=0
4407	13.743048		LinksysG_2	Acknowledgement	Acknowledgement
4408	13.744784		LinksysG_2	Acknowledgement	Acknowledgement
4409	13.744938	Netgear_28	LinksysG_2	Data	Data, SN=3554, FN=0
4410	13.745041		Netgear_28	Acknowledgement	Acknowledgement
4411	13.747418	Netgear_28	LinksysG_2	Data	Data, SN=3555, FN=0
4412	13.748244	Netgear_28	LinksysG_2	Data	Data, SN=3555, FN=0
4413	13.751171	Netgear_28	LinksysG_2	Request-to-send	Request-to-send
4414	13.754962	Netgear_28	LinksysG_2	Request-to-send	Request-to-send
4415	13.798448	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3556
4416	13.901875	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3557
4417	14.003289	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3558
4418	14.106051	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3559
4419	14.208276	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3560
4420	14.310754	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3561
4421	14.412985	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3562
4422	14.515182	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3563
4423	14.617585	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3564
4424	14.720368	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3565
4425	14.822729	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3566
4426	14.924799	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3567
4427	15.027313	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3568
4428	15.129814	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3569
4429	15.232142	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3570
4430	15.334872	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3571
4431	15.437282	Netgear_28	Broadcast	Beacon frame	Beacon frame, SN=3572
4432	15.500078	LinksysG_2	Netgear_28	Null function (No data	Null function (No da
4433	15.500339		LinksysG_2	Acknowledgement	Acknowledgement
4434	15.502928	LinksysG_2	Netgear_28	Data	Data, SN=2083, FN=0
4435	15.503013		LinksysG_2	Acknowledgement	Acknowledgement
4436	15.503168	Netgear_28	LinksysG_2	Data	Data, SN=3573, FN=0
4437	15.503282		Netgear_28	Acknowledgement	Acknowledgement
4438	15.505789	LinksysG_2	Netgear_28	Data	Data, SN=2084, FN=0

Figure 35 Captured frame digest in FTP duration under Deauth flooding attacks in windows STA on condition of RBN=3 & SND=24

5.3 Discussion and limitation

5.3.1 Discussion

We implemented the two-phase filter mechanism to defend 802.11 DoS attacks and the mechanism has some benefits. First, the implementation is 802.11 backward compatible. It means when you utilize the mechanism, you can just update the firmware of the legacy (b, g) devices to solve the security issues for defending deauthentication and disassociation flooding attacks.

Second, the beneficial feature of our defending mechanism is that it utilizes the sequence number sub-field of the 802.11 MAC header, and the main structure of the

802.11 frame is not modified. So, the 802.11 (b, g) would only be modified in small ways to elevate the security of defending deauthentication and disassociation flooding attacks.

Third, the other advantage of the two-phase filter mechanism is the lightweight computing feature. We do not implement complex encryption algorithm. More complex encryption algorithm consumes more computing and memory resources, and maybe leads to other type of resource-exhausted DoS attacks.

For the 802.11i standard, you can also utilize our defending mechanism to the standard. As we discussed in chapter 3, the DoS attacks issues against 802.11i may not be considered properly. 802.11i followed the 802.11 specification. 802.11i did not delete or encrypt (authentication) the 802.11 deauthentication and association processes before 802.1X authentication and key negociations. 802.11i preserves 802.11 authentication processes and just define it as open system. These problems make 802.11i vulnerable to the DoS attacks. Our defending DoS mechanism can be utilized to solve such problems.

5.3.2 Limitation

Our two-phase mechanism is suitable to be implemented in prevailing 802.11 (b, g) equipment as we disussed in 5.2.1. We can use the existing WEP key for preliminary DoS security issues, though we know WEP key is easy cracked. Even though we did not discuss the key exchange issues, it should be researched in the future.

802.11i specification defined pre-shared key and other key exchange issues. We can exploit pre-shared key just as the WEP key does. If we want to utilize the existing 802.1X key handling architecture, we would need to modify the 802.11 standard documents. The association negociation between STA and AP should be processed after 802.1X handshakes has completed while the STA and AP shared some secret. We

exploit the shared secret to generate the same random bit string; then our ramdom bit authentication mechanism would work smoothly.

Our experiments have some inevitable deviation. We test the implemention in a common laboratory. Some radio noises may happen and disturb our radio equipment. The normal FTP duration with no attacks launched, varied even on the same equipment. To minimize variation error, we must repeet each experiment at least 10 times, and average the results. We did not use sophisticated equipment to detect the stability of the radio in our labortary. If we did so, maybe the deviation could be limited.

Chapter 6 Conclusion and future work

6.1 Conclusion

We designed an efficient mechanism to defend the deauthentication and disassociation flooding attacks against 802.11 and 802.11i networks. With the integration of both the random bit authentication and the sequence number filtering mechanism, our approach is robust in defending deauthentication and disassociation flooding attacks. Our experiments proved that the two-phase filter mechanism is feasible.

Our designs are not only suitable to defend the deauthentication and disassociation flooding attacks, but also can be adopted to defend similar DoS attacks. For example, the EAPOL-Failure message and EAPOL-Logoff message attacks are such attacks in 802.11i network. Such messages are send in clear over the air. There is no key-based authentication to protect these frames. If we utilize the two-phase filter mechanism to defend such DoS attacks, the clear messages will be authentication with the random authentication bits. The DoS attacks caused by forged EAPOL-Failure and EAPOL-Logoff will be defended or alleviated.

The resources of the wireless devices, including APs and wireless network cards, are limited. Our random bit authentication mechanism utilizes the unused bits of the control field in the 802.11 MAC header. It is simplier than those that implemented complex encryption and decryption mechanisms to protect the frames. The consumption of the computing and memory resources is lightweight. One of the superior advantage is that our designs will not give rise to other kinds of

resource-exhausted DoS attacks.

802.11 devices are prevailing over the word. The cost to upgrade all the deployed 802.11 hardwares is heavy for a company or corporation. Our design is backward compatible. We can upgrade the firmware of the deployed devices to improve the vulnerabilities to the DoS attacks.

The 802.11 security issues are important. 802.11i improves the issues of the data integraty and confidentiality in wireless network. The availability of the network is basic and an important issue. The DoS attacks against the WLANs should not be neglected.

6.2 Future work

Our experiments are implemented in 802.11b nework. We analyzed the similar vulnerabilities of the DoS attacks in 802.11i, but did not really implement the attacks. We will implement or simulate our designs to defend EAPOL-Failure and EAPOL-Logoff DoS attacks in 802.11i network. These frames are transmitted in clear and easily spoofed by the attacker. The vulnerability of such messages is similar to the deauthentication and disassociation attacks.

As for the limitation of the hardware decribed in section 4.3.2, we could not insert random authentication bits into the control field of the 802.11b MAC header in the managed mode. Some critical fuctions of the management frames were implemented in the firmware. We will implement our designs in the firmware, we forcast that our implementation will be successful.

Reference

[1] IEEE Standard 802.11i. “Local and metropolitan area networks- Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 6: Medium Access Control (MAC) Security Enhancements”. IEEE Std 802.11i-2004

[2] IEEE Standard 802.11. “Local and metropolitan area networks- Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Security Enhancements”. ANSI/IEEE Std 802.11, 1999 Edition.

[3] D. Chen, J. Deng,, P. K. Varshney. “Protecting wireless networks against a Denial of Service attack based on virtual jamming”. In Poster Session of MobiCom2003, San Diego, CA, September, 2003.

[4] Jesse R. Walker. “Unsafe at any key size: an analysis of the WEP encapsulation”. Tech. Rep. 03628E, IEEE 802.11 committee, March 2000, http://grouper.ieee.org/groups/802/11/Documents/DocumentHolder/0-362.zip.

[5] Stubblefield, A., Ioannidis, J., and Rubin, A. “Using the Fluhrer, Mantin, and Shamir attack to break WEP”. In Proceedings of the 2002 Network and Distributed Systems Security Symposium, 2002, pages: 17–22.

[6] Changhua He, John C Mitchell. “Security analysis and improvements for IEEE 802.11i”. Network and Distributed System Security Symposium Conference Proceedings, 2005, http://www.isoc.org/isoc/conferences/ndss/05/proceedings/papers/NDSS05-1107.pdf

[7] Mike Kershaw. Kismet 2005-08-R1, 2005. http://www.kismetwireless.net.

[8] Reyk Floeter. Wireless lan security framework: void11. 2002, http://www.wlsec.net/void11/.

[9] Jouni Malinen. Host AP driver for Intersil Prism2/2.5/3, hostapd, and WPA Supplicant, 2005, http://hostap.epitest.fi/.

[10] Scott Fluhrer, Itsik Mantin and Adi Shamir. “Weaknesses in the key scheduling algorithm of RC4”, Eighth Annual Workshop on Selected Areas in Cryptography, August 2001.

[11] Henric Johnson, Arne Nilsson, Judy Fu, S. Felix Wu, Albert Chen and He Huang. “SOLA: A one-bit identity authentication protocol for access control in IEEE 802.11”. GLOBECOM, IEEE Global Telecommunications Conference, vol. 21, no. 1, November 2002, pages: 777–781.

[12] H. Wang, A. Velayutham, and Y. Guan. ”A Lightweight Authentication Protocol for Access Control in IEEE 802.11”. In Proceedings of IEEE Globecom 2003, San Francisco, CA, December 1-5, 2003.

[13] J. Bellardo, and S. Savage. “802.11 Denial-of-Service attacks: real vulnerabilities and pra

ctical solutions". In Proceedings of the 12th USENIX Security Symposium, Washington, D.C., August 4-8, 2003.

[14] Ferreri F., Bernaschi M., Valcamonici L. "Access points vulnerabilities to DoS attacks in 802.11 networks". Wireless Communications and Networking Conference, Vol. 1, March 2004, pages: 634–638.

[15] Changhua He, John C. Mitchell. "Analysis of the 802.11i 4-way handshake". In Proceed ings of the 2004 ACM workshop on wireless security, ACM Press, New York, USA, 2 004, Pages: 43–50.

[16] Bernard Aboba. "Issues in pre-standard IEEE 802.11i implementations". http://www.drizzl e.com/~aboba/IEEE/prestand.html.

[17] Airsnort, http://airsnort.shmoo.com/.

[18] Anton T. Rager. WEPCrack, http://wepcrack.sourceforge.net/.

[19] Ethereal, http://www.ethereal.com/.

[20] Kui Ren, Hyunrok Lee, Kyusuk Han, Park, J., Kwangjo Kim. "An enhanced lightweight authentication protocol for access control in wireless LANs". In Proceedings of 12th IE EE International Conference on Volume 2, Nov 16-19. 2004, Pages: 444–450.

[21] Ping Ding, JoAnne Holliday, Aslihan Celik. "Improving the Security of Wireless LANs by Manag ing 802.1X Disassociation", In Proceedings of the IEEE Consumer Communications and Networkin g Conference, Las Vegas, NV, January 2004.

[22] Boland, H.; Mousavi, H. "Security issues of the IEEE 802.11b wireless LAN". Electrical and Computer Engineering, Canadian Conference, Vol 1, May 2004 Pages: 333-336.

[23] Joshua Wright. "Detecting Wireless LAN MAC Address Spoofing". GCIH, CCNA, 2003, http://home.jwu.edu/jwright/papers/wlan-mac-spoof.pdf.

[24] Chibiao Liu. "802.11 Disassociation Denial of Service attacks". http://www.mnlab.cs.depa ul.edu/seminar/spr2005/WiFiDoS.pdf.

[25] Jodi Haasz. "Re: P802.11w - Amendment to Standard [FOR]. Information Technology-Telecommunications and Information Exchange between systems-Local and Metropolitan networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) a nd Physical Layer (PHY) specifications: Protected Management Frames", IEEE 802.11w approved letter, March 22 2005, http://standards.ieee.org/board/nes/projects/802-11w.pdf.

[26] Jesse Walker. Status of Project IEEE 802.11 Task Group w: Protected Management Fra mes. http://grouper.ieee.org/groups/802/11/Reports/tgw_update.htm.

[27] Wikipedia, the free encyclopedia. "IEEE 802.11". http://en.wikipedia.org/wiki/802.11.

[28] IEEE P802.11-TGn, Status of Project IEEE 802.11n: Standard for Enhancements for Hig her Throughput, http://grouper.ieee.org/groups/802/11/Reports/tgn_update.htm.

[29] CERT Coordination Center. "Denial of Service Attacks". http://www.cert.org/tech_tips/deni al_of_service.html.

[30] Wikipedia, the free encyclopedia. "Denial-of-service attack". http://en.wikipedia.org/wiki/D enial-of-service_attack