JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 25, 1485-1500 (2009)

Using Random Bit Authentication to Defend IEEE 802.11 DoS Attacks*

YING-SUNG LEE, HSIEN-TE CHIEN AND WEN-NUNG TSAI
Department of Computer Science and Information Engineering
National Chiao Tung University
Hsinchu, 300 Taiwan

IEEE 802.11 networks are insecure. Wired Equivalent Privacy (WEP), the security mechanism used in 802.11, was proved to be vulnerable. IEEE 802.11i, the security enhancement, concentrates only on integrity and confidentiality of transmitted frames. Either version did not properly handle the network availability. Because management frames are not authenticated, {802.11, 802.11i} networks are susceptible to Denial of Service (DoS) attacks. In this paper, we designed a random bit authentication mechanism as a defense against DoS attacks. Random bits are placed into unused fields of the management frames. Access Point (AP) and station (STA) can then authenticate each other according to these authentication bits. The effectiveness of our mechanism is demonstrated through experimental results.

Keywords: wireless network security, denial of service, lightweight authentication, deauthentication and disassociation flooding attacks, vulnerability

1. INTRODUCTION

Wireless access has become a popular option for network connectivity. Securing this access option, therefore, is essential for the prolonged adoption of this technology. However, Wired Equivalent Privacy (WEP), the security scheme adopted by the original IEEE 802.11 standard, is flawed as a group of researchers had demonstrated in 2001 [32]. Also in the same year, S. Fluhrer, *et al*. [11] described a ciphertext-only attack on RC4 as used in WEP. Not long after, Adam Stubblefield and AT&T were the first to publicly verified Fluhrer's attack [6].

Due to the growing demand to plug the security hole, IEEE established a dedicated task group to strengthen the security scheme, which becomes the future 802.11i amendment. Meanwhile, the Wi-Fi Alliance announced an interim specification, Wi-Fi Protected Access (WPA), roughly based on the then released IEEE 802.11i draft. WPA compliant products started to appear in mid-2003. In June 2004, IEEE 802.11i was finally ratified, and replaced RC4 with the Advanced Encryption Standard (AES).

Security designers generally focus on implementing three security services: confidentiality, integrity and availability. Some researchers found that 802.11i only concentrated on maintaining confidentiality and integrity; availability is of a lesser concern. Even after introducing Robust Security Network Association (RSNA), 802.11i still appears to be vulnerable to Denial of Service (DoS) attacks [7].

However, some of that vulnerability may be inherent in the protocol design. IEEE 802.11 standard utilizes management frames for transmission preparation. Yet, these

Received November 21, 2007; revised March 10 & May 21, 2008; accepted August 1, 2008.
Communicated by Tzong-Chen Wu.
* The preliminary work of this paper has been presented in International Computer Symposium 2006.

frames are not encrypted and are not authenticated. Therefore, falsified deauthentication and/or disassociation requests could successfully deny services from legitimate users.

This paper is organized into five sections. Section 2 briefly discusses some researches on DoS attacks in 802.11 networks. In section 3, we present our proposed protocol that could minimize the damage caused by deauthentication and disassociation flooding attacks. Next, our experimental results are summarized in section 4. Finally, we conclude our contributions and provide some directions for future works in section 5.

2. RELATED WORKS

Wireless networks are especially susceptible to DoS attacks. Many researchers have already discovered numerous attacking strategies. In this section, we summarize some of their findings and proposed defenses.

2.1 DoS Attacks Against 802.11 Networks

Because management and control frames are not protected, adversaries could exploit this fact to launch DoS attacks on 802.11 networks. The most efficient exploit is to flood the surroundings with huge amounts of deauthentication or disassociation frames.

2.1.1 Deauthentication and disassociation flooding attacks

An 802.11 station (STA) must first authenticate and then associate itself with an access point (AP) before communication occurs. Stopping the communication requires either the STA or the AP to request deauthentication. Unfortunately, deauthentication is not cryptographically protected. Consequently, spoofed deauthentication messages could easily disconnect any active connection. Similarly, disassociation messages can also be exploited in the same way.

Since the receiver would not verify the source of deauthentication/disassociation notification messages, an attacker could effortlessly convince the receiver those notifications come from the AP or an already authenticated STA. According to the 802.11 protocol, the receiver must respond immediately by leaving the authenticated/associated state. Communication is disrupted until the affected STA re-associates itself with an AP. However, as long as the attacker keeps broadcasting deauthenticaiton/disassociation frames, re-authentication/re-association may be difficult or even impossible.

J. Bellardo and S. Savage have shown in [14] that delaying responses to deauthentication or disassociation requests can be a good defense. Receivers now would observe subsequent packets from their communication partners. Since a legitimate node would never send more data after making deauthentication or disassociation requests, data packets, if received, indicates a bogus request that should be discarded.

However, their approach is not without drawbacks. Not only did their defensive measure introduce longer delay in the handoff process but also enable an attacker to hijack a legal session by injecting forged data frames that cancels valid disconnection requests. The hijacker would then be able to assume the role of the disconnecting node and to start transmitting data through the AP undetected.

2.1.2 Traffic jamming DoS attacks

Traffic jamming attacks deny services to legal users by trying to exhaust resources on network devices. Since management frames must be processed once received, computing and/or memory resources are easily exhausted through this type of attacks.

F. Ferreri, *et al*. [15] developed a simple attacking tool, wfit (wireless frame injection tool), based on the Radiate library which was built on top of an old version of the HostAP driver. Their attacking tool was used to launch probe, authentication and association flooding attacks against 4 models of APs (Enterasys RoamAbout R2, Netgear ME102, 3COM AP 8000 and HostAP). Their experiments showed that most models experience serious performance degradation when attacked.

The authors failed their attempt to enhance Linux HostAP driver with built-in detection and defense capability as they found that the exploit enabling probe flooding attacks were at the firmware level, and therefore could not be mitigated at the software driver level. No better solutions were suggested in their paper.

Traffic jamming attacks also includes virtual carrier sense attacks [4], power saving mode attacks, and many others mentioned in [15].

2.2 DoS Attacks Against 802.11i Networks

Since management frames continue to be unprotected under 802.11i deployment, the DoS attacks described in section 2.1 still threaten 802.11i networks. C. He, *et al*. even claimed in [7] that severity for DoS threats increases when 802.11i security measures were applied. During their security analysis on the message flow in Robust Security Network Association (RSNA) procedure, they found that RSNA establishment during network capability discovery, authentication, and association stages could be disrupted as key management frame exchanges are not protected. In the same way, EAPOL (Extensible Authentication Protocol over LAN)-Start, EAPOL-Success, EAPOL-Logoff, and Deauthentication/Disassociation messages, due to their no-key-based-authenticated nature, are playground for the attackers to explore.

2.2.1 Deauthentication and disassociation attacks against 802.11i network

Ding, Holliday and Celik [22] devised an anti-DoS application for 802.1X enabled 802.11i network environment. Their Central Manager (CM) application, acting like an authentication server (AS), also tracks STAs to further mitigate DoS attacks.

Their design requires that an AP to forward any disassociation frames received from STAs to the CM. Then, the CM will contact the requesting STA to verify its intention to disassociate. The AP only processes the disassociation requests once the STA confirms its intention. However, their design does not consider the fact that disassociation frames could come from an AP. In such a case, the AP will not receive the disassociation frames and thus cannot forward them to the CM for verification. Furthermore, current authentication servers, as well as the 802.11i standard, would need to be modified. Their scheme would not work in 802.11 networks that do not implement 802.1X.

2.2.2 EAPOL-failure and EAPOL-logoff message attacks

Several DoS attacks exploit the unprotected EAP (Extensible Authentication Protocol) messages in 802.1X authentication procedure. For example, EAPOL-Failure messages and EAPOL-Logoff messages could be forged to disconnect the supplicant [15, 16, 23]. EAPOL-Failure messages force the receiving STA to stop negotiating with the network nodes. Similarly, EAPOL-Logoff messages cause the receiving AP to log off the sending STA. Hence, by jamming the network with these two message types, the attacker could deny useful communication between the AP and STAs.

P. Ding, *et al.* [22] also suggested their Central Manager (CM) as a way to mitigate EAPOL-Failure and EAPOL-Logoff message attacks. However, this solution also shares the same disadvantages as the CM approach mentioned earlier.

2.3 One-bit Lightweight Authentication

Statistical One-bit Lightweight Authentication (SOLA) protocol [12] is an identity authentication protocol proposed to detect unauthorized access in 802.11 networks. STAs and the AP would need to agree on a secret key used to generate identical random authentication stream. The communicating parties then successively extract one bit from this stream and insert it into the MAC layer header for identity authentication.

The major purpose of SOLA protocol is to detect an attack. SOLA protocol offers a statistical way to identify the origin of the packets. The authors claim that the SOLA protocol is well suited in a resource-constrained wireless environment. Furthermore, developing a SOLA-based framework for detecting DoS attacks is possible. The framework developed could also discover adversaries who attempt to attack the network by guessing the identity authentication bit.

H. Wang, *et al.* [13] followed up on the lightweight authentication idea, but criticized the synchronization algorithm of [12]. They developed a workable synchronization algorithm and incorporated the algorithm into an IEEE 802.11 network environment. They concluded that their authentication mechanism is light-weight, simple, continuous-authenticating, highly efficient, and fault tolerant.

Both research groups focused on synchronization algorithms and the statistical analysis. Implementation issues are not sufficiently considered.

2.4 Enhanced Lightweight Authentication

K. Ren, *et al.* [21], pointed out that synchronization problems exist in Johnson's lightweight authentication work. The problems are attributed to the frame loss in the wireless networks. They also criticized the inefficiencies of the protocol designed by H. Wang *et al.* [13].

In [21], the same research team also proposed an enhanced lightweight authentication protocol for wireless LANs. They analyzed the redundancy in the MAC header, and found that 3 bits are available. To begin communicating, a common random bit stream generator is shared between the sender and the receiver. A 3-bit unit is extracted from the generated random bit stream for each outgoing frame and the unit is inserted into the control field before transmitting that frame. The legitimate receiver is able to verify the

3-bit authentication value stored in the incoming data frame since the receiver could generate identical random bit stream as the sender could.

The authors of [15] offered another statistical way to identify data origin for detecting attack scenarios. They asserted that the protocol is fully compatible with current IEEE 802.11 frame structure and provides a highly efficient identity authentication scheme.

3. PROPOSED PROTOCOL TO MITIGATE 802.11 DOS ATTACKS

In this section, we describe our design of a random bit authentication mechanism that is capable of protecting an 802.11 network from deauthentication and disassociation flooding attacks. Before our design is explained in section 3.2, 802.11 management frames are first analyzed in section 3.1 to determine the available header fields that can be used to insert random authentication bits.

3.1 Unused Bits of 802.11 Management Frame

IEEE 802.11 standard defines three types of frames: management, control and data. Despite their types, all frames share the same structure that includes a 2-octet frame control field, and a 0-2312 octet(s) frame body field. In this section, the frame control and frame body fields of management frames are analyzed to determine the superfluous bits that could be utilized in our proposed scheme.

3.1.1 Management frame control field analysis.

Fig. 1 illustrates the frame control field in a management frame. The type and the subtype fields need values to indicate the frame type. “To DS” and “From DS” fields are both zero as management frames neither originate from nor terminate on distribution systems. “More Fragments” may happen for some management frames. “Power Management” and “More Data” are to indicate the power management mode of a STA. The “Order” bit would be set if transferring data frames with Strictly Ordered service class 2.

After the brief analysis, the Retry and the WEP bits in the control field are the only two bits left for random bit insertion.

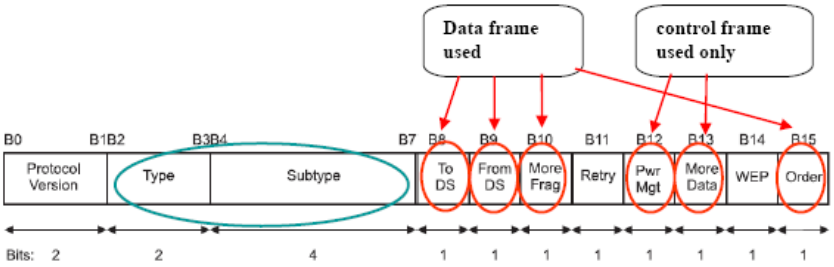

Fig. 1. Management frame control field in use [3].

3.1.2 Management frame body analysis

Management frames are also classified into many subtypes. In this section, only the subtypes pertaining to the authentication and association process are analyzed since only they are relevant to the scope of this paper.

The frame body of the authentication subtype consists of authentication algorithm number, authentication transaction sequence number, status code, and a challenge text. Even though 16 bits are reserved for authentication algorithm number, only two values are currently defined in the 802.11 specification. Thus, 15 bits are redundant for this field. Similarly, the authentication transaction sequence number has 16 bits reserved, but 13 bits are redundant. Status code and challenge text have values that are dependent on the authentication algorithm used. Hence, they will not be considered in our scheme to avoid any side effect.

The frame body of the deauth/disassoc subtype includes only a 16-bit reason code. The reason codes for values 1-9 are defined in 802.11 standard, and they are expanded to require 5 bits to represent in 802.11i addendum. Hence, we only recruit 11 bits in these subtypes to maximize applicability of our proposed scheme.

Regarding the association and reassociation management frames, the capability information is common in all the request and response frame bodies. According to the 1999 edition of the 802.11 specification, that field does not use bit number 5 to 15. Hence, at least eleven bits are available in these frame subtypes.

Based on the analysis in this and the previous sections, the frame control field could donate 2 bits, and the frame body could contribute at most 11 bits. Hence, our DoS defense mechanism could make use of a maximum of thirteen bits to protect the operations of the authentication/association procedures.

3.2 Random Bit Authentication for Management Frame

Some assumptions are made in this paper. Key sharing is assumed to have happened between the communicating nodes. The shared key will be used to generate session keys that can be used to generate a common random bitstream. This paper assumes the communicating partners use the same random number generator for key generation. Diffie-Hellman algorithm is assumed for key exchange. Furthermore, both the sender and the receiver are assumed to have chosen identical bitstream generation algorithm. The algorithm is assumed to be public, while the shared key is kept private.

The nodes in the same basic service set (BSS) utilize the shared session key and algorithm to generate an identical random bit stream independently. The stream is divided into equal-sized chunks, each having "*N*" authentication bits. Let's call it "*N* random bits". Each chunk is given an index number. In our design, only 8 chunks are required in 802.11 (b, g) environment.

The usage of our random bit authentication mechanism is illustrated in Fig. 2. When a node (AP or STA) sends (de)authentication or (dis)association frames, it inserts the current 3-bit unit into the unused bit positions of each frame, and then advances the index to point to the next unit. The receiving node should find that the random authentication bits in the incoming frame matches the corresponding bitstream unit on the receiver; otherwise, the incoming frame is rejected.

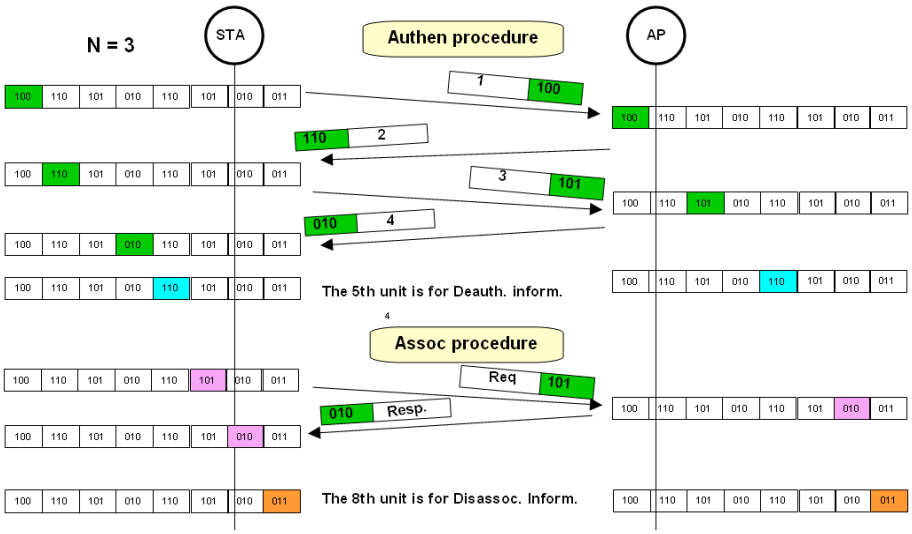

Fig. 2. Random bit authentication for authentication/association procedures.

Consider Fig. 2 once again under the scenario that an attacker is mounting deauthentication or disassociation flooding attacks. Normally, the 5th and the 8th units would be matched for legitimate deauthentication and disassociation frames, respectively. However, the attacker would not know the values for those units for certain; he/she must keep guessing the authentication bits until a match were obtained. Alternatively, the attacker could take a "brute force" approach and cycle through all the possible values of the random bits. Take 3-bit random authentication unit as an example. The attacker can successively substitute the values from 0 to 7 as the authentication bits used in the attacking frame. One out of the 8 spoofed deauth/disassoc frames would pass the authentication test. The success rate of an attacker to disconnect the session between the AP and the STA is thus 1/8 per cycle in this case. If the number of authentication bits used is increased, the success rate for achieving DoS is decreased exponentially.

4. EXPERIMENTAL RESULTS

In this section, our experimental results are presented and discussed. First, our implementation scenarios, utilized tools and testing procedures are described. Next, experimental results are illustrated and the limitations in our experiments are explained.

4.1 Testing Environment

Four laptops were used to create our testing environment. Table 1 summarizes the hardware and software configuration for each laptop. All the wireless network card drivers and necessary software are open source. However, some relevant codes are modified to suit our experiments.

Table 1. Hardware/Software configuration in our experiments.

Function	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 2.8GHz	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 P4 1.73GHz	1.00GB	Netgear 802.11b MA401 (Chipset: Prism2)	Linux FC3 Kernel: 2.6.9-1.667	Host AP driver with 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

4.1.1 Tools and utilities

- **Host AP** is a driver for 802.11b cards with Intersil Prism2/2.5/3 chipset. Those wireless network cards 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.
- **Kismet** [8] is an 802.11 layer 2 wireless network detector, sniffer, and intrusion detection system. It was selected to gather 802.11 frames information in our experiments. Kismet works with wireless cards that support raw monitoring mode, and is able to sniff 802.11{a, b, g} traffic. It identifies networks by passively collecting packets, detecting both standard named and hidden networks; and inferring that no beacon is present via data traffic [7].
- **Ethereal** [20] is a GUI network protocol analyzer. It is utilized to display and analyze the packets captured by Kismet. In addition to analyzing the captured data, the "IO Graphs" tool is utilized to construct the graphs of the FTP sessions.
- **void11** [9] is a free implementation of some basic 802.11b attacks, and also includes some original features. This tool can be used to generate flooding of deauthentication notification, authentication request, and association requests. Some APs will deny any service after the request flooding is launched. Some APs or 802.11 cards become non-responsive for a period after void11 launched DoS flooding attacks [7]. Gvoid11 is the new graphical user interface of void11.

Void11 version 0.2.0 was used in our experiments. 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.

4.1.2 Testing procedures

Transferring a standard test file, whose size is 21,872,640 bytes, using FTP from the STA to the Host AP, is unchanged in the three separate experiments conducted. In an 802.11b network, the transfer should take about 35 seconds to complete. After the file was allowed to transmit for 15 seconds, later referred to as the "waiting time", deauthen-

tication or disassociation flooding attacks are launched for about 10 seconds from the attacker node (void11). The 15 second "waiting time" plus the 10 second attack time is well within the 35 seconds file transfer window. The duration of the FTP sessions are measured to determine the effect of the attack. FTP sessions are monitored and recorded by the monitor node (Kismet).

For each experiment, the testing procedure was repeated for at least 10 times, and the duration was averaged. The testing procedure for each experiment was described below:

(a) Normal FTP sessions were used as our baseline model.
(b) Perfect defense mechanism was simulated by ignoring all the deauthentication and the disassociation frames received by the Host AP. This is to determine the pure overhead caused by the existence of the flooding attacks.
(c) The effectiveness of our random authentication mechanism was determined by plotting the average file transfer duration with respect to the number of random bits used. The number of random bits tested ranged from 0 to 9.

When we tried to insert random authentication bits to the unused bits of the (de)authentication and (dis)association frames, we encountered some problems. We were able to 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. However, we were only successful in Master mode (Host AP mode), but not in managed mode (STA mode).

Other drivers have been tried, for example, the Linux-wlan-ng driver, but failed also. In consequence, an alternative method must be devised to test our design. As mentioned in section 3.2, one possible way to disarm our proposed defense is to try all the possible combinations to guess the random authentication bits in "brute force". Suppose the attacker knows that $N = 3$. The attacker would then transmit attacking frames with the random bits ranging from 0 to 7. In this case, the probability that the attacker succeeds is 1/8.

To emulate this behavior in another way, the attacking node (void11) could send its deauthentication or disassociation frame with one of the unused bit (random authentication bit) set to 1 (Bit15 in our implementation) with the probability of 1/8 if the number of random authentication bits is 3. The receiver (Host AP) was programmed to accept the deauth/disassoc request only when Bit15 = 1, and to ignore the frame when Bit15 = 0. This alternative strategy was used to test our random bit authentication mechanism.

4.2 Testing Results

Experiment 1: Bandwidth Consumptions of Normal FTP Sessions.

Normal FTP sessions between the STA and Host AP were recorded 10 times. The average file transfer duration was used as a baseline, and was found to be 35.5 seconds to transfer the standard test file. In Fig. 3, a typical behavior for Normal FTP session is depicted from the "IO Graph" of the ethereal program.

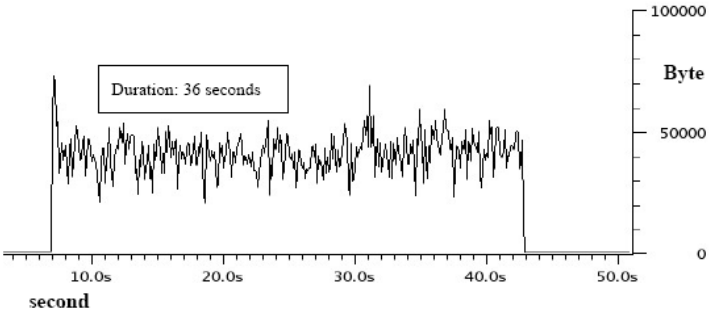

Fig. 3. Normal FTP session.

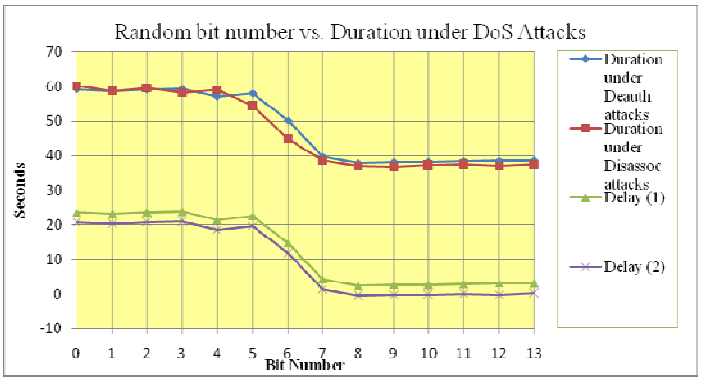

Fig. 4. Protection strength related to the number of random bits used.

Experiment 2: Bandwidth Consumptions of FTP Sessions under Attacks.

In this experiment, the extra bandwidth consumption due to deauthentication and/or disassociation flooding attacks was determined. File transfer duration for the standard test file was recorded and averaged during deauth/disassoc flooding attacks. All attacking frames were ignored by the Host AP to simulate the perfect protection. The duration, averaged over 10 trials, was found to be 38.4 seconds for deauthentication flooding attacks, and 38.1 seconds for the disassociation flooding attacks, respectively.

Experiment 3: Random Bit Authentication Defending Mechanism.

Fig. 4 showed that the more random authentication bits used, the less affected the communication is to the effect of deauthentication and disassociation flooding attacks. The figure also showed that $N = 5$ is not sufficient to defend DoS attack. The deauthentication flooding attacks still succeeded despite the existence of our security measure. To aid our experiments, void11 was modified to spoof STA's MAC address and send deauthentication messages to AP in a tight loop. We found that the attacker node could send about 80 forged deauthentication or disassociation frames per second. Therefore, the attacker node could make about 80/32 = 2.5 successful attempts on average every second.

After the number of bits were raised to 7, the average successful attempts on average per second became less than 1. No significant improvement was observed for even higher number of random bits as observed in Fig. 4.

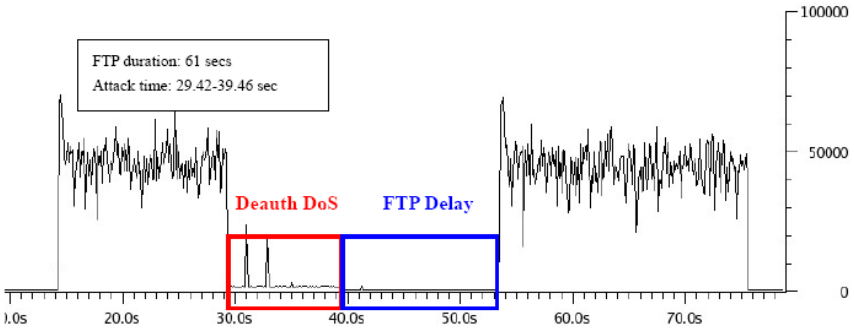

Fig. 5. Deauth. attack when no protection is applied ($N = 0$).

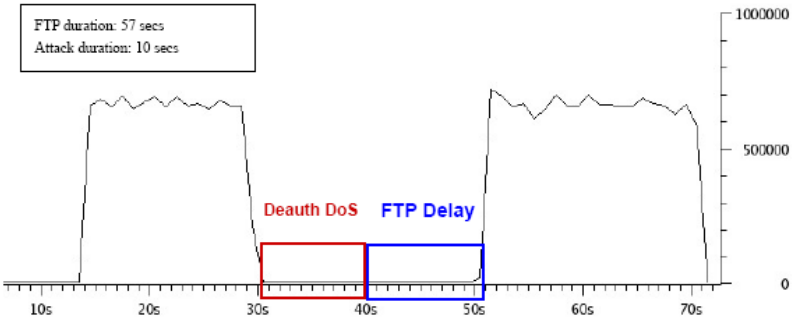

Fig. 6. FTP delay needed to recover from a deauth. attack.

Fig. 5 showed that, if not protected, the FTP session would be blocked when deauthentication started to flood. However, this figure also showed that the FTP session is not recovered immediately after the attack stopped, but instead delayed for an additional 13 seconds. We call this phenomenon “FTP delay”.

We speculate that the MAC layer connection may be recovered immediately, but the FTP session may not be right after the conclusion of an attack. The evidences below support that speculation.

In Fig. 6, TCP frames during FTP session were captured during the attack scenario. The figure exhibits the FTP delay phenomenon like that in Fig. 5. Next, a continuous stream of ICMP requests was issued before starting FTP session. Fig. 7 captures the result. From the graph, it can be observed that the echo response time are longer when the network loading is high during the file transfer. During the attack, the echo response message stopped since all communications were blocked by the deauthentication flooding. However, the echo response message resumed after the attack stopped and showed small time value during the “FTP delay” period.

In order to understand the performance of our method under diverse situations, the same experiments were performed using different file sizes and the delays relative to the normal FTP sessions were measured. The results obtained were shown in Table 2. For each combination of file size and number of random bits, the file transfer was initiated 10 times under 10 seconds of continuous deauthentication attacks. The transfer times without any incidents were also taken and averaged over 10 trials. Based on these

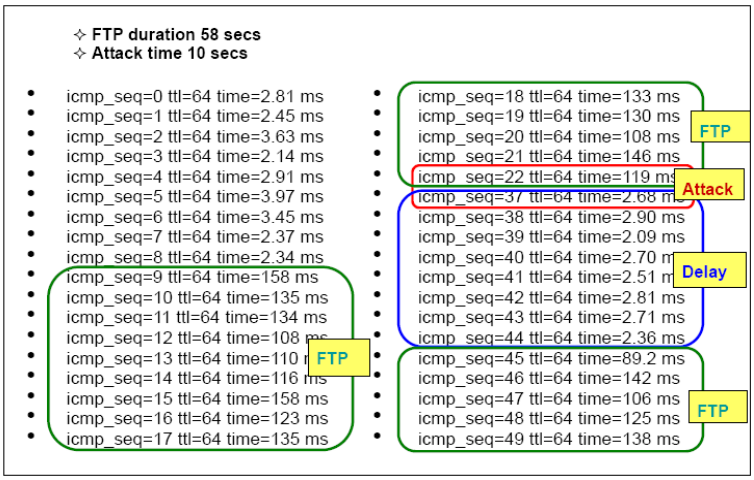

Fig. 7. Ping during a deauth. attacked FTP session.

Table 2. FTP transfer delay caused by deauthenticaiton attacks.

	Number of Random Bits													
Filesize	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1k	23.7	23.6	23.6	23.8	22.5	20.6	15.3	2.5	2.5	2.7	2.7	2.9	3.1	3.0
2k	23.6	23.4	23.8	23.5	22.7	22.5	14.5	2.7	2.6	2.8	2.9	3.0	2.9	3.1
4k	23.5	23.7	23.5	23.6	22.5	22.3	14.9	2.6	2.6	2.7	2.8	2.8	3.0	3.0
8k	23.6	23.6	23.5	23.7	22.4	20.5	14.7	2.8	2.7	2.7	3.0	3.1	2.9	2.9
64k	23.7	23.6	23.7	23.6	22.7	22.5	15.1	2.7	2.5	2.8	2.7	2.9	2.8	3.2
512k	23.6	23.5	23.6	23.7	22.1	22.0	14.6	2.5	2.7	2.6	2.9	2.8	2.9	2.8
1M	23.7	23.5	23.6	23.8	21.9	22.3	14.8	2.8	2.7	2.5	2.8	2.9	2.9	3.0
2M	23.7	23.7	23.5	23.6	22.5	22.5	15.2	3.1	2.8	2.8	2.7	2.9	3.1	3.0
4M	23.6	23.6	23.7	23.7	22.6	22.4	14.7	2.6	2.7	2.6	2.7	3.1	3.0	3.1
8M	23.5	23.7	23.5	23.8	22.5	22.3	14.6	3.9	2.6	2.7	2.9	2.8	2.9	3.0
16M	23.7	23.6	23.6	23.5	22.7	22.3	14.5	4.2	2.8	2.7	2.8	2.9	2.9	3.1
32M	23.7	23.7	23.6	23.8	22.3	22.2	14.6	2.5	2.6	2.6	3.0	2.9	3.1	2.9

experimental results, we found that the transfer delay does not depend on the size of the file being transferred but rather is solely dependent on the number of random bits used.

As shown in Table 2, we found that no matter what the file size is, the delay time is almost 23 seconds if the number of random bits used is less than or equal to 5. In order to find out the reason behind this observation, we also repeat the same experiments except to replace the wireless environment with cabled networks. The DoS attacks were simulated through the unplugging of network cables. The results obtained are consistent with the experiments under wireless environment. Therefore, we speculated the cause might lies within the characteristic of network protocols. We spent much time and effort to look for answers in the huge volumes of RFC documents. Finally, through RFC 2988, RFC 2581, and RFC 1122, we found that the delay is caused by the TCP retransmission.

In order to study the effectiveness of our proposed scheme under different protocols, file transfers using HTTP were also performed using different file sizes. However, similar results were also observed. Since TCP is the underlying transport layer protocol for HTTP, delay caused by TCP retransmission process is also observed in the HTTP experiments. We thus conclude that all TCP-based application protocols would exhibit similar delay pattern when tested. This conclusion is further verified when we used TFTP, an UDP-based protocol, for file transferring. The transmission was observed to resume immediately once the attack stopped.

5. CONCLUSION AND FUTURE WORK

We designed an efficient, but simple, mechanism to guard against the DoS attacks that exploits the weakness inherent in the deauthentication and disassociation procedures in an 802.11 network. Using the random bit authentication, our approach is shown through the experimental results to reduce the impact caused by deauthentication and disassociation flooding attacks.

Although we have only used the random bit authentication as a defense mechanism for the two flooding attacks, our design can also be adopted to defend similar DoS attacks. For example, EAPOL-Failure and EAPOL-Logoff messages are sent over the wireless network unencrypted. They also lack key-based authentication mechanism to protect these frames. Hence, our random bit authentication scheme can also be used to prevent forged EAPOL-Failure and EAPOL-Logoff messages from seeping into 802.11 networks unnoticed.

Since wireless devices, such as access points and wireless NICs, have limited resources, we designed the random bit authentication mechanism to embed its authentication information into the inactive bits of the control field in the MAC header. The defensive power is derived from the unpredictability of a random bitstream, and not from complicated encryption algorithm. Hence, the consumption of the computation and bandwidth resources is lightweight for our approach. Because of this, an added advantage is that our design can be adapted to protect the system from other kinds of resource-draining DoS attacks.

In this paper, we only performed the experiments in an 802.11b based network environment. Although the newer 802.11i amendment exhibit similar vulnerabilities, we did not pursue any further on the effectiveness of our design in the updated version. In the future, we plan to test our design under an 802.11i based network, and also try to demonstrate the applicability of our design for other attacks that exploit management frame authentication vulnerability. One possible direction is to tackle the problem of spoofed EAPOL-Failure and EAPOL-Logoff messages.

REFERENCES

1. Y. S. Lee, H. T. Chien, and W. N. Tsai, "Using random-bit authentication to defend 802.11 networks," in *Proceedings of International Computer Symposium*, 2006, pp. 777-782.

2. 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.
3. 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.
4. D. Chen, J. Deng, and P. K. Varshney, "Protecting wireless networks against a denial of service attack based on virtual jamming," in *Proceedings of the 9th ACM Annual International Conference on Mobile Computing and Networking*, 2003.
5. J. R. Walker, "Unsafe at any key size: An analysis of the WEP encapsulation," Technical Report 03628E, IEEE 802.11 Committee, 2000, http://grouper.ieee.org/groups/802/11/Documents/DocumentHolder/0-362.zip.
6. A. Stubblefield, J. Ioannidis, and A. Rubin, "Using the Fluhrer, Mantin, and Shamir attack to break WEP," in *Proceedings of Network and Distributed Systems Security Symposium*, 2002, pp. 17-22.
7. C. He and J. C. Mitchell, "Security analysis and improvements for IEEE 802.11i," in *Proceedings of the 12th Annual Network and Distributed System Security Symposium*, 2005, http://www.isoc.org/isoc/conferences/ndss/05/proceedings/papers/NDSS05-1107.pdf.
8. M. Kershaw, Kismet 2005-08-R1, 2005, http://www.kismetwireless.net.
9. R. Floeter, Wireless LAN Security Framework: Void11, 2002, http://www.wlsec.net/void11/.
10. J. Malinen, Host AP driver for Intersil prism2/2.5/3, hostapd, and WPA supplicant, 2005, http://hostap.epitest.fi/.
11. S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," in *Proceedings of the 8th Annual Workshop on Selected Areas in Cryptography*, 2001, http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf.
12. H. Johnson, A. Nilsson, J. Fu, S. F. Wu, A. Chen, and H. Huang, "SOLA: A one-bit identity authentication protocol for access control in IEEE 802.11," in *Proceedings of IEEE Global Telecommunications Conference*, Vol. 21, 2002, pp. 777-781.
13. H. Wang, A. Velayutham, and Y. Guan, "A lightweight authentication protocol for access control in IEEE 802.11," in *Proceedings of IEEE Global Telecommunications Conference*, 2003, pp. 1384-1388.
14. J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions," in *Proceedings of the 12th USENIX Security Symposium*, 2003, pp. 15-28.
15. F. Ferreri, M. Bernaschi, and L. Valcamonici, "Access points vulnerabilities to DoS attacks in 802.11 networks," *IEEE Wireless Communications and Networking Conference*, Vol. 1, 2004, pp. 634-638.
16. C. He and J. C. Mitchell, "Analysis of the 802.11i 4-way handshake," in *Proceedings of the ACM Workshop on Wireless Security*, 2004, pp. 43-50.
17. B. Aboba, "Issues in pre-standard IEEE 802.11i implementations," http://www.drizzle.com/~aboba/IEEE/prestand.html.
18. AirSnort, http://airsnort.shmoo.com/.

19. A. T. Rager, "WEPCrack," http://wepcrack.sourceforge.net/.
20. Ethereal, http://www.ethereal.com/.
21. K. Ren, H. Lee, K. Han, J. Park, and K. Kim, "An enhanced lightweight authentication protocol for access control in wireless LANs," in *Proceedings of the 12th IEEE International Conference on Networks*, Vol. 2, 2004, pp. 444-450.
22. P. Q. Ding, J. N. Holliday, and A. Celik, "Improving the security of wireless LANs by managing 802.1X disassociation," in *Proceedings of IEEE Consumer Communications and Networking Conference*, 2004, pp. 53-58.
23. H. Boland and H. Mousavi, "Security issues of the IEEE 802.11b wireless LAN," *Canadian Conference on Electrical and Computer Engineering*, Vol. 1, 2004 pp. 333-336.
24. J. Wright, "Detecting wireless LAN MAC address spoofing," SecurityFocus Bugtraq: Whitepaper, 2003, http://forskningsnett.uninett.no/wlan/download/wlan-mac-spoof.pdf.
25. C. Liu, "802.11 disassociation denial of service attacks," http://www.mnlab.cs.depaul.edu/seminar/spr2005/WiFiDoS.pdf.
26. J. 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) and Physical Layer (PHY) specifications: Protected Management Frames," *IEEE 802.11w Approved Letter*, 2005, http://standards.ieee.org/board/nes/projects/802-11w.pdf.
27. J. Walker, "Status of project IEEE 802.11 task group w: Protected management frames," IEEE P802.11 – Task Group w – Meeting Update, http://grouper.ieee.org/groups/802/11/Reports/tgw_update.htm.
28. IEEE 802.11, http://en.wikipedia.org/wiki/802.11.
29. IEEE P802.11 – Task Group n – Meeting Update, Status of Project IEEE 802.11n: Standard for Enhancements for Higher Throughput, http://grouper.ieee.org/groups/802/11/Reports/tgn_update.htm.
30. CERT Coordination Center, "Denial of service attacks," http://www.cert.org/tech_tips/denial_of_service.html.
31. Denial-of-service attack, http://en.wikipedia.org/wiki/Denial-of-service_attack.
32. N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: The insecurity of 802.11," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, 2001, pp. 180-189.
33. B. Aslam, M. H. Islam, and S. A. Khan, "802.11 disassociation DoS attack and its solutions: A survey," in *Proceedings of the 1st Mobile Computing and Wireless Communication International Conference*, 2006, pp. 221-226.

Ying-Sung Lee (李英宗) received his M.S. degree in Computer Science and Information Engineering from National Chiao Tung University, Hsinchu, Taiwan in 2006. He received his B.S. degree in Mathematics from Fu Jen Catholic University, Hsinchuang, Taiwan, in 2004. He is currently pursuing his Ph.D degree in Computer Science in National Chiao Tung University. His research interests include computer network security in general, and service availability in particular.

Hsien-Te Chien (簡先得) received his M.S. degree in Computer Science and Information Engineering from National Chiao Tung University, Hsinchu, Taiwan in 2006. He is currently the head of the dean's office at the Kuo-Kuang Elementary School in Taichung, Taiwan.

Wen-Nung Tsai (蔡文能) has been studying the Ph.D. program in Electrical Engineering and Computer Science in Northwestern University, U.S.A. in 1990s. He received his M.S. degree in Computer Science and Information Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1979. He is now an Associate Professor of Computer Science Department, National Chiao Tung University, Hsinchu, Taiwan, since 1996. His research interests include computer networks, network security, operating system design and mobile computing.