# Using Random Bit Authentication to Defend IEEE 802.11 DoS Attacks

Ying-Sung Lee, Hsien-Te Chien, Wen-Nung Tsai
*Department of Computer Science and Information Engineering,*
*National Chiao-Tung University*
*{leeyt, htchien, tsaiwn}@csie.nctu.edu.tw*

## ABSTRACT

*The security of IEEE 802.11 networks has always been a heavily discussed issue. WEP, the security mechanism adopted in 802.11, was proved to be vulnerable and easily attacked. The enhanced version of 802.11*[1], *802.11i*[2], *concentrates on maintaining integrity and confidentiality when transmitting 802.11 frames. Either version did not consider the network availability aspect properly. Because management frames are not protected by any key-based authentication, Denial of Service (DoS) attacks is a threat to an {802.11, 802.11i} network.*

*In this paper, we designed a random bit authentication mechanism to protect 802.11 networks from DoS attacks. Random bits are placed into unused fields of the management frames. AP and STA can then authenticate each other according to these authentication bits. The effectiveness of our mechanism in defending DoS attacks is demonstrated using our experimental results.*

## 1: INTRODUCTION

In a WLAN environment, security designers basically focus on three security services: confidentiality, integrity and availability. Some researchers found that 802.11i only concentrates on confidentiality and integraty, and overlooks availability issue. 802.11i appears susceptible to DoS attacks even after RSNA is implemented [4].

Launching DoS attacks within an {802.11, 802.11i} networks is easy. An adversary with moderate skills can use free attacking tools to successfully mount a DoS attack on his laptop equipped with a cheap 802.11 wireless card. Although we may not stop DoS attacks completely, we could impose a relatively higher cost on the attackers to deter them from disrupting network services.

Certain vulnerabilities are inherent in the protocol design with deauthentication and disassociation among the examples. 802.11 transmitting sessions are established by exchanging management frames. However, these frames are not protected by any key-based authentication, and are transmitted unecrypted. Therefore, an adversary can fake a deauthentication and/or disassociation request and successfully deny legitimate users from their rightful service.

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 association flooding attacks. Next, our experimental results are summarized in section 4. Finally, we conclude our results and provide some directions for future works in section 5.

## 2: RELATED WORKS

WLANs, due to their inherent openness, are quite vulnerable to DoS attacks. Many researchers have discovered numerous attack methods and we summarize some of their findings and proposed defense in this section.

### 2.1: DOS ATTACKS AGAINST 802.11 NETWORKS

Many vulnerabilities of an 802.11 network are derived from the lack of protections on its management and control frames. For instance, some adversaries could exploit CSMA/CA mechanisms for launching DoS attacks.

Amongst the exploitations, the most efficient DoS attack is the flooding of deauthentication or disassociation frames. On the Internet exist many tools that allow anyone to execute such attacks.

**2.1.1: Deauthentication and disassociation flooding attacks under 802.11 networks.** When an 802.11 STA wishes to connect to the network, it must first authenticate and then associate itself to a selected AP. To terminate the connection, the STA or the AP can send out a deauthentication request to the other node. Unfortunately, the deauthentication process is not protected by any cryptographic means. Consequently, an attacker can pretend to be the AP or the STA and disconnect an active connection by spoofing the deauthentication message. Similarly, diassociation messages can also be exploited this way. Figure 1 illustrates the scenario of the deauthentication and disassociation attacks [10].

When a deauthentication notification arrives, the receiver will immediately leave the authenticated state until authentication is reestablished. If the attacker saturates a network with deauthentication frames, no communication can take place.until the attack stops.

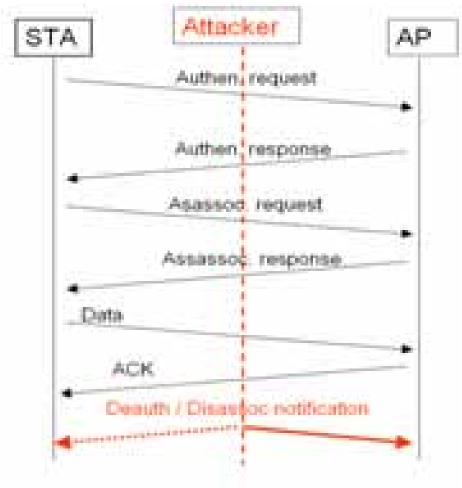


**Figure 1 Deauth and Disassoc attacks**

J. Bellardo and S. Savage [10] designed one defensive mechanism by the method of delaying the responses of deauthentication or disassociation requests (e.g., queue such requests for 5-10 seconds before responding). Receivers now gain the opportunity to observe subsequent packets from their communication partners. Since a legitimate node would never generate more data packets once it wants to disauthenticate or disassociate, a data packet after the fact indicates a bogus request and that request should be discarded.

However, there are some drawbacks in their approach. Their defensive measure not only delays the handoff process but also creates new vulnerabilities when mobile clients roam between access points.

**2.1.2: Traffic Jamming DoS attack.** The traffic jamming DoS attack refers to the attack type that tries to exhaust resources of network devices (e.g. AP) to such a point that the devices will be unable to provide network services for legal network nodes.

Ferreri, F. et al. [11] developed a simple attacking tool, named wfit (wireless frame injection tool), based on the Radiate library which is built on top of an old version of the HostAP driver. They demonstrated their attacking tool by using it to launch probe, authentication and association flooding attacks.

Although they had also attempted to design a detection and defense mechanism in a Linux HostAP, they could not mitigate probe flooding attacks at the software driver level. No better solutions have been suggested in their paper.

Other attacks such as virtual carrier sense DoS attack [3], power saving mode attack and many others also fall into the category of traffic jamming attacks.

## 2.2: DOS ATTACKS AGAINST 802.11I NETWORKS

IEEE institute asserts that 802.11i supports the strongest data confidentiality and authentication protocols; however, the seemingly lack of emphasis on the availability issue has resulted in 802.11i being vulnerable to DoS attacks. Changhua He et al have claimed in [4] that the threat of DoS is even more severe in 802.11i than in 802.11.

Since the DoS problem is not solved in the 802.11i ammendment, the DoS attacks as mentioned in section 2.1 are also effective under an 802.11i deployment.

**Deauthentication and disassociation attacks against 802.11i network**

Ping Ding, JoAnne Holliday and Aslihan Celik [15] devised an application called Central Manager (CM) to defend 802.11i network with 802.1x implemented against DoS attacks.

Their system works as follows. After an AP receives a disassociation frame from a STA, the AP will forward the frame to the CM. The CM then calls that STA to confirm its intention to be disassociated. The AP will proceed to disassociate the requesting STA only if the CM receives a confirmation. Otherwise, the disassociation request is ignored.

**EAPOL-Failure and EAPOL-Logoff message attacks**

Other than the disassociation attack, several DoS attacks exploit the unprotected EAP messages in 802.1x authentication. An adversary can forge the EAPOL-Failure message and the EAPOL-Logoff message to disconnect the supplicant [11][12][16].

Ping Ding, et al. [11] also suggested using their Central Manager to defend against EAPOL-Failure and EAPOL-Logoff message DoS attacks.

## 2.3: ONE-BIT LIGHTWEIGHT AUTHENTICATION

Statistical One-bit Lightweight Authentication (SOLA) protocol [8] is a new identity authentication protocol proposed to detect unauthorized access in 802.11 networks. The main idea is for STAs and the AP to agree on a secret key so they can both generate identical random authentication stream, and then the communicating party will add one bit from this stream 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 packet origins for the purpose of access control. The authors claim that SOLA protocol is well suited in a resource-constrained wireless 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.

H. Wang, et al. [9] followed up on the lightweight authentication ideas, but criticized the synchronization algorithm of [8]. They also developed a workable synchronization algorithm.

Both of the researchers focused on syncronization algorithms and the statistical analysis. Implementation issues are not sufficiently considered.

**Enhanced lightweight authentication**

Kui Ren, et al. [14], found that some servere synchronization problems exist in Johnson's work on the lightweight authentication due to the frame loss in the wireless networks. They also criticized the inefficiencies of the protocol designed by Wang et al. [9].

In [14], they also proposed an enhanced lightweight authentication protocol for access control at the MAC layer in wireless LANs. They examined the redundancy existed in the MAC header, and adopted 3-bit authentication mechanism. To begin communicating, the sender and the receiver establish a common random bit stream generator by sharing a seed value. The random bit stream generator is used to output 3-bit unit at a time to be inserted into outgoing data frame's control field. Upon receiving a frame, the legitimate receiver will then be able to compute a 3-bit random value that is identical to the 3-bit authentication value stored in the incoming data frame.

The authors of [14] also offered a statistical way to identify the data origin for the purpose of detecting an attack. They asserted that the protocol is fully compatible with current IEEE 802.11 frame structure and the protocol provides a highly efficient identity authentication scheme.

## 3: PROPOSED PROTOCOL TO COUNTER DOS ATTACKS IN 802.11 NETWORKS

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 we explain our design in section 3.2, we first analyze 802.11 management frames in section 3.1 to determine the header fields that are available to be used for our random authentication bits.

### 3.1: UNUSED BITS OF 802.11 MANAGEMENT FRAME

The general management frame format can be found in [1]. In the following sections, we performed some analyses on the frame control and frame body field to find out the unused bits.

**3.1.1: Management frame control field analysis.** Figure 2 illustrates the frame control field of the MAC header in the management frame. The frame type can be determined with the type field and the subtype field.

In the management frame, "To DS", "From DS" and/or "More Fragments" bits are set accordingly. A management frame is never fragmented. The "Power Management" and the "More Data" bit are used only in the control frame to indicate the power management mode of a STA. The "Order bit" is used in data frames being transferred with Strictly Ordered service class [1].

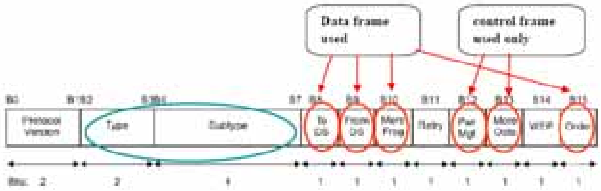


**Figure 2 Unused control fields in a management frame [1]**

**3.1.2: Management frame body analysis.** Table 1 displays the structure of the authentication frame body. The authentication algorithm number is either 1 or 0, and requires only a single bit to represent yet 16 bits are reserved. That gives us 15 unused bits. Similarly, the authentication transaction sequence number only requires 3 bits to represent, and leaves us 13 useable bits.

| 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 1 Athentication frame body [1]**

The deauthentication and disassociation frame body only contain a reason code. In the 802.11 standard [1], nine reason codes have been defined, and require only 4 bits to represent. Twelve unused bits can thus be used for random authentication bits. For 802.11i [2], a 5-bit reason code is adopted, and we only have 11 unused bits to work with.

The association request and response frame bodies are as structured in Table 2. Their capability information field has its 11 (B5 to B15) bits reserved as described in the 802.11 standard [1]. The reassociation request and response frame also include similar informations in their frame bodies. Their capability information fields also contribute 11 bits that can be used for random authentication bits. However, in 802.11b, only 8 (B8-B15) bits are reserved; others are defined.

| 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 2 Association req./resp. frame bodies [1]**

The analysis in this section and the previous has determined the unused bits in the header and body of an 802.11 management frame. The number of unused bits is enough to implement our DoS defense scheme.

### 3.2: RANDOM BIT AUTHENTICATION FOR MANAGEMENT FRAME

We assumed that all the communicating nodes had shared the same key, and one session key will be generated for each communicating group based on the shared key. We will not discuss the key generation and exchange issues in this paper. Furthermore, we assumed that the communicating nodes had agreed on a common algorithm for generating identical random bitstreams.

After the nodes in the same basic service set generated identical random bitstreams independently, each node divides the bitstream into equal length units; each unit consists of "N" authentication bits. Figure 3 shows an example of using 3 bits as the unit of division. We number each unit sequentially from left to right, and use 8 units in our design under 802.11 {b, g}.

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

**Figure 3 Example of the bitstream generated and divided when N = 3**

Figure 4 illustrates the random bit authentication protocol under deauthentication or disassociation flooding attacks. Since the attacker does not know the values of the 5th and 8th units, he/she must keep guessing the random authentication bits until he/she successfully guessed the bits on his targeted victim. Alternatively, the attacker can use "brute force" and try out all the possible combinations. As illustrated in Figure 4, the attacker can cycle through the value of 0 to 7 as the guessed authentication bits. The success rate of an attacker to disconnect is thus 1/8 per cycle. If we use longer authentication bits, the success rate of such attack will decrease exponentially.

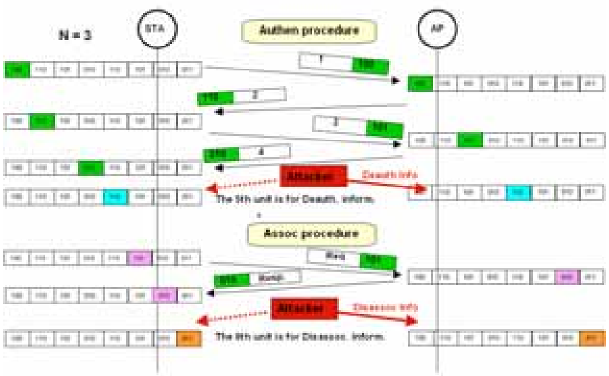


**Figure 4 Scenario of attacks under random bit authentication for authen. & assoc. procedures**

We encountered some problems while trying to change the system to insert random authentication bits to the unused bits of the (de)authentication and (dis)association frames. On our Linux based platform decribed in the next section, we could modify the Host AP driver to insert a maximum of 4 random authentication bits into the frame control field (at B9, B12, B13 and B15) in the MAC header of the authentication and association frames. 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), and not in Managed mode (STA mode). We asked Jouni Malinen [7], the auther of the Host AP driver, for help, and he told us that we failed because in Managed mode, the frame control field is processed in the firmware of the prism2/2.5/3 based 802.11b card instead of through the Host AP driver.

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

To attack our design, it is conceivable that the attacker will use "brute force" to guess the random bits. For example, the attacker could cyclically place the values from 0 to 7 at the random bit positions of the forged frames when we use 3-bit random bits unit. In this case, the success probability of the attacker is 1/8. In general, the success rate of such attack is $1/2^N$ when the number of random authentication bits used is *N*.

Taking advantage of the observation in the previous paragraph, we configured the attacking node (using void11[6]) to send attacking frames with Bit15 set to 1 once every 8 frames if the number of random authentication bits used is 3. The receiving Host AP will pass frames with Bit15 set to 1, and drops frames with Bit15 set to 0. Therefore, the success rate of the attack is $1/8$. We exploited this alternative design to test our random bit authentication mechanism.

## 4: EXPERIMENTAL RESULTS

In this section, we describe our implementation setup, utilized tools and procedures. Then, we present our experimental results and discuss the implications and and the limitations in our experiments.

### 4.1: TESTING ENVIRONMENT

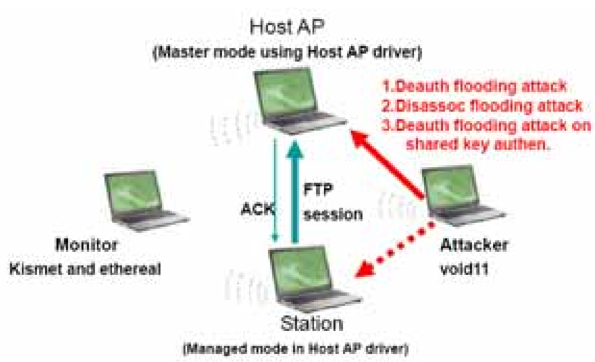


**Figure 5 Experimental Setup**

**Figure 5** illustrates our experimental setup. We utilized 4 Linux based laptops PCs equipped with Intersil's

Prism2/2.5/3 802.11b PCMCIA cards. All the hardware devices used are bought from electronic stores, and softwares are open source, as listed in Table 3. We describe the tools and utilities to conduct our experiments in the next section.

| 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 3 Hardware and Software Configuration in Our Experiments**

**4.1.1: Tools and utilities.**

**a. Host AP**

Host AP is a driver for the 802.11b cards with Intersil Prism2/2.5/3 chip. The firmware of such cards takes care of time-critical tasks like beacon sending and frame acknowledging, but leaves other management tasks to the device driver.

**b. Kismet**

Kismet [5] is an 802.11 layer 2 wireless network detector, sniffer, and intrusion detection system. It works with wireless cards supporting raw monitoring mode, and can sniff 802.11{a,bg} traffic. Kismet identifies networks by passively collecting packets, detecting standard named and hiden networks, and inferring the absense of beaconing via data traffic [4].

**c. Ethereal**

Ethereal [13] is a GUI network protocol analyzer, and can be used to display and analyze the packets captured by Kismet. Its "IO Graphs" tool is utilized to reconstruct the FTP session activities.

**d. void11**

Void11 [6] is a free implementation of some basic 802.11b attacks, and other original features. It implements the deauthentication DoS attack by flooding wireless networks with deauthentication packets and spoofed BSSID. The authenticated stations will then drop their network connections. Void11 can also be configured to flood APs with authentication or association request packets with spoofed random station MAC addresses. Some APs will deny any service while others stop responding for a period after void11 launched DoS flooding attacks.

We used void11 version 0.2.0 in our experiments, and issued "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.** Our experiment is conducted in three phases. They are as follows. The first phase tests normal FTP sessions to establish our baselines. The second phase ignores all deauthentication and disassociation attacking frames to simulate the perfect defense. The third phase enables the random bit authentication mechanism to study its effectiveness in blocking deauthentication and the disassociation flooding attacks.

## 4.2: TESTING RESULTS

**Experiment 1: Normal FTP Sessions**

We uploaded a file of 21,872,640 bytes using FTP protocol from a STA to the Host AP, and recorded the durations of 10 normal FTP sessions. The average upload time is 35.5 seconds as shown in Table 4.

| 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 4 Durations of Normal FTP sessions**

**Experiment 2: FTP Sessions under Attacks**

We must consider the bandwidth consumption during deauthentication and disassociation flooding attacks since attacking frames consume wireless bandwidth regardless of their success.

In this experiment, we launched deauthentication and disassociation flooding attacks, and instructed the Host AP to drop all attacking frames. Table 5 lists the durations of FTP sessions under this setting.

| 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 5 Durations of FTP sessions under deauth/diassoc flooding attacks with all attacking frames dropped**

**Experiment 3: FTP Sessions with Random-bit Authentication Enabled**

In this experiment, we study the variations of the number of authentication bits on the file upload time during deauthentication and disassociation attacks. As shown in Figure 6, the more random authentication bits used, the more effective the system in defending such attacks. This figure shows that using 5 random bits is inadequate in defending DoS attacks. Only after the number of bits is increased beyond 7 would the flooding attacks to almost have no effect on the file upload delay.

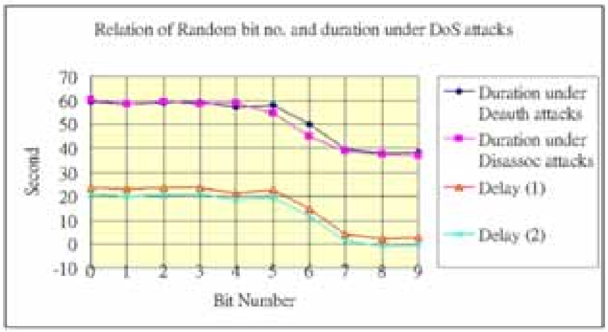


**Figure 6 Number of Random bits used vs FTP durations and delays during deauth/disassoc flooding attacks**

## 5: CONCLUSION AND FUTURE WORK

We designed an efficient, but simple, mechanism to defend against the DoS attacks on 802.11 networks. Using the random bit authentication, our approach can defend against deauthentication and disassociation flooding attacks as shown by our experimental results.

Our design can also be adopted to defend similar DoS attacks. For example, EAPOL-Failure and EAPOL-Logoff messages also gives rise to DoS attacks due to their lack of authentication protocol.

Our random bit authentication mechanism recycles the unused bits of the control field in the MAC header to embed authentication information. Also, no complicated encryption and decryption algorithm are used. Hence, our approach does not consume computation and bandwidth resources of the network. Another notable advantage of our approach is its ability to adapt to protect the system from other kinds of resource-hogging DoS attacks.

In this paper, we only implemented our experiments in 802.11b based nework. In the future, we plan to test our design under an 802.11i based network, and also try to demonstrate our claim that our design can indeed protect network users from EAPOL-Failure and EAPOL-Logoff DoS attacks.

## REFERENCE


[1] 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.

[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] 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] 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

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

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

[7] Jouni Malinen. Host AP driver for Intersil Prism 2/2.5/3, hostapd, and WPA Supplicant, 2005, http://hostap.epitest.fi/.

[8] 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.

[9] 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.

[10] J. Bellardo, and S. Savage. "802.11 Denial-of-Service attacks: real vulnerabilities and practical solutions". In Proceedings of the 12th USENIX Security Symposium, Washington, D.C., August 4-8, 2003.

[11] 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.

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

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

[14] 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 IEEE International Conference on Volume 2, Nov 16-19. 2004, Pages: 444–450.

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

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