A lightweight solution for defending against deauthentication/disassociation attacks on 802.11 networks

Thuc D. Nguyen† Duc H. M. Nguyen† Bao N. Tran† Hai Vu⋆ Neeraj Mittal⋆
† Vietnam National University, Hochiminh City, Vietnam
⋆ University of Texas at Dallas, Richardson, TX, USA

Abstract

In this paper we investigate a special type of denial of service (DoS) attack on 802.11-based networks, namely deauthentication/disassociation attack. In the current IEEE 802.11 standards, whenever a wireless station wants to leave the network, it sends a deauthentication or disassociation frame to the access point. These two frames, however, are sent unencrypted and are not authenticated by the access point. Therefore, an attacker can launch a DoS attack by spoofing these messages and thus disabling the communication between these wireless devices and their access point. We propose an efficient solution based on a one way hard function to verify that a deauthentication frame is from a legitimate station. We implement our solution on some 802.11 devices and the experimental results show that our protocol is highly effective against this DoS attack.

Keywords: 802.11 networks, Deauthentication, Disassociation, DoS attacks, Wireless security

1. Introduction

IEEE 802.11-based networks have been very successful because they only require inexpensive hardware devices operating on free spectrum with low cost deployment. Due to their popularity, 802.11 networks have been the target for a large number of attacks. Researchers and industrial companies have been trying to fix the vulnerabilities in 802.11 networks by proposing a number of protocols and standards (such as WEP, WPA, EAP, 802.11i, 802.1x). However, some flaws are still not addressed by any of these protocols, one of which is the deauthentication/disassociation attack described as follows.

802.11 networks can operate in ad-hoc mode or infrastructure mode. In this paper we are only concerned with 802.11 networks operating in infrastructure mode, in which a wireless client (in this paper we use the term "client" and "station" interchangeably) needs to associate with an access point (AP), before data messages can be further exchanged. Before associating with the AP, the client needs to authenticate itself to the AP. If a station (STA) wants to disassociate with an AP, it sends a disassociation frame to that AP. In case the station wants to gracefully leave the network, it sends a deauthentication frame to the AP. Similarly, when the AP wants to disconnect a client, it sends a disassociation frame to that client. In case the AP wants to disassociate with all the STAs (for instance, it reboots before upgrading the firmware), it broadcasts the disassociation frame to all clients. However, the current description of 802.11 standards specifies that the deauthentication frame and the disassociation frame not be authenticated. Because the deauthentication and disassociation frames are unencrypted and unauthenticated, an attacker can easily spoof these frames (by spoofing the MAC address of the client or the AP) thereby disconnecting the client from the AP, effectively launching a DoS attack. Even though the deauthentication frame and the disassociation frame are similar, spoofing the deauthentication frame is more effective since it requires the STAs and the AP to perform the authentication again in order to resume the connectivity.

One trivial solution to this vulnerability is to modify the authentication framework such that the AP and STAs could authenticate all the management messages in 802.11 networks, including the deauthentication/disassociation frames. However, this solution has two problems. First, millions of legacy devices that have already been deployed may not be able to support the required cryptographic primitives to mutually authenticate the management frames [15]. Second, authenticating all management frames may lead to a new DoS attack in which the attacker floods the AP with a large number of spoofed management frames, depleting the computation resources of the AP. Thus, a lightweight and efficient solution for defending against deauthentication/disassociation attacks is desirable.

In this paper we present a new protocol based on a one way hard function to defend against the deauthentication/disassociation attacks, which we now refer to as "Farewell attack". Our solution does not require legacy de-

vices to support new cryptographic primitives, thus it can be widely deployed as an extension to the current 802.11 standards.

The rest of the paper is organized as follows. In Section 2 we discuss in detail how Farewell attack is launched and existing solutions to defend against it. We present our solution for defending against a Farewell attack in Section 3. In Section 4, we show that our solution can effectively defend against the Farewell attacks. Finally, we conclude the paper in Section 5.

2. Farewell attacks and related work

2.1. Farewell attacks

In [14], Aslam et al. describe an association process as a three steps process with four states:

1. Unauthenticated and unassociated
2. Authenticated and unassociated
3. Authenticated and associated
4. Authenticated, associated and 802.1x authenticated

Initially both client and AP are in state 1. In order to join a network, a client scans all the channels to find an AP. After identifying the preferred AP, the client and the AP perform mutual authentication by exchanging several messages. They can either use *Open Authentication* or *Shared Key Authentication* [13]. In Open Authentication, the system authenticates anyone who requests to join the network. In Shared Key Authentication, a secret key is shared between the client and the AP. The client and AP go through a sequence of challenge-response in which the message may be encrypted with "WEP" using the shared key. Upon completion of the authentication, both client and AP move to state 2. In state 2, the client associate to the AP and both of them transit to state 3. In state 3, the client can now send data packets to the AP. However, if the 802.1x protocol is supported, then the 802.1x authentication messages will be exchanged between the client and the AP. On successfully finishing 802.1x authentication, both the client and AP move to state 4.

Note that, on receiving a disassociation message the state machine of the client and AP move back to state 2 no matter where they were in state 4 or state 3. Similarly, on receiving a deauthentication message, the state machines transit to state 1. On receiving the disassociation and deauthentication frame, the client and AP clears the relevant states and keys in the memory. The deauthentication and disassociation frames are unauthenticated and unencrypted, thus are sources of persistent flaws in 802.11 networks.

As explained above, to break the communication between the clients and their AP, an attacker can simply send out a spoofed deauthentication or disassociation frame. There are a number of tools that enable an attacker to spoof the source MAC address of any device, such as: SpoofMAC [9], Airsnarf [2], MAC Changer [7]. Note that if the attacker spoofs a deauthentication or a disassociation frame of the AP with a broadcast destination MAC address, then effectively all clients associated to the AP will be disconnected.

The Farewell attack is simple but can cause serious damage, because the attacker can stop the communication using only limited resources without requiring any special technical skill. The attacker even does not need to break the authentication protocol or to obtain shared secret keys between the STAs and the AP. If the attacker sends a disassociation frame, the victim clients must set up a new association session with the AP. If the attacker sends a deauthentication frame, the victim STAs must perform a new authentication session with the AP in order to resume connectivity. In [15], Bellardo et al. implement the attacks and show that this attack is simple and effective. At the moment, a number of tools such as Airjack [1], KisMAC [5], Void11 [10], WLAN-jack [11], FATA-jack [4], CommView [3] can be used to launch Farewell attack.

2.2. Related work

There are a number of solutions that have been proposed to defend against Farewell attack, as summarized in [14, 17]. Some of the important solutions are discussed below:

- *Approach*: eliminating the deauthentication and disassociation frames, or enqueueing them for a fix interval of time (for instance, 10 seconds) [15].
  Issues: there may be a period of time where a STA associates with multiple APs concurrently, which may cause routing/handoff problems [14, 17].
- *Approach*: using Reverse Address Resolution Protocol (RARP) to detect spoofed frames [16].
  Issues: the attackers may spoof the IP address of the client to break the RARP. Moreover, the solution does not work if multiple IP addresses are assigned to the same network card [14, 17].
- *Approach*: detecting spoofed frames based on frame sequence number [22, 12, 18, 23].
  Issues: if the sequence number are assigned deterministically, the attacker may sniff the frames sent by the client to predict the sequence number of the next frame [14, 17].
- *Approach*: developing a lightweight authentication protocol for management frames, such as using 1 bit for authentication [19, 21].
  Issues: errors in wireless medium may break the authentication, and the probability of an attacker to guess the authentication bit correctly is high (50%) [14, 17].

- *Approach*: modifying the current authentication framework to authenticate deauthentication and disassociation frames.
  Issues: this requires the clients to be able to support the modified authentication framework. This is not possible for millions of legacy devices that cannot support cryptographic primitives required by the authentication framework [15]. Moreover, if the framework includes a centralized authentication server like in 802.1x, then this solution suffers from the single point of failure problem and DoS attacks on the server, which has to process a large number of deauthentication frames flooded by the attacker [14, 17].

We develop a lightweight scheme for authenticating the management frames. However, instead of using sequence number, we use a one way function, thus our scheme is computationally infeasible to break. That means only the management frames send by legitimate STAs and APs are accepted. Our scheme does not depend on advance cryptographic primitives, thus all 802.11 devices can implement our solution via firmware upgrade.

3. Solution to deauthentication/disassociation attacks

3.1. Letter-envelop protocol

In this paper we propose a lightweight authentication protocol, which we call "Letter-envelop" protocol, that can defend against the Farewell attack. The protocol works based on the "factorization problem", which is known to be one way hard: given a large number $N = p * q$ (where p and q are two large prime numbers), it is computationally infeasible to compute p and q. However, given p and q, it is easy to compute N. The "Letter-envelop" protocol is as follows:

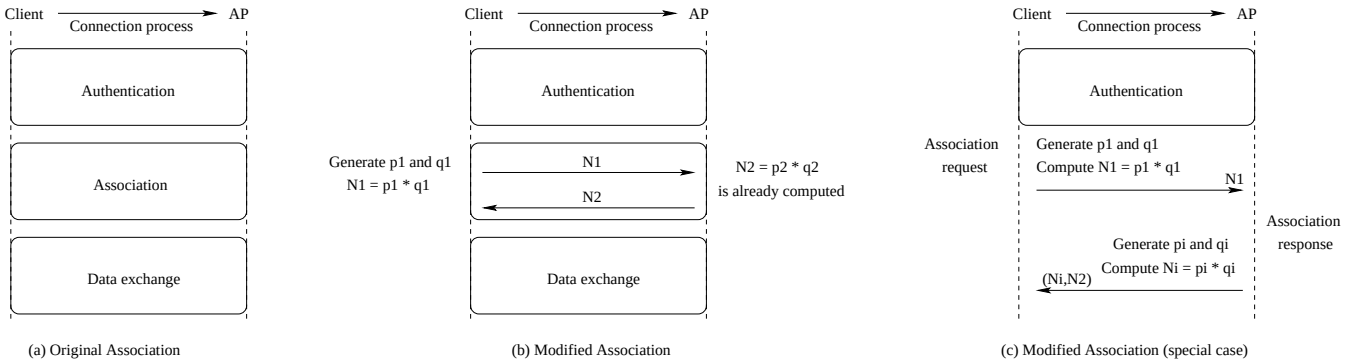
- Initially, the client randomly generates primes p_1 and q_1, then computes $N_1 = p_1 * q_1$. Similarly, the AP generates p_2, q_2 and computes $N_2 = p_2 * q_2$.
- During the authentication process between the client and the AP, the client sends an "envelop" that contains N_1 to the AP, and AP sends an "envelop" containing N_2 to the client.
- When the client wants to disconnect from the AP, it sends either the deauthentication or the disassociation frame to the AP, together with p_1 to the AP; we call this number "letter". If this "letter" corresponds to the "envelop" previously sent, i.e. $p_1|N_1$ (p_1 divides N_1) then the frame is authenticated and will be processed accordingly. Otherwise, the frame is rejected.
- Similarly, if the AP wants to disconnect from the client, it sends the disassociation/deauthentication frame together with p_2. The STA disconnects itself from the AP if $p_2|N_2$.
- The "Letter-envelop" protocol works because:
  - Since p and q are two large primes, even though the attacker can obtain N, it is difficult for her to correctly "guess" p. This is because the attacker must solve the hard "factorization" problem, which is intractable. Spoofing p will easily be detected, since the division operation N/p can be efficiently performed by the legitimate AP and STA.
  - Since the factorization of N is unique provided p and q are primes (meaning that there does not exist a pair $(p', q') \neq (p, q)$ such that $p' * q' = p * q = N$), only the client or AP who generated the "envelop" N can prove that they are the legitimate owner of the "letter" p, and thus can send the legitimate deauthentication/disassociation frame.

3.2. Implementation of Letter-envelop protocol

In 802.11 standards, the association session follows the authentication (Figure 1(a)). We modify the association process such that the client and the AP can authenticate each other whenever they receive a deauthentication or disassociation frame.

The modified association session (Figure 1(b)) is as follows:

- After the authentication is finished, the client randomly generates two large primes number p_1 and q_1, computes $N_1 = p_1 * q_1$ and includes N_1 in the Association Request frame (by putting the number in the frame body) sending to the AP.
- On receiving the Association Request frame, the AP checks whether the Association ID (AID) of the client exists in the memory. There are two cases:
  - Case 1: if the AID does not exist, then the STA has not been associated with the AP. The AP then stores N_1 in the memory corresponding to the client's record. If this is the first client associating to the AP, then the AP randomly generates two large primes p_2 and q_2, computes $N_2 = p_2 * q_2$ and includes it in the Association Response frame sending to the client. If there are clients already associated to the AP, then the AP just send the value of N_2 that has been previously sent to other clients. Note that the AP just needs one value of N_2 for all the STAs associated to



(a) Original Association (b) Modified Association (c) Modified Association (special case)

Figure 1. Association protocol

it. Whenever the AP wants to disconnect with all STAs, it just needs to broadcast a single disassociation frame containing N_2.

  - Case 2: The AID does exist, then the AP discards the Association Request frame.

- On receiving the Association Response frame, the client stores N_2 in the memory corresponding to the AP it is associating with.
- When the AP receives the deauthentication frame or disassociation frame with a value of "letter" k from a client, it checks in the memory the values of N_1 corresponding to that client. If $k|N_1$ then the AP clears the information related to that client in the memory.

Code	Reason
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 non-authenticated station
7	Class 3 frame received from non-associated station
8	Disassociated because sending station is leaving (or has left) BSS
9	Station requesting (re)association is not authenticated with responding station
10-65535	Reserved

Table 1. Reason code

802.11 standards do not specify any information except source and destination MAC addresses to verify the management frames, thus an attacker could easily launch the Farewell attack to disconnect the client from the AP. With the modified association protocol, the attacker cannot do that anymore.

In 802.11 standards, the deauthentication frame and the disassociation frame include the reason code in the frame body. Each of the reason code (Table 1) corresponds to a situation because of which the frame is sent. We consider situations where the client wants to disassociate from the AP and show how the new association protocol can effectively defend against the Farewell attacks:

1. The client is leaving (reason code = 3 or 8)
   - The client send a disassociation frame that includes p_1 to the AP
   - The AP looks for the value of N_1 that corresponds to the client in the AID table. If such value does not exist, ignore the frame. If N_1 is found, the AP checks if $p_1|N_1$. If it does, then accept and process the disassociation frame, otherwise ignore the frame.

   In this case, if the attacker attemps to spoof the disassociation frame of a legitimate client, she needs the MAC address of the client as well as the correct value of p_1 that the client has used to compute N_1. The MAC address can be sniffed, but p_1 cannot be efficiently guessed. Thus she cannot pretend to be an legitimate client to disassociate the client from the AP.
2. The AP goes offline (reason code = 3)
   - The AP broadcasts a disassociation frame that includes the value of p_2 to all clients in the network.
   - When receiving this frame, the client checks whether $p_2|N_2$. If it does, then the client disassociates with the AP, otherwise ignore the frame.

   Similarly to the case where the client disassociates with the AP, if the attacker wants to disconnect the clients, she needs to correctly guess p_2, which is intractable. Thus this attack cannot be easily launched by the attacker.

3. Reason code = 2,6,7,9. In this case the client has not been authenticated or has not been associated with the AP, thus the frame will be ignored.
4. Reason code = 1 (unspecified). The frame will be ignored.
5. Reason code = 5 (AP cannot handle any more client). This is a special case and rarely happens in reality. However, if supporting this situation is desirable, we suggest the modified protocol as follows. During the authentication, each client receives two values from the AP: N_2 is the common value which can be used by the AP to broadcast to all clients in case it goes offline; and $N_i(i \geq 3)$ that can be used by the AP to disassociate each individual client. If the AP wants to disassociate a single client to reduce the network's load, it sends the value of $p_i(i \geq 3)$ correspond to that client only. The client will accept to disassociate with the AP if $p_i|N_i(i \geq 3)$, otherwise it will ignore the frame. The modified Association protocol is illustrated in Figure 1(c).

4. Experiments

4.1. Farewell attacks on commercial 802.11 devices

As described in Section 2, there are many tools that can be used to launch the Farewell attacks, based on one principle: using the packet generator to send packets to the AP or the client. We use CommView for Wifi [3] for our experiment, since it enables us to capture the frames, modify the frame header as well as generate new frames to launch the attack. We continuously send deauthentication and disassociation frames to two commercial APs (Planet ADW-4301 and DLINK 624+) with spoofed MAC address of legitimate clients and as a result the clients are disconnected just in less than 2 seconds. This result is similar to what has been shown in [15].

4.2. Farewell attacks on modified 802.11 networks with Letter-envelop protocol

In this experiment we implement a simple system consisting of one AP, one client and an attacker. We simulate one legitimate client associating to the AP and one attacker trying to launch the Farewell attack. The legitimate client and the AP both are installed with the modified Association protocol that we proposed. We use a PC equipped with a wireless card to simulate the AP. The functionalities of this "AP" are exactly the same as other off-the-shell APs on the market. Our AP is implemented with two different authentication mechanisms: Open Authentication and Shared Key Authentication.

We use the following tools and library for the client and AP:

- Madwifi-0.9.3.3 [8]: this is an open source device driver for wireless cards that use Atheros chipset running on Linux operating systems. We reprogram the device driver to make it work as an kernel module for the client and AP following 802.11 standards with modified Association protocol described above.
- LibTomMath 0.41 [6]: this is a platform-independent library for manipulating large numbers. We modify this library so that it can be compiled with Madwifi as part of our kernel module for the client and AP running on Linux operating system.

The configuration of our system is as follows:

- One PC (CPU: Intel Celeron 3GHz, RAM: 1GB, HDD: 80GB) functioning as an AP.
- One PC (CPU: Intel Celeron 1.73GHz, RAM: 512MB, HDD: 80GB) functioning as a legitimate client. This client continuously sends ICMP ping packets to the AP to check the connection with the AP.
- One PC (CPU: Intel Core Duo 1.6Ghz, RAM: 512MB, HDD: 80GB) running CommView for WiFi to launch the Farewell attack.

We conduct the experiment as follows. We continuously send deauthentication and disassociation frames with spoofed MAC address of the client (to the AP) and of the AP (to the client) at the rate of 10 frames/second. If the AP can detect the frame to be a spoofed frame, they will ignore the frame and will not disconnect the client. Otherwise it will disconnect the client and clear information related to that client in the memory. We use different size of primes p and q as 64, 128, 256 and 512 bits. The corresponding value of $N = p * q$ would be 128, 256, 512 and 1024 bits.

The results of the experiments (Table 2) show that our solution is completely effective against the Farewell attack, none of the attacks is successful.

Length of N (bits)	Defense against Farewell attack	
	AP	Client
128	Yes	Yes
256	Yes	Yes
512	Yes	Yes
1024	Yes	Yes

Table 2. Experimental results

We also perform microbenchmarking to measure the time it takes for commercial handheld devices, which have hardware configuration close to that of commercial access points and wireless stations (CPU- 200 Mhz, RAM- 32

Mb), to perform operations such as generating primes, multiplication, division. We test 2 systems as follows: Nokia N80 (CPU- 220 Mhz, RAM- 40 Mb, OS- Symbian v9.1) and Nokia N70 (CPU- 220 Mhz, RAM- 30 Mb, OS- Symbian v8.1a). The results are as follows.

Operations	Time (seconds) for 512 bits number	
	N80	N70
Generate primes p and q	6.4863	10.8493
$N = p * q$	0.0156	0.0232
N/p	0.0158	0.2760

Table 3. Microbenchmarking results

The benchmarking results indicate that our solution can be efficiently implemented on commercial products. Our only concern is the time it takes for the AP to generate primes in case it needs to use one pair of prime for each client. In this case we suggest that the AP use pseudo primes, which are much faster to generate. Another option is that the AP can pre-generate a number of primes and store them in its database. One may argue that the attacker can also pre-generate all the primes and try to match the primes that the AP generates. However, it is shown in [20] that the probability that a number which is less than p is prime is about $1/\ln(p)$. If p is a 512-bit number, then there will be $\frac{2^{512}}{\ln(2^{512})} = \frac{2^{512}}{512\ln(2)} > \frac{2^{512}}{512} = 2^{503}$ primes. This large number of primes makes it infeasible for the attacker to store all the primes in order to match with the primes generated by the AP.

5. Conclusion

In this paper we apply the factorization problem to develop Letter-envelop protocol which can help 802.11 networks defend against Farewell attacks. Being employed as an extension to current 802.11 standards, the protocol can be easily deployed to the existing systems as well as future 802.11 devices. The APs and clients just need to upgrade the firmware to patch our modified protocol.

The experimental results show that our protocol is effective against Farewell attacks. However, the current device driver that we modified only works on wireless devices with Atheros chipset running on Linux. Our protocol would be more widely applicable if other hardware vendors upgrade the drivers that implement our protocol for their devices.

References

[1] Airjack: sourceforge.net/projects/airjack.

[2] Airsnarf: airsnarf.shmoo.com.

[3] CommView: http://www.tamos.com/products/commwifi/.

[4] FATA-jack: http://www.wi-foo.com/soft/attack/fata_jack.c.

[5] KisMAC: binaervarianz.de/projekte.

[6] LibTomMath: http://math.libtomcrypt.com/.

[7] MAC Changer: www.alobbs.com.

[8] MadWifi: http://madwifi.org/.

[9] SpoofMAC: www.klcconsulting.net/smac.

[10] Void11: www.wlsec.net/void11.

[11] WLAN-jack: 802.11ninja.net.

[12] F. Anjum, S. Das, P. Gopalakrishnan, L. Kant, and B. Kim. Security in an insecure WLAN network. In *International Conference on Wireless Networks, Communications and Mobile Computing*, pages 292–297, Maui, Hawaii, June 2005.

[13] W. Arbaugh, N. Shankar, Y. Wan, and K. Zhang. Your 802.11 Wireless Network Has No Clothes. *IEEE Wireless Communications*, 9(6):44–51, December 2002.

[14] B. Aslam, M. Islam, and S. Khan. 802.11 Disassociation DoS Attack and Its Solutions: A Survey. In *Proceedings of the First Mobile Computing and Wireless Communication International Conference*, pages 221–226, Amman, Jordan, September 2006.

[15] J. Bellardo and S. Savage. 802.11 Denial-of-Service attacks: real vulnerabilities and practical solutions. In *Proceedings of the 12th conference on USENIX Security Symposium*, pages 15–28, Washington, DC, 2003.

[16] E. D. Cardenas. MAC Spoofing: An introduction. In *www.giac.org/practical/GSEC*.

[17] Chibiao Liu. 802.11 Disassociation Denial of Service (DoS) attacks: www.mnlab.cs.depaul.edu/seminar/spr2005.

[18] F. Guo and T. Chiueh. Sequence Number-Based MAC Address Spoof Detection. In *Proceedings of 8th International Symposium on Recent Advances in Intrusion Detection (RAID)*, Seattle, Washington, September 2005.

[19] H. Johnson, A. Nilsson, J. Fu, S. 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 (GLOBECOM)*, Taipei, Taiwan, November 2002.

[20] K. H. Rosen. *Elementary Number Theory and Its Application, 3rd Edition*. 1993.

[21] H. Wang and A. Velayutham. An enhanced one-bit identity authentication protocol for access control in IEEE 802.11. In *Proceedings of IEEE Military Communications Conference (MILCOM)*, October.

[22] E. Wright. Detecting Wireless LAN MAC Address Spoofing. In *http://forskningsnett.uninett.no/wlan/download/wlan-mac-spoof.pdf*.

[23] H. Xia and J. Brustoloni. Detecting and Blocking Unauthorized Access in Wifi Networks. In *Proceedings of 3rd Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*, Athen, Greece, May 2004.