DENIAL-OF-SERVICE ATTACKS AGAINST THE 4-WAY WI-FI HANDSHAKE

Mathy Vanhoef and Frank Piessens

imec-DistriNet, KU Leuven

ABSTRACT

The 4-way Wi-Fi handshake is used to negotiate fresh pairwise keys, and authenticates both the client and Access Point (AP). We analyze this handshake, and discover several new denial-of-service (DoS) attacks against it. Interestingly, our attacks work even if Management Frame Protection (MFP) is enabled.

The first attack abuses the observation that messages in the 4-way handshake undergo link-layer encryption once the pairwise key is installed. More precisely, when message 4 of the handshake is dropped, the handshake times out. The second attack is similar to the second one, but induces the AP into sending the first message 4 with link-layer encryption. Again, this causes the handshake to time out. In the third attack, an adversary waits until the victim completes the 4-way handshake. Then she initiates a rekey by injecting a malformed 4-way handshake messages, causing several implementations to disconnect the client from the network. Finally, we propose countermeasures against our discovered attacks.

KEYWORDS

Network Protocols, Wi-Fi, 802.11, Denial-of-Service attacks, 4-way handshake

1. INTRODUCTION

Nowadays, wireless networks are usually based on the 802.11 standard, which is more widely known under the name Wi-Fi. This standard has gained major attraction over the years, and is currently used in a plethora of scenarios, ranging from personal use to reliability-critical industrial use. Because adversaries can monitor (and interfere with) wireless transmissions remotely, it is essential to protect the privacy and security of transmitted data. Initially, the 802.11 standard provided Wired Equivalent Privacy (WEP) to protect data. Unfortunately, it contained major design flaws, and is considered completely broken [1, 2, 3]. Instead, nearly all modern networks rely on Wi-Fi Protected Access (WPA) to encrypt data [4].

Both version 1 and 2 of WPA use a 4-way handshake for authentication, and for the negotiation of fresh pairwise keys. Even Wi-Fi networks that use 802.1x authentication, i.e., those that require a username and password, will use the 4-way handshake during the last step of the connection phase. As a result, is it critical that the 4-way handshake is reliable, and does not contain any security flaws. Given its high importance, several works have formally analyzed the security of this handshake [5, 6, 7]. However, even though these works discovered and addressed

Natarajan Meghanathan et al. (Eds) : NeTCoM, CSEIT, GRAPH-HOC, NCS, SIPR - 2017
pp. 85– 94, 2017. © CS & IT-CSCP 2017 DOI : 10.5121/csit.2017.71508

one denial-of-service (DoS) attack against the 4-way handshake, in practice implementations of the handshake still contain several deficiencies. In this work, we perform a detailed study of implementations of the 4-way handshake, and discover several new denial-of-service attacks against it.

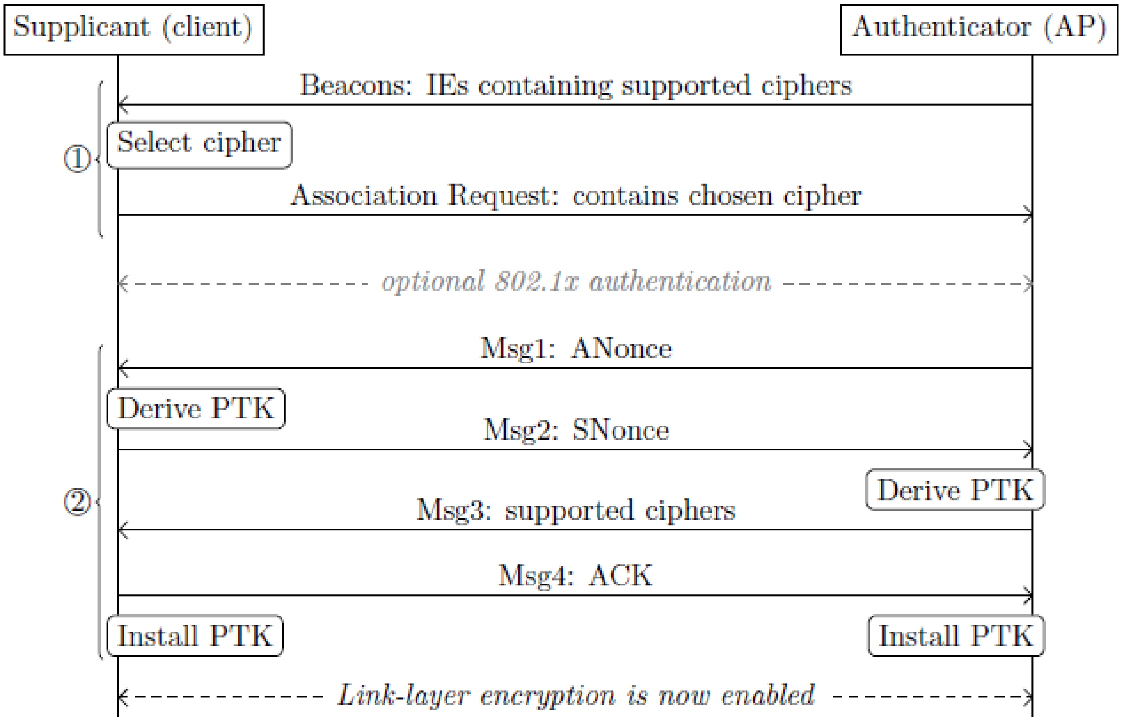
Supplicant (client)
Authenticator (AP)
Beacons: IEs containing supported ciphers
①
Select cipher
Association Request: contains chosen cipher
optional 802.1x authentication
Msg1: ANonce
Derive PTK
Msg2: SNonce
②
Derive PTK
Msg3: supported ciphers
Msg4: ACK
Install PTK
Install PTK
Link-layer encryption is now enabled

Figure 1: Simplified overview of the messages exchanged when connecting to a network, including the actions of the client (supplicant) and AP (authenticator). The 4-way handshake is illustrated in stage 2.

The remainder of this paper is structured as follows. First, Section 2 introduces the 4-way handshake and Management Frame Protection (MFP). Our novel denial-of-service attacks against the 4-way handshake are presented in Section 3. In Section 4 we propose countermeasures to the identified attacks. Finally, we present related work in Section 5 and conclude in Section 6.

2. BACKGROUND

In this section we first explain how stations discover nearby networks. Then we introduce the 4-way handshake, and its high-level interaction with link-layer confidentiality protocols. Finally, we explain how the Management Frame Protection (MFP) amendment of 802.11 works.

2.1. Network Discovery

An Access Point (AP) periodically broadcasts beacons to advertise its presence. This is illustrated in stage 1 of Figure 1. These beacons include the supported link-layer encryption algorithms (i.e., the supported ciphers) that are supported by the AP. This is either the Temporal Key Integrity Protocol (TKIP) or the CTR CBC-MAC Protocol (CCMP). When a client wants to connect to an

AP, and has selected a supported encryption algorithms to use, it sends an association request to the AP. This association request contains the encryption algorithm (cipher) selected by the client.

2.2. The 4-way Handshake

In order to start the 4-way handshake, the client and Access Point (AP) must first posses a shared secret called the Pairwise Master Key (PMK). The PMK is commonly derived from a pre-shared key, or from an 802.1X handshake. The 4-way handshake verifies that both entities posses the same PMK, generates a fresh Pairwise Transient Key (PTK), and confirms the selection of the cipher suite. It also synchronizes the installation of the PTKs. Note that once a PTK has been installed, all traffic is encrypted at the link-layer.

Stage 2 of Figure 1 shows the messages exchanged during the 4-way handshake. Simplified, the first two messages contain random nonces to generate a fresh PTK, and the last two messages protect against downgrade attacks. The handshake messages are defined using EAPOL frames, and we use the notation message n to refer to a specific message. After the client has transmitted message 4, it installs the PTK, while the AP installs the PTK after receiving message 4. Finally, to handle missed frames, the AP will retransmit the previous message if it did not receive a response. If the client already received this message, it will transmit a new response.

Finally, in an existing connection it is possible to rekey the PTK by executing a new 4-way handshake. During this rekey handshake, the EAPOL frames undergo link-layer encryption using the currently installed PTK.

2.3. Management Frame Protection (MFP)

A client can disconnect from a network by sending a deauthentication or disassociation frame to the AP. By default, these messages are not authenticated. As a result, an adversary can forge them to forcibly disconnect a client from a network. Continuously injecting these forged deauthentication packets causes a denial-of-service attack [8]. Fortunately, this attack can easily be prevented by using Management Frame Protection (MFP). This feature was introduced in the 802.11w amendment of the 802.11 standard. When this amendment is enabled, an adversary can no longer forge deauthentication or disassociation frames in order to disconnect a client.

3. DENIAL-OF-SERVICE (DoS) ATTACKS

In this section we present three novel denial-of-service (DoS) attacks against implementations of the 4-way handshake. The first exploits a race condition between installing the pairwise key and sending message 4. In the second attack we make the handshake fail by blocking message 4. Finally, we also discovered that injecting a malformed message 1 can cause the client to disconnect from the AP.

3.1. Encrypted Message 4 Race Condition

In the 802.11 standard, installing the PTK and sending an EAPOL message are both done by calling primitives in the MAC Sublayer Management Entity (MLME). However, in an actual implementation, these MLME primitives do not have to correspond similar interfaces [9, x6.3.1]. In practice, we indeed see that several operating systems use different kernel interfaces for installing the PTK and sending EAPOL messages. For instance, on Linux the nl80211 kernel interface is can be used to install the PTK, while the handshake messages are transmitted by the sendto system call. This means there is no guarantee in which order these two actions will be performed, and hence the PTK may be installed before message 4 is transmitted [10]. Additionally, message 4 may still be queued for transmission due to a busy medium, while the PTK is already being installed. As a result message 4 may undergo link-layer encryption, and will therefore be rejected by the AP. This will eventually cause the handshake to time out.

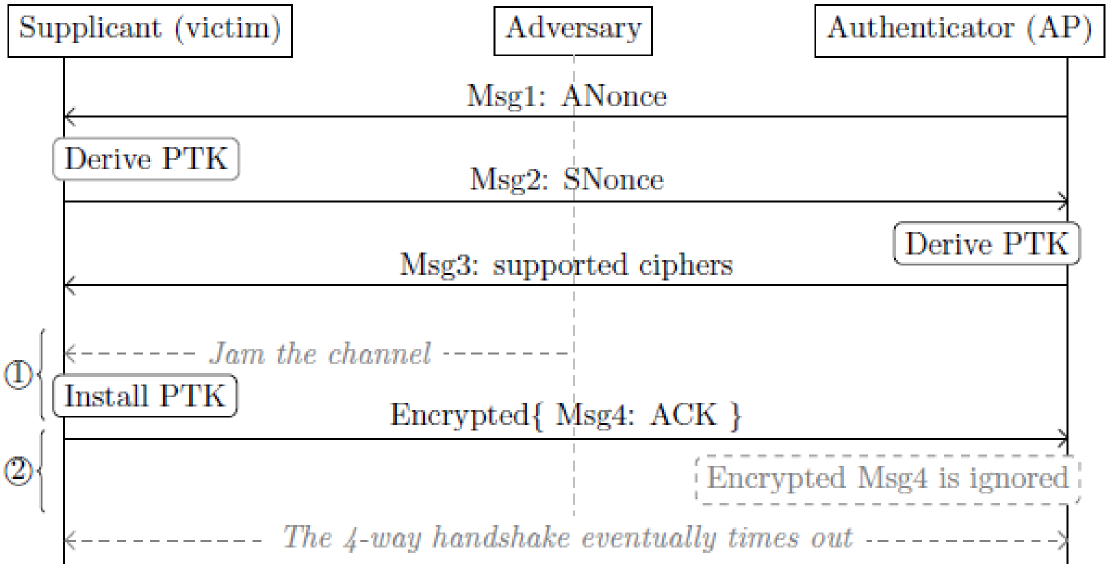

Figure 2: Encrypted message 4 race condition. Here the victim is induced into installing the pairwise key (PTK) before sending message 4.

Figure 2 illustrates this attack more clearly. At stage 1 of the attack, the adversary briefly jams the wireless channel. As a result, the victim queues message 3 for transmission until the wireless channel is no longer busy, i.e., until the adversary stops jamming. While message 4 is queued, the victim already installs the pairwise key. When the adversary now stops jamming, the victim will send message 4 using link-layer encryption, since the PTK has already been installed. However, the AP has not yet installed the pairwise key, and therefore will reject frames that underwent link-layer encryption. This illustrated in stage 2 of Figure 2. Since the AP never receives message 3, the 4-way handshake will eventually time out. We tested this attack against OpenBSD 6.0, which was acting as a client using a Sitecom WL-172 v1 wireless NIC, and confirmed the vulnerability.

3.2. Blocked Message 4

If message 4 of the handshake does not reach the AP, the handshake will never successfully complete. This is because the client installs the PTK after transmitting message 4, meaning it now only accepts frames that are encrypted at the link-layer. However, the AP will retransmit message 3 without link-layer encryption when it did not receive message 4. The client rejects this unencrypted message 3, and hence will not retransmit message 4. Eventually, the AP reaches its retransmission limit, and will abort the handshake. An attacker can abuse this as an e cient denial-of-service attack, by selectively jamming message 4. Note that selectively jamming frames is possible using cheap Wi-Fi USB dongles [11].

Figure 3 illustrates this attack. During state 1 of the attack, the adversary blocks message 4 from arriving at the AP using a selective jammer. Immediately after the victim transmitted message 4, she will install the pairwise key (PTK). At this point, the victim only accepts frames that are encrypted at the link-layer. When the AP now retransmits message 3 without link-layer encryption, the victim will reject it. This is illustrated in stage 2 of Figure 3. Finally, since the AP never receives message 4 of the handshake, the handshake will eventually time out and be aborted.

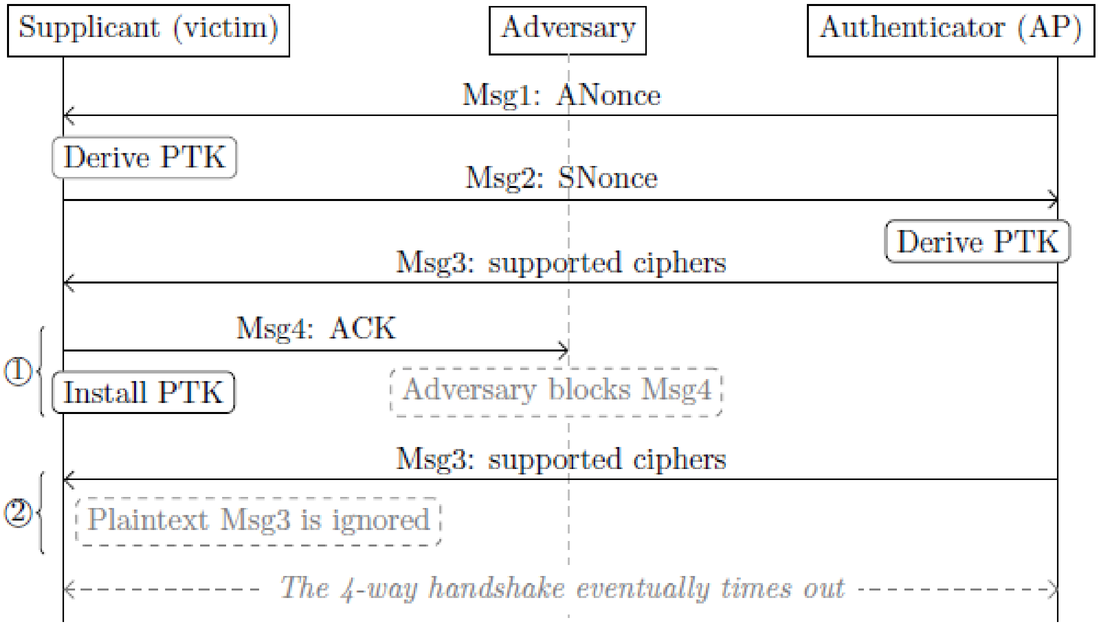

Figure 3: Blocked message 4 attack. The first transmission of message 4 is blocked, after which the victim installs the PTK, meaning retransmissions of message 3 will be ignored.

We remark that some implementations of the 4-way handshake always accept plaintext handshake messages, even when a PTK has been installed. That is, by testing several implementations, we discovered that some accept plaintext EAPOL frames (i.e., handshake frames) at any moment (see column 2 in Table 1). Allegedly this is done because some implementations transmit certain EAPOL frames without link-layer encryption (e.g. group key updates), even though both the client and AP already installed the PTK [12]. Even if plaintext handshake message are always accepted, the attack in Figure 3 still causes a denial-of-service. This is because nearly all implementations will transmit message 4 using link-layer encryption once a PTK has been installed, and the AP will reject these frames since it did not yet install the PTK (similar to the attack in Figure 2). Interestingly, this behaviour in contradicts the 802.11 standard. More precisely, the standard states that in the initial 4-way handshake, message 4 should always be sent without link-layer encryption [9, x11.6.6.5]. However, few implementations follow this requirement. Only MediaTek (re)transmits message 4 without link-layer encryption (see column 3 in Table 1).

3.3. Failed Rekey Using a Malformed Message 1

In the precious section we observed that several implementations accepted plaintext EAPOL frames, even though they already installed the PTK (recall column 1 in Table 1). We can abuse this behaviour to inject a forged message 1 towards the client. Note that this is valid behavior, since the AP can decide at any moment to refresh the pairwise keys (PTK) by starting a new 4-way handshake. More importantly, if this message contains invalid or malformed data, the client will abort the handshake, and subsequently disconnect from the network.

One way to create a malformed message 1, is by including an invalid PMKID alongside the ANonce (see stage 1 of Figure 4). Recall from section 2 that the PMK is the shared secret between the client and AP, and can either be derived from a preshared key, or from an 802.1X authentication. In general though, it is possible for the client and AP to share multiple valid PMKs. Therefore, the first message of the 4-way handshake may include a PMKID, which identifies the specific PMK that will be used [9, x11.6.6.2]. Note that the PMKID is essentially just a hash of the secret PMK. Interestingly, we found that many clients will abort the handshake and disconnect from the network when message 1 contains an unknown PMKID. Hence an adversary can cause a denial-of-service during an initial 4-way handshake, by injecting a message 1 with an unknown PMKID. Additionally, if the victim always accepts plaintext EAPOL frames, this message can even be injected after the initial 4-way handshake completed. Put differently, then our denial-of-service attack works even when the client has already installed the PTK. This attack latter variant of the attack is illustrated in stage 1 Figure 4, where the adversary injects a plaintext message 1 while the victim already installed a PTK.

In contrast to injecting deauthentication frames to disconnect a client, injecting a malformed message 1 is possible even if Management Frame Protection (MFP) is enabled. Indeed, if MFP is enabled, an adversary cannot forge deauthentication or disassociation frames. However, message 1 of the handshake can still be forged, and hence can still be abused to perform an efficient denial-of-service attack.

We tested this attack against Linux's wpa_supplicant. This confirmed that the client aborts the handshake and disconnects from the network when receiving a message 1 containing an invalid PMKID.

Table 1: Behaviour of several 4-way handshake implementations. The second column shows whether EAPOL frames that did not undergo link-layer encryption are accepted even if a PTK is installed. The third column shows whether message 4 is retransmitted without link-layer encryption during an initial 4-way handshake.

Implementation	Plaintext Reception	Plaintext (Re)transmission
FreeBSD	Yes	No
NetBSD	Yes	No
OpenBSD	No	No
Linux	Yes	No
Android	Yes	No
Windows	No	No
Apple	No	No
MediaTek	Yes	Yes
Broadcom	No	No

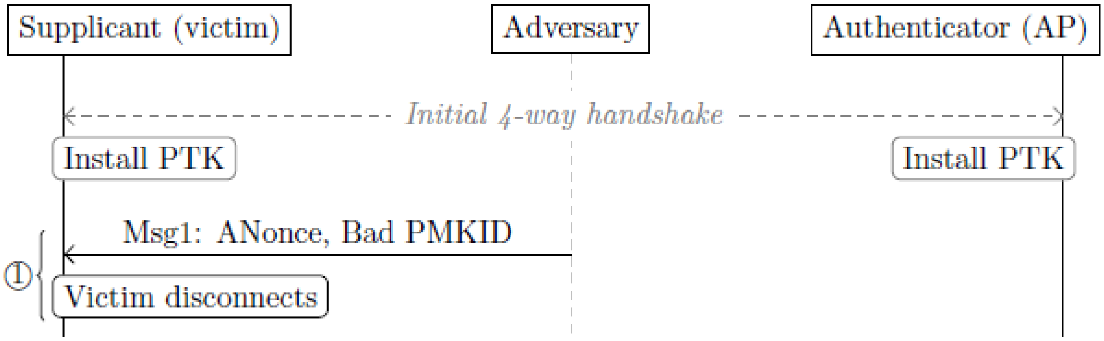

Figure 4: Instigating a failed rekey using a malformed message 1. This messages contains an invalid PMK ID, and in response the victim will disconnect from the AP.

4. PROPOSED COUNTERMEASURES

In this section, we propose countermeasures against the denial-of-service attacks we discovered, and explain their advantages and disadvantages.

4.1. Plaintext EAPOL Frames

The first two issues discussed in Section 3 can be dealt with simultaneously. In particular, we can simply send all EAPOL frames without link-layer encryption. Note that this does not negatively impact security, because sensitive information in EAPOL frames is already encrypted, and the full frame is also authenticated. Hence link-layer encryption is not required to protect the EAPOL frames used in the 4-way handshake. Not encrypting EAPOL frames at the link-layer is also consistent with RFC 5247, which states that \presence or absence of lower layer security is not taken into account in the processing of EAP messages" [13, x3.4]. Additionally, the formal security proof of the 4-way handshake also not require link-layer encryption. In other words, it is safe to send all EAPOL frames without link-layer encryption [5].

Unfortunately, only sending plaintext EAPOL frames introduces some compatibility issues. In particular, some implementations require that these frames are encrypted during a rekey. Sending them without encryption would therefore break compatibility. Nevertheless, in a first step, implementations can be modified so they always accept plaintext EAPOL frames. Additionally, they should always send plaintext EAPOL frames during the initial 4-way handshake even if a PTK is already installed. This does not introduce compatibility issues, but does solve the encrypted message 4 race condition, as well ass the dropped message 4 issue of Section 3. Once most implementations accept plaintext EAPOL frames at any moment, we can also send plaintext EAPOL frames during a rekey.

4.2. Handling a Malformed Message 1

To prevent the malformed message 1 attack of Section 3.3, an implementation should ignore such malformed messages. This is the approach that OpenBSD is currently using. There, if the client receives a malformed handshake message, it is simply dropped. Hence the client remains connected to the network, awaiting the real message 1 from the AP.

Another modification that should be made is that, during a rekey handshake, the authenticity of message 1 should be validated by the currently installed PTK. Note that during the initial 4-way handshake, message 1 is sent unauthenticated, since the AP does not yet know the PTK (recall Figure 1). However, during a rekey we can authenticate message 1 using the PTK that was negotiated in the previous 4-way handshake. This assures an adversary cannot forge any handshake messages once the initial 4-way handshake has completed.

5. RELATED WORK

Several works have analyzed the security of the 4-way handshake [5, 6, 7, 14]. In particular, He et al. discovered a denial-of-service vulnerability [5, 7], which led to the standardization of a sightly improved design of the 4-way handshake [15]. In their DoS attack, the adversary injects a forged message 1 using a different ANonce than the one the real AP is using. This causes the client to generate an invalid PTK, making the handshake fail. The solution is to make the client always use the same SNonce in a specific handshake, and to verify the ANonce when receiving message 3 of the handshake [5]. Note that message 3 is authenticated, and hence cannot be forged by an attacker. The advantage of our DoS attacks is that they are not yet addressed in the official 802.11 standard, and that several implementations are still affected by them.

Several other DoS attacks also exist against Wi-Fi networks that abuse different parts of the protocol. Arguably the most well-known of these is a deauthentication attack, where an adversary forges deauthentication frames to disconnect the client from the network [16]. This is possible because, by default, these messages are not authenticated. However, nowadays this attack can easily be prevented by enabling protected management frames [9, x4.5.4.9]. In contrast, our attacks remain possible even when protected management frames is enabled.

Several DoS attacks also exploit weaknesses in the link-layer encryption protocol called TKIP. In particular, Glass and Muthukkumarasamy [17] abuse the TKIP Michael countermeasures to make a Wi-Fi network unusable for one minute. Vanhoef and Piessens improved this attack, by removing the requirement of a man-in-the-middle position [4]. The advantage of our DoS attacks is that they can be execute on any Wi-Fi network, even if the network does not support the TKIP encryption algorithm.

Finally, Konings et al. found several DoS vulnerabilities in the physical and MAC layer of 802.11 [19]. Some of these make the network unusable for one minute, while others only do so for a brief amount of time. A survey of DoS attacks at the physical and MAC layer is given by Bicakci and Tavli [20]. Generally, these attacks require injecting a large amount of frames, while our attacks require a lower amount of packets to be injected.

6. CONCLUSION

Implementations of the 4-way handshake (still) contain several denial-of-service vulnerability. This in spite of previous analysis and security proofs of the 4-way handshake. Our attacks can be mitigating by always sending and accepting plaintext EAPOL frames. Because this may introduce compatibility issues, we first recommend doing this only during the initial 4-way handshake. This does not introduce compatibility issues, while already making the initial 4-way handshake more secure and robust. In a second step, implementations can also send plaintext EAPOL frames during a rekey. Finally, when implementations receive a malformed message 1, e.g., if the message contains an unknown PMKID, the message should be ignored and dropped.

REFERENCES

[1] S. R. Fluhrer and D. A. McGrew, \Statistical analysis of the alleged RC4 keystream generator," in FSE, 2000.

[2] A. Stubblefield, J. Ioannidis, A. D. Rubin, et al., \Using the uhrer, mantin, and shamir attack to break wep.," in NDSS, 2002.

[3] A. Bittau, M. Handley, and J. Lackey, \The final nail in WEP's coffin," in IEEE SP, 2006.

[4] M. Vanhoef and F. Piessens, \Practical verification of WPA-TKIP vulnerabilities," in ASIA CCS, pp. 427{436, ACM, 2013.

[5] C. He and J. C. Mitchell, \Analysis of the 802.1 i 4-Way handshake," in WiSe, ACM, 2004.

[6] C. He, M. Sundararajan, A. Datta, A. Derek, and J. C. Mitchell, \A modular correctness proof of IEEE 802.11i and TLS," in CCS, 2005.

[7] J. Mitchell and C. He, \Security analysis and improvements for IEEE 802.11i," in NDSS, 2005.

[8] S. Park, K. Kim, D. Kim, S. Choi, and S. Hong, \Collaborative QoS architecture between DiffServ and 802.11e wireless LAN," in Vehicular Technology Conference, 2003.

[9] IEEE Std 802.11-2012, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Spec, 2012.

[10] J. Malinen, \wpa keyhandshake question / bug." Retrieved 19 March 2017 from http://lists.shmoo.com/pipermail/hostap/2005-May/010370.html, 2005.

[11] M. Vanhoef and F. Piessens, \Advanced Wi-Fi attacks using commodity hardware," in ACSAC, 2014.

[12] J. Malinen, \Re: Dealing with retransmitted EAPOL msg 3/4 and 4/4." Retrieved 19 March 2017 from www.spinics.net/lists/hostap/msg03309.html, 2017.

[13] B. Aboba, D. Simon, and P. Eronen, \Extensible authentication protocol (EAP) key management framework." RFC 5247, 2008.

[14] L. Wang and B. Srinivasan, \Analysis and improvements over DoS attacks against IEEE 802.11i standard," in NSWCTC, 2010.

[15] IEEE Std 802.11-2016, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Spec, 2016.

[16] J. Bellardo and S. Savage, \802.11 denial-of-service attacks: real vulnerabilities and practical solutions," in Proc. of the 12th USENIX Security Symp., 2003.

[17] S. M. Glass and V. Muthukkumarasamy, \A study of the TKIP cryptographic dos attack," in 15th International Conference on Networks, IEEE, 2007.

[18] M. Morii and Y. Todo, \Cryptanalysis for RC4 and breaking WEP/WPA-TKIP," IEICE Transactions, pp. 2087{2094, 2011.

[19] B. Konings, F. Schaub, F. Kargl, and S. Dietzel, \Channel switch and quiet attack: New DoS attacks exploiting the 802.11 standard," in LCN, 2009.

[20] K. Bicakci and B. Tavli, \Denial-of-service attacks and countermeasures in IEEE 802.11 wireless networks," Comput. Stand. Interfaces, vol. 31, no. 5, 2009.