Operating Channel Validation: Preventing Multi-Channel Man-in-the-Middle Attacks Against Protected Wi-Fi Networks

Mathy Vanhoef
imec-DistriNet, KU Leuven
Mathy.Vanhoef@cs.kuleuven.be

Nehru Bhandaru
Broadcom Ltd.
nehru.bhandaru@broadcom.com

Thomas Derham
Broadcom Ltd.
thomas.derham@broadcom.com

Ido Ouzieli
Intel Ltd.
ido.ouzieli@intel.com

Frank Piessens
imec-DistriNet, KU Leuven
Frank.Piessens@cs.kuleuven.be

ABSTRACT

We present a backwards compatible extension to the 802.11 standard to prevent multi-channel man-in-the-middle attacks. This extension authenticates parameters that define the currently in-use channel.

Recent attacks against WPA2, such as most key reinstallation attacks, require a man-in-the-middle (MitM) position between the client and Access Point (AP). In particular, they all employ a multi-channel technique to obtain the MitM position. In this technique, the adversary acts as a legitimate AP by copying all frames sent by a real AP to a different channel. At the same time, the adversary acts as a legitimate client by copying all frames sent by the client to the channel of the real AP. When copying frames between both channels, the adversary can reliably manipulate (encrypted) traffic. We propose an extension to the 802.11 standard to prevent such multi-channel MitM attacks, making exploitation of future weaknesses in protected Wi-Fi networks harder, to practically infeasible. Additionally, we propose a method to securely verify dynamic channel switches that may occur while already connected to a network.

Finally, we implemented a prototype of our extension on Linux for both the client and AP to confirm practical feasibility.

1 INTRODUCTION

Traditional attacks against protected Wi-Fi networks only require an attacker to sniff and (optionally) inject packets. For example, Wired Equivalent Privacy (WEP) can be broken by passively sniffing packets [7], dictionary attacks against WPA2 merely require a passive capture of the 4-way handshake [5], and breaking Wi-Fi Protected Setup (WPS) only relies on interactions with an Access Point (AP) [22]. Put differently, none of these attacks require a man-in-the-middle (MitM) position between the client and AP. In contrast, recent attacks do require a MitM position. This is because in these attacks, the adversary must be able to reliably manipulate (i.e. block, delay, or modify) encrypted packets. For example, certain key reinstallation attacks against WPA2 require the ability to block packets [20]. Similarly, certain attacks against (WPA-)TKIP require a MitM position [15, 18], as do other recent attacks against the 4-way handshake and encryption algorithms of WPA2 [9, 19, 21].

To obtain a MitM position in a protected Wi-Fi networks, these recent attacks rely on a multi-channel technique [9, 18–20]. This is also called a channel-based MitM position [18]. It is very effective, stealthy, reliable, and to the best of our knowledge, the only method usable under all practical circumstances (see Section 5.1). The idea behind a multi-channel MitM is to clone the AP on a different channel, trick a client into connecting to the AP on this rogue channel, to then forward frames between both channels so the client and AP can communicate. This enables an adversary to reliably delay, block, or modify frames sent between the client and AP. Since all recent attacks that require a MitM rely on this multi-channel technique [9, 18–20], preventing it mitigates these attacks, and hardens implementations against future weaknesses.

The idea behind our defense is to cryptographically authenticate the parameters that define the operating channel. This require us to first define an unambiguous encoding of the operating channel, and then authenticating this information when connecting to a protected Wi-Fi network. We also propose a method to securely verify dynamic channel switches that may occur while clients are already connected to a network. Note that the general Wi-Fi industry is receptive of defining such a mechanism to prevent multi-channel MitM attacks [17, Slide 18], and we are working on submitting our proposal for inclusion in the 802.11 standard [3]. Finally, we implemented our proposal to confirm practical feasibility.

To summarize, our main contributions are:

- We propose a backwards-compatible extension to the 802.11 standard to detect and prevent multi-channel MitM attacks.
- We design a mechanism to securely confirm dynamic channel switches by extending the SA query procedure.
- We implement and evaluate our proposed extension.

The remainder of this paper is organized as follows. Section 2 introduces relevant aspects of the 802.11 standard, and Section 3 introduces the multi-channel MitM attack. Our defense is proposed in Section 4, and evaluated in Section 5. Finally, we discuss related work in Section 6 and conclude in Section 7.

2 BACKGROUND

This section introduces aspects of the 802.11 standard [11] that are essential for understanding both the multi-channel MitM attack, and our proposed defense against it.

WiSec '18, June 18–20, 2018, Stockholm, Sweden

ACM ISBN 978-1-4503-5731-9/18/06…$15.00
https://doi.org/10.1145/3212480.3212493

2.1 Protected Wi-Fi Networks

All modern protected Wi-Fi networks rely on a Robust Security Network (RSN) association to guarantee secure communications. A subset of RSN is certified by the Wi-Fi Alliance for interoperability under the more well-known name Wi-Fi Protected Access version two (WPA2). The most important features of a RSN association are mutual authentication, and encryption of data frames once a session key has been negotiated. Several handshakes exists that provide both mutual authentication and session key negotiation:

The 4-way handshake. This handshake is used in all WPA2 networks, and is always executed when connecting to a certain network for the first time. It consists of four messages, which are defined using EAPOL-Key frames. All messages except the first are integrity protected (authenticated) using a Message Integrity Code (MIC).

Authenticated Mesh Peering Exchange (AMPE). This is the equivalent of the 4-way handshake for mesh networks. It consists of two Open and two Confirm messages, and also contains a Close message to terminate the connection. All frames are authenticated.

Fast BSS Transition (FT) handshake. This handshake is used to roam from one AP to another . Handshake messages are encapsulated in authentication and (re)association frames. Only handshake data in (re)association frames is authenticated with a MIC.

Fast Initial Link Setup (FILS) handshake. This handshake establishes a secure link and Internet connection in only 100ms. Handshake messages are encoded in authentication and (re)association frames. Similar to the FT handshake, only handshake data encoded in (re)association frames is authenticated.

Tunneled Direct-Link Setup (TDLS). This protocol establishes a direct tunnel between two clients. To secure the tunnel, the DTLS PeerKey (TPK) handshake is used, where its messages are sent through the AP. To assure the negotiated key is secret, there must already be a secure tunnel between both clients and the AP.

Group Key (GTK) handshake. This handshake is not used to negotiate a session key, but to transport a new group key to all associated clients. It is defined using EAPOL-Key frames, and consists of two messages which are both protected using a MIC.

Unfortunately, recently most of these handshakes were found vulnerable to key reinstallation attacks [20]. This attack tricks a victim into reinstalling an already-in-use key, causing nonce reuse in the encryption algorithm, voiding any security guarantees. Performing most key reinstallation attacks requires a MitM position. Thankfully, implementations can be patched in a backwards-compatible manner to prevent key reinstallations. Moreover, the Wi-Fi Alliance recently announced the WPA3 certification, which mandates support of more modern security features [27]. However, none of these features will prevent multi-channel MitM attacks.

2.2 Management Frame Protection (MFP)

Since 2018, the WPA2 certification mandates that management frames are protected using MFP [26]. Before this it was an optional feature, meaning that it currently is still possible to forge management frames in most networks. For instance, deauthentication frames can be forged to forcibly disconnect clients from Wi-Fi networks [2]. However, such attacks can be prevented by enabling Management Frame Protection (MFP).

When a station, i.e. a client or AP, supports MFP, robust management frames are cryptographically protected. This encompasses disassociation, deauthentication, and robust action frames [11, §12.2.8]. In an infrastructure network, nearly all action frames are robust. For an overview of which action frames are not robust, see [11, §9.4.1.11]. Unicast robust management frames are both encrypted and authenticated, but group-addressed frames are only authenticated.

Authentication and (re)association frames are not protected when using MFP. This is because these frames are sent before keys are negotiated. Although forged authentication frames pose no problem, since the AP can simply ignore them [11, §11.3.2], forged (re)association frames cause the AP to reset the existing connection. To defend against this, MFP adds a Security Association (SA) tear down protection procedure. Under this procedure, when the AP receives a (re)association request for an already-associated client that uses MFP, the AP will not change the state of the client. Instead, the AP transmits a SA query request to the client [11, §11.3.5.3]. In case the client responds with a SA query response, nothing happens. But if the client does not respond timely, subsequent association requests are processed without a SA query procedure. Therefore, as long as the real client is connected and responding to the SA query requests, the connection will not be reset. Finally, we remark that a client can also send SA query requests to the AP [11, §11.14].

2.3 WNM Sleep Mode

Wireless Network Management (WNM) sleep mode enables clients to sleep for arbitrary long periods. To enter or exit this sleep mode, the client sends a WNM-sleep mode request frame. The AP replies in both cases with a WNM-sleep mode response frame. While in WNM sleep mode, clients do not have to wake up for group key updates. If the group key was renewed while the client was asleep, and MFP is enabled, the WNM-sleep mode response frame will contain the new group key. Otherwise, if MFP is not used, the AP will execute a new group key handshake when the client wakes up.

2.4 Channel Properties

Stations can operate on several frequencies and bandwidths. Currently, the 802.11 standard supports the traditional 20 MHz bandwidth channels, as well as 40, 80, and 160 MHz bandwidths. These wide-bandwidth channels also define a primary 20 MHz channel that can be used when the full wide-bandwidth channel is occupied. Finally, 80+80 MHz channels are also supported. Here, two non-contiguous 80 MHz channels are used as one virtual channel.

3 MULTI-CHANNEL MITM

This section explains the multi-channel MitM attack [18]. Note that its goal is not to decrypt traffic, but to enable reliable manipulation of (encrypted) frames. We also provide an overview of both existing and novel attacks that rely on this MitM position.

3.1 Background

Reliably monitoring and manipulating traffic in a protected Wi-Fi network is not trivial. First, when monitoring frames, some will be missed due to background noise. Second, it is hard to block and modify frames. While a selective jammer can block some frames, it is unreliable, and does allow the attacker to receive the full frame [18]. Another method to manipulate traffic is by establishing a MitM position. However, this is difficult in a protected Wi-Fi network because the negotiated session key depends on the MAC address of both the client and AP. Hence, if we use a rogue AP with a different MAC address than the real AP, and then forward frames between the client and AP, the handshake will fail. Using the same MAC address as the real AP is also not an option, because then the client and AP would simply communicate with each other directly.

3.2 Obtaining a Multi-Channel MitM Position

To reliably intercept all traffic, the adversary can clone the AP on a different channel, and forward traffic between both channels [18]. Here the adversary first copies all beacons of the AP, which is on channel A, to a different channel B. Note that this requires two Wi-Fi antennas: one operating on channel A, and one operating on channel B. Once the rogue AP is visible on channel B, we force clients to connect to it. Originally a continuous jammer was used to accomplish this [18], by jamming the channel of the real AP. When the client has switched to channel B, the jammer is stopped, so frames can now be forwarded between the client and AP.

3.3 Channel Switch Announcements (CSAs)

We discovered a novel method to force clients into connecting to the rogue AP. In particular, an adversary can forge Channel Switch Announcements (CSAs) to force a client to switch to the rouge AP's channel. Normally these announcements are sent when the AP is switching to a different channel. For instance, when an AP operates on certain 5 GHz channels and detects (weather) radar pulses, it must switch to a different channel to comply with regulations.

Channel switch announcements can be broadcasted using three different frames [11, §9.4.2.19]. The first is by including a CSA element inside a beacon. The second method is by including a CSA element inside a probe response. And the third technique is to send action frames with a CSA element to associated clients. The first two methods use unprotected management frames, and the third method is protected when MFP is enabled. As a result, even when MFP is used, an adversary can forge CSA elements inside beacons and probe responses to force clients to switch channels.

3.4 Security Impact

The multi-channel MitM has been used in several works. For example, it was used to attack the (WPA-)TKIP encryption protocol [18], to perform downgrade attacks against the 4-way handshake [19, 21], to manipulate encrypted web traffic [9], and to trigger key reinstallations against WPA2 [20]. In general, this MitM position can be used to reliably manipulate, delay, replay, and block frames. Apart from these known attacks, the MitM can also be abused for other purposes. We will list new attacks to further illustrate the impact of the MitM, but a leave more extensive analysis as future work:

3.4.1 SA Query Suppression. The MitM position can be used to bypass the SA query mechanism used to defend against unprotected management frames when MFP is enabled. In particular, after obtaining a MitM position, the adversary can inject (re)association frames, and block the resulting SA Query frames. This means subsequent unprotected (re)association frames will be accepted, which will reset the existing connection, causing a DoS (recall Section 2.2).

3.4.2 Manipulate Capabilities. When copying beacons, probe responses, association frames, and so on, the adversary can modify advertised or requested capabilities that are not securely authenticated. One example are the supported bitrates of a device (see Section 5.1). A systematic analysis of all capabilities and their security impact is out of scope for this paper, and left as future work.

3.4.3 Influence Timing Measurements. The 802.11 standard also contains a Fine Timing Measurement (FTM) procedure [11, §11.24.6]. This procedure allows two devices to determine the distance between themselves with high accuracy [25]. We conjecture an adversary can alter the resulting range estimation with a MitM position.

3.4.4 A-MSDU Message Confusion. A MitM position can also be abused to convert an ordinary MAC Service Data Unit (MSDU) frame into an Aggregated MSDU (A-MSDU). This is possible because the header specifying whether a frame is a normal or A-MSDU frame is not authenticated. While the Payload Protected (SPP) A-MSDU feature would prevent this attack, few networks support this. Converting a MSDU into an A-MSDU frame changes how the decrypted data is interpreted by the receiver. This voids the security guarantees of authenticated encryption, and we conjecture it can be abused to leak (a small amount of) encrypted data.

4 OPERATING CHANNEL VALIDATION

In this section we present our extension to prevent multi-channel MitM attacks. First we unambiguously represent the current channel, then we show how to authenticate this information, and finally we propose a method to securely verify dynamic channel switches.

4.1 Operating Channel Information

Our defense will authenticate the operating channel that is used between two stations. For example, we authenticate the channel used to receive and send frames, for both an AP and each associated client. This requires us to precisely encode the channel being used, since any ambiguity might otherwise be exploited by attackers. To accomplish this, three problems must be addressed. First, properties of a physical channel can depend on the regulatory domain (i.e. country) a device is operating in. We handle this by relying on global operating classes as defined in Table E-4 of [11]. An operating class represents a set of channels in a specific regulatory domain. To pick a unique channel within the selected regulatory domain, the global operating class is combined with a channel number. This matches the practice of modern standards such as Neighbor Awareness Networking [23] and Wi-Fi Agile Multiband [24].

The second problem is how to precisely encode wide-bandwidth channels such as 40, 80, or 160 MHz channels. Recall from Section 2.4 that we must also specify the 20 MHz primary channel being used. Interestingly, usage of a wide-bandwidth channel is already encoded by the selected global operating class. And when combined with the channel number of the primary 20 MHz, this fully identifies the wide-bandwidth channel. For instance, usage of a 40 MHz channel is specified using a global operating class identifier. A channel number then defines the primary 20 MHz channel, which together with the

ID	Length	ID Ext.	Operating Class	Primary Channel No	Frequency Segment 1

Figure 1: The Operating Channel Information (OCI) element. All fields are encoded by one byte.

global operating class defines the full 40 MHz channel (i.e. whether the other 20 MHz is above or below the primary 20 MHz).

The third problem is encoding an 80+80 MHz wide-bandwidth channel (recall Section 2.4). Note that these also use a primary 20 MHz channel as a fall-back when the full-bandwidth channel is unavailable. To define an 80+80 MHz channel, we use the operating class to specify usage of 80 Mhz channels, and use the primary channel number to define the primary 20 MHz channel. This implicitly defines the first 80 MHz channel (commonly called frequency segment 0). A second channel number, called frequency segment 1, now defines the other 80 MHz channel.

Figure 1 illustrates the resulting encoding. The first three fields are used to define the Operating Channel Information (OCI) type-length-value element. The operating class field contains the global operating class as defined in Annex E of [11]. Together with the primary channel number this defines the used channel and the primary 20 MHz channel for wide-bandwidth channels. Finally, for 80+80 MHz channels, the frequency segment 1 channel number defines the second segment of the 80+80 MHz channel. This last field is only used for 80+80 MHz channels, and set to zero otherwise.

For example, a typical 20 MHz network on channel 11 is represented by a global operating class identifier of 81 with a primary channel number of 11. The frequency segment 1 field is not used and set to zero. As a more complex example, take an 80 MHz network operating on channel 155 (i.e. in the frequency range 5735-5815 MHz) with 153 as its primary 20 MHz channel. This channel is represented using a global operating class identifier of 128, and a primary channel number of 153. The frequency segment 1 field is not used and set to zero. As a final example, take an 80+80 MHz network with its first 80 MHz segment on channel 155, and with its primary 20 MHz on channel 153, and its secondary 80 MHz segment on channel 42. Then this 80+80 MHz channel would be represented using a global operating class identifier of 128, a primary channel number of 153, and a frequency segment 1 channel number of 42.

4.2 Operating Channel Validation

Our goal is to detect and prevent multi-channel MitM attacks, before encrypted data is transmitted. We accomplish this by authenticating and validating exchanged OCI elements. Recall that this element represents the current operating channel of a station. Since encrypted data can only be exchanged after installing a session key, a natural location to perform channel validation is during session key negotiations. In other words, we will include (authenticated) OCI elements in every handshake that negotiates a session key.

The OCI element included in a handshake message describes the channel used by the transmitter, with the operating class representing the widest bandwidth being supported. For example, if an 80 MHz AP has an associated client that only supports 40 MHz, the AP must include an OCI element representing the 80 MHz channel. In turn, this client will construct an OCI element that represents the 40 MHz channel it is using. Additionally, the OCI element must represent this channel, even if the handshake message happens to be sent over a smaller bandwidth channel. For example, when the full 40 MHz bandwidth is (temporarily) occupied, and the client uses the primary 20 MHz channel to send the handshake message, the included OCI must still represent the 40 MHz channel.

On reception of a handshake message that must contain an OCI element, the receiver verifies that: (1) an OCI element is present; and (2) the primary channel (and operating class) used to communicate with the other station matches the one in the OCI element; and (3) the maximum bandwidth used to communicate with the other station matches the bandwidth of the operating class in the OCI; and (4) for 40 MHz channels, the location of the non-primary channel matches the operating class in the OCI; and (5) frequency segment 1 matches the one in the OCI when using an 80+80 MHz channel. In case any of these checks fail, the handshake message should be silently ignored, causing the handshake to eventually timeout. Additionally, no channel switches are allowed during a handshake (see Section 4.4). In case there is a channel switch, the handshake must be aborted [3], after which it can optionally be restarted.

Several handshakes exist that negotiate a session key where OCI elements must be included (for implementation details see [3]):

The 4-way Handshake An OCI element must be included in message 2 and 3 of the handshake. The element can either be directly included as an extra information element in the key data field of the EAPOL-Key frame, or encoded using a new Key Data Encapsulation (KDE) entry.

The AMPE Handshake An OCI element must be directly included in all Open and Confirm messages. The element will automatically be authenticated using a pairwise master key.

The FT Handshake An OCI element must be included in both (re)association frames. The OCI element can be added to the Fast BSS Transition Element (FTE) as a subelement. Note that the FTE, and hence also the OCI, is authenticated.

The FILS Handshake An OCI element must be included in both (re)association frames. It can be directly included as an information element, and will then automatically be authenticated by the FILS handshake.

The TDLS PeerKey Handshake Channel validation is not required for this handshake. Recall from Section 2.1 that handshake messages are sent through the AP. This connection with the AP was already validated using one of the above handshakes.

To further harden implementations, we also recommend (but do not require) including an OCI element in the group key handshake.

4.3 Backwards Compatibility

Our extension must be backwards-compatible with devices that do not support operating channel validation. To this end, we add a new flag to the RSN Capabilities bit-field [11, §9.4.2.25.4]. In particular, we propose the Operating Channel Validation Capable (OCVC) flag. A client or AP sets this flag if it supports channel validation. When both stations support it, channel validation will be used.

Downgrade attacks are prevented because the RSN capabilities field is authenticated using the negotiated session key. As a result, an attacker is not able to modify this field in an attempt to prevent channel validation from being used. In other words, if both stations support channel validation, it will always be used.

4.4 Verifying Unprotected Channel Switches

A network may change the operating channel while clients are already connected to it. To inform clients about the channel switch, the AP broadcasts Channel Switch Announcements (CSAs). Recall from Section 3.3 that CSA elements in beacons and probe responses are not protected, meaning an attacker can forge them. To prevent an attacker from abusing CSA elements to obtain a multi-channel MitM, a naive idea would be to require MFP and only send and accept CSAs in action frames. However, although CSA elements would then always authenticated, there would still be no guarantee that the client received the CSA element and indeed switched channels. This may result in the client and AP being on different channels, which an adversary can abuse to obtain a MitM position. To prevent this, a client must always (securely) confirm the receipt of a CSA element. We will also use this confirmation procedure to securely verify unauthenticated channel switch announcements.

To securely confirm channel switches, we assume MFP is enabled, and require that stations verify all received CSA elements using a SA query procedure. More precisely, after switching to the new channel, the client must initiate a SA query with the AP. Similar to the inclusion of an OCI element inside handshake messages, both the SA query response and request must include an OCI that describes the channel used by the transmitter, with the operating class representing the widest bandwidth being supported. Note that SA query frames are authenticated if MFP is used, meaning the OCI element will also be authenticated. When receiving a SA query frame that must contain an OCI element, the receiver performs the same checks as when receiving a handshake message that must contain an OCI element (see Section 4.2). If any of the checks fail, the client is deauthenticated from the network. If the SA query procedure times out, the client must not switch back to the previous channel, as this may violate DFS regulations, Moreover, switching back may even enable short-term MitM attacks. Instead, if the SA query times out, the connection must be terminated. After sending or receiving the initial channel switch announcement, a station must pause transmission (Tx) and reception (Rx) of frames until the SA query has completed successfully. This is illustrated in Figure 2. Pausing transmission and reception prevents an adversary from obtaining a temporary MitM by forging CSAs, which would cause the client, but not the AP, to switch channels.

Note that a station may be in sleep mode when the AP is changing channels (recall Section 2.3). This means it will miss the channel switch announcement. We solve this by not mandating SA queries from clients in WNM-sleep mode. Instead, we require that stations include an OCI element in the WNM-sleep mode exit request and response frames. The WNM frames effectively replace the SA query when the client wakes up and discovers the AP on a new channel.

We do remark that a temporary partial MitM remains possible. In particular, an adversary can block CSAs from arriving at a client, causing the client to stay on the old channel. The adversary can then capture and store frames sent by the client. Before the AP disconnects the client due to a SA query timeout, the adversary sends a CSA to the client so it will switch channels and perform a valid SA query. Now the adversary can forward the previously captured frames to the AP. Note that the attack is only possible if the AP performs channel switches, the attacker can trigger a channel switch, and the attacker manages to jam all CSAs. This is non-trivial, especially because most APs broadcast multiple beacons with CSAs, which an attacker must all successfully block. We also remind the reader that our goal is not to make attacks impossible. Instead, if an essential security feature of 802.11 contains a deficiency, the goal of our defense is to make it harder to exploit said deficiency.

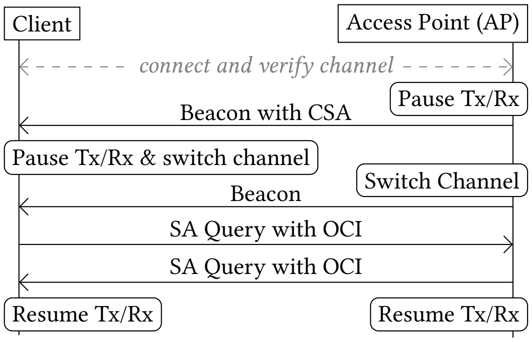

Figure 2: Channel switch with SA Query. The client waits for a beacon on the new channel before initiating the SA Query.

5 EVALUATION

In this section we evaluate the security and feasibility of our proposed extension, and discuss a proof-of-concept implementation.

5.1 Security Considerations

Our defense assumes that the client is within range of the AP. If this is not the case, the adversary can act as a repeater to obtain a MitM position [15]. Moreover, the attacker could clone a far-away network by forwarding frames over the internet [18, §5.3]. However, this is only possible if the adversary knows which network(s) the victim will connect to. In other words, with operating channel validation, large-scale attacks against many clients become significantly more difficult because the adversary would not know which network(s) to repeat.

The adversary can also try to obtain a partial MitM by manipulating the advertised frame reception capabilities of a client or AP. For example, the attacker can forge beacon frames (on the channel of the real AP) that advertise extra bitrates that are not supported by the AP. If the victim connects based on the forged beacons, it marks these extra bitrates as usable. The subsequent handshake will successfully complete, since supported bitrates, and frame reception capabilities in general, are not cryptographically verified. Moreover, both the AP and client are on the same channel, meaning channel validation will also be successful. When the client now uses the extra bitrates, the AP cannot receive these frames. However, the adversary does receive them, and can then manipulate and forward these frames. Apart from supported bitrates, other features that can be abused are support for LDPC encoded frames, support for short guard intervals, operating mode notifications, and so on. The disadvantage of this attack is that only frames sent using an unsupported frame reception feature can be intercepted. Moreover, the attack is only possible if there are frame reception capabilities that one device supports, but the other does not. As a result, this attack is less general then a multi-channel MitM.

We conclude that, although channel validation does not prevent all (partial) MitM attacks, it makes them significantly harder.

5.2 Future Work

Defending against other MitM attacks is interesting future work. First, the repeater attack of Section 5.1 might be detected by measuring the unique physical channel response between two stations. Previous work already showed that this property can be used to establish a secret key between two stations with only commodity hardware [14]. Second, the attack of Section 5.1 involving unsupported frame reception capabilities can be detected by verifying all advertised capabilities during the RSN handshake.

Proving there are no unknown edge cases where a MitM is possible by modeling physical protocol properties [16] is also useful.

5.3 Proof-of-Concept

We implemented our proposed extension by modifying an open source Wi-Fi client and AP, namely wpa_supplicant and hostapd.[1] To construct the OCI element, we request the configuration parameters of the wireless card using the nl80211 kernel interface. The Linux kernel must also be modified to assure a client does not dynamically change the maximum bandwidth. Otherwise, unauthenticated information such as the HT Operation element in e.g. the beacon can change the maximum bandwidth the client uses. Finally, we confirmed that clients that do not support channel verification can still connect with an AP that does support it, and vise versa.

6 RELATED WORK

Vanhoef and Piessens introduced the multi-channel MitM attack to break (WPA-)TKIP [18], and later also used it to attack other protocols [19–21]. The multi-channel MitM was also used by Van Goethem et al. to reveal information about encrypted web traffic [9]. Ohigashi and Morii relied on a repeater position as a MitM, and used this to attack (WPA-)TKIP [15]. However, they do not explain how to determine which Wi-Fi network to repeat. This means that in practice it is hard to obtain a repeater position. Konings et al. showed that channel switch announcements can be abused as a denial-of-service (DoS) attack [12]. Here CSAs are forged to trick a victim into switching to a different channel. Douglas et al. demonstrated that spoofing radar pulses causes an AP to switch channels [6]. There is no known countermeasure against this attack.

Francillon et al. replayed a car's raw physical signal to its corresponding smart key, with as goal to unlock and start the car [8]. Hu et al. describe replay attacks against ad hoc networks to disrupt routing protocols [10]. Note that our extension does not defend against replay attacks where the adversary acts as an repeater.

Finally, to the best of our knowledge, all other Wi-Fi based MitM attacks do not manipulate legimate encrypted traffic. Instead, they are MitM attacks against unprotected networks [13], against enterprise authentication mechanisms [4, 28], or are used to enable dictionary attacks against the 4-way handshake [1], and so on.

7 CONCLUSION

We proposed an extension to the 802.11 standard to prevent multi-channel MitM attacks, while assuring backwards compatibility with older devices. Although not all MitM variants are prevented by our extension, the remaining MitM attacks are less powerful and only possible under specific conditions. Therefore, when utilizing our extension, it becomes significantly harder to exploit existing (and future) vulnerabilities in protected Wi-Fi networks.

ACKNOWLEDGMENTS

This research is partially funded by the Research Fund KU Leuven. Mathy Vanhoef holds a Postdoctoral fellowship from the Research Foundation Flanders (FWO).

REFERENCES

[1] Dylan Ayrey. 2016. WPA2-HalfHandshake-Crack. (2016). Retrieved 28 February 2018 from https://github.com/dxa4481/WPA2-HalfHandshake-Crack

[2] John Bellardo and Stefan Savage. 2003. 802.11 denial-of-service attacks: real vulnerabilities and practical solutions. In *USENIX Security*.

[3] Nehru Bhandaru, Thomas Derham, Mathy Vanhoef, and Ido Ouzieli. 2018. Defense against multi-channel MITM attacks via Operating Channel Validation. (March 2018). Retrieved 7 May 2018 from https://mentor.ieee.org/802.11/

[4] Aldo Cassola, William Robertson, Engin Kirda, and Guevara Noubir. 2013. A Practical, Targeted, and Stealthy Attack Against WPA Enterprise Authentication. In *NDSS*.

[5] darkAudax. 2010. Tutorial: How to Crack WPA/WPA2. (2010). Retrieved 20 February 2018 from https://www.aircrack-ng.org/doku.php?id=cracking_wpa

[6] Brett Douglas, Greg Corsetto, and Douglas Chan. 2007. Greenfield Mode and DFS. (2007). Retrieved 27 February 2018 from https://mentor.ieee.org/802.11/

[7] Scott Fluhrer, Itsik Mantin, and Adi Shamir. 2001. Weaknesses in the key scheduling algorithm of RC4. In *SAC*.

[8] Aurélien Francillon, Boris Danev, and Srdjan Capkun. 2011. Relay attacks on passive keyless entry and start systems in modern cars. In *NDSS*.

[9] Tom Van Goethem, Mathy Vanhoef, Frank Piessens, and Wouter Joosen. 2016. Request and Conquer: Exposing Cross-Origin Resource Size.. In *USENIX Security*.

[10] Yih-Chun Hu, Adrian Perrig, and David B Johnson. 2006. Wormhole attacks in wireless networks. *IEEE journal on selected areas in communications* (2006).

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

[12] Bastian Könings, Florian Schaub, Frank Kargl, and Stefan Dietzel. 2009. Channel switch and quiet attack: New DoS attacks exploiting the 802.11 standard. In *LCN*.

[13] Mike Lynn and Robert Baird. 2002. Advanced 802.11b Attack. In *Black Hat USA*.

[14] Suhas Mathur, Wade Trappe, Narayan Mandayam, Chunxuan Ye, and Alex Reznik. 2008. Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In *MobiCom*.

[15] Toshihiro Ohigashi and Masakatu Morii. 2009. A practical message falsification attack on WPA. *Proc. JWIS* (2009).

[16] Patrick Schaller, Benedikt Schmidt, David Basin, and Srdjan Capkun. 2009. Modeling and verifying physical properties of security protocols for wireless networks. In *CSF*.

[17] Dorothy Stanley. 2017. TGm IEEE Nov 2017 Agenda. (2017). Retrieved 20 February 2018 from https://mentor.ieee.org/802.11/

[18] Mathy Vanhoef and Frank Piessens. 2014. Advanced Wi-Fi attacks using commodity hardware. In *ACSAC*.

[19] Mathy Vanhoef and Frank Piessens. 2016. Predicting, Decrypting, and Abusing WPA2/802.11 Group Keys. In *USENIX Security*.

[20] Mathy Vanhoef and Frank Piessens. 2017. Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2. In *CCS*.

[21] Mathy Vanhoef, Domien Schepers, and Frank Piessens. 2017. Discovering logical vulnerabilities in the Wi-Fi handshake using model-based testing. In *ASIA CCS*.

[22] Stefan Viehböck. 2011. Brute forcing Wi-Fi protected setup. (2011). Retrieved 20 April 2018 from packetstorm.foofus.com/papers/wireless/viehboeck_wps.pdf

[23] Wi-Fi Alliance. 2017. *Neighbor Awareness Networking Technical Spec. Version 2.0.*

[24] Wi-Fi Alliance. 2017. *Wi-Fi Agile Multiband Technical Spec. Version 1.1.*

[25] Wi-Fi Alliance. 2017. *Wi-Fi Certified Location: Indoor location of Wi-Fi.*

[26] Wi-Fi Alliance. 2018. Discover Wi-Fi: Security. (2018). Retrieved 25 February 2018 from https://www.wi-fi.org/discover-wi-fi/security

[27] Wi-Fi Alliance. 2018. Wi-Fi Alliance introduces security enhancements. Retrieved 19 Feburary 2018 from https://wi-fi.org/news-events/newsroom/wi-fi-alliance-introduces-security-enhancements. (2018).

[28] Joshua Wright. 2003. Weaknesses in LEAP challenge/response. In *DEF CON*.

[1] This code is available at https://github.com/vanhoefm/hostap-channel-validation