Exploiting Race Condition for Wi-Fi Denial of Service Attacks

Karim Lounis

Queen's University, Canada

SIN'20
Nov 4-7, 2020

Joint work with: Prof. Mohammad Zulkernine

Outline

Background

- WPA2-PSK security mechanism
- DoS attacks on WPA2-PSK
- Race condition-based vulnerability

Contributions

- DoS attack using MFP
- DoS attack using incorrect password
- DoS attack using WPA3-SAE
- Countermeasure

Wi-Fi Security

Wi-Fi (Wireless Fidelity) offers a number of security mechanisms:

1. **WEP (Wired Equivalent Privacy) [1999].** Provides authentication through challenge-response mechanism, encryrption thru RC4 and a 24-bit IV (Initialization Vector), and data integrity thru CRC-32.
2. **WPA (Wi-Fi Protected Access) [2003].** Based on a draft version of the 802.11i standard. Use TKIP: RC4 for encryption with longer IV (48-bit) and Michael algo for data integrity.
3. **WPA2 (Wi-Fi Protected Access 2) [2004].** Implementing 802.11i standard. Uses CCMP: AES-128 for encryption and AES-CBC-MAC for data integrity. (It also supports TKIP)
4. **WPA3 (Wi-Fi Protected Access 3) [2021?].** Augments WPA2 with additional security functions (e.g., enforce MFP 802.11w).

And other mechanisms, such as **WPS** and **OWE**.

WPA2-PSK Authentication Phases

In WPA2-PSK (Pre-shared Key), the authentication runs in three phases:

1. **Authentication Phase.** Exchange two management frames, one authentication request and one authentication response.

   At this point, the authenticating parties are considered authenticated w.r.t. IEEE 802.11 standard.

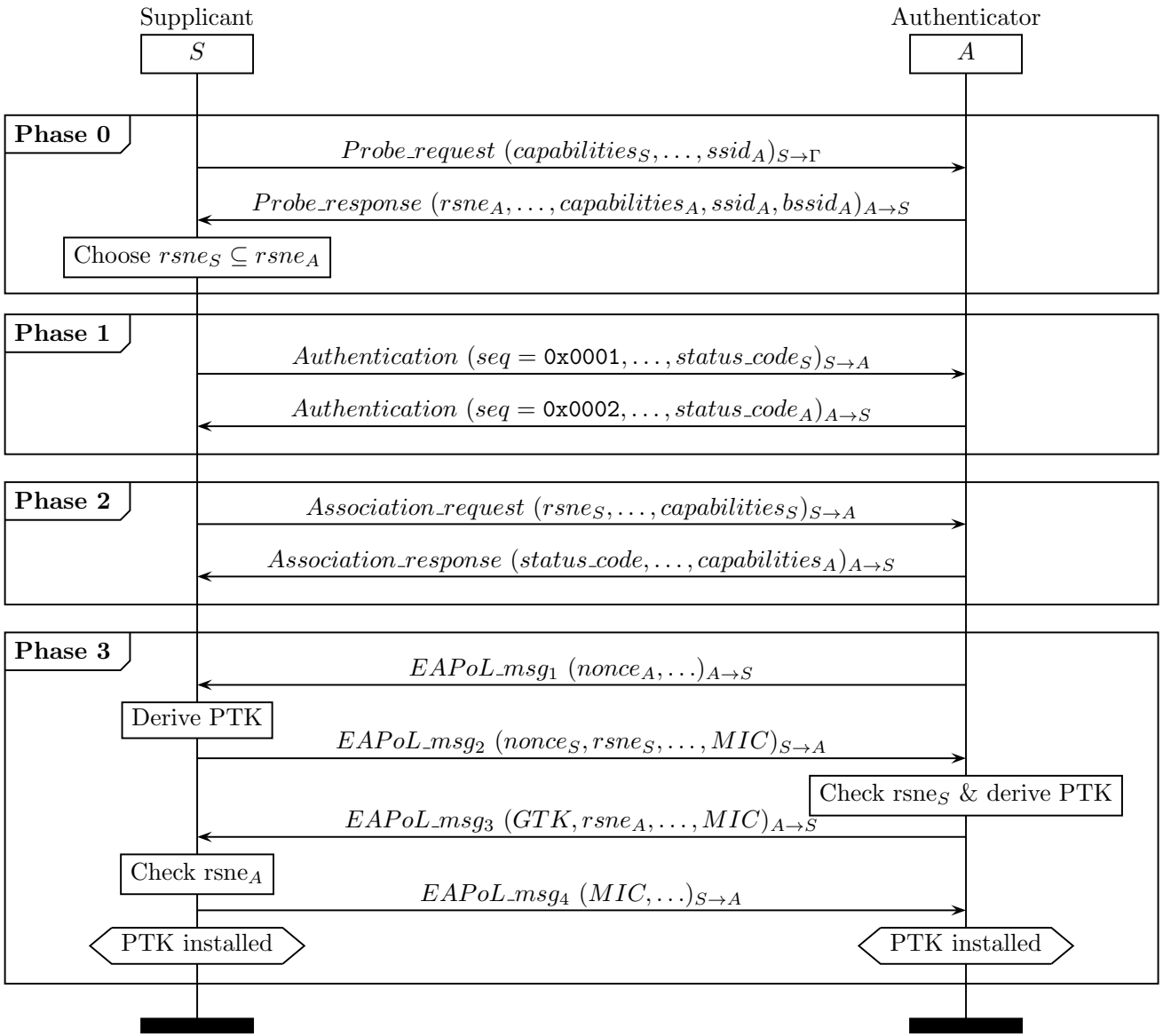
2. **Association Phase.** Exchange two management frames, one association request and one association response.

   At this point, the authenticating parties are considered associated w.r.t. IEEE 802.11 standard.

3. **4-Way-Handshake Phase.** Exchange four EAPoL messages.

   At this point, the authenticating parties will authenticate each other (w.r.t. 802.11i), and derive and install the PTK (Pairwise Transient Key) using the derived PMK.

WPA2-PSK Authentication Phases (MSC)



DoS attacks on WPA2-PSK

WPA2-PSK has been demonstrated to be vulnerable to different types of DoS attacks that can be categorized into three classes:

1. **Thru Management Frames.** By spoofing management frames an attacker usually impersonates the access point and generates attacks such as: deauthentication, deassociation, and sleep deprivation

   These attacks can be mitigated by enforcing MFP (Management Frame Protection), i.e., 802.11w.

2. **Thru Protocol Misuse.** An attacker may abuse MAC-layer protocols to access the radio, e.g., greedy behavior on CSMA/CA or RTS/CTS.

   There exist some detection approaches but not in the standard of IEEE 802.11.

3. **Thru Jamming.** The attacker generates random signals on the network operating radio channel to create interference.

   Techniques such as FHSS can be applied to limit the impact of jamming attacks.

Race-Condition Vulnerability

Source of the Vulnerability. The race-Condition vulnerability results from the conception that existing authentication protocols, in general, lack intelligence (there is no notion of smart authentication protocols).

This lack of intelligence comes from the fact that an authenticating party moves to the next step of the protocol based on the first message that it receives from the other authenticating party at a given stage of the protocol execution.

If `Alice` sends a message `m` to `Bob` multiple times (e.g., `n` instances) during an authentication then `Bob` will just consider processing the first instance of those messages and never goes back in the trace.

Race-Condition Vulnerability

Source of the Vulnerability. The race-Condition vulnerability results from the conception that existing authentication protocols, in general, lack intelligence (there is no notion of smart authentication protocols).

This lack of intelligence comes from the fact that an authenticating party moves to the next step of the protocol based on the first message that it receives from the other authenticating party at a given stage of the protocol execution.

If `Alice` sends a message `m` to `Bob` multiple times (e.g., `n` instances) during an authentication then `Bob` will just consider processing the first instance of those messages and never goes back in the trace.

The issue is that nothing can prove to `Bob` that the first instance of message `m` (that it will process) came from `Alice` and not from another source. It could be a spoofed message sent from `Charlie`.

Race-Condition Vulnerability

Why race condition. We refer to this vulnerability by race-condition, as the legitimate party (unawarely) and the attacker (maliciously) will concurrently run the protocol and the output of the protocol will be oriented and affected by the order of the reception of the messages.

Depending on the protocol's specifications, the authentication protocol may behave in a way that is beneficial to the attacker.

For example, one of the straightforward decision that is taken by the protocols is to abort the communication. This would constitute a potential flaw to generate DoS attacks.

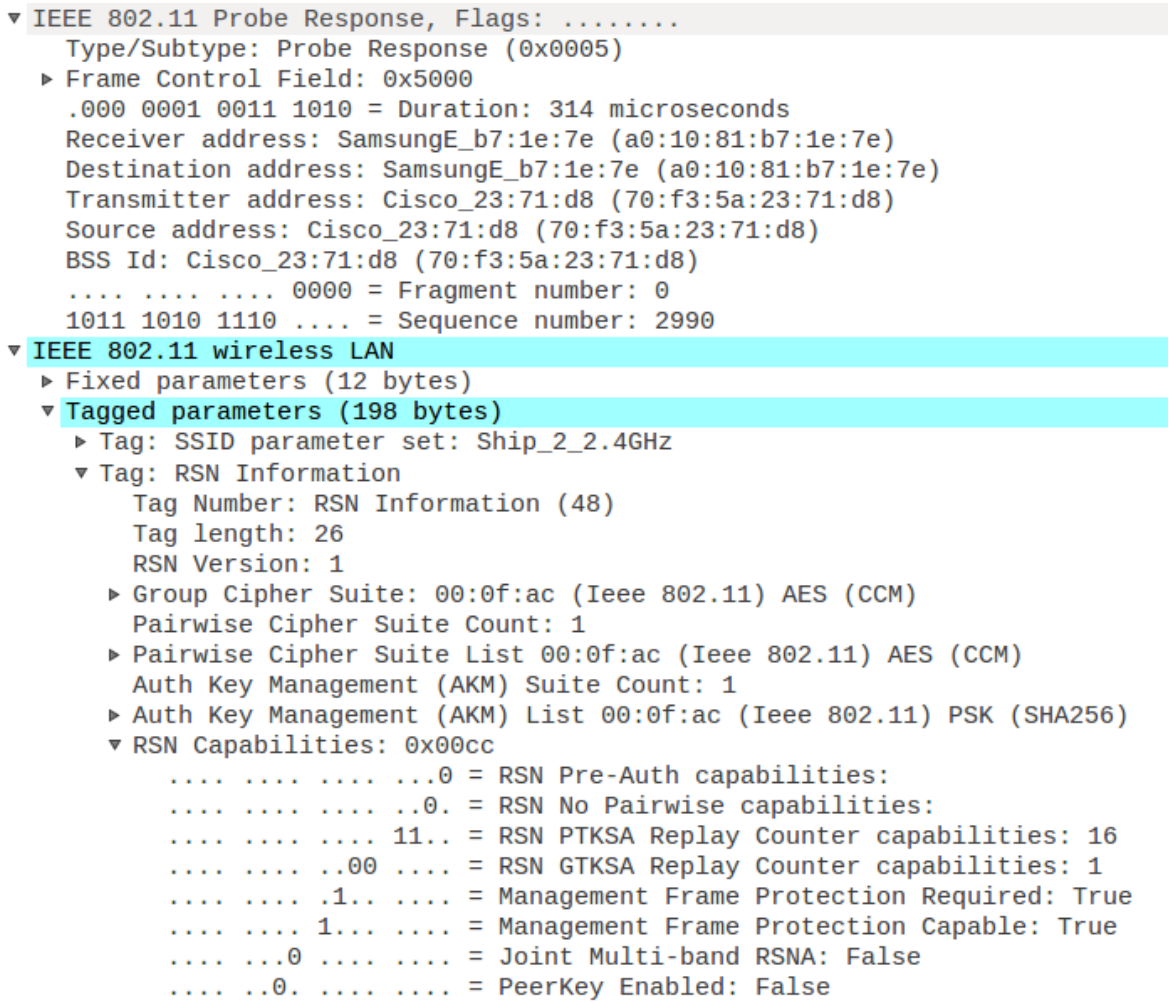
Experimental Testbed

To perform the attacks, we have used the following devices:

1. A laptop HP ProBook 6560b, running Linux Ubunt 16.04 LTS OS and hostapd-2.7.
2. Two Wi-Fi supplicants, a smartphone Samsung J7-2016 (Android 8.1.0) and a tablet Huawei MediaPad M5 lite (Android 8.0.0).
3. A Wi-Fi access point, Cisco WAP150, that is MFP-capable.
4. A Desktop, Dell precision T7500, running Linux Ubuntu 16.04 LTS OS along with a USB-dongle (ODROID Wi-Fi Module 4). It also runs airdump-ng and Wireshark for traffic monitoring and analysis.
5. A wireless router, Kisslink WR1410.

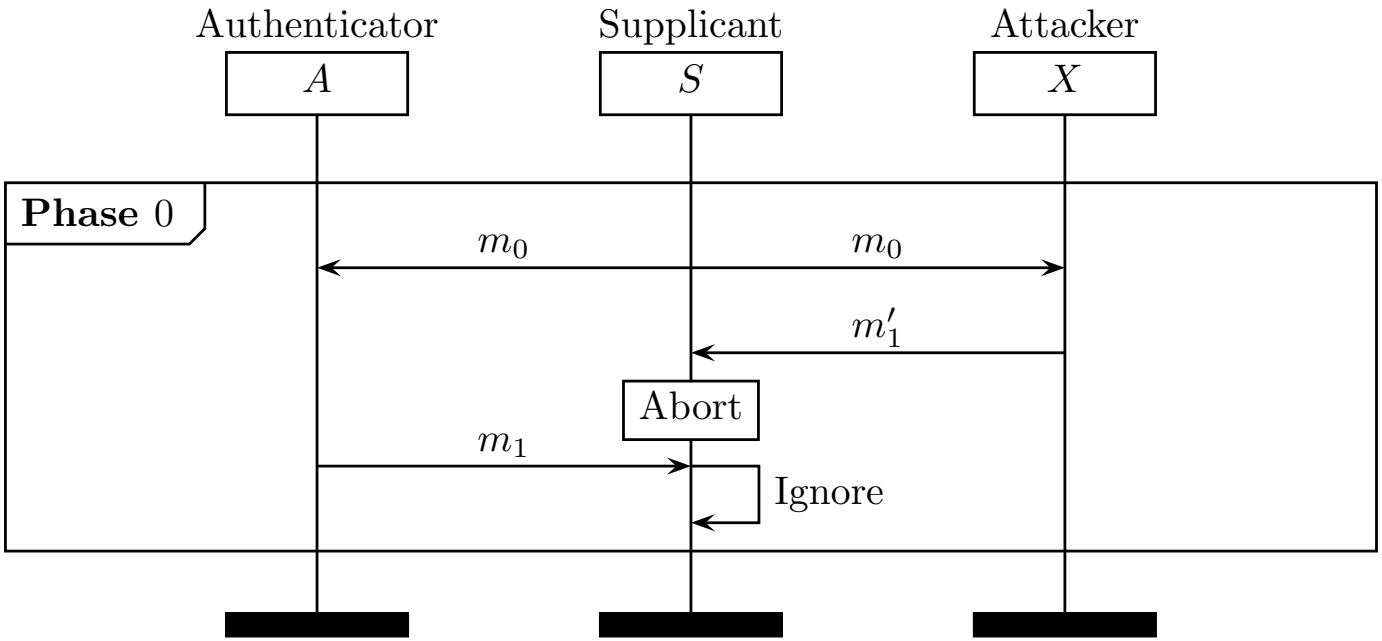
DoS attack using MFP (Management Frame Protection)

Observation. If a Wi-Fi supplicant is not MFP-capable, it will **abort** getting authenticated to the access point that requires MFP.



DoS attack using MFP (m_1' before m_1)

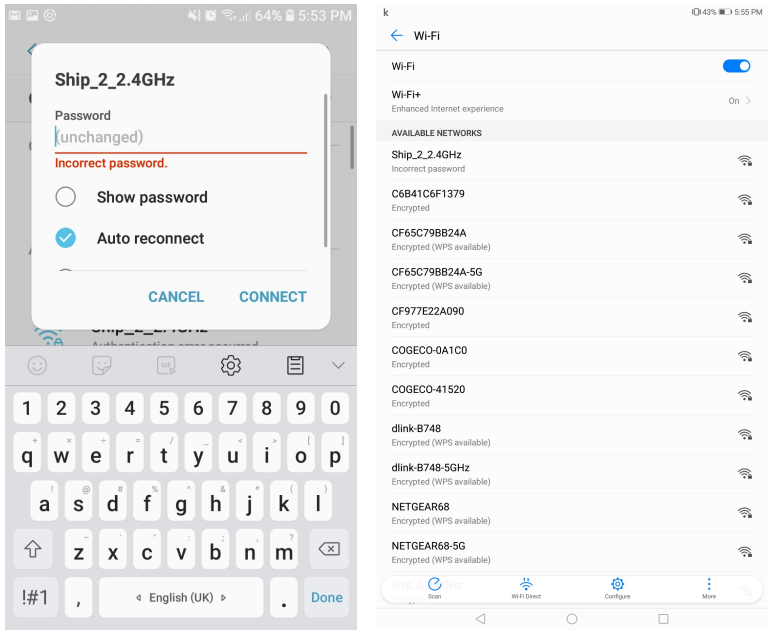
Attack Scenario. Installing an evil twin that operates MFP may result in confusing the supplicant and forcing the authentication to fail.



$$
\begin{aligned}
m_0 &= Probe\_request\ (capabilities_S, \ldots, ssid_A)_{S \to A},\\
m_1 &= Probe\_response\ (rsne_A, \ldots, capabilities_A, ssid_A, bssid_A)_{A \to S},\\
m_1' &= Probe\_response\ (rsne_X, \ldots, capabilities_A, ssid_A, bssid_A)_{A \to S}.
\end{aligned}
$$

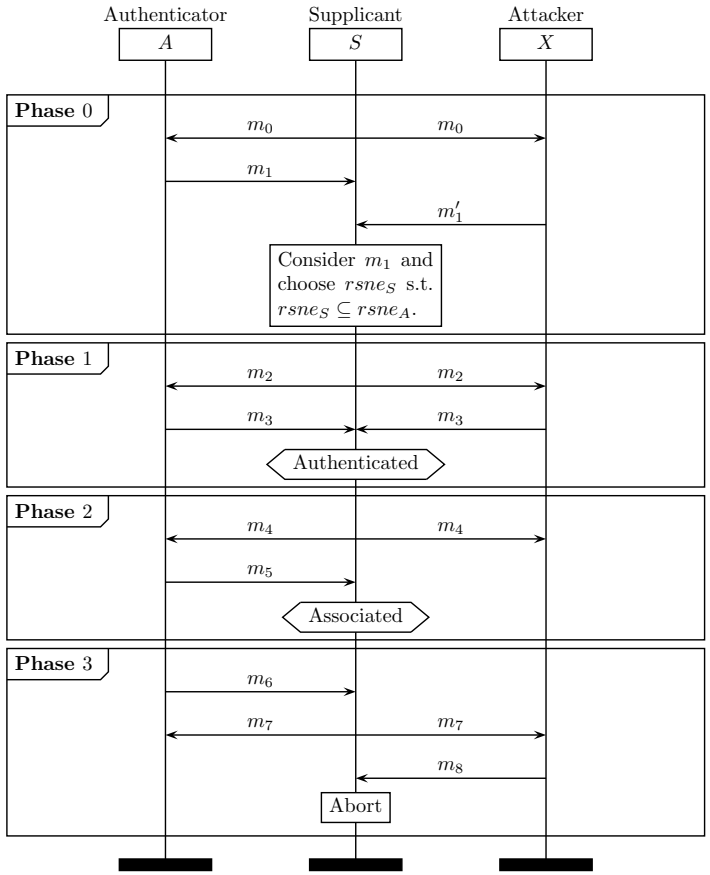
DoS attack using MFP (m_1' before m_1)

Attack Generation. We have started both access points, the attacker access point (Cisco WAP150), which is MPF-enabled, and the legitimate access point (the laptop running hostapd), which is not MFP-capable.



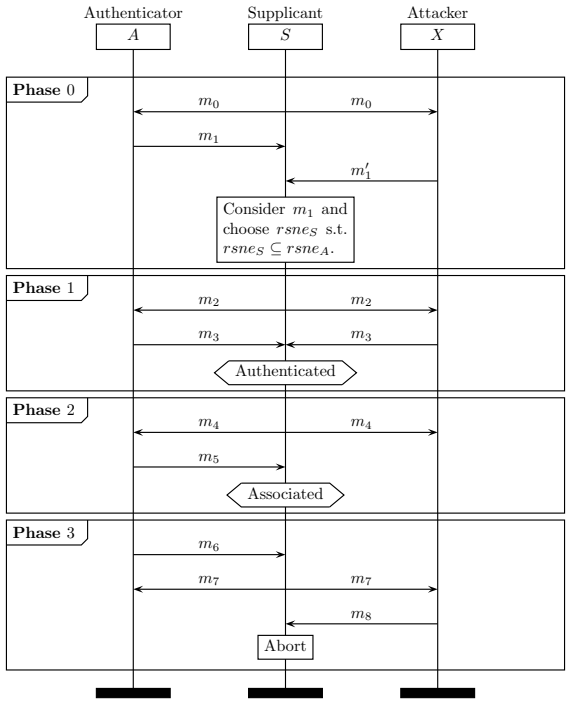
For **5 minutes** time frame, the supplicants (smartphone and tablet) could not get successfully connected (**DoS successful**).

DoS attack using MFP (m_1 before m'_1)



$m_0 = Probe\_request\ (capabilities_S, \ldots, ssid_A)_{S \to A}$,
$m_1 = Probe\_response\ (rsne_A, \ldots, capabilities_A, ssid_A, bssid_A)_{A \to S}$,
$m'_1 = Probe\_response\ (rsne_X, \ldots, capabilities_A, ssid_A, bssid_A)_{A \to S}$,
$m_2 = Authentiction\ (seq = \texttt{0x0001}, \ldots, status = \text{“}Successful\text{”})_{S \to A}$,
$m_3 = Authentiction\ (seq = \texttt{0x0002}, \ldots, status = \text{“}Successful\text{”})_{A \to S}$,
$m_4 = Association\_request\ (rsne_S, \ldots, capabilities_S)_{S \to A}$,
$m_5 = Association\_response\ (status = \texttt{0x0001}, \ldots, capabilities_A)_{A \to S}$,
$m_6 = EAPoL\_msg_1(nonce_A, \ldots)_{A \to S}$,
$m_7 = EAPoL\_msg_2(nonce_S, rsne_S, \ldots, MIC)_{A \to S}$,
$m_8 = Disassociation\ (status = \texttt{0x0007}, \ldots)_{A \to S}$.

DoS attack using MFP (m_1 before m_1')



$m_0 = Probe\_request\ (capabilities_S, \ldots, ssid_A)_{S \to A}$,
$m_1 = Probe\_response\ (rsne_A, \ldots, capabilities_A, ssid_A, bssid_A)_{A \to S}$,
$m_1' = Probe\_response\ (rsne_X, \ldots, capabilities_A, ssid_A, bssid_A)_{A \to S}$,
$m_2 = Authentiction\ (seq = \texttt{0x0001}, \ldots, status = \text{“}Successful\text{”})_{S \to A}$,
$m_3 = Authentiction\ (seq = \texttt{0x0002}, \ldots, status = \text{“}Successful\text{”})_{A \to S}$,
$m_4 = Association\_request\ (rsne_S, \ldots, capabilities_S)_{S \to A}$,
$m_5 = Association\_response\ (status = \texttt{0x0001}, \ldots, capabilities_A)_{A \to S}$,
$m_6 = EAPoL\_msg_1(nonce_A, \ldots)_{A \to S}$,
$m_7 = EAPoL\_msg_2(nonces_S, rsne_S, \ldots, MIC)_{A \to S}$,
$m_8 = Disassociation\ (status = \texttt{0x0007}, \ldots)_{A \to S}$.

The attacker will manage to disassociate the supplicant during the 4-way-handshake: Class 3 frame received from nonassociated STA.

DoS attack using incorrect password

Observation. During the 4-Way-handshake, the authenticator checks whether the supplicant has correctely derived the keys, and hence, hold the correct WPA2 password.

If it finds that the password is incorrect, it sends a disassociation frame with the respective status code.

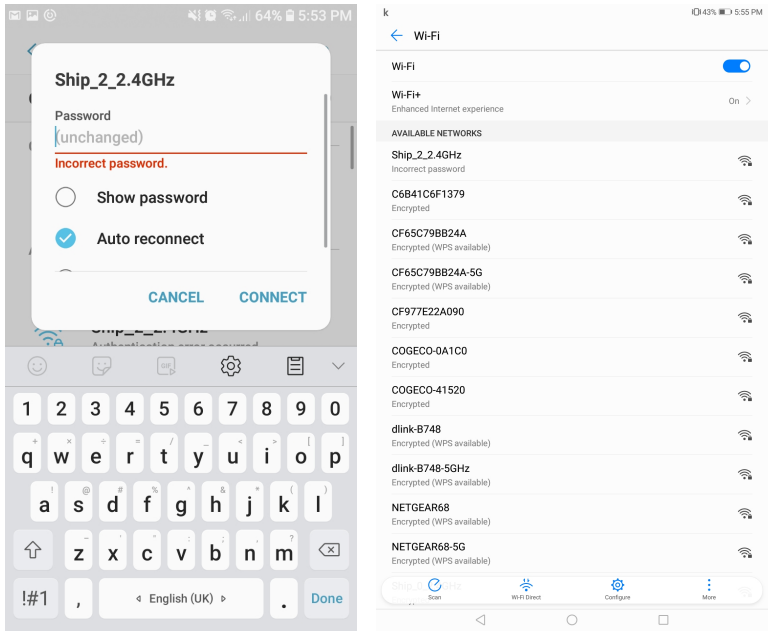
The supplicant aborts the authentication upon the reception of that frame.

Attack Scenario. If we configure a supplicant to automatically connect to a Wi-Fi access point and install an evil twin to that access point with the incorrect password, a race-condition will take place.

The supplicant may get misled by the evil twin and believe that the configured password is the wrong one.

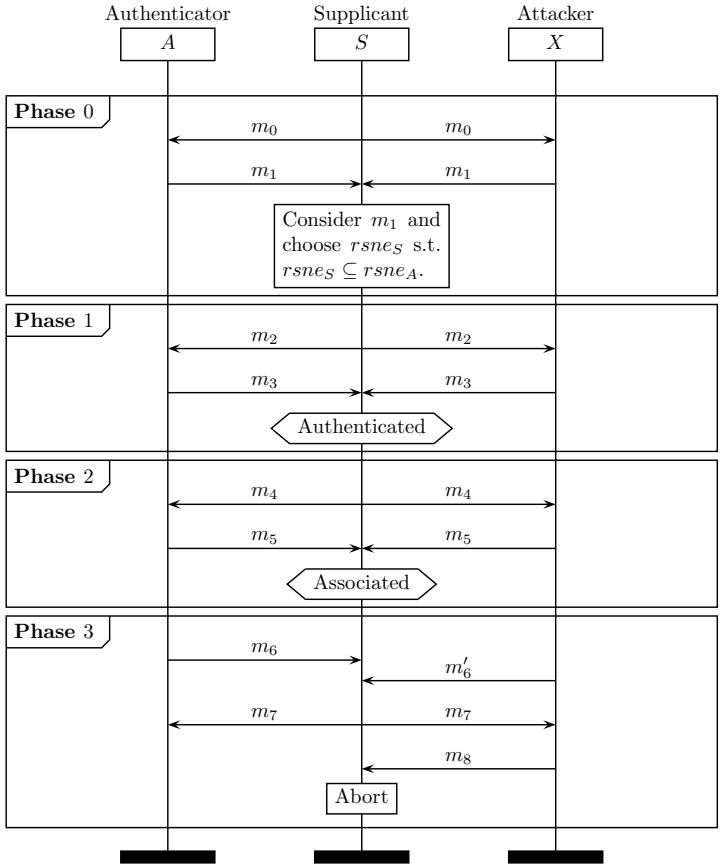
DoS attack using incorrect password

Attack Generation. We have started both access points, the attacker access point (laptop running hostapd), and the legitimate access point (Kisslink WR1410), and tried to connect the supplicant to the legitimate access point.



For **5 minutes** time frame, the supplicants (smartphone and tablet) could not get successfully connected (**DoS successful**).

DoS attack using incorrect password



$m_0 = Probe\_request\ (capabilities_S, \ldots, ssid_A)_{S \to A}$,
$m_1 = Probe\_response\ (rsne_A, \ldots, capabilities_A, ssid_A, bssid_A)_{A \to S}$,
$m_2 = Authentiction\ (seq = \texttt{0x0001}, \ldots, status = \text{“}Successful\text{”})_{S \to A}$,
$m_3 = Authentiction\ (seq = \texttt{0x0002}, \ldots, status = \text{“}Successful\text{”})_{A \to S}$,
$m_4 = Association\_request\ (rsne_S, \ldots, capabilities_S)_{S \to A}$,
$m_5 = Association\_response\ (status = \texttt{0x0001}, \ldots, capabilities_A)_{A \to S}$,
$m_6 = EAPoL\_msg_1(nonce_A, \ldots)_{A \to S}$,
$m'_6 = EAPoL\_msg_1(nonce_X, \ldots)_{A \to S}$,
$m_7 = EAPoL\_msg_2(nonce_S, rsne_S, \ldots, MIC)_{A \to S}$,
$m_8 = Disassociation\ (status = \texttt{0x000e}, \ldots)_{A \to S}$.

DoS attack using incorrect password

```
▼ IEEE 802.11 Disassociate, Flags: ........
    Type/Subtype: Disassociate (0x000a)
  ▶ Frame Control Field: 0xa000
    .000 0001 0011 1010 = Duration: 314 microseconds
    Receiver address: HuaweiTe_d4:c9:53 (90:17:c8:d4:c9:53)
    Destination address: HuaweiTe_d4:c9:53 (90:17:c8:d4:c9:53)
    Transmitter address: ComnectT_97:48:45 (4c:6e:6e:97:48:45)
    Source address: ComnectT_97:48:45 (4c:6e:6e:97:48:45)
    BSS Id: ComnectT_97:48:45 (4c:6e:6e:97:48:45)
    .... .... .... 0000 = Fragment number: 0
    0001 1111 1100 .... = Sequence number: 508
▼ IEEE 802.11 wireless LAN
  ▼ Fixed parameters (2 bytes)
      Reason code: Message integrity code (MIC) failure (0x000e)
```

Disassociation frame sent from the attacker's access point to the tablet Huawei M5 during the incorrect password attack.

The frame was intercepted by the desktop monitoring system (running airdump-ng) and analyzed using Wireshark.

DoS attack using WPA3

Observation. If a supplicant does not support the cipher-suites that the authenticator proposes (i.e., rsne field in probe response). Then, it will fail to connect to that authenticator (access point).

The supplicant aborts the authentication upon the reception of that frame.

Attack Scenario. If we configure a supplicant to automatically connect to a WPA2-PSK access point and install an evil twin to that access point but running WPA3-SAE, then a race-condition will take place.

The supplicant may get misled by the evil twin and believe that the access point for which it was configured has upgraded its security.

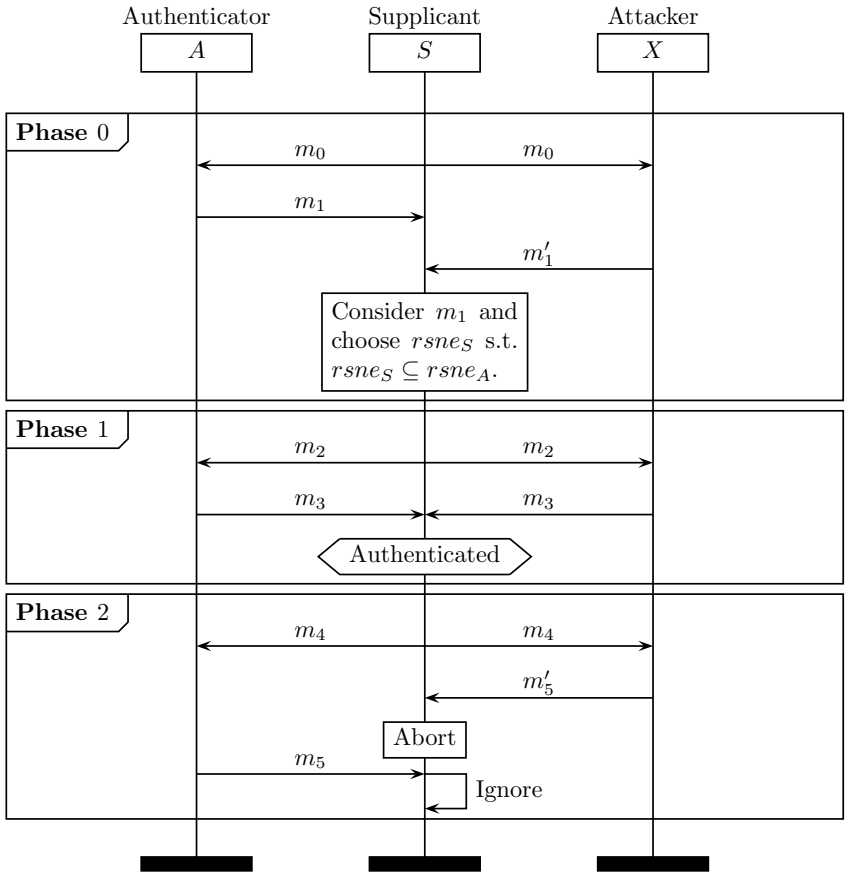
DoS attack using WPA3

Attack Generation. We have started both access points, the attacker access point (laptop running hostapd configured for WPA3-SAE), and the legitimate access point (Kisslink WR1410), and tried to connect the supplicant to the legitimate access point.

The supplicants displayed the network information in two different ways: WPA2-Enterprise (Smartphone) and Nothing (tablet)).

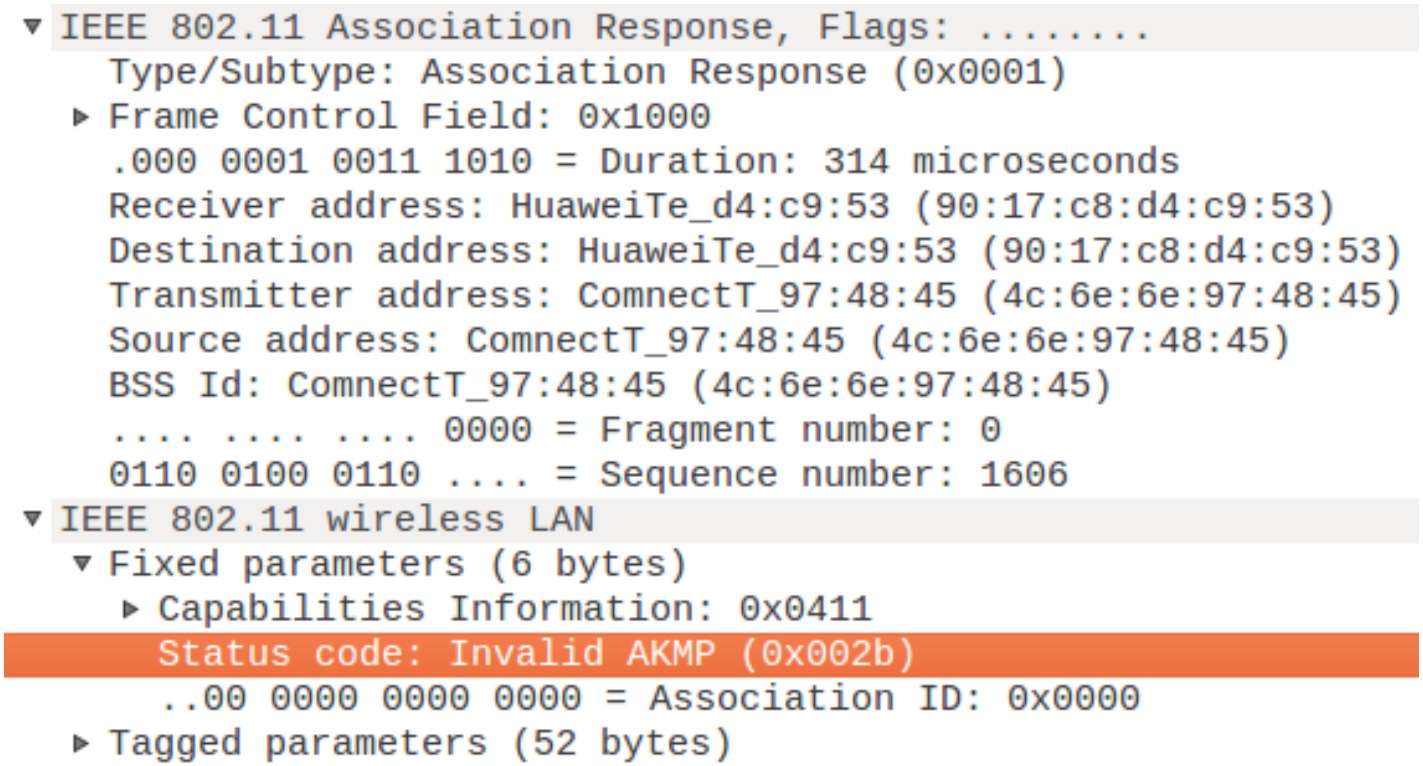
For **5 minutes** time frame, the supplicants (smartphone and tablet) could not get successfully connected (**DoS successful**).

DoS attack using WPA3



$m_0 = Probe\_request\ (capabilities_S, \ldots, ssid_A)_{S \to A}$,
$m_1 = Probe\_response\ (rsne_A, \ldots, capabilities_A, ssid_A, bssid_A)_{A \to S}$,
$m'_1 = Probe\_response\ (rsne_X, \ldots, capabilities_A, ssid_A, bssid_A)_{A \to S}$,
$m_2 = Authentiction\ (seq = \texttt{0x0001}, \ldots, status = \text{“}Successful\text{”})_{S \to A}$,
$m_3 = Authentiction\ (seq = \texttt{0x0002}, \ldots, status = \text{“}Successful\text{”})_{A \to S}$,
$m_4 = Association\_request\ (rsne_S, \ldots, capabilities_S)_{S \to A}$,
$m'_5 = Association\_response\ (status = \texttt{0x002b}, \ldots, capabilities_A)_{A \to S}$.
$m_5 = Association\_response\ (status = \texttt{0x0001}, \ldots, capabilities_A)_{A \to S}$.

DoS attack using WPA3



Association response frame sent from the attacker's access point to the tablet Huawei M5 during the WPA3-based attack.

The frame was intercepted by the desktop monitoring system (running airdump-ng) and analyzed using Wireshark.

Mitigating race condition-based attacks

To mitigate the previous attacks, we have proposed the following algorithm as a *first step toward an efficient solution*:

Algorithm 1 Authentication Protocol Stage Decision

```
1:  procedure MOVE_OR_ABORT
2:      Δt ← v                          ▷ Stage time frame duration v > 0
3:      i ← 0
4:      while (Δt > 0) do
5:          Receive(m)                  ▷ Receive message m
6:          B[i] ← m                    ▷ Buffer message m
7:          i ← i + 1
8:          Δt ← Δt − τ                 ▷ τ time to buffer a message
9:      for j ← 0, i − 1 do             ▷ Process buffered messages
10:         If (Φ(B[j]) == 1) then return True
11:     return False                    ▷ Abort authentication
```

The goal is to make future supplicants and authenticators **smarter**.

Mitigating race-condition attacks

Of course the current countermeasure has some disadvantageous, mainly, a delay that is caused to make a decision after the reception of each individual messages.

Algorithm 1 Authentication Protocol Stage Decision

```
1:  procedure MOVE_OR_ABORT
2:      Δt ← v                          ▷ Stage time frame duration v > 0
3:      i ← 0
4:      while (Δt > 0) do
5:          Receive(m)                  ▷ Receive message m
6:          B[i] ← m                    ▷ Buffer message m
7:          i ← i + 1
8:          Δt ← Δt − τ                 ▷ τ time to buffer a message
9:      for j ← 0, i − 1 do             ▷ Process buffered messages
10:         If (Φ(B[j]) == 1) then return True
11:     return False                    ▷ Abort authentication
```

However, those delays will only affect the response time during the authentication. **We believe that scarifying some milliseconds to guarantee a successful connection is worthwhile.** .

Conclusion

- We have introduced the race-condition vulnerability.

  Although this vulnerability has never been **explicitly** discussed in the literature as such, we do not claim it as novel.

- By exploiting the vulnerability, we have demonstrated the feasibility of three DoS attacks on WPA2-PSK.

  Thus far, there exist no countermeasure in the standard 802.11i that can mitigate the presented attacks.

- We have proposed a possible countermeasures to mitigate the attacks.

  Again, we do not claim that the proposed solution is 100% complete but rather a first step toward an efficient countermeasure, as we have not implemented it and evaluated it. That will be one of our future work.

- Thank You.