28. WPA

28. Wireless Local Networks: WPA2

28.1. Cheat sheet

- Layer: Link (2)
- Purpose: Communicate securely in a wireless local network
- Vulnerability: On-path attackers can learn the encryption keys from the handshake and decrypt messages (includes brute-forcing the password if they don't know it already)
- Defense: WPA2-Enterprise

28.2. Networking background: WiFi

Another implementation of the link layer is WiFi, which wirelessly connects machines in a LAN. Because it uses wireless connections over cellular networks, WiFi has some differences from wired Ethernet, but these are out of scope for this class. For the purposes of this class, WiFi behaves mostly like Ethernet, with the same packet format and similar protocols like ARP for address translation.

To join a WiFi network, your computer establishes a connection to the network's **AP (Access Point)**. Generally the AP is continuously broadcasting beacon packets saying "I am here" and announcing the name of the network, also called the **SSID (Service Set Identifier)**. When you choose to connect to a WiFi network (or if your computer is configured to automatically join a WiFi network), it will broadcast a request to join the network.

If the network is configured without a password, your computer immediately joins the network, and all data is transmitted without encryption. This means that anybody else on the same network can see your traffic and inject packets, like in ARP spoofing.

28.3. Protocol

WPA2-PSK (WiFi Protected Access: Pre-Shared Key) is a protocol that enables secure communications over a WiFi network by encrypting messages with cryptography.

In WPA2-PSK, a network has one password for all users (this is the WiFi password you ask your friends for). The access point derives a **PSK (Pre-Shared Key)** by applying a password-based key derivation function (PBKDF2-SHA1) on the SSID and the password. Recall from the cryptography unit that password-based key derivation functions are designed to be slower by a large constant factor to make brute-force attacks more difficult. Sanity check: Why might we choose to include the SSID as input to the key derivation function?[1]

When a computer (client) wants to connect to a network protected with WPA2-PSK, the user must first type in the WiFi password. Then, the client uses the same key derivation function to generate the PSK. Sanity check: Why can't we be done here and use the PSK to encrypt all further communications?[2]

[1] By including the SSID, two different networks with the same password will still have different PSKs.

[2] Because everyone on the network would use the same PSK, so others on the same network can still decrypt your traffic.

To give each user a unique encryption key, after both the client and the access point independently derive the PSK, they participate in a handshake to generate shared encryption keys.

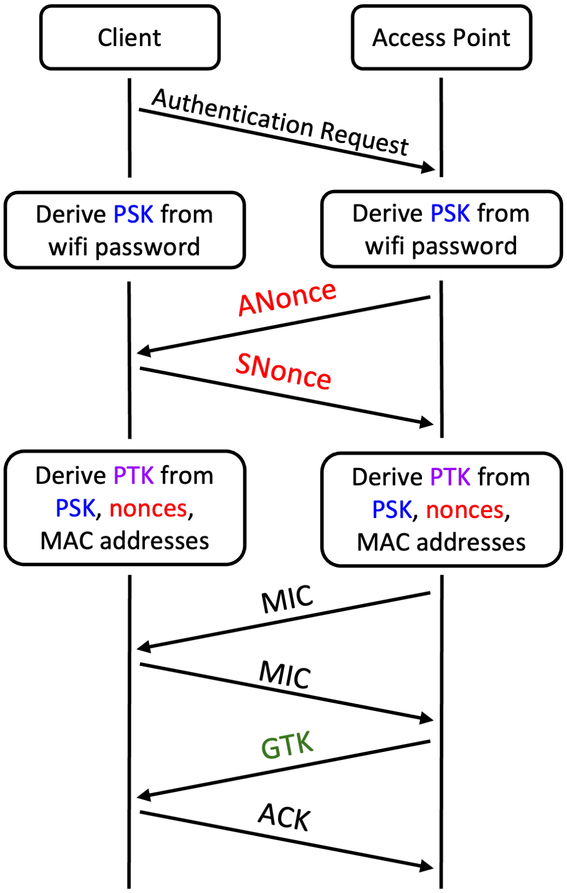
Client
Access Point
Authentication Request
Derive PSK from wifi password
Derive PSK from wifi password
ANonce
SNonce
Derive PTK from PSK, nonces, MAC addresses
Derive PTK from PSK, nonces, MAC addresses
MIC
MIC
GTK
ACK

Figure 1: Diagram of the WPA2 handshake

1. The client and the access point exchange random nonces, the ANonce and the SNonce. The nonces ensure that different keys will be generated during each handshake. The nonces are sent without any encryption.
2. The client and access point independently derive the **PTK (Pairwise Transport Keys)** as a function of the two nonces, the PSK, and the MAC addresses of both the access point and the client.
3. The client and the access point exchange MICs (recall that these are MACs from the crypto unit) to check that no one tampered with the nonces, and that both sides correctly derived the PTK.
4. The access point encrypts the **GTK (Group Temporal Key)** and sends it to the client.
5. The client sends an ACK (acknowledgement message) to indicate that it successfully received the GTK.

Once the handshake is complete, all further communication between the client and the access point is encrypted with the PTK.

The GTK is used for messages broadcast to the entire network (i.e. sent to the broadcast MAC address, `ff:ff:ff:ff:ff:ff`). The GTK is the same for everyone on the network, so everyone can encrypt/send and decrypt/receive broadcast messages.

In practice, the handshake is optimized into a 4-way handshake, requiring only 4 messages to be exchanged between the client and the access point.

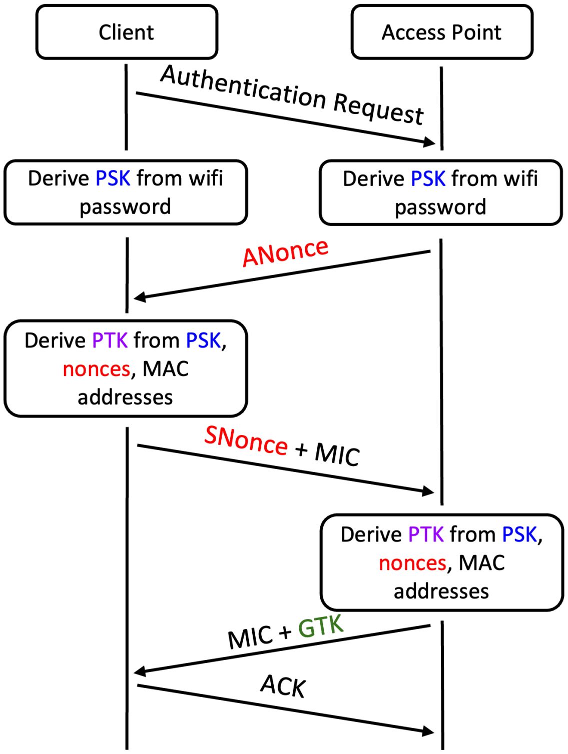
Client
Access Point
Authentication Request
Derive PSK from wifi password
Derive PSK from wifi password
ANonce
Derive PTK from PSK, nonces, MAC addresses
SNonce + MIC
Derive PTK from PSK, nonces, MAC addresses
MIC + GTK
ACK

Figure 2: Diagram of the optimized WPA2 handshake used in practice

1. The access point sends the ANonce, as before.
2. Once the client receives the ANonce, it has all the information needed to derive the PTK, so it derives the PTK first. Then it sends the SNonce and the MIC to the access point.
3. Once the access point receives the SNonce, it can derive the PTK as well. Then it sends the encrypted GTK and the MIC to the client.
4. The client sends an ACK to indicate that it successfully received the GTK, as before.

28.4. Attacks

In the WPA2 handshake, everything except the GTK is sent unencrypted. Recall that the PTK is derived with the two nonces, the PSK, and the MAC addresses of both the access point and the client. This means that an on-path attacker who eavesdrops on the entire handshake can learn the nonces and the MAC addresses. If the attacker is part of the WiFi network (i.e. they know the WiFi password and generated the PSK), then they know everything necessary to derive the PTK. This attacker can decrypt all messages and eavesdrop on communications, and encrypt and inject messages.

Even if the attacker isn't on the WiFi network (doesn't know the WiFi password and cannot generate the PSK), they can try to brute-force the WiFi password. For each guessed password, the attacker derives the PSK from that password, uses the PSK (and the other unencrypted information from the handshake) to derive the PTK, and checks if that PTK is consistent with the MICs. If the WiFi password is low-entropy, an attacker with enough compute power can brute-force the password and learn the PTK.

28.5. Defenses: WPA2-Enterprise

The main problem leading to the attacks in the previous section is that every user on the network uses the same secrecy (the WiFi password) to derive private keys. To solve this, each user needs an different, unique source of secrecy. This modified protocol is called **WPA2-Enterprise**. AirBears2 is an example of WPA2-Enterprise that you might be familiar with.

Instead of using one WiFi password for all users, WPA2-Enterprise gives authorized users a unique username and password. In WPA2-Enterprise, before the handshake occurs, the client connects to a secure authentication server and proves its identity to that server by providing a username and password. (The connection to the authentication server is secured with TLS, which is covered in a later section.) If the username and password are correct, the authentication server presents both the client and the access point with a random **PMK (Pairwise Master Key)** to use instead of the PSK. The handshake proceeds as in the previous section, but it uses the PMK (unique for each user) in place of the PSK (same for all users) to derive the PTK.

WPA-2 Enterprise defends against the attacks from the previous section, because the PMK is created randomly by a third-party authentication server and sent over encrypted channels to both the AP and the client. However, note that WPA2-Enterprise is still vulnerable against another authenticated user who executes an ARP or DHCP attack to become a man-in-the-middle.