KRACK attack (802.11i four-way handshake attack)

Vadim Davydov
vadim.davydov@epfl.ch

I. Introduction

In 2017, the group of researchers from KU Leuven university [3] found serious vulnerabilities in the WPA2 protocol. This protocol is an updated wireless device certification program which was created to ensure confidentiality, integrity and accessibility of the information in modern Wi-Fi networks. These technologies rely on the 4-way handshake defined in the 802.11i amendment of 802.11 [3]. An attacker, who is in the victim's area, can use Key Reinstallation Attack to read information that was previously considered to be encrypted. In this short report, the attack mechanism is explained and different variations of the attack are discussed.

II. 802.11i protocol description

This section contains the details of the 802.11i protocol and is divided into four parts.

A. Used keys

In this subsection the used keys in the system are presented. Normally, there are three main keys:

- PMK (Pairwise Master Key): derived from a pre-shared password in a personal network;
- PTK (Pairwise Transient Key, 384 bits): derived from PMK, ANonce (random nonce, which is generated on authenticator side), SNonce (random nonce, which is generated on supplicant (client) side) and MAC addresses of both the supplicant and authenticator;
- GTK (Group Temporal Key): generated by the Authentificator.

In turn, after deriving the PTK, the key itself is divided into:

- KCK (Key Confirmation Key, bits 0–127 of the PTK): to protect the authenticity of the sending frame;
- KEK (Key Encryption Key, bits 128–255 of the PTK): to protect the Key Data field of the frame (see subsection B);
- TK (Temporal Key, bits 256–383 of the PTK): to protect normal data frames with a data-confidentiality protocol.

B. EAPOL frames

Every message in the 4-way handshake is defined using EAPOL frames (cf. Figure 1).

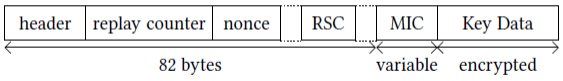

Fig. 1: The structure of the EAPOL frame.

- Header: contains the information about which message in the handshake a particular EAPOL frame represents;
- Replay Counter: is used to detect replayed frames;
- Nonce: this field is used to transport random nonces to derive a fresh session key;
- RSC (Receive Sequence Counter): contains the starting packet number of the group key;
- MIC (Message Integrity Check): the value which is stored in this field and KEK are used for protecting the authenticity of the frame;
- Key Data: has the group key itself, which is encrypted using KEK.

C. Confidentiality and integrity protocols

To encrypt the data frame, confidentiality and integrity protocols are used. Depending on implementation, there are three possible variants:

- TKIP (Temporal Key Integrity Protocol)
  Encryption algorithm: RC4
  Message authencity: Michael algorithm
- CCMP (Counter Mode Cipher Block Chaining Message Authentication Code Protocol)
  Encryption algorithm: AES-CCM (Counter Mode with CBC-MAC)
  Message authencity: AEAD algorithm (Authenticated Encryption with Associated Data algorithm)
- GCMP (Galios Counter Mode Protocol)
  Encryption algorithm: AES-GCM (Galois/Counter)
  Message authencity: AEAD algorithm

To denote that a frame is encrypted and authenticated using a data-confidentiality protocol, the following notation $Enc_k^n\{...\}$ is used, where n is the nonce and k is the key. The key k in case of unicast traffic is PTK and GTK in case of a group traffic.

D. 4-way handshake

Figure 2 illustrates the whole protocol which consists of three main steps: assosiation, four-way handshake and group key handshake. $MsgN(r, Nonce; GTK)$ represents message N of the 4-way handshake, having a replay counter r, and with the given nonce (if present). GTK is stored in the key data field and is encrypted using the KEK. The two notations $Data(...)$ and $GroupData(...)$ are used to represent an ordinary unicast or group addressed data frame, respectively.

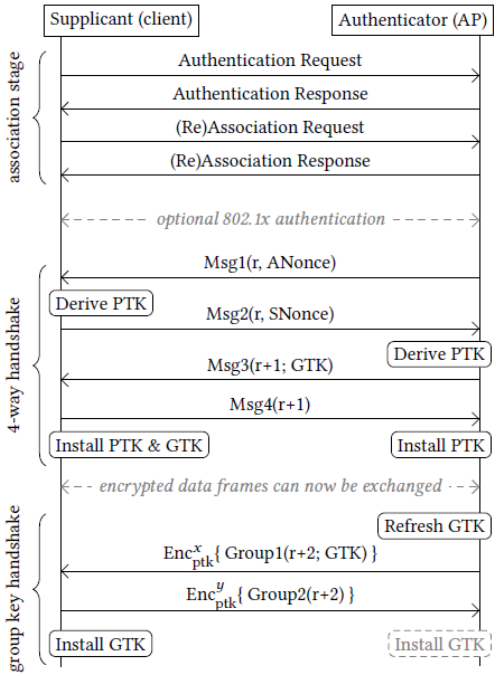

Fig. 2: The communication protocol between Supplicant and Authenticator

- **Association.**
  At this stage of the protocol, no actual authentication takes place and Open System authentication is used which allows any client to autenticate. It is a system which provides identification of the client by using the wireless adapter's MAC address. After authentication, the client is associated with the network, which is done by sending an association request to the AP.

- **4-way handshake.**
  There are 4 messages to be sent on this stage.
  1) Msg1: The authenticator uses this message only to send ANonce to the supplicant.

2) Msg2: After successful delivery of the first message, the supplicant generates its own value of SNonce and sends it in message 2 to the authenticator.
3) Msg3: This message is used to verify to the supplicant that the authenticator knows the PMK and is thus a trusted party. Secondly, it gives the information that the authenticator is ready to install and start using the data encryption keys. Finally, the messge is used for sending the group key to the supplicant.
4) Msg4: This message is used to verify to the authenticator that the keys are about to be installed.

- **Group key handshake.**
  The authenticator initiates the handshake by sending a group message 1 to all clients. The supplicant acknowledges the receipt of the new group key by replying with group message 2.

III. KEY RE-INSTALLATION ATTACK

Before talking about the attacks, let's consider three properties of the standard 802.11i [2]:

1) The AP retransmits message 1 or 3 if it did not receive a reply.
2) The client should install the PTK after processing and replying to message 3.
3) On reception of message 4, the Authenticator verifies that the Key Replay Counter field value is that one that has been used at the current 4-way handshake.

The main idea of the attack is to force the Supplicant to reinstall the key. The Client normally installs the key after receiving message 3. The AP retransmits message 3 if it did not receive a reply and as such key reinstallation can be achieved by triggering retransmissions of message 3 by preventing message 4 from arriving to the authenticator. To mount the attack, the first step is establishing a position between the supplicant and authenticator. It is not easy because the generated keys depend on the MAC addresses of the Supplicant and the Authenticator. If the adversary uses a rogue access point with a different MAC addresses, the handshake will fail. In this case, the channel-based MitM attack is used. The main idea of this attack is cloning the access point and forcing the client to reconnect to the rogue AP by continuous jamming the real channel.

The main point why this attack works is that the nonce, being used by the data-confidentiality protocol, can be reused. The impact of nonce reuse caused by the attack depends on the data-confidentiality protocol. All three protocols use a stream cipher to encrypt frames. Therefore, reuse of a nonce always implies reuse of the keystream. This can be used to decrypt, replay or even forge packets.

There are several different attacks to the protocol depending on the implementation. In this report we will look at two examples of them. First, we will look at one of the variations of 4-way handshake attacks. We will focus on the situation when the victim accepts a plaintext message 3 retransmission, which is presented in figure 3.

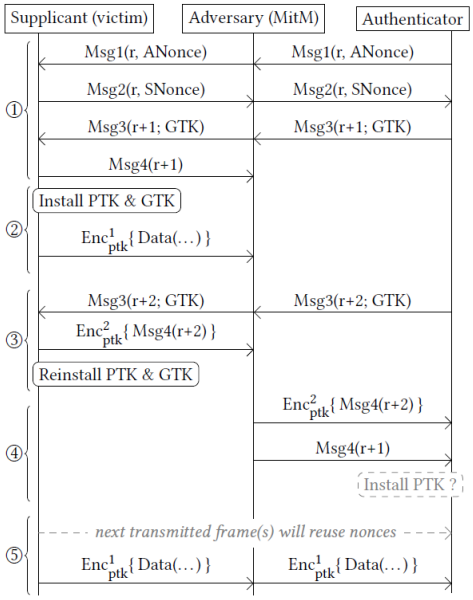

Fig. 3: Attacking 4-way handshake

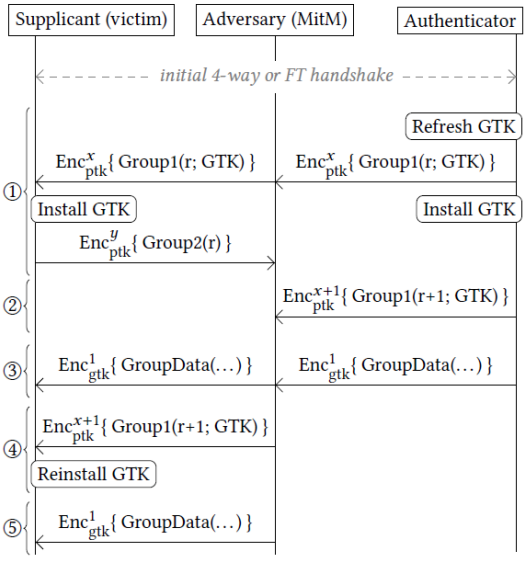

Fig. 4: Attacking group handshake

The attack works as follows. The adversary uses a channel-based MitM attack and blocks message 4 from arriving to the authenticator. Due to this, the Authenticator, who didn't receive the response, retransmits message 3. Consequently, PTK and GTK are reinstalled. As a result, it resets the nonce and replay counter used by the data-confidentiality protocol and we have

two encrypted messages with the same nonces. Due to the properties of used data-confidentiality protocols, the data can be decrypted.

The goal of the stage 4 in the figure 4 is to complete the handshake at the authenticator side. The victim already installed the PTK, and message 4 is encrypted. The problem is that the 802.11 standard dictates the retransmitted message 4 to be sent in plaintext in the initial 4-way handshake, but nearly all clients send it using encryption. Obviously, if the authenticator did not yet install the PTK, it rejects encrypted message 4. But, as it is written above, the Key Replay Counter field value can be one that was used on this 4-way handshake before. That means the old unencrypted message can be sent and will be accepted by the Authenticator.

The group key handshake attack is shown in figure 4. An important notice is that networks periodically refresh the group key to assure that only recently authorized clients posses this key. The main goal of the attack, as in the previous one, is to force the client to reinstall the key which will lead to reusing the nonce. Below the variation of the attack is presented.

As shown in figure 4, when the client receives the group message 1, he installs the new GTK and sends back the group message 2. The adversary blocks this message from arriving at the AP. Hence, the Authenticator will retransmit a new group message 1. Now, the adversary waits until data frame is transmitted and forwards it to the Supplicant. Further, the adversary forwards the retransmitted group message 1 from stage 2 to the victim. As a result, the Supplicant will reinstall the GTK and will reinitialize a replay counter. This trick allows to replay the data frame and the client accepts this frame because its replay counter was reinitialized.

IV. Countermeasures

As the main problem in the system is that protocol reuses nonces, the logical countermeasure is to check if the system uses already-in-use keys and not to reset replay counters [3]. In this case, if the adversary tries to send the ”old” message, which was intercepted, the system will refuse it. For example, speaking about attacking 4-way handshake, when the client receives a retransmitted message 3, he should not reinstall the key. In this case, there will be no reusing nonces and the attacker can not break a confidentiality protocol. Another point is that the data-confidentiality protocol should provide some protection against nonce reuse.

V. Conclusion

In this report key-reinstallation attacks were described [3]. It is important to notice that there are several different implementations of the 802.11i protocol depending on what is the system, and the number of attacks is not limited by this report. One more important remark is that the attacks do not always work. For example, Windows and iOS do not accept retransmissions of the message 3 and are not vulnerable to the 4-way handshake attack. However, these systems are vulnerable to the group-key handshake attacks.

References

[1] W.-F. Alliance. Wi-fi protected access: Strong, standards-based, interoperable security for today’s wi-fi networks. *White paper, University of Cape Town*, pages 492–495, 2003.

[2] I. . W. Group et al. Ieee standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements–part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments. *IEEE Std*, 802(11), 2010.

[3] M. Vanhoef and F. Piessens. Key reinstallation attacks: Forcing nonce reuse in wpa2. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1313–1328. ACM, 2017.