Design and Implementation of IEEE 802.11i in WIRE1x

Yuan-Yao Shih, Ming-Hung Yang, Shih-Chieh Liao, Chien-Chia Chen, Jui-Yi Chen, Yi-Ken Ho, and Jyh-Cheng Chen
Wireless Internet Research and Engineering Laboratory
National Tsing Hua University
Hsinchu, Taiwan

November 15, 2007

Abstract

This document presents the design and implementation of IEEE 802.11i in WIRE1x. The implementation is integrated into WIRE1x version 2.2, and later versions. WIRE1x is an open-source implementation of IEEE 802.1x client (supplicant) developed by the Wireless Internet Research & Engineering (WIRE) Laboratory, National Tsing Hua University. IEEE 802.11i specifies security mechanisms for wireless LANs. The 802.11i uses 802.1X for authentication and Robust Security Network (RSN) for keeping track of associations. It also includes two algorithms, Temporal Key Integrity Protocol (TKIP) and Counter mode with CBC-MAC Protocol (CCMP), to enhance the encryption.

1. Introduction

The Wireless Local Area Network (WLAN) has become more popular and commonly used in recent years. The traditional wired network that uses physical lines provides poor mobility. In addition, the WLAN provides great mobility, which is the best advantage.

The transmission medium of the WLAN is the air. Data are transmitted through radio, which means everyone who is within the range of the radio can intercept the data that only you should receive. In order to secure the WLAN data transmission, IEEE 802.11 standards define Wired Equivalent Privacy (WEP), which intends to provide the confidentiality as the wired network does.

Unfortunately, WEP connection can be cracked with software within minutes because of several serious weaknesses. The relatively short size of key and the possibility of initialization vector (IV) collisions limit the security in WEP; hence, WPA is proposed in response of these serious weaknesses in WEP.

WPA implements the Draft 3 of IEEE 802.11i standard, and was intended to replace WEP while IEEE 802.11i was still in progress. The WPA is designed to use either with an IEEE

802.1x authentication server, which distributes various keys to each user, or a less secure mode called “pre-shared key” (PSK) mode. In WPA, data are encrypted using the RC4 stream cipher, with a 128-bit key and a 48-bit IV. Moreover, the WPA uses Temporal Key Integrity Protocol (TKIP), which dynamically changes the keys, and to combine with the double sized IV (the WEP has only 24-bit IV), defeats the well-known key recovery attacks on WEP. A more secure Message Integrity Code (MIC) is used in WPA. The MIC used in WPA includes a frame counter that prevents replay attacks. Increasing the size of keys and IVs, reducing the number of packets sent with related keys, and adding a secure message verification system (MIC), WPA makes it much more difficult to crack a wireless network.

IEEE 802.11i is an amendment to IEEE 802.11 standard specifying security mechanisms. IEEE 802.11i is a superset of WPA specification. Thus, the Wi-Fi Alliance refers to their approved, interoperable implementation of IEEE 802.11i as WPA2. It contains IEEE 802.1x and four-way handshake for key-exchange, Robust Security Network (RSN) for keeping track of associations and AES based Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP) to provide confidentiality, integrity and origin authentication. In order to improve the strength of data encryption, IEEE 802.11i makes use of the Advanced Encryption Standard (AES) block cipher, while WEP and WPA use the RC4 stream cipher. IEEE 802.11i is definitely the most secure solution to 802.11 wireless networks for now. As a result, it is urgent for WIRE1x to support this new standard.

The rest of this report is organized as follows. The next chapter gives an overview of security standards for IEEE 802.11 wireless networks. Then, chapter 3 provides a roughly introduction of WIRE1x, our enhancement plan and the method to implement. System descriptions will be given in chapter 4 and we will conclude the report in the last chapter.

2. Standards to Build Robust Security Wireless Networks

2.1 Wireless Network Security Problems

Since wireless networks broadcast data by radio, they are easy to be eavesdropped. Besides, there are many kinds of threats against Local Area Network (LAN) security, like Denial of Service (DoS), Masquerading, Message Modification, etc. As a result, we need a combination of security features to be built into the wireless networking to ensure the security problems for Wireless LAN can be fully prevented.

The followings are the most common objectives for the security features built into wireless networking:

1. **Confidentiality:** To ensure unauthorized clients cannot read any data from the network.

2. **Integrity:** Make sure all the changes, either intentional or unintentional, of data during the transmission can be detected.

3. **Availability:** To ensure that all devices or clients can access the network and get the resources whenever they need.

4. **Access Control:** Make it possible to restrict some or all rights of devices or clients to access the network or specific resources in a network.

2.2 WEP

Before IEEE 802.11i and WPA, IEEE 802.11 wireless networks used Wired Equivalent Privacy (WEP) [1] to secure itself. WEP is a part of IEEE 802.11 standard and ratified in 1999.

For confidentiality, WEP uses the stream cipher RC4. The CRC-32 (Cyclic Redundancy Check) checksum is used for integrity. WEP Key size can be 40, 104, or 232 bits. In fact, the real WEP size is 64, 128, or 256 bits while some fields are used for the initialization vector (IV). The standard for WEP only supports 40-bit WEP key at first, but later, many vendors offer non-standard extension to the size of WEP key (up to 232 bits) to try to improve the protection strength of WEP.

There are two methods of authentication can be used with WEP: open system and shared key. The following is the brief introduction to these two methods:

In an open system authentication, a supplicant (STA) can associate itself with the Access Point (AP) and then attempt to authenticate without using WEP keys. Then after the authentication and association, WEP key can be used for encrypting the data, but the supplicant and the AP should have the right keys.

In a shared key system, WEP will be used for authentication. The whole process of the authentication is a four-way challenge-response handshake indicated in Figure 1:

1. The supplicant sends an authentication request to the AP

2. AP sends back a clear-text challenge and requests the supplicant to encrypt packet using the WEP key.

3. The supplicant sends the encrypted packet with the challenge text back to the AP

4. The AP decrypts the packet and checks the correctness of the challenge text, and then the AP will send back the result to the supplicant.

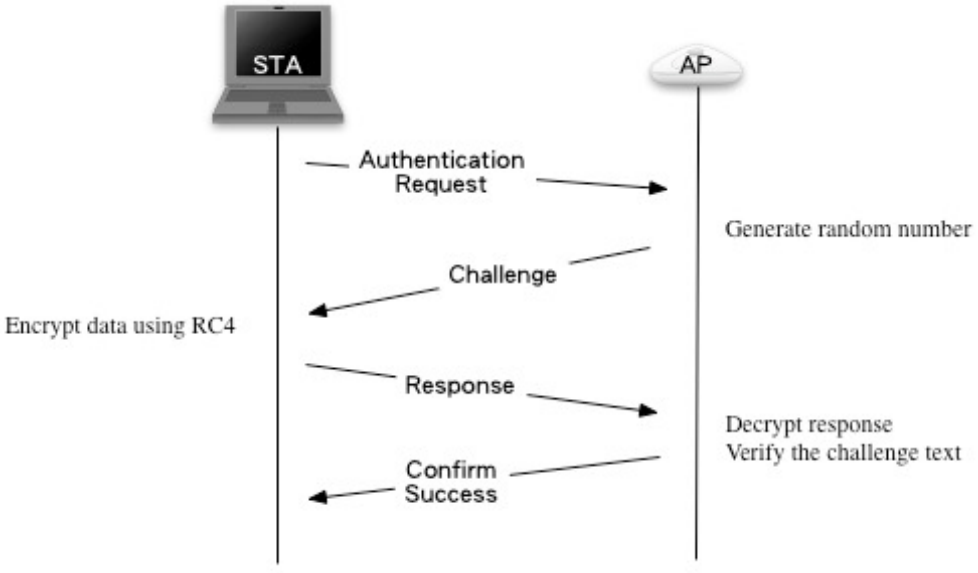


Figure 1: Shared Key Four-way Challenge-response Handshake Message Flow

But several serious weaknesses have been found for WEP. Key size is one of the major security limitations in WEP. Ideally, larger key size means stronger protection. In general, the longer key size requires interception of more packets in the network to crack the key, but the cracker can easily simulate necessary traffic of packets. In addition, the possibility of IV collisions and altered frames cannot be lower by increasing the size of the key.

Second, CRC-32 used by WEP can be cracked by bit flipping attacks. That means a cracker can know which CRC-32 bits will change when data bits are modified. The creators of WEP tried to avoid this problem by producing an integrity check value (ICV) from encrypted CRC-32. But unfortunately, it has been proved that the WEP ICV cannot offer any additional protection against bit flipping attacks.

Besides, an attacker without authorization can capture the authentication messages transmitted between STA and AP. Then these messages might be retransmitted to trick the receiver into unauthorized operations. This kind of attack is called a ‘replay attack’. WEP offers no features such as a sequence number, time-stamp, or other temporal random data for the receiver to check if the message has been received before. Thus, WEP cannot prevent the replay attacks.

As a result, A WEP connection can be easily cracked with some software within minutes. Therefore, Wi-Fi Protected Access (WPA) [2] was introduced by the Wi-Fi Alliance in 2003.

2.3 WPA

The design of Wi-Fi Protected Access (WPA) is based on a Draft 3 of IEEE 802.11i standard, and it was proposed to ensure the release of more security Wireless LAN (WLAN) products before IEEE group could officially introduce 802.11i when the serious weaknesses of WEP had been well known at that time.

Due to the weakness of WEP, WPA introduces some improvements. First, WPA can either be used with an IEEE 802.1x authentication server where each user will be given different keys or be used in a less secure "pre-shared key" (PSK) mode where every client is given the same pass-phrase.

Moreover, for data encryption, WPA uses the RC4 stream cipher, with a 128-bit key and a 48-bit IV, which is quite similar to the WEP. But unlike the WEP, there is a major improvement for WPA to use the Temporal Key Integrity Protocol (TKIP), which can dynamically change keys when it is used and have much larger initialization vector, so related-key attack, which can be used to crack WEP, is successfully blocked by WPA.

In addition, unlike the insecure CRC-32 in WEP, WPA uses a more secure Message Authentication Code (MAC, also known as MIC for Message Integrity Code), which includes a frame counter that can prevent replay attacks.

Through the improvements mentioned above, WPA successfully provides a more secure WLAN and makes breaking into the network more difficult.

2.4 IEEE 802.11i

Due to the introduction of "Fluhrer, Mantin, and Shamir (FMS) attack" [3] that allows an attacker to reconstruct the key from a number of collected RC4 encrypted messages, IEEE 802.11i [4] standard (or WPA2) finally showed up in 2004 to replace less secure WEP and WPA. IEEE 802.11i standard not only adapts all the improvements included in WPA, but also introduces a new AES-based algorithm called CCMP, which is considered fully secure.

Besides, 802.11i also introduces the concept of Robust Security Network (RSN). The RSN enhances the weak security of WEP and provides better protection for the wireless link by allowing the creation of Robust Security Network Associations (RSNA) only. An RSNA is a logical connection between wireless clients. In addition to the RSN, The IEEE 802.11i has another class of security framework: pre-RSN. The pre-RSN is similar to the legacy one in the original IEEE 802.11 specification: open system, shared key, and WEP. The structure of RSN and pre-RSN is plotted in Figure 2.

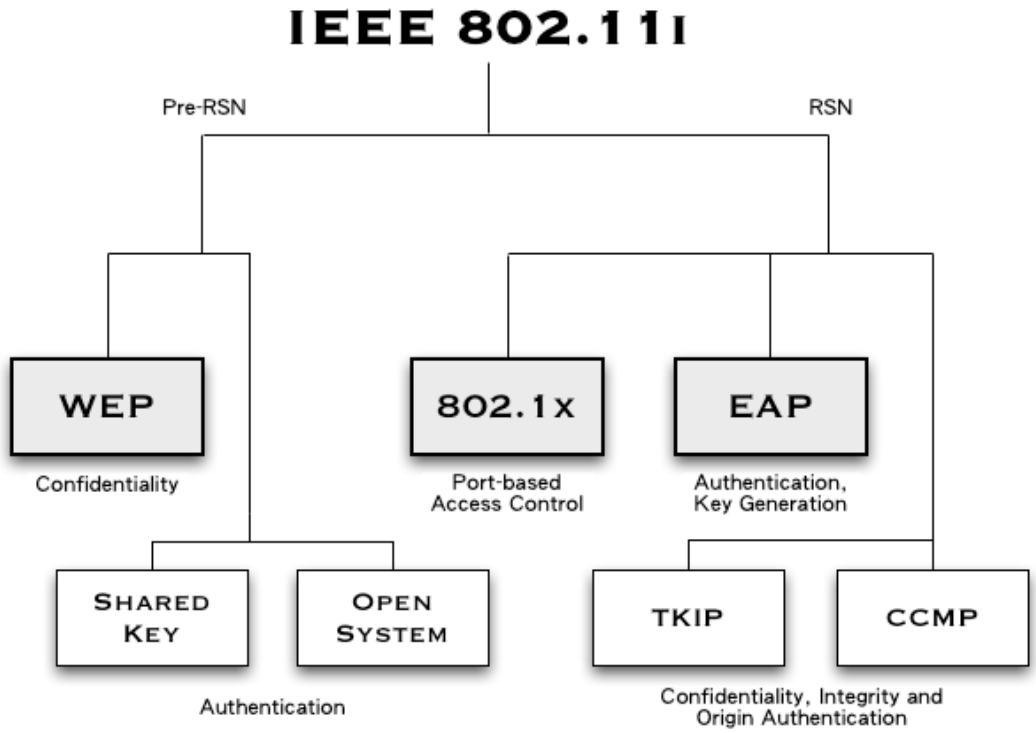


Figure 2: Tree Structure for pre-RSN and RSN

For the confidentiality, integrity and origin authentication, the 802.11i provides two algorithms: Temporal Key Integrity Protocol (TKIP) and Counter mode with CBC-MAC Protocol (CCMP). TKIP is adapted from WPA and uses the RC4 stream cipher, whereas CCMP uses a better cipher algorithm, the Advanced Encryption Standard (AES) algorithm.

Then for the key management, IEEE 802.11i provides two different methods: manual and automatic key management. The manual method requires users to install a pre-shared key (PSK) before establishing an association. The automatic method adapts the 802.1x model to do the key management services, but it only supports RSN framework.

The IEEE 802.11i key management scheme, also called "The Four-Way Handshake" is a protocol that is used to make sure both the supplicant (STA) and the authentication server behind the Authenticator (AP) share a same pairwise master key (PMK). The PMK is distributed in two methods, PSK or IEEE 802.1x [5] port-based access control, mentioned in last paragraph. The followings describe the behavior of the handshake plotted in Figure 3:

1. The AP sends a nonce-value (ANonce) to the STA. Now the STA has all attributes needed to construct the Pairwise Transient Key (PTK).

2. The STA generates a MIC and sends it with another nonce-value (SNonce) to the AP.

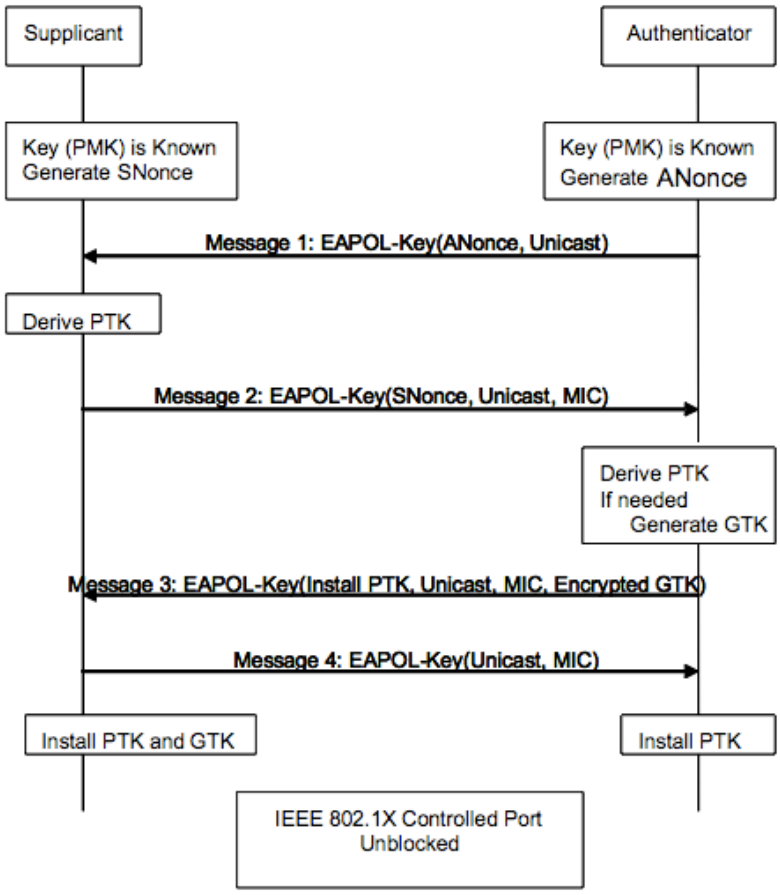


Figure 3: IEEE 802.11i Four-way Handshake Message Flow

3. The AP generates the GTK and another MIC. Then the AP sends them to STA. There is a sequence number that will be used in the next multicast or broadcast frame included in this frame to prevent the replay attacks.

4. After receiving the message 3 sent by the AP, STA sends an ACK (acknowledge) to the AP. This completes the handshake.

In addition to the Four-Way Handshake, IEEE 802.11i also has Group Key Handshake, which is used by the AP to send a new GTK to a STA. The GTK needs to be updated when a wireless client joins or leaves the network. Besides, it may need to be updated due to the expiry of a specific timer in the AP or upon the STA initiation. Group Key Handshake is a two-way handshake plotted in Figure 4 and described in the following:

1. The AP sends a new GTK to every STA in the network. The GTK is encrypted, and there is a MIC sent with it to prevent the GTK from being tampered.

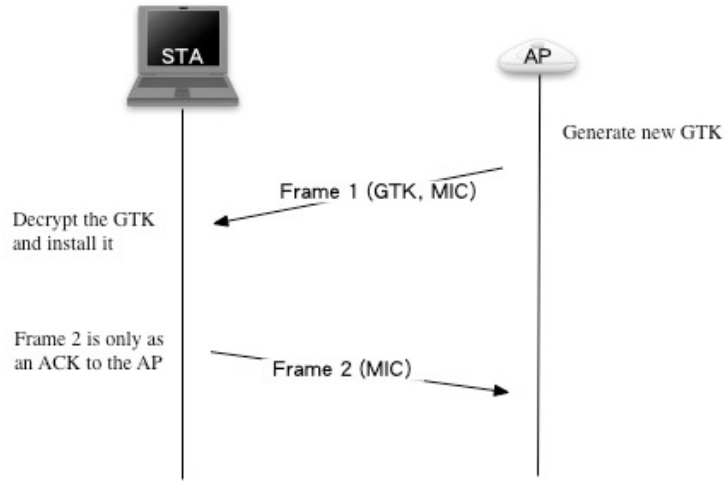


Figure 4: IEEE 802.11i Group Key Handshake Message Flow

2. When the STA receives the new GTK, it sends back an ACK to the AP.

IEEE 802.11i adopts IEEE 802.1x standard in many ways. The next section will briefly describe this standard.

2.5 IEEE 802.1x

IEEE 802.1x is a standard for port-based network access control mechanism. It provides an authentication and authorization mechanism for devices attached to a LAN port.

IEEE 802.1x is based on Extensible Authentication Protocol (EAP) [6] which is an authentication framework providing some common functions and a negotiation of the desired authentication mechanism. EAP only defines message formats, so every protocol that includes it needs to specifically define a way to encapsulate EAP messages. In IEEE 802.1x, it defines EAP over LANs (EAPOL) to encapsulate EAP messages.

IEEE 802.1x defines several terms related to authentication: supplicants, authenticators and authentication servers (The basic structure is plotted in Figure 5):

- **Supplicant:** An entity is similar to a client, like the STA, that will be authenticated by an authenticator.
- **Authenticator:** An entity, like the AP, handles communications between authentication servers and supplicants.

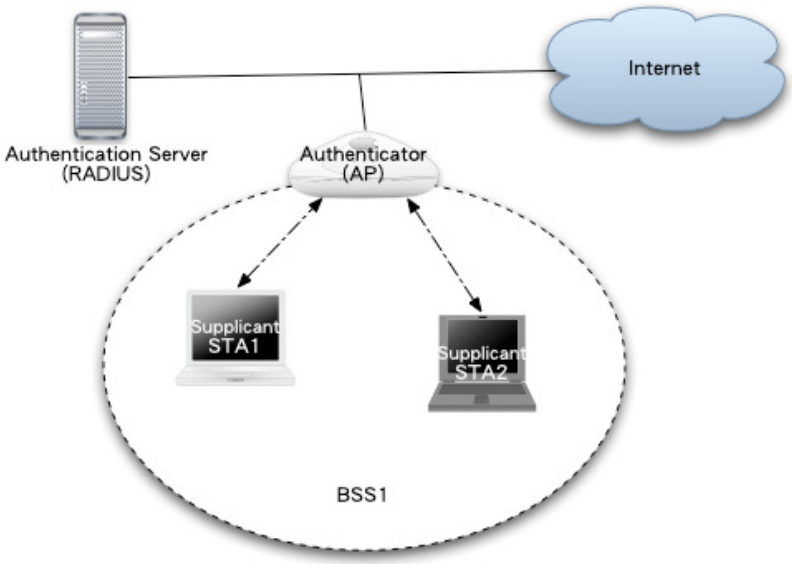


Figure 5: Overview of IEEE 802.1x Basic Structure

- **Authentication Servers:** An entity, such as a RADUIS server, provides an authentication service to an authenticator.

According to IEEE 802.1x, the port controlled by the authenticator is initially in the unauthenticated state. Messages sent to the authenticator will be redirected to the authentication server through authenticator's Process Access Entity (PAE), which is responsible for the authentication-related mechanism, and then the server will start to process the authentication. If finally the authentication is successful, the port controlled by the authenticator will change to the authenticated state. Thus, supplicant can access network services or resources through this port.

3. WIRE1x

3.1 Introduction to WIRE1x

WIRE1x is an open source implementation of IEEE 802.1x client (supplicant) developed by Wireless Internet Research and Engineering (WIRE) Laboratory, National Tsing Hua University. This software provides users a convenient way and easy-to-use GUI to access the network by various authentication mechanisms defined by IEEE 802.1x. The implementation of WIRE1x is based on Open1x, which is a free Linux software designed for WLAN authentication. Many users use Microsoft Windows, but Windows itself only supports few authentication mechanisms; besides, there is very little free software that could provide a comprehensive and convenient authentication solution. Thus, WIRE1x is designed for supporting various versions of Microsoft

Windows, and much more methods of authentication mechanisms is supported in WIRE1x to make up the shortness of the software included in Windows.

The structure of WIRE1x, plotted in Figure 6, is combined by three major components:

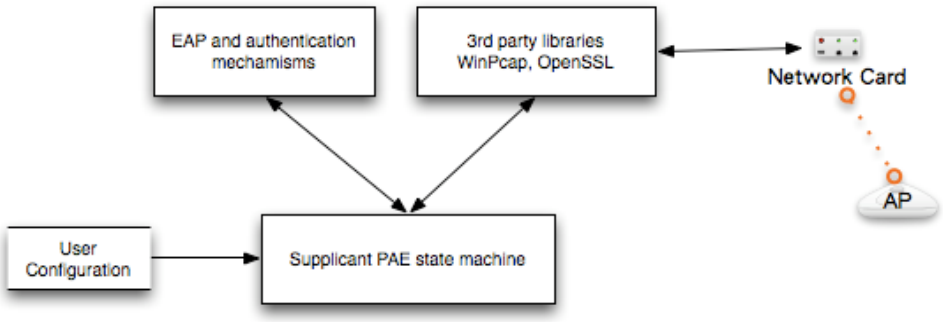


Figure 6: Basic Structure of WIRE1x

1. **Supplicant PAE state machine:** This state machine is based on the specification defined in IEEE 802.1x standard. It defines the behavior of WIRE1x during the process of authentication.

2. **EAP and authentication mechanisms:** WIRE1x supports many different kinds of authentication mechanisms in EAP [7]. In the latest version (2.2), WIRE1x supports four different EAP authentication mechanisms, including EAP-MD5 [8], EAP-PEAP [9], EAP-TLS [10], and EAP-TTLS [11]. And after the construction of EAP secure tunnel is accomplished, WIRE1x support three different kinds of key-management mechanisms: WEP Keys, WPA, and 802.11i in the latest version.

3. **3rd party libraries:** WIRE1x makes use of many free third party libraries, such as WinPcap [12], and OpenSSL [13]. WinPcap is used for capturing and transmitting frames between AP and supplicant, and it also acts as a packet filter to help WIRE1x to receive EAP frames only. The aim of OpenSSL is to encrypt and decrypt messages required by the TLS authentication methods. WIRE1x also uses it to load server certificates, client certificates, and the client private key in different formats that are required during the process of authentication.

3.2 Enhancement in WIRE1x

Our major objective for WIRE1x is to let it support the IEEE 802.11i Standard (WPA2). As we mentioned in early chapters. IEEE 802.11i is the most secure framework for IEEE 802.11

WLAN for now, and it is now widely adapted by almost all network device vendors. Moreover, many big corporations, organizations, and schools have already chosen IEEE 802.11i as their security scheme for the Wireless LAN. Thus, it is quite important for WIRE1x to support this standard.

The implementation of IEEE 802.11i in WIRE1x supports both TKIP and CCMP security protocol, and it is compatible for WPA, too. But in this version, we are not going to implement the pre-shared key mode, which is less secure for 802.11i.

4. System Description

In this chapter, we are going to describe the implementation details including the new architecture and features.

4.1 System Architectures

The whole process of authentication in IEEE 802.11i, plotted in Figure 7, can be divided into four phases:

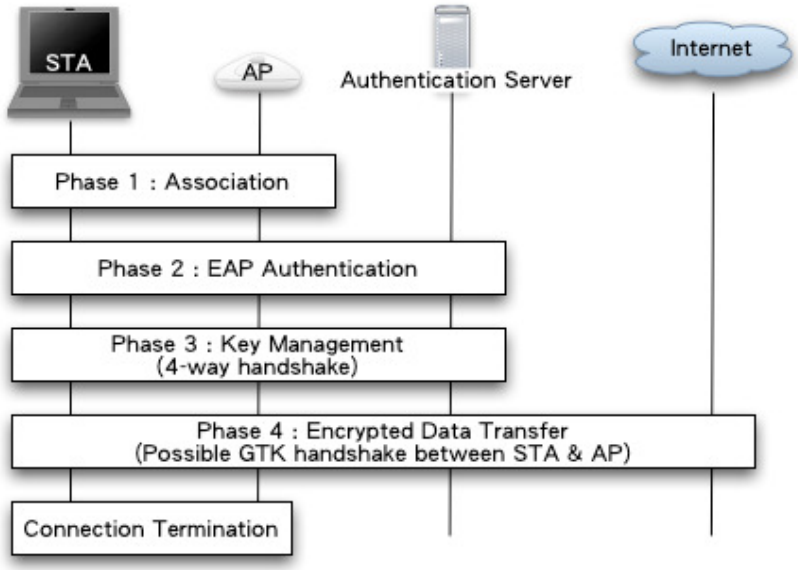


Figure 7: The Four-phase of The Whole Process of IEEE 802.11i Authentication and Key Management

1. **Establishing the IEEE 802.11 association:** First, STA needs to associate the AP, and they both exchange the information about what kinds of key-management mechanisms it

supported. Thus, they can decide which method they should use for the remaining process of authentication. The message flow of this phase is showed in Figure 8.

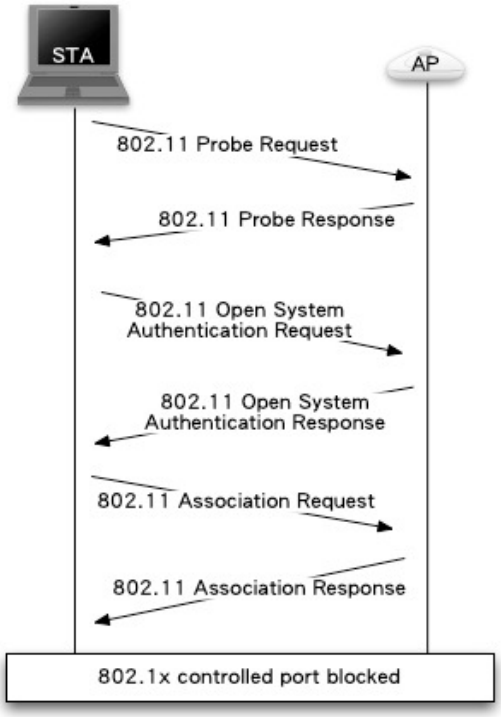


Figure 8: IEEE 802.11 Association Phase Operation Message Flow

2. **IEEE 802.1x EAP authentication:** At this step, both STA and AP exchanging EAP authentication frames try to establish an EAP secure tunnel based on the IEEE 802.1x standard. The message flow of this phase is showed in Figure 9.

3. **Establishing pairwise and group keys:** STA and AP do the 4-way handshake we've discussed in the previous chapters inside the EAP secure tunnel. After the success of the 4-way handshake, STA now has finished the authentication process with the AP

4. **Delivery of subsequent group keys:** If the AP later changes the GTK, it sends the new GTK, using the Group Key Handshake, to every STA that is still in the network and has finished the process of authentication.

In the last version of WIRE1x (2.1), the service Wireless Zero Configuration included in Microsoft Windows handles the first step, and the second step has been implemented in WIRE1x. But the implementation of these two steps needs a little modification to meet the standard. For example, at the first step, WIRE1x should have the ability to retrieve the RSN information ele-

ment, which stores the information about what kinds of authentication mechanisms the device can use and about to use, sent by the AP.

Since there is another team which is responsible for the implementation of first and second

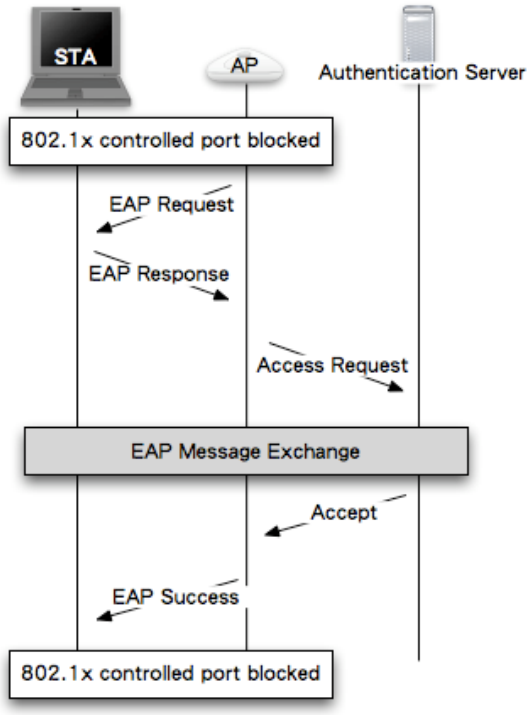


Figure 9: IEEE 802.1x Authentication Phase Operation Message Flow

steps, we only need to focus on the last two steps of the process: the Four-way Handshake and the Group Key Handshake.

The flow chart of the whole process is indicated in Fig. 10. Then Fig. 11 is the architecture of the implementation of the process in WIRE1x and how it processes an IEEE 802.11i EAP frame.

1. When WIRE1x derive a frame and find out it's an EAP frame during the key process of the authentication, the 'eapol_process_key' function will determine which type of key the frame includes. For an IEEE 802.11i frame, the key data within the frame will be sent to the 'WPA2_process_key' function.

2. The 'WPA2_process_key' function will call the function 'eapcrypt_get_keyblock' to get the key-material from the packet. Then the 'WPA2_determine_key' function will parse

the header of the frame to determine which type of the key we get and sends it to the corresponding functions to process the message.

3. If it's a Four-way Handshake frame, the state machine will act according to the data in this frame. If it's a GTK handshake frame, the 'eapol_key_WPA2_do_gtk' function will read the frame and retrieve the new GTK from the frame.

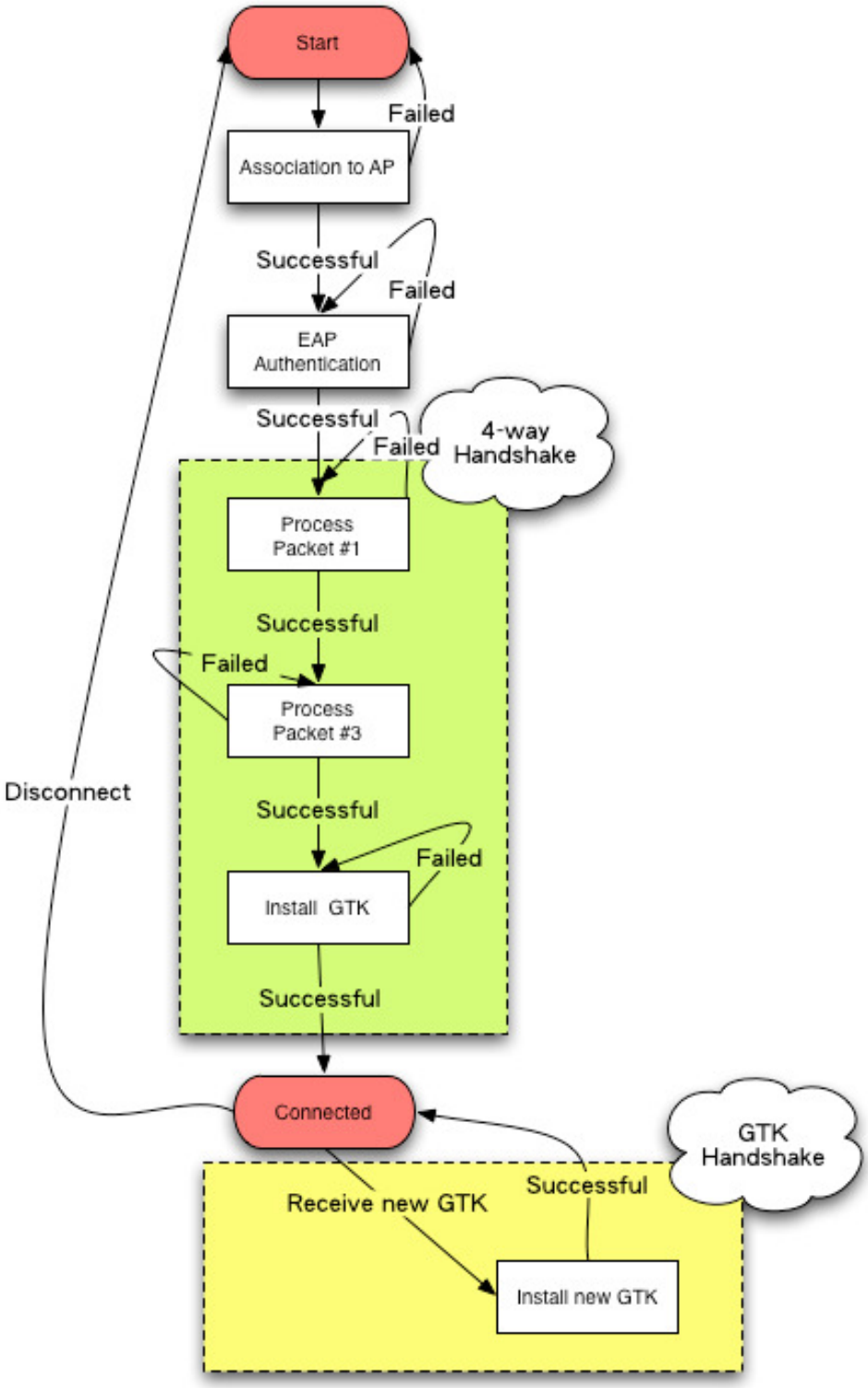


Figure 10: Flow Chart of Whole Process of IEEE 802.11i Authentication

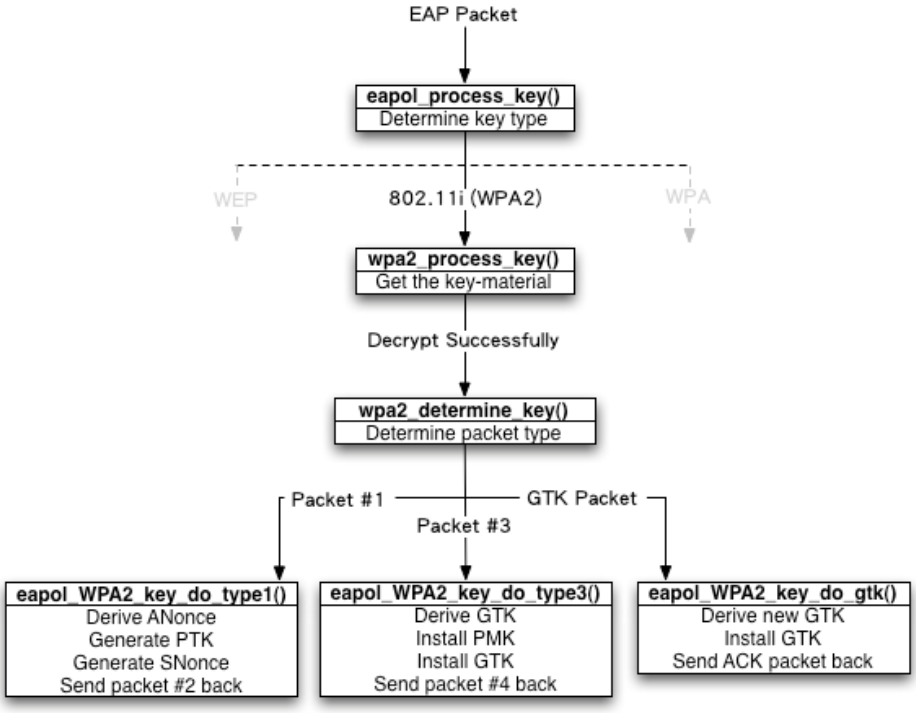


Figure 11: Flow Chart of Processing IEEE 802.11i Authentication Packet

4. After the processing of the frame, the ‘ WPA2_process_key’ function might send a response to the AP according to the standard.

4.2 The Four-way Handshake

In the implementation of WIRE1x, there are two functions to handle the corresponding frame sent from the AP during the 4-way handshake:

1. **eapol_key_WPA2_do_type1:** Handling the first packet of the Four-way Handshake. This function first derives the ANonce from the frame, it generates the SNonce, and the ‘eapol_key_wpa2_gen_ptk’ function is called to generate the PTK. Then it constructs 2nd packet, which includes the SNonce and the PTK and sends it to the AP.

2. **eapol_key_WPA2_do_type3:** Handling the third packet in the Four-way Handshake. It first ensures the correctness of this packet by checking the reply number within the frame. After retrieving the group key with the third packet. Then it calls ‘wpa_set_key’ function to set the pairwise key and group key. Finally, it constructs fourth packet, which is an ACK message and sends it to the AP.

4.3 The Group Key Handshake

For WIRE1x, there is one function ' **eapol_key_WPA2_do_gtk**' responsible for the group key handshake.

On reception of first packet of the handshake, this function first checks the correctness of this packet by verifying the reply number within the packet. Then it calls the 'eapol_key_type2_process_keydata' to decrypt the frame to derive the GTK and configure the new GTK into the device. Finally, this function responds by creating and sending 2nd packet of the Group Key Handshake to the AP.

4.4 MIC Process

In every packet of the 4-way handshake and the group key handshake, the MIC is necessary to ensure the integrity of the data within the packet. Thus, in WIRE1x, there are three functions to handle the MIC related tasks:

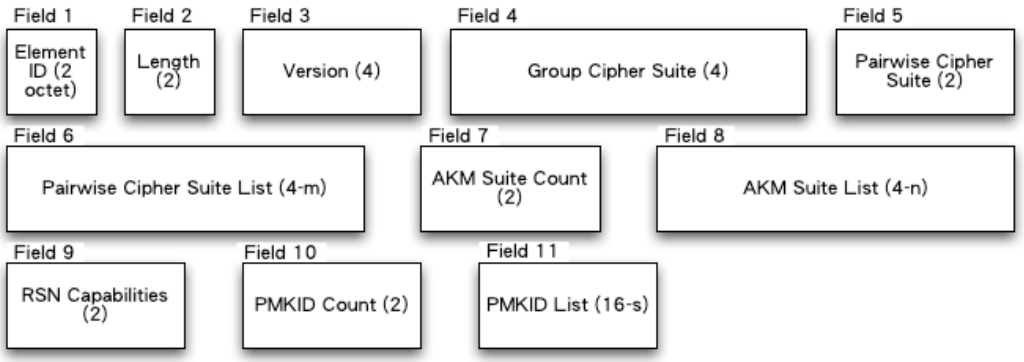
1. **mic_process:** This function calculates the MIC value for a key packet and return the value. It supports both MD5 HMAC and SHA1 HMAC.

2. **mic_wpa_validate:** A key packet will be sent into this function, and it will derive the MIC value by calling the function 'mic_process'. Then it compares the value to the MIC value stored in the header of this packet to ensure the integrity.

3. **mic_wpa_populate:** This function will calculate the MIC value for a given packet by calling the 'mic_process' function. Then it sticks the value into the header of the packet.

4.5 RSN information element

The RSN information element (RSNIE) contains the information about a network device and its capabilities, including its authentication capabilities and what kinds of encryption it supports as illustrated in Fig. 12. When an AP that is going to establish an association will send its RSNIE



m denotes the pairwise cipher suite count, n the AKM suite count, and s is the PMKID count.

Figure 12: RSN information element format

to the STA. When the STA receive the element, it can decide to use what kinds of authentication mechanisms to connect to this AP.

In WIRE1x, a function called 'eapol_key_type2_cmp_ie' is used to compare RSNIEs.

At the association phase, WIRE1x will receive the RSNIE sent from the AP. Then the RSNIE will be stored. During the Four-way handshake, WIRE1x will verify each RSNIE derived from packets the AP sent to ensure the RSNIE is the same as that one the AP sent at the association phase. In addition, there is also a RSNIE in each packet WIRE1x sends to the AP during the Four-way Handshake. WIRE1x will fill the RSNIE field within those packets according to what kinds of authentication standard we use.

5. Conclusions

As the popularity of wireless networks grows, the significant of security issues in the wireless networks is much more essential than before. The security issue has been considered as a major restriction that directly influences the willing of customers to adopt the wireless networks. Thus, the existence of a secure-enough framework for WLAN is very important.

IEEE 802.11i standard introduces the advanced security enhancement for wireless networks. It eliminates various serious security weaknesses in WEP and WPA. Thus, IEEE 802.11i is the most secure standard for IEEE 802.11 WLAN now. It definitely will become popular and be widely adopted by almost everyone who chooses IEEE 802.11 standard to build the wireless networks.

Our implementation of IEEE 802.11i in WIRE1x only support to function with an authentication server, but there is another mode called 'PSK', which does not require a separate authentication server. Although PSK mode is less secure, it is easy to deploy since we don't have to spend extra time setting up the server. For individual and home use, PSK is considered secure enough, so PSK mode is also called 'Personal mode'. As a result, the implementation of IEEE 802.11i PSK mode will be a major topic for the next version of WIRE1x.

Our goal is to provide software, which is secure and easy to use for wireless networks. We also attempt to see that WIRE1x can provide users a pleasant experience of using wireless networks. The implementation of IEEE 802.11i in WIRE1x will undoubtedly be a big help for WIRE1x to achieve these goals. With the improvements we have done so far, we expect that WIRE1x can have a great impact on the both fields of industry and academic of WLAN.

6. Reference

[1] IEEE Standard 802.11b-1999, “LAN/MAN Specific Requirements – Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: High Speed Physical Layer in the 5 GHz band,” 1999.

[2] “Wi-Fi_Protected_Access”, Wi-Fi Alliance, http://www.wi-fi.org/knowledge_center/wpa/

[3] Scott R. Fluhrer, Itsik Mantin and Adi Shamir, “Weaknesses in the Key Scheduling Algorithm of RC4. ” Selected Areas in Cryptography 2001

[4] IEEE Standard 802.11i-2004, “Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications. Amendment 6: medium access control (MAC) security enhancements,” July 2004.

[5] IEEE Standard 802.1X-2001, “IEEE standard for local and metropolitan area networks, port-based network access control,” Oct. 2001.

[6] RFC 3748: Extensible Authentication Protocol (EAP) (June 2004)

[7] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz, “Extensible Authentication Protocol (EAP),” IETF RFC 3748, Jun. 2004.

[8] R. Rivest, “The MD5 Message-Digest Algorithm,” IETF RFC 1321, Apr. 1992.

[9] A. Palekar, D. Simon, J. Salowey, H. Zhou, G. Zorn, and S. Josefsson, “Protected EAP protocol (PEAP), version 2,” IETF Internet Draft, <draft-josefsson-pppext-eap-tls-eap-10.txt>, work in progress, Oct. 2004.

[10] B. Aboba, D. Simon, “PPP EAP TLS Authentication Protocol,” IETF RFC 2716, Oct. 1999.

[11] P. Funk and S. Blake-Wilson, “EAP tunneled TLS authentication protocol version 1 (EAP-TTLSv1)," IETF Internet Draft, <draft-funk-eap-ttls-v1- 00.txt>, work in progress, Feb. 2005.

[12] “WinPcap” http://www.winpcap.org/

[13] “OpenSSL” http://www.openssl.org/