Implementation of WIRE1x WPA Module

Fu-Wen Chen, Ruei-Jiun Wang, Jian-Liang Lin, 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 implementation of Wi-Fi Protected Access (WPA) in 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. The versions of WIRE1x before 2.1 support only Wire Equivalent Privacy (WEP). It has been proved that WEP can be cracked easily within a short period of time. The motivation for developing WIRE1x WPA module is to provide a better security mechanism over wireless networks. The architecture of WIRE1x WPA module is discussed in this document.

1. Introduction

The Wireless Local Area Network (WLAN) becomes more and more popular in recent years because of the convenience and ease of use. The promotion of wireless networking technology results in the growing of services and applications. IEEE 802.11 [1] is a set of standards for WLAN computer communication and it is widely adopted for broadband wireless internet access.

Since wireless networks transmit messages using radio, they are susceptible to eavesdropping. In order to secure IEEE 802.11 wireless networks, IEEE 802.11 standards defined a security mechanism of Wired Equivalent Privacy (WEP), whose goal is to protect the confidentiality of user data from eavesdropping. Therefore, WEP has been integrated by manufacturers into their IEEE 802.11 hardware and has been in widespread use.

Unfortunately, WEP is not as secure as we expected. Although WEP use the well-known RC4 [2] cipher, it still contains many serious security flaws. WEP has proved to be vulnerable [3] because it adopts relatively short Initialization Vectors (IVs) and keys that remain static. In order to enhance security of IEEE 802.11, a new standard called IEEE 802.11i [4] has been released in July 2004.

IEEE 802.11i is an amendment to IEEE 802.11 standard specifying security mechanisms for wireless networks. This standard defines three security mechanisms, which are key management and establishment, encryption, and authentication improvement. However, it takes too much time to accomplish the development of IEEE 802.11i; when more and more attention is paid to the wireless network security, it's urgent to find a substitute for IEEE 802.11i.

Hence, WPA was proposed by the Wi-Fi Alliance, an industry trade group, which owns the trademark to the Wi-Fi name and certifies devices that carry that name. WPA implements a subset of IEEE 802.11i and is compatible with WEP; hence, IEEE 802.11b network interface cards can support WPA. Besides, WPA can be applied to an IEEE 802.1x authentication server, which distributes different keys to each user. In the mechanism of WPA, user data are encrypted using the RC4 stream cipher, which is also used in WEP. The major improvement in WPA over WEP is the Temporal Key Integrity Protocol (TKIP), which dynamically changes keys as the system is used. Moreover, the length of IV in WPA is longer than the one in WEP. For this reason, some attacks can be avoided, such as the well known key recovery attacks [5] on WEP.

The rest of this document is organized as follows. Section 2 takes an introduction to WEP, and the flaws of WEP will be discussed. In section 3, we first take an overview on WPA and then describe the handshake and TKIP in WPA. Section 4 depicts the implementation details of WIRE1x WPA module. Finally, section 5 offers some conclusions.

2. Introduction to WEP

WEP is IEEE 802.11's encryption mechanism implemented in the medium access control (MAC) layer that most radio network interface card and access point vendors support. In the rest of this section, we give an overview of WEP and show the security flaws in WEP algorithm.

2.1 The WEP protocol

WEP uses stream cipher RC4 for confidentiality and cyclic redundancy check (CRC) checksum for integrity. A CRC is an error-detecting code that can check whether the frame is correct or not. There are two different key sizes of WEP. One is standard 64-bit WEP, which uses a 40-bit key concatenated with a 24-bit IV to produce a 64-bit keystream by using RC4 algorithm. The other one is the extended 128-bit WEP, which uses a 104-bit key size and is in common use. In the WEP protocol, IV is generated randomly, and it lengthens the life of the key because stations can change IV for each frame transmission. The encryption details are depicted as follows.

As indicated in Fig. 1, the plaintext P, which will be encrypted, is composed of a message M and a CRC c(M). The keystream generated by RC4 algorithm is a long sequence of pseudorandom bytes as a function of the IV v and the key k. This keystream is denoted by RC4(v, k). WEP combines the plaintext with the keystream through a bitwise exclusive-or (XOR) process, which produces the ciphertext and is given by (1).

$$C \quad = \quad P \text{ XOR } RC4(v,\ k) \qquad (1)$$

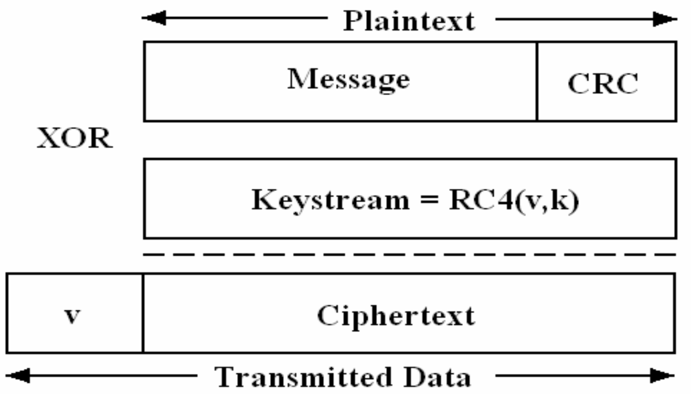


Fig. 1: Encrypted WEP frame [3]

Before transmission takes place, WEP includes the IV, which is unencrypted, within the first few bytes of the frame body. Finally, the sending station transmits the IV and the ciphertext over the radio link. On the other side, the receiving station uses this IV along with the key supplied by the user of the receiving station, which is the same as the key k, to decrypt the ciphertext. The decryption process is simply the inverse of the encryption process. First, the receiving station regenerates the keystream RC4(v, k) and then combines the keystream with the ciphertext through a bitwise XOR process to recover the initial plaintext. This process is given by (2).

$$\begin{aligned} P' &= RC4(v, k) \text{ XOR } C \\ &= RC4(v, k) \text{ XOR } (P \text{ XOR } RC4(v, k)) \end{aligned}$$

$$= P \tag{2}$$

Finally, the receiving station verifies the integrity of the frame data by splitting the plaintext into a message M’ and a checksum c’ and re-computing a new checksum c(M’) and checking whether c(M’) is the same as c’ or not. Only the frame data with a valid checksum will be accepted by the receiving station.

2.2 Authentication of WEP

An access point (AP) must authenticate a supplicant before the supplicant can associate with the AP or communicate with the network. The IEEE 802.11 standard defines two types of WEP authentication: Open System and Shared Key.

Open System Authentication: As indicated in Fig. 2, whenever clients want to connect to the network, clients must send an authentication request to the AP first, and then the AP authenticates clients. If the authentication is finished successfully, then clients associate with the AP and join the network. During the authentication, clients need not provide their credentials to the AP. Therefore, any client can connect to the network regardless of its WEP keys. After the authentication and association, WEP can be used to encrypt data frames, and clients need have correct keys to decrypt data frames.

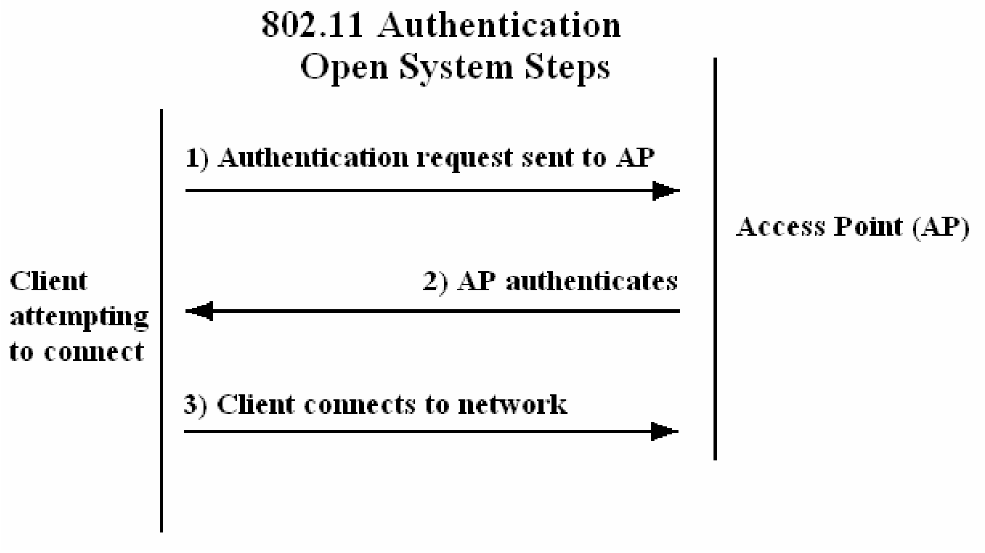

802.11 Authentication
Open System Steps
1) Authentication request sent to AP
Access Point (AP)
Client attempting to connect
2) AP authenticates
3) Client connects to network

Fig. 2: IEEE 802.11 authentication open system steps [6]

Shared Key Authentication: When two devices use Shared Key Authentication, as indicated in Fig. 3, clients will send an authentication request to the AP. The AP will send a challenge text to clients. After receiving the challenge text, clients have to encrypt it by using the configured WEP key and send it back to the AP in another authentication request. The AP decrypts the text and compares it with the challenge text that the AP has sent. Depending on the result of this comparison, the AP sends back a positive or negative response to clients; if the response is positive, client can connect to the network. After the authentication, clients can use WEP to decrypt data frames.

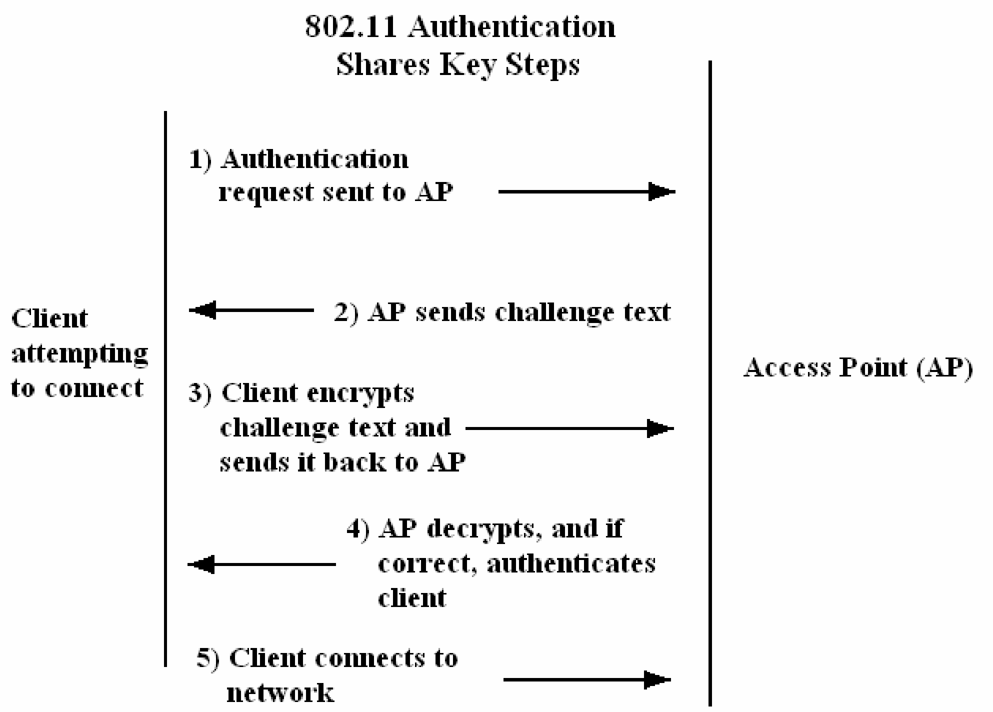

802.11 Authentication
Shares Key Steps
1) Authentication request sent to AP
2) AP sends challenge text
Client attempting to connect
3) Client encrypts challenge text and sends it back to AP
Access Point (AP)
4) AP decrypts, and if correct, authenticates client
5) Client connects to network

Fig. 3: IEEE 802.11 authentication shared key steps

2.3 Deficiencies of WEP

WEP was a nice choice to avoid eavesdropping in the past. However WEP has been found that it can be easily attacked if attackers can get enough data frames. WEP is vulnerable because of its relatively short IVs and static keys. With only 24-bits, WEP must use the same IV for different packets eventually. For the busy network, using the same IV for different packets isn't a rare event; it might happen within an hour or so. Therefore, if an attacker can get enough similar data frames based on the same IV, he can get the plaintext by using XOR. The procedure is given by (3). This

$$\begin{aligned} C_1 &= P_1 \text{ XOR } RC4(v,k) \\ C_2 &= P_2 \text{ XOR } RC4(v,k) \\ C_1 \text{ XOR } C_2 &= (P_1 \text{ XOR } RC4(v,k)) \text{ XOR } (P_2 \text{ XOR } RC4(v,k)) \\ &= P_1 \text{ XOR } P_2 \end{aligned} \tag{3}$$

procedure can cancel out the effect of the keystream; moreover, if any plaintext of the messages was known, the other plaintexts can be known immediately.

Although this problem can be improved by using dynamic keys, IEEE 802.11 doesn't provide any function that supports the exchange of keys among stations. As a result, WEP can’t effectively protect data frames from eavesdropping. In order to replace WEP with a more secure mechanism, WPA was proposed. We will introduce WPA in the next section.

3. WPA

WPA is a mechanism which was designed to regulate the security for data protection on network. The design of WPA is based on Draft 3 of the IEEE 802.11i standard [7]. IEEE 802.11i is designed to replace WEP due to the vulnerability of WEP, but it takes too much time to complete it. Therefore WPA was designed as an intermediate measure to take the place of WEP while IEEE 802.11i was prepared. We will introduce WPA in the following paragraphs.

3.1 Introduction to WPA

There are three enhancements in WPA: key management and establishment, encryption enhancement, and authentication enhancement. These improvements are discussed in the following paragraphs.

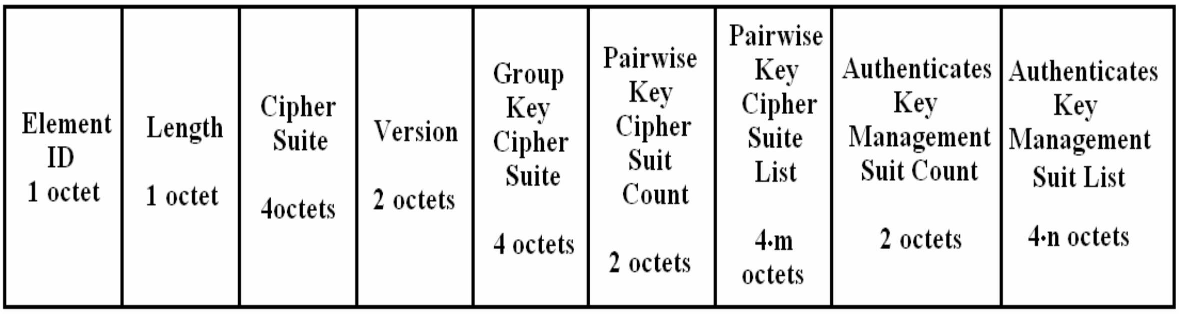
Authentication: In the enhancement of authentication, WPA utilizes IEEE 802.1x for its authentication and key management services. It adds two elements into the IEEE 802.11 architecture: IEEE 802.1x port and authentication server (AS). In IEEE 802.1x, the port represents the association between a supplicant and an AP. Besides, APs and supplicants shall use IEEE 802.11 open system authentication which has been described in section 2 when they use WPA.

Key management and establishment: In the enhancement of key management and establishment, there are two ways to support key distribution: manual key management and automatic key management. The administrator must configure the key manually in manual key management. This way is also called Pre-shared key mode, which is recommended for home use only since it is less secure. Contrarily, the automatic key management relies on not the administrator but IEEE 802.1x to support key management services. This way has more security than Pre-shared key mode.

Encryption: The major enhancement of encryption in WPA over WEP is TKIP, which dynamically changes keys as the system is used. TKIP uses longer IVs to lower the possibility of IV repeats so that some attacks could be avoided. Furthermore, Message Integrity Code (MIC) is used to verify the integrity of messages. The details of TKIP and MIC will be described later.

In addition to the above enhancements, WPA introduces a new component called WPA information element (WPA IE), which is used to negotiate security information between a supplicant and an AP. WPA IE lists authentication and pairwise key cipher suite selectors, and a single group key cipher suite selector. The details of WPA IE are illustrated below.

As shown in Fig. 4, WPA IE contains cipher suite selectors fields, authenticated key management fields, etc. We describe each field as follows: Element ID shall be DD in hexadecimal. Length is the number of octets in the information fields except for Element ID and Length itself. Cipher Suite should be 0050F201 in hexadecimal. The Version field indicates the version number of WPA IE. For example, version 1 represents the station may support IEEE 802.11 Open System Authentication. The Pairwise Key Cipher Suit Count indicates the number of cipher suites in the Pairewise Key Cipher Suite List. The Pair Wise Cipher Suite is used for the communication between two peers that are associated with each other. With cipher suites, both peers could know the security level supported by each other. Similarly, Authenticated Key Management Suite Count indicates the number of Authenticated Key Management Suites in the relative field. The Authenticated Key Management Suite tells the mechanism of key management.



Element ID	Length	Cipher Suite	Version	Group Key Cipher Suite	Pairwise Key Cipher Suit Count	Pairwise Key Cipher Suite List	Authenticates Key Management Suit Count	Authenticates Key Management Suit List
1 octet	1 octet	4octets	2 octets	4 octets	2 octets	4·m octets	2 octets	4·n octets

Fig. 4: WPA IE format

The following figure shows the association establishment process between a supplicant and an authenticator in WPA which assumes that pre-shared key is not used. As shown in Fig. 5 the supplicant attaches to the authenticator using IEEE 802.11 authentication and association process in the beginning of the establishment. In the meantime, the security information and capabilities are negotiated by using WPA IE. Then the supplicant mutually authenticates with the authentication server using IEEE

802.1x authentication process. During the process, Extensible Authentication Protocol (EAP) messages are exchanged between the supplicant, the authenticator, and the authentication server. After this authentication process, if the supplicant and the authentication server authenticate each other successfully, both generate a Pairwise Master Key (PMK) from the authentication independently. Then the authentication server transmits the PMK to the authenticator through a secure channel such as TLS. This PMK is used to derive and verify a Pairwise Transient Key (PTK) in four-way handshake and guarantees fresh session keys between the supplicant and the authenticator. Afterward, the group key handshake is initiated. In the group key handshake, the group key is generated and refreshed. This key ensures the security of broadcast and multicast messages spreading in the wireless network. So far, we have taken an overview on WPA, and the details of four-way handshake, group key handshake, TKIP, and MIC would be depicted in the following paragraphs.

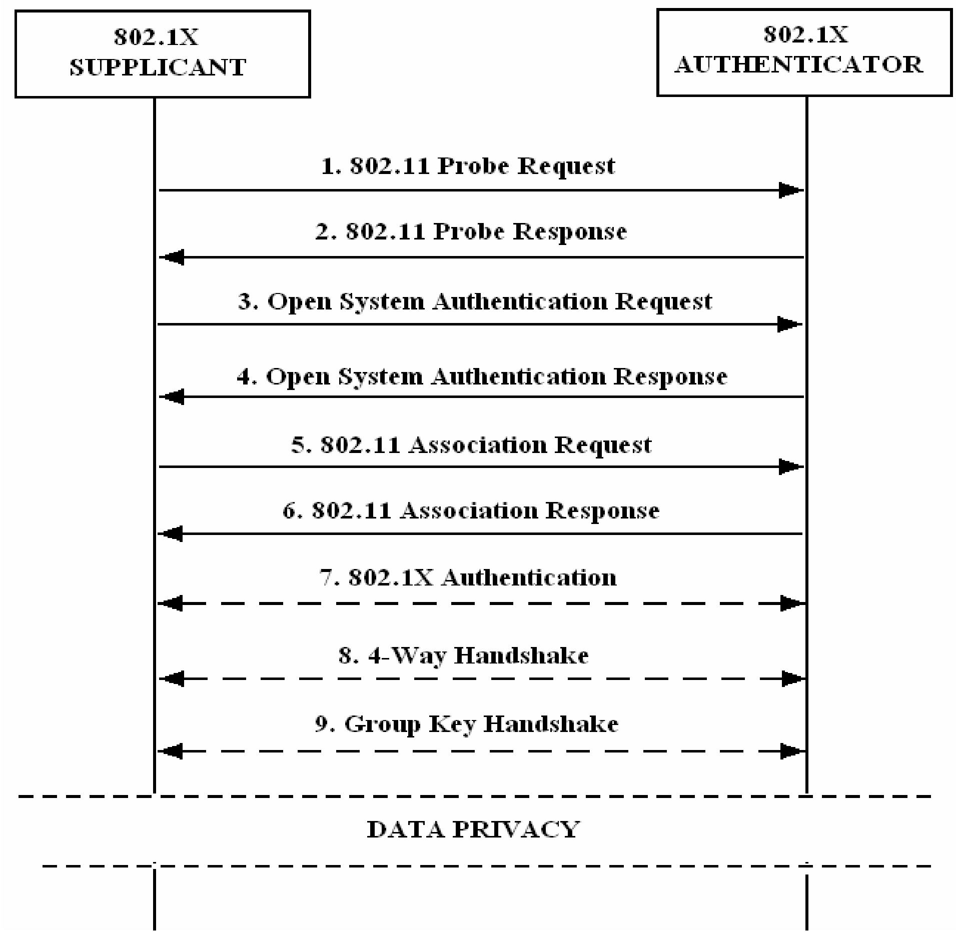


Fig. 5: The association in WPA [8]

3.2 Four-way handshake

WPA defines an IEEE 802.1x protocol called four-way handshake which confirms the liveness of the stations communicating with each other over the IEEE 802.11 link and guarantees the freshness of the session key. Moreover, in order to secure the IEEE 802.11 link, four-way handshake binds PMK to the MAC addresses of the communicating stations and synchronizes the usage of the key. Four-way handshake proceeds by using IEEE 802.1x; in other words, messages exchanged in four-way handshake are in EAPOL-Key format. We depict in detail each message of four-way handshake as follows.

As shown in Fig. 6, the authenticator first sends an EAPOL-Key message to the supplicant. The format of EAPOL-Key message is indicated in Fig. 7. The first message contains key information and an Anonce, which is in the Key Nonce field. The Anonce, also called key material, is a random or pseudorandom value generated by the authenticator, and it will never be reused in four-way handshake; therefore, replay attack can be avoided. After receiving this message, the supplicant first validates the message by checking the Replay Counter field in the messages. The Replay Counter should be greater than the value kept in the supplicant; otherwise, the supplicant discards the message. If the Replay Counter is valid, the supplicant generates a new nonce called Snonce. Then the supplicant derives the Pairwise Transient Key (PTK) by using an algorithm called Pseudo-Random Function (PRF) with some inputs such as Anonce, Snonce, PMK, etc. After the derivation of PTK, the supplicant sends back the second message to the authenticator. The second message contains key information, Snonce, the supplicant's WPA IE, and MIC. MIC is a cryptographic digest used for integrity, and it is computed using the EAPOL-Key MIC Key which is the first sixteen bytes of PTK as indicated in Fig. 8. We will discuss the details of MIC later.

On receiving the second message, the authenticator checks the Replay Counter as the supplicant does when first receiving the message. If the Replay Counter is valid, the authenticator then derives PTK with the same inputs as the supplicant derives PTK. Because of the same inputs, the PTK derived by the authenticator is the same as the one derived by the supplicant. Then the authenticator checks MIC for integrity. If the MIC is not valid, this message will be silently discarded. Besides, the authenticator compares the WPA IE in the message with the one contained in the Association/Reassociation request received from the supplicant earlier. Once these are not identical, the association will be terminated. Otherwise, the authenticator sends the third message to the supplicant. The third message contains key information, Anonce, MIC, and the authenticator's WPA IE.

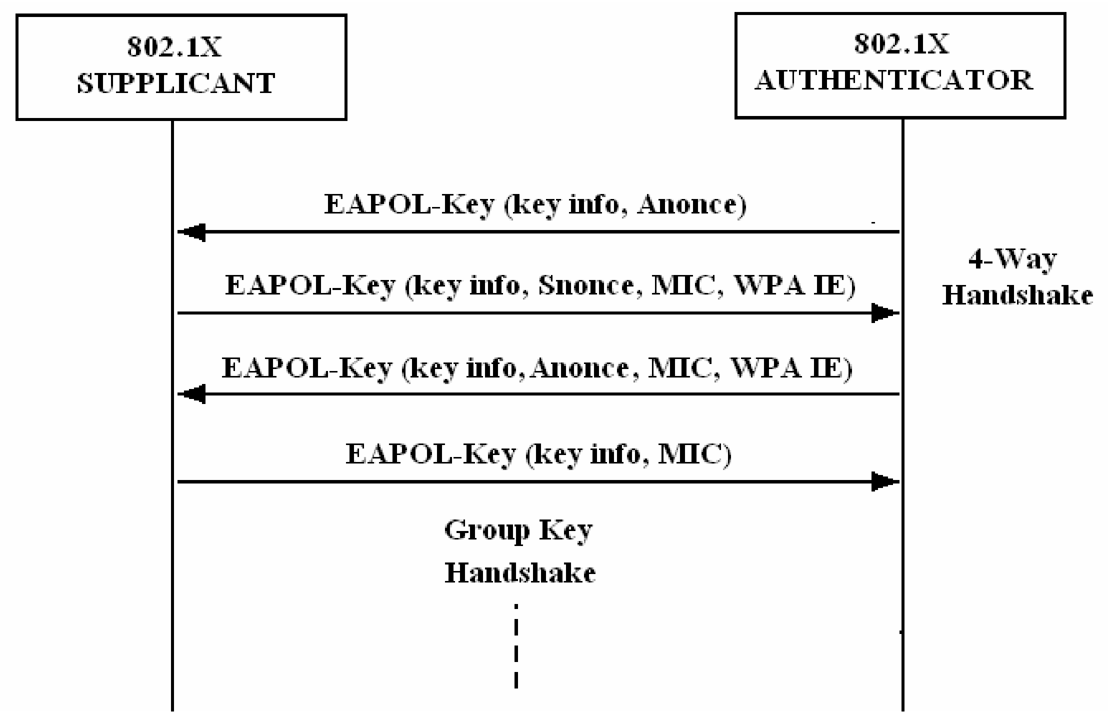
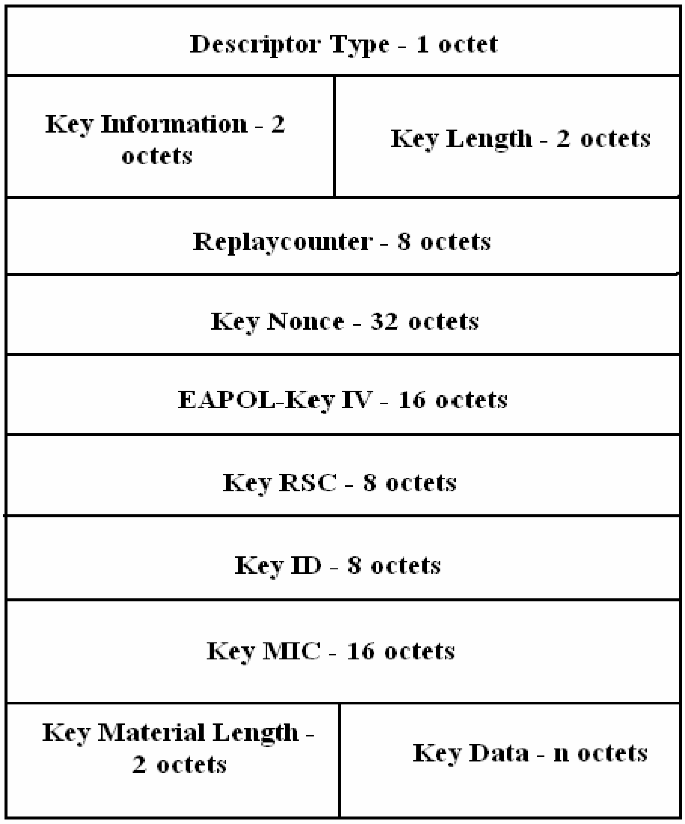

802.1X
SUPPLICANT
802.1X
AUTHENTICATOR
EAPOL-Key (key info, Anonce)
EAPOL-Key (key info, Snonce, MIC, WPA IE)
EAPOL-Key (key info, Anonce, MIC, WPA IE)
EAPOL-Key (key info, MIC)
4-Way
Handshake
Group Key
Handshake

Fig. 6: Four-way handshake



Descriptor Type - 1 octet	
Key Information - 2 octets	Key Length - 2 octets
Replaycounter - 8 octets	
Key Nonce - 32 octets	
EAPOL-Key IV - 16 octets	
Key RSC - 8 octets	
Key ID - 8 octets	
Key MIC - 16 octets	
Key Material Length - 2 octets	Key Data - n octets

Fig. 7: EAPOL-Key format [7]

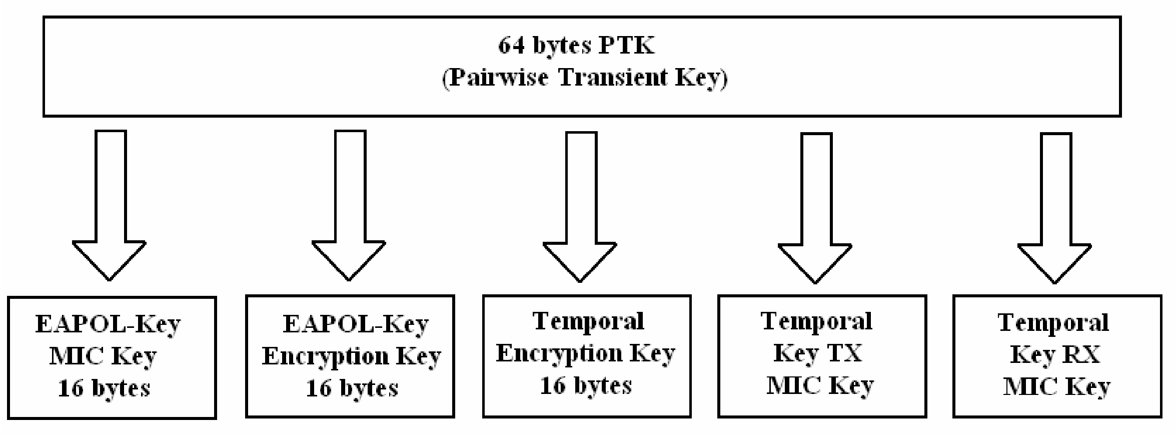

64 bytes PTK
(Pairwise Transient Key)
EAPOL-Key MIC Key 16 bytes
EAPOL-Key Encryption Key 16 bytes
Temporal Encryption Key 16 bytes
Temporal Key TX MIC Key
Temporal Key RX MIC Key

Fig. 8: PTK format

Upon receiving the third message, the supplicant checks the Replay Count and compares the Anonce with the one in the first message. The two Anonces must be the same or the message will be discarded. The supplicant then compares the WPA IE with the one received earlier in the Beacon or Probe Response. If these two WPA IEs are not identical, the supplicant will disassociate from the authenticator. Otherwise, if the WPA IE is valid, the suppliant checks the MIC. If the MIC is valid, the supplicant will send the fourth message to the authenticator. The fourth message contains key information and MIC. After the fourth message is sent, the supplicant configures the PTK into the IEEE 802.11 MAC.

Once the authenticator received the fourth message, it checks the Replay Counter and MIC as before. Then the authenticator configures the PTK into the MAC. This message is used to acknowledge the authenticator that the supplicant has installed the PTK. The above shows the details of four-way handshake.

3.3 Group key handshake

The WPA also defines an IEEE 802.1x protocol called group key handshake which is used by the authenticator to send the Group Transient Key (GTK) to the supplicants so that the supplicants would be able to receive broadcast messages. Group key handshake, like four-way handshake, exchanges the messages in EAPOL-Key format. The authenticator may initiate this as the final stage of authenticating a supplicant.

As shown in Fig. 9, the group key handshake is performed after the four-way handshake. The authenticator first sends a message to the supplicant. The first message contains key information, MIC, and GTK. GTK is encrypted using

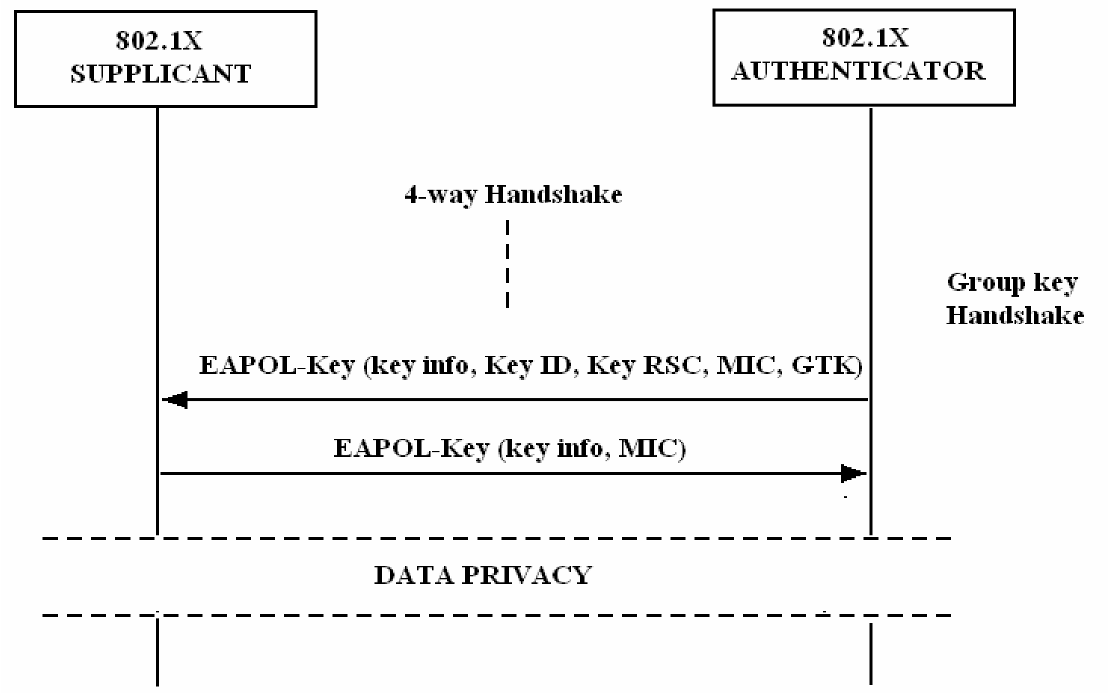


Fig. 9: Group key handshake

EAPOL-Key Encryption Key which is the second sixteen bytes of PTK as indicated in Fig. 8. On reception of the first message, the supplicant check if this message is valid as it does in four-way handshake. If the message is valid, the supplicant decrypts the GTK using the same EAPOL-Key Encryption Key. Then the supplicant configures the GTK into its MAC. Finally, the supplicant sends back a message with key information and MIC to the authenticator. After the authenticator validates this message, the group key handshake is completed.

3.4 TKIP

We has mentioned four-way handshake and group key handshake and described the stations would configure the keys into the MAC. The keys are installed to use TKIP to protect data frames. TKIP is a security protocol which is used in WPA. Although TKIP still uses RC4 cipher to generate keystreams, it improves WEP's flaw by using new algorithms:

1. The sender calculates a cryptographic MIC and appends it to the messages. The receiver checks the MIC when receiving the massages. If the MIC is invalid, the receiver will discard the message.
2. TKIP uses a TKIP sequence counter (TSC), or extended IV, to assign numbers to the sending messages. The receiver will drop the messages which are out of order.
3. TKIP uses the mixing function to combine the temporal key and the TSC into

the WEP seed, which includes the IV. The receiver can use the mixing function to compute the same WEP seed to decrypt messages.

The details about TKIP encapsulation are shown in Fig. 10. At first, TKIP computes the MIC over the message's source address, destination address, and payload with the key and appends the computed MIC to the message. TKIP will assign an incrementing TSC value to each message, and then it mixes the TSC value, key, and transmitter address to generate the Phase 1 Key or TKIP mixed Transmit Address and Key (TTAK). By using TTAK and IV, TKIP can generate WEP seeds, which are represented as a WEP IV and RC4 key. Finally, TKIP passes these seeds with each massage to WEP encapsulation and generates the cipher text.

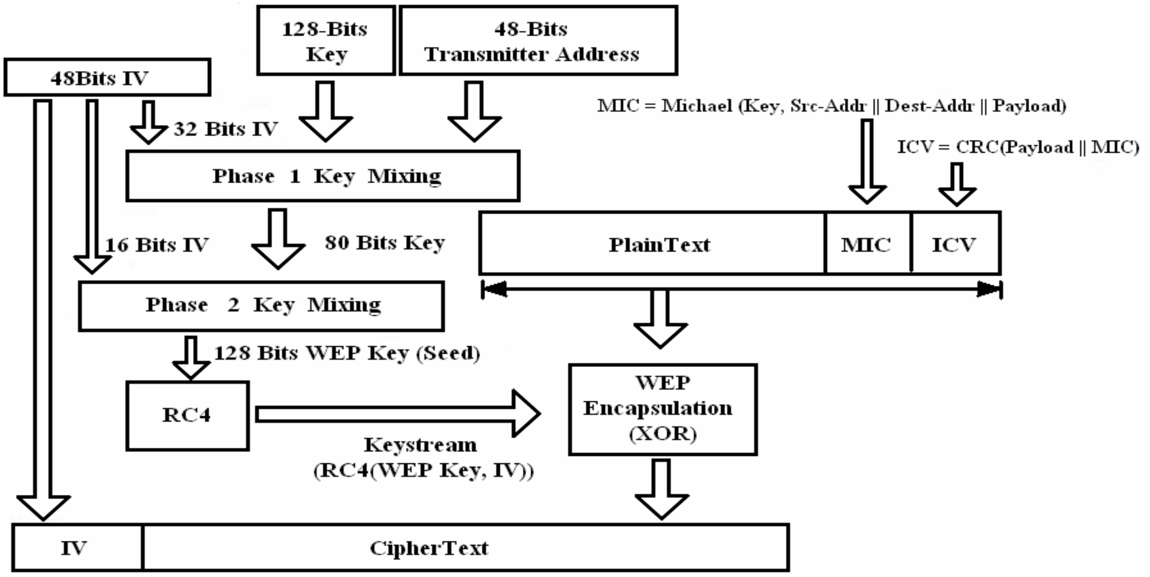


Fig. 10: TKIP Encapsulation Block Diagram

The details about TKIP decapsulation are shown in Fig. 11. At first, TKIP can derive the IV from the received data. TKIP extracts the TSC and key id from the IV and checks if the TSC is valid. TKIP discards the data with invalid TSC or constructs the WEP seed by using the mixing function as it does in encapsulation. TKIP represents the WEP seed as a WEP IV and RC4 key, and passes these with the received cipher text to WEP decapsulation. After decapsulating the cipher text, TKIP verifies the Integrity Check Value (ICV). If the ICV is valid, the receiver verifies the MIC. Both ICV and MIC should be valid; otherwise, the receiver discards the packet. Finally, if all verifications are finished successfully, TKIP delivers the packet to the upper layer, and TKIP decapsulation finishes.

Compare WPA with WEP, WPA uses more elements to generate different keystreams for each message; therefore, it's difficult to attack the wireless network by

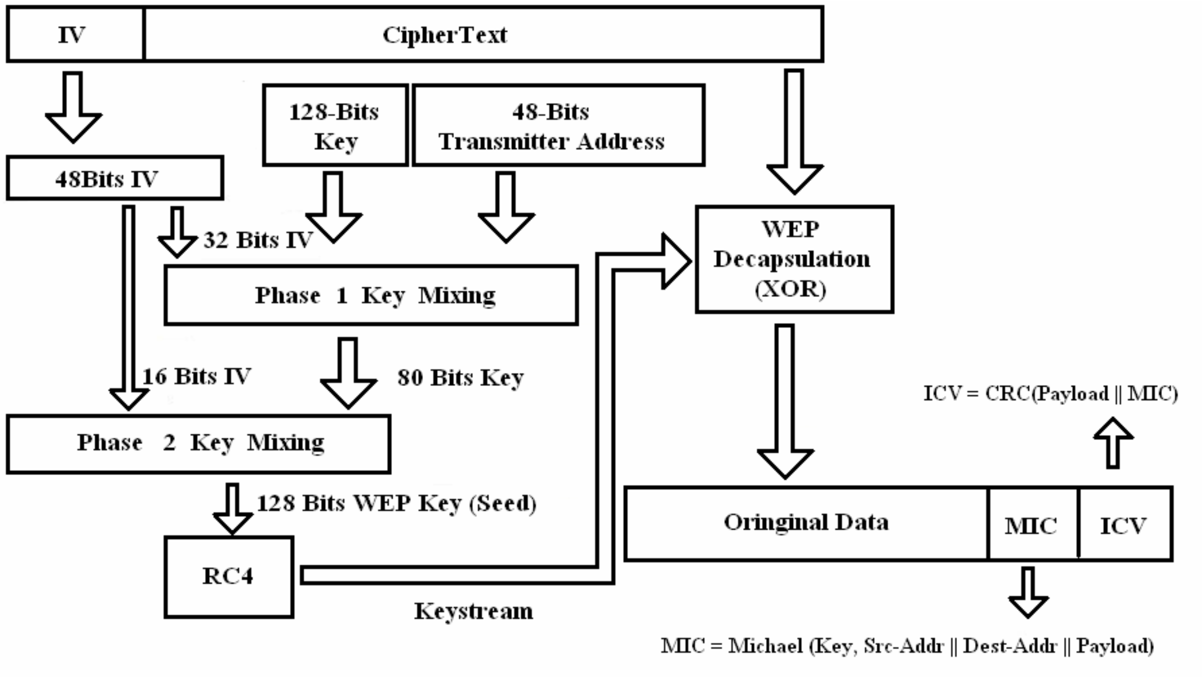


Fig. 11: TKIP Decapsulation Block Diagram

getting lots of packets. In brief, WPA is a more secure protocol against eavesdropping than WEP.

3.5 MIC

MIC is an important element in TKIP and is used to verify the integrity of messages. MIC value is calculated by Michael algorithm. This algorithm uses the key, source address, destination address, and payload to generate a 64-bits MIC value and this value is appended to the message before delivering the message. In the receiving side, the receiver can also use the key, source address, destination address, and payload to generate the same MIC value as the one in the sending message after the cipher text is decapsulated. If the MIC value generated by the receiver is not identical to the one generated by the sender, the message will be discarded. By using MIC, receivers can check the integrity of message. Using MIC is more reliable than using only CRC value to ensure the message's integrity.

4. System Description

This section describes the implementation details of WIRE1x WPA module. The addition of WPA is an enhancement of the original WIRE1x. The information of WIRE1x could be found in the WIRE1x web site [9]. WIRE1x WPA module is implemented based on the supplicant PAE state machine which is the core of the

WIRE1x. Many functions in the original WIRE1x are also used in WIRE1x WPA module. Besides, some open source libraries are still in use. For example, WinPcap [10] is used to capture and send frames, and OpenSSL [11] is used to handle cryptographic hash functions. The details of these libraries will not be discussed in this article; we show only the main architecture of WIRE1x WPA module as below.

After finishing the IEEE 802.1x process, the AP sends an EAPOL-Key message to start four-way handshake. This is the beginning of WIRE1x WPA module. Then the supplicant derives PTK and installs this session key in four-way handshake. Group key handshake proceeds after four-way handshake is done. GTK is derived and installed in group key handshake. The above briefly describes four-way handshake and group key handshake. We then describe WIRE1x WPA module architecture as below.

WIRE1x WPA module is implemented in three files: eapol.cpp, WPA.cpp and mic.cpp. eapol.cpp identifies a WPA type EAPOL-Key message. WPA.cpp implements most functions of handshakes, i.e. the processes of four-way handshake and group key handshake are handled in WPA.cpp. The last file, mic.cpp, only deals with the functions related to MIC such as the validation and the calculation of MIC. We depict the details in the following paragraph.

As shown in Fig. 12, the supplicant prepares to do four-way handshake and group key handshake. We demonstrate the flow and depict the architecture of WIRE1x WPA module with Fig. 12, Fig. 13, and Fig. 14.

1. If the supplicant select WPA as his secure protocol, the AP will send a WPA type EAPOL-Key message to the supplicant after the supplicant PAE state machine transit to the AUTHENTICATED state. This message is identified as an EAPOL-Key message by eapol_decode_packet(). Then eapol_process_key() generates a keyblock with the information derived in the process of the authentication and checks the type of this EAPOL-Key message.
2. After recognizing the message as a WPA message. WPA_process_key() first derives PMK which is the first 32 bytes of the keyblock generated in eapol_process_key(). Then WPA_process_key() calls WPA_determine_key() to handle the message. If WPA_determine_key() finishes successfully and there is an message prepared to send to the AP, this message will be sent out. However, if there are some errors in this message or the AP doesn't receive this message, the AP will retransmit the first message to the supplicant, and the AP can retransmit a message at most twice.
3. WPA_determine_key() first checks the MIC if there is a MIC in the message.

The mic_wpa_validate() is called to verify the MIC. If the MIC is not valid, this message will be discarded silently. Otherwise, eapol_key_WPA_do_type1() will be called to handle the first message.

4. In eapol_key_WPA_do_type1(), the replay counter of the message will be checked to ensure security. The replay counter should be greater than the value kept in the supplicant; otherwise, the message will be discarded. After verifying the correctness of the message, PTK is calculated by using PRF algorithm. After PTK is calculated, the supplicant prepares the second message. In order to generate the second message in four-way handshake, wpa_gen_ie() is called to generate WPA IE of the supplicant. Besides, the MIC of the second message is calculated for integrity and this is completed by mic_wpa_populate(). So far, the second message is ready, and it will be sent by calling send_frame() in WPA_process_key().
5. The AP will send the third message to the supplicant if the second message is correct. With the same process as the first message, the third message will be verified. Like eapol_key_WPA_do_type1(), eapol_key_WPA_do_type3() is called to handle the third message. After the third message is validated, send_frame() will be called to send out the fourth message. This time send_frame() is called not in WPA_process_key() but in eapol_key_WPA_do_type3() because PTK will be installed later. Finally, wpa_set_key() is called to install PTK.
6. After four-way handshake is completed, group key handshake starts. The AP sends a group key message to the supplicant, and eapol_key_WPA_do_gtk() is called to deal with group key handshake. After validating this message, the supplicant needs to derive GTK. Since GTK is encrypted, rc4_skip() is called to decrypt GTK. After deriving GTK, wpa_set_key() is called to install this key. Then a message is made to acknowledge the AP. Now group key handshake is also finished. The above describes the details of WIRE1x WPA module.

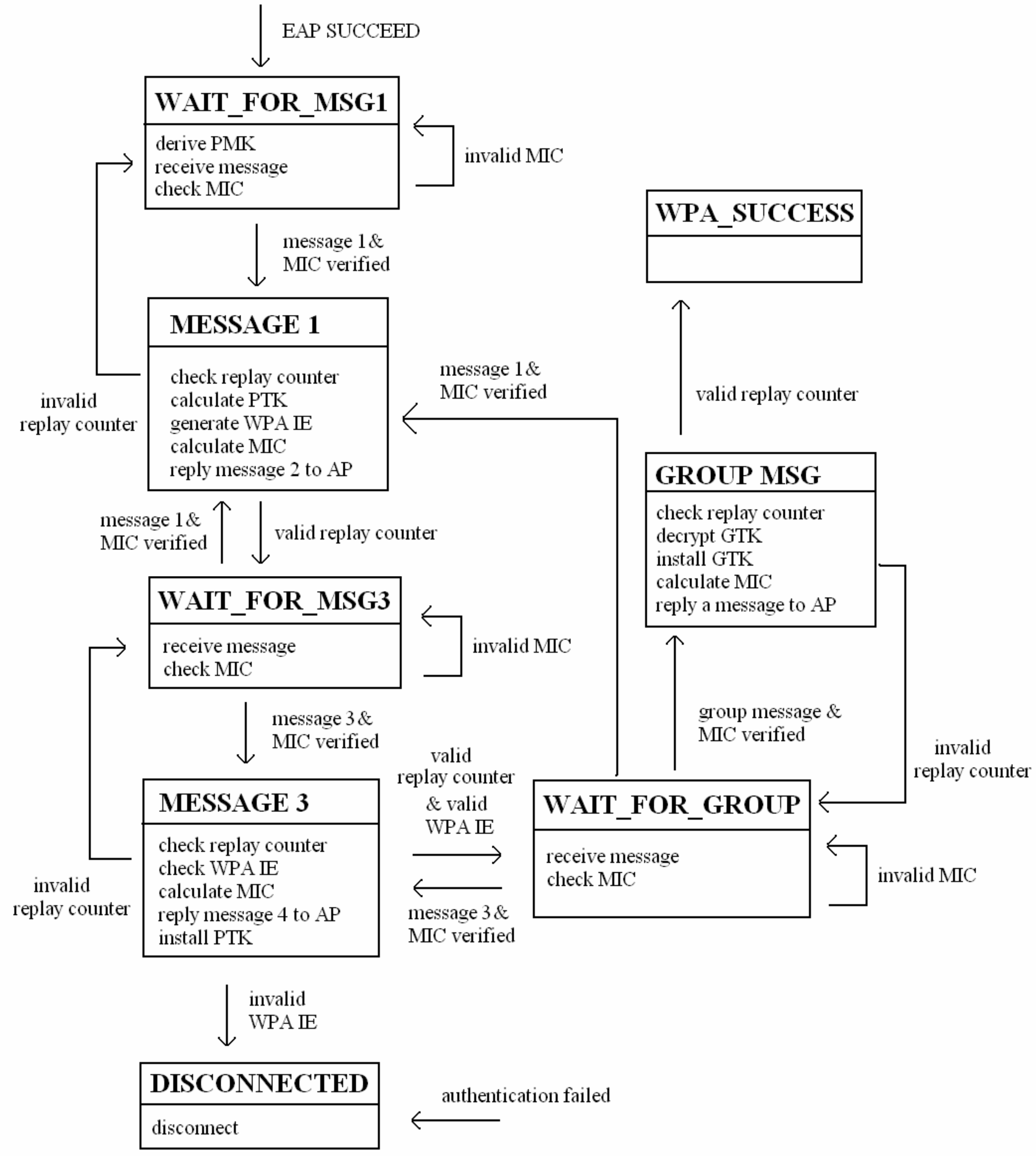


Fig. 12: State diagram of WPA

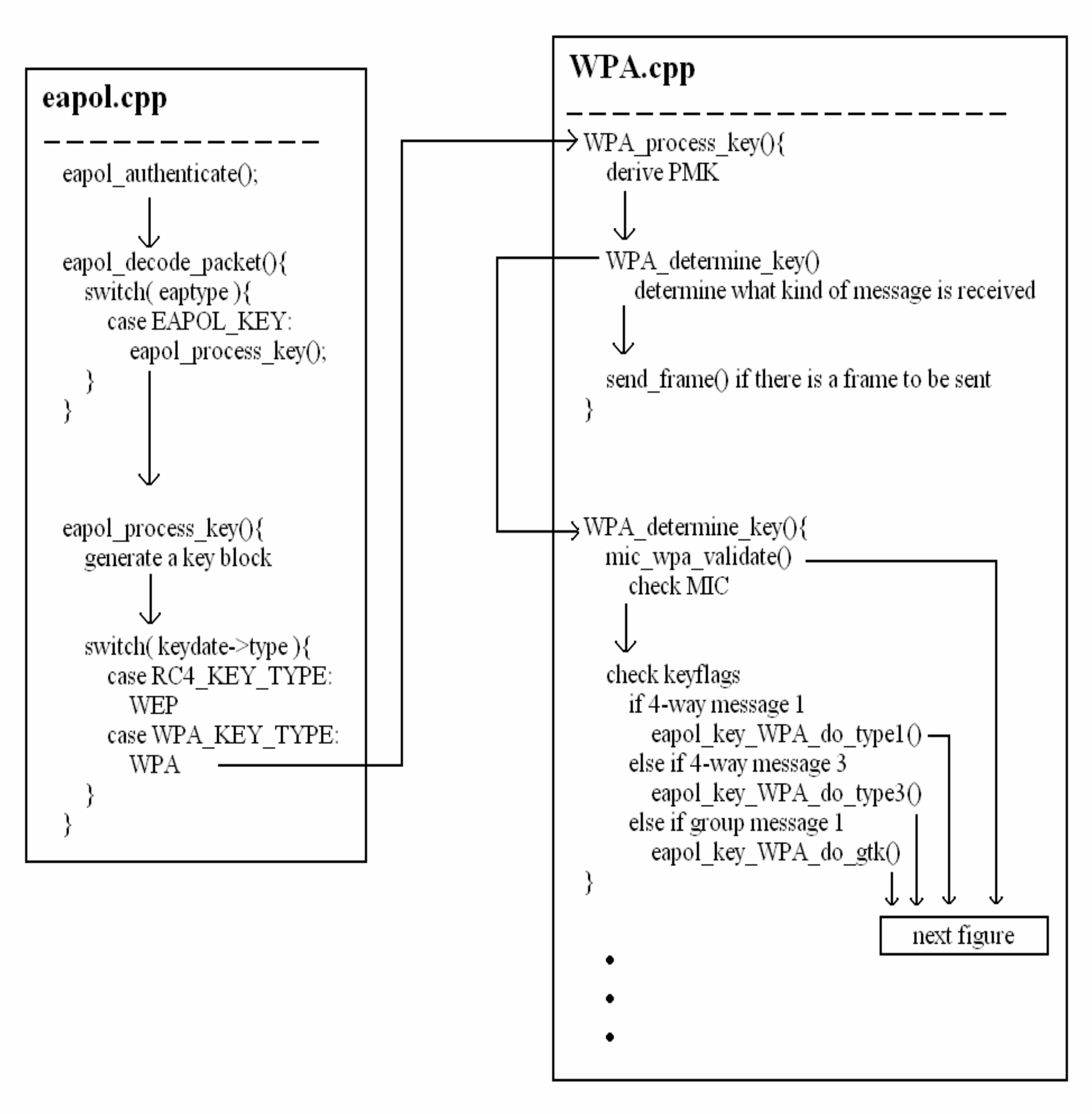
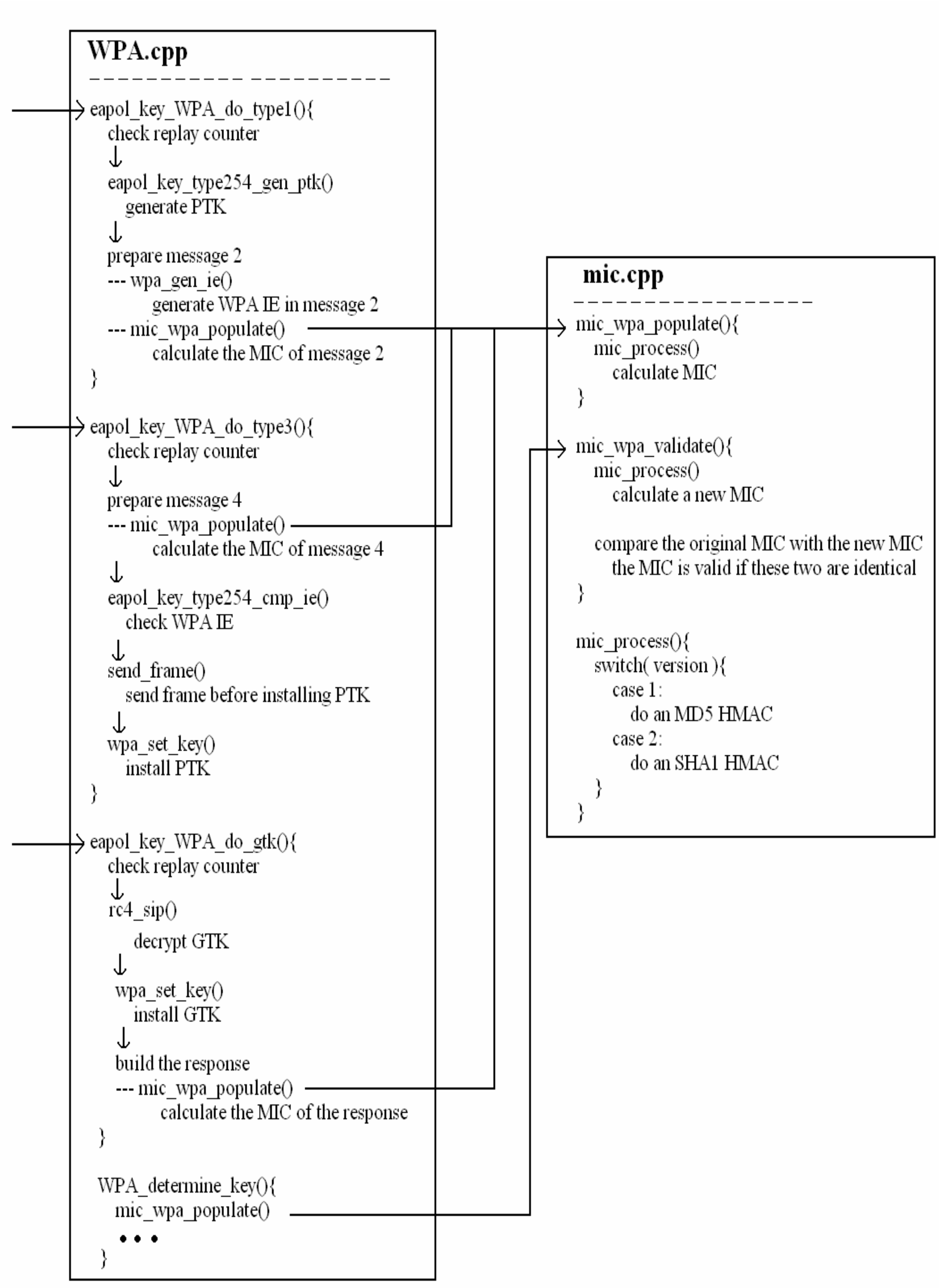


Fig. 13: UML diagram for WIRE1x WPA module part I



```
eapol_key_WPA_do_type1(){
  check replay counter
  ↓
  eapol_key_type254_gen_ptk()
    generate PTK
  ↓
  prepare message 2
  --- wpa_gen_ie()
        generate WPA IE in message 2
  --- mic_wpa_populate()
        calculate the MIC of message 2
}

eapol_key_WPA_do_type3(){
  check replay counter
  ↓
  prepare message 4
  --- mic_wpa_populate()
        calculate the MIC of message 4
  ↓
  eapol_key_type254_cmp_ie()
    check WPA IE
  ↓
  send_frame()
    send frame before installing PTK
  ↓
  wpa_set_key()
    install PTK
}

eapol_key_WPA_do_gtk(){
  check replay counter
  ↓
  rc4_sip()
    decrypt GTK
  ↓
  wpa_set_key()
    install GTK
  ↓
  build the response
  --- mic_wpa_populate()
        calculate the MIC of the response
}

WPA_determine_key(){
  mic_wpa_populate()
  • • •
}
```

```
mic_wpa_populate(){
  mic_process()
    calculate MIC
}

mic_wpa_validate(){
  mic_process()
    calculate a new MIC

  compare the original MIC with the new MIC
    the MIC is valid if these two are identical
}

mic_process(){
  switch( version ){
    case 1:
      do an MD5 HMAC
    case 2:
      do an SHA1 HMAC
  }
}
```

Fig. 14: UML diagram for WIRE1x WPA module part II

5. Conclusion

Due to the popularity of wireless networks, security in wireless networks becomes more and more important. Although IEEE 802.11 standards define a security mechanism, WEP, the security of WLAN is still vulnerable. Therefore, to access the wireless networks through a much more secure mechanism is necessary. This is why WPA was proposed.

To describe WPA, we first introduce the encryption and decryption algorithm of WEP and demonstrate major security flaws in the WEP protocol in this article. After the introduction to WEP, the security enhancements in encryption and authentication developed by WPA are discussed. With the comparison of WEP and WPA, the weakness in WEP and the advance in WPA are obvious. Since the previous versions of WIRE1x support only WEP, in order to provide users with a more secure wireless network, making WIRE1x support WPA is required. The WIRE1x WPA module is hence developed. The implementation details of WIRE1x WPA module are depicted in this article and the implementation follows the specification, Draft 3 of the IEEE 802.11i standard.

It is admitted that WPA is more secure than WEP. However, WPA is not a perfect security mechanism because it still uses RC4 cipher as its encryption algorithm. There are vulnerabilities in RC4 algorithm [12]. Although the occurrence of WPA could temporally improve security services in today's IEEE 802.11 wireless LANs, there will be a better security mechanism after all.

In order to make WIRE1x more powerful, to develop a more secure mechanism in WIRE1x is necessary. Besides, WIRE1x can still be enhanced in many ways. There is still a long way for WIRE1x to be more complete. After all, WIRE1x is expected to have a more significant impact on WLAN.

References

[1] ANSI/IEEE Std 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1999.

[2] R. L. Rivest. "The RC4 Encryption Algorithm," Technical report, RSA Data Security, Inc., March 12, 1992.

[3] Nikita Borisov, Ian Goldberg, David Wagner, "Intercepting mobile communications: the insecurity of 802.11." MOBICOM 2001, pp180–189

[4] IEEE Standard 802.11i, "Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications. Amendment 6: medium

access control (MAC) security enhancements," July 2004.

[5] A. Stubblefield, J. Ioannidis, and A. Rubin, "A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP)," ACM Trans. Inform. Syst. Security, vol.7, no.2, pp.319–332, May 2004.

[6] "Wireless Basics," http://documentation.netgear.com/reference/sve/wireless/

[7] IEEE Std 802.11i/D3.0, Wireless LAN Medium Access (MAC) and physical layer (PHY) specifications: Specification for Enhanced Security, November 2002.

[8] E. Sithirasenan, V. Muthukkumarasamy, "IEEE. 802.11i WLAN Security Protocol – A Software. Engineer's Model," Proc. of 4. th. AusCERT Asia. Pacific Information Technology Security Conference, pages 39–50, May 2005.

[9] "WIRE1x," http://wire.cs.nthu.edu.tw/wire1x/

[10] "WinPcap," http://winpcap.polito.it/

[11] "OpenSSL," http://www.openssl.org/

[12] S. Fluhrer, I. Mantin, and A. Shamir, Weaknesses in the Key Scheduling Algorithm of RC4, Proc. of SAC2001, Lecture Notes in Computer Science, vol.2259, 2001, pp.124.