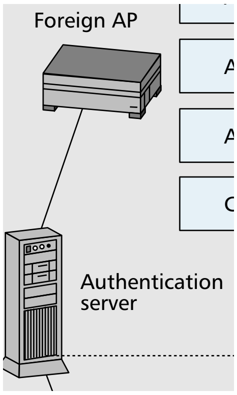
WIRELESS LAN SECURITY AND IEEE 802.11I

JYH-CHENG CHEN, MING-CHIA JIANG, AND YI-WEN LIU
NATIONAL TSING HUA UNIVERSITY



In order to enhance security of the IEEE 802.11 standard, a new standard called IEEE 802.11i is being developed. In addition to introducing key management and establishment, it also defines encryption and authentication improvements.

ABSTRACT

This article reviews wireless LAN security with a focus on the evolving new IEEE 802.11i standard. The major security enhancements in encryption and authentication defined by 802.11i are illustrated. In addition, the newly introduced key management in 802.11i is discussed. Because 802.11i incorporates IEEE 802.1X as its authentication enhancement, 802.1X with consideration of roaming users is depicted. Both intrasubnet and intersubnet roaming are illustrated.

INTRODUCTION

Conventional Internet users have been bound to wired connections. Wireless communications, however, have broken this restriction and provide ubiquitous access to the Internet. In addition, increased flexibility strongly motivates wireless network technologies. Today, the deployment of wireless local area networks (WLANs) is sometimes even more economical and efficient than installing wired networks in a whole building. With promotion of wireless networking technologies and their market, services and applications are growing tremendously. The IEEE 802.11 standard [1] for WLANs is one of the most widely adopted standards for broadband wireless Internet access.

However, security over a wireless environment is more complicated than in a wired environment. Due to the wide open nature of wireless radio, many attacks could make the network insecure. The IEEE 802.11 standard has defined the following two basic security mechanisms for secure access to IEEE 802.11 networks:

- Entity authentication, including *open system* and *shared key* authentication
- Wired Equivalent Privacy (WEP)

Both have proved to be vulnerable.

In order to enhance security of IEEE 802.11, a new standard called IEEE 802.11i [2] is being developed. The objective of 802.11i is to enhance the 802.11 security aspects. In addition to introducing *key management and establishment*, it also defines *encryption* and *authentication* improvements. In order to manage security keys automatically, 802.11i defines key management and establishment algorithms, which are first introduced in the 802.11 standards. As conventional WEP is known to be vulnerable, 802.11i is specifying enhanced encryption to provide stronger privacy. 802.11i also incorporates IEEE 802.1X [3] as its authentication enhancement. 802.1X is now widely deployed in many IEEE 802 series standards with the Remote Authentication Dial In User Service (RADIUS, Internet Engineering Task Force, IETF, RFC 2865) as the authentication server. RADIUS could provide authentication, authorization, and accounting (AAA) services, but is still unlikely to resolve all security threats in wireless networks. Therefore, Diameter (IETF RFC 3588) is evolving to improve RADIUS to provide better security.

This article first discusses 802.1X with consideration of roaming users. It then reviews:

- Authentication enhancement
- Key management and establishment
- Encryption enhancement

as defined in the IEEE 802.11i draft.[1]

IEEE 802.1X

The IEEE 802.1X standard defines a mechanism for port-based network access control to provide compatible authentication and authorization mechanisms for devices interconnected by various 802 LANs. It could also be used to distribute security keys for 802.11 WLANs by enabling public key authentication and encryption between access points (APs) and mobile nodes (MNs). In 802.1X, the *port* represents the association between MN and AP. There are three main components in the 802.1X authentication system: *supplicant*, *authenticator*, and *authentication server* (AS). A supplicant is usually an MN requesting WLAN access. An authenticator represents the network access server (NAS). In 802.11 networks it is normally an AP. A RADIUS server is commonly used as the authentication server, although other types of AAA servers such as Diameter could also serve as the authentication server. In 802.11, the authentication server might be physically integrated into an AP.

THE IEEE 802.1X FRAMEWORK

As indicated in Fig. 1 [3], both supplicant and authenticator have a port access entity (PAE) that operates the algorithms and protocols associated with the authentication mechanisms. The authenticator PAE controls the authorized/unau-

[1] *IEEE 802.11i is still evolving; only its draft was available when this article was completed.*

1536-1284/05/$20.00 © 2005 IEEE

Once the supplicant is authenticated successfully, the controlled port in the authenticator is authorized. Packets from the supplicant will now go through the controlled port of the authenticator to backend networks to acquire the necessary services.

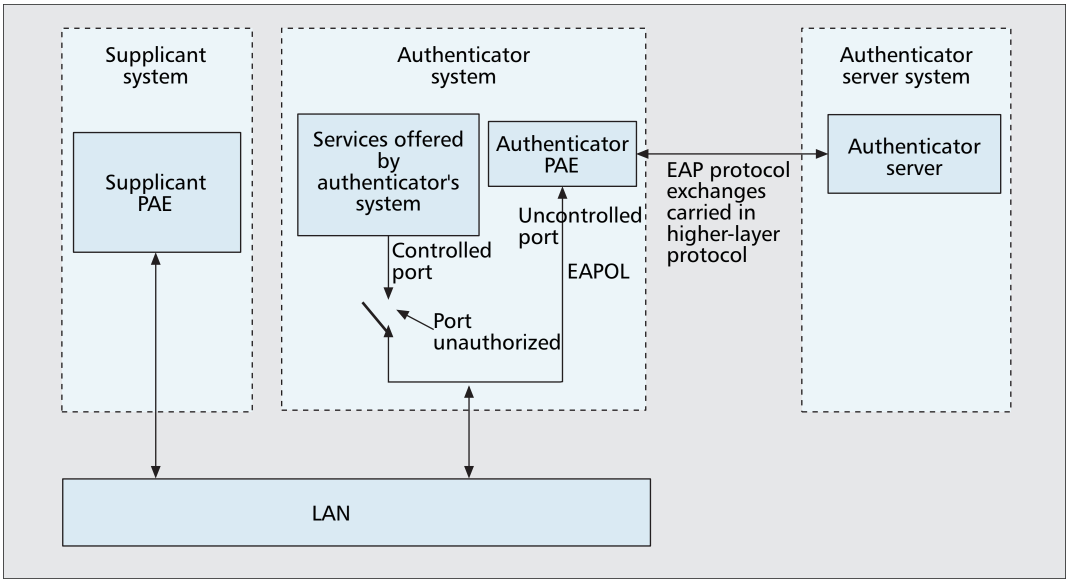


Figure 1. *IEEE 802.1X framework.*

thorized state of its *controlled port* depending on the outcome of the authentication processes. Before the supplicant is authenticated, the authenticator uses an *uncontrolled port* to communicate with the supplicant PAE. The authenticator will block all traffic except 802.1X messages before the supplicant is authenticated. The 802.1X standard leverages Extensible Authentication Protocol (EAP, IETF RFC 2284) to provide a number of authentication schemes, including Message Digest 5 (MD5, IETF RFC 1321), Transport Layer Security (TLS, IETF RFC 2716), Tunneled TLS (TTLS) [4], Protected Extensible Authentication Protocol (PEAP) [5], and smart cards such as EAP SIMs [6]. 802.1X also defines EAP over LANs (EAPOL) that encapsulates EAP messages between the supplicant and authenticator. EAP messages from the supplicant are relayed to the authentication server by the authenticator PAE. In order to let the RADIUS server authenticate users using EAP, the authenticator PAE encapsulates the same EAP messages in RADIUS packet format and sends them to the RADIUS server, assuming it has been adopted as the authentication server. The encapsulation is known as RADIUS-encapsulated EAP with the EAP-Message attribute, which is defined in RADIUS Extensions (IETF RFC 2869) for supporting EAP within RADIUS. Once the supplicant is authenticated successfully, the controlled port in the authenticator is authorized. Packets from the supplicant will now go through the controlled port of the authenticator to backend networks to acquire the necessary services.

Figure 2 depicts a typical 802.1X message exchange with both the supplicant PAE and authenticator PAE state transitions. The supplicant and authenticator PAE state machines are shown in Figs. 3 and 4, respectively. In Fig. 2 the digits associated with each flow do not represent the order of the flow. Instead, the digits in rectangles refer to the supplicant PAE state in Fig. 3, and the digits in circles refer to the authenticator PAE state in Fig. 4.

After the MN and AP complete the 802.11 association, both the MN and AP will transit to the CONNECTING state in their PAE state machines. However, the port is unauthorized at this moment, and the 802.1X authentication process just starts. As indicated in Fig. 2, the MN (supplicant) sends an EAPOL-Start frame to the AP (authenticator) to initialize the authentication process. When the AP receives EAPOL-Start, it replies with an EAP-Request/Identity to obtain the MN's identity. The MN transits to the ACQUIRED state when it receives EAP-Request/Identity from the AP. The MN then sends back an EAP-Response/Identity containing the MN's identity in response to the EAP-Request/Identity. If the AP receives the EAP-Response/Identity, the authenticator PAE state will transit to the AUTHENTICATING state. In the AUTHENTICATING state, the authenticator PAE encapsulates the EAP-Response/Identity message in RADIUS-Access-Request as an attribute (EAP-Message attribute) and sends it to the RADIUS server. In response to the RADIUS-Access-Request, the RADIUS server will challenge the MN by sending a RADIUS-Access-Challenge to the AP, which then relays the message in the form of EAP-Request/Auth to the MN. When the MN receives EAP-Request/Auth, the supplicant PAE state transits to the AUTHENTICATING state and replies with an EAP-Response/Auth, which is also relayed to the RADIUS server by the AP in the format of RADIUS-Access-Request. Depending on the authentication scheme, there might be some more message exchanges. The RADIUS server then determines whether the MN should be accepted or denied access to the network services. Figure 2 depicts three cases thereafter that are separated by dotted lines. If authentication succeeds, the RADIUS server sends a RADIUS-Access-Accept to the AP. On receipt of RADIUS-Access-Accept the authenticator PAE state transits to the AUTHENTICATED state and sends an EAP-Success

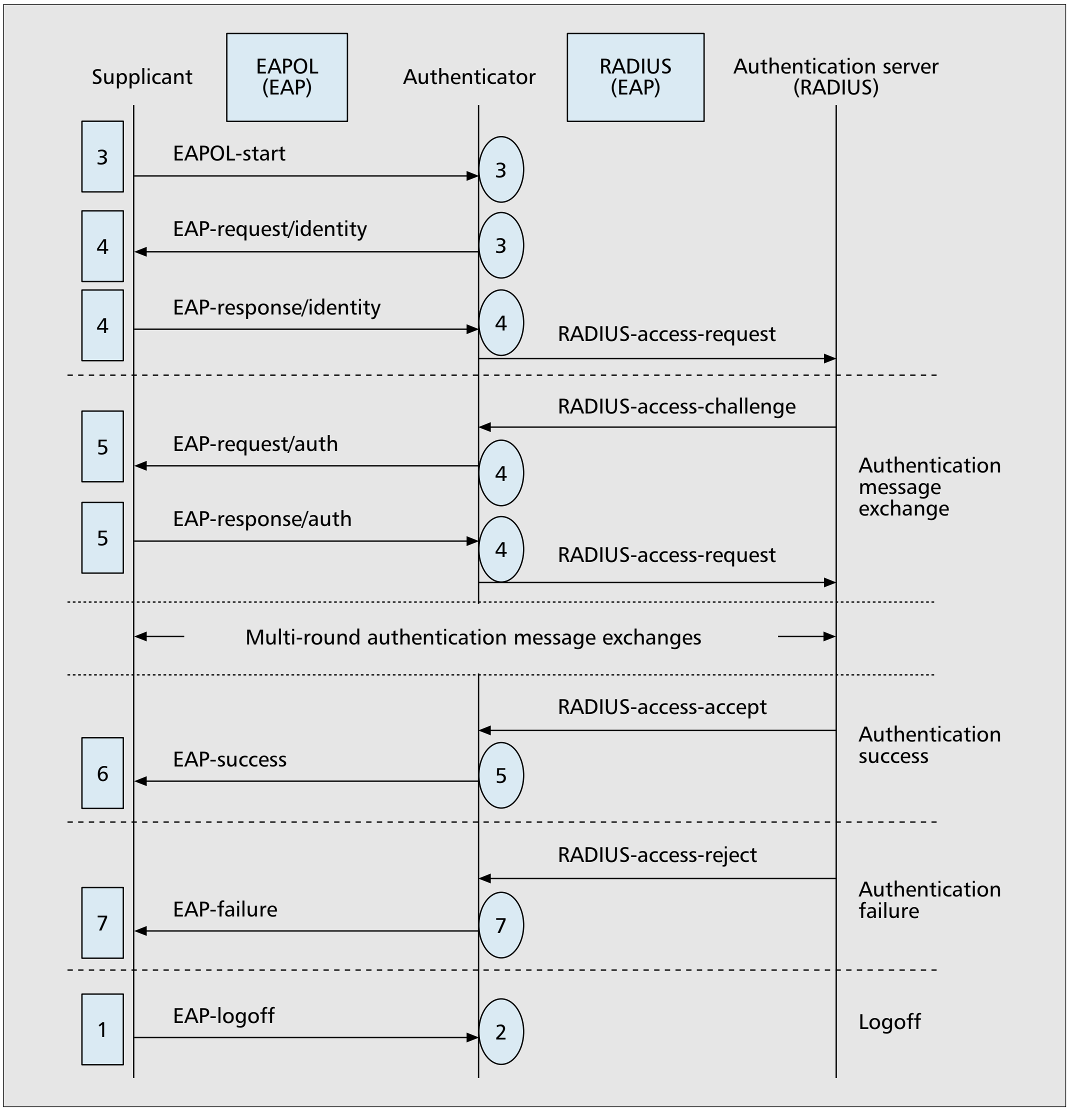


Figure 2. *Example flows of IEEE 802.1X message exchange.*

Administrators would not need to configure APs frequently when users are added or removed from the system. In addition, proxy RADIUS is also able to relay authentication messages to other RADIUS servers to authenticate the users whose profiles are maintained in other servers.

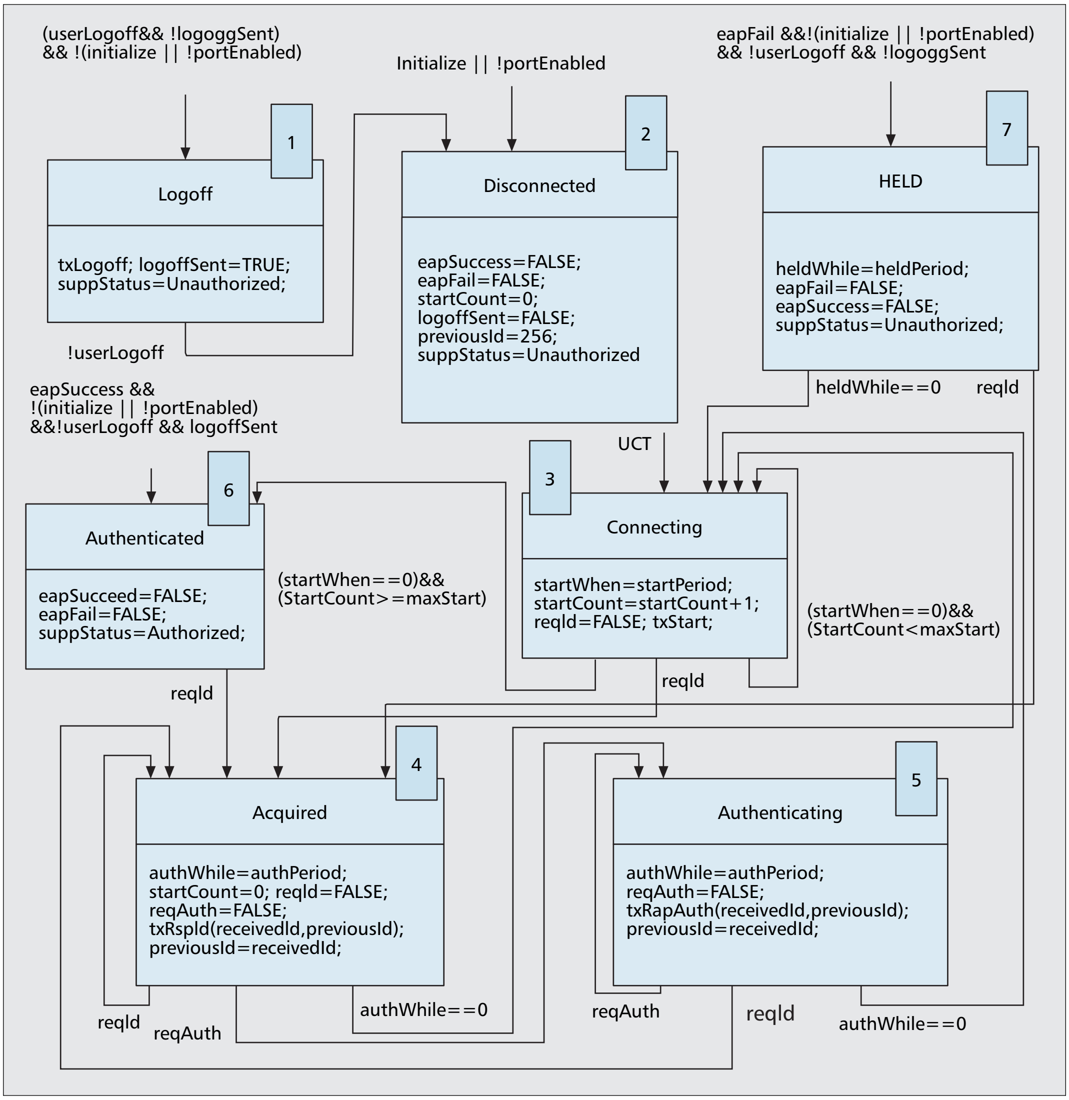
message to the MN to indicate the success of authentication. The controlled port of the AP shown in Fig. 1 is thus authorized. After receiving EAP-Success, the MN transits to the AUTHENTICATED state and the whole authentication process is completed. On the other hand, RADIUS-Access-Reject is sent by the RADIUS server and relayed to the MN by the AP in the message of EAP-Failure if the authentication fails. In this case, both the AP and MN transit to the HELD state, and the whole authentication fails. The controlled port is thus still unauthorized. If the MN is authenticated and wants to perform a logoff procedure from the current AP, the MN originates an EAPOL-Logoff packet to the AP. After that, the controlled port of the current AP transits to unauthorized state immediately. The supplicant and authenticator will transit to the LOGOFF state and DISCONNECTED state, respectively.

Usually there are several APs connected to the same authentication server. Because user profiles are stored in a centralized database, APs in the system could lighten the load of storing large amounts of data such as medium access control (MAC) addresses, user names, and passwords. Administrators would not need to configure APs frequently when users are added or removed from the system. In addition, proxy RADIUS is also able to relay authentication messages to other RADIUS servers to authenticate users with profiles maintained in other servers.

IEEE 802.1X with Roaming Users

This section discusses the mobility issues in 802.1X-enabled networks in terms of intrasubnet and intersubnet roaming, respectively. In intrasubnet roaming, an MN hands off to a different AP within the same IP subnet. It would need to re-authenticate with the new AP. Because the MN is still in the same IP subnet, it does not involve a change of IP address. Intersubnet handoff, however, would need to get a new IP address and keep the ongoing IP connections alive. This article uses Mobile IP (IETF RFC 3344) as an example to illustrate intersubnet handoff in IEEE 802.1X networks.

After the IEEE 802.1X authentication with the foreign AP is completed, as described earlier, both the MN and the AP are in the Authenticated state. Therefore, MN is able to send Mobile IP registration messages and perform subsequent Mobile IP procedures. Thus, IEEE 802.1X and Mobile IP could work independently.



(userLogoff&& !logoggSent) && !(initialize || !portEnabled)
Initialize || !portEnabled
eapFail &&!(initialize || !portEnabled) && !userLogoff && !logoggSent
1
Logoff
txLogoff; logoffSent=TRUE; suppStatus=Unauthorized;
2
Disconnected
eapSuccess=FALSE; eapFail=FALSE; startCount=0; logoffSent=FALSE; previousId=256; suppStatus=Unauthorized
7
HELD
heldWhile=heldPeriod; eapFail=FALSE; eapSuccess=FALSE; suppStatus=Unauthorized;
!userLogoff
heldWhile==0
reqId
eapSuccess && !(initialize || !portEnabled) &&!userLogoff && logoffSent
UCT
6
Authenticated
eapSucceed=FALSE; eapFail=FALSE; suppStatus=Authorized;
(startWhen==0)&& (StartCount>=maxStart)
3
Connecting
startWhen=startPeriod; startCount=startCount+1; reqId=FALSE; txStart;
(startWhen==0)&& (StartCount<maxStart)
reqId
reqId
4
Acquired
authWhile=authPeriod; startCount=0; reqId=FALSE; reqAuth=FALSE; txRspId(receivedId,previousId); previousId=receivedId;
5
Authenticating
authWhile=authPeriod; reqAuth=FALSE; txRapAuth(receivedId,previousId); previousId=receivedId;
reqId
reqAuth
authWhile==0
reqAuth
reqId
authWhile==0

Figure 3. *Port access entity (PAE) state machine in supplicant.*

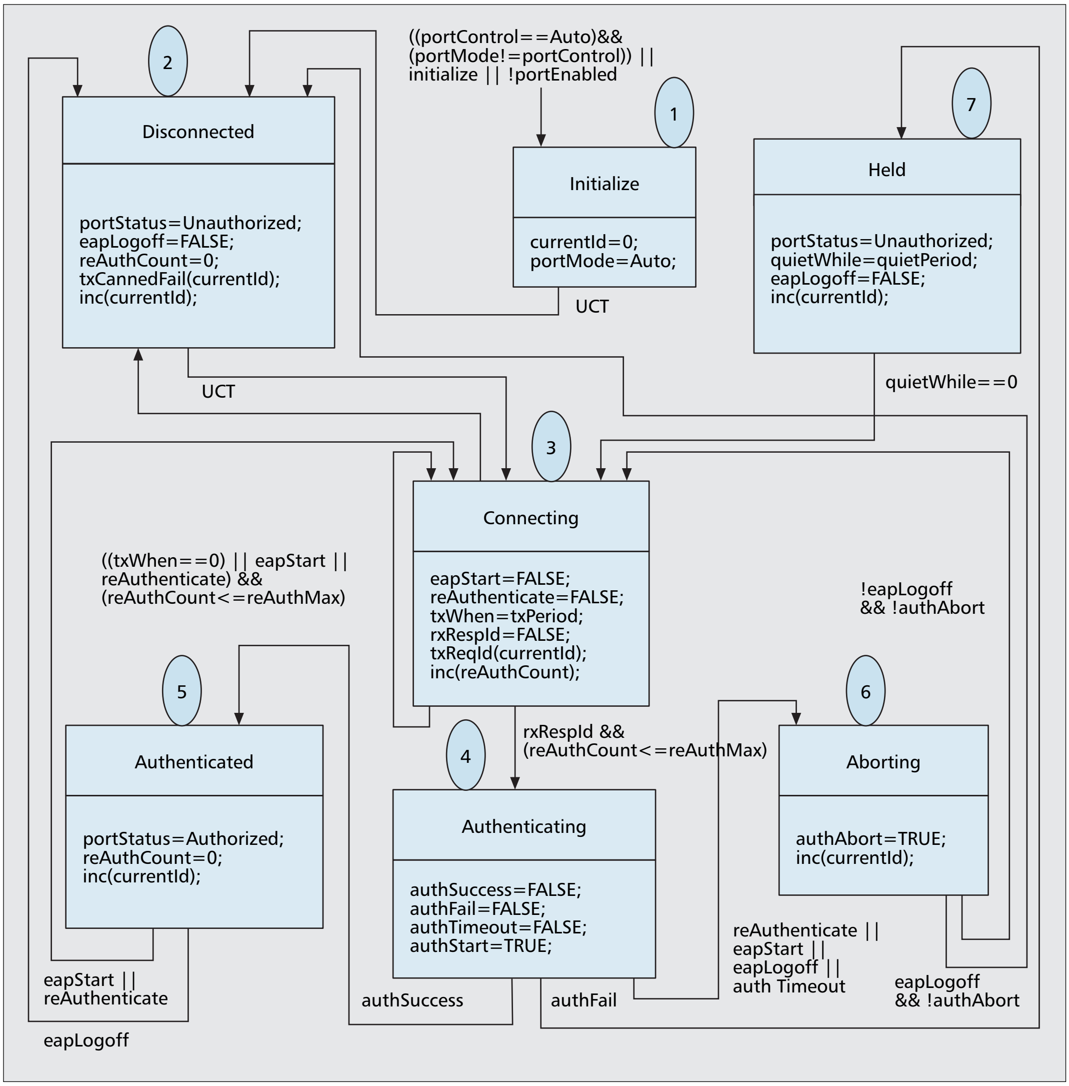
Intrasubnet roaming: According to IEEE 802.1X, an MN should re-authenticate with a new authenticator or authentication server when it roams to another 802.1X-enabled network. Figure 5 depicts a typical architecture in which an MN moves from a home AP to a foreign AP within the same IP subnet. Figure 5 also indicates the state transitions of the supplicant and authenticator during handoff.

Assuming the MN is already authenticated successfully with its home AP, both supplicant and authenticator PAE states are in the AUTHENTICATED state, as indicated in Fig. 5. When MN roams to a foreign AP, it first proceeds to do the 802.11 reassociation process with the foreign AP. The PAE state machine of the foreign AP will transit to the CONNECTING state once the reassociation is successfully completed. The foreign AP then sends an EAP-Request/Identity to the MN. After receiving the EAP-Request/Identity, the MN will transit from the AUTHENTICATED to the ACQUIRED state. It then sends back an EAP-Response/Identity. Afterwards, the message exchange follows the standard 802.1X authentication described earlier. Both the foreign AP and the MN will proceed to the AUTHENTICATED state eventually if the MN is authenticated successfully. Re-authentication then is completed.

Intersubnet roaming: When roaming to a new IP subnet, the MN's IP address in the old subnet is invalid in the new subnet. With only 802.1X re-authentication, the MN is able to reassociate with the foreign AP, but it cannot connect to the new IP subnet. Therefore, a protocol such as Mobile IP should be used to support mobility management in the IP layer. After 802.1X authentication with the foreign AP is completed, as described earlier, both the MN and AP are in the AUTHENTICATED state. Therefore, the MN is able to send Mobile IP registration messages and perform subsequent Mobile IP procedures. Thus, 802.1X and Mobile IP could work independently.

IEEE 802.1X WITH DIAMETER

Because network entities have increased in complexity, the deficiencies in RADIUS have been identified (IETF RFC 3127). Because IEEE 802.1X does not mandate the type of authentica-



■ **Figure 4.** *PAE state machine in authenticator.*

Because network entities have increased in complexity, the deficiencies in RADIUS have been identified. Because IEEE 802.1X does not mandate the type of authentication server, Diameter could be used to improve the security weaknesses in Radius.

tion server, Diameter (IETF RFC 3588) could be used to improve the security weaknesses in RADIUS.

Diameter provides a base protocol that can be extended from various applications to support AAA services. The Diameter-EAP application [7] could be employed for an 802.1X network. Usage of Diameter in an 802.1X system is similar to that of RADIUS. The major difference is in the replacement of RADIUS messages with Diameter messages. For intersubnet roaming, Diameter also specifies a Mobile IPv4 application [8].

IEEE 802.11i

The IEEE 802.11 Working Group has been working on MAC enhancement for several years. In May 2001, the MAC enhancement was split into different task groups. Task Group E (TGe) is responsible for quality of service (QoS). Task Group I (TGi) is working on security.

One of the major missions of 802.11 TGi is to define a robust security network (RSN). The definition of an RSN according to IEEE 802.11i draft [2] is *a security network that only allows the creation of robust security network associations* (RSNAs). That is, in an RSN the associations between all stations including APs are built on a strong association/authentication called an RSNA, which is also defined by the 802.11 TGi as: *an RSNA depends on 802.1X to transport its authentication services and deliver key management services*. A security association is defined as *the context providing the state (cryptographic keys, counters, sequence spaces, etc.) needed for correct operation of the IEEE 802.11 cipher suites*. RSNA includes a novel *four-way handshake* mechanism to provide robust session key management. By leveraging IEEE 802.1X, the four-way handshake, and the enhanced cryptographic algorithms, communication links in 802.11 wireless are securely protected.

The IEEE 802.11i Framework

The 802.11i standard defines two classes of security framework for 802.11 WLANs: RSN and pre-RSN. A station is considered RSN-capable equipment if it is capable of creating RSNAs. Otherwise, it is pre-RSN equipment. A network that only allows RSNA in associations with RSN-capable equipments is called an RSN security

The IEEE 802.11i standard defines two classes of security framework for IEEE 802.11 WLANs: RSN and pre-RSN. A station is considered to be RSN-capable equipment if it is capable of creating RSNAs. Otherwise, it is pre-RSN equipment.

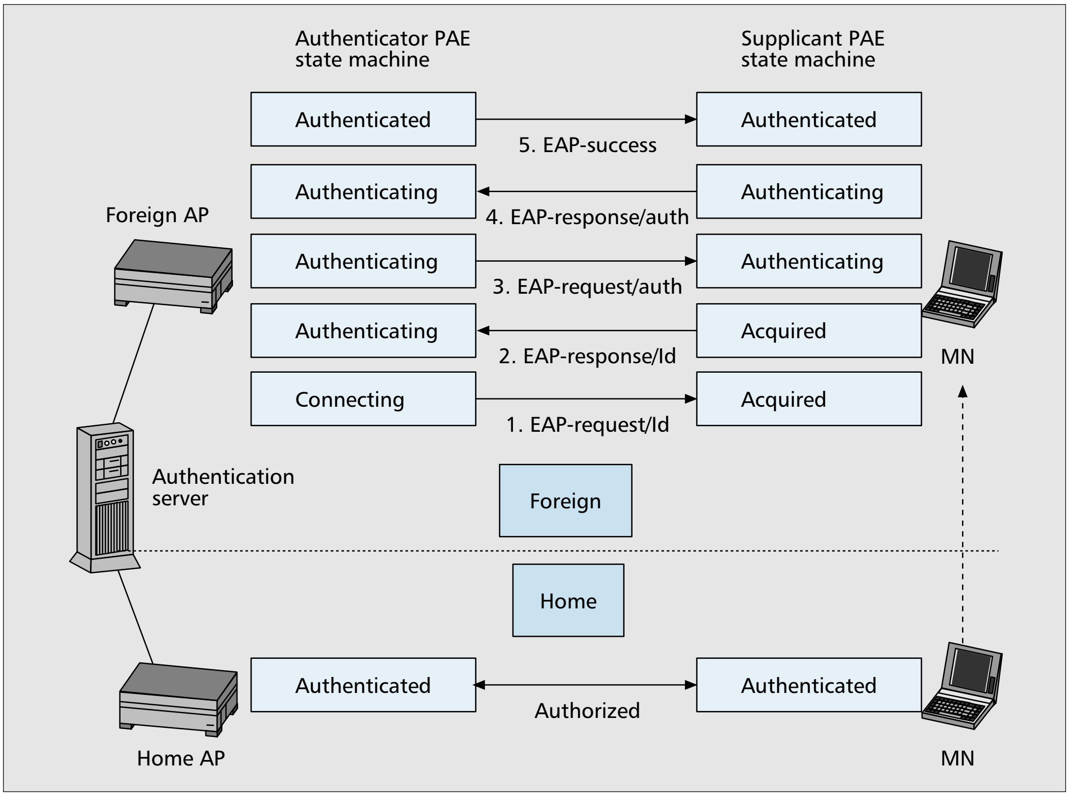


Figure 5. *State transition in roaming.*

framework. A network that allows pre-RSNA associations between stations is called a pre-RSN security framework. The major difference between RSNA and pre-RSNA is in the four-way handshake. If the four-way handshake is not included in the authentication/association procedures, stations are said to be pre-RSNA.

Pre-RSN: Pre-RSN security consists of two security subsystems:

- IEEE 802.11 entity authentication
- WEP

The IEEE 802.11 entity authentication includes *open system* authentication and *shared key* authentication. In open system authentication, there is no authentication algorithm. A station is authenticated simply based on its identity. Shared key authentication, on the other hand, authenticates a station based on a secret key known to the authentication requester and responder. It requires the privacy mechanism to be implemented in WEP.

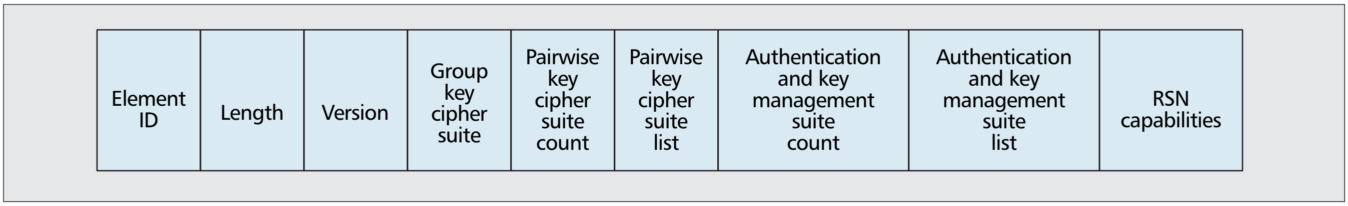
RSN: In addition to enhancing the security in pre-RSN, RSN security defines key management procedures for 802.11 networks. It also enhances the authentication and encryption in pre-RSN.

Authentication enhancement: 802.11i utilizes 802.1X for its authentication and key management services. It incorporates two components into the 802.11 architecture: the *802.1X port* and *authentication server* (AS). The 802.1X port represents the association between two peers. There is a one-to-one mapping between the 802.1X port and the association. As discussed earlier, an 802.1X port will allow general traffic to pass only when the authentication is successfully completed. The AS could be a standalone server or integrated into an AP. Although the protocol between the AS and AP is not specified by 802.11i, there should be a secure channel such as TLS (IETF RFC 2246) or IPSec (IETF RFC 2401) between the AP and AS. An EAP that supports mutual authentication should be used in an RSN. That is, the authentication requester and responder must be able to authenticate each other. EAP-MD5, for instance, cannot meet this requirement.

Key management and establishment: Two ways to support key distribution are introduced in 802.11i: *manual* and *automatic key management*. Manual key management requires the administrator to manually configure the key. Automatic key management is available only in an RSNA. It relies on 802.1X to support key management services. More specifically, a four-way handshake is used to establish each transient key for packet transmission.

Encryption enhancement: In order to enhance confidentiality, two advanced cryptographic algorithms are developed: Counter-Mode/CBC-MAC Protocol (CCMP) and Temporal Key Integrity Protocol (TKIP). In RSN, CCMP is mandatory. TKIP is optional and recommended only to patch pre-RSNA equipment.

IEEE 802.11i specifies an *RSN information element (RSN IE)* that carries RSN security information including RSN capabilities, authentication, and cipher key selectors. An RSN IE could be used to distinguish pre-RSN stations and RSN-capable stations. RSN-capable stations shall include the RSN IE in beacons, probe response, association and reassociation request, and the second and third messages of the four-way handshake. On the



Element ID	Length	Version	Group key cipher suite	Pairwise key cipher suite count	Pairwise key cipher suite list	Authentication and key management suite count	Authentication and key management suite list	RSN capabilities

■ Figure 6. *RSN IE format.*

other hand, there is no RSN IE in messages sent by pre-RSN stations. As shown in Fig. 6, the RSN IE contains a list of authentication and cipher selector fields for communications. The value of the *Element ID* field in Fig. 6 should always be 48 in decimal. The *Length* field indicates the number of octets in the information fields excluding the Element ID and Length fields. The *Version* field shows the version number of the RSNA protocol. The *Pairwise Key Cipher Suite Count* indicates the number of pairwise key cipher suites contained in the *Pairwise Key Cipher Suite List* field. The *Pairwise* field refers to two entities that are associated with each other. The *Pairwise Key Cipher Suite*, therefore, is the cipher suite being or to be associated between communicating peers. Similarly, the *Authentication and Key Management Suite Count* indicates the number of authentication and key management suites contained in the *Authentication and Key Management Suite List* field. In the *RSN Capabilities* field, the requested or advertised capabilities are filled in. By using this field, the receiver can know the security mechanisms the sender supports or is requesting.

Generally speaking, the RSN IE carries robust security information that indicates the authentication and cipher algorithms the communicating parties will use. Stations and APs can learn the security capabilities of the communicating peers and negotiate with each other by the RSN IE carried in an association/reassociation request, probe response, beacons, or other messages. Correspondent security procedures will then be executed.

Figure 7 shows an example of establishing an RSNA between a supplicant (station) and an authenticator (AP) in a basic service set (BSS). It assumes there is no preshared key. *Flows 1–6* are the 802.11 association and authentication process prior to attaching to the authenticator. During the process, security information and capabilities can be negotiated using the RSN IE. The authentication in *flows 3 and 4* refers to 802.11 open system authentication. After the 802.11 association is completed, the 802.1X authentication indicated in *flow 7* of Fig. 7 is initiated. EAP messages will be exchanged between the supplicant, authenticator, and authentication server, although the authentication server is not depicted in Fig. 7. If the supplicant and authentication server authenticate each other successfully, both independently generate a *pairwise master key* (PMK). The authentication server then transmits the PMK to the authenticator through a secure channel (e.g., IPsec or TLS). The four-way handshake then uses the PMK to derive and verify a *pairwise transient key (PTK)*. Therefore, the session key between the supplicant and authenticator is guaranteed to be fresh. After that, the group key handshake proceeds as indicated in *flow 9*. The group key handshake is used to generate and refresh the group key, which is shared between a group of stations and APs. By using this key, broadcast and multicast messages can be securely exchanged over the air.

The following sections review the authentication enhancement, key management and establishment, and encryption enhancement, respectively, defined in IEEE 802.11i.

Authentication Enhancement

In the original 802.11 standard, a station should first associate with an 802.11 AP. It then is able to access the WLAN service. An example of the process is shown by *flows 1–6* in Fig. 7. After finding an AP by receiving the Probe Response, the mobile station needs to proceed to the following two steps: *802.11 entity authentication* and *association*. Before associating with an AP, the station needs to accomplish 802.11 entity authentication. As discussed earlier, there are two authentication schemes: *open system* and *shared key* authentication. Open system authentication allows a station to be authenticated without having a correct WEP key. There are two message exchanges. The first message sending from supplicant (mobile station) to authenticator (AP) is used to expose the identity of the station. Based on the identity, the authentication result is sent from the authenticator back to the station. There is no authentication algorithm. In shared key authentication, there are four message exchanges. The first message containing the identity of the station is delivered from the station to the AP. The AP will then send a challenge packet to the mobile station. The mobile station is required to encrypt the challenge packet using the shared WEP key and send the encrypted result back to the AP. If the challenge packet is encrypted correctly, the supplicant is authenticated successfully. The authentication result is sent to the station in the fourth message. If the station is authenticated successfully, it proceeds to the 802.11 association. The mobile station should transmit an Association Request to the AP. The AP then sends back an Association Response to the station.

Shared key authentication in 802.11 is not adopted by 802.11i. Instead, it incorporates 802.1X as the authentication solution for the RSN. As depicted in Fig. 7, 802.1X is performed after 802.11 open system authentication and association. IEEE 802.1X provides a *port-based* network access control mechanism to protect against unauthorized access. Details of 802.1X

IEEE 802.11i also supports pre-authentication. A station could pre-authenticate with an AP before roaming. A station could initiate an EAPOL-Start message through the serving AP to inform the new AP to start the IEEE 802.1X authentication, thus reducing handoff latency.

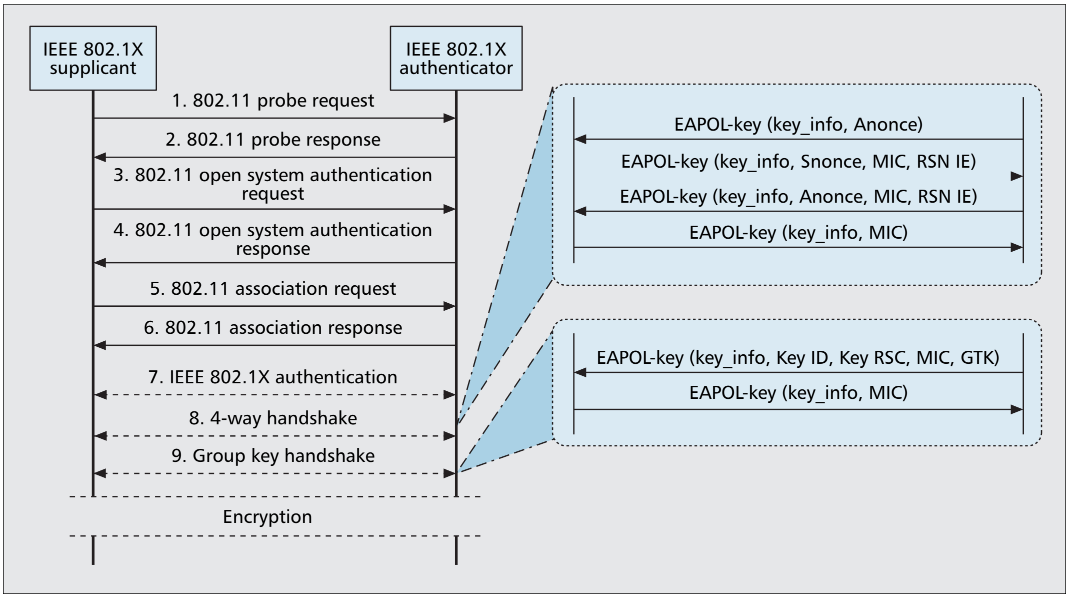

IEEE 802.1X supplicant
IEEE 802.1X authenticator
1. 802.11 probe request
2. 802.11 probe response
3. 802.11 open system authentication request
4. 802.11 open system authentication response
5. 802.11 association request
6. 802.11 association response
7. IEEE 802.1X authentication
8. 4-way handshake
9. Group key handshake
Encryption
EAPOL-key (key_info, Anonce)
EAPOL-key (key_info, Snonce, MIC, RSN IE)
EAPOL-key (key_info, Anonce, MIC, RSN IE)
EAPOL-key (key_info, MIC)
EAPOL-key (key_info, Key ID, Key RSC, MIC, GTK)
EAPOL-key (key_info, MIC)

Figure 7. *Example flows of RSNA establishment.*

have been discussed. Please note that Fig. 7 depicts the establishment of an RSN. The two message exchanges of *flows 3 and 4* for open system authentication should not be replaced by the four message exchanges of shared key authentication.

IEEE 802.11i also specifies a more robust security framework utilizing 802.1X, a four-way handshake, and a group key handshake to authenticate and authorize stations. The four-way and group key handshakes are described in the next section. After the station is authenticated successfully, the cryptographic keys are configured as well. The station is thus able to send and receive unicast and broadcast frames in a secure manner. Moreover, IEEE 802.11i also supports pre-authentication. A station could pre-authenticate with an AP before roaming. A station could initiate an EAPOL-Start message through the serving AP to inform the new AP to start the IEEE 802.1X authentication, thus reducing handoff latency.

Key Management and Establishment

This section discusses the four-way handshake and group key handshake, respectively.

Four-way handshake: The RSNA defines a 4-way handshake to perform several functions such as confirming the liveness of the communicating stations, guaranteeing the freshness of the session key, installing the cryptographic key, and confirming the installation of the key. The four-way handshake is achieved using 802.1X. Specifically, messages exchanged in the four-way handshake are in the *EAPOL-Key* format. EAPOL-Key is defined in IEEE 802.1X and could be used to exchange cryptographic keying information. Figure 7 depicts the message flows of four-way handshake.

In a four-way handshake, the authenticator first sends out a message to the supplicant. The first message contains key information and an *Anonce*. An Anonce is a nonce, which is also called *key material*, generated by the authenticator. A nonce essentially is a random or pseudo-random value. In the four-way handshake, Anonce will never be reused. Therefore, the RSN is safe against replay attack.

After receiving the first message, the supplicant validates the message by checking the Replay Counter field in the message. The Replay Counter is a sequence number, which shall be incremented by each EAPOL-Key message. If the Replay Counter is less than or equal to the value kept in the supplicant, the supplicant discards the message. Otherwise, the supplicant generates a new nonce called Snonce. By using an algorithm called *Pseudo-Random Function* (PRF) with Anonce, Snonce, PMK, and other information as inputs, the supplicant derives a *pairwise transient key* (PTK). The supplicant then sends back the second message containing key information, Snonce, the supplicant's RSN IE, and the *message integrity code* (MIC) back to the authenticator. The MIC is a cryptographic digest used to provide integrity service.

Upon receiving the second message, the authenticator validates the message by checking the replay counter. The process is similar to that in the supplicant when receiving the first message. It then derives the PTK if the second message is validated. Because the authenticator uses the same algorithm and the same inputs, the PTK derived by the authenticator will be the same as the one in the supplicant. The authenticator also verifies the MIC. The packet is discarded silently if the MIC is not valid. In addition, the authenticator compares the received RSN IE bitwise with the one contained in the Association/Reassociation Request received earlier from the supplicant. If these are not exactly identical, the association is terminated. Otherwise, the authenticator sends the third message to the supplicant. The third message includes the key information, Anonce, MIC, and the authenticator's RSN IE.

On reception of the third message, the supplicant first verifies the message by checking the

Replay Counter and Anonce fields. It then compares the RSN IE with the one received previously in the Beacon or Probe Response. The supplicant will disassociate from this AP if the RSN IEs are different. If the RSN IE is correct, the supplicant further checks the MIC. The supplicant sends back the fourth message to the authenticator if the MIC is valid. The fourth message comprises the key information and MIC.

Once the fourth message is received by the authenticator, the authenticator checks the replay counter as before. The authenticator then verifies the MIC if the replay counter is valid. The four-way handshake is completed if the MIC is valid. The fourth message is used to acknowledge to the authenticator that the supplicant has installed the PTK. The PTK is only known by the supplicant and authenticator. It is used as a key to encrypt data.

Group key handshake: The RSNA also defines a group key handshake that enables the authenticator to deliver the *group transient key* (GTK) to the supplicant so that the supplicant can receive broadcast messages. Like the four-way handshake, the messages exchanged in the group key handshake also use the EAPOL-Key format. Figure 7 depicts the message flows of the group key handshake.

As indicated in Fig. 7, the group key handshake is performed after the four-way handshake. The authenticator first sends a message that includes key information, MIC, and GTK to the supplicant. The GTK is encrypted using the EAPOL-Key encryption key (KEK), and the MIC is computed over the body of the EAPOL-Key message using the EAPOL-Key confirmation key (KCK). Both the KEK and KCK are parts of the PTK. Upon receiving the message, the supplicant checks the replay counter. It then uses the KCK to verify the MIC. The supplicant will decrypt the GTK with KEK if the replay counter and MIC are valid. The supplicant then configures the GTK into its 802.11 MAC. In addition, it also replies with a message that includes key information and the MIC to the authenticator. Similarly, the authenticator validates the replay counter and MIC.

Encryption Enhancement

The WEP algorithm is primarily used to protect wireless communications from eavesdropping. It is also capable of preventing unauthorized access. Thus, WEP provides both confidentiality and integrity services. WEP relies on the secret key shared between a mobile station and an AP. WEP uses the RC4 stream cipher. Before sending data, the sender needs to compute the integrity check value (ICV) with a cyclic redundancy check-32 (CRC-32) algorithm. The sender then encrypts the data frame and ICV. The ciphertext consists of the encrypted data and ICV. The WEP bit in the MAC header should be set as well. When the receiver receives a MAC frame with the WEP bit set, it will use the shared WEP key to decrypt the payload.

It is known that WEP has been cracked. WEP is vulnerable because of the short length of the initialization vector (IV) and the static secret key. IVs are used to concatenate the shared secret key to produce different RC4 key sequences for each packet. The IV is generated at random and is also included in the packet. With only 24 bits of IV, WEP will eventually use the same IV for different data packets, which is known as *IV collision*. When collecting enough packets based on the same IV, an attacker could find out the shared value (i.e. the key sequence or secret key) among the communicating parties. The static nature of the shared secret key causes another security issue. Because the original 802.11 does not provide any mechanism for key management, the system administrator and a user in general use the same shared key for a long period of time. The same WEP key is even shared between all stations in the same BSS or extended service set (ESS). This nature provides attackers plenty of time to monitor and hack into WEP-enabled WLANs.

To amend the flaws in WEP, 802.11i has developed a better algorithm, TKIP, as an interim standard. TKIP, initially referred to as WEP2, is also based on RC4 encryption. However, it is implemented in a different way that addresses the vulnerabilities of WEP. TKIP defines a temporal key (TK), a 128-bit secret key shared by encryptor and decryptor. The TK might be common among many parties. The encryptor and decryptor must use the RC4 stream cipher. Each party must ensure that no IV value is used more than once with the current TK. The IV is extended to a 48-bit counter starting with zero. Implementations must ensure that the TK is updated before the full 48-bit IV space is exhausted. TKIP also employs a packet *sequence counter* to order the MAC protocol data unit (MPDU). The receiver should drop out-of-order MPDUs. It could thus protect against replay attack. Moreover, TKIP combines the TK with the client's MAC address and then adds a relatively large 16-bit IV to produce the key to encrypt data. This ensures that each computer uses a different key for encryption. TKIP basically applies the same encryption as WEP, but it utilizes the IEEE 802.1X EAPOL protocol to refresh the temporal keys to prevent key reuse. This provides dynamic key distribution that significantly enhances the security provided by WEP. TKIP can be adapted into older IEEE 802.11 products by just upgrading through relatively simple firmware patches. This is especially favorable for vendors. In addition, equipment that only supports the old WEP will still be capable of interoperating with TKIP-enabled devices. TKIP is optional in 802.11i.

Because TKIP uses the same RC4 encryption as WEP, it is considered as a short-term solution for WLAN security. In addition to TKIP, 802.11i also defines CCMP as a long-term solution. CCMP employs the stronger encryption of the Advanced Encryption Algorithm (AES), which uses the CCM mode (IETF RFC 3610) with a 128-bit key and a 128-bit block size of operation. The CCM mode combines counter-mode (CTR) and cipher block chaining message authentication code (CBC-MAC). CTR is used to encrypt the payload and the MIC to provide confidentiality service. CBC-MAC computes the MIC to provide authentication and integrity services. CCM requires a fresh TK for every session and

RSNA also defines a group key handshake that enables the authenticator to deliver the group transient key to the supplicant so that the supplicant could receive broadcast messages. Like the 4-way handshake, the messages exchanged in the group key handshake also use the EAPOL-Key format.

> Although the emerging IEEE 802.11i standard could potentially improve the security services in today's IEEE 802.11 wireless LANs, it is expected that more work is needed to develop a more secure WLAN environment.

needs to refresh the TK when the packet number (PN) is repeated. The PN is incremented for each MPDU and can be used to prevent a replay attack with the receiver's replay counter. The PN and key ID are encoded in the CCMP header. Although CCMP could provide much stronger security services, it requires additional hardware (co-processor) to improve encryption performance. Therefore, older 802.11 hardware will not be upgradeable in many cases. CCMP is mandatory in 802.11i.

Summary

Due to the nature of wireless media, unauthorized access is easier than in wired networks. Although IEEE 802.11 is proverbially insecure, it is widely deployed. IEEE 802.11i, therefore, is aiming to enhance security in IEEE 802.11 networks.

This article presents the security enhancements in encryption and authentication developed by IEEE 802.11i. In addition, the newly introduced key management in 802.11i is also discussed. The RSN is expected to fulfill many security requirements. However, the coordination of whole systems is still a challenge. It involves intercompatibility between different domains, as well as backward compatibility between new and old systems. The usability of new software and hardware will also determine the acceptance of the new standard by end users. Although the emerging IEEE 802.11i standard could potentially improve security services in today's IEEE 802.11 wireless LANs, it is expected that more work is needed to develop a more secure WLAN environment.

Acknowledgments

This work was sponsored in part by the Ministry of Education (MOE) Program for Promoting Academic Excellence of Universities under the grant number 89-E-FA04-1-4, National Science Council under grant numbers 91-2219-E-007-023 and 92-2213-E-007-019, and the Industrial Technology Research Institute under contracts T1-92019-3 and 2F-92050-4.

References

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

[2] IEEE Std 802.11i/D4.1, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Security Enhancements," July 2003.

[3] IEEE Std 802.1X-2001, "Port-Based Network Access Control," June 2001.

[4] P. Funk and S. Blake-Wilson, "EAP Tunneled TLS Authentication Protocol (EAP-TTLS)," IETF Internet draft, draft-ietf-pppext-eap-ttls-03.txt, Aug. 2003, work in progress.

[5] A. Palekar *et al.*, "Protected EAP Protocol (PEAP) Version 2," IETF Internet draft, draft-josefsson-pppext-eap-tls-eap-07.txt, Oct. 2003, work in progress.

[6] H. Haverinen and J. Salowey, "EAP SIM Authentication," IETF Internet draft, draft-haverinen-pppext-eap-sim-12.txt, Oct. 2003, work in progress.

[7] P. Eronen, T. Hiller, and G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application," IETF Internet draft, draft-ietf-aaa-eap-04.txt, Feb. 2004, work in progress.

[8] P. R. Calhoun *et al.*, "Diameter Mobile IPv4 Application," IETF Internet draft, draft-ietf-aaa-diameter-mobileip-16.txt, Feb. 2004, work in progress.

Biographies

JYH-CHENG CHEN [SM] (jcchen@cs.nthu.edu.tw) is an associate professor in the Department of Computer Science and the Institute of Communications Engineering, National Tsing Hua University, Hsinchu, Taiwan. Prior to joining National Tsing Hua University as an assistant professor, he was a research scientist at Telcordia Technologies (formerly Bellcore), Morristown, New Jersey, from August 1998 to August 2001. He received his Ph.D. degree from the State University of New York at Buffalo in 1998. He is co-author of the book *IP-Based Next-Generation Wireless Networks* published by Wiley in January 2004.

MING-CHIA JIANG (jmc@wire.cs.nthu.edu.tw) received his B.S. degree in computer science and information engineering from National Central University, Chungli, Taiwan in 2001, and his M.S. degree in computer science from National Tsing Hua University in 2003. He is now with Ambit Mircosystems Corporation, Hsinchu, Taiwan.

YI-WEN LIU (timl@wire.cs.nthu.edu.tw) received his B.S. degree from the Department of Computer Science, National Tsing Hua Universit in 2002. He is now an M.S. student in the same department.