ARUBA™
The Mobile Edge Company

App Note

Wireless Client Security Demystified

Abstract
Wireless client security today is an alphabet soup of acronyms, standards and protocols. This document provides an overview of the most common and popular wireless security techniques. This includes:

- ♦ 802.11i
- ♦ Wireless Protected Access (WPA)
- ♦ Wireless Protected Access 2 (WPA2)
- ♦ 802.1x
- ♦ TKIP
- ♦ AES
- ♦ EAP (PEAP, EAP-TLS, EAP-TTLS, LEAP)
- ♦ xSec

Recommended Reading
The following pre-requisite documentation is highly recommended before reading this document:

- ♦ *802.11 Wireless Networks: The Definitive Guide* - Matthew S. Gast (O'Reilly)
- ♦ *Real 802.11 Security: Wi-Fi Protected Access and 802.11i* – Jon Edney, William A. Arbaugh (Addison Wesley)

Table of Contents

© 2005 Aruba Networks

Wireless Client Security Overview

Overview

There are many different protocols, standards and acronyms associated with wireless client security today. This section attempts to list the most common and useful.

The fundamentals of security

Before we can talk about wireless security as an entire organic whole, it is important to discuss the different types of protocols available. These are the building blocks that are used to create security standards such as 802.11i and Wireless Protected Access (WPA). All of the protocols mentioned in this document fall into one of the following categories:

- **Encryption** – privacy, i.e. protecting your data from eavesdroppers
- **Authentication** – proving who you are
- **Access control** – restricting network resources based on a given criteria

Before we can fully discuss the different protocols, it is important to understand whether the protocol is an authentication, encryption or access control protocol and how it fits into a greater 802.11 security framework.

Most wireless security schemes will employ at least one authentication and one encryption protocol. Some include access control as well. This ensures wireless devices are authenticated (we know who they are), their data is encrypted (no-one can eavesdrop) and their ability to reach network resources is controlled. This last can be very important since not all wireless users or devices are trusted equally.

Encryption

Because wireless networks broadcast messages using radios that operate in the unlicensed band of the RF spectrum, any device equipped with a receiver powerful enough can hear it. This makes Wi-Fi (802.11) particularly susceptible to eavesdropping, data compromises, and man in the middle attacks. The following is a list of commonly used encryption protocols that attempt to solve the problem of *confidentiality*[1]:

- Wired Equivalent Privacy (WEP)
- Temporal Key Integrity Protocol (TKIP)
- Advanced Encryption Standard (AES-CCMP)

All of these protocols are used to encrypt data before it is transmitted over the wireless network. These aid in foiling the attempts of a malicious attacker form intercepting and interpreting the information. These protocols are explained more fully later in this document.

[1] Confidentiality is used to refer to the privacy and integrity of information

© 2005 Aruba Networks

Wireless Client Security Overview continued

Authentication

Authentication ensures that only users or devices that have identified themselves are allowed on the network. This is important from a security standpoint – data does not need to be encrypted, but we should ensure only people we know are allowed to connect. Authentication is focused on identification – who or what is making the request. The following is a list of authentication protocols and mechanisms commonly used:

- SSID (Service Set Identifier)
- Extensible Authentication Protocol (EAP)

The original 802.11 standard counted the ability of a wireless device to know the name of the SSID as a form of authentication. This is based on the fact that a wireless access point can “hide” an SSID by refusing to broadcast it or respond with the name when a client probes it for a list of available SSIDs. This document will not discuss this authentication mechanism further, other than to note that it can be useful as a means of separating wireless users and traffic. However it should always be paired with other, stronger, security mechanisms.

The Extensible Authentication Protocol (EAP) validates user credentials against an authentication server. EAP, in its many variations and flavors, is a very popular protocol for wireless authentication.

Access control protocols

Access control can take many forms and is used to restrict network resource access. This is very useful for *user differentiation.* For example, a guest user on a wireless network typically only needs Internet access and does not require (and should not have) access to internal network servers and resources. Access control can take many forms and multiple types can be used together for even greater flexibility and functionality:

- Access Control Lists/Firewalls
- 802.1x

Access Control Lists (ACLs) and firewalls are a common means of controlling network access. Most networks today employ some form of firewall – typically in a De-Militarized Zone (DMZ). However it is worth noting that because anyone with a radio receiver can intercept wireless data, a firewall that is only located in the DMZ is not useful. To be truly effective, a firewall needs to be as close to the wireless packets as possible for the greatest flexibility and control.

© 2005 Aruba Networks

Wireless Client Security Overview continued

Beyond simple ACLs, a stateful firewall can not only provide stateful inspection and access control on a per-packet and per-flow basis, it can also aid in user differentiation and classification. A wireless-aware firewall can also participate in WLAN control mechanisms – for example, if a user or device attempts to access a resource they are forbidden by policy, the firewall can communicate with the WLAN and blacklist the client. This is a very sophisticated level of network access control that works very well either on its own or coupled with other techniques such as 802.1x.

802.1x is a network access protocol designed to determine whether a device should be allowed to connect to the network – even before the device is given Layer 2 functionality (DHCP, etc.). It is almost always deployed with other protocols such as EAP and TKIP to provide a complete authentication, encryption and network access security model.

Putting it all together

Now that we have covered the basic protocols and language of wireless security, let's talk about entire wireless security eco-systems. There are three very popular standards today that are widely referred to:

- 802.11i
- xSec
- Wi-Fi Protected Access (WPA)
- Wi-Fi Protected Access 2.0 (WPA2)

802.11i

IEEE 802.11i is an amendment to the original 802.11 standard. It improves upon the original security mechanisms and is intended as the next-generation definition for Wi-Fi networks. 802.11i is the direct successor of Wired Equivalent Privacy (WEP), which was shown to have severe security weaknesses.

802.11i introduces the concept of a Robust Security Network (RSN). In RSN wireless devices need to support additional capabilities. Some of these new capabilities include AES-CCMP – a more secure encryption algorithm that is a replacement for TKIP.

These new features can require new hardware and software drivers; making a fully compliant RSN network incompatible with existing WEP equipment.

Wireless Client Security Overview continued

xSec

xSec was created to address a requirement by the US government for a FIPS (Federal Information Processing Standard)-certifiable version of the 802.11i standard. The xSec protocol was co-developed by Aruba Networks and Funk Software.

xSec is unique in that it is available for both **wired** as well as wireless clients. With this, organizations can offer the exact same security protocols and standards regardless of the connection medium.

WPA

When WEP, the security in the original IEEE 802.11 standard, was broken, the IEEE immediately began work on a new, next-generation wireless standard called 802.11i. However, like all standards, it took some time for the 802.11i standard to be completed and ratified. In the meantime, wireless networks everywhere were vulnerable to attack.

To fill this gap, a group of companies interested in promoting the growth of wireless LANs was created. This group, called the Wi-Fi Alliance, created an interim, de facto standard called Wi-Fi Protected Access (WPA) that was based on an early draft of the 802.11i standard.

WPA implements the majority of the 802.11i standard and is designed to work with all wireless network interface cards, although not necessarily all first generation access points.

WPA2

WPA2 is the Wi-Fi Alliance name for its own implementation of the mandatory parts of the 802.11i standard. It is available today from many vendors and hardware manufacturers. The Wi-Fi Alliance offers a certification process which guarantees interoperability and adherence to common standards requirements.

Both WPA and WPA2 provide good security, with these significant differences:

- WPA is fully backwards compatible with all wireless NICs, but not necessarily all WLAN systems (APs, controllers, etc.)
- WPA2 is often not supported fully by older equipment
- In a mixed WPA or WPA2 environment often one or the other must be chosen – however some vendors offer a mixed mode that allows both to operate at the same time
- WPA2 offers both AES-CCMP and TKIP for encryption, WPA offers TKIP only

WEP Encryption

Overview

The very first wireless security scheme was the Wired Equivalent Privacy (WEP) protocol. It was included in the original IEEE 802.11 standard and was intended to secure data transmitted over 802.11 networks.

How WEP keys work

WEP depends upon a static pre-shared key that is configured on every wireless client before it can connect to the WLAN. The client passes this key to the wireless access point (AP). This key is then used to encrypt the communications between the client and AP. WEP uses the stream cipher RC4 in combination with CRC-32 checksums for integrity. WEP keys can be of varying lengths and are described as *bits*.

Standard 64-bit WEP uses a 40-bit key, to which a 24-bit initialization vector (IV) is concatenated to form the RC4 traffic key. A popular alternative is the 128-bit WEP protocol which uses a 104-bit key size. A 128-bit WEP key is almost always entered by users as a string of 26 Hexadecimal (Hex) characters (0-9 and A-F). Each character represents 4 bits of the key. 4 * 26 = 104 bits. Adding the 24-bit IV creates what is then called a *128-bit WEP key*. A 256-bit WEP system is available and even larger sizes could theoretically be supported. Unfortunately, key size has turned out to not be the major security limitation in WEP.

WEP security

WEP was intended to provide comparable confidentiality to a traditional wired network, hence the name. However, the way the RC4 cipher and IV bits by WEP has been shown to be flawed. Numerous attacks have been published which show how to recover WEP keys passively within minutes or hours.

The major drawbacks of WEP include:

- **Static keys are easy to break** – the IV value is too short and keys are constructed such that they are vulnerable to reuse or weak key attacks; breaking the key compromises every wireless device and all traffic
- **Message integrity** – there is no effective detection of message tampering or replay
- **No key management** – the protocol directly uses the master key with all clients. This static key must be shared somehow with all clients before they are allowed to associate with the WLAN. WEP makes no provision for sending out new keys
- **No key rotation/updates** – there is no automated mechanism for changing keys. Often, WEP networks go for months or years with the same key simply because of the inconvenience of manual key updates

© 2005 Aruba Networks

WEP Encryption continued

Some vendors have offered a variation of WEP that attempts to solve some of these issues. This is often referred to as *Dynamic WEP*. Dynamic WEP is a typically an implementation of the 802.1x protocol that uses WEP for the encryption. Thus, the 802.1x protocol is used for key management and distribution; however, once the key is established it functions like static WEP. 802.1x is covered later in this document.

When to use WEP

Although WEP has been widely discredited as a wireless security mechanism, it does have its uses even today. These include:

- Simple devices that do not support more advanced protocols, e.g. wireless phones or older wireless client devices[2]
- Home use

Whenever possible, dynamic WEP is preferable to static WEP – however not all devices support this.

[2] In this case, it is strongly recommended that static WEP devices be deployed in conjunction with another technology such as a VPN client or MAC authentication.

TKIP Encryption

Overview

TKIP is an encryption protocol used in 802.11 networks and is part of the IEEE 802.11i standard. TKIP was designed as a successor to WEP that could be implemented without replacing legacy hardware. This was necessary because the breaking of WEP keys had left Wi-Fi networks vulnerable and without a viable link-layer security solution. It was deemed that solving this problem could not wait for new hardware to become available.

How TKIP works

TKIP was designed to work in a very similar way to WEP. This was by design and intention. TKIP is a suite of algorithms that works as a "wrapper" to WEP, which allows users of legacy WLAN equipment to upgrade to TKIP without replacing hardware. TKIP uses the original WEP programming and concept of a pre-shared key but "wraps" additional code at the beginning and end to encapsulate and modify it. Like WEP, TKIP uses the RC4 stream encryption algorithm as its basis. The new protocol, however, encrypts each data packet with a unique encryption key. These keys are also much stronger cryptographically than those of its predecessor, WEP.

TKIP includes four additional algorithms:

- A cryptographic message integrity check
- An initialization-vector (IV) sequencing mechanism that hashes the IV bits and changes the rules for the size, reuse and values of the IV[3]
- Change the encryption key for every frame
- A mechanism to distribute and change the broadcast keys automatically – this creates a hierarchy of keys and protects the master key

[3] These changes effectively address WEP’s weaknesses that made it vulnerable to key reuse and replay attacks

TKIP Encryption continued

TKIP security

TKIP is superior to WEP in many ways and solves many of its problems. However TKIP does have some drawbacks. These includes:

- **No key rotation/updates** – the same static pre-shared master key (PSK) must be shared somehow with all clients before they are allowed to associate with the WLAN. The TKIP protocol makes no provision for automatically generating new master keys
- **Weak encryption algorithm** – TKIP still relies on the RC4 cipher which is not considered the highest level of security available. RC4 is not approved for government use

TKIP has been shown to be a cryptographically sound protocol that has no known major weaknesses today. However, the very security conscious may want to consider an entirely new protocol that does not rely on backwards compatibility and thus could be made stronger and incorporate new features.

The 802.11i standard specifies the Advanced Encryption Standard (AES) in addition to TKIP. AES offers a higher level of security and is approved for government use. As organizations replace older wireless equipment, AES is expected to become the accepted encryption standard for WLAN security.

AES-CCMP Encryption

Overview

The 802.11i standard also includes the Advanced Encryption Standard (AES) crypto cipher and an accompanying operating mode, or encryption algorithm, Counter-Mode/CBC MAC Protocol (CCMP)[4]. AES is the successor to the Data Encryption Standard (DES) and satisfies US government security requirements and has been adopted as an official encryption standard.

! **Important:** To build a true 802.11i RSN, the use of AES-CCMP is mandated by the 802.11i standard.

How AES-CCMP works

The CCMP protocol is based on the AES encryption cipher using the Counter Mode with CBC-MAC (CCM) mode of operation. The CCM mode combines Counter (CTR) mode privacy and Cipher Block Chaining Message Authentication Code (CBC-MAC) authentication. CCMP, in the context of Wi-Fi security, is often referred to as AES-CCMP or simply AES.

AES processing in CCMP must use AES 128-bit key and 128-bit block size. Per FIPS 197 standard, the AES algorithm (a block cipher) uses blocks of 128 bits, cipher keys with lengths of 128, 192 and 256 bits, as well as a number of rounds 10, 12 and 14 respectively.

AES-CCMP security

AES-CCMP introduces a higher level of security from past protocols by providing protection for the MAC protocol data unit (MPDU) and parts of the 802.11 MAC headers. This protects even more of the data packet from eavesdropping and tampering.

AES-CCMP is superior to WEP and TKIP in many ways:

- AES-CCMP was built from the ground up specifically for 802.11 encryption – it goes far beyond the RC4 steam cipher used by WEP and TKIP
- AES-CCMP offers greater data privacy by encrypting parts of the 802.11 header

[4] AES is often referred to as the encryption protocol used by 802.11i, however AES itself is simply a block cipher. The actually encryption protocol is CCMP.

AES-CCMP Encryption continued

The major drawbacks of AES include:

- **No key rotation/updates** – the same static pre-shared master key (PSK) must be shared somehow with all clients before they are allowed to associate with the WLAN. The AES-CCMP protocol makes no provision for automatically generating new master keys
- **Hardware requirements** – AES-CCMP is not backwards compatible with legacy Wi-Fi hardware. This means AES-CCMP deployments may require a firmware or hardware upgrade

EAP Authentication

Overview

The Extensible Authentication Protocol (EAP) or RFC 3748 is, very simply, a transport protocol that has been optimized for authentication. It is important to note that EAP is not, in itself, an authentication protocol. The EAP protocol expands on authentication methods used by the Point-to-Point Protocol (PPP). EAP can support multiple authentication mechanisms such as token cards, smart cards, digital certificates, one-time passwords, and public key encryption. This section will focus on the popular EAP/authentication combinations and discuss how each works within a wireless security framework.

In this document, EAP is always used in a wireless LAN context – therefore a more correct name for the EAP protocol is *EAPOL*, or EAP over LAN.

How EAP works

Here's how it works: a user requests connection to a wireless network through an access point. The access point requests identification data from the user and transmits that data to an authentication server. The authentication server asks the access point for proof of the validity of the credentials. After the access point obtains that verification from the user and sends it back to the authentication server, the user is connected to the network as requested.

The way the identification credentials are requested and transmitted is the difference between the different versions of EAP.

EAP flavors

There are many different combinations of EAP and authentication types. A complete listing is available at http://www.iana.org/assignments/eap-numbers. the following list offers a description of the most popular versions as well as some design considerations:

- **EAP-MD-5** (Message Digest) is a EAP authentication type that provides base-level EAP support. EAP-MD-5 is typically not recommended for wireless LAN implementations because it may allow a user's password to be derived. It also provides for one-way authentication only - there is no mutual authentication of wireless client and the network. More importantly it does not provide a means to derive dynamic, per session wired equivalent privacy (WEP) keys

- **EAP-TLS** (Transport Layer Security) provides for certificate-based and mutual authentication of the client and the network. It relies on client-side and server-side certificates to perform authentication and can be used to dynamically generate user-based and session-based WEP keys to secure subsequent communications between the WLAN client and the access point. One drawback of EAP-TLS is that certificates must be managed on both the client and server side. For a large WLAN installation, this could be a very cumbersome task

EAP Authentication continued

- **EAP-TTLS** (Tunneled Transport Layer Security) was developed by Funk Software and Certicom, as an extension of EAP-TLS. This security method provides for certificate-based, mutual authentication of the client and network through an encrypted channel (or "tunnel"), as well as a means to derive dynamic, per-user, per-session WEP keys. Unlike EAP-TLS, EAP-TTLS requires only server-side certificates

- **LEAP** (Lightweight Extensible Authentication Protocol), is a proprietary EAP authentication type used primarily in Cisco Aironet WLANs. It encrypts data transmissions using dynamically generated WEP keys, and supports mutual authentication. LEAP has been shown to be easily broken with publicly available tools; compromising user credentials and accounts.

- **EAP-FAST** (Flexible Authentication via Secure Tunneling) is another proprietary EAP developed by Cisco. Instead of using a certificate, mutual authentication is achieved by means of a PAC (Protected Access Credential) which can be managed dynamically by the authentication server. The PAC can be provisioned (distributed one time) to the client either manually or automatically. EAP-FAST has seen some popularity, but has been shown to have security issues.

- **PEAP** (Protected Extensible Authentication Protocol) provides a method to transport securely authentication data, including legacy password-based protocols, via 802.11 wireless networks. PEAP accomplishes this by using tunneling between PEAP clients and an authentication server. Like the competing standard Tunneled Transport Layer Security (TTLS), PEAP authenticates wireless LAN clients using only server-side certificates, thus simplifying the implementation and administration of a secure wireless LAN.

  ➲ **Note:** On Cisco ACS server 3.1 and lower, Cisco only supports Cisco-PEAP, which only allows the use of a smart card as the inner authentication protocol. Microsoft PEAP is considered the mainstream PEAP and uses MS-CHAPv2 as the inner authentication protocol.

- **EAP-MSCHAP2** is an EAP encapsulation of MS-CHAP-v2 (RFC-2759). MS-CHAP-v2 requires a user name and password. This protocol is rarely used on its own. Instead it is typically used inside of a PEAP-encrypted tunnel

Each combination of EAP plus an authentication type offers a unique approach to authentication. Which one to use is generally determined by the level of security required, the amount of administrative/management overhead desired, and the limitations of the clients (supplicants) that will implement EAP as well as the capabilities of the RADIUS servers used in the deployment.

EAP Authentication continued

EAP comparisons The following table provides a side by side comparison of the EAP types:

	MD5	TLS	TTLS	PEAP	LEAP	FAST
Feature / Benefit						
Client-side authentication/certificate required	No	Yes	No	No	No	No (PAC)
Server-side authentication/certificate required	No	Yes	Yes	Yes	No	No (PAC)
Authentication method	One-way	Mutual	Mutual	Mutual	Mutual	Mutual
Deployment complexity	Low	High	Moderate	Moderate	Moderate	Moderate to high
Security strength	Low	Highest	High	High	Low	Medium to high

EAP security Based on this table, we can draw some reasonably clear conclusions:

- TLS, while very secure, requires client certificates to be installed on each wireless workstation. Installing and maintaining a PKI infrastructure must be part of any TLS installation and does create more administrative overhead. If a working PKI already exists, TLS is a very good option

- TTLS addresses the certificate issue by tunneling TLS, and thus eliminating the need for a certificate on the client side. If a working PKI structure does not exist, this is an option worth considering

- LEAP is one of the earliest EAP implementations; however inherent security flaws have now made it less popular and it is not recommended

© 2005 Aruba Networks

EAP Authentication continued

- EAP-FAST promises to be as easy as LEAP but as secure as PEAP, however it has different implementation and operational modes that, ultimately, offer a compromise. The highest security, ultimately, ends up looking very similar to PEAP – without the widespread client support that PEAP enjoys

- PEAP works similarly to EAP-TTLS in that it does not require a certificate on the client side and is natively supported by many client operating systems. PEAP is the protocol of choice when client-side certificates are not required

802.1x Access Control

Overview

802.1x is an IEEE standard designed to enforce authentication of a client before Layer 2 access to the network is permitted. The 802.1x protocol consists of three parts:

- **Supplicant**, or client, is software running on a device trying to gain access to the network
- **Authentication server** is the system that validates client credentials and determines if the client should be allowed access. The authentication server must be a RADIUS[5] server
- **Authenticator** is a piece of software running on the device that communicates with both the supplicant and the authentication server and enforces the authentication server's deny or permit directive

The supplicant and the authenticator use the Extensible Authentication Protocol (EAP) to securely negotiate authentication. There are several different types of EAP in use today. During the authentication process, the authentication server and the supplicant negotiate which type of EAP they will use for the authentication transaction. The choice of EAP must be mutually supported by both devices.

How 802.1x works

Here are the steps that must occur before 802.1x will allow a device access to the network:

1. A wireless client device (supplicant) requests access to a WLAN. An authenticator (access point/mobility controller) asks for the supplicant's identity.[6]
2. The supplicant responds to the authenticator with identity data that will establish its credentials[7]

[5] Remote Authentication Dial-In User Service (RADIUS) is defined in RFC-2865 and other, later documents. It was originally designed to provide centralized authentication, authorization and access control (AAA) for dial-up ISP connections. The 802.1x standard mandates an AAA server as the back-end server. RADIUS is the only AAA server supported today. Theoretically, other types of AAA servers could be used, but nothing has been standardized or implemented as of this writing.

[6] No other traffic than EAP traffic is allowed at this point, i.e. the "port" is closed.

[7] EAP, the protocol used to transport authentication messages, was originally used for dial-up PPP. The identity was the user name and was sent in the clear (not encrypted). A malicious sniffer might capture this and learn the user's identity. *Identity hiding* is therefore used; the real identity is not sent before an encrypted session is established

802.1x Access Control continued

3 After the identity has been sent, the authentication process begins. The authenticator re-encapsulates the EAPOL messages to RADIUS format and passes them to the authentication server[8]

4 Each authentication process is slightly different, depending on the type of EAP authentication used, however, at some point the authentication server will send a success or failure message

5 If the authentication server transmits a success message, the authenticator opens the “port” for the supplicant and network access is granted. If a failure message is sent, the port is not opened. This supplicant is free at this point to try again

802.1x security

802.1x offers a powerful access control mechanism. Unlike any other protocol discussed in this document, 802.1x ensures a client device has absolutely no network access before authentication and validation.

The advantages of 802.1x are:

- High level of network access control
- Can work with other protocols to provide authentication and encryption
- Key distribution and rotation

The major drawbacks of 802.1x are:

- **No encryption** – the 802.1x protocol does not include an encryption algorithm
- **No authentication** – the 802.1x protocol does not include authentication, thus it must be coupled with a protocol such as EAP
- **RADIUS requirement** – the 802.1x protocol only support RADIUS[9] today as a de facto standard for authentication server messages. This means a RADIUS authentication server, or proxy, must be used

With all of these issues, 802.1x is still an extremely effective security protocol when coupled with other standards.

[8] During the authentication process, the authenticator simply translates and relays packets between the supplicant and the authentication server

[9] Not all RADIUS servers support, or are configured for, every EAP type. Always consult your RADIUS documentation before implementation

802.11i

Overview

IEEE 802.11i is an amendment to the original 802.11 standard specifying security mechanisms for wireless networks (Wi-Fi). The draft standard was ratified on June 24, 2004, and supersedes the previous security specification, Wired Equivalent Privacy (WEP), which was shown to have severe security weaknesses.

How 802.11i works

The 802.11i architecture contains the following components:

- 802.1X for authentication (using EAP and an authentication server)
- RSN for keeping track of associations
- AES-based CCMP to provide confidentiality (data privacy), integrity and origin authentication
- A four-way authentication handshake

802.11i uses the concept of a Robust Security Network (RSN). In an RSN, wireless devices need to support additional capabilities. In a true RSN, only WLAN access points/system only allow RSN mobile devices to connect and places rigorous security constraints on the process.

Because it is recognized that not all wireless NICs will be RSN-capable, an intermediate network has been defined, Transitional Security Network (TSN). A TSN is similar to an RSN in concept and architecture, but does not implement all of the mandatory capabilities and mechanisms.

802.11i security

802.11i is by far the strongest security commercially available for wireless networks. RSN-compatible equipment is readily available today and can be implemented as software-only versions for some older cards[10]. 802.11i is most robust, scalable, and secure solution today. Its main appeal is to enterprise users where key management and administration have been a major headache and have often stopped security-conscious wireless implementations.

802.11i has been designed from the ground up using proven technologies. Although no security system can ever be considered totally unbreakable, 802.11i security is a dependable solution and shows no weaknesses at this time.

[10] Software-only implementations are notable for their poor performance compared to hardware-based solutions

802.11i continued

As an entire security eco-system, 802.11i combines many of the fundamental protocols discussed in this document. It offers a complete solution:

- **Access control** – offers a high level of network access control through mechanisms such as 802.1x
- **Encryption** – offers AES-CCMP as proven, highly regarded cryptographic algorithms that go far beyond the RC4 stream cipher used by WEP and TKIP. AES-CCMP also goes farther by encrypting parts of the 802.11 header
- **Authentication** – is offered using the industry standard EAP and its associated authentication types
- **Key generation/distribution and rotation** - the ability to derive a hierarchy of keys from a master key, this is typically performed by EAP and 802.1x

© 2005 Aruba Networks

xSec

Overview

xSec enjoys all of the same security benefits as 802.11i with the addition of higher levels of encryption. It is essentially a slight modification of the 802.11i/WPA 2.0 standard with the intent of making it FIPS compliant and certifiable.

How xSec works

The xSec architecture contains the following components:

- 802.1X for authentication (using EAP and an authentication server)
- RSN for keeping track of associations
- AEC-CBC-256 and HMAC-SHA1 to provide confidentiality (data privacy), integrity and origin authentication
- A four-way authentication handshake

xSec functions essentially the same as 802.11i. The major differences are: wired and wireless functionality and higher encryption levels. Like 802.11i, xSec requires a new software client (available from Funk Software) and has higher hardware requirements.

xSec security

xSec meets the requirements for FIPS 140-2 level certification and the U.S. Department of Defense (DoD) direction 8100.2 for secure, Layer 2 data transmission. xSec is the only wireless protocol that satisfies these requirements today. As of this writing, it has passed initial validation and is waiting final certification.

Unlike 802.11i, xSec does not support TKIP or AES-CCMP. Instead, it uses AEC-CBC-256 for encryption. This standard is already FIPS-140-2 certified and approved for government use. The replacement of AES-CCMP with AEC-CBC-256 is a major difference between 802.11i and xSec. It brings several advantages. One of the most significant is that xSec only supports 256-bit encryption keys. AES-CCMP only supports 128-bit keys.

This protocol is also unique in that it is available, without modification, for both wired and wireless devices. This is unlike the 802.11i protocol, which is wireless only, or the 802.1x protocol which supports wired devices, but with lower-grade security options such as MD5.

Another unique feature of this protocol is that it offers complete encryption of transmitted packets. 802.11i offers partial encryption, but xSec takes this a step further to ensure complete data privacy.

xSec continued

xSec encompasses all of the benefits of 802.11i and improves upon them even further:

- **Access control** – offers a high level of network access control through mechanisms such as 802.1x
- **Encryption** – offers AEC-CBC-256, a encryption protocol approved for government use for Layer 2 data transmissions, with HMAC-SHA1 for complete 802.11 header encryption
- **Authentication** – is offered using the industry standard EAP and its associated authentication types
- **Key generation/distribution and rotation** - the ability to derive a hierarchy of keys from a master key, this is typically performed by EAP and 802.1x
- **Universal media support** - xSec is unique in that it works across both wired and wireless networks

© 2005 Aruba Networks

Wi-Fi Protected Access (WPA)

What is WPA?

Wi-Fi Protected Access (WPA) was developed by the Wi-Fi Alliance as an interim solution aimed at addressing the weakness of WEP-based wireless networks. WPA has, rightly, been admired as a masterpiece of retro engineering. It addresses the weaknesses of WEP and the result is a very secure security system that is backwardly compatible with most existing Wi-Fi compliant equipment. WPA is a practical solution that will provide more than adequate security for most wireless network applications.

How WPA works

WPA is designed for use with an 802.1X/EAP authentication server, which distributes different keys to each user. However, it can also be used in a less secure "pre-shared key" (PSK) mode, where every user is given the same *passphrase* – a passphrase is similar to a password. The Wi-Fi Alliance calls the pre-shared key version *WPA-Personal* and the 802.1X authentication version *WPA-Enterprise*.

Unlike the 802.11i standard that uses AES-CCMP by default, WPA data is encrypted using TKIP's RC4 stream cipher, with a 128-bit key and a 48-bit initialization vector (IV). One major improvement in WPA over WEP is the use of the Temporal Key Integrity Protocol (TKIP), which dynamically changes keys as the system is used. When combined with the much larger IV, this defeats the well-known key recovery attacks that were discovered with WEP.

In addition to authentication and encryption, WPA also provides vastly improved payload integrity when compared to WEP. The cyclic redundancy check (CRC) used in WEP is inherently insecure; it is possible to alter the payload and update the message CRC without knowing the WEP key. A more secure message authentication code (MIC) is used in WPA.[11] The MIC used in WPA includes a frame counter, which prevents replay attacks from being executed; this was another weakness in WEP.

[11] This is called MIC for *Message Integrity Code*, which also happens to be based on an algorithm called Michael

Wi-Fi Protected Access (WPA) continued

WPA security

By any measure, WPA is a very strong security system. Any system using WPA today will have addressed the major shortcomings of the original 802.11 standard.

Major features of WPA include:

- Use of 802.1x for access control and authentication
- TKIP encryption that is far stronger than WEP and fixes many issues with larger keys, IVs and changing keys
- Key management and distribution scheme (PSK is also still supported)
- MIC packet integrity checking prevents packet replay attacks
- Backwards-compatible with most 802.11 network cards

The major drawbacks of WPA include:

- Backwards-compatibility limits crypto operations – thus encryption is still ultimately based on RC4, as is WEP
- TKIP is not FIPS-certified or approved for US government use
- WPA, as an interim solution, is not compatible with pure 802.11i/RSN environments

WPA2

What is WPA2?

Wi-Fi Protected Access v2.0 (WPA2) is an implementation of the IEEE 802.11i standard tested and certified by the Wi-Fi Alliance. It is interoperable with any other 802.11i-compliant system. WPA2 implements the mandatory elements of 802.11i. In particular, the Michael algorithm is replaced by a message authentication code, CCMP, that is considered fully secure and RC4 is replaced by AES.

How WPA2 security works

WPA2 works just like the 802.11i standard and supports all of the mandatory features and capabilities.

For more information on WPA2, please refer to the 802.11i section of this document.

Recommendations

Overview

Navigating the jungle of complex, competing security standards, acronyms and buzzwords is a tough job. It can be very confusing and yet, the need for wireless security drives us to make a decision.

Helpful questions to ask

When determining what security mechanisms to use, the following questions can be very helpful:

1. Is this for personal use or enterprise use?
2. Are the wireless network adapters in each client very old or relatively new?
3. Is a Public Key Infrastructure (PKI) already in place?
4. Is a US government-sanctioned level of security required?
5. What type(s) of RADIUS servers are being used in the enterprise?

Personal versus enterprise

If the wireless project is relatively small – for example, a WLAN limited to a private home – certain limitations can more readily be accepted. For instance, a personal WLAN is typically comprised of only a few access points and a few clients. A RADIUS server may not be available to support dynamic key generation/rotation. While the need for security may be high, the need for automatic, computation complexity is not. Therefore a good security implementation in this case would be to use WPA-TKIP with a strong pre-shared key (PSK).

This eliminates much of the complexity of WPA-TKIP (server certificates, RADIUS, etc.) while still enjoying a high level of security.

An enterprise solution however, will typically involve hundreds or thousands of client devices. A PSK simply does not make sense in this case and would represent a large security hole. Also, enterprises typically have EAP-capable authentication servers available which can be readily configured. PEAP, as the most popular and well-supported EAP type, is highly recommended.

An enterprise should always look to WPA-TKIP as a minimum solution and strongly consider WPA2-AES. The following describes when WPA2 may be required or strongly desirable.

Network adapters

If the wireless network adapter in the clients is very old, WPA2 may not be feasible as it is unlikely to be supported. If WPA2 is required, it may be possible to install a software supplicant client that can perform the advanced calculations in software rather than hardware. However this functionality comes at the price of performance.

Recommendations continued

Public Key Infrastructures

All strong EAP authentication types (such as PEAP) require, at minimum, a server-side certificate on the authentication server. Some EAP types, such as EAP-TLS, go even further and require client-side certificates as well.

While a single server certificate can be purchased or provisioned fairly easily as a one-time event, client certificates are much more burdensome. None of the standards described allow for a completely automatic mechanism for provisioning client certificates[12]. Certainly, even if such a protocol existed, it would still require the existence of a Certificate Authority (CA) and the rest of a full-blown PKI. This has a real management cost.

If an enterprise-wide PKI solution already exists, a client distribution mechanism has generally also been implemented. This can make the adoption of protocols such as EAP-TLS as simple as PEAP. In this case, it is generally preferable to go with the improved security EAP-TLS offers since it comes at little to no additional cost.

If a PKI does not already exist, but EAP-TLS is still desired, it can still be achieved, with a slight modification. PKIs are generally regarded as large-scale projects in their own right that can be time-consuming to deploy. Therefore, implementing a new PKI in conjunction with a WLAN that relies on it is not recommended. Instead, the following phased approach has been show to work very well:

1. Deploy the wireless LAN with WPA-TKIP or WPA2-AES and PEAP
2. Test and validate PEAP functionality
3. Install PKI infrastructure
4. Test client certificates (EAP-TLS) in a limited pilot while the majority of clients continue to use PEAP
5. Once PKI is operational and functioning correctly with EAP-TLS, convert the rest of the clients to EAP-TLS

➲ **Note:** Most of the work in an EAP environment consists of correctly configuring the RADIUS server, installing the server certificate and configuring the clients. Once a valid client certificate is installed, switching from PEAP to EAP-TLS is trivial.

[12] Microsoft offers an auto-enrollment mechanism for rolling out client certificates automatically in Windows-only environments. For more information please refer to: http://www.isaserver.org/img/upl/vpnkitbeta2/autoenroll.htm

Recommendations continued

Government-sanctioned security

Some organizations are required to use protocols that have been approved for use by the US government, i.e. FIPS certification. Also some organizations (such as financial institutions) prefer to use the strongest possible security measures available as a general course of action. In these cases, the use of WPA-AES is not sufficient and xSec must be used instead. xSec requires an xSec-aware client, modern hardware and a complete, end to end, RSN infrastructure.

RADIUS servers

An important consideration for almost all authentication protocols described in this document is the type of RADIUS server used in the organization. Different servers may or may not support the desired EAP type if 802.11i or 802.1x are being used. The following table outlines the most common versions of RADIUS servers and their support of various EAP types:

	MD5	TLS	TTLS	PEAP	LEAP	FAST
RADIUS Server						
Cisco ACS 3.2 and higher	Yes	Yes	No	Yes	Yes	Yes
Microsoft IAS	Yes	Yes	No	Yes	No	No
Funk Steel Belted RADIUS	Yes	Yes	Yes	Yes	Yes	Yes
InfoBlox	Yes	Yes	No	Yes	Yes	No
FreeRADIUS	Yes	Yes	No	Yes	Yes	No
Radiator	Yes	Yes	Yes	Yes	Yes	No

➲ **Note:** This table is correct as of the writing of this document. For the most up to date information, please refer to the documentation for your specific RADIUS server.

© 2005 Aruba Networks