Controller-Based Wireless LAN Fundamentals

An end-to-end reference guide to design, deploy, manage, and secure 802.11 wireless networks

Jeff Smith • Jake Woodhams
Robert Marg

ciscopress.com

Controller-Based Wireless LAN Fundamentals

Jeff Smith, Jake Woodhams, Robert Marg

Published by:
Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Printed in the United States of America

First Printing November 2010

Library of Congress Cataloging-in-Publication data is on file.

ISBN-13: 978-1-58705-825-7

ISBN-10: 1-58705-825-1

Warning and Disclaimer

This book is designed to provide information about controller-based wireless local-area networks. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an "as is" basis. The authors, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Corporate and Government Sales

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact: **U.S. Corporate and Government Sales** 1-800-382-3419 corpsales@pearsontechgroup.com

For sales outside the United States, please contact: **International Sales** international@pearsoned.com

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers' feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through email at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Publisher: Paul Boger

Associate Publisher: Dave Dusthimer

Executive Editor: Mary Beth Ray

Managing Editor: Sandra Schroeder

Senior Development Editor: Christopher Cleveland

Senior Project Editor: Tonya Simpson

Copy Editor: John Edwards

Editorial Assistant: Vanessa Evans

Indexer: Tim Wright

Manager, Global Certification: Erik Ullanderson

Business Operation Manager, Cisco Press: Anand Sundaram

Technical Editors: Saurabh Bhasin, Sujit Ghosh

Proofreader: Sheri Cain

Book Designer: Louisa Adair

Cover Designer: Sandra Schroeder

Composition: Mark Shirar

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

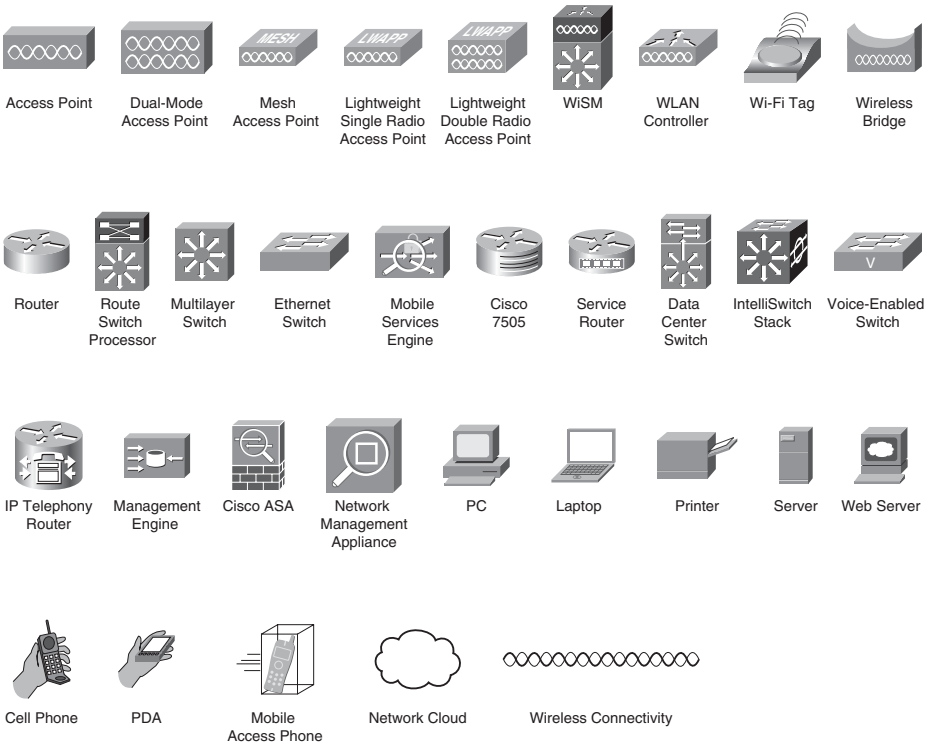
Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

Contents at a Glance

Contents

Icons Used in This Book



Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- **Boldface** indicates commands and keywords that are entered literally as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a **show** command).
- *Italic* indicates arguments for which you supply actual values.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate an optional element.
- Braces ({ }) indicate a required choice.
- Braces within brackets ([{ }]) indicate a required choice within an optional element.

Introduction

Since the first Cisco Press book on the fundamentals of 802.11 networks, much has changed. Wireless local-area networks (WLAN) have grown dramatically in size and scope, creating changes for traditional wireless deployment architectures. Many of the challenges with prior generations of wireless networking designs have been greatly reduced with development of the Cisco Wireless Unified Network (CUWN) architecture. The Cisco Unified approach has allowed wireless networks to efficiently scale to meet new requirements and business needs. WLAN controllers rapidly became the dominant wireless architecture and have been standardized through the IETF Control and Provisioning of Wireless Access Points (CAPWAP) protocol. CAPWAP defines an industry-standard protocol for how controllers manage wireless access points. The other foundational change to wireless has been the increased performance supported by the 802.11n amendment to the IEEE 802.11 wireless LAN standard. While the controller removed the management and scaling constraints, 802.11n is removing many of the performance constraints with wireless networking. The results have been a shift from wireless networks being a convenience to becoming mission-critical for most organizations.

This book focuses on introducing the concepts and reasoning behind controller-based wireless and higher-performance wireless networks. The elements of wireless LAN controller (WLC)–based WLANs are introduced, and the goal is to provide an update on many of the fundamentals that have been introduced since the publication of the Cisco Press 802.11 fundamentals book. This book is targeted toward IT engineers who are new to wireless controllers, 802.11n, Wireless Control System (WCS), wireless multicast, and mission-critical wireless networks. This book should be viewed as an introduction to wireless controllers and the knowledge needed to support life cycle design and support of controller-based architectures. Organizations new to wireless controllers and those migrating from legacy wireless networks to controller-based and 802.11n wireless networks will find this book to be a valuable guide.

This book also has a wealth of knowledge gleaned from the authors' experiences in terms of guidelines, deployments, and configuration of wireless LAN architectures.

Goals and Methods

The goal of this book is to introduce you to the concepts and principles of mission-critical, high-performance, controller-based wireless LAN deployments. To accomplish this, the book includes the following elements:

- **Drivers for the migration to controller architecture:** The book covers the evolution, challenges, and reasoning about why WLAN controllers have become essential in modern WLAN deployments.
- **Elements of the controller-based wireless architecture:** The book introduces each of the elements in the end-to-end controller-based wireless architecture.
- **Wireless controller WLAN design and implementation:** The book includes details on how to design and implement WLANs with wireless controllers.

- **Wireless security with a wireless LAN controller:** This book covers the key principles of security WLANs with a wireless LAN controller.
- **Wireless multicast design and implementation:** This book covers the key principles of both wired and wireless multicast implementations within the wireless LAN controller.

Who Should Read This Book

This book is planned and written for network engineers who design, configure, implement, and maintain wireless networks, with an emphasis on WLAN controller basics, 802.11n principles, wireless architecture designs, and wireless network management. This book also focuses on the "newbie" to wireless networking in the hopes of clarifying the alphabet soup commonly known as IEEE 802.11 and introducing the key principles to designing and managing a wireless architecture.

How This Book Is Organized

Although this book can be read from cover to cover, it is designed to be flexible and allow you to easily move between chapters and sections of chapters to learn just the information that you need.

The book covers the following topics:

- **Chapter 1, "The Need for Controller-Based Wireless Networks":** This chapter describes the evolution of WLANs and the principles that drove the development of the wireless LAN controller.
- **Chapter 2, "Wireless LAN Protocols":** This chapter explains radio wave fundamentals so that you have the basis for understanding the complexities of deploying wireless LANs.
- **Chapter 3, "802.11n":** This chapter describes the key features provided by 802.11n. 802.11n increases the performance and reliability of WLAN through protocol improvements and the support for multiple input, multiple output (MIMO) radio systems. The concepts on how MIMO improves performance are presented.
- **Chapter 4, "Cisco Unified Wireless LAN Security Fundamentals":** This chapter explains the key elements of wireless security in the unified wireless networks.
- **Chapter 5, "Design Considerations":** This chapter provides a background and consideration for designing unified wireless networks.
- **Chapter 6, "Cisco Unified Wireless LAN Architectures":** This chapter explains the details of the unified wireless LAN architecture.
- **Chapter 7, "Troubleshooting":** This chapter focuses on how to troubleshoot client issues using the unified wireless networks.

- **Chapter 8, "Introduction to WCS":** This chapter provides an overview of the key features and principles and configuration steps that you should complete when deploying a wireless LAN management solution. This chapter also navigates you through the key menus within the WCS management platform, highlighting the key elements in planning, designing, and configuring the Cisco Unified Wireless Network architecture.
- **Chapter 9, "Next-Generation Advanced Topics: Multicast":** This chapter focuses on both the wired and wireless design and implementation phases of multicast. This chapter takes a deep approach to multicast, with a detailed emphasis of controller-based multicast design best practices and principles.

Chapter 4

Cisco Unified Wireless LAN Security Fundamentals

"New Vulnerability Allows Hackers to Penetrate Wireless Networks!" screams the headline in the newspaper or periodical. Perhaps the accompanying article describes some new theoretical vulnerability announced by a security research group that (surprise!) offers wireless LAN (WLAN) security consulting services. Or maybe it's a WLAN vendor that, quite naturally, not only "discovered" the new vulnerability but also offers the industry's "only" or "best" solution. Or maybe the accompanying article contains a sensationalistic description of how some "white hat" hacker demonstrated a new tool to exploit a WLAN or network attack vector at a security conference. Quite often, what's "new" is just a variant of what's old—a new exploit tool for a well-known vulnerability, for example. But then, every once in a while, articles of this ilk describe a significant new development that gravely impacts the industry.

Unfortunately, many journalists—even those writing for industry and technical publications—struggle to grasp even the fundamentals of WLAN technology, let alone the intricacies and complexities of WLAN security threats and their full ramifications on network design and implementation. It's shocking how often vulnerabilities common only in consumer WLAN implementations are applied in hysterical, sweeping generalizations to all wireless networks.

This is not to say that there aren't real security threats with WLAN networks; there definitely are some significant security challenges for WLAN network designers and operators. But the challenges are, for the most part, manageable when reality is filtered out of all the hype and the problem domain is well understood. Indeed, we often observe that the WLANs our customers deploy are more secure than their companion wired networks!

This chapter discusses the fundamentals of wireless LAN security in the context of the Cisco Unified Wireless Network (CUWN). An in-depth discussion and analysis of WLAN security can be its own book. In fact, there are already a number of excellent books available on the topic of WLAN security. Some favorites are listed in the references at the end of this chapter.

This chapter begins with an introduction of the security risks with WLAN technologies and continues with an explanation of technology building blocks that address and mitigate the risks.

When you are done reading this chapter, you should have sufficient background information on WLAN security. The security concepts discussed in this chapter are woven throughout the fabric of the CUWN. Indeed, one of the real benefits of the CUWN architecture is that it simplifies the design, deployment, and operations of security for your WLAN.

Understanding WLAN Security Challenges

You should know the vulnerability points of any network you are trying to secure and understand how the bad guys try to exploit them. How else do you separate the real from the hype and design sensible security policies and select the right WLAN security technologies?

This would be a good place for one of those hackneyed quotes about the importance of knowing your enemy from the likes of Sun Tzu's *The Art of War*. But all the good quotes we know of have already been used *ad nauseum* by other authors. So we'll spare you (and ourselves).

Instead, let's move right into discussing the security risks. The discussion that follows centers on the places where WLANs have security exposures as opposed to specific attacks and flaws. Basically this is because books have a long life, and by the time you read this, today's latest, greatest WLAN exploits might be old news. But the risk points remain the same. The risks discussed are as follows:

- Vulnerabilities inherent to the radio transmission medium
- Vulnerabilities inherent to the standards definitions
- Vulnerabilities inherent to mobility
- Readily available profiling and attack tools
- Misconfigured wireless devices and clients
- Rogue access points and devices

After concluding the following sections, you should have a good overview of the real risks associated with WLANs and be ready to take a closer look at the building blocks that address these vulnerabilities.

Vulnerabilities Inherent to the Radio Transmission Medium

WLANs have inherent vulnerabilities arising from the use of the airwaves and radio waves as the transmission medium. The two significant problem areas are

- Physical containment of transmissions
- Use of the unlicensed radio spectrum

The sections that follow look at these problem areas in greater detail.

Physical Containment Problem

With an Ethernet LAN, eavesdropping or attacking the network from the inside requires physical access to the network. Typically, an attacker must be able to connect a machine to a switchport in the network somewhere. Violating the network's security requires violating physical security.

This is not the case with WLANs. The basic physics of the transmission medium creates a physical containment problem. WLANs use radio signals over the air as the physical transmission medium. After a radio signal leaves its source, whether it is an access point or a wireless client, the signal travels through the air in many directions, and you have little or no control over the signal propagation.

Any listener with an antenna tuned to the right frequency and within range of the WLAN can "hear" the transmissions of both clients and access points. Skilled attackers know how to use high-gain directional antennas to profile and eavesdrop on WLAN networks from far away. But even relatively unskilled attackers can hear your WLAN pretty easily with simple tools.

If an attacker can hear transmissions in the unlicensed spectrum using readily available equipment from the WLAN coverage area, it's only logical that the attacker can also transmit into the WLAN coverage area relatively easily to cause big problems. Attackers might do this for one or more reasons. For example, the attacker might simply be trying to create a denial of service by using up available radio channel time. The attacker could also be trying to spoof a legitimate wireless device. It's not uncommon for an attacker to spoof a legitimate access point to try to trick wireless clients into connecting to the attacker.

Unlicensed Radio Spectrum Problem

The physical containment problem is exacerbated by the use of the unlicensed radio spectrum in both the 2.4-GHz and 5-GHz bands. Other types of wireless networks—for example, the cellular phone carrier networks—enjoy a certain amount of "security by obscurity" because they have a dedicated radio spectrum allotted to them by a regulatory agency. While that doesn't solve the physical containment problem, it makes it much harder for an eavesdropper or attacker because he has to obtain or build special equipment and tools to attack the network. This typically requires sophisticated knowledge and technical skills. It is also illegal.

On the other hand, with WLANs, attackers can use off-the-shelf equipment and open-source software attack tools. The skills and knowledge level required are moderate. Also, because the spectrum is unlicensed, the legal questions are much more abstruse.

WLANs can be susceptible to competition with non-802.11 devices that use the same radio channels in the unlicensed spectrum. From the 802.11 WLAN's perspective, this

competition is considered noise, and if strong enough, can significantly degrade the network's performance. Common products that use some of the same spectrum as the WLAN are Bluetooth wireless devices, 2.4-GHz cordless phones, and microwave ovens. Legitimate devices don't represent a security problem per se, but they can affect WLAN availability, creating a de facto denial of service. Malicious attackers can use jammers to the same effect.

One of our favorite stories from Cisco sales lore comes from a customer bake-off between Cisco and a competitor for a large WLAN deal. The Cisco pilot was going very poorly and Cisco engineers were completely baffled because every failed test case in the pilot environment worked perfectly in Cisco labs. After many sleepless nights and much consternation, the mystery was solved. The competitor was camping out in a van outside the test environment with a doorless microwave oven jamming the airwaves during the pilot!

While this story is almost certainly apocryphal, it does illustrate how a legitimate product can be used nefariously and how the unlicensed spectrum makes the WLAN susceptible to RF jamming attacks.

Vulnerabilities Inherent to the Standards Definitions

The underlying IEEE 802.11 standards definitions have some inherent vulnerabilities, which fall into two categories:

- Authentication and encryption weaknesses
- Unauthenticated management and control frames

The sections that follow look at some of the details.

Authentication and Encryption Weaknesses

Put simply, authentication controls access to the network and networked resources by using techniques that identify who and which devices are allowed onto the network and those that are not. Encryption protects data frames in transit on the network, using cryptographic algorithms to obfuscate the frame content. When you consider the vulnerabilities inherent to the transmission medium, it's pretty obvious why both authentication and encryption are really important security concepts with WLANs.

The original IEEE 802.11 specification was released in 1997 and called out a mechanism for authentication and data privacy called Wired Equivalent Privacy, or WEP for short. This name is telling because it reflects the original goals of the standards designers—to provide a wireless data privacy mechanism roughly equivalent to what you get with a wired Ethernet network. In other words, it was supposed to be as hard to break WEP encryption as it is to violate an enterprise's physical security to gain access to the wired network. The WEP standard was designed to be a trade-off between "reasonably strong" security and implementation simplicity and exportability.

WEP is based on the shared-secret concept. Both end devices of a WLAN connection share a secret WEP key. The WEP key can be used to authenticate wireless devices; if a device has the secret WEP key, it must be authorized!

The WEP key is also used to encrypt data transmissions between each end of the WLAN connection. The original 1997 version of the 802.11 specification called out 40-bit WEP keys. In 1999, the specification allowed expanding the key length to 104 bits. These keys are statically configured on the devices that will use the WLAN.

As 802.11 WLAN technology started to take off, a lot of smart people in the cryptographic community started to take a good look at WEP as a security mechanism. In 2000 and 2001, several landmark papers were published detailing critical problems with WEP. If you're really interested, these papers are listed in the references at the end of this chapter, and they make for excellent reading to combat insomnia.

Not long after these papers were published, exploit tools appeared on the scene. These tools are now readily available on the Internet and are pretty easy to use, even for novices. So the most important thing to know about WEP is that it is irreversibly cracked and should never be used. It bears repeating: WEP is totally ineffective for data privacy because of cryptographic flaws; don't use it.

Recognizing that WEP was not the answer to WLAN security, the IEEE formed the 802.11i task group to come up with a robust security scheme for the future. The 802.11i task group's work was ratified in 2004.

While the 802.11i standard was in draft form, the Wi-Fi Alliance released its own requirements based on a subset of the 802.11i standard. The first iteration of these requirements was called Wi-Fi Protected Access (WPA). An update to these requirements is based on the complete, ratified 802.11i standard and is called Wi-Fi Protected Access Version 2 (WPAv2). The industry as a whole has moved toward 802.11i/WPAv2-based security, and that's where you should be too. Later in the chapter, you will learn more about WPAv2.

Unauthenticated Management Frames

Recall from the basics of 802.11 WLANs that there are three kinds of frames: control, management, and data frames. Discussions of WLAN security weaknesses are incomplete without noting that the 802.11 specification lacks an authentication mechanism for management frames.

The lack of authentication for management frames opens the door to a variety of denial of service (DoS) attacks. For example, an attacker runs a tool that spoofs disassociation and/or deauthentication management frames from the access point.

These DoS attacks can be run in conjunction with other attacks. For example, if you have a WLAN using Lightweight Extensible Authentication Protocol (LEAP) for authentication, an assailant could spoof deauthentication messages to all the users connected to an access point in the hopes of capturing username and password hash combinations when the client devices reauthenticate. If some username and password hash combinations get retrieved, an offline dictionary attack is used to crack as many passwords as possible.

LEAP is covered later in the chapter, but the attack just described is why LEAP should not be used anymore. Coverage of LEAP has been included solely for historical and educational reasons.

Vulnerabilities Inherent to Mobility

The freedom offered by mobility is why we love wireless technologies; however, when it comes to WLANs, the same mobility that is the primary driver for adopting the technology also creates some security challenges.

One of the big challenges in an enterprise is figuring out how to handle roaming end users securely. Wireless clients regularly leave their association with one access point to reassociate with another access point. These wireless clients cannot just reassociate; they must be reauthenticated and generate new encryption keys. This means that the wireless client devices must carry some kind of security context with them so that the system can support fast reauthentication and rekeying if you want to avoid adversely affecting latency-sensitive applications like voice.

There are other problems inherent to mobility that are less related to technology and more related to end-user behavior. Suppose that you've deployed a secure WLAN in your enterprise, using the strongest authentication and encryption technologies available. You are confident in the strength of your WLAN security in your enterprise. But then, how do you secure the laptops of your road warriors when they connect to public hotspot WLAN networks in airports and coffee shops?

Consider what could happen when a senior executive in your enterprise uses her laptop to connect to an open network in an airport. There is probably some of your enterprise's important intellectual property and strategic information stored on that laptop in the form of documents, spreadsheets, PowerPoint presentations, and emails. You don't want an attacker compromising that valuable data. You also don't want that laptop catching a virus from another computer connected to the same wireless network. That laptop needs to be protected!

A similar problem arises with home WLAN networks. Home WLAN devices are very common these days. Usually, these are commodity devices from the local electronics megastore that don't always support the strongest security. Most end users aren't all that technical and don't pay much attention to security. These users get easily confused configuring security settings. Walk around any neighborhood with a WLAN sniffer and you will see that most home users don't give much attention to the physical containment problem either. Now when your enterprise users take their laptops home and connect to their home WLANs, how do you trust that the device is not vulnerable?

It can get worse too. A former neighbor of ours is a telecommuter, working as a marketing consultant for a large Fortune 500 company. His employer supplied him with a hardware Virtual Private Network (VPN) solution. Quite innocently, he decided it was a great idea to add a wireless access point so that he could enjoy working outside on nice days. And as you'd expect from a marketing consultant, his access point was configured with weak security. It never occurred to the neighbor that computers connected behind the

hardware VPN client have a free ride onto his employer's corporate network through the VPN tunnel.

Misconfigured Wireless Devices and Clients

The previous section reviewed some of the security challenges inherent to mobility. Another issue that is often related is misconfigured wireless client devices. Client devices usually get misconfigured when users tinker with the client supplicant settings on their own, usually when they are trying to set up their home WLAN or connect to a public WLAN hotspot.

Wireless network devices, like access points, can get misconfigured too. We've been on a customer site where we (temporarily) crippled WLAN security during troubleshooting. This isn't necessarily a dumb thing to do; in this customer's case, we were troubleshooting issues with wireless client associations and were eliminating authentication and encryption as variables while working on RF issues. But it's pretty easy to forget to turn the security back on, especially after an all-night troubleshooting session!

Enterprise-class WLAN implementers can be presented with a dizzying array of configuration options, especially when it comes to some of the authentication and encryption settings. Even the most experienced network manager can make mistakes and inadvertently leave the network exposed in some way.

Rogue Access Points and Devices

Consider an enterprise with a wireless network deployment utilizing Extensible Authentication Protocol–Transport Layer Security (EAP-TLS) authentication and Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP) for privacy. Don't worry if these concepts are foreign to you because they'll be explained shortly. Suffice it to say, for now, that this is a very strong authentication and encryption approach.

Now suppose though, that in some of the buildings, the access points are placed improperly so that some labs and conference rooms along the building periphery get poor radio coverage and users have difficulty connecting to the wireless network, and when they do connect, they experience very bad performance. Take it as axiomatic that people love the freedom of wireless mobility, so the poor end-user experience in the conference rooms and labs creates an unintended incentive for employees to deploy "rogue" access points.

Some employee, almost inevitably and quite innocently, will bring a cheap, commodity access point from the local electronics superstore into one of the conference rooms or labs with poor coverage, find a free Ethernet jack, and deploy an unauthorized, rogue access point, probably with weak security at best.

Clearly, this represents a catastrophic network security hole. The conference room or lab locations along the periphery of the building almost guarantee that the access point radio signals will be accessible from outside the building. Attackers frequently look for poorly

secured WLANs to exploit, and it doesn't take much in the way of technical skill to find them. If and when an unauthorized user associates to the rogue, the user has free access to the enterprise network and can do all sorts of nefarious things.

In this example, the authorized wireless network is securely implemented with strong authentication and encryption. But while there is no glaring weakness with the official wireless network, there is a serious wireless security problem!

This example illustrates what Cisco has often called the "frustrated insider" rogue access point. These are rogue access points deployed by insiders out of frustration because of no wireless access or rotten WLAN performance.

There's an entire different class though of "malicious attacker" rogue access points and devices. These are rogue wireless devices implemented by the bad guys for the singular purpose of compromising your network. It's not hard to imagine a parasitic attacker tailgating an employee in your enterprise to bypass building security, then finding an available Ethernet jack and deploying a rogue access point that he can later exploit from outside the building.

There is also software readily available that can turn any computer with a wireless network interface card into a software-based access point. Attackers use these software-based access points to entice wireless clients to connect to them. After a wireless client connects, the attacker attempts to trick the wireless client into giving up valuable information, or else the attacker compromises the client device in some way. This attack vector is especially effective in public hotspot environments.

Readily Available Profiling and Attack Tools

So far, you've learned about the vulnerability characteristics of the radio transmission medium, vulnerabilities in the standards definitions, vulnerabilities introduced by mobility, and the challenge of rogue access points.

All the problems are exacerbated by the proliferation of profiling and attack tools on the Internet that exploit the basic vulnerabilities in WLANs. Many of these tools are very easy to get started and not hard to use. There are bootable Linux CDs that include all the latest tools and client card drivers that make running these attacks "chimp simple."

Addressing the WLAN Security Challenges

The security challenges presented by 802.11 WLANs can seem daunting; however, do not despair, because there are solutions! Based on experience, WLAN deployments are usually more secure than their parallel, wired networks. Now that you've been introduced to the major vulnerabilities in 802.11 WLANs, this section takes a look at the solutions.

Table 4-1 documents a mapping between the major WLAN security challenges and the solutions.

Table 4-1 *WLAN Security Challenges and Solutions*

WLAN Security Challenge	WLAN Security Solution(s)
Vulnerabilities inherent to the radio transmission medium	Strong authentication and privacy
Vulnerabilities inherent to the standards definitions	Strong authentication and privacy, Management Frame Protection
Vulnerabilities inherent to mobility	Fast, secure roaming; secure management and policies
Misconfigured wireless devices and clients	Secure management and policies
Rogue access points and devices	Rogue AP management and wireless intrusion prevention systems, secure management and policies
Readily available profiling and attack tools	Secure management and policies

The sections that follow take a more in-depth look at the solutions.

Background on Strong Authentication and Privacy

If you think about the challenges presented by the radio transmission medium and the standards definition, the most logical and only practical mitigation strategies introduce strong authentication and data privacy through encryption. The need for strong authentication and data privacy extends to roaming clients, which must be reassociated and reauthenticated quickly while securely preserving data privacy. But what are the specific requirements for strong authentication and privacy? To answer that question, you need to consider how the 802.11 WEP-based security model is flawed.

How WEP Encryption Works

Previously, this chapter established that WEP is broken on the cryptographic front and shouldn't be used; however, it's useful to look briefly at how WEP works to establish some baseline knowledge for later in the chapter.

The WEP encryption is based on the symmetric RC4 cipher algorithm developed by Ron Rivest at RSA Security Inc. A symmetric cipher algorithm uses the same encryption key for encryption and decryption. Figure 4-1 illustrates the entire WEP processing model for a packet.

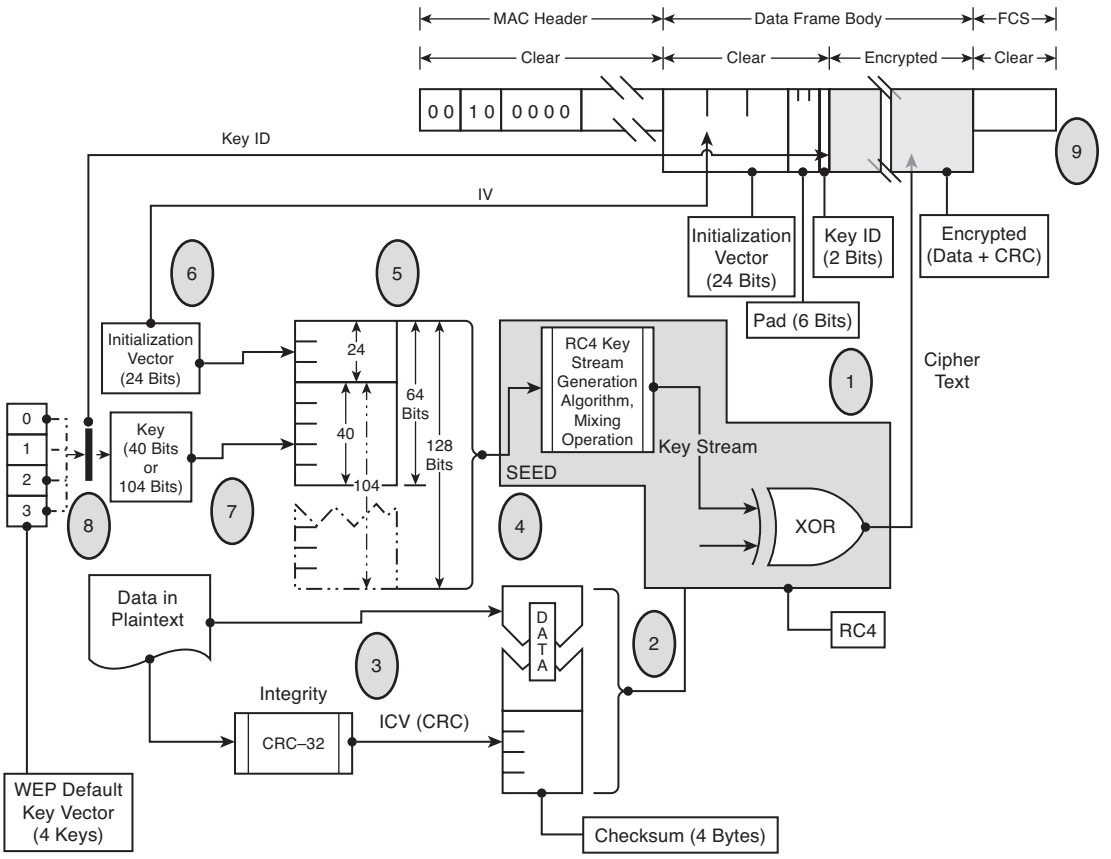


Figure 4-1 *WEP Processing Model*

There are a couple of things to note about RC4 first. RC4 encrypts data one byte at a time. This is because RC4 is a stream cipher, as opposed to a block cipher, which operates on chunks of data. For each byte of data input into the encryption algorithm, an encrypted byte of data is output. Note also that the RC4 algorithm is reversible. Inputting plain text and the encryption key into the RC4 algorithm yields cipher text. Inputting the cipher text and the same encryption key into the RC4 algorithm yields the plain text.

Notice how the WEP key is prepended with an IV, or initialization vector, prior to being fed into the RC4 algorithm. The IV is a 24-bit value that is different per packet, ensuring a different encryption key per packet. Note that we said *different* instead of *unique*. You can never truly have a unique encryption key per packet in the WEP processing model because the WEP key portion is static and there are only 2^{24} IVs available. Eventually, the IV space will be exhausted and the WEP processing model needs to start reusing IVs. Also, the encryption key per packet cannot be unique because there will be multiple

client devices using the same connection. Each client has its own IV sequence, meaning that multiple clients could use the same IV.

The IV is not kept secret because the receiver needs to know which IV has been used to encrypt the packet in order to decrypt the packet. In fact, as Figure 4-1 shows, the IV is transmitted in the clear.

The combined IV and WEP key are fed into the RC4 algorithm. Specifically, the combined value is used to seed the RC4 *key stream generator*. The purpose of the key stream generator is to output a pseudorandom sequence of bytes, called the *key stream*, that are used to scramble the plain text. Remember though, that every time an identical IV and WEP key pair are input into the RC4 algorithm, the same key stream will be generated.

The plain text is a combination of the data in the MAC protocol data unit (MPDU), which is a chunk of data from a higher-layer application and a 4-byte Integrity Check Value (ICV). The ICV is a value computed over the MPDU data using the CRC-32 algorithm. The ICV is intended to provide protection against message tampering in transit. That's why the CRC-32 value is computed before encryption.

The combined data and ICV are then encrypted by the RC4 algorithm using the key stream. The encryption mechanism used by the RC4 algorithm is simply the bitwise exclusive OR (XOR) operation. The plain text is XORed with the key stream, and the result is considered the cipher text.

The cipher text is inserted into the data frame body along with the clear text IV and the key ID of the WEP key used for this particular packet. Remember that the 802.11 specification calls for up to four WEP keys, so the receiver needs to know which key to use for decryption. The MAC header is prepended to the data frame body, and the frame check sequence (FCS) is computed. Next, the entire frame is handed down the protocol stack to be transmitted by the radio.

When the frame is received on the receiving end, the process is essentially reversed. The IV is stripped out of the frame's data body, the key ID is used to select the correct WEP key, and the IV and WEP key are combined and fed into the key stream generator. The encrypted cipher text is extracted from the frame body and XORed with the key stream, yielding the plain text. The ICV is computed over the data portion and compared to the ICV computed by the sender. If it matches, the data in the frame has not been tampered with and it is accepted. If the ICV does not match, the frame is rejected because the system assumes that the data has been modified in transit.

So now that you understand the WEP processing model, the section that follows examines how the 802.11 specification uses WEP and why it's broken!

How WEP Is Broken

To determine why WEP isn't a good security solution, you need to look at some practical problems and then look at the cryptographic problems with WEP.

For now, ignore the cryptography and consider just the practical aspects of using static WEP for data privacy. Remember that WEP keys have the following properties:

- **They are static:** They can't be changed except by reconfiguring all access points and stations.
- **They are shared:** All access points and stations share the same WEP keys.

The static property of WEP keys creates a management headache when it comes to key distribution. It isn't much of a challenge to configure WEP keys in a small office or home network when there are only one or a few access points and a few end stations. If there are many access points and end stations, however, this is a major problem. Configuring WEP keys isn't just a problem when devices are initially provisioned; as you are about to learn, WEP keys should be frequently rotated to mitigate some cryptographic weaknesses.

Now, think about the shared key property. Basically, this means that every WLAN device on the network has the capability to decrypt any other device's encrypted frames. Maybe that doesn't matter in some deployments, but then again, it probably does in others. Do you want your colleagues decrypting and reading some of your email messages?

Those are just some of the practical problems intrinsic to the specification. There are some major cryptographic problems with WEP, too. As previously noted, as 802.11 WLAN technology started to take off, a lot of smart people in the cryptographic community took a good long look at WEP as a security mechanism and identified catastrophic flaws.

We'll keep it simple here and not go into all the details of birthday paradoxes, bit-flipping, and weak IVs. The reference material at the end of the chapter provides that information. In a nutshell, the big cryptographic problems center around several areas: IV choice, reuse, transparency, flaws in the integrity-checking mechanisms, and weak RC4 keys.

802.11 Authentication

In the base 802.11 specification, there are basically two kinds of authentication: open authentication and shared-key authentication. Figure 4-2 illustrates open authentication.

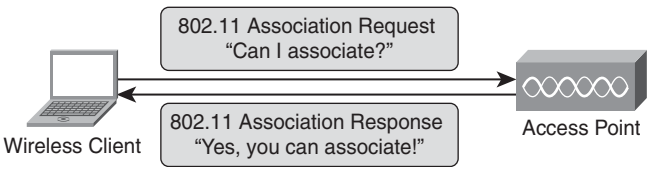


Figure 4-2 *802.11 Open Authentication*

As you can see, open authentication is exactly that—completely open. It isn't designed to authenticate the end-user device in any way. As long as the association request includes a valid service set identifier (SSID) on the access point, the device is allowed access to the network. When open authentication is used with static WEP encryption, there is some access control because each end-user device needs to know the WEP key to transmit any data after associating with an access point.

You might ask, why would a standards body allow free and open access to the WLAN? Well, there are many applications where open access to the network satisfies business objectives. Many universities view the WLAN as an open resource and thus provide no authentication or access control. Open authentication is sometimes used for guest access solutions in enterprises and supplemented with some kind of portal-based access control. It's also used in most public hotspots, typically supplemented with the service providers' access and authentication framework layered on top of the 802.11 open authentication.

Now what about shared-key authentication? Figure 4-3 illustrates shared-key authentication.

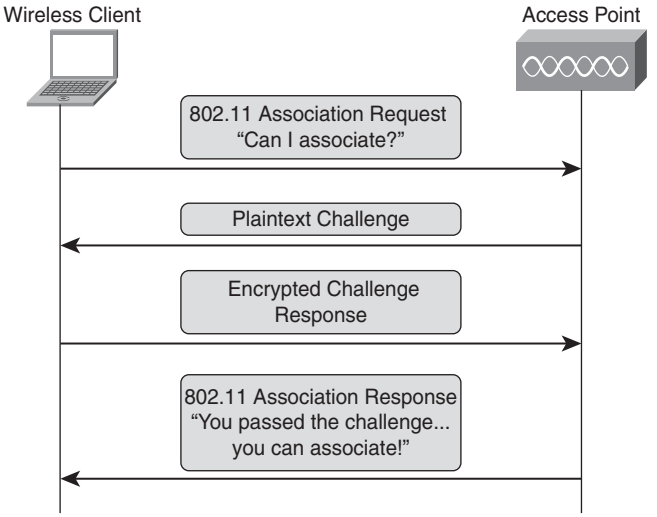


Figure 4-3 *802.11 Shared-Key Authentication*

As you can see from Figure 4-3, the authentication is provided by a challenge, challenge-response mechanism. The access point sends an arbitrary 128-bit value as challenge text to the station requesting access. The station encrypts the challenge text using the WEP key and sends this as the challenge response. If the WEP key is shared correctly between the AP and the end-user device, the AP should be able to decrypt the challenge response and compare it to the challenge text. If it matches, the station is allowed access. Simple enough.

If you take a closer look, however, there's a mutual authentication deficiency. The AP is authenticating the station, but what allows the station to trust that the challenge response

is from the correct AP? The answer is...nothing! The station has no way of knowing whether the challenge text comes from a legitimate AP or an AP impersonator.

But it gets worse. An eavesdropper watching this transaction just saw a plain-text message (the challenge) and its corresponding cipher text (the challenge response). Now, if the eavesdropper takes the plain text and XORs it with the cipher text, she gets the key stream corresponding to that IV!

Unfortunately, the 802.11 specification allows for IV reuse, so if the vendor implementation is sloppy, the eavesdropper might be able to use the recovered material to authenticate herself and start sending packets into the network. Worse, the next time the eavesdropper sees a frame with that IV, she can use the recovered key stream to decrypt the cipher text by simply XORing the cipher text with the key stream. If she's patient, she can build enough of a key stream database to decrypt a lot of cipher text.

As you can see, shared-key authentication is worse than no authentication because it provides a false sense of security.

Oh, and maybe you noticed something else about both open and shared-key authentication. Curious? There is no user-based authentication. Device-based authentication, like 802.11 open and shared-key authentication, does not prevent unauthorized users from accessing networked resources from authorized machines. Think of the problem presented by lost or stolen assets!

Addressing the Strong Authentication and Privacy Challenges

After having reviewed the issues with the base 802.11 specification and WEP for authentication and privacy, a set of basic requirements for real authentication and privacy emerges:

- User-based authentication
- Mutual authentication
- Dynamic, per-session, and per-user cryptographic keys
- Larger IV space than 2^{24}
- Stronger cryptographic algorithms than RC4
- Backward compatibility with RC4-only-capable client devices
- Strong message integrity checks
- Scalability and manageability

These are the basic requirements for strong authentication and privacy addressed by the IEEE 802.11i task group charter. The work of the 802.11i task group resulted in a new 802.11 amendment that defines the following:

- **Two types of networks:** The Transition Security Network (TSN) and the Robust Security Network (RSN)

- **New data privacy and integrity models:** Temporal Key Integrity Protocol (TKIP) and CCMP
- An authentication framework based on IEEE 802.1X and EAP
- Dynamic cryptographic key management

TKIP includes techniques to protect cryptographic keys through key mixing and packet counters and a Message Integrity Check (MIC) algorithm. TKIP is designed to be backward compatible with legacy WEP equipment.

CCMP is an algorithm for data privacy and integrity based on the Advanced Encryption Standard (AES). CCMP is stronger than TKIP but is not backward compatible with legacy equipment that cannot support AES.

The IEEE 802.1X standard defines an algorithm for port-access control that requires client devices to authenticate before being given network access. The authentication is accomplished through the EAP and RADIUS protocols. The key management algorithm is designed to generate dynamic cryptographic keys to address the weaknesses in WEP.

An RSN is a network that allows only TKIP and/or CCMP. A TSN allows both RSN and WEP machines.

While the IEEE 802.11i task group was working on completing the 802.11i standard, the Wi-Fi Alliance adopted Wi-Fi Protected Access (WPA) as an industry standard. WPA was based on the parts of the 802.11i standard that were available and uncontroversial at the time. After 802.11i was ratified, though, the Wi-Fi Alliance adopted the full document as its WPA Version 2 standard.

In a nutshell, WPA includes TKIP, 802.1X authentication, and dynamic key management. WPAv2 adds CCMP. Both WPA and WPAv2 have the following modes:

- **Personal mode:** Allows pre-shared keys for authentication
- **Enterprise mode:** Requires 802.1X authentication

We will look at the enterprise mode here.

Cisco literature often uses a four-ingredient model, as illustrated by Figure 4-4, to abstract basic wireless security requirements:

- **The authentication framework:** The scaffolding necessary for the authentication and encryption algorithms
- **The authentication algorithm:** Provides a secure way of validating user credentials
- **The data privacy algorithm:** The cryptographic mechanism for obfuscating data in transmission
- **The data integrity algorithm:** Protects messages in transit from tampering

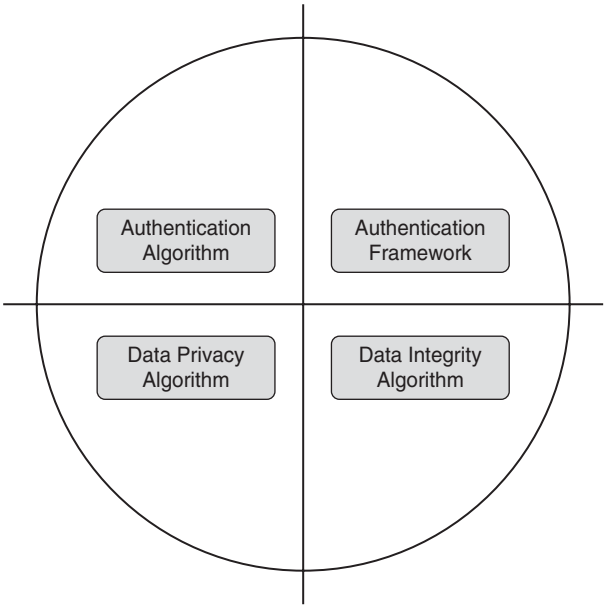


Figure 4-4 *WLAN Authentication and Data Privacy*

Authentication Framework

The authentication framework provides the scaffolding for the authentication and encryption algorithms and mechanisms by specifying the protocols and processes. In the raw 802.11 specification, the authentication framework is provided by the authentication management frame. This management frame facilitates the open and shared-key authentication algorithms, without doing the actual authentication.

802.11i specifies the use of the IEEE's 802.1X port-based access control specification for access control and the IETF's EAP authentication framework with RADIUS as the protocol for enterprise-grade wireless authentication.

EAP is an authentication framework defined in IETF RFC 3748 and RFC 3579 that is used to authenticate supplicants. After the supplicant is authenticated, the authenticator removes the port access restrictions and the authenticated device can begin transmitting and receiving data frames.

EAP, as defined in RFCs 2284 and 3579, is an authentication framework only. There are many different authentication algorithms based on the EAP framework, defined either in other RFCs or as proprietary protocols. The following section covers EAP types in more detail.

RADIUS is the protocol used to authenticate the supplicant. EAP over RADIUS messages, as defined in RFC 2869, transport the authentication messages between the authenticator and authorization server. In RADIUS terminology, the authenticator is the Network Access Server (NAS) or RADIUS client.

Figure 4-5 illustrates the 802.11i authentication framework.

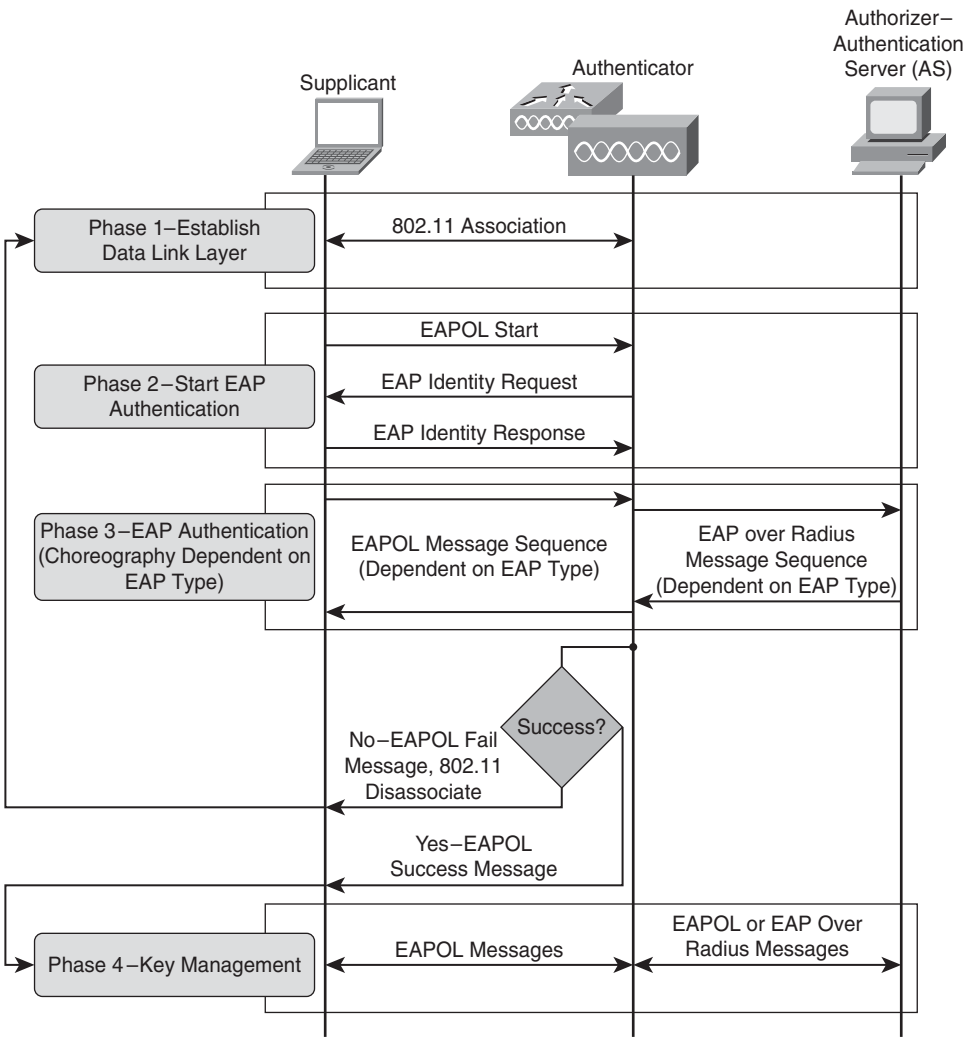


Figure 4-5 *802.11i Enterprise Authentication Framework*

Note that 802.1X authentication has three parties:

- **Supplicant:** The client device requesting access to network resources.
- **Authenticator:** The network unit that controls access to the network. It acts as an intermediary between the supplicant and the authentication server. In the CUWN, the WLAN controller is the authenticator.

- **Authentication server:** Grants or denies permission to the supplicant based on the authentication algorithm and user-based access credentials. The authentication server is the RADIUS server.

Initially, the wireless station (the supplicant) establishes data link layer connectivity through the standard 802.11 open authentication. Immediately then, the authenticator blocks further wireless station access to the network.

Next, the client typically issues an Extensible Authentication Protocol over LAN (EAPoL) start message to begin the authentication process. The authenticator, the WLC, responds with an EAPoL Identity request.

In the next phase, the device identity is authenticated through an exchange of EAP and RADIUS messages. We've deliberately abstracted this phase at this point because the mechanics vary by EAP algorithm. Suffice it to say, though, that in this phase, some type of mutual authentication is necessary to establish two-way trust. The network trusts the supplicant only after it is authenticated, and the supplicant can only trust the network after it authenticates the network in some way. The supplicant cannot implicitly trust the network because some attacker could be spoofing the network elements.

Assuming success in authentication, the key management phase follows. We'll look at key management in more detail in the context of data privacy. During key management, a dynamic encryption key is negotiated for the particular session.

Now finally, after full authentication and session key derivation, the authenticator lifts the 802.1X port restrictions and the supplicant is allowed access to the network.

Authentication Algorithm

Recall in the previous sections that the authentication framework includes the use of the EAP, but the authentication algorithm performs the actual authentication of devices based on some type of credentials. The authentication algorithm is defined by an EAP type, of which there are many. Neither 802.11i nor the Wi-Fi Alliance specifies an EAP type, but the Wi-Fi Alliance currently certifies interoperability with several EAP types as part of the WPA certification tests: EAP-TLS, EAP Tunneled Transport Layer Security (EAP-TTLS), Protected EAP (PEAP), and EAP Subscriber Identity Module (EAP-SIM). Additionally, there are Cisco-developed EAP types—Lightweight EAP (LEAP) and EAP with Flexible Authentication through Secure Tunneling (EAP-FAST)—that are widely adopted.

The sections that follow cover all of these EAP types, though some in greater detail than others, starting with the Cisco-developed EAP types.

LEAP

Prior to the ratification of 802.11i and the WPA certifications, Cisco introduced a proprietary EAP type called Lightweight Extensible Authentication Protocol (LEAP). Some literature refers to LEAP as Cisco-EAP.

Initially, LEAP support was limited to Cisco Aironet–branded client cards and supplicant software. Cisco drove wide adoption of LEAP into the market through the Cisco

Compatible Extensions (CCX) program by making LEAP a mandatory part of the CCX version 2 specification.

Let's look at the choreography of LEAP in Figure 4-6 because it illustrates some important things.

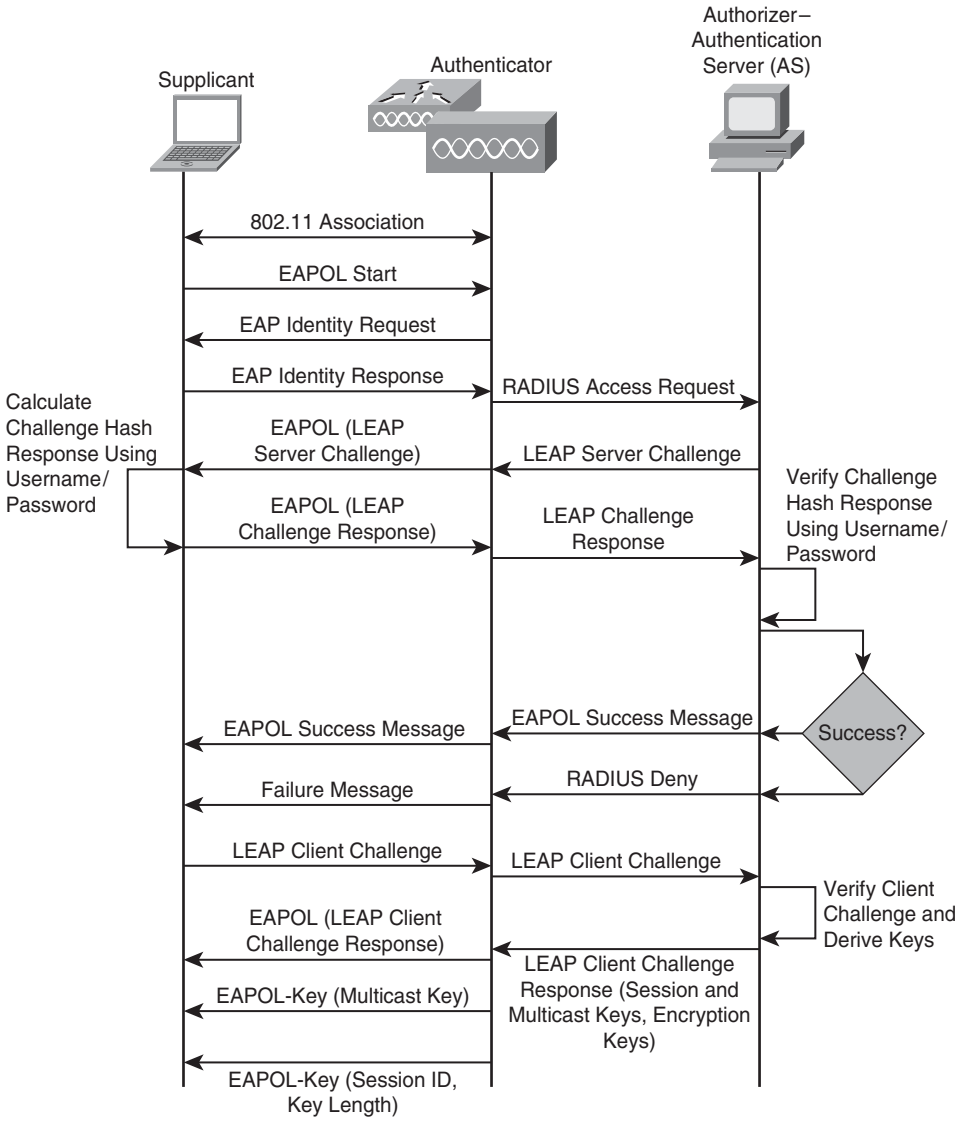


Figure 4-6 *Lightweight EAP*

As shown in Figure 4-6, LEAP has several important characteristics. It includes mutual authentication and is password based. It is also *lightweight* in the sense that all that is

really needed to deploy LEAP is support for it on the three parties of the 802.1X authentication model.

But notice also in Figure 4-6 how there is nothing in LEAP's choreography to obfuscate the mutual challenge/challenge-response messages. In 2004, a tool called ASLEAP was released that exploited weaknesses in the hashing mechanisms in those challenge/challenge-response messages to recover user credentials. ASLEAP attacks can be mitigated with strong passwords, but that adds more management overhead and isn't totally effective.

So LEAP is now deprecated and should not be used. But the wide adoption of LEAP was a watershed industry development. LEAP was the first EAP type defined specifically for WLANs. It also provided a viable alternative to WEP at a time when this alternative was desperately needed and there was not a standards-based alternative. Most importantly, it proved the viability of 802.1X with EAP as an authentication framework for WLANs before standards ratification.

EAP-FAST

EAP-FAST is an acronym for EAP with Flexible Authentication through Secure Tunneling. Cisco has published the EAP-FAST specification as IETF RFC 4851.

EAP-FAST preserves the lightweight characteristic of LEAP but provides a mechanism to protect the exchange of credentials during authentication. EAP-FAST uses something called a *Protected Access Credential (PAC)* as a unique shared-secret between the supplicant and server. The PAC is used during the EAP-FAST algorithm to establish a TLS tunnel that protects the exchange of user authentication messages. The tunnel establishment is called the *outer authentication*; the user authentication inside the tunnel is called in the *inner authentication*.

Figure 4-7 illustrates the EAP-FAST choreography.

During EAP-FAST Phase 0, the PAC is provisioned on the supplicant using the Authenticated Diffie-Hellman Protocol. The supplicant can subsequently securely store the PAC, obviating Phase 0 for future authentications.

During Phase 1, the PAC is used for mutual device authentication that results in a session tunnel key. This session key is used to secure the authentication exchanges in Phase 2.

Finally, in Phase 2, the user authentication is performed in the TLS tunnel established during Phase 1.

Now, it's important to note that the devil is in the details of EAP-FAST implementation. When automatic PAC provisioning is enabled, an attacker can intercept the PAC and subsequently use that to compromise user credentials.

There is also a potential attack vector, where an attacker deploys a rogue AP offering service to clients. During Phase 1, the attacker rejects the PAC during Phase 1, triggering a return to the PAC-provisioning phase. Many supplicants are configured to prompt the user to accept a new PAC. If the target in this attack accepts the new PAC, the attacker can compromise user credentials when EAP-FAST moves to Phase 2 on the rogue network. Strong rogue AP management and proper supplicant configuration mitigate this attack.

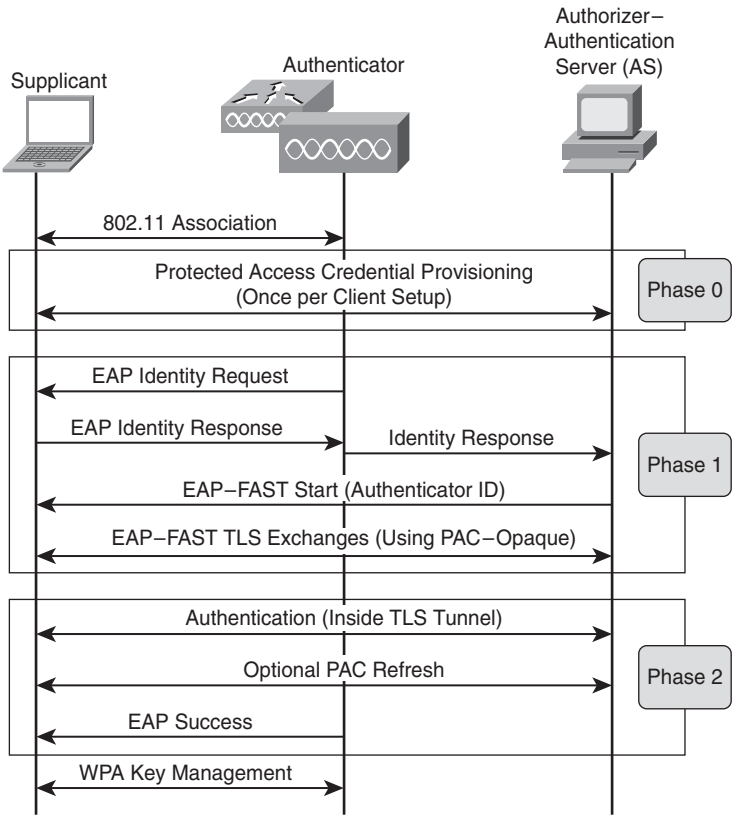


Figure 4-7 *EAP-FAST*

The PAC is issued on a per-user and per-authentication server basis, so if a new user logs in to the network from a device, or if a device is logging in to a part of the network serviced by a different RADIUS server, a new PAC file must be provisioned. This does present some obvious challenges from a management perspective in light of the potential attack vectors just discussed.

EAP-FAST allows manual PAC provisioning and server certificate use during the PAC provisioning phase to mitigate the threats and management challenges.

PEAP

Protected EAP (PEAP) is similar to EAP-FAST in that it establishes a TLS tunnel to protect authentication messages; however, PEAP uses a server certificate for outer authentication and to negotiate the cryptographic keys necessary for the TLS tunnel. Figure 4-8 illustrates the choreography of PEAP.

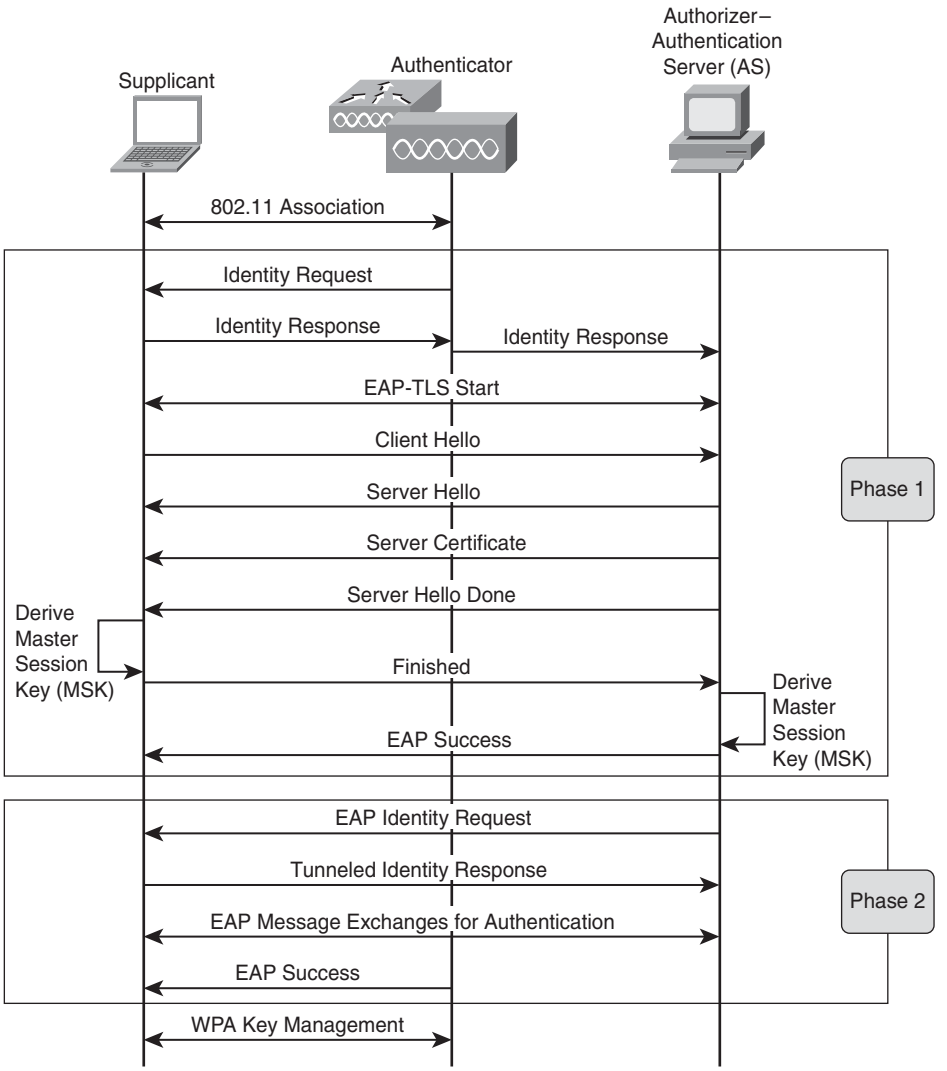


Figure 4-8 *PEAP*

PEAP inner authentication in the TLS tunnel comes in two flavors:

- **PEAP-MSCHAPv2:** PEAP-MSCHAPv2 is the most common version of PEAP, so common in fact that usually when people use the term *PEAP*, they are referring to this version of PEAP. PEAP-MSCHAPv2 uses the Microsoft Challenge Handshake Authentication Protocol version 2 (MSCHAPv2) for the inner authentication method.
- **PEAP-GTC:** GTC is an acronym for Generic Token Card, which represents the inner authentication method, which uses hardware token cards to generate one-time passwords for users.

EAP-TLS

RFC 5216 defines EAP-TLS. Like PEAP, EAP-TLS uses certificate exchanges for outer authentication in establishing a TLS tunnel. But, as shown in Figure 4-9, EAP-TLS adds client-side authentication: The client device must present a certificate that is authenticated by the server.

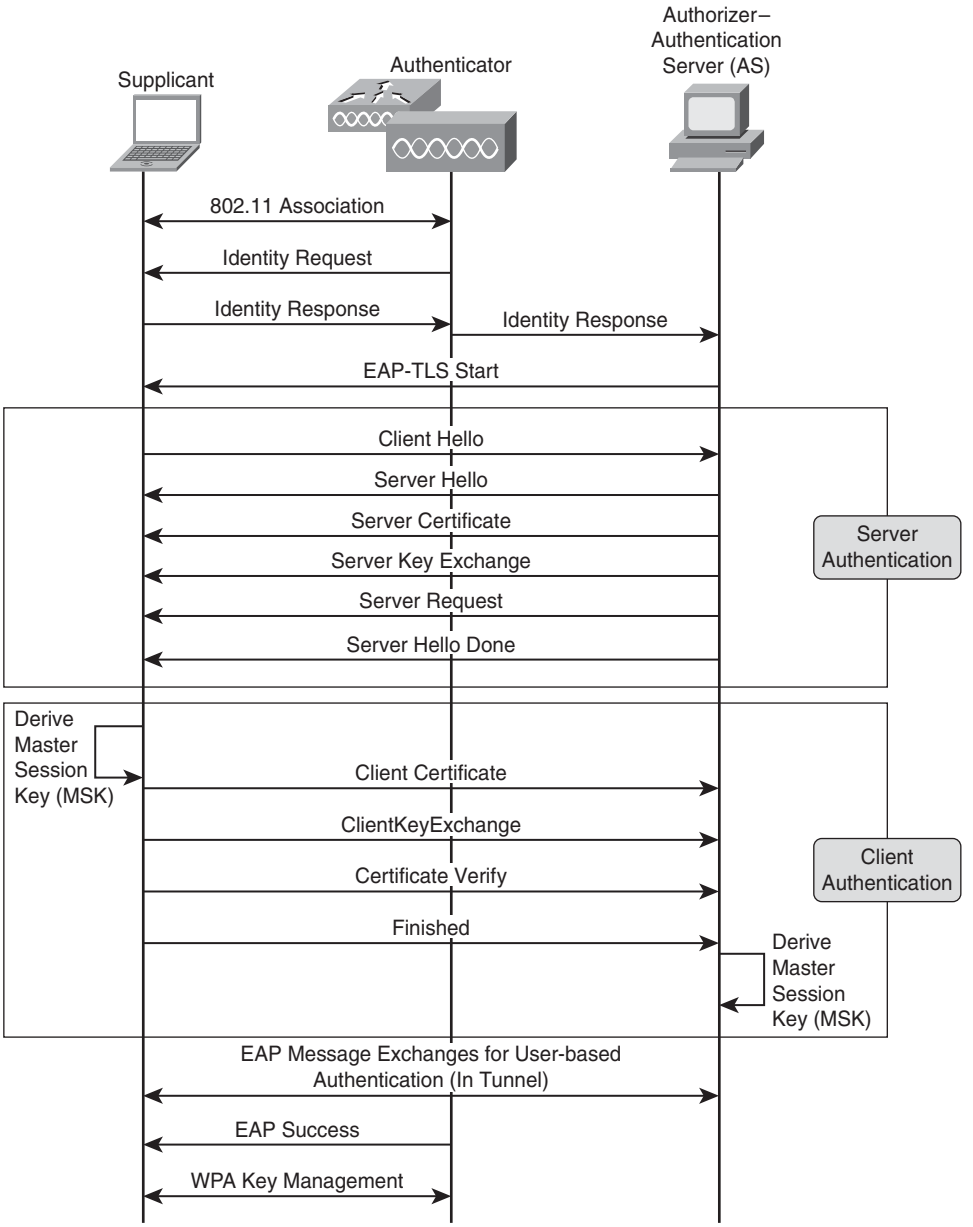


Figure 4-9 *EAP-TLS*

The power of this mutual certificate exchange is that it mitigates theoretical man-in-the-middle attacks to which other authentication types might be vulnerable.

After the mutual exchange is completed, the inner authentication takes place, leading up to the keying phase.

EAP-TTLS

RFC 5281 defines EAP-TTLS as an authentication algorithm extending the functionality of EAP-TLS to a number of legacy inner authentication methods. It is listed here for completeness because the Wi-Fi Alliance includes EAP-TTLS in its WPA interoperability certifications.

EAP-SIM

RFC 4186 defines a protocol for authentication and session key derivation through Global System for Mobile Communications (GSM) Subscriber Identity Modules (SIM) infrastructure. The choreography of EAP-SIM is out of the scope of this book but is listed because it is included in the Wi-Fi Alliance WPA test suite.

Selecting an EAP Type

After all this discussion of EAP choreography, which EAP type should you use? The answer is, it depends.

One of the decision criteria is what EAP types are supported on your client devices. At the time of this writing, the latest versions of Microsoft Windows have native support for PEAP, EAP-TLS, LEAP, and EAP-FAST (with the appropriate patches). The latest versions of Mac OS have support for LEAP, PEAP, EAP-FAST, EAP-TLS, and EAP-TTLS.

If your device's OS does not support the EAP type you select, you'll have to install additional supplicant software. You might decide to do this anyway because add-on supplicants often have additional features you want. Sometimes add-on supplicants just work better than native OS supplicants.

The other major decision factor customers wrestle with is how much management overhead to incur. Each EAP type requires some setup and maintenance, and there are usually trade-offs between manageability and security.

For example, EAP-TLS is probably the most secure of the EAP types covered; however, it's also the most difficult to manage because it requires a PKI infrastructure and certificates on every client device. You might decide that a lighter-weight authentication algorithm such as PEAP or EAP-FAST is good enough.

Data Privacy and Integrity

The previous section looked at how the 802.11i and WPA standards address the problem of missing access control and authentication in WEP. This section looks at how the standard addresses encryption key management.

Remember that with WEP, there were two big problems with key management:

- A big management challenge distributing WEP keys.
- The keys are shared between all devices in the WLAN.

The 802.11i standard addresses these problems by calling out a mechanism for dynamic encryption key generation.

Before going into detail, consider the two kinds of keys:

- **Master keys:** Used to generate transient keys. The master key exists as long as a supplicant's authentication is valid, so it can exist across multiple 802.11 associations.
- **Transient keys:** Change with each session.

The two phases of the key management process are

- Master key establishment
- Key exchange

With master key establishment, the master key is shared between station and authenticator and is subsequently used to negotiate keys that are used to encrypt data. There are two types of master keys:

- **Pairwise master key (PMK):** Used for unicast traffic; there is a PMK instance for each supplicant.
- **Group master key (GMK):** Used for multicast and broadcast traffic; group keys are shared among all stations and the AP.

When using 802.1X/EAP authentications, the PMK is established following mutual authentication between the supplicant and authentication server. Figure 4-10 illustrates the key management phase.

In Figure 4-10, you can see that following mutual authentication, the PMK is established between the supplicant and the authentication server. The supplicant and the authentication server derive the PMK separately. The authentication server then hands the PMK to the authenticator. The 802.11i specification does not dictate how the PMK is passed to the authenticator, although it does recommend EAPoL; WPA on the other hand, specifies that this must be done through RADIUS.

After the PMK is handed off to the authenticator, the supplicant and authenticator negotiate a pairwise transient key (PTK) through a cryptographic four-way handshake. As you can see in Figure 4-10, the four-way handshake begins with the authenticator and supplicant each generating a cryptographic nonce value. In the first message then, the authenticator sends its nonce to the supplicant. As shown in Figure 4-11, the supplicant uses the two nonces and the PMK to generate the PTK.

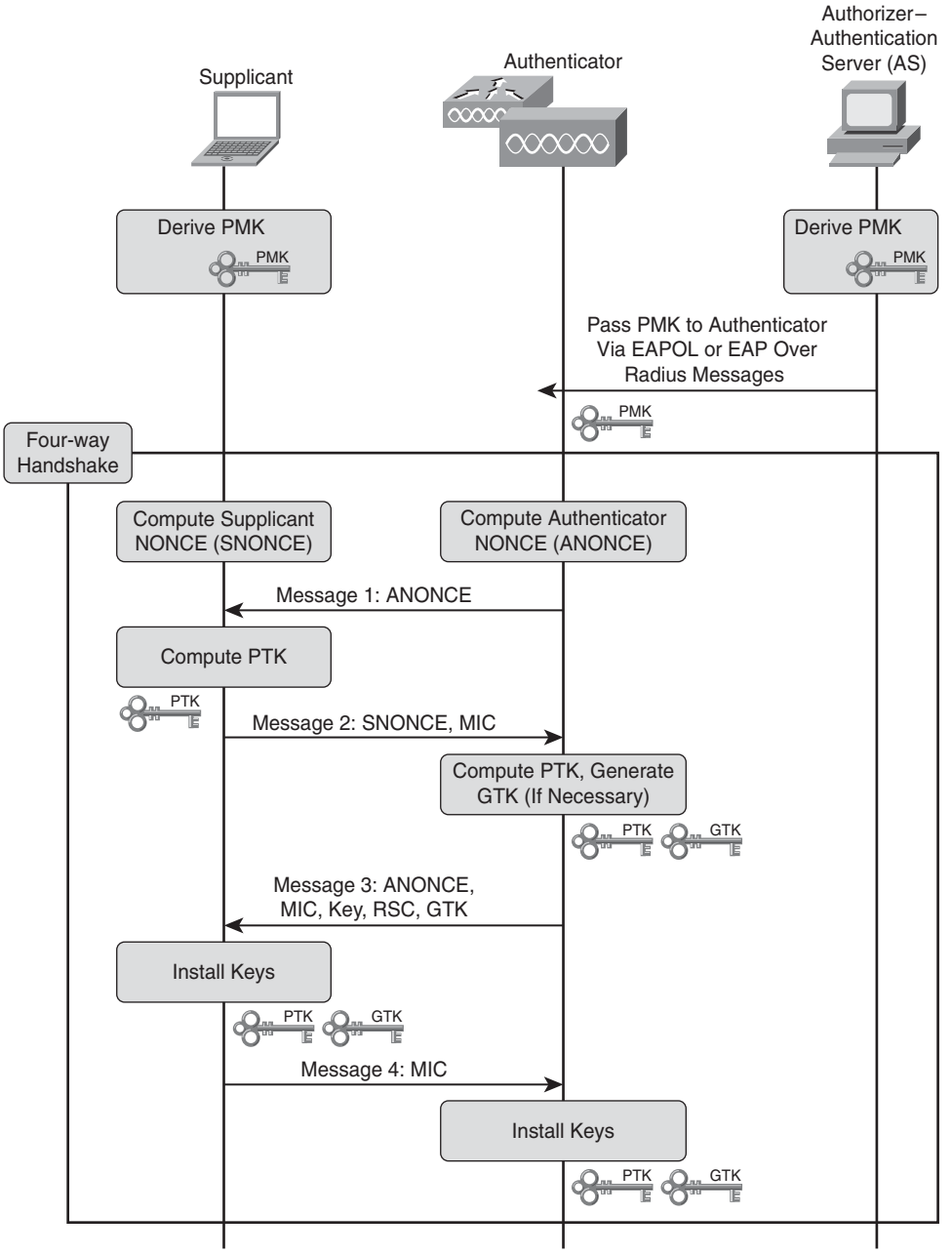


Figure 4-10 *Key Management Phase*

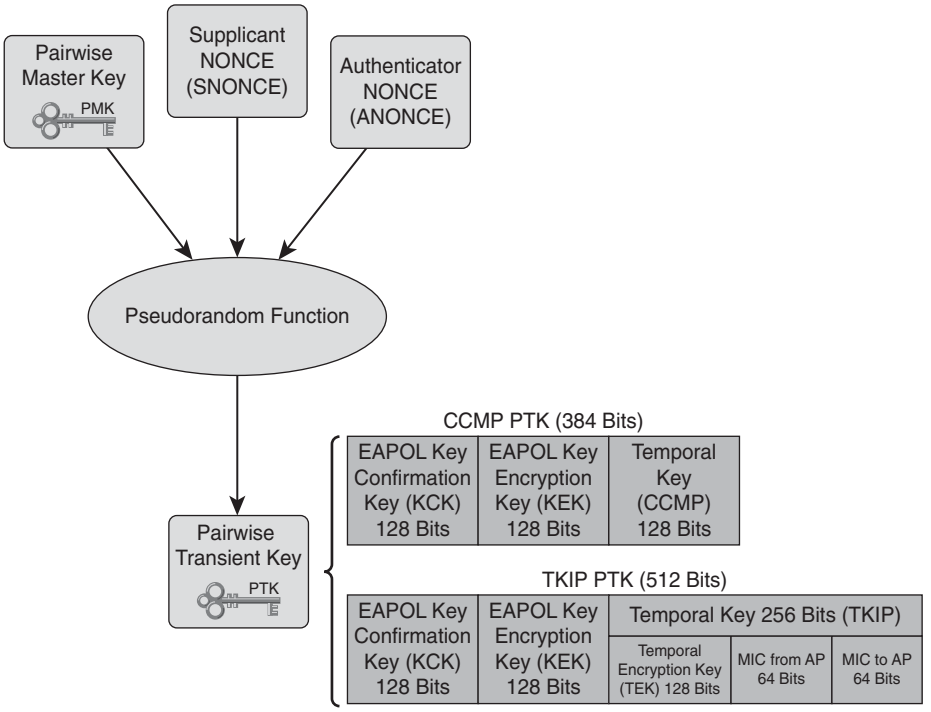


Figure 4-11 *Pairwise Transient Key Generation*

In Figure 4-10, you can see that after generating the PTK, the supplicant sends a message back to the authenticator with its own nonce and a message integrity check (MIC) that proves the supplicant knows the PMK. After verifying the MIC, the authenticator generates the PTK using the methodology shown in Figure 4-11.

The authenticator will generate a group transient key (GTK) from the GMK if necessary. The GTK is used for encrypting multicast and broadcast traffic. Figure 4-12 illustrates the process for generating the GTK.

In the third message of the four-way handshake illustrated in Figure 4-11, the authenticator sends its nonce, a MIC based on the PMK, the receive sequence counter (RSC), and the GTK. The RSC is the current sequence counter for the GTK. The GTK requires a sequence counter to prevent an attacker from replaying a broadcast message. After the supplicant verifies the MIC, it installs the PTK and GTK. It then sends the confirmation message to the authenticator that includes a MIC. After the authenticator confirms the MIC, it installs the PTK and GTK. At this point, the wireless station can begin sending and receiving data. The data is encrypted with part of the PTK.

Take a moment to revisit the generation of the GTK. Figure 4-12 illustrates the process for generating the GTK from the GMK.

An obvious question is, where does the GMK come from? The answer is that the authenticator generates the GMK. The authenticator then uses the GMK to generate the GTK.

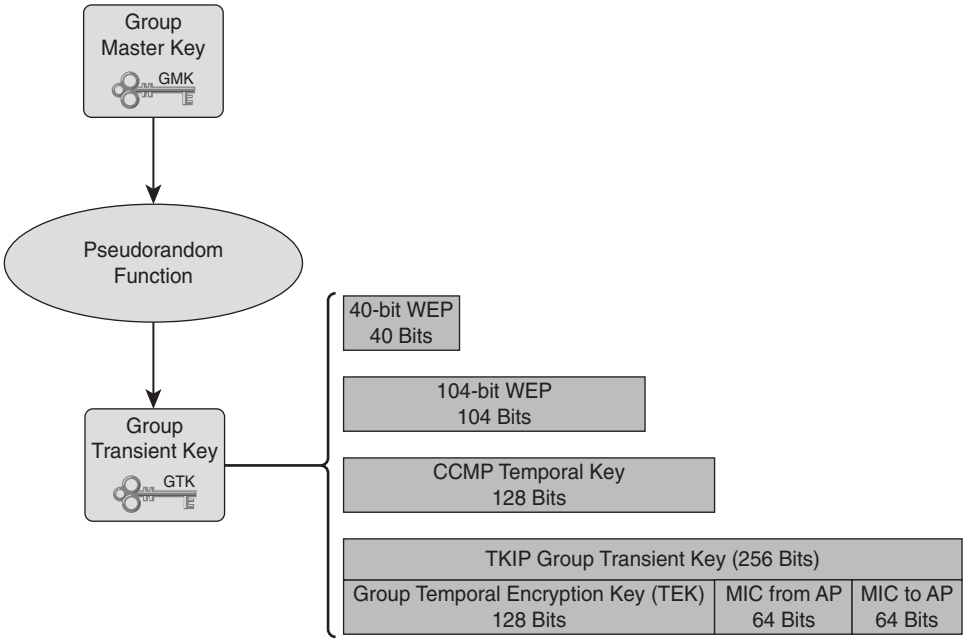


Figure 4-12 *Group Transient Key Generation*

As you probably have deduced, all devices associated to an AP share the GTK. The GTK will be regenerated in the following situations:

- When a MIC failure occurs (more on that in the next section)
- When a client disassociates or is deauthenticated
- After a specified interval

When a new GTK is generated, it is sent in an encrypted message to each associated wireless station. This is a two-way handshake process, as illustrated in Figure 4-13.

You should have noticed in Figures 4-11 and 4-12 that there is both a pairwise and group key hierarchy and that this hierarchy is different for CCMP and TKIP. The sections that follow examine CCMP and TKIP in greater detail. For now, note that the PTK is made up of several keys:

- EAPoL Confirmation Key
- EAPoL Encryption Key
- Temporal Key

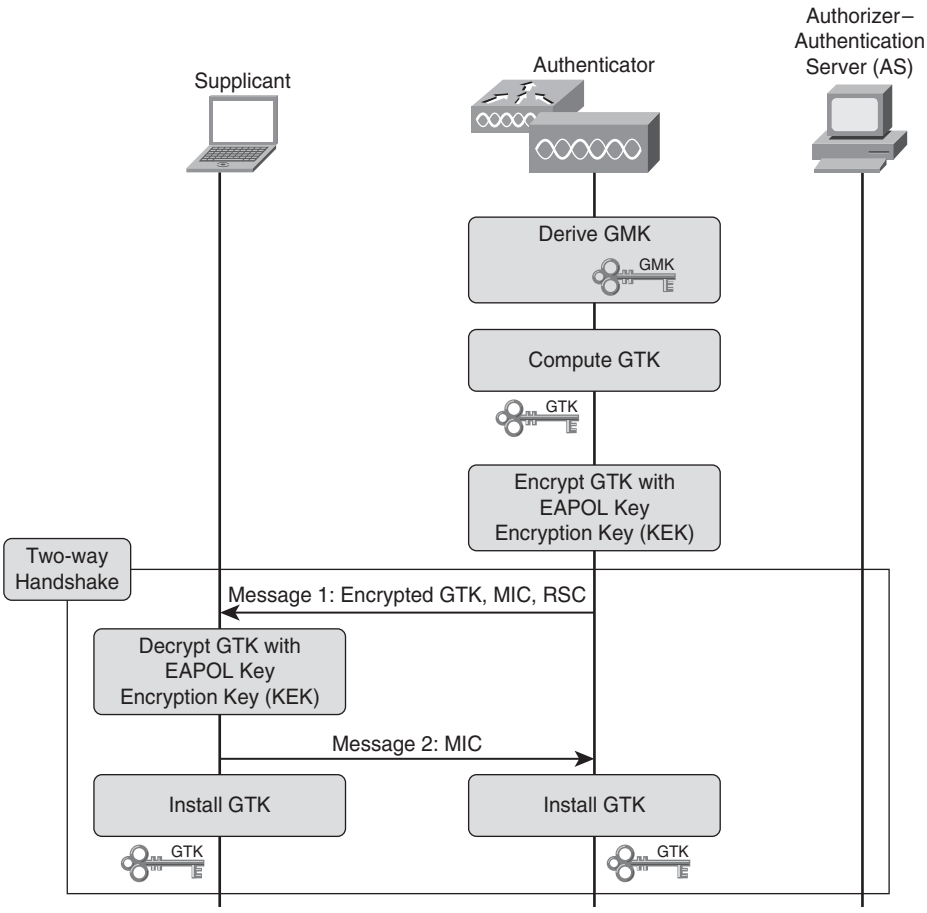


Figure 4-13 *Group Transient Key Two-Way Handshake*

In the case of TKIP, the Temporal Key is actually a Temporal Encryption Key and the MICs from the four-way handshake. The Temporal Encryption Key is used in the data encryption and decryption. The EAPoL keys are used to secure transactions across the air whenever encryption keys need to be sent from the authenticator to the supplicant.

Temporal Key Integrity Protocol (TKIP)

At this point, you should understand how the standards bodies have addressed WEP deficiencies when it comes to authentication and key management. Now it's time to look at how the cryptographic flaws have been addressed. This section looks at TKIP in detail, while the next section examines CCMP.

Recall that the basic cryptographic problems with WEP centered on the following items:

- IV choice and reuse
- ICV flaws
- "Weak" RC4 keys

The design challenges before the standards bodies for TKIP were that TKIP had to "fix" the WEP cryptographic deficiencies and at the same time had to work with legacy 802.11 equipment. Basically, this meant that TKIP had to work within the mechanical framework of WEP using RC4.

The TKIP designers settled on a solution that involves three protocols to supplement WEP:

- Michael MIC
- TKIP Sequence Counter (TSC)
- Key-mixing algorithm

Michael MIC is a message integrity check mechanism to augment the WEP ICV. Remember that the WEP ICV failed to protect against bit-flipping. TKIP uses the Michael hashing algorithm instead that is not vulnerable to bit-flipping. Essentially, the Michael MIC is a hash value calculated from a 64-bit key derived from one-half of the lower-order 128 bits of the PTK (see Figures 4-11 and 4-12) and a padded MAC service data unit (MSDU) from the network layer. The padded MSDU is the MSDU with some additional fields, including the source and destination MAC addresses and some additional special octet values. This padded MSDU is used only for the Michael MIC calculation.

The Michael MIC was selected because it is computationally inexpensive. The 802.11i task group felt that this was important because the algorithm needed to run on legacy hardware with limited CPU horsepower. As such, Michael MIC is a trade-off between cryptographic strength and computational complexity. The 802.11i amendment itself calls Michael MIC "weak protection against active attacks."

So the 802.11i amendment calls out some countermeasures. Specifically, 802.11i requires that all Michael MIC failures be logged and that, in the event that two Michael MIC failures (on either the AP or the client) occur within one minute, both the station and AP are to disable all packet reception and transmission and the AP is to deauthenticate and disassociate all wireless clients. The ICV is always checked before the MIC to make it harder for an attacker to create deliberate MIC failures, which in itself could be a DoS attack.

The TKIP Sequence Counter (TSC) is used to strengthen the IV and prevent replay attacks on the system. The TSC is a 48-bit counter that starts at 0 and is incremented with each packet.

The TSC protects against replay attacks in two ways:

- Each device records the highest TSC value it has seen for a MAC address. If a packet arrives with a TSC value less than or equal to the current highest TSC value, the packet is discarded because the receiver assumes that the packet has been replayed.
- The TSC is used in the key-mixing algorithm for both encryption and decryption algorithms, so if the TSC is tampered with in transit, the ICV and MIC will fail and the packet will be discarded.

Figure 4-14 illustrates the TKIP packet format.

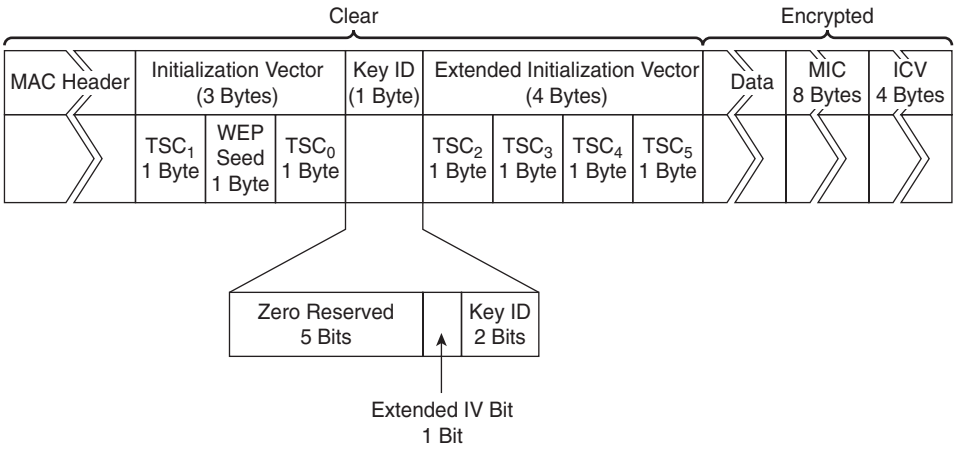


Figure 4-14 *TKIP Packet Format*

In Figure 4-14, you can see that the 48 bits of the TSC are split up into 6 bytes, TSC_0 through TSC_5. The low-order bytes, TSC_0 and TSC_1, are joined with a specially crafted byte to create a standard 24-bit IV that is designed to avoid FMS weak keys. Bytes TSC_2 through TSC_5 are placed in a new field called the Extended IV field. A bit is set in the Key ID byte to indicate that the Extended IV is in use. We will see shortly how all 6 bytes of the TSC are used in the key-mixing algorithm.

The TSC protects against IV reuse because a TSC is never repeated for a given key; when a new key is generated, the TSC is reset to 0. So even if an attacker were to retrieve a key stream for a packet, the key stream would be worthless.

The third protocol that TKIP introduces is a key-mixing algorithm that ensures a unique per-packet 128-bit WEP key generated from the Temporal Encryption Key (TEK). As shown in Figures 4-11 and 4-12, the TEK is the high-order 128-bits of the PTK.

Figure 4-15 illustrates the TKIP key-mixing algorithm.

In Phase 1, the most significant 32 bits of the TSC are "mixed" with the transmitter's 48-bit MAC address and the high-order 80 bits of the TEK. The output is an 80-bit TKIP Transmit Address and Key (TTAK).

The TTAK is "mixed" with the entire TEK and TSC in Phase 2. The output is a 128-bit WEP key that is used to seed the RC4 keystream generator. The first 24 bits of the 128-bit key are the IV constructed from the least significant TSC bytes and the specially crafted byte, as shown previously in Figure 4-14.

TKIP provides strong protection, but the 802.11i task group wanted to do better. Particularly important was to have a solution that satisfies the U.S. government FIPS-140-2 certification requirements and that calls for AES encryption. That solution is CCMP.

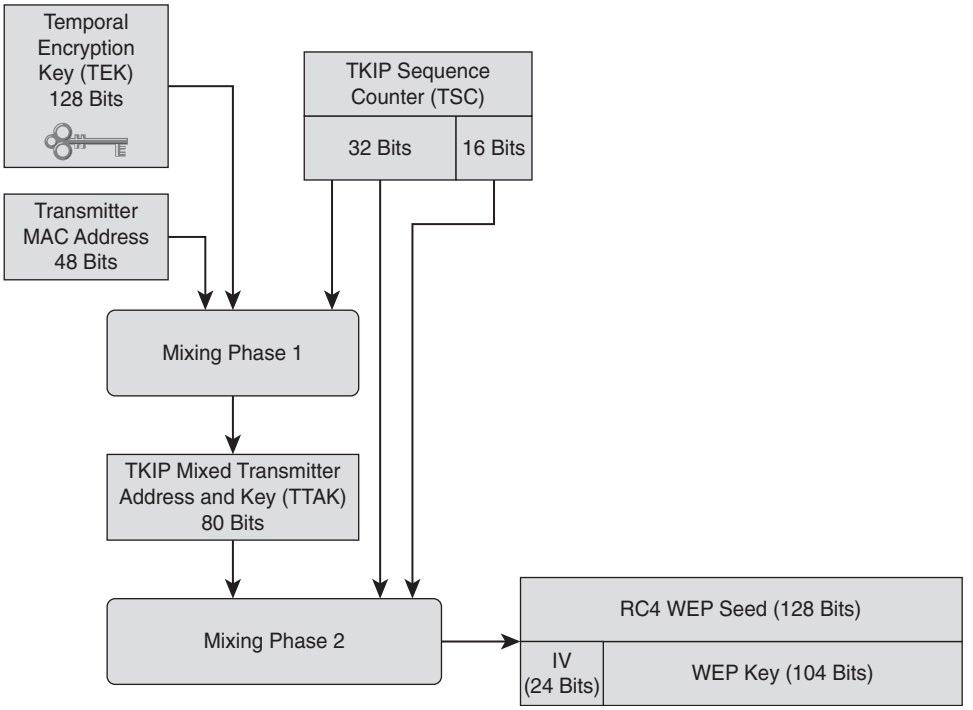


Figure 4-15 *TKIP Key-Mixing Algorithm*

Counter/CBC-MAC Protocol (CCMP)

CCMP is the strongest confidentiality, integrity, and replay protection protocol suite available for WLAN security. It is based on AES and is the core of the 802.11i RSN and the WPAv2 specification. CCMP is thus the preferred solution if all the WLAN devices in the system can support it.

An in-depth discussion of AES is beyond the scope of this chapter. We will just look at it in the abstract as it is applied to CCMP. The *Counter* part of the CCMP name is from an AES *counter mode* that provides the data privacy. The *CBC-MAC* part of the name comes from the *Cipher Block Chaining Message Authentication Code* that is used as a message integrity check.

WEP and TKIP rely on the RC4 stream cipher for encryption. In contrast, AES is a block cipher. A block cipher operates on chunks of data as opposed to a stream of bytes.

Figure 4-16 illustrates how AES counter mode works to encrypt plain text.

The plain text is chopped into 128-bit chunks. A 128-bit counter is AES encrypted and then XORed with the first 128-bit chunk of plain text. The counter is then incremented and the process repeated on the next 128-bit chunk. The process is repeated until the entire plain text is exhausted. Then the 128-bit chunks of cipher text are concatenated to produce the full encrypted text.

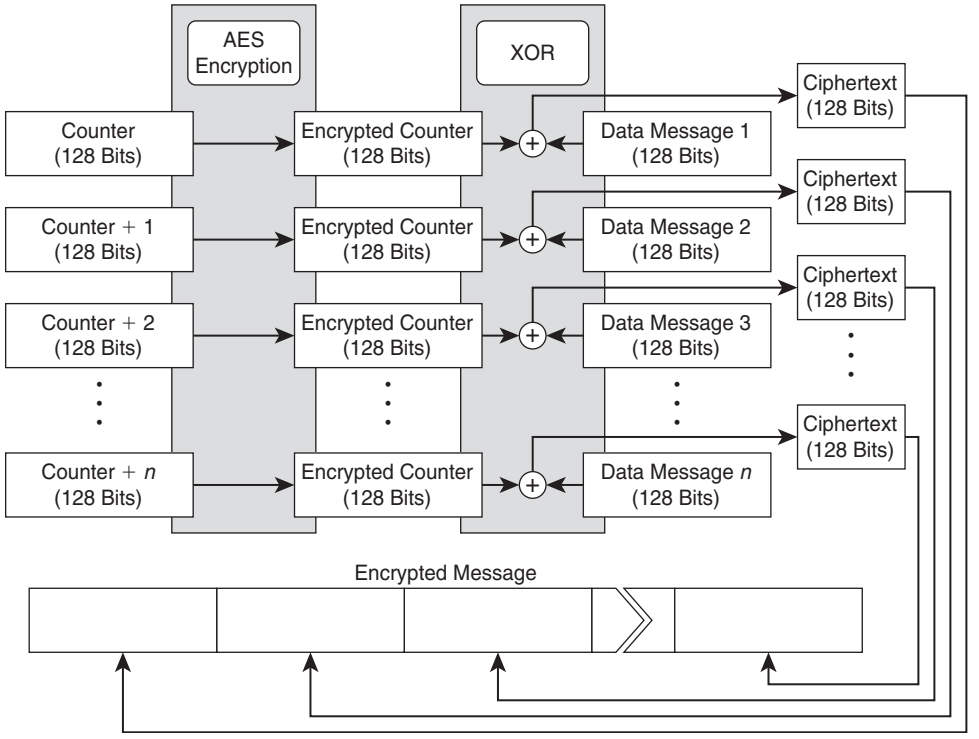


Figure 4-16 *AES Counter Mode*

Now, the decryption of this encrypted text is really simple. The decrypting entity chops up the encrypted text into 128-bit chunks. It then uses the same algorithm shown in Figure 4-16, starting with the same counter value. But instead of XORing the encrypted counters with plain text, the encrypted counters are XORed with the 128-bit chunks of cipher text. The output is the decrypted chunks of plain text. These chunks are then reassembled as the full plain-text message.

Figure 4-17 illustrates the AES CBC-MAC mode.

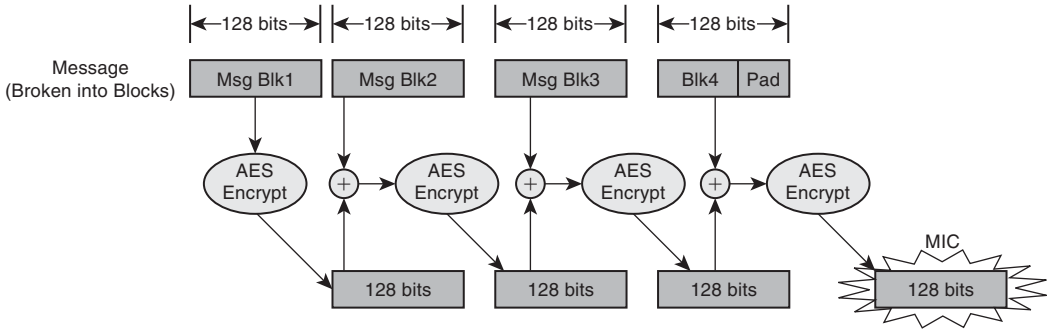


Figure 4-17 *AES CBC-MAC Mode*

As you can see from Figure 4-17, the plain text is again broken into chunks. The first chunk is AES encrypted to produce a chunk of cipher text. This chunk of cipher text is then XORed with the next block of plain text and then AES encrypted. The process is then repeated until the plain-text chunks are exhausted. The final value is the MIC.

So now that you understand the mechanics of the CCM algorithm, the next step is to see how it is applied. Figure 4-18 shows the CCMP packet format.

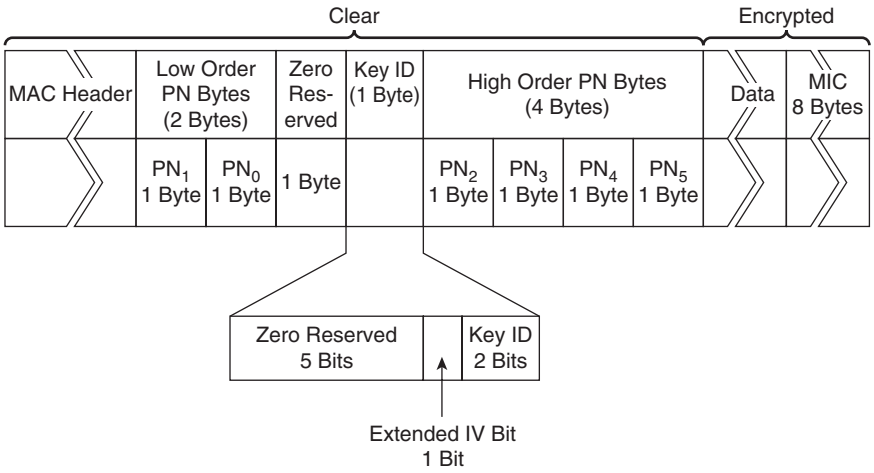


Figure 4-18 *CCMP Packet Format*

What you'll notice is that this looks a lot like the TKIP packet format. Instead of the TSC, though, CCMP uses a 48-bit packet number (PN) that is split into two parts. The Extended IV bit is also set in the KeyID field.

The MIC is a 64-bit value computed with CBC-MAC mode. The plain-text input into the CBC-MAC mode is the plain-text data plus the parts of the MAC header that cannot be changed in the event of a retransmission. The parts of the MAC header included in the MIC calculation are called the *Additional Authentication Data (AAD)*.

Figure 4-19 shows the CCMP encapsulation process.

You've already seen how the CCM algorithm encrypts the data using AES counter mode and computes the MIC through CBC-MAC. The encryption key input into the AES encryption algorithm is the CCMP Temporal Key derived from the PTK.

Remember that AES counter mode needs a counter value that is unique per packet. This counter value is the nonce shown in Figure 4-19. The nonce is assembled from the priority and source address fields from the MAC header and the packet number.

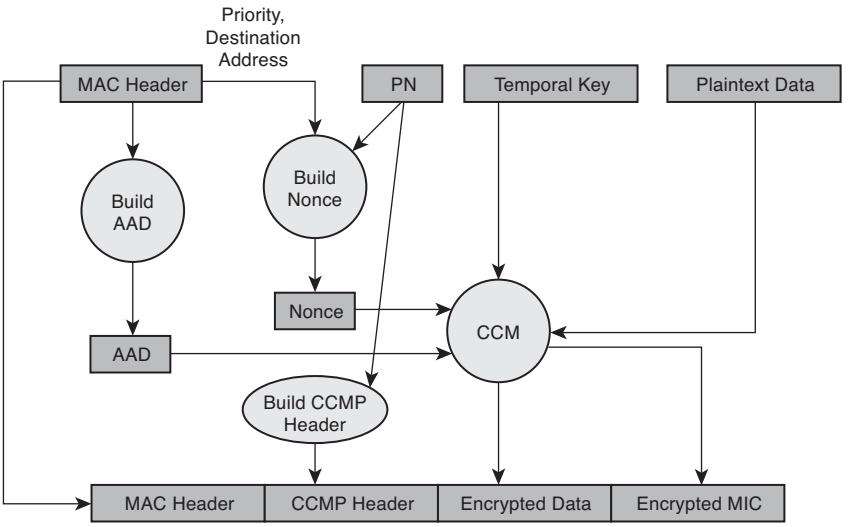


Figure 4-19 *CCMP Encapsulation Process*

Figure 4-20 shows the CCMP decapsulation process.

When a packet is received, the receiver extracts the PN and compares it against the value it has stored. If the PN has increased, the receiver knows that the packet has not been replayed.

The receiver knows the CCMP Temporal Key and can compute the nonce from the MAC and CCMP headers. It can then run the AES counter mode algorithm to decrypt the cipher text. As we've already seen, AES counter mode decryption is essentially the same as AES counter mode encryption, except that the operations are performed on the cipher text.

The MIC is then computed and compared to the decrypted MIC value to verify that the packet has not been tampered with.

Alternative Approaches to Authentication and Data Privacy

There are, of course, alternative approaches to WPA/WPAv2 Enterprise. A full 802.1X/EAP authentication solution is overkill for most home and many small-office WLAN implementations. The IEEE standards and WFA committees recognized this from the beginning and so allow an implementation using pre-shared keys (PSK). In a PSK implementation, there is a shared secret—the pre-shared key (PSK)—between the wireless station and authenticator. This shared secret is used as a passphrase to negotiate per-session encryption keys with the authenticator, as described in the key management sections.

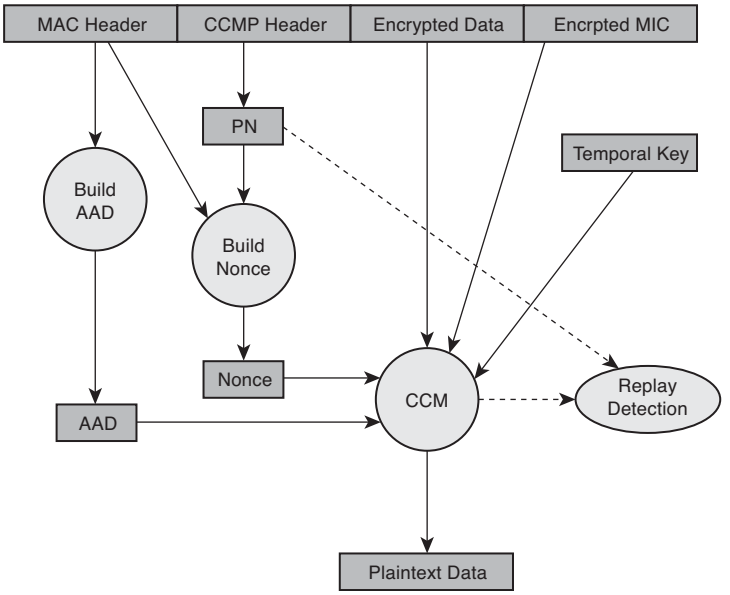


Figure 4-20 *CCMP Decapsulation Process*

We'd be remiss if we didn't at least mention overlay options as an approach to securing WLANs. Overlays typically allow 802.11 open authentications to the wireless network and leverage back-end mechanisms for user authentication and data privacy and to protect networked resources. Some customers, usually universities, don't bother with user authentication and data privacy, treating the WLAN as a public resource.

Some WLAN implementers simply don't trust anything other than VPNs, or for some reason cannot or won't implement 802.1X authentication. Typically, VPN solutions treat the 802.11 network as completely untrusted, treating the WLAN as a logically separate entity from the rest of the network. Usually, the WLC is firewalled off from the rest of the network and the APs reside on their own VLAN infrastructure. Traffic aggregated at the controller is directed to the VPN concentrator. You can see how this creates a lot of complexity in the wired network, which is why VPN overlay solutions have mostly been deprecated in favor of the wireless standards-based approaches.

Another overlay variant is to use browser redirects for web authentication, either through the native capabilities on the controller or through an external web-authentication box. These architectures were discussed in the context of guest access.

Rogue Access Point Detection and Wireless Intrusion Prevention

Remember the earlier discussion of rogue access points and devices and the other potential vulnerabilities presented by readily available profiling and attack tools? A comprehensive wireless security approach usually requires some type of rogue device detection and wireless intrusion prevention system (WIPS).

The most crude detection technique is "sneaker net." An IT staffer laces up his or her "sneakers" and walks the campus with a WLAN sniffer or other reconnaissance tools, looking for unauthorized wireless devices and potential attackers. This isn't a bad approach for small deployments, where it's usually pretty effective and good exercise too!

However, sneaker net just doesn't scale to a large or geographically dispersed deployment. Furthermore, it's pretty easy to defeat and pretty low yield for the amount of effort. One large enterprise we frequently work with used sneaker net to find rogues. At first, the IT staff found the sneaker net technique pretty effective. But after a few months, they began to find the technique was pretty much good only as physical exercise. It was a lot of effort for a pretty low yield; they rarely found rogues but still suspected there were many of them out there connected to their network.

After reevaluating their approach, they decided to take advantage of embedded rogue detection and location capabilities in their Cisco WLAN. Sure enough, as soon as they began using the embedded tools, many rogues were identified and located. One of the employees guilty of rogue deployments confessed that he had figured out the IT staff's rogue hunting schedule and simply removed the rogue until the IT staffer walked by!

All modern WLAN vendors including Cisco implement over-the-air rogue detection and WIPS using the entire WLAN system. Typically this involves sampling the RF environment at the AP radio interfaces and applying higher-level forensic analysis at the WLC or management software. For WIPS, this involves comparing traffic patterns against signatures that profile known attacks. For rogue AP detection, this involves collecting 802.11 beacons from the environment and culling out the beacons that are sourced from devices not belonging to the WLAN system. Can you see the problem with this approach, though?

We've run a WLAN sniffer in the Cisco office in midtown Manhattan and identified upwards of 350 different WLAN networks by simply capturing beacons. Each of these networks is probably a legitimate network belonging to someone. The beacons are just propagating out of their coverage area. You should be able to see that separating real rogues from other legitimate 802.11 networks is nontrivial, especially if you're trying to make the classification manually.

Cisco has tools that give you pretty granular location of rogue APs, and this is pretty helpful in classifying true rogues. Cisco has also implemented algorithms for identifying rogue devices as connected to the wired network and tracing the rogue to an actual switchport. When you can determine that a rogue AP is connected to your network, you can be pretty certain that it is in fact a rogue AP.

After you classify an AP as a rogue, Cisco gives you tools to suppress the rogue. The most effective of these techniques allows you to shut off the switchport the rogue is connected to. Essentially, this renders most rogues useless. Cisco also has the ability to do *over-the-air containment*. When the WLAN system sees legitimate wireless clients connecting to a rogue, 802.11 deauthentication frames with the rogue MAC source spoofed are sent to the client.

The CUWN has two tiers of offerings. Figure 4-21 shows the base-level rogue AP and WIPS solution.

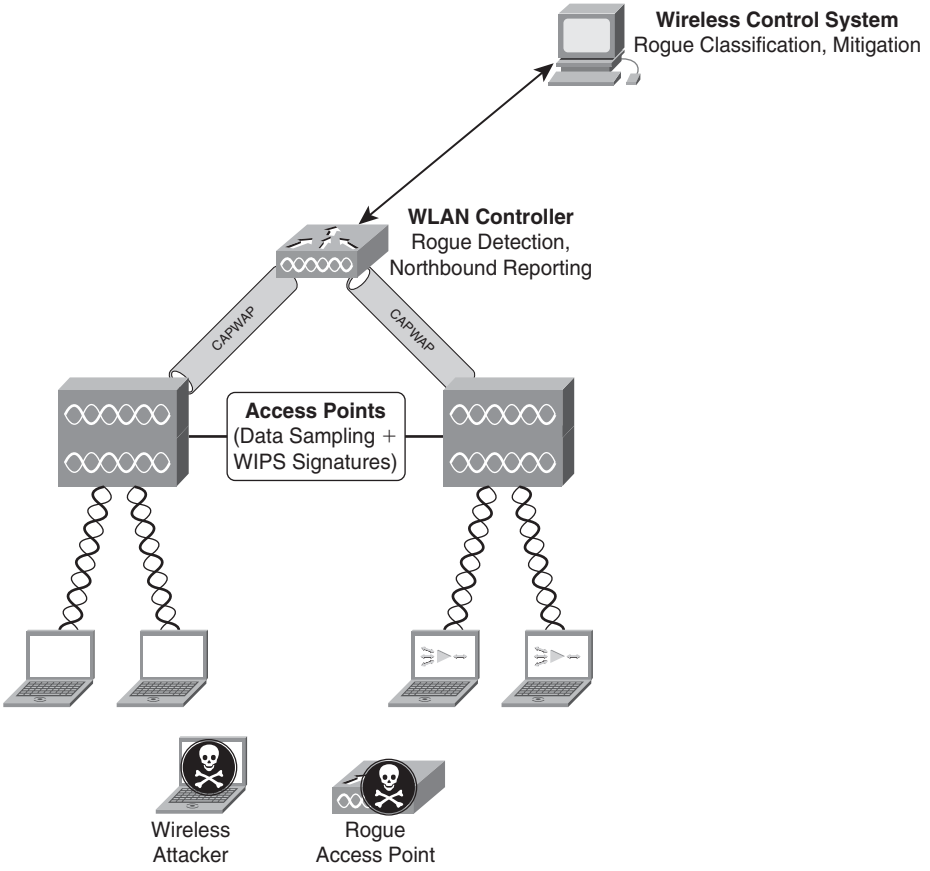


Figure 4-21 *Rogue AP and WIPS Base Solution*

In this base solution, the APs sample the environment during normal operation. APs have on-board WIPS signatures that, when triggered, create alerts that get forwarded through the WLC to WCS, where logic and rules are applied.

As far as rogue AP detection is concerned, the WLC collects sampled data from the APs and identifies potential rogue devices. The WCS takes this information and applies classification logic. After it is classified, the network administrator can initiate rogue AP containment strategies.

Figure 4-22 shows the more sophisticated solution with Cisco Adaptive WIPS running on the Mobility Service Engine (MSE) platform.

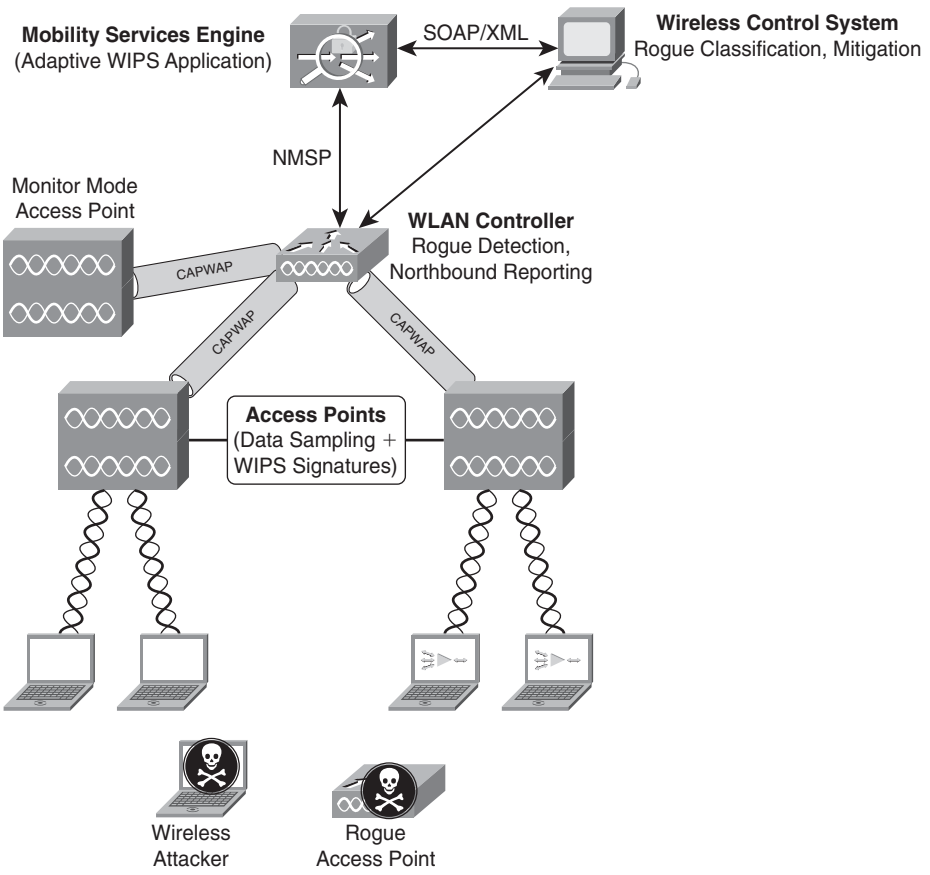


Figure 4-22 *Rogue AP and WIPS Base Solution*

This more advanced solution contains a superset of the base solution's functionality. There are more signatures and a more accurate and sophisticated classification engine running on the MSE. Also, this solution potentially uses dedicated *monitor mode* APs, dedicated to sampling the air, so there is significantly more data for the system to work with.

Further details on rogue AP management and WIPS are a chapter unto themselves. Furthermore, at the time of this writing, the details and nuances of both solutions were under analysis and discussion by the Cisco engineering teams, so any details we might include now might be obsolete when you read them. We suggest you work closely with your Cisco representatives for the details if you're interested in these topics.

Secure Management and Security Policies

It's always important to remember that any device in a network can be attacked. WLAN infrastructure devices are no different in this respect than routers and switches. It's always a good idea to follow sensible secure management practices.

A discussion on secure management practices is a chapter unto itself. We won't spend much time on it. Basically, you should try to follow these general security recommendations:

- Use centralized authentication and authorization (TACACS+ for example) for administrative device access if possible.
- Select secure management protocols such as SSH, HTTPS, and SNMPv3 over insecure protocols such as Telnet, HTTP, SNMPv1, and SNMPv2, if possible.
- When you can't use centralized authentication and authorization and/or secure management protocols, try to find other ways of securing access. For example, use obscure and strong passwords and SNMP community strings. You can also use access control lists (ACL) to restrict management access to specific hosts.
- Disable all unnecessary services on WLAN devices.
- Limit management connectivity to specific platforms using ACLs or an equivalent mechanism.

This is obviously not a comprehensive list, but you get the idea. There are some good papers referenced at the end of the chapter that will help you with secure management practices.

Protecting WLAN client devices is fundamentally a desktop management problem. The best thing you can do is make sure that your mobile users are running host-based security solutions like personal firewalls and maintaining current antivirus software. Furthermore, you can eliminate many of the problems related to misconfigured wireless clients by preprovisioning supplicant profiles for your end users. There's a little bit more work up front, but it's definitely worth it.

It's always a good idea to periodically audit your WLAN configurations to be sure that there aren't any unintentional security holes. Cisco offers tools to periodically check AP configurations and provide alerts when security policies are violated. The CUWN enforces security policies at the controllers, so policy monitoring and enforcement are de facto parts of the architecture.

Summary

There is a lot of information in this chapter about WLAN security. We examined the potential vulnerabilities in an 802.11 WLAN. The risks discussed were as follows:

- Vulnerabilities inherent to the radio transmission medium
- Vulnerabilities inherent to the standards definitions
- Vulnerabilities inherent to mobility
- Readily available profiling and attack tools
- Misconfigured wireless devices and clients
- Rogue access points and devices

We then looked in detail at how the 802.11 WEP standard was broken and how the newer 802.11i, WPA, and WPAv2 standards solve the WEP problems and allow strong authentication and data privacy through 802.1X/EAP and TKIP and CCMP. We also looked at overlay options.

The chapter concluded with a look at other aspects of WLAN security, including rogue device detection and mitigation, wireless intrusion detection and protection, secure management practices, techniques for protecting WLAN client devices, and security configuration policy monitoring and enforcement, as well as how to deal with RF containment challenges.

This has been a lot of information for sure, but you should now know enough about WLAN security to understand what it takes to secure a WLAN deployment.

References

For additional insight on wireless LAN security fundamentals, consult the books, RFCs, standards, white papers, and technical documentation listed in the sections that follow.

Books on WLANs, WLAN Security, and General Network Security

1. Shankar, K., Sundaralingam, S., Balinsky, A., and Miller, D. *Cisco Wireless LAN Security*. Indianapolis, Indiana: Cisco Press; 2005.
2. Fleck, B. and Potter, B. *802.11 Security*. Cambridge, Massachusetts: O'Reilly; 2002.
3. Edney, J. and Arbaugh, W. *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*. Reading, Massachusetts: Addison-Wesley; 2003.
4. Flickenger, R. *Wireless Hacks*. Cambridge, Massachusetts: O'Reilly; 2003.

5. Vladimirov, A.A., Gavrilenko, K.V., and Mikhailovsky, A.A. *Wi-Foo: The Secrets of Wireless Hacking*. Reading, Massachusetts: Addison-Wesley Professional; 2004.

6. Peikari, C. and Fogie, S. *Maximum Wireless Security*. Indianapolis, Indiana: Sams Publishing; 2002.

7. Swaminatha, T.M. and Elden, C.R. *Wireless Security and Privacy: Best Practices and Design Techniques*. Reading, Massachusetts: Addison-Wesley; 2002.

8. Roshan, P. and Leary, J. *802.11 Wireless LAN Fundamentals*, Indianapolis, Indiana: Cisco Press; 2003.

9. Convery, S. *Network Security Architectures*. Indianapolis, Indiana: Cisco Press; 2004.

10. Kaeo, M. *Designing Network Security*. Indianapolis, Indiana: Cisco Press; 1999.

11. O'Hara, B. and Petrick, A. *IEEE 802.11 Handbook*. Piscataway, New Jersey: IEEE Press; 2005.

Relevant RFCs and Standards Documentation

1. **RFC 2196**, "Site Security Handbook." Fraser, B. IETF. ftp://ftp.rfc-editor.org/in-notes/rfc2196.txt. September 1997.

2. **RFC 3579**, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)." Aboda, B. and Calhoun, P. IETF. ftp://ftp.rfc-editor.org/in-notes/rfc3579.txt. September 2003.

3. **RFC 3748**, "Extensible Authentication Protocol (EAP)." Aboda, B., Blunk, L., Vollbrecht, J., Carlson, J., and Levkowetz, H. IETF. ftp://ftp.rfc-editor.org/in-notes/rfc3748.txt. June 2004.

4. "Wi-Fi Protected Access: Strong, standards-based, interoperable security for today's Wi-Fi networks." www.wi-fi.org/membersonly/getfile.asp?f=Whitepaper_Wi-Fi_Security4-29-03.pdf. Wi-Fi Alliance. April 2003.

5. "Deploying Wi-Fi Protected Access (WPA™) and WPA2™ in the Enterprise." www.wi-fi.org/membersonly/getfile.asp?f=WFA_02_27_05_WPA_WPA2_White_Paper.pdf. Wi-Fi Alliance. March 2005.

White Papers on WLAN Security

1. "SAFE: Wireless LAN Security in Depth—version 2." Convery, S., Miller, D., Sundaralingam, S., et al. www.cisco.com/en/US/partner/netsol/ns340/ns394/ns171/ns128/networking_solutions_white_paper09186a008009c8b3.shtml.

2. "Wireless LAN Security White Paper." www.cisco.com/en/US/partner/products/hw/wireless/ps430/products_white_paper09186a00800b469f.shtml.

Technical Documents Describing WEP Flaws

1. "An Inductive Chosen Plaintext Attack Against WEP/WEP2." Arbaugh, W.A. Submission to the IEEE-802.11. doc# IEEE 802.11-01/230. www.cs.umd.edu/~waa/attack/v3dcmnt.htm. May 2001.
2. "802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions." Bellardo, J. and Savage, S. Proceedings of the USENIX Security Symposium, Washington, D.C. www.cs.ucsd.edu/users/savage/papers/UsenixSec03.pdf.
3. "Intercepting Mobile Communications: The Insecurity of 802.11." Borisov, N., Goldberg, I., and Wagner, D. 7th Annual Conference of Mobile Computing and Networking. July 2001.
4. "Weaknesses in the Key Scheduling Algorithm of RC4." Fluhrer, S., Mantin, I., and Shamir, A. In Proc. 8th Workshop on Selected Areas in Cryptography. LNCS 2259. www.crypto.com/papers/others/rc4_ksaproc.ps. Springer-Verlag. 2001.
5. "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP, Revision 2." Stubblefield, A., Ioannidis, J., and Rubin, A.D. AT&T Labs. www.uninett.no/wlan/download/wep_attack.pdf. August 21, 2001.
6. "Unsafe at Any Key Size: An Analysis of the WEP Encapsulation." Walker, J. IEEE doc# 802.11-00/362. October 2000.

This page intentionally left blank

Index

Symbols & Numerics

A

B

C

D

E

F

G–H

I

J–K

L

M

N–O

P

Q–R

S

T

U

V

W