3. If IV1 = IV2, proceed to Step 4.
4. C1 XOR C2 = {Plaintext P1 XOR [Stream Cipher RC4 with key generated through the use of K, IV1]} XOR {Plaintext P2 XOR [Stream Cipher RC4 with key generated through the use of K, IV2]} = P1 XOR P2, the XOR of the two plaintexts.

With the Exclusive Or of the two plain-text items known corresponding to the transmitted cipher text items, dictionary attacks can be applied to determine the plain-text items.

WEP security upgrades

Because of the weaknesses in WEP security, IEEE 802.11 established Task Group i (TGi) to develop approaches to address WEP problems. TGi had to consider a number of issues and constraints. One path was to redesign 802.11 security so as not to include any legacy WEP functions. Another path was to upgrade WEP security while keeping the same WEP architecture. Both approaches were chosen, resulting in the completely new 802.11i standard and the upgraded WEP encryption and integrity method called the *Temporal Key Integrity Protocol* (TKIP). The latter approach was necessary to accommodate the huge base of existing wireless WEP devices already deployed and to have improved security in place because of the anticipated delay in developing and finalizing the 802.11i standard. The installed WEP implementations have hardware-based WEP functions that cannot be easily modified, so the TKIP solution was chosen because it can be installed as a software upgrade to the legacy systems. In addition, because of the limited additional computing capability remaining on extant access points, the TKIP upgrade could not be computing resource–intensive. TKIP uses the *802.1X authentication architecture* as a basis for secure key exchange, so the next section briefly describes 802.1X as a precursor to an overview of the TKIP algorithms.

802.1X authentication

802.1X is a port-based authentication mechanism that operates under the *Extensible Authentication Protocol (EAP)* transport protocol (RFC 2284). For wireless LANs, the EAP protocol is known as EAP over LAN (EAPOL). EAPOL is applied to the exchange of challenges and responses between client stations, or *supplicants*, as they are called in the protocol, and an authentication server. The third entity in 802.1X is the *authenticator*, a dual access control port, similar to the access point. The authentication server is usually a RADIUS server, but other authentication servers can be employed. In this discussion, a RADIUS server is used. EAPOL supports a number of protocols, including Transport Layer Security (TLS), RFC 2246. A typical authentication process employing EAPOL proceeds as follows:

1. The supplicant sends credentials to the RADIUS server.
2. The RADIUS server provides credentials to the supplicant.
3. Upon mutual authentication, the protocol is used to establish session keys.
4. The session keys are used to encrypt the client station message.

In more detail, the sequence occurs in the following steps:

1. A conventional 802.11 association is established.
2. At this point, all non-802.1X traffic is blocked.
3. The RADIUS server sends a challenge to the supplicant (client station).
4. The client hashes the user-provided password as a response to the RADIUS server. This hash is sent to the RADIUS authentication server through the authenticator.
5. The RADIUS server uses the same process to compute the hash based on its database of user passwords.
6. If a match of the hashes is obtained, the RADIUS server generates a dynamic WEP secret key and sends it to the authenticator.
7. The WEP secret key is sent to the client via EAPOL key frames.
8. The secret keys are updated at specified intervals.

Because employing 802.1X for WEP encryption does not eliminate weak IV and IV collision vulnerabilities, TKIP was developed to address these and other WEP security weaknesses.

Temporal Key Integrity Protocol

TKIP is built around the existing WEP security algorithm because of the necessity of not adding complex cryptographic algorithms whose execution would far exceed the spare CPU cycles available on most of today's deployed access points. Table 16-5 lists the upgrades provided by TKIP in terms of the security weaknesses addressed.

TABLE 16-5

TKIP Upgrades for WEP Weaknesses

Weakness	TKIP Upgrade
Correlation of IVs with weak keys	Per-packet key mixing function
Replay	IV sequencing discipline
Key reuse	Rekeying approach
Susceptibility to forgery	Message Integrity code (MIC) called Michael

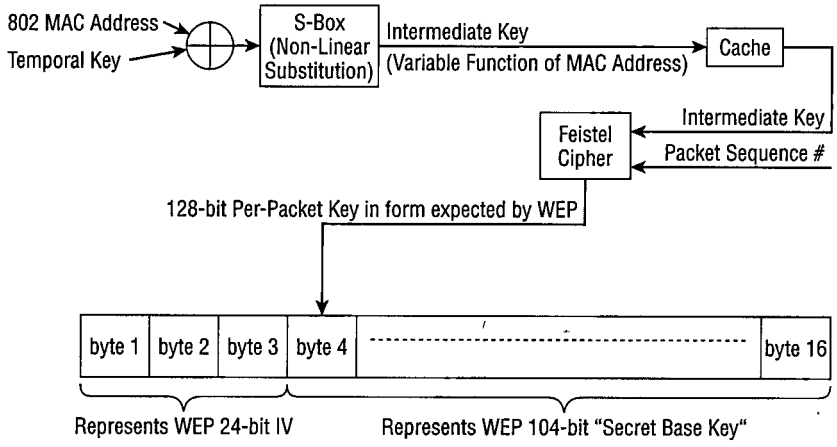
Per-packet mixing function

The TKIP *per-packet key mixing function* addresses the problem of correlating IVs with weak keys by using a key that varies with time, or temporal key, as the WEP secret base key. It then uses the packet sequence counter and temporal key to construct the per-packet key and IV. These

operations hide the relationship between the IV and the per-packet key and are illustrated in Figure 16-16.

FIGURE 16-16

TKIP per-packet mixing function



The process in Figure 16-16 shows that using the Exclusive Or function for the local MAC address with the temporal key results in different client stations and access points generating correspondingly different intermediate keys. Thus, the per-packet encryption keys are different at every client station. The result of the total process is a 16-byte packet that corresponds to the input that is expected by existing WEP hardware.

IV sequencing discipline

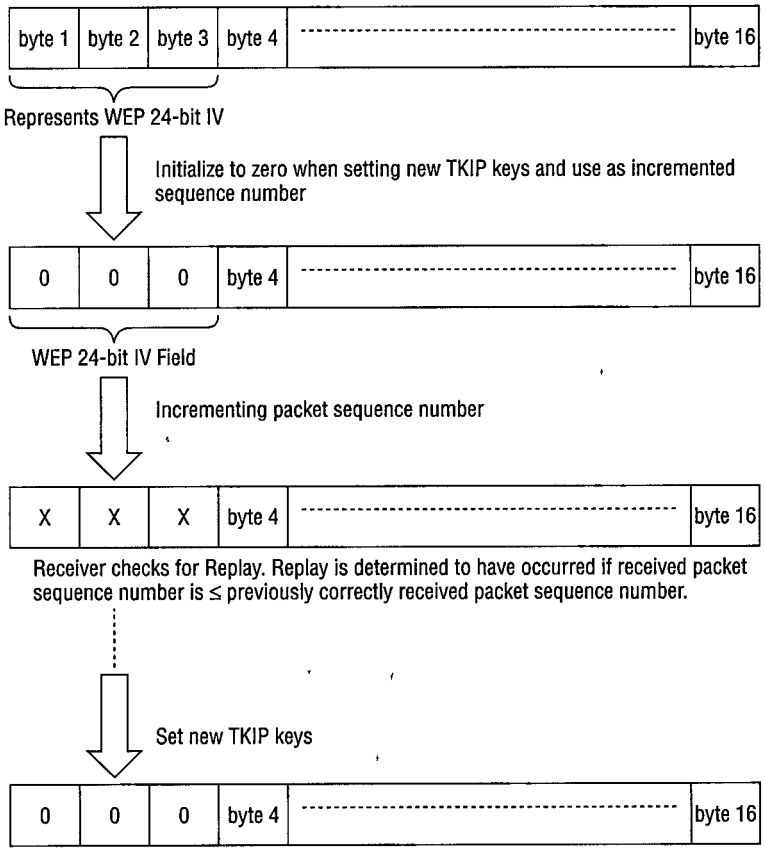
As a control against replay attacks, TKIP applies an IV sequencing discipline in which a receiver determines if a packet is out of sequence. If that condition is true, the receiver assumes it is a replay and discards the packet. A packet is defined as out of sequence if its IV is less than or equal to that of a previously correctly received packet. By using the WEP IV field as a packet sequence number, the procedure for detecting and countering replays is as follows:

1. New TKIP keys are used.
2. Receiver and transmitter initialize the packet sequence number to zero.
3. As each packet is transmitted, the packet sequence number is incremented by the transmitter.
4. The IV sequencing discipline is applied to determine if a packet is out of sequence and a replay has occurred.

This procedure is illustrated in Figure 16-17.

FIGURE 16-17

TKIP replay sequence checking



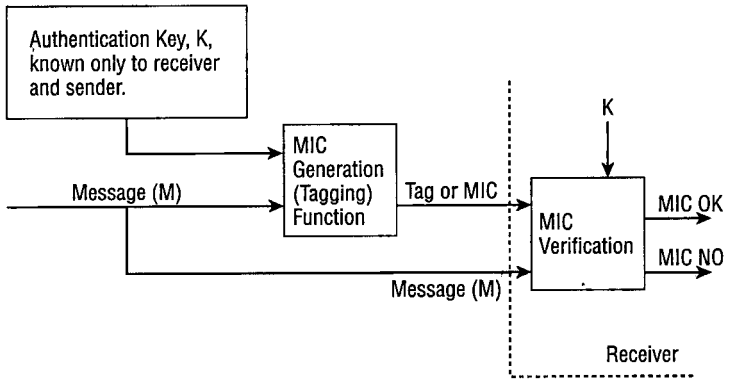
Message Integrity Codes against forgery

An ideal *Message Integrity Code (MIC)* is a unique, unambiguous representation of the transmitted message that will change if the message bits change. Thus, if an MIC is calculated using an authentication key by a transmitting entity and sent with the message, the receiver can similarly calculate another MIC based on the message and compare it to the MIC that accompanied the message. If the two MICs are identical, in theory, the message was not modified during transmission.

In TKIP, the 64-bit MIC is called *Michael* and was developed by Niels Ferguson, an independent cryptography consultant based in Amsterdam, Holland. The TKIP MIC process is illustrated in Figure 16-18.

FIGURE 16-18

TKIP MIC generation and verification



Rekeying against key reuse

To protect against key reuse, 802.1X uses a hierarchy of master keys, key encryption keys, and temporal keys. The 802.1X temporal keys are used in the TKIP authentication and confidentiality processes. A temporal key set comprises a 64-bit key for the MIC process, as described in the previous section, and a 128-bit encryption key. A different set of temporal keys is used in each direction when an association is established. The material used to generate the temporal keys must be protected from compromise and this protection is accomplished by use of key encryption keys. The master key is needed to set up the key encryption keys. This process is summarized as follows:

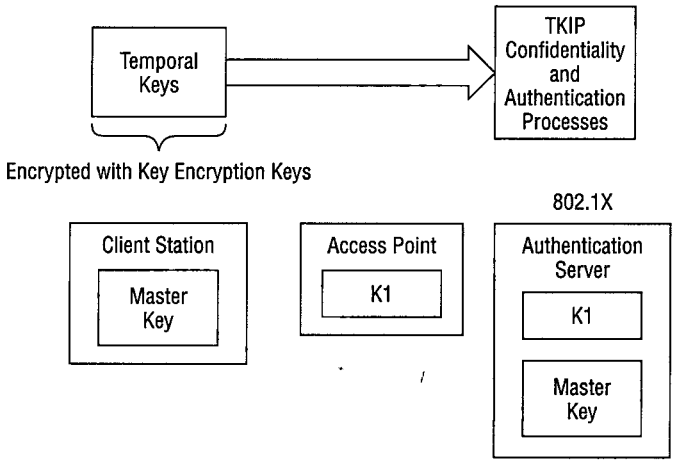
- 802.1X defines that the authentication server and client station share a secret key, the master key.
- 802.1X defines that the authentication server and access point share a secret key, derived by the authentication server and client station from the master key and distributed by the authentication server to the access point.
- A new master key is used with each session (a session covers the time from authentication to when the key expires, is revoked, or when a client station no longer communicates).
- The master key is used to protect the communication of key encryption keys between a client station and the access point.

- The key encryption keys are employed to protect the transmitted keying material used by the access point and client to generate sets of temporal keys.
- The pairs of temporal keys are used for integrity protection and confidentiality of the data.

Figure 16-19 shows the relationships and locations of the three types of keys.

FIGURE 16-19

Key hierarchy for rekeying



802.11i

The 802.11i wireless security standard was ratified in June of 2004. The IEEE 802.11 committee considers this specification a long-term solution to wireless security. It incorporates TKIP, 802.1X, and the Advanced Encryption Standard (AES). AES is a block cipher and, in 802.11i, processes plain text in 128-bit blocks. It uses the following set of keys:

- **A symmetric master key** — Possessed by the authentication server and client station for the positive access decision
- **A pairwise master key (PMK)** — A fresh symmetric key possessed by the access point and client station and used for authorization to access the 802.11 medium
- **A pairwise transient key (PTK)** — A collection of the following operational keys:
  - **Key encryption key (KEK)** — Used to distribute the group transient key (GTK), which is an operational temporal key used to protect multicast and broadcast data
  - **Key confirmation key (KCK)** — Binds the PMK to the client station and access point
  - **Temporal key (TK)** — Protects transmitted data

Thus, 802.11i employs a 128-bit key, combines encryption and authentication, uses temporal keys for both functions, and protects the entire 802.11i packet. In relation to the authentication server and EAP, RADIUS and EAP-TLS are not officially a part of 802.11i, but are de facto standards for use in 802.11i.

The next sections explore the AES and its employment in 802.11i because it is the major component of and provides the increased security capabilities in the new standard.

AES Counter and Cipher-Block Chaining modes

The two modes of operation of AES relative to 802.11i are Counter (CTR) and Cipher-Block Chaining (CBC).

In the CTR mode of operation, AES employs a monotonically increasing counter. The encryption process in the CTR mode is summarized as follows and is shown in Figure 16-20:

1. The Message, M, is broken into 128-bit blocks: M1, M2, ... Mn.
2. The key is determined.
3. The counter is initialized to zero.
4. For each block processed, increment the counter by one.
5. For each block, the counter value is encrypted.
6. The encrypted counter value is XORed with the plain-text block, Mi, to generate the cipher text block, Ci.
7. When all the plain-text blocks have been encrypted, the initial counter value is prepended to the cipher text blocks to generate the message ($counter_0$) C= ($counter_0$){C1, C2, ... Cn}.
8. The message is transmitted.
9. The receiver decrypts the message by reversing the process. It uses the prepended initial counter value as a starting point.

For security, the CTR mode requires a new, different key for every session.

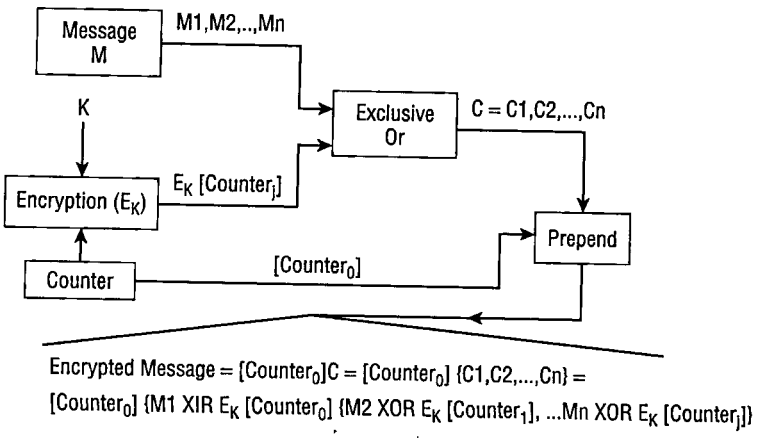
The AES CBC mode employs an initialization vector for enhanced security and operates in the following steps:

1. The Message, M, is broken into 128-bit blocks: M1, M2, ... Mn.
2. A random initial IV value is chosen.
3. This first IV value is XORed with plain-text block M1.
4. Encrypted block C1 is generated by encrypting the result of the XOR in the previous step with the encryption key, K. C1 also becomes the next IV to be used in the XOR function with M2.
5. This process iterates until all plain-text blocks are encrypted.

6. The message to be transmitted is assembled by prepending the initial IV to the cipher text C= C1, C2, ... Cn.
7. The receiver performs decryption by using the prepended initial IV value and reversing the process.

FIGURE 16-20

AES CTR mode



A different, initial IV must be used for each new message to maintain security.

The steps in the CBC mode are shown in Figure 16-21.

The AES CBC mode can also be employed to generate an MIC and ensure that a message has not been modified during transmission. The MIC is generated as follows:

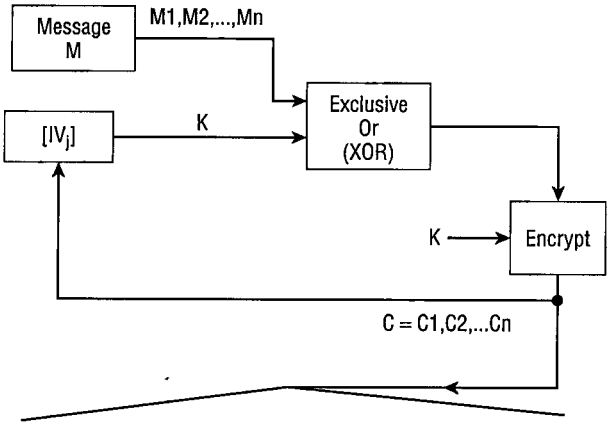
1. The Message, M, is broken into 128-bit blocks: M1, M2, ... Mn.
2. An initial IV value that is known to the transmitter and receiver is chosen.
3. This first IV value is XORed with plain-text block M1.
4. A Tag block, MIC1, is generated by encrypting the result of the XOR in the previous step with the encryption key, K. MIC1 also becomes the next IV to be used in the XOR function with M2.
5. This process iterates until the last Tag block, MICn, is generated.
6. The Tag block, MICn, is appended to the transmitted message as an integrity check.

The receiver generates an MICn using the same algorithm and initial IV as the transmitter and compares it to the MICn received with the message. If the values match, the message is assumed to have been transmitted without modification.

Figure 16-22 illustrates AES MIC generation.

FIGURE 16-21

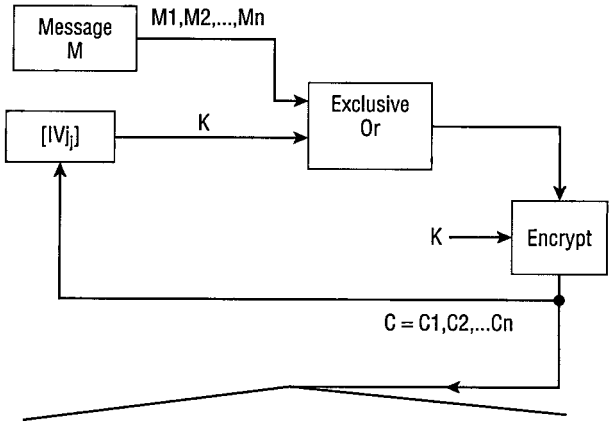
AES CBC mode



Encrypted Message = $[IV_0]C = [IV_0]\{C1,C2,...,Cn\} =$
$[IV_0]\{E_K[M1 \text{ XOR } IV_0], E_K[M2 \text{ XOR } [IV_1], ..., E_K[Mn \text{ XOR } [IV_j]]\}$

FIGURE 16-22

CBC mode for MIC generation



Encrypted Message = $[IV_0]C = [IV_0]\{C1,C2,...,Cn\} =$
$[IV_0]\{E_K[M1 \text{ XOR } IV_0], E_K[M2 \text{ XOR } [IV_1], ..., E_K[Mn \text{ XOR } [IV_j]]\}$

Application of AES in 802.11i

The AES is applied in 802.11i in the form of the AES — Counter with CBC-MAC (AES-CCM) protocol. AES-CCM applies the AES CTR mode for confidentiality of data and combination CBC-MAC mode for data integrity.

AES-CCM uses the same AES key for encryption and for generating an MIC. In addition, AES-CCM employs a 48-bit packet sequence counter. This counter is then applied in the CTR mode and in the generation of the CBC-MAC mode initialization vector. The following steps describe this process:

1. Concatenate the source MAC address, the packet sequence counter, a 16-bit per-packet block counter, and a 16-bit string to form the CTR mode counter and CBC MAC-IV. The 16-bit string differentiates the two concatenation results as being the CTR mode counter or the CBC-MAC IV.
2. Increment the packet sequence counter.
3. The CCM-MAC IV and secret AES key are used to compute an MIC over the message packet, including the source and destination addresses.
4. Truncate the MIC to 64 bits.
5. Encrypt the packet and append MIC, applying the CTR mode counter and secret AES key.
6. Insert the packet sequence counter number in between the 802.11 header field and the encrypted message data.
7. Transmit the packet.

On the receiving end, the packet sequence counter is obtained from the message packet and checked for replay. If the message is valid, the packet sequence counter is used to generate the CTR mode counter and the CBC-MAC IV. Then, the process steps used in the transmission process are reversed.

The AES-CCM mode protects against forgeries through the use of an MIC, protects against replays by checking the packet sequence counter, encrypts the source and destination addresses, and does not use an initialization vector or counter value with the same AES secret key.

Additional 802.11i capabilities

802.11i provides for pre-authentication for roaming and, also, a Pre-Shared Key (PSK) mode. In this mode, there is no authentication exchange and a single private key can be assigned to the entire network or on a per-client station pair. PSK is amenable for use in ad-hoc and home networks. The PSK mode uses the PKCS#5v2.0PBKDF2 key derivation function to produce a 256-bit PSK from an ASCII string password. RFC 2898, PKCS #5: Password-Based Cryptography Specification Version 2.0 describes this operation, which applies a pseudorandom function to derive keys. The PSK mode is vulnerable to password/passphrase guessing using dictionary attacks.

Tools for testing and security wireless

The following are some tools that can be used to test and validate the security of a wireless network:

- **Kismet** is an 802.11 layer2 wireless network detector, sniffer, and intrusion detection system. Kismet will work with any wireless card that supports raw monitoring (rfmon) mode, and can sniff 802.11b, 802.11a, and 802.11g traffic. It will work on most Linux and UNIX platforms.
- **bsd-airtools** is a package that provides a complete toolset for wireless 802.11b auditing. It contains a WEP cracking application, a netstumbler clone, and a few tools for Prism2 debug modes. Most of the utilities only fully work with a Prism2 chipset-based card.
- **Aircrack** is a 802.11 WEP key cracker. It implements the so-called Fluhrer-Mantin-Shamir (FMS) attack, along with some new attacks by a talented hacker named KoreK. When enough encrypted packets have been gathered, aircrack can almost instantly recover the WEP key. It runs under Linux and Windows.
- **AirSnort** is a wireless LAN (WLAN) tool that recovers encryption keys. AirSnort operates by passively monitoring transmissions and computing the encryption key when enough packets have been gathered. It uses the Prism2 chipset.
- **Hotspotter** passively monitors the network for probe request frames to identify the preferred networks of Windows XP clients, and will compare it to a supplied list of common hotspot network names. If the probed network name matches a common hotspot name, Hotspotter will act as an access point to allow the client to authenticate and associate with it.
- **Wellenreiter** is a wireless network discovery and auditing tool. Prism2, Lucent, and Cisco based cards are supported. It can discover networks (BSS/IBSS), and automatically detects ESSID broadcasting or non-broadcasting networks and their WEP capabilities and the manufacturer. DHCP and ARP traffic are decoded and displayed to give you further information about the networks. An ethereal/tcpdump-compatible dumpfile and an Application savefile will be automatically created. There are two versions for Linux, a GTK/Perl version and a newer C++ version with a QT front end for desktop and an Opie front end for Linux handhelds such as the Zaurus.
- **WepLab** is a tool designed to teach how WEP works, what different vulnerabilities it has, and how they can be used in practice to break a WEP-protected wireless network. WepLab is more of a Wep Security Analyzer, designed from an educational point of view. The author has tried to leave the source code as clear as possible, running away from optimizations that would obfuscate it. Weplab works under any flavor of Linux for i386 and PPC, MacOSX and Windows NT/2000/XP.
- **Prismtumbler** is a wireless LAN (WLAN) that scans for beacon frames from access points. Prismstumbler operates by constantly switching channels and monitors any frames received on the currently selected channel. Prismstumbler uses AirSnort.

- **WEPCrack** is a tool for breaking 802.11 WEP secret keys. WEPCrack was the first of the WEP encryption cracking utilities.
- **SNR tool** helps the network administrator collect signal/noise-rate statistics from Lucent Wireless AccessPoint devices via SNMP, store it in a MySQL database, and view summary graphs via CGI-module.
- **APTools** is a utility for Windows and UNIX that queries ARP Tables and Content-Addressable Memory (CAM) for MAC Address ranges associated with 802.11b access points. It will also utilize Cisco Discovery Protocol (CDP) if available. If a Cisco Aironet MAC address is identified, the security configuration of the access point is audited via HTML parsing.
- **The Rice Monarch Project** develops protocols for adaptive mobile and wireless networking. The project was formerly hosted at CMU.
- **KOrinoco** is is a KDE clone of the Lucent Orinoco client manager.
- **Wavemon** is a monitoring application for wireless network devices. It currently works under Linux with devices that are supported by the wireless extensions by Jean Tourrilhes (included in Kernel 2.4 and higher), e.g. the Lucent Orinoco cards.
- **GNOME Wireless Applet** is a wireless link quality monitor panel applet for GNOME. It reads the link quality out of `/proc/net/wireless` and reports quality by altering color, like a mood ring.
- **Gkrellm wireless plug-in** monitors the signal quality of your wireless networking card (if its driver supports the Linux wireless extension API or you use Freebsd's wi0 interface).
- **NetStumbler** displays wireless access points and SSIDs, channels, checking whether WEP encryption is enabled and signal strength. NetStumbler can connect with GPS technology to accurately log the precise location of access points.
- **Ministumbler** is a smaller version of NetStumbler designed to work on PocketPC 3.0 and PocketPC 2002 platforms. It provides support for ARM, MIPS, and SH3 CPU types.
- **Btscanner** allows you to extract as much information as possible from a Bluetooth device without the requirement to pair. It extracts HCI and SDP information, and maintains an open connection to monitor the RSSI and link quality.
- **Fake AP** is the polar opposite of hiding your network by disabling SSID broadcasts. Black Alchemy's FakeAP generates thousands of counterfeit 802.11b access points. As part of a honeypot or as an instrument of your site security plan, FakeAP confuses Wardrivers, NetStumblers, Script Kiddies, and other scanners.
- **Redfang v2.5** is an enhanced version of the original Redfang application that finds non-discoverable Bluetooth devices by brute-forcing the last six bytes of the device's Bluetooth address and doing a `read_remote_name()`.
- **SSID Sniff** is a tool to use when looking to discover access points and save captured traffic. It comes with a configured script and supports Cisco Aironet and random prism2 based cards.

- **WiFi Scanner** analyzes traffic and detects 802.11b stations and access points. It can listen alternatively on all 14 channels, write packet information in real time, and search access points and associated client stations. All network traffic may be saved in the libpcap format for post analysis.
- **wIDS** is a wireless IDS. It detects the jamming of management frames and could be used as a wireless honeypot. Data frames can also be decrypted on-the-fly and re-injected onto another device.
- **WIDZ** is a proof-of-concept IDS system for 802.11 wireless networks. It guards access points (APs) and monitors local frequencies for malicious activity. It detects scans, association floods, and bogus/Rogue APs. It can also be integrated with SNORT or RealSecure.

Bluetooth

Bluetooth is a peer-to-peer, short-range protocol named after Harald Bluetooth, the king of Denmark in the late 900s. It is used to connect cellular phones, laptops, handheld computers, digital cameras, printers, and so on. It is defined in IEEE standard, IEEE 802.15 and has the following characteristics:

- **FHSS** — Hops 1,600 times per second among 79 RF channels
- **Transmission rate** — 1 Mbps
- **Transmission distance** — About 30 feet
- **Frequency band** — 2.4 Ghz to 2.5 Ghz
- **Transmitting power** — 1 milliwatt, which minimizes interference with other networks (cell phones can transmit up to 3 watts of power)
- **Transmission range extension** — Range can be extended to 300 feet by increasing transmitting power to 100 milliwatts
- **Number of devices on the network** — 8

Because FHSS is used, other Bluetooth networks can exist in the same area without any mutual interference. Bluetooth devices operate by setting up a personal area network (PAN) called a *piconet* based on the devices' assigned addresses. A Bluetooth piconet operates in the following manner:

- As an ad hoc network.
- All Bluetooth devices are peer units.
- Different piconets have different frequency hopping sequences to prevent interference.
- All devices on the same piconet are synchronized to the frequency hopping sequence for that piconet.
- One device operates as a master and the other devices operate as slaves (point-to-multipoint topology).

- A maximum of seven active slaves can exist on a piconet, each assigned a 3-bit active member address.
- Up to 256 inactive (*parked*) slaves that are synchronized to the frequency-hopping sequence can be assigned to the piconet. They can activate rapidly because they are synchronized.

Bluetooth security uses challenge response protocols for authentication, a stream cipher for encryption, and dynamic session keys.

Wireless Application Protocol

The Wireless Application Protocol (WAP) is widely used by mobile devices to access the Internet. Because it is aimed at small displays and systems with limited bandwidth, it is not designed to display large volumes of data. In addition to cellular phones and PDAs, WAP is applied to network browsing through TV and in automotive displays. It has analogies to TCP/IP, IP, and HTML in wired Internet connections and is actually a set of protocols that covers Layer 7 to Layer 3 of the OSI model. Because of the memory and processor limitations on mobile devices, WAP requires less overhead than TCP/IP.

WAP has evolved through a number of versions, the latest being version 2.0. WAP 2.0 includes support for the transmission and reception of sound and moving pictures over telephones and other devices, as well as providing a toolkit for development and deployment of new services, such as Extensible Hypertext Markup Language (XHTML).

The WAP architecture comprises the following levels:

- **Application layer** — Contains the wireless application environment (WAE) and is the direct interface to the user. The Application layer includes the following:
  - The Wireless Markup Language (WML)
  - A microbrowser specification for Internet access
  - WMLScript (development language)
- The Handheld Device Markup Language (HDML) is a simpler alternative to and actually preceded WML. HDML contains minimal security features, however. Another alternative is Compact HTML (C-HTML). Used primarily in Japan through NTT DoCoMo's i-mode service, C-HTML is essentially a stripped-down version of HTML. Because of this approach, C-HTML can be displayed on a standard Internet browser.
- **Session layer** — Contains the Wireless Session Protocol (WSP), which facilitates the transfer of content between WAP clients and WAP. This layer provides an interface to the WAE through the following activities:
  - Connection creation and release between the client and server
  - Data exchange between the client and server
  - Session suspend and release between the client and server

- **Transaction layer** — Provides functionality similar to TCP/IP through the Wireless Transactional Protocol (WTP). WTP provides transaction services to WAP, including acknowledgment of transmissions, retransmissions, and removal of duplicate transactions.
- **Security layer** — Contains Wireless Transport Layer Security (WTLS). WTLS is based on Transport Layer Security (TLS) and can be invoked similar to HTTPS in conventional Web browsers. WTLS supports privacy, data integrity, DoS protection services, and authentication. WTLS provides the following three types of authentication:
  - **Class 1 (anonymous authentication)** — The client logs on to the server, but in this mode, neither the client nor the server can be certain of the identity of the other.
  - **Class 2 (server authentication)** — The server is authenticated to the client, but the client is not authenticated to the server.
  - **Class 3 (two-way client and server authentication)** — The server is authenticated to the client and the client is authenticated to the server.

  Authentication and authorization can be performed on the mobile device using smart cards to execute PKI-enabled transactions. A specific security issue that is associated with WAP is the WAP GAP. A WAP GAP results from the requirement to change security protocols at the carrier's WAP gateway from the wireless WTLS to Secure Sockets Layer (SSL) for use over the wired network. At the WAP gateway, the transmission, which is protected by WTLS, is decrypted and then re-encrypted for transmission using SSL. Thus, the data is temporarily in the clear on the gateway and can be compromised if the gateway is not adequately protected. To address this issue, the WAP Forum has put forth specifications that will reduce this vulnerability and support e-commerce applications. These specifications include WMLScript Crypto Library and the WAP Identity Module (WIM). The WMLScript Crypto Library supports end-to-end security by providing for cryptographic functions to be initiated on the WAP client from the Internet content server. These functions include digital signatures originating with the WAP client and the encryption and decryption of data. The WIM is a tamper-resistant device, such as a smart card, that cooperates with WTLS and provides cryptographic operations during the handshake phase. A third alternative is to use a client proxy server that communicates authentication and authorization information to the wireless network server.
- **Transport layer** — Supports the Wireless Datagram Protocol (WDP), which provides an interface to the wireless networks. It supports network protocols such as GSM, CDMA, and TDMA. It also performs error correction.

The Public Key Infrastructure (PKI) for mobile applications provides for the encryption of communications and mutual authentication of the user and application provider. One concern associated with the mobile PKI relates to the possible time lapse between the expiration of a public key certificate and the reissuing of a new valid certificate and associated public key. This "dead time" may be critical in disasters or in time-sensitive situations. One solution to this problem is to generate one-time keys for use in each transaction.

Future of Wireless

Over the past 10 years or so, an alternative to wired LAN structures has evolved in the form of the wireless LAN. The first-generation wireless LAN products operated in the unlicensed 900–928 MHz Industrial Scientific and Medical (ISM) band, with low range and throughput offering (500 Kbps). They were subject to interference and came to market with little success in some applications. But they enjoyed a reputation of being inexpensive due to breakthroughs in semiconductor technologies. On the other hand, the band became crowded with other products in a short time, leaving no room for further development. The second generation in 2.40–2.483 GHz ISM band WLAN products boosted by the development of semiconductor technology was developed by a huge number of manufacturers. Using spread spectrum technology and modern modulation schemes, this generation's products were able to provide data rates up to 2 Mbps, but again the band became crowded since the most widely used product in 2.4 GHz is the microwave oven, which caused interference. Third-generation products assembled with more complex modulation in the 2.4 GHz band allow an 11 Mbps data rate. In June 1997, the IEEE finalized the initial standard for wireless LANs: IEEE 802.11. The first fourth-generation standard, HiperLAN, came as a specification from the European Telecommunication Standard Institute (ETSI) Broadband Radio Access Network (BRAN) in 1996, operating in the 5 GHz band. Unlike the lower frequency bands used in prior generations of WLAN products, the 5 GHz bands do not have large potential interferers such as microwave ovens or industrial heating systems as was true in 900 MHz and 2.4 GHz. In late 1999, the IEEE published two supplements to the 802.11: 802.11b and 802.11a, following the predecessors' success and interest from the industry. ETSI's next-generation HiperLAN family, HiperLAN/2, was proposed in 1999, operating on the same band with its predecessor, with the goal of providing high-speed (raw bit rate 54 Mbps) communications access to different broadband core networks and moving terminals.

Broadband wireless – WiMax

Broadband 802.16 wireless technology (WiMax) can help service providers meet these challenges because it has the ability to seamlessly inter-operate across various network types. It also provides the flexibility to support very high bandwidth solutions where large spectrum deployments (i.e., > 10 MHz) are desired. As a result, 802.16 can leverage existing infrastructure, keeping costs down, while delivering the bandwidth needed to support a full range of high-value, multimedia services. 802.16 technology can provide wide area coverage and quality of service capabilities for applications ranging from real-time delay sensitive Voice-over-IP (VoIP) to real-time streaming video — all to ensure that subscribers get the performance they expect for all types of communications. Industry standards will help contribute to economies of scale for 802.16 solutions, so that high performance can be provided at reasonable cost.

WiMax and 3G cellular technologies

WiMAX could be a serious threat to 3G because of its broadband capabilities, distance capabilities, and ability to support voice effectively with full QoS. This makes it an alternative to

cellular in a way that Wi-Fi can never be, so that while operators are integrating Wi-Fi into their offerings with some alacrity (looking to control both the licensed spectrum and the unlicensed hotspots), they will have more problems accommodating WiMAX. But as with Wi-Fi, it will be better for them to bring down their own networks than let independents do it for them, especially as economics and performance demand force them to incorporate IP into their systems. Handset makers such as Nokia, Erickson, and Samsung will be banking on this as they develop smart phones that support WiMAX as well as 3G. WiMAX can slash the single biggest cost of deployment: access charges for linking a hotspot to a local phone or cable network. A high frequency version of 802.16 would allow entrepreneurs to blast a narrow, data-rich beam between antennas miles apart. A standards-based long distance technology will avoid many of the problems of high upfront costs, lack of roaming, and unreliability — problems that those pioneers encountered — but it will still need to gain market share rapidly before 3G takes an unassailable hold. Given the current slow progress of 3G, especially in Europe, and the unusually streamlined process of commercializing WiMAX, the carriers are indulging in wishful thinking when they say nothing can catch up with cellular.

Beyond the future: IEEE 802.20

Meanwhile, another, separate IEEE standard in development seems to have significant overlap with WiMAX and IEEE 802.16e: the IEEE 802.20 standard. WiMAX and 802.16e are targeted for mobile users moving at speeds of up to 60 mph inside a WiMAX region (laptop users moving across a corporate campus, for example). But 802.20 is focused more on high-speed mobile users traveling across an extended metropolitan area at speeds of up to 150 mph. WiMAX/802.16 also differs from 802.20 in that it supports substantially higher data rates (up to 70 Mbps) than 802.20 (up to 1 Mbps). Both WiMAX/802.16e and 802.20 provide for mobility while enabling broadband connections across a much larger area than Wi-Fi and at higher data rates than what is commonly available to mobile clients today. Barring unexpected problems with the technology, it's likely we'll see both 802.16 and 802.20 products and services entering the market over the next few years, and we'll have to wait to see which standard gains traction for various user groups and applications.

Future architecture and building of wireless networks would depend on a variety of factors such as quality of service, transmission efficiency and range, bandwidth allowed, and mobility of the devices involved. With the increase in speed and range of wireless devices and communication, networks that were constrained basically to LANs have been able to grow and achieve MAN (Metropolitan Area Network) standards. Hotspots and other public area networks have shown proliferation over the past couple of years to substantiate that wireless networks will make the eventual difference. The emergence of WiMax/Broadband wireless devices as a standard has made this transition plausible. However commercial implications of wireless devices have been on the back burner because of issues such as security, transition cost, and management policy. Yet in the future, wireless technologies have the ultimate potential to coexist with conventional wireline networks to achieve higher advancements in the field of communication and networking.

Summary

Nearly every industry has benefited from wireless technology. Hospitals and medical professionals can get instant updates on patients without being physically present at the hospital. Travelers can get confirmation of flight schedules on the run. Many commercial vendors have set up wireless network access available to their customers that may heighten customer interest. Many other applications could be conceived easily and implemented without great difficulty. However, wireless technology also has some of its own drawbacks: wireless channels may not be as fast (they have less bandwidth) compared to conventional wired channels. Also, the range of wireless access may not be very high. Security may be highly affected as wireless networks become more popular because administrators cannot direct the flow of wireless information easily, and coding and channel access schemes are different compared to wired channels. This will necessitate equipment manufacturers' adding the functionalities.

This chapter also reviewed the electromagnetic spectrum and focused on the UHF band for cellular phone communications. The major components of the cellular phone network were described, including the mobile station, cell tower, subscriber identity module, base transceiver station, and mobile switching center. The chapter explained TDMA, FDMA, and CDMA technologies along with a subset of CDMA, spread spectrum technology. In particular, DSS, FSS, and OFDM spread spectrum implementations were discussed. The chapter reviewed different generations of cellular systems development, including AMPS, TACS, NMT, GSM, UMTS, and IMT-2000. The chapter also explained and summarized the 802.11 wireless LAN standard, including its various upgrades and instantiations, such as 802.11, 802.11a, 802.11b, 802.11g, and 802.11i. The related 802.11 wireless security issues were explored and the various solutions to the original 802.11 WEP security deficiencies were developed. You also learned a little about Bluetooth piconets and the WAP protocols.