FORTRESS
TECHNOLOGIES™

Wireless Link Layer Security (*w*LLS) Architecture

Version 1.0

A White Paper By
Bill McIntosh

Fortress Technologies
Corporate Headquarters
4025 Tampa Road, Suite 1111
Oldsmar, Fl 34677
813-288-7388

Abstract

Wireless Link Layer Security (*w*LLS) is a layer 2 security protocol that delivers point-to-point security for wireless network communications. It was designed to provide a simple, robust level of strong security in wireless devices, solving the security inadequacies uncovered in the current IEEE 802.11b wireless standard. The IEEE 802.11b standard includes an encryption mechanism, defined as the Wired Equivalent Privacy (WEP), to provide the same level of security found in wire-line networks. However, it is known that WEP provides limited security, subjecting the user to the same vulnerabilities present in wire-line networks. The intent of wLLS is to take security to the next level for wireless networks by providing an advanced level of security similar to the VPN technology available on wire-line networks.

*w*LLS was engineered with the intent to seamlessly integrate with 802.11b and provide an application independent security solution for sensitive wireless communications. *w*LLS provides secure frame and packet transmissions for wireless data networks through the automation of all critical security operations including encryption, authentication, data integrity checking, key exchange and data compression. This protocol has been designed using techniques from Fortress' Secure Packet Shield technology, which has been proven and validated by independent government agencies and is deployed in various government, military and commercial sectors. *w*LLS, modular and state driven, has a small footprint and is computationally light, making it ideal for embedded solutions or easy integration into any standard hardware platform over various operating systems.

Table of Contents

1 Governing Principles of *Wireless* Link Layer Security

- Highly secure in all wireless and mobile environments.

- Small footprint for easy integration into NIC cards, access point devices and mobile devices.

- Operates with IEEE 802.11, IEEE 802.11a, IEEE 802.11b and WEP.

- Allows higher-level protocols, like IPX, to pass securely, therefore making it independent of any information above the data link layer.

- Computationally light and runs on standard hardware so it can operate in embedded devices that do not have a high level of computational power.

- Easily integrates into hardware platforms and operating systems.

2 How does it Work?

2.1 IEEE 802.11 Basic Summary

IEEE 802.11 WLAN is designed to support a network where most decision-making is distributed to the mobile stations. This architecture has several advantages, including being very tolerant of faults in all of the WLAN equipment and eliminating any possible bottlenecks a centralized architecture would introduce. The architecture is very flexible, easily supporting both, small transient networks and large semi-permanent or permanent networks. In addition, deep power-savings modes of operation are built into the architecture and protocols to prolong the battery life of mobile equipment without losing network connectivity. The IEEE 802.11 architecture comprises several components: the Station, the Access Point (AP), the wireless medium, the Basic Service Set (BSS), the Distribution System (DS, the network that ties the APs together), and the Extended Service Set. The architecture also includes station services and distribution services. It's recommended that you read (1) in order to understand the basics of IEEE 802.11. (O'Hara and Petrick 7)

The architecture also embeds a level of indirection that has not been present in previous LANs. It is this level of indirection, handled entirely with the IEEE 802.11 architecture and transparent to protocol users of the IEEE 802.11 WLAN that provides the ability of a mobile station to roam throughout a WLAN and appear to be stationary to the protocols above the MAC that have no concept of mobility. This "sleight of hand" is performed by IEEE 802.11 allows all the existing network protocols to run over a WLAN without any special considerations. (O'Hara and Petrick 8)

2.2 IEEE 802.11 WEP

IEEE 802.11 incorporates MAC-level privacy mechanisms to protect the content of data frames from eavesdropping. This is because the medium for the IEEE 802.11 WLAN is significantly different from that of a wired LAN. The IEEE 802.11 Wired Equivalent Privacy (WEP) mechanism provides protection at a level that is felt to be equivalent to that of a wired LAN. (O'Hara and Petrick 74)

WEP is an encryption mechanism that takes the content of a data frame, its frame body, and passes it through an encryption algorithm. The result then replaces the frame body of the data frame and is transmitted. Data frames that are encrypted are sent with the WEP bit in the frame control field of the MAC header set. The receiver of an encrypted data frame passes the encrypted frame body through the same encryption algorithm used by the sending station. The result is the original, unencrypted frame body. The receiver then passes the unencrypted results up to higher layer protocols. (O'Hara and Petrick 74)

WEP is an RC-4-based, 40 bit encryption mechanism that encrypts the data portion of the frame. WEP relies on a secret key that is shared between a mobile station and an access point. The secret key is used to encrypt frames before they are transmitted and an integrity check is used to ensure that frames have not been modified in transit. The standard does not discuss how the shared key is established. In practice, most installations use a single key that is shared between all mobile stations and access points.

A stream cipher, the RC-4 encryption algorithm operates by expanding a short key into an infinite pseudo-random key stream. The sender XORs the key stream with the plaintext to produce ciphertext. The receiver has a copy of the same key and uses it to generate identical key streams. XORing the key stream with the cipher text yields the original plaintext.

Currently 128-bit RC-4 encryption is available. However, because of the stream cipher it's only marginally better than the 40-bit RC-4 encryption.

Unfortunately, high-end equipment can break 40-bit encryption in a matter of seconds. In addition, WEP has another major loophole. To achieve mobility within a campus, all access points are set to use the same key. This key is stored on all users' computers enabling an intruder to easily obtain the common key and gain access to the entire network.

2.3 Where does *wireless* Link Level Security (*w*LLS) Fit

Based on Fortress' Secure Packet Shield (SPS) technology, *Wireless* Link Layer Security (*w*LLS) is a true virtual private networking security protocol that operates on the data link layer to provide point-to-point protection of wireless communications. Thus it is independent to upper layer protocols. *w*LLS protects data in an IEEE 802.11 frame with a strong encryption and authentication process. The *w*LLS protocol can be used with any existing IEEE 802.11 wireless system. Since the entire *w*LLS software is small, (under 200k bytes), it can easily be embedded in any device.

*w*LLS is designed using standard encryption methods and a modified Diffie-Hellman key exchange to automatically build its security associations. It uses standard encryption algorithms such as DES, 3DES or IDEA and does data integrity checking and authentication.

3 Technical Overview of *w*LLS

3.1 Why is *w*LLS so Efficient?

*w*LLS has been designed as a self-contained, single efficient software program to save processing cycles and memory. *w*LLS only needs a small amount of memory, approximately 200k bytes, for it to operate.

The main efficiency advantage in *w*LLS is its key exchange mechanism. Unlike other security protocols, *w*LLS requires only 2 steps and 4 frames to complete the key exchange. This makes it up to 20 times faster than other key exchange methods.

3.2 Encryption

The strength of encryption rests in:

- The cryptographic strength of the algorithm (how resistant it is against crypto-analytical methods of cracking); and

- The length of the encryption key.

The longer the key, the tougher it is to crack it with brute force methods. *w*LLS uses the 128-bit International Data Encryption Algorithm (IDEA) as a default. Other algorithms like DES, 3DES and Skipjack are also available. If necessary, other algorithms can be customized into the product.

3.3 Wireless Key Exchange and Packet Flow

Working at the data link layer, *w*LLS is used for point-to-point protection of communications between two wireless/mobile devices, such as from a Mobile Station (MS) to an Access Point (AP), as shown in figure 3.2-1.

On power-up, each station has a Hard Key, a Public Static Key and a Private Static Key. These keys are generated from a Signature, a private code that is assigned to each *w*LLS entity, and a unique device parameter.

In figure 3.2-1, if an MS sends a frame to an AP it will embed its Public Static Key 1 in the frame, encrypted with its Hard Key, and send it to the AP. The AP decrypts the Public Static Key 1 with its Hard Key, generates a reply with its Public Static Key 2 and sends it back to the MS.

The MS then generates the Common Static Key created from its Private Static Key 1 and the AP Public Static Key 2. The MS uses this Common Static Key to encrypt its Public Dynamic Key 1 sent to the AP. The AP decrypts the Public Dynamic Key 1, generates a reply to the MS containing the Public Dynamic Key 2 and sends it to complete the key exchange process. Communications from this point forward will be encrypted.

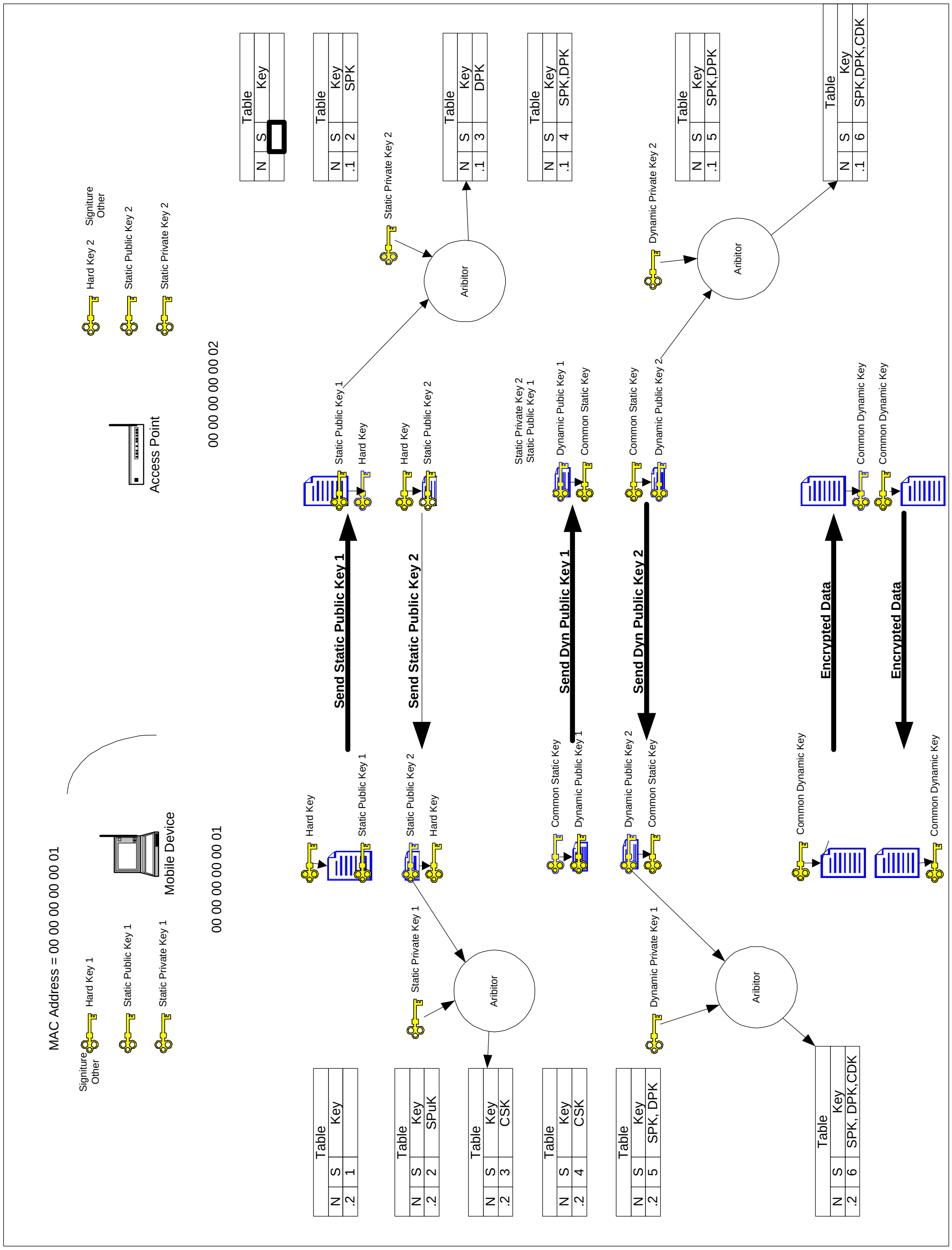


Table		
N	S	Key
.2	1	

Table		
N	S	Key
.2	2	SPuK

Table		
N	S	Key
.2	3	CSK

Table		
N	S	Key
.2	4	CSK

Table		
N	S	Key
.2	5	SPK, DPK

Table		
N	S	Key
.2	6	SPK, DPK,CDK

Table		
N	S	Key

Table		
N	S	Key
.1	2	SPK

Table		
N	S	Key
.1	3	DPK

Table		
N	S	Key
.1	4	SPK,DPK

Table		
N	S	Key
.1	5	SPK,DPK

Table		
N	S	Key
.1	6	SPK,DPK,CDK

Figure: 3.2-1: Frame Flow

3.4 Wireless LLS State Machine

Figure 3.3-1 is the high level *w*LLS State machine. It's not the intent of this white paper to go over the details of the state machine. This is to show that the *w*LLS is a true Virtual Private Networking security protocol.

3.4.1 Output Frame Processing

After the frame is received for processing its state is looked up in the Dynamic Database. The State will determine what Key will be used to encrypt or decrypt the frame as shown in figure 3.3-1a.

3.4.2 Input Frame Processing

If a clear text frame is received its segmentation policy needs to be checked. If its segmentation policy is standard, it will be dropped. If its segmentation policy is segmented 1 then it will be passed.

If an encrypted frame is received, it an attempt will be made to decrypt it using the common dynamic key. If the encryption fails, the frame will be dropped.

3.4.3 Key Processing

Frames are processed as shown in figure 3.3-1b. If the destination MAC exists in the dynamic database, the state dispatcher is invoked. If not, the dynamic database is set with the new MAC address and the process will go to State 1. In Summary:

- State 1: Discover and Exchange Public Static Key
- State 2: Compute Common Static Key
- State 3: Complete the Common Static Key
- State 4: Exchange Public Dynamic Key
- State 5: Compute Common Dynamic Key
- State 6: Finish Computing Common Dynamic Key
- State 7: Error: Bad Static Key to ID Pair
- State 8: Frame is not from a wLLS Peer

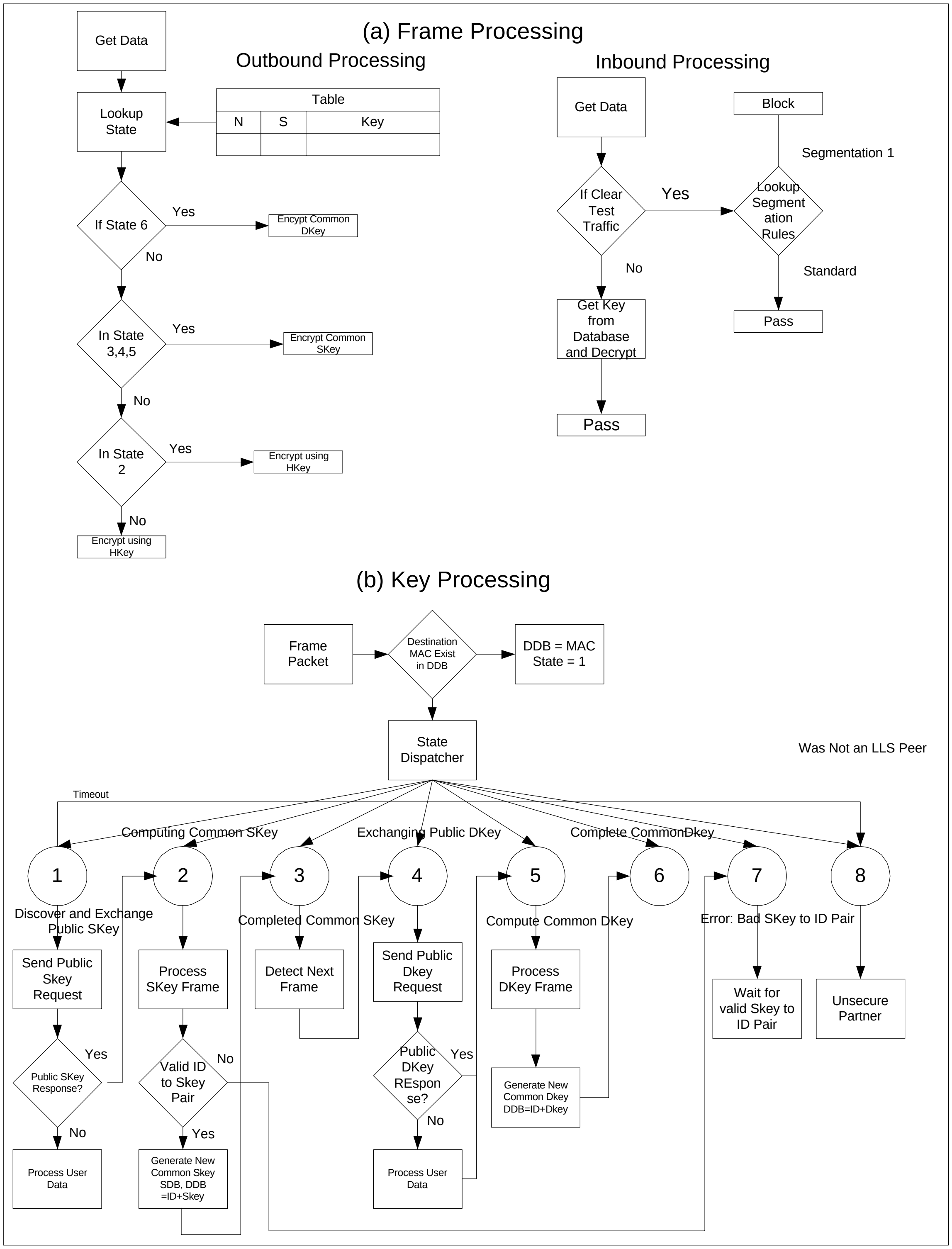


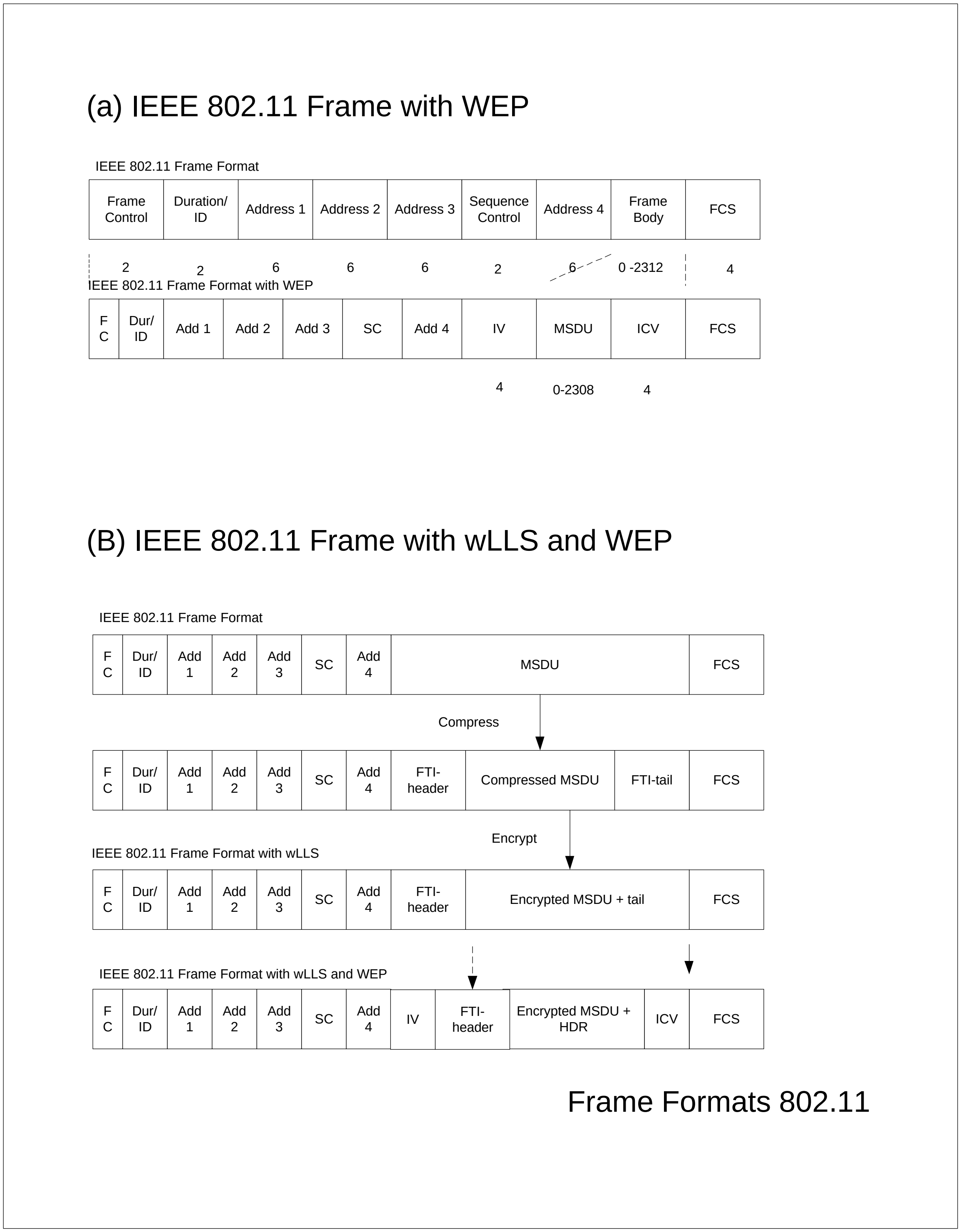
Figure 3.3-1: State Machine

3.5 Secure Frame Handling

3.5.1 IEEE 802.11 with *w*LLS Frame Format

Both WEP and *w*LLS add fields to the frame as shown in figure 3.4-1. In figure 3.4-1a the packet’s WEP bit in the header would be set and the Initialization Vector (IV) and Integrity Check Value (ICV) would be added to the packet. The data portion of the packet goes through the RC-4 based, 40 bit encryption mechanism.

wLLS can be combined with WEP but is strong enough to be utilized by itself. In figure 3.4-1b both wLLS and WEP are utilized. The FTI-header and FTI-tail is added to the packet and the data is compressed. From here the compressed data and the FTI-tail are encrypted using a cryptographic algorithm such as IDEA or 3DES. The frame is forwarded to the wireless driver where the WEP bit is set in the header and the IV and ICV is added to the packet. The data portion of the packet goes through the RC-4 based, 40 bit encryption mechanism.



(a) IEEE 802.11 Frame with WEP

IEEE 802.11 Frame Format

Frame Control	Duration/ ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	Frame Body	FCS
2	2	6	6	6	2	6	0 -2312	4

IEEE 802.11 Frame Format with WEP

F C	Dur/ ID	Add 1	Add 2	Add 3	SC	Add 4	IV	MSDU	ICV	FCS
							4	0-2308	4	

(B) IEEE 802.11 Frame with wLLS and WEP

IEEE 802.11 Frame Format

F C	Dur/ ID	Add 1	Add 2	Add 3	SC	Add 4	MSDU	FCS

F C	Dur/ ID	Add 1	Add 2	Add 3	SC	Add 4	FTI-header	Compressed MSDU	FTI-tail	FCS

IEEE 802.11 Frame Format with wLLS

F C	Dur/ ID	Add 1	Add 2	Add 3	SC	Add 4	FTI-header	Encrypted MSDU + tail	FCS

IEEE 802.11 Frame Format with wLLS and WEP

F C	Dur/ ID	Add 1	Add 2	Add 3	SC	Add 4	IV	FTI-header	Encrypted MSDU + HDR	ICV	FCS

Frame Formats 802.11

Figure 3.4-1: Frame Formats

3.5.2 Integrity Checking

The hash in the FTI-header on the sender's side is calculated after encryption and checked on the receiver's side before decryption, thus providing an integrity test of the data in transit.

The hash in the FTI-tail on the sender's side is calculated before encryption and checked on the receiver's side after decryption to provide an integrity test of the encryption system. The hashes differ if the common secret key is not identical on both sides.

3.5.3 Authentication

The second encrypted hash also serves as a means of strong authentication since the static and dynamic common crypto keys used to encrypt the hash are secret and shared by the sender and the receiver only.

This encrypted hash in the dynamic key-frame verifies that the dynamic public key is truly coming from the sender, thus authenticating the sender. The same encrypted hash in a regular frame authenticates that the sender of the frame is who it claims to be.

3.6 The Mobile Databases

Each mobile device will have a Static Database (SD) and a Dynamic Database (DD). The Static Database is used to keep the static keys along with other crypto information. The dynamic database is used to keep the MAC Address, its state and the current active key. The active key will change based on the state. None of the dynamic keys or common crypto keys are store in persistent memory.

3.7 Local Roaming

Because of the dynamic nature of *w*LLS when a mobile station moves from one access point to another, as shown in figure 3.6-1, a new key exchange will occur automatically. The mobile user will not know this has occurred. An IEEE 802.11 re-association will occur with the new AP. A new key exchange will occur between the mobile station and the AP. Any queued packets will automatically be transferred from the old AP to the new AP through the DS. The association and the *w*LLS information will be invalidated on the old AP.

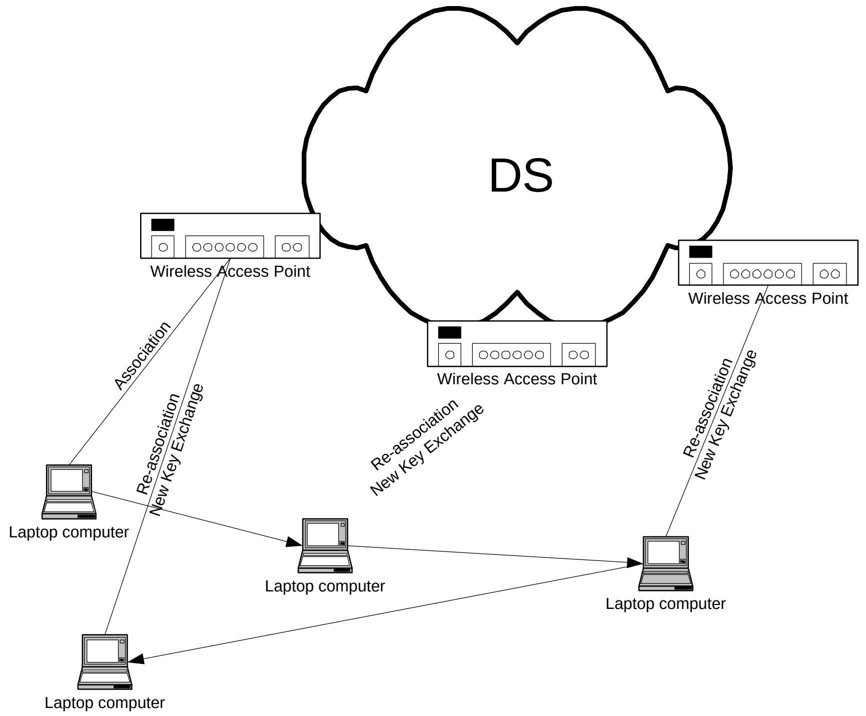


Figure 2.7-1: Roaming with the Wireless LAN

4 Wireless LAN with *w*LLS Applications

4.1 Wireless LAN without Privacy

Wireless LANs are the most vulnerable of all the IEEE 802 LAN types. Without WEP or *w*LLS turned on, any untrustworthy person such as a competitor could stand outside of your building with a common laptop computer and read frames coming and going to your workstation as shown in figure 4.1-1. These frames could contain customer contacts, price sheets, specifications, memos or anything that may give him a competitive advantage.

4.2 Wireless LAN with WEP

WEP will give you a level of privacy as shown in figure 4.2-1. It would be unlikely that a competitor could steal your data. However, this would be easy for a junior hacker though. The junior hacker could perform the following attacks using a standard laptop computer with any packet analyzer that can be downloaded from the Internet:

- Passive Attacks to Decrypt Traffic: Intercept all wireless traffic, until an IV collision occurs. By XORing two frames that use the same IV, the attacker obtains the XOR of the two-plaintext messages. The resulting XOR can be used to infer data about the contents of the two messages. IP traffic is often very predictable and includes a lot of redundancy. This redundancy can be used to eliminate many possibilities for the contents of messages. Further educated guesses about the contents of one or both of the messages can be used to statistically reduce the space of possible messages, and in some cases it is possible to determine the exact contents.

- Active Attacks to Inject Traffic: If the hacker knows the exact plaintext for one encrypted message, he can use this knowledge to construct correct encrypted frames. The procedure involves constructing a new message, calculating the CRC-32, and performing bit flips on the original encrypted message to change the plaintext to the new message.

- Active Attacks form both Ends: The previous attack can be extended further to decrypt arbitrary traffic. In this case, the attacker makes a guess not about the contents, but rather the headers of a frame. The information is usually quite easy to obtain or guess. All that is necessary to guess is the destination IP address. Armed with the IP address, the attacker can flip appropriate bits to transform the destination IP address to send the packet to a machine he controls, somewhere in the Internet, and transmit it using a rogue mobile station.

- Table Based Attacks: The small space of possible initialization vector allows a hacker to build a decryption table. With the plaintext for a frame, he can compute the RC4 key stream generated by the IV it used. This key stream can be used to decrypt all other frames that use the same IV. Over time the hacker can build a table of IVs and corresponding key streams.

4.3 Wireless LANs with *w*LLS

The only way to get security in the wireless LAN is to use a strong security solution such as *w*LLS. The following example illustrates how *w*LLS is not subjected to the same weaknesses as WEP:

- Passive Attacks to Decrypt Traffic: Even after the hacker has intercepted all the wireless traffic he would have a difficult time tying to break the encryption. Assuming *w*LLS is using the 128 bit IDEA encryption algorithm, the hacker would need a significant amount of processing power and years to break the algorithm. This is because IDEA is a symmetric algorithm; it operates on 64-bit blocks of plaintext, encrypting 64-bit blocks of plain text into 64-bit blocks of cipher text. Each 64-bit super bloc is divided into four 16-bit blocks. IDEA operates in 17 rounds, where the odd and even rounds are different. The 128-bit key is expanded into 52 16-bit keys, from which four are used in the odd rounds and two are used in the even rounds. IDEA uses three reversible operations, a bitwise exclusive, a slightly modified add, and a slightly modified multiply. The reversibility of operation is important to run IDEA backwards, e.g., to decrypt. As you can see IDEA is infinitely more advanced and secure than the RC-4 algorithm.

- Active Attacks to Inject Traffic: Even if the hacker knows the exact plaintext for one encrypted message it would be nearly impossible for him to construct correct encrypted frames. Since *w*LLS frames are both compressed and encrypted it would be impossible to do any form of bit flipping and calculation of the CRC-32.

- Active Attacks form both Ends: The header information is compressed and encrypted inside of the frame, therefore, impossible to guess. Any bit flipping would be useless in this frame to try to change the IP address of the frame.

- Table Based Attacks: The 128 bit IDEA encryption algorithm has a very long key that makes it difficult for a hacker to crack it with a brute force method. The dynamic key is also changed every 2 hours. The user can adjust this value for added security.

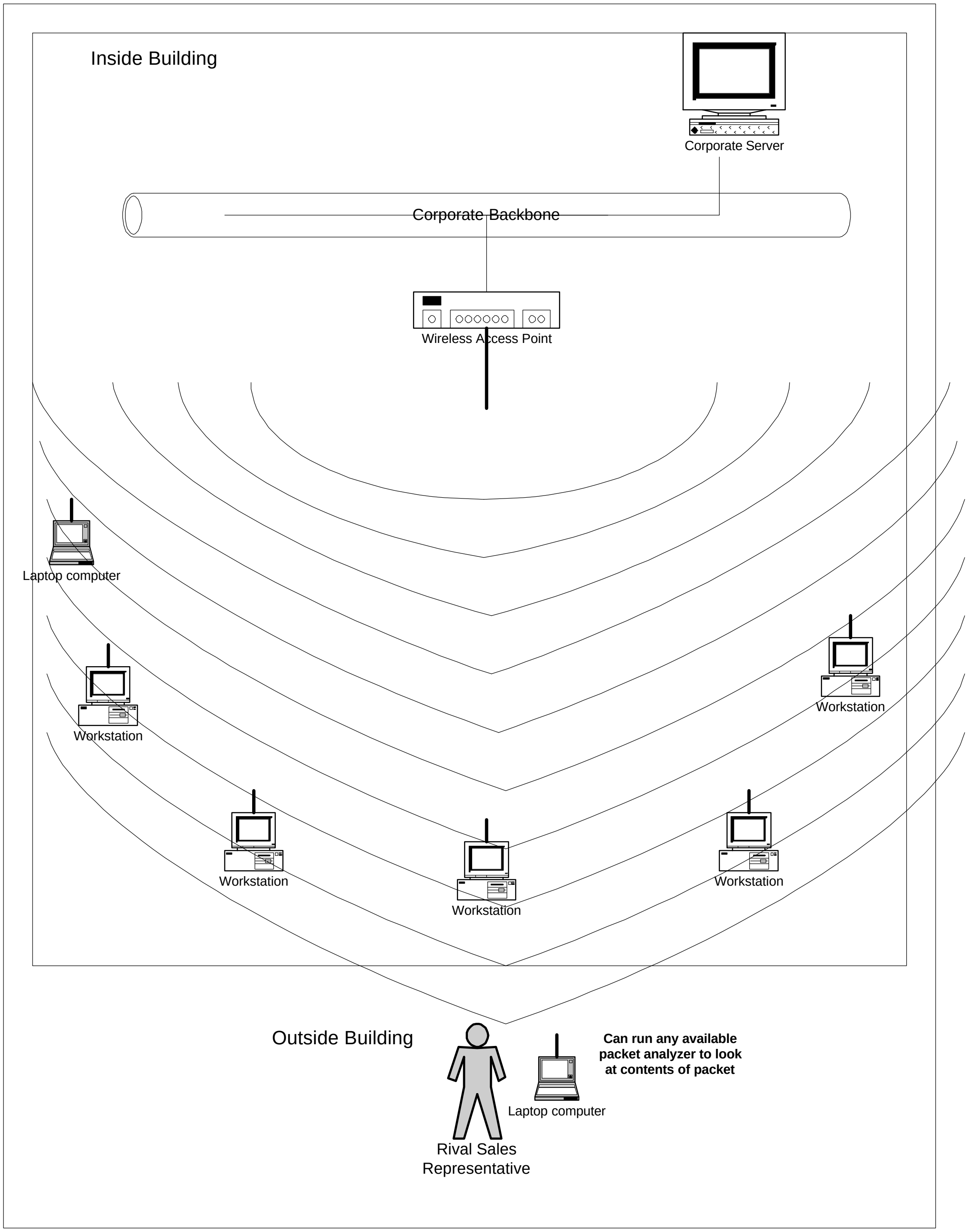


Figure: 3-1: A Wireless LAN without and Security

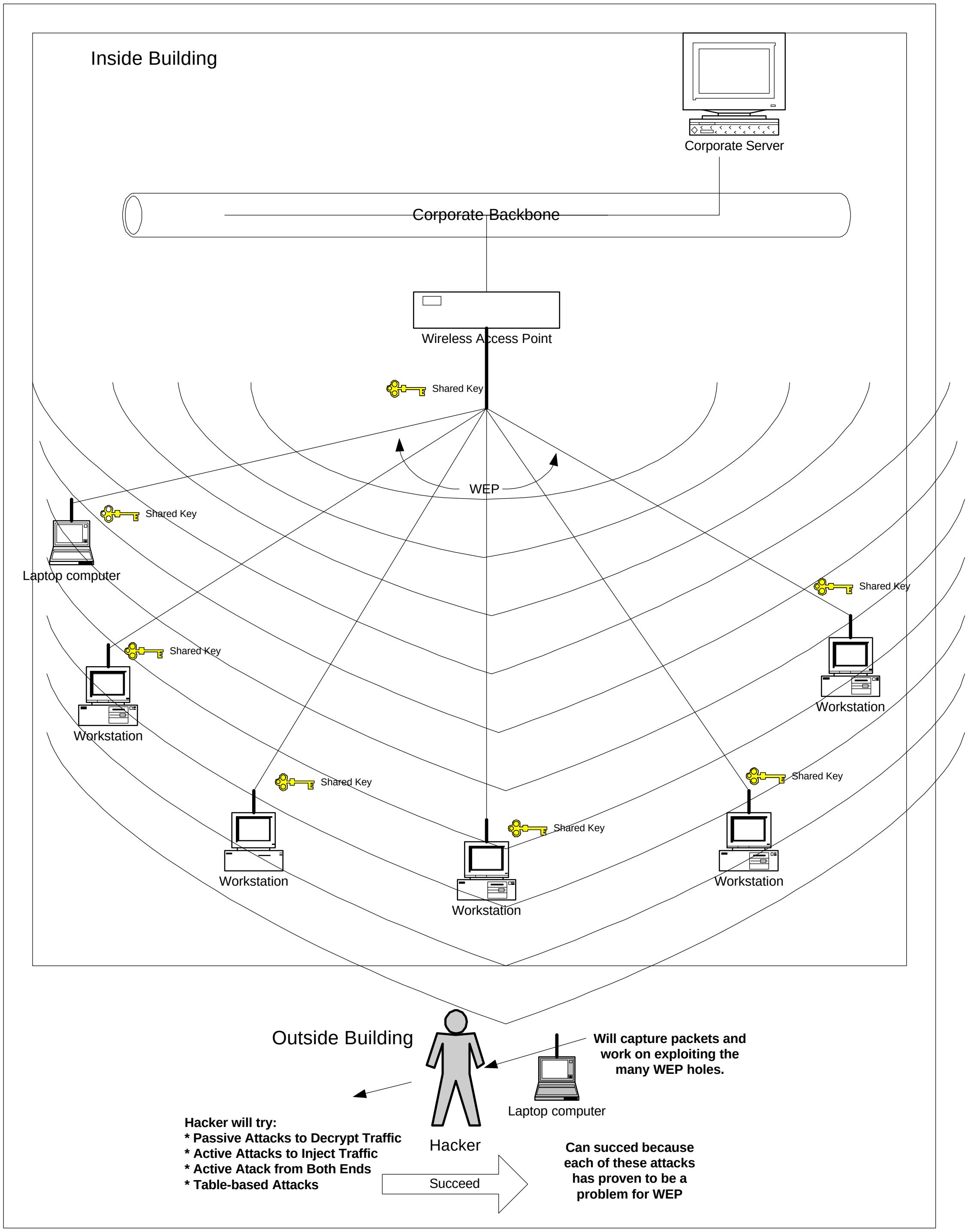


Figure 3-2: A Wireless LAN with WEP

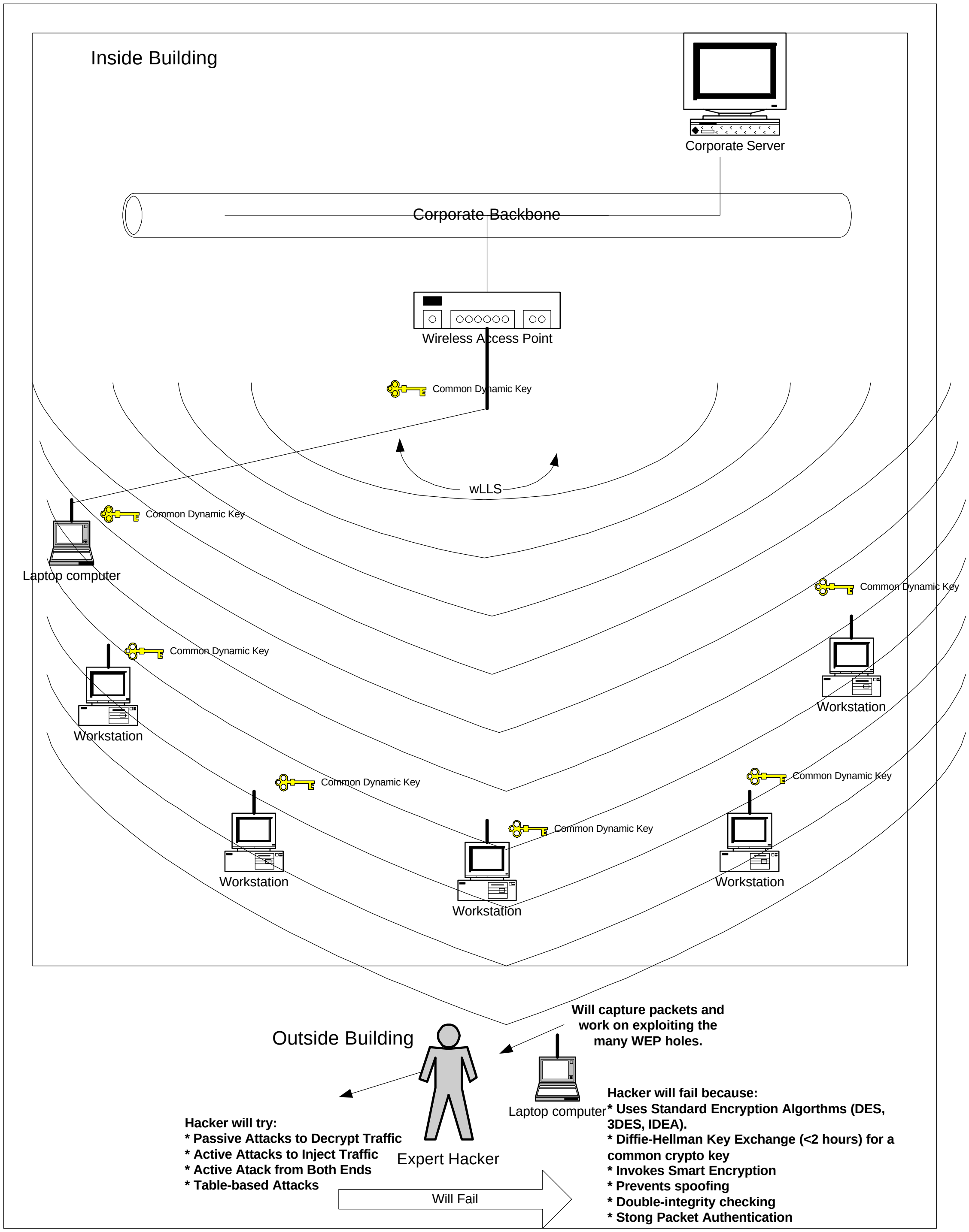


Figure 3-3: A Wireless LAN with *w*LLS

Works Cited

Bob O’Hara and Al Petrick. IEEE 802.11 Handbook: A Designer’s Companion. New York: IEEE 1999