PARANOIA Security Standard for Wireless Networks

Richard Hu **Pius A. Uzamere II** **Fei Xing**

Abstract
Current wireless security is flawed. There is a distinct lack of authentication and encryption. Currently the Wired Equivalent Privacy protocol can be broken with common programs available via the Internet. Furthermore, many system administrators do not know how to properly protect their wireless networks. This paper offers a technical solution, a proposed IEEE standard PARANOIA. PARANOIA incorporates some existing technologies and others that are in the process of being developed. In addition to explaining the PARANOIA approach, this paper examines competition to PARANOIA as well as other modalities that attempt to solve the problem of wireless security.

1. Introduction

Wireless networks are growing in popularity due to price cuts of wireless networking components; wireless cards that were once over $100 dollars now only cost $50. As laptops also grow more popular and less expensive, more and more corporations and homes will use wireless networks. Despite their growing popularity, wireless networks do not offer the same level of security as wired networks. When a person buys a wireless network access point and card, he often assumes that the wireless network will offer the same security and protection that a wired network does. This assumption is false and can lead to many problems.

For example, a study done by PC Magazine in major cities such as New York, Boston, and San Jose showed that only 39% of the 808 networks had the 802.11b Wired Equivalent Privacy protocol (WEP) enabled [1]. Furthermore, some of the wireless access points (APs) did not have the default administrator password changed so the settings to those APs could have been modified to give the attacker full administrative access to the network.

Even more dangerous, the mere fact that WEP is enabled does not provide strong protection against hackers, lulling some users into a false sense of security. To illustrate just how open these networks are, one only needs to consider the availability of easy-to-use cracking tools for WEP. For example, AirSnort is a program that is being distributed at http://airsnort.sourceforge.net that supposedly determines a WEP key in seconds after listening to at least 100 MB of traffic. Current wireless security contains many exploitable flaws. Legislation and policy are not adequate to solve these problems. Our method PARANOIA will maximize security under the current wireless infrastructure.

2. Background on IEEE 802.11 Structure and Flaws

The design of PARANOIA focuses on dealing with the security holes present in the current standard for wireless communication, IEEE 802.11. To ease the later discussion of PARANOIA, we will first talk about how 802.11 works and its vulnerabilities.

IEEE 802.11 provides two modes of connection between stations, ad-hoc mode and infrastructure mode. A station, as Håkan Andersson, from RSA Laboratory points out is "any device that contains an IEEE 802.11 interface to the wireless medium" [2]. Ad-hoc mode allows direct station to station connection, where as in infrastructure mode, the station's communication is mediated by an access point. We will be dealing with infrastructure mode in this paper.

IEEE 802.11 standard includes mechanisms to provide the users with both confidentiality and authentication. Confidentiality, as defined by MIT Professor Frans Kaashoek, is "limiting information access to authorized principals" and authentication is "verifying the identity of a principal for the authenticity of a message (its origin and integrity)" [3]. IEEE 802.11 standard uses Wired Equivalent Privacy (WEP) protocol, a shared-secret key encryption algorithm, to protect confidentiality of wireless data transfer. The standard also provides for authentication through open system authentication and shared key authentication.

2.1 Infrastructure Mode

Infrastructure mode is also known as Basic Service Set (BSS). It provides wireless connectivity all wireless devices within transmission range through access points (see Figure).

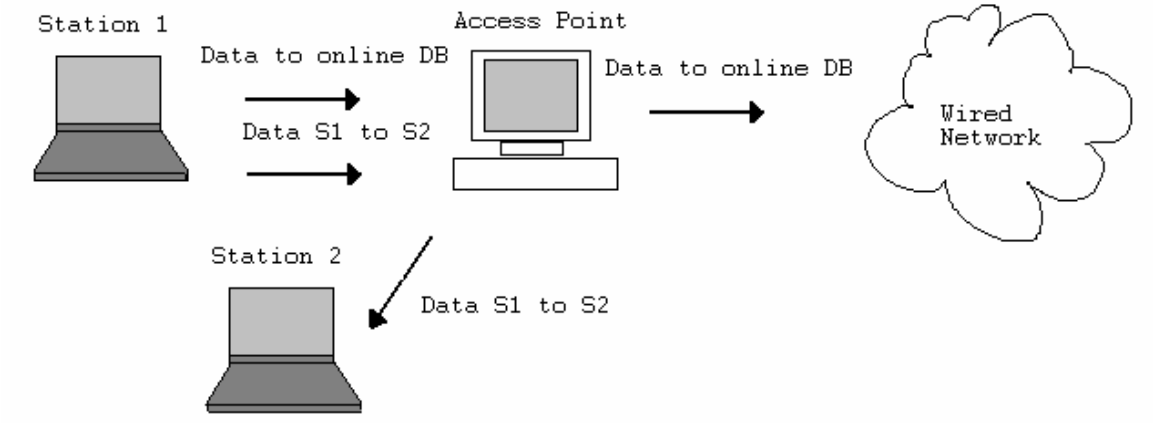

Figure 2-1 Sample Infrastructure Network [4]

An access point forwards data from each station to the appropriate network for either wired or wireless transfer. Access points also allow users to roam within the wireless network -- access points hand off the user's connection from one to another automatically, as the user moves in and out of transmission range of one access point to another. Corporations usually use infrastructure mode networks because users often need services that cannot be provided other stations within transmission range, such as access to remote

databases and the Internet. Connectivity in infrastructure mode also scales up easily with size of network coverage area-- just add another access point.

2.2 Infrastructure Mode Network Details

In an infrastructure mode network, when a station wishes to connect to the network it must find an access point and then establish a connection, or an association, with an access point. Both steps are done via messages, or management frames, sent between the station and an access point. Once a connection has been established the station can exchange data through the access point with the network.

The process of finding an access point and establishing an association has the following three states:

1. Unauthenticated and unassociated,
2. Authenticated and unassociated, and
3. Authenticated and associated [4].

State 1: Unauthenticated and Unassociated:

When the wireless station is searching for an access point via its built-in scanning function, it is in State 1, unauthenticated and unassociated. The station finds an access point either via listening for an access point's beacon management frame or through knowing the access point's unique network name, otherwise named Service Set IDentifiers (SSID). Access points send out beacon management frames periodically to allow a station waiting to connect to find those access points within transmission range. A station wishing to connect to a particular access point with known SSID sends out a probe request management frame to locate the desired access point.

State 2: Authenticated and Unassociated

Once the station finds an access point, both the station and the access point go through a series of message exchange to authenticate each other's identity. Authentication is used to verify that both station have the authorization to communicate within a given transmission range. There are two mechanisms for authentication provided by IEEE 802.11: Open System Authentication and Shared Key Authentication (see Section **2.4** for more details). With open system authentication, a station requests authentication at a particular access point via a message. The access point then determines whether to grant a connection to the station and responds accordingly to the station with a message. Depending on the access point response, the station either proceeds to complete the connection process or discontinues. Unlike open system authentication, shared key authentication uses WEP to determine if a station has access authorization. This method assumes that the station shares a WEP key with the one it is attempting to connect to. The station attempts to connect to an access point by sending an authentication request management frame. The access point responds with an authentication management frame with 128 bit generated challenge text. To proceed with authentication, the station encrypts the challenge text with the shared secret key and sends the text encrypted back to the access point. If the access point is able to decrypt it using the shared secret key then the station is authenticated.

State 3: Authenticated and Associated
After both parties have been authenticated, the station is now in State 2, authenticated and unassociated. To become associated, the station sends an association request frame to the access point, and the access point accepts the request via an association response frame. Now the station becomes part of the network and can send and receive data to the network via the access point. Each station is only allowed to connect to one access point, where as each access point can connect to multiple stations. Once a station is associated with a particular access point, the station may chose to roam to another access point. The station simply repeats the above process to establish an association with the new access point. When a new association is completed, the old connection is broken off.

2.3 Wired Equivalent Encryption Protocol

Wired Equivalent Privacy protocol is the encryption mechanism defined by the IEEE 802.11 standard. WEP is based on RC4 PRNG (Ron's Code 4 Pseudo Random Number Generator) developed by Ron Rivest. It uses a shared secret key for both encryption and decryption for data communication between stations. In IEEE 802.11 the distribution of shared secret key to stations is not standardized. WEP has several known weaknesses that will be discussed later in this section.

2.3.1 WEP Basics

WEP uses the RC4 encryption algorithm, which is a type of stream cipher. It means that the cipher expands a "short key into an infinite pseudo-random key stream" [5]. To encrypt, the plaintext is XORed with the key stream by the sender. Because the receiver has the same key, the receiver can generate the same key stream as the sender. To decrypt the ciphertext, the receiver simply XORs the key stream with the ciphertext.

Using just XORing allows for several attacks so WEP has some built-in defenses. WEP uses an Integrity Check(IC) field to prevent an attacker from changing the plaintext by taking advantage of plaintext-ciphertext one-to-one correspondence. The IC field is computed using 32-bit cyclical redundancy check (CRC-32). Both the plaintext and the checksum are sent encrypted to the receiver. To increase the security of WEP, a 24-bit Initialization Vector (IV) is concatenated with the 40-bit shared secret key to produce a different RC4 key for each packet. This is done to prevent statistical attacks to obtain plaintext on XORs of captured ciphertext encrypted with the same key stream. The IV is sent unencrypted to the receiver.

To send a packet, the sender does the following to prepare the plaintext (see Figure 2-2):

1. Compute the IC using CRC-32 over the message plaintext.
2. Concatenate the IC to the plaintext (M).
3. Choose a random initialization vector (IV) and concatenate this with the secret key.
4. Input the secret key k + IV into the RC4 algorithm to produce a pseudorandom key sequence.

5.Encrypt the plaintext M + IC by doing a bit-wise XOR with the pseudorandom key sequence under RC4 to produce the cipher text.
6.Communicate the IV to the peer by placing it in front of the cipher text. [3]

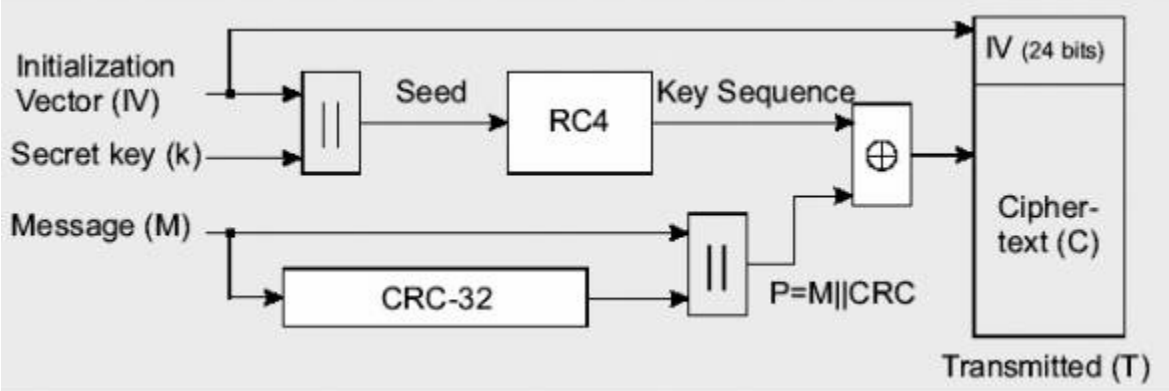

Figure 2-2 WEP Encryption [2]

The actual data sent by the sender is composed of the following encrypted plaintext and IC plus the unencrypted IV (see Figure 2-3).

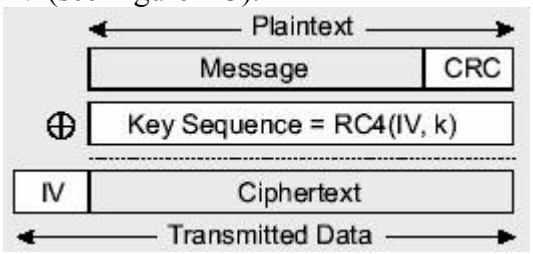

Figure 2-3 Transmitted Data [2]

When the receiver receives the packet, it decrypts the transmitted data using the key stream generated by IV from the packet and its own copy of the shared key (see Figure 2-4). The receiver can check the integrity of the recovered plaintext by computing the IC from the plaintext and compare it with the one from the packet. If the two checksums are equal then the message is verified.

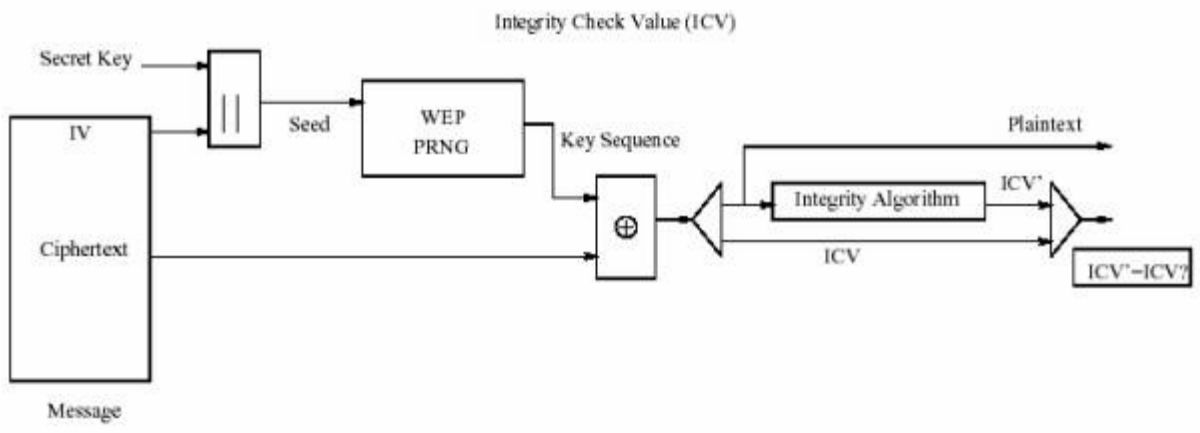

Figure 2-4 WEP decryption [2]

In shared key authentication, the shared key used in data transmission is also used to verify the identity of a station when it attempts to connect to an access point. Open system authentication stations use WEP only for encryption. Both authentication methods are discussed in more detail in Section **2.4**.

2.3.2 WEP Weaknesses

In the previous section, two attacks against RC4 are mentioned: packet modification and statistic attack. Even though WEP theoretically provides defenses against both, they are not implemented correctly so as to provide sufficient protection against attacks.

First, the IC field is designed to protect data integrity but CRC-32 is linear. This means that it is flipping a bit in the message causes a set number of bits to flip in the IC. Nikita Borisov, Ian Goldberg, and David Wagner find that this allows an attacker to do the following:

1. Compute the bit difference of two CRCs based on the bit difference of the messages over which they are taken.
2. Change the message by flipping arbitrary bits in an encrypted message and correctly adjust the checksum so that the resulting message appears valid. This can happened because flipping bits carries through after an RC4 decryption. [5]

The initialization vector is sent unencrypted in a transmitted packet. This allows easily capture of ciphertext with the same IV. The reuse of IV with the same shared secret key cannot be avoided because with 24 bits there are 16777216 total possible IVs – a relatively small number for current computing power. According to Borisov, Goldberg, and Wagner:

> a busy access point, which constantly sends 1500 byte packets at 11Mbps, will exhaust the space of IVs after 1500*8/(11*10^6)*2^24 = ~18000

> seconds, or 5 hours. (The amount of time may be even smaller, since many packets are smaller than 1500 bytes.) [5]

Once an attacker obtains a collection of ciphertexts, he can easily mount a statistical attack to get the plaintext. Vendor specific built-in functions in wireless cards may also increase the chance of IV collision. As noted by Borisov, Goldberg, and Wagner in their work:

> A common wireless card from Lucent resets the IV to 0 each time a card is initialized, and increments the IV by 1 with each packet. This means that two cards inserted at roughly the same time will provide an abundance of IV collisions for an attacker. [5]

Adding to this problem, changing the IV with each packet is defined as optional by the IEEE 802.11 standard, thereby increasing the chance of IV collision. There are a number of known attacks that exploit this weakness of WEP, which will be discussed further in Section **3**.

2.4 Open System Authentication and Shared Key Authentication

IEEE 802.11 standard comes with two methods of authentication or verification of authorization to communicate: open system authentication and shared key authentication.

2.4.1 Open System Authentication and Shared Key Authentication Basics

IEEE 802.11 defaults to open system authentication or the "NULL authentication" algorithm [4]. It allows any station request for authentication at any access point. The access point grants authentication according to its own set standards, it may respond to any station or to only select ones. Stations connected in network using open system authentication may listen to all plaintext data transferred over the network. This type of authentication is used where the network administrator chooses not to deal with security at all and ease-of-use is more important. Some stations may need to authenticate with another station. This can be done by directly sending an authentication management frame to that station. The sending station writes its identity in the frame. The receiving station then responds back with a frame to say whether it recognizes the identity of the sending station.

The second method, shared key authentication, assumes the requesting station and the access point know a shared secret key and authenticates a station using the key to ensure better verification of identity than the open system method. Stations and access points using shared key authentication must use WEP. The shared key is stored in each station in a write-only form. The method of key distribution is not specified in the 802.11 standard.

The authentication process is the following (see Figure 2-5):

1. A requesting station sends an initial authentication request management frame to the access point.

2. When the access point receives an initial authentication frame, the access point will reply with an authentication management frame consisting of 128 bytes of random challenge text generated by using the WEP pseudo-random number generator (PRNG), the shared secret key and the IV. This is sent in the clear without encryption.
3. The requesting station will then copy the challenge text into an authentication frame, encrypt it with the shared key and a new IV, and then sends the frame to the access point.
4. The receiving access point will decrypt the received frame using the same shared key and received IV. It then checks the validity of the CRC checksum and compares the challenge text with that sent in the first message. If a match occurs, the responding station will reply with an authentication indicating a successful authentication. If not, the responding AP will send a negative authentication. [3]

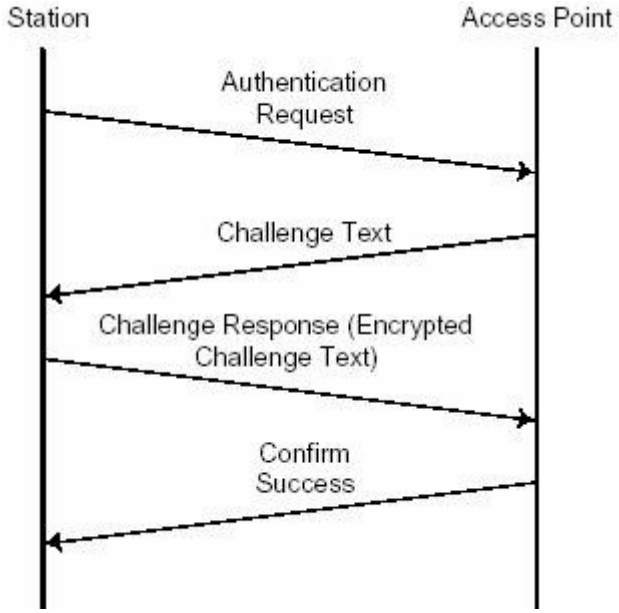

Figure 2-5 Shared Key Authentication Process [2]

The format of the authentication management frame is the same for all messages sent in this process (see tables in Figure 2-6a and 2-6b).

Management Frame Format								
Size in octet	2	2	6	6	6	2	0-2312	4
Field	Frame	Duration	Dest	Source	BSSID	Seq #	Frame	FCS

Name	Control		Addr	Addr			Body	

Figure 2-6a Authentication Management Frame [4]

Frame Body for an Authentication Management Frame						
Size in octet	2	2	2	1	1	128
Field Name	Algorithm number	Seq Num	Status Code	Element ID	Length	Challgenge Text

Figure 2-6b Authentication Management Frame [4]

Both open system authentication and shared key authentication have weaknesses that can be exploited in their design. These weaknesses will be described in the next section.

2.4.2 Authentication Weaknesses

Open System Authentication
Open system authentication is designed to provide an open network; as such the security of network implemented using this method alone has no guarantee of security. The key weakness of this method showed in experimentation by Arbaugh, Shankar, and Wan is that even with those stations that do perform mutual authentication, "the authentication management frames are sent in the clear even when WEP is enabled" [4].

Shared Key Authentication
For shared key authentication, the WEP PRNG used is the "critical component of process, since it is the actual encryption engine" [3]. Unfortunately WEP has several weaknesses that can be exploited as shown in Section **2.3.2**. This makes shared key authentication untrustworthy. Because shared key authentication requires the access point and the connecting station to have the same key, security risks in a LAN increases with the number of access points and connecting stations. As mentioned earlier in Section **2.3.2**, the probability of IV collision increases with number of messages and it also scales up with number of access points and stations. Additionally, because the same shared secret key is used by the entire network, the key needs to be changed frequently to reduce security risk. Unfortunately, under the current IEEE 802.11 standard, changing a shared secret key has to be done manually at every access point. In addition to the fallibility of WEP, the reliability of shared key distribution is also questionable. For example, Arbaugh, Shankar, and Wan cite in their study that "protocols with well-known vulnerabilities, e.g. authenticated Diffie-Hellman key agreement" are included in some vendor products [4].

3. Known Attacks on Wireless Networks

There are two general types of attacks that all networks are vulnerable to: passive attack and active attack. They have the following definitions:

> Passive Attack - An attack in which the intruder overhears a message destined to a principal and makes a copy for analysis.
> Active Attack - An attack in which the intruder can create, delete, and manipulate messages destined for a principal (including substituting one message for another by replaying a message that was copied earlier). [3]

Unfortunately, the weaknesses mentioned in the previous sections combined with the nature of wireless data transmission make wireless networks more susceptible to both types of attacks than wired networks. The attacker has an added advantage in wireless network due to the nature of data transmission over air: wireless network "permits an attacker access beyond the physical security perimeter" [4]. The following analysis is not intended to be an exhaustive listing of possible attacks but rather to provide a starting point for the discussion of securing wireless networks.

3.1 Passive attack

Because data transmission is done over air, eavesdropping is considerably easier on wireless networks than wired ones, "when one sends a message over the radio path, everyone equipped with a suitable transceiver in the range of the transmission can eavesdrop the message" [7]. The hardware needed can be obtained at a reasonable price from vendors such as Sony and Toshiba. Attackers can "listen" to the data transmission but the sender and receiver has no way to detect it.

Once passive attackers gain access to data traffic, they can obtain the information sent through statistical analysis. For example, an attacker can exploit the WEP weakness of repeating IVs decrypt WEP encrypted data. In WEP, the IV is sent in the clear attached to the encrypted payload. An attacker can capture packets with the same IV and obtain the XORs of the packets. "The resulting XOR can be used to infer data about the content of two messages", because "IP traffic is often very predictable and includes a lot of redundancy" [5]. Once the attacker obtains one plaintext message, he can now find the key stream used to encrypt the message. Because a WEP encoded message is a simple XOR of the key stream with the plaintext, with the plaintext and the cipher text the attacker can now obtain the key stream and decrypt all messages with the same IV. The effectiveness of this attack increases with the number of messages with the same IV. The attacker can also increase the effectiveness of this attack via a chosen plaintext attack. An attacker can send a chosen message from outside of the wireless network through the wireless network. He can then capture the encrypted version and obtain the key stream.

3.2 Active Attacks

In this section, we will discuss some of the possible general attacks on wireless networks. This discussion is not intended to be an exhaustive listing of all possible attacks on wireless networks; instead we present some of the more dangerous ones.

3.2.1 Social Engineering

A social engineering attack is defined as the following:

> An outside hacker's use of psychological tricks on legitimate users of a computer system, in order to obtain information he needs to gain access to the system. [6]

For example, an attacker may gain access to a person's password by finding out personal background and/or by mounting a dictionary attack. Such attacks can only be prevented through good personal practices and are not addressed by the technical design of PARANOIA.

3.2.2 Impersonation

To gain access to wireless network, the access point identifies the user's network interface cards (NIC). It does not identify users through passwords. If an attacker gains access to a user's computer even if he logs on as guest, he can then impersonate legitimate users without detection. Unfortunately, under 802.11 per-user authorization is not possible—a guest user may even have the same privilege as the administrator. The attacker can also use a man-in-the-middle attack, "where an attacker is relaying messages between two principals and impersonating the principals to each other" [3]. If an attacker obtains the key stream used to encrypt data, then by exploiting the WEP weakness of repeating IVs, the attacker can also read captured messages and construct new valid messages in his impersonation. Instead of impersonating a user, the attacker may also insert access points as legitimate ones because 802.11 does not require mutual authentication.

3.2.3 Packet Modification, Decryption, and Injection

As shown before, WEP encryption has several weaknesses that can be exploited. An attacker can make use of the linearity of CRC checksum and modify transmitted data by bit flipping. The redundancy in IP packets and predictability of IP packet format also aids the attack by limiting possible content. The attacker may be able to flip the correct bits of the IP header to trick the access point into sending an decrypted version of the packet to an IP address of his own choice. Borisov, Goldberg and Wagner's study shows that even with a firewall, "if a guess can be made about the TCP headers of the packet, it may even be possible to change the destination port on the packet to port 80 which allows it to be forwarded through most firewalls" [5]. Note that the packet will be decrypted by the access point before forwarding to the World Wide Web.

The attacker may be also able to find the key stream used through statistical analysis of captured messages with the same IV. Or, the attacker may use a chosen-plaintext attack to obtain a copy of the ciphertext and calculate the key stream. Once a key stream is found, he can then decrypt all packets encrypted with a discovered key stream. The attacker can also use discovered shared secret key with a brute force method to discover new key streams. With a known key stream the attacker can construct new valid packets to stations and access points with the same shared secret key without detection.

3.2.4 Denial of Service

There are a few ways to launch denial of service attacks on a wireless network. First, the attacker can easily forge disassociation messages since they are sent unencrypted and

unauthenticated. It is easy for an attacker to simply capture a legitimate disassociation message and modify it. Second, the attacker may be able to generate radio interference to disable transmission or by disabling physical connection of access points. PARANOIA is not meant to handle the latter attack; it provides security and not performance optimization.

4. PARANOIA

Our design for PARANOIA incorporates both ideas that are already in the market and novel ideas created by our team. MAC access lists, SSID, public-key encryption, and 802.1x are all features recommended by IEEE and IETF task forces as well as Christopher Murphy of MIT Information Services. The Hand-off Protocol and Ring Authentication are mechanisms that were innovated by our team.

4.1 MAC Access List

Medium Access Control (MAC) addresses are implemented in security for wired networks and should therefore be considered for wireless networks as well [18]. Each network adapter, whether wired or wireless, that conforms to the IEEE standard must contain a unique MAC address. This MAC address is embedded in the device and can be used to identify itself. Currently, the IEEE acts as the registration authority for all MAC addresses. Manufacturers who wish to conform to the IEEE standard register for addresses with the IEEE. There are full or partial MAC address look-ups available on the Internet so given a MAC address, anyone can figure out the manufacturer and type of that network adapter. Some wired networks only allow people with certain MAC addresses to access the network. Certain commercial wireless access points already have this system installed; however, all wireless access points should be required to have this feature in a new IEEE standard.

Wireless networks should utilize MAC address access lists. It is useless to have wireless cards that have unique identifiers and not to have wireless access points that check these identifiers. While it is possible for a firewall or gateway to check the MAC addresses of wireless clients, this practice raises two issues.

The first of these is that MAC addresses are part of the data link layer of Ethernet. Therefore, it would be more reasonable from a system's point of view to check them in the data link layer, not in the application layer, where a firewall resides. Second, by having a firewall or gateway check the MAC address instead of the access point, it still allows the untrusted client to get access to the wireless part of the network for even a few moments. That is enough time for the client to possibly gain control of some other computer on the wireless network or launch a different attack against the wired network. In order to prevent this intrusion, the access point should check MAC addresses against an access list. This access list could be updated periodically from a central server as long as it is the access point that checks the addresses. Also, an access point could be configured to alert the system administrator if there have been too many failed attempts at connecting to that access point. This practice warns the administrator that a possible attack could be occurring on the network.

4.1.1 Why MAC Access Lists Are Not a Singular Solution

MAC address access lists provide machine-level authentication but can still be circumvented. Currently, there exist firmware updates and programs that can change which MAC address is transmitted. An attacker could just run a program to continually change the MAC address of his network adapter until he found a valid one. In addition, an eavesdropper could listen continually to a network with a radio antenna until he found a valid request for connection which would include a valid MAC address. The eavesdropper would then use that MAC address later to gain access to the wireless network. However, using a MAC address access list does prevent random people from accessing the wireless network.

4.2 Service Set Identifier (SSID)

In a secure network, an unauthorized person should not be able to connect to the wireless access point. However, with current wireless conventions, the wireless access points advertise their presence to possible eavesdroppers and hackers – a practice that makes these access points targets for misuse. After a hacker has discovered a wireless network in the area, he can monitor the traffic going through the network and eventually use some kind of attack whether it be known plaintext or chosen plaintext or known ciphertext to crack the WEP protection. SSID prevents idle attacks. Service Set Identifier was developed by Lucent and is implemented in some commercial wireless access points.

Essentially, SSID hides networks by giving them names. If a group of access points form one wireless network, then they must all have the same SSID. In order to gain access to a network that has SSID enabled, the client must know the name of the network. This name is equivalent to having a phone number assigned to the network. Only people who know the phone number can call the network for access. People who do not know the phone number cannot access the network at all. A random attacker who is looking for a vulnerable network will, in all likelihood, pass up a network with SSID enabled because it would require additional resources for him to access the network. This follows the Bovinity Principle suggested by Professor Zittrain of Harvard Law School's Berkman Center [8]: "Small fences keep in large animals." In other words, even seemingly small barriers will tend to deter most people. Most casual eavesdroppers will be deterred because they do not want to attempt to figure out the SSID.

4.2.1 Why Service Set Identifiers Are Not a Singular Solution

SSID is a good deterrent, but there is still a relatively simple way to circumvent SSID. An eavesdropper with a radio antenna could eavesdrop on the frequency and thus procure the SSID for the network because the SSID is sent across the wireless network in plaintext. However, this attack requires that the eavesdropper already have some specially prepared wireless listening device. SSID will deter most casual attackers and "script-kiddies"(people who are well-versed in software but with little experience in hardware).

4.3 Key Management and 802.1x

One of the largest problems with wireless communication is key management. The current encryption system uses a pseudo-random key generator to generate the shared

secret key for both the wireless client and the wireless access point. However, there is an existing shared secret key which forms the basis of the new shared secret key. The problem is that these existing keys are often reused which allows hackers to have an easier time to crack the message.

802.1x solves the key management problem. In 802.1x there are three major components: a supplicant (the client), the authenticator (which is a wireless access point in our case), and an authentication server usually implemented through RADIUS or a similar scheme. The client first attempts to connect to the wireless access point. It responds by first creating a port for passing only EAP (Extensible Authentication Protocol) packets from the supplicant to the authentication server. The wireless access point also suppresses all other ports and traffic from the supplicant such as HTTP, DHCP, etc. The client then sends an EAP-start request to the wireless access point to request the beginning of authentication. The wireless access point, the gatekeeper, responds with an identity request of the client. The client responds with his identity, which the authentication server verifies through some algorithm such as the ring authentication scheme we describe in Section **4.6**. The server then returns either a success or rejection. [9]

Dynamic key exchange, a proprietary addition to 802.1x, adds increased security. Included in the accept message from the access point to the client are session keys. These session keys are used to build, sign, and encrypt an EAP key message that is sent to the client. The client then uses the contents of the message to define applicable encryption keys. This method provides a different key for each session to reduce the probability that eavesdroppers will have enough time to decrypt the key.

802.1x provides a better authentication system than the one that is currently in place because it certifies a user's identity. The current system authenticates the identity of the machine attempting to connect. Unfortunately, all machines are built by some commercial manufacturer. In almost all of these implementations, specific mechanisms to protect against mutating the identity of a machine are non-existent and a machine can be used to pretend to be another machine. However, a user certifies himself with a personal password or personal digital certificate. These are much more difficult to forge because when these methods are used properly, because forging them generally requires personal knowledge of the user. Therefore 802.1x is a better authentication mechanism for wireless networks.

4.3.1 Why Key Management and 802.1x Are Not a Singular Solution

Despite all this improvement in authentication, 802.1x is not an end-all solution by itself. Because the client must send some form of personal identification, the data must be encrypted to prevent eavesdroppers from stealing the data. This will be discussed in the next section. Also, there are two attacks that can bypass 802.1x. The first is session-hijacking where the attacker pretends to be an access point and sends a message to a client telling them that they have been disconnected. The unsuspecting client is actually not disconnected so there remains an open connection to the access point. The attacker can then use that access point until it times out.

The other attack is a man-in-the-middle attack where the attacker pretends to be an access point and fools a client into giving the attacker his information. The attacker then acts like that client to an access point, thus gaining complete access to the network. The first attack is defeated by encryption explained in Section **3.4,** which disallows the attacker from using the session even if the client thinks that it is disconnected. The second attack is discussed in Section **3.7**. [10]

4.4 Public-key Encryption and AES

The encryption afforded by WEP does not provide adequate security for wireless networks. This cipher is a stream cipher, which means that both the sender and the intended recipient must have knowledge of the same stream. In the case of WEP, this stream is based on a random initialization vector and a "shared secret." Unfortunately, as was demonstrated before, an attacker does not need to be aware of the shared secret in order to crack the encryption.

A better implementation would be to utilize public-key encryption such as RSA for the initial contact and use Advanced Encryption Standard (AES), a symmetric block cipher, for further communication. Each access point and each wireless card has some reservoir of private and public key pairings meant for public-key encryption. A client sends an access point a request for connection. The access point then responds with a temporary public key. This public key will have a time to live associated with it on the access point. After the time to live has passed then the client will have to start over in order to connect. Because public keys are such that they do not reveal information about the private key, it is acceptable for the access point to send the public key unencrypted to an unauthenticated client. After the client receives the public key, it encrypts its personal information for authentication (described in Section **3.3**) as well as a temporary public key. The access point uses that public key to encrypt the session key (also described in Section **3.3**) for that connection with that client. Afterwards, all communication is transferred using the session key as the key for AES. This process is shown below in Figure 3-1**:**

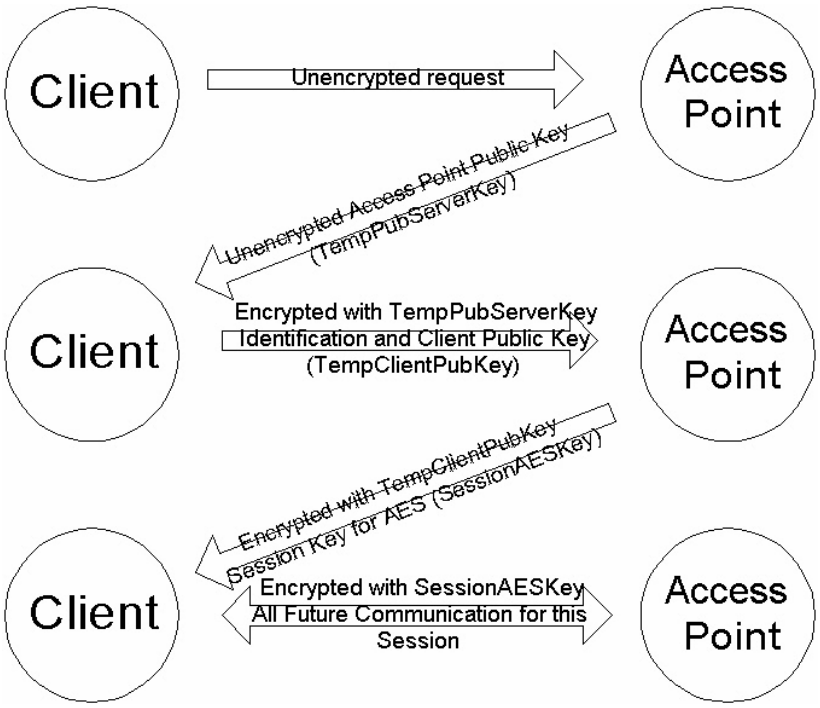

Figure 3-1

AES is a symmetric block cipher which means that it uses the same algorithm and key to both encrypt and decrypt messages. AES or Advanced Encryption Standard is a new standard for encryption as defined by the Federal Information Processing Standard by the NIST. It has been adopted by the US government as the new standard in cryptography. The last standard, DES, was considered difficult to crack for twenty years. DES supported a maximum of 64-bit encryption [18]. In contrast, AES supports far larger keys. It currently supports 128, 192, and 256 bit keys and this maximum can be increased in the future. While not as secure as 256-bit public-key encryption, it is more computationally efficient which is ideal for extended communication between an access point and a client.

Public-key encryption is optimal for connecting to an access point. The problem with shared-key encryption is that both the sender and the recipient must know the shared-key. Unfortunately with wireless networks, there is currently no reliable way implemented to ensure that a party is authentic without sending sensitive information over the network. However, because the client has not been authenticated yet, the message is unencrypted. The primary problem is that there are a great number of clients that may attempt to connect to a wireless access point. By using public-key encryption, all of these unauthenticated clients can receive a temporary key to encrypt their authentication without the hassle of some kind of secure key distribution that is necessary for shared secret key algorithms. Furthermore, this widespread public key does not significantly damage the integrity of the algorithm.

4.4.1 Why Public-key Encryption and AES Are Not a Singular Solution

There remain some caveats to public-key encryption. The first is that with enough time, a public key can be manipulated in a fashion such to reveal the private key. This attack

can be realized in a variety of manners: brute-force computing, known plaintext, or chosen plaintext. The access point solves this problem by only having one active public key and by assigning it a time to live. After the time to live expires, the public key is changed to another randomly determined public key. By doing this, it reduces the window of opportunity for an eavesdropper to crack the private key and by having a randomly selected public key; it becomes difficult for an eavesdropper to happen again across the same key. However, there exist only a limited number of private and public key pairs. If an access point used public-key encryption for all communication with its clients then it becomes much more likely for keys to be recycled. Therefore, after the initial connection and authentication, the access point switches to AES.

4.5 Hand-off protocol

If an administrator cannot prevent an eavesdropper or hacker from accessing the network, the next best thing would be to locate the general vicinity of the hacker or an insecure hub. Then the administrator would be able to secure the hub, manually disconnect the hacker by eliminating the compromised hub, or in the worst case, be able to alert authorities to the exact location of the attacker. However, no mechanism for this type of intelligence gathering exists aside from having the system administrator walk around the entire wireless network with a wireless laptop that is running a packet sniffer. This method is both inefficient and does not give the location of the intruder. By utilizing the hand-off protocol that is used for roaming networks, we can locate where the attack is originating from.

The Hand-Off Protocol that we are proposing takes advantage of existing technologies and merely adds their features to wireless access points. When a computer is logged into a wireless network, it is given information about the strength of the connection to the network. The strength of the connection to the network is determined by the maximum of the strengths of all the wireless access points in the range of the client and the strength of a single wireless access point is inversely proportional to the distance from the client with some normalization with regards to the medium. For example, the strength of a wireless access point is reduced if it has to travel through lead walls instead of air. The client therefore automatically determines which access point to connect to. Therefore, the client has a general idea of the location of the access point. This technology can also be applied in reverse. If an access point were to also be able to determine the signal of strength coming in from a client, it could determine the relative position of the client. Because the access point can also pick up other traffic in its radius, with multiple access points with overlapping coverage, it becomes easier and easier to triangulate the position of the attacker. Furthermore, this information could be logged to a computer that was connected to the wireless access point in a manner identical in the way that a firewall can log who attempts to connect to it.

The algorithm for location with only one wireless access point is simple; the intruder is in a locus of points that form a circle of radius X around the wireless access point, where X is inversely proportional to the signal strength of the client. The algorithm for multiple hubs is demonstrated below in Figure 3-2:

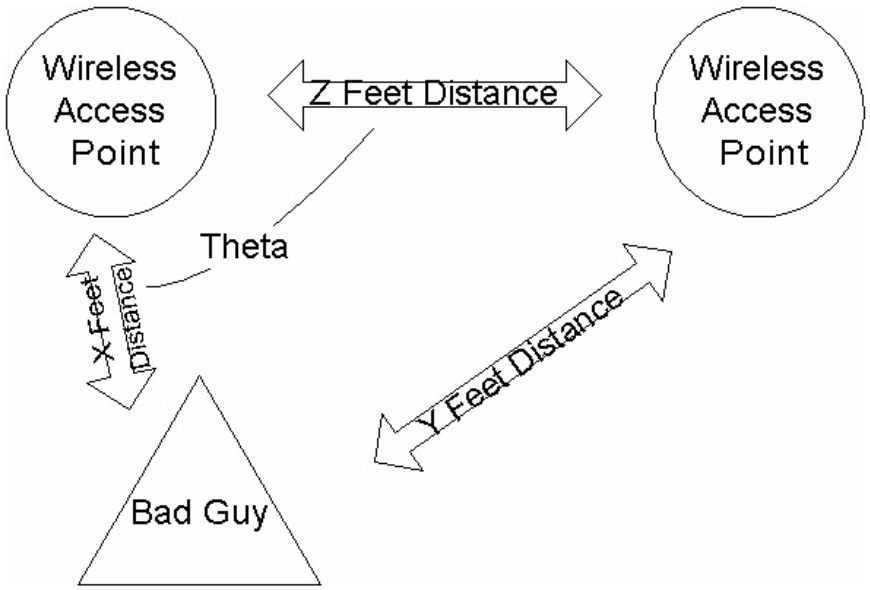

Figure 3-2

X and Y distances can be determined from measuring signal strength. Z is a pre-known distance from when the administrator set up the network. $\theta = \arccos(\frac{x^2 + z^2 - y^2}{2 * x * y})$ from the Law of Cosines. Given this angle and the distance X, we can figure out the location of the attacker because, from geometry, an angle from a line and distance can define only two points. Checking two spots for an attacker requires minimal effort. With three hubs that overlap, we achieve exact positioning by triangulation.

This protocol would be useful for system administrators. If a system administrator notices that an intruder has compromised or is attempting to compromise the network, the administrator can quickly check the logs and see which access point the signals are coming from. An attack can be noticed by a variety of methods which include a spike in the number of failed connect attempts to a particular wireless access point, or a sudden surge in bandwidth going to a particular wireless computer. The administrator could then check the logs for that access point and see the signal strength of the intruding computer. This would narrow down the location of the intruder to a more manageable approximate area. Furthermore, if there are multiple wireless access points, then the approximate area decreases because the system administrator can rule out the areas where other wireless access points have greater strength. This strategy is extremely useful in determining the location of "parking lot" attackers. Parking lot attackers are people who sit right outside a building and siphon bandwidth from a wireless network. Even if the administrator could not apprehend the attacker in time, he could record the data for future notice. This technology is extremely useful for corporate system administrators or anyone else who runs a number of overlapping wireless access points.

4.6 Ring Authentication

PARANOIA proposes "Ring Authentication" -- a new method of authenticating clients to access points based on a type of signature developed by Ronald Rivest, Adi Shamir, and Yael Tauman in their paper, "How to Leak a Secret."

In this paper, Rivest, Shamir, and Tauman introduce the concept of a ring signature, described as such:

> [A ring signature is] a digital signature that specifies a set of possible signers, such that the verifier can't tell which member actually produced the signature. Unlike group signatures, ring signatures have no group managers, no setup procedures, and no coordination: any user can sign on behalf of any set to which he belongs, and he can choose a new set for each message without getting the consent or the assistance of the other members. The only requirement is that each possible signer is already using some public key signature scheme, such as RSA. [11]

Our ring authentication scheme provides a secure, reliable method of authenticating users to the network, while providing a level of anonymity to privileged users. Even if an eavesdropper is targeting a specific user in an effort to utilize a known plaintext attack or chosen ciphertext attack on the user's identity, the effort will be futile because ring authentication provides authentication without anyone knowing the actual identity of the signer. Therefore, the eavesdropper cannot determine which user is signing the authentication and thus cannot target a specific user.

The ring authentication scheme assumes that each user has been assigned a public key and an immutable private key whose sole purpose is to generate signatures and thus will never be transmitted across the network. Here is an overview of the process:

1. The access point beacons will broadcast the public keys of a subset of n clients with permission to access the network. This subset is pseudo-randomly chosen at every broadcast.
2. A client (the *initiator*) wishing to authenticate to the access point (the *respondent*) will add itself to this subset and treat this final set of $n+1$ clients as the ring. The initiator will use its private key and the public keys received in the broadcast to generate the ring signature. This ring signature will be transmitted to the access point.
3. The access point will use this ring signature to verify that all members of the ring (including said client) have permission to access the network. Note that if this ring signature is intercepted, the only information it will yield is a set of possible users who could have requested authentication. Using this information to determine a secret key is intractably difficult.
4. If the ring represented by the signature is deemed to be a subset of the privileged users, then the client is granted access, because this implies that the client is privileged.

4.6.1 Generating the Ring Signature

Outlined here is the algorithm for generating the ring signature using RSA keys, as described by Rivest, Shamir, and Tauman. This outline is taken directly from the discussion they provide of how the algorithm is implemented, but for ease of understanding, certain words ("signer," for instance) have been replaced with their corresponding terms with regards to our application of the algorithm for network authentication purposes.

Let the ring of n+1 clients be the set $R = \{R_1, R_2, \ldots, R_{n+1}\}$ where the client requesting access] is R_c, $1 \leq c \leq n+1$. Given the authentication management frame m to be signed, initiator client's secret key S_c and the sequence of privileged clients' public keys (plus the initiator's) $\{P_1, P_2, \ldots, P_{n+1}\}$, the initiator computes the ring signature as follows:

1) Choose a key: The initiator first computes the symmetric key k as the hash of the management frame m to be signed:

$$k=h(m)$$

2) Pick a random glue value: The signer picks an initialization (or "glue") value v uniformly at random from $\{0,1\}^b$.

3) Pick random x_i's: The signer picks random x_i for all the other ring members $1 \leq i \leq n$, $i \neq c$ uniformly and independently from $\{0,1\}^b$, and computes

$$y_i = g_i(x_i) .$$

4) Solve for y_s: The initiator solves the following ring equation for y_s:

$$C_{k,v}(y_1,y_2, \ldots, y_r) = v$$

By assumption, given arbitrary values for the other inputs, there is a unique value for y_s satisfying the equation, which can be computed efficiently.

5) Invert the signer's trap-door permutation: Fifth, the initiator uses its knowledge of its trapdoor in order to invert g_s on y_s to obtain x_s:

$$x_s=g_s^{-1}(y_s).$$

6) Output the ring signature: The signature on the management frame m is defined to be the $(2(n+1)+1)$-tuple:

$$\{P_1, P_2, \ldots, P_{n+1}; v; x_1, x_2, \ldots, x_{n+1}\}.$$

4.6.2 Verifying a ring signature

The access point may verify an alleged ring signature:

$$\{P_1, P_2, \ldots, P_{n+1}; v; x_1, x_2, \ldots, x_{n+1}\}$$

on the management frame m as follows.

1. Apply the trap-door permutations: First, for $I = 1,2, \ldots, n+1$ the access point computes

$$y_i = g_i(x_i) .$$

2. Obtain *k*: The access point hashes the message to compute the encryption key *k:*

$$K=h(m).$$

3. Verify the ring equation: The access point checks that the y_i's satisfy the fundamental equation.

$$C_{k,v}(y_1,y_2, \ldots , y_r) = v$$

If this ring equation is satisfied, the access point accepts the signature as valid and the initiator is authenticated. [11]

4.7 The Complete PARANOIA Standard

So far, each method that we have discussed in this paper on how to protect a wireless network has some sort of flaw; none of them provide *ipso facto* bulletproof security. MAC addresses can be imitated and forged. SSID's can be sniffed with a radio antenna and the proper hardware. 802.1x is vulnerable to man-in-the-middle attacks and to session-hijacking. Lastly, encryption is always vulnerable with enough brute force attacks. However, when these four are combined and used in conjunction with the hand-off protocol and the ring authentication scheme, they form a formidable yet still easily implementable standard.

A typical wireless session with our PARANOIA standard works as such.

1. A client will send out a request for network "X".
2. A wireless access point for network X will reply with its public key for encryption purposes. It is important to notice that the client will not see wireless network Y because it has not specifically asked for wireless network Y. The public key for that wireless access point has a time to live counter on it before it changes. This change prevents someone from having enough time to brute-force the key.
3. The client now sends its MAC address over in a message. This message is encrypted with the access point's public key.
4. If the access point determines that this MAC address is on its access list, then it will be using 802.1x to authorize the user.
5. The client encrypts its user identity and authentication again with the current temporary public key and finally the access point returns with either an accept or reject management frame. The authentication is via the ring authentication scheme whereby the client attaches his signature to the frame in such a way that an eavesdropper cannot listen and attack a specific user.
6. With dynamic key exchange, a key is assigned to this user just for this session. This key becomes the shared secret key for AES for all communication between the access point and this client.

PARANOIA first prevents all sorts of casual attackers. The only known way that SSID can be compromised is with hardware, specifically some sort of radio antenna and a

computer attached to it. This protection inhibits attacks from casual attackers such as "script-kiddies" who may be talented in software and cracking encryption algorithms but who may not have the hardware or knowledge available to build the necessary equipment to eavesdrop on wireless traffic. Furthermore, depending on the number of connections and disconnections to the network, the number of opportunities to eavesdrop the SSID could be very small. For example, if a wireless network is for home use, the person using the network may only connect to it once or twice a day because that person is merely roaming around his apartment with his computer and has no need to repeatedly connect and disconnect. Secondly, even if an attacker manages to intercept the SSID, the attacker still has to imitate a valid MAC address. Previously, an attacker could gain this knowledge in a similar manner to figuring out the SSID, via radio antenna. However, with our new system, this MAC address is encrypted with public-key encryption with a changing public key. Therefore, in order to gain access, an attacker has to either crack the public-key encryption which is extremely difficult or an attacker has to listen to the network and capture someone's MAC address identification message which is encrypted with a certain public key and wait until the wireless access point broadcasts that same public key. This process takes a significant amount of time and therefore is not conducive to casual attackers.

PARANOIA also protects against more dedicated and malicious attackers. By using public-key encryption with a public key that has a certain time to live, the network is protected against code-crackers. Even a weaker algorithm such as DES, although not a public-key algorithm, takes three days to crack using a special $250,000 DES cracking machine [19]. Assuming similar time to crack the public-key encryption algorithm that we chose to use, it will take a cracker three days to defeat one public key. However, there are many other public keys to consider. This forces an attacker to be extremely dedicated and to have a good deal of time. If an attacker merely wants to eavesdrop on the content between the client and an access point, then the attacker has to be able to attempt to defeat AES encryption. AES encryption is rumored to be extremely difficult to decrypt because of the size of the keys. Currently, DES, which can be cracked in three days, uses 56 bit keys. This means that there are $7.2 * 10^{16}$ possible keys. However, AES uses 128 bit keys, which means that there are $3.4 * 10^{38}$ keys. This increase is 21 orders of magnitude greater than DES. Furthermore, AES also contains support for 192 and 256 bit keys as well as scaling for even larger keys if necessary. In addition, the PARANOIA ring authentication algorithm hides the identity of the actual signer from any hackers.

Lastly, PARANOIA provides mechanisms that limit the amount of damage that an attacker can do. If AES proves to be faulty and there exists some clever way to crack it in the same amount of time it takes to crack DES (we are assuming that AES will not be worse than the existing federal standard), 802.1x provides for per-session keys such that an attacker will only be able to listen to one user's session at a time. If it becomes known that a certain conversation is being compromised, then the client can merely disconnect and reconnect for a new randomly selected session key. This action forces the attacker to start the entire process again. Also, with the hand-off protocol, an administrator who has discovered suspicious activities on the network can figure out the approximate position of

the attacker. Because we have already established that cracking PARANOIA requires a great amount of resources and ability, chances are that someone who can crack PARANOIA will probably not be attacking the home network of the average user. Instead, this hacker would use his abilities to crack the network of a large corporation to justify the cost in cracking the network. This large corporation will be likely to have many hubs and some dedicated administrators. This combination together with the PARANOIA hand-off protocol increases the accuracy of determining the location of the hacker and thus catching the attacker before major damage is done.

Like any security system, PARANOIA has its downfalls. While managing the MAC address access list, SSID, and public-key encryption for initial connection requests are simple and atomic matters, utilizing 802.1x for session keys and the hand-off protocol for catching attackers requires active effort on the part of the system administrator. However, we feel that MAC address access lists, SSID, and public-key encryption are enough of a deterrent for casual attackers who might attempt to raid a home network. Any company who decides to use the PARANOIA standard must already have a system administrator to regulate their wired network as well. Therefore, PARANOIA gives sufficient coverage to both the commercial and civilian area. Another pitfall of PARANOIA is the computing power required. Both public-key encryption and AES require much more computing power than DES which is why they have not been used for communication yet. Currently, it is believed that AES can run on a separate math co-processor. Therefore, we believe that with the inclusion of Moore's Law, it should not be long before it is possible to either include a math co-processor on a wireless card or for the primary chip to be able to handle the computations.

There remains one attack that is effective against PARANOIA, namely the man-in-the-middle attack which was discussed in Section **3.3**. The only apparent way to guard against this attack is to implement some kind of authentication mechanism on the access point side such that each access point also has to reliably authenticate itself to each wireless client (i.e. mutual authentication). We feel that this is an overly expensive operation to perform. Each wireless access point must be assigned a unique identifier. Furthermore, each wireless card will have to know the name of each wireless access point that it can connect to. For a large network, this process will be difficult to synchronize because if one access point changes, then all wireless cards in the building will have to be modified whereas if one laptop is added to the network, since all access points route to a central server, only the server needs to be modified.

In addition, adding SSID gives the client enough warning to know that something is amiss if an attacker tries this attack. If a client asks for network X and receives two different responses, it knows something is amiss. Even if network X had two wireless access points which overlap, the hand-off protocol and the way current roaming works dictates that only the closer hub would respond based on signal strength. Therefore, if a client sees two hubs respond, then something is wrong and the client is made aware to be extremely careful. We feel that this process conveys enough of a warning to a client.

5. Alternatives to PARANOIA

There are other proposed solutions to the wireless problem currently on the market, but all of these have deficiencies that lead the authors to believe that our *sui generis* PARANOIA approach is the best solution. The other attempts fall into three main categories: virtual private networks, commercial wireless security schemes, and the IEEE's new TKIP.

5.1 Virtual Private Networks (VPNs)

As PC World describes it,

> A virtual private network is a secure connection between two segments of a network, with one end being your office's network gateway (an entrance to the network, such as a router), and the other end being your PC or a gateway to another network, say, in a remote office. Those two segments connect over a public network . . . . [13]

One problem with VPNs is that, although they can provide some level of security for small wireless networks, they can prove to be too costly and difficult to put into practice for larger networks [13]. This is largely due to the fact that the way most VPNs are implemented does not bundle the VPN into the access point, but rather places wireless network users in the same situation as remote dialup users in the sense that they must authenticate to the virtual private network or firewall. One way of looking at this is that the access point is "behind the firewall."

Unfortunately, the topology of most high traffic volume networks is such that access points act like repeaters that bounce packets from access point to access point, allowing communication to route around the network before authentication even takes place. An unauthenticated user can then intercept data passing along these insecure access points, even though the unscrupulous user may not be able to defeat the VPN-provided security along the backbone of the network. [14]

Overall, VPNs have some useful features, but they are no replacement for a well-implemented wireless standard. Rather, they are a well-suited solution for network administrators who have resigned themselves to maintaining an insecure network and want to try to "batten up the hatches." Using a VPN to lessen the risks of an insecure wireless protocol is analogous to maintaining a bank vault with a broken lock and consoling one's self by putting up an electric fence with barbed wire around the bank; it really does not get at the heart of the matter, but works mainly as a stopgap measure.

5.2 Commercial Wireless Security Schemes

All of the commercial schemes have an inherent problem: they are proprietary. This means that they generally require special client software and same-manufacturer access points and cards. This is a big problem for network owners who have already invested in different brands of equipment or who simply wish to have freedom of choice in which products they purchase. Also, this tends to lead to a clunky interface, rather than a transparent security protocol.

5.3 TKIP

TGi, the task group convened by the IEEE, has proposed a new temporary solution called Temporal Key Integrity Protocol (TKIP). This protocol is interesting in that it would be not only standardized, but would work with the existing wireless equipment by way of firmware upgrades. TKIP uses a fast-packet re-keying approach, whereby the encryption keys are changed frequently (approximately every 10,000 packets). This helps alleviate the problem, but does not solve it completely. It still relies on the inherently poor encryption of WEP for keys. This means that the system is still very breakable. Primarily, TKIP is being endorsed because it can be quickly deployed; it is more of a business solution to keep vendors happy than a technical solution to keep wireless networks secure.

6. Using the Law, Social Norms, and the Market to Aid in Securing Wireless

There are four fundamental modalities of effecting change in cyberspace, namely code, law, social norms, and market [12]. In the arena of wireless networking, it appears that the strongest methods of improving security lie in the code. However, the modalities of law and social norms can also be powerful when employed in conjunction with PARANOIA.

United States law already has provisions dealing with the breaching of security of networks in general. These provisions provide a *prima facie* proscription of breaching or aiding and abetting in the breach of the security of networks – wireless or not. Stronger language could be added to these provisions that will add incentive to maintainers of large networks to keep their networks secure.

One existing piece of doctrine on this matter is the Computer Fraud and Abuse Act. This section of the US Code (until being revised by the National Information Infrastructure Protection Act of 1996) proscribed the transmission of information over networks with "reckless disregard of a substantial and unjustifiable risk that the transmission will [be used to cause damage or denial of service to computer users and systems]," making this action a federal offense. The current law is less broad, specifically proscribing only intentional harm rather than negligence unless trespass is involved. [15,17]

We feel that this stronger language with modification should be re-introduced to §1030. What is the opposition to this? One issue is that administrators are afraid that with such statute, system administrators hosting blatantly insecure wireless networks will be prosecuted. After all, it can successfully be argued that the current wireless infrastructure harbors a "substantial and unjustifiable" risk of abuse and denial of service attacks. Following from these concerns, there are two primary facts to consider while drafting legislation along these lines. First, there is currently no standardized alternative to the current insecure standard. Thus, the government would be hard-pressed to prosecute anyone for maintaining an insecure network, lest it require that network administrators invent new security measures. This, of course, is unrealistic.

The second and perhaps more important of the reasons is this: in practice, the government loathes the idea of imprisoning private citizens for what amounts to not being computer savvy enough. To illustrate, imagine a family that owns and operates a small café. Eventually, this family decides to foray into the internet café niche by installing a wireless network. Unfortunately, the default configuration of the network is insecure. If a hacker piggybacks the network's bandwidth for illicit purposes, should the family be prosecuted in federal court? The intuitive answer to this, is no.

Thus, the government rarely if ever prosecuted for this offense when it existed in §1030. Presently, this issue is most directly addressed by what one may call "private sheriffs" [8]. A case in point is the not-for-profit organization called MAPS (Mail Abuse Prevention System). This group maintains, among other things, a list of mail systems that maintain "open relays." Open relays allow users to send email without authentication. According to MAPS, "they unwittingly provide a conduit between a spammer and some number (usually a very high number, tens or hundreds of thousands) of spam victims" [16]. MAPS does more than maintain a list however; they encourage subscribers (mainly internet service providers or ISP) "blackhole" the perpetrators' servers. To have a server blackholed is to have all packets originating from the servers dropped. This effectively makes communication using the server in question impracticable.

Private sheriffs notwithstanding, it is clear that if these proposed amendments to §1030 were enforced, there would be incredible pressure on network administrators to ensure the security of wireless networks. The government should commission a taskforce to analyze the status of wireless standards on the market (e.g. PARANOIA) and use its findings to answer a simple question: Are there reasonable methods of maintaining secure wireless networks on a large-scale? When the answer to this is yes, the provision recommended should be brought back to §1030. To make this law enforceable, the legislature should add language to §1030 that stipulates that owners of large networks (as defined by some appropriate threshold) are the ones to be held responsible if they show reckless disregard for risk of abuse of their networks. This eliminates the problem of the law causing any person or small business with a router to fear criminal penalties for their inexperience, while targeting the most fertile grounds for large-scale mayhem caused by hackers. Furthermore, this will send a clear message to system administrators that, in order to be indemnified from the consequences of §1030, they must put a good faith effort towards adopting standards with adequate security – especially when the solutions are as cost effective as PARANOIA.

Enforcement of the new, stronger §1030 combined with standardization of PARANOIA will cause the PARANOIA standard to proliferate quickly among enterprise level vendors. Eventually, the implementation of the new standards will trickle down to the average consumer, leading to a significantly more secure wireless infrastructure. Along with this widespread use of PARANOIA will come an increased expectation of wireless security from the public, leading to compelling market pressure to keep security as a high priority. Hopefully this will lead network administrators and future wireless ISPs to develop internal best practices policies that will aid in preserving a reasonable level of security.

7. Conclusion

In this paper, we have discussed the flaws behind the current 802.11b system as well as our technical implementation for the IEEE PARANOIA standard which will improve wireless network security. Through the use of a diverse set of tools to combat different attack strategies, we compensate for the deficiency of each individual tool and eliminate different attack strategies. In the end PARANOIA consists of six major tools: MAC access lists, SSID, public-key and AES encryption, 802.1x for key management, hand-off protocol for detecting intruders, and ring authentication for hiding the identity of the user connecting. We also look at other modalities as solutions to the wireless security problem and come to the conclusion that implementing PARANOIA in conjunction with market pressure to accept this new standard is the best method to achieving wireless security.

Acknowledgements

We would like to thank all the professors of MIT's *Ethics and Law on the Electronic Frontier* and of Harvard Law School's *Internet and Society* for introducing us into the real world where technology has to integrate with not only other technology but also with law and doctrine. We would especially like to thank Professor Hal Abelson of MIT and Joe Pato of Hewlett-Packard's Internet Security Laboratory for advising us on this paper.

References

[1] Craig Ellison. *Wireless LANs at Risk.* PC Magazine April 9, 2002.

[2] Wireless LAN. Katholieke Universiteit Leuven, 2001. http://www.esat.kuleuven.ac.be/~brodiers/h239/security.php

[3] J. H. Saltzer, and M. Frans Kaashoek. Topics in the Engineering of Computer Systems. 6.033 Class Notes, Draft Release 1.14. MIT EECS Department: Cambridge, MA, 2002.

[4] Arbaugh, Shankar, and Wan. *Your 802.11 Wireless Network has No Clothes*. Uniersity of Maryland, 2001. http://www.cs.umd.edu/~waa/wireless.pdf

[5] Borisov, Goldberg, and Wagner. *Security of WEP Algorithm.* Berkeley University. http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html

[6] Palumbo, John. *Social Engineering*. SANS Institute, 2000. http://rr.sans.org/social/social.php

[7] Uskela, Sami. *Security in Wireless Networks*. Helsinki University of Technology, 1997. http://www.tml.hut.fi/Opinnot/Tik-110.501/1997/wireless_lan.html#Threats

[8] Lecture from Professor Jonathan Zittrain. Harvard Law School 2002.

[9] Geier, Jim. *802.1x Offers Authentication and Key Management*. 802.11Planet.com 2002. http://www.80211-planet.com/tutorials/article/0,4000,10724_1041171,00.html

[10] Schwartz, Ephraim. *Researchers Crack New Wireless Security Spec*. InfoWorld 2002. http://www.infoworld.com/articles/hn/xml/02/02/14/020214hnwifispec.xml

[11] Rivest, Shamir, and Tauman. *How to Leak a Secret*. MIT 2001. http://theory.lcs.mit.edu/~rivest/RivestShamirTauman-HowToLeakASecret.pdf

[12] Lessig, Lawrence. The Future of Ideas. Random House, 2001.

[13] *Virtual Private Networks*. PCWorld 2000. http://www.pcworld.com/hereshow/article/0,aid,15915,00.asp

[14] Wexler, Joanie. *VPN and Wireless LAN Security*. NetworkWorld 2001. http://www.nwfusion.com/newsletters/wireless/2001/00960610.html

[15] Computer Fraud and Abuse Act §1030(a), 18 U.S.C.

[16] http://mail-abuse.org/rbl/candidacy.html#ByRelaying

[17] U.S. Department of Justice, Legislative Analysis of the 1996 National Information Infrastructure Protection Act, 2 Electronic Info. Pol'y & L. Rep. 240, 240 (1997). Also available here: http://www.usdoj.gov/criminal/cybercrime/1030_anal.html

[18] Interview with Christopher Murphy, MIT Information Services

[19] *Fed Encryption Standard Exposed*, Wired Magazine http://www.wired.com/news/technology/0,1282,13800,00.html