K.K.WAGH COLLEGE OF EDUCATION AND RESEARCH

Seminar Report on

# Wi-Fi (802.11) SECURITY

Seminar By –
Jigar Shah

Guided by –
Prof. G. K. Kharate

**K. K. Wagh**
**COLLEGE OF EDUCATION AND RESEARCH**
**NASHIK**

# *CERTIFICATE*

This is to certify that

Mr. Jigar A. Shah

Has Successfully Delivered

Seminar on

## “Wi – Fi Security”

Towards the Partial Fulfilment of Bachelor’s

Degree In Information Technology

At Pune University

During Academic Year 2004 - 2005

| | | |
|---|---|---|
| **Prof. J. K. Bharadwaj** | **Prof. G. K. Kharate** | **Prof. Nandurkar** |
| [Seminar Guide] | [H.O.D] | [Principal] |

## ABSTRACT

As a security-conscious network manager, you've listened to vendors when installing their new wireless LAN products. You're confident about your network security and sleep fairly well at night. That's fairly well.

Then, one morning, you come in the office only to find your entire corporate Web site replaced by hacker's web page. The resulting audit reveals that the attackers came in through the wireless network. But how?

One of the ugly truths vendors don't tell you is that wireless networks are inherently less secure than their wired counterparts. Despite what you do, putting network ports in the air opens holes in your network that weren't previously there - holes that must be plugged. It's like putting Ethernet jacks on the outside of the building. What are you to do? Most people won't argue about the incredible convenience and productivity increases possible with wireless networks, but no one can afford that convenience at the expense of compromised network security.

## KEYWORDS

Wi – Fi Wireless Fidelity

| | |
|---|---|
| WECA | Wireless Ethernet Compatibility Alliance |
| WEP | Wired Equivalent Privacy |
| WPA | Wi – Fi Protected Access |
| VPN | Virtual Private Network |
| STA | Station (Wi – Fi client machine like Laptop) |
| AP | Access Point (Central Hub for connecting Wireless STA) |
| QPSK | Quadrature Phase Shift Key (Modulation Scheme for Wi – Fi network) |
| RSN | Robust Secure Network |

# INDEX

# INTRODUCTION

Because of their flexibility, affordability, and ease of installation, the use of wireless local area networks (wireless LANS, WLANs, and Wi-Fi) are increasing at a tremendous rate. MDR (research group) estimates, there are currently more than 75 million wireless LANs in use worldwide. META Group and MDR estimate that 95% of corporate laptop computers that will be shipped in 2005 will be equipped for wireless operation. An equal amount of wireless support devices, such as access points, routers, printers, scanners, and handhelds, are also being produced to meet the demand for wireless.

With the increasing deployment of 802.11 (or Wi-Fi) wireless networks in business environments, IT organizations are working to implement security mechanisms that are equivalent to those existing today for wire-based networks. An important aspect of this is the need to provide secure access to the network for valid users. Existing wired network jacks are located inside buildings already secured from unauthorized access through the use of keys, badge access, and so forth. A user must gain physical access to the building in order to plug a client computer into a network jack. In contrast, a wireless access point (AP) may be accessed from off the premises if the signal is detectable (for instance, from a parking lot adjacent to the building).

# 1. OVERVIEW OF Wi-Fi

The IEEE 802.11b standard is a specification for Wireless Local Area Networks (WLAN). The Wireless Ethernet Compatibility Alliance (WECA) acts as a certification organization for products that interoperate with one another via the IEEE 802.11b standard. Products that achieve certification are deemed Wi-Fi compliant. Wi-Fi systems transmit data in the unlicensed 2.4GHz ISM band. Data is transmitted on BPSK and QPSK constellation at 11Mbps. Wi-Fi uses Direct Sequence Spread Spectrum (DSSS) technology for transmission.

WLAN can operate in two modes: Ad - hoc mode and Infrastructure mode. In ad-hoc mode, stations communicate directly with each other. An ad-hoc network is formed “on the fly”, when mobile devices within proximity of each other have a need to communicate and no pre-existing network infrastructure is in place near their location. An ad-hoc has no connection to the “outside world”. Fig 1.1 shows operation of Wi – Fi in Ad hoc mode.

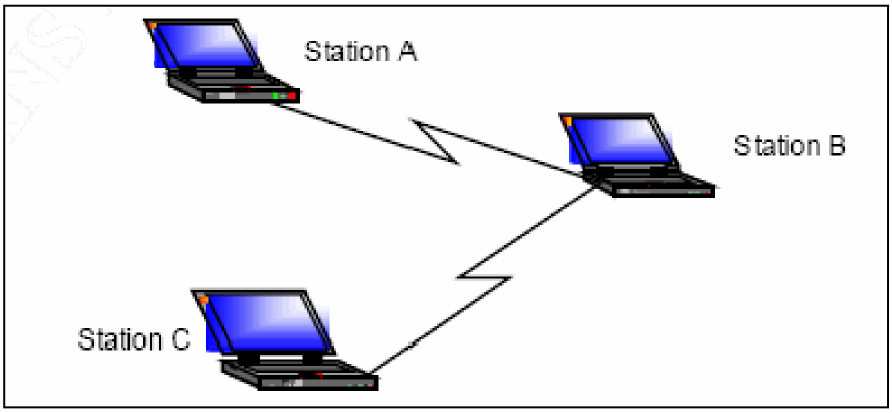


*Fig. 1.1: WLAN in Ad hoc mode*

In Infrastructure mode there is an Access point (AP) which acts like a central wheel in distributing data. This is known as infrastructure mode because the access point is coordinating the Wi – Fi LAN from a fixed point and often providing a connection to wired Ethernet network. Fig 1.1 shows operation of Wi – Fi in Ad hoc mode.

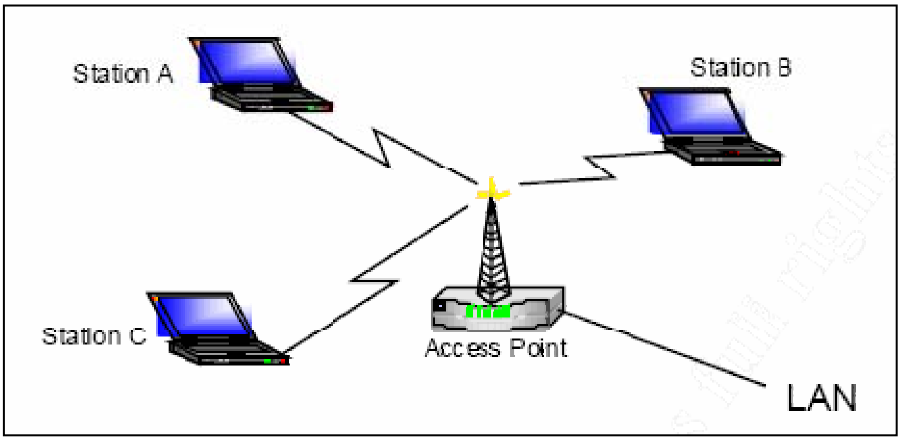


*Fig. 1.2: WLAN in Infrastructure mode*

The first WLAN standard was adopted in 1997. This standard defined the Media Access Control (MAC) and Physical (PHY) layers for a wireless LAN.

*Media Access Control Layer (MAC)*

The Media Access Control Layer provides three services:

- Reliable data delivery from the physical wireless media to the upper layers of the OSI reference model.
- A controlled access method from the upper layers to the wireless media. This method is called Carrier-Sense Multiple Access with Collision Avoidance
- (CSMA/CA). CSMA/CA is similar to the collision detection access method used in 802.3 Ethernet LANs.
- The authentication services and the Wireless Equivalent Privacy (WEP), which is the encryption service for data transmitted on the WLAN.

*Physical Layer (PHY)*

The physical layer is the interface between the MAC and the wireless media where frames are transmitted and received.

> *"As wireless networks become ubiquitous extensions of wire networks, problems with rogue access points will wane — though accidental network associations and attacks against mobile laptops will increase. This makes it very important to understand the risks of wireless LAN laptops and other devices that are present in every organization."*
>
> **Gartner Research**

## 2. Threats to WLAN Network

As wireless LAN deployments increase, so does the challenge to provide these networks with security. Wireless LANs face the same security challenges as their wired counterparts, *and more*. Because the medium for wireless is air, wireless LANs' have the added issue of securing data that travels the airwaves. This has given momentum to a new generation of hackers who specialize in inventing and deploying innovative methods of hijacking wireless communications.

### *3.1 Types of attacks*

Best way to start understanding security for any system or protocol is by learning possible attacks on it. Some of these attacks are similar to attacks on wired network but are done in unexpected way.

Attack methods can be classified as "passive" and "active". Passive attacks include eavesdropping (i.e. snooping). Active attacks are subdivided into "forgery", "message modification," and "denial of service".

**Snooping** as the name suggests, is simply accessing private information. This information could be used for an advantage, such as getting company secrets to help your own business or stock purchase information. Encryption can be used to make snooping difficult.

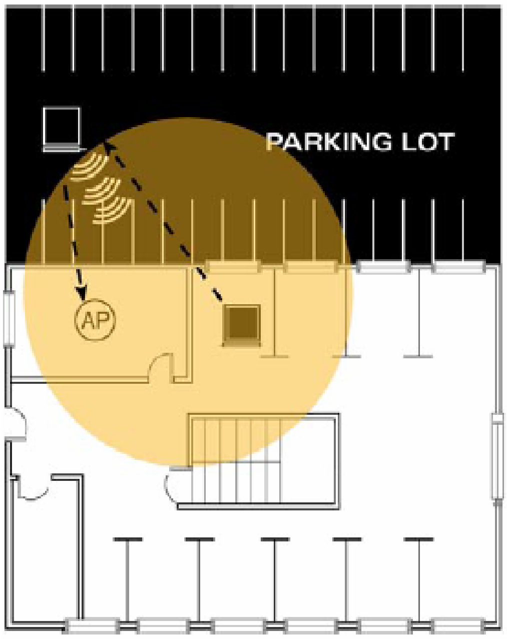
> *"Wireless LANs are too easy to install and manipulate, and users and criminals will continue to take advantage of opportunities to disrupt or damage enterprise networks."*
>
> ***Gartner, September, 2003***

The attacker is required either to know the secret encryption key or to use some clever technique to recover the encrypted data.

**Modifications** to data can be achieved in some no obvious ways. It is generally done by hanging destination address field (IP address) on a message; you could cause that message to be forwarded to you across the internet, instead of its intended recipient. You can get decrypted message from internet. Generally this attack is accomplished with some other attacks like masquerading.

### 3.1.1 Man – in – Middle attack

In this type of attack, an intruder causes a legitimate client to connect to an intruder's AP, and then the intruder connects to the valid enterprise AP. All communication between the legitimate client and the network now flows through the attacker – allowing him to modify data, delete data, or insert data. This attack again depends on a compromised encryption protocol, so solving the surveillance issue also solves this one. Essential to thwarting such attacks is the ability to detect man-in-the-middle attacks that automatically quarantine the attacked user from the network, blocking the MITM attack from being successful.



*Fia. 3.1 Man – in – the – middle* attack

> *"Unmanaged wireless LANs can jeopardize entire enterprise networks, data, and operations."*
>
> ***Forrester Research, Inc.***

A second class of impersonation attack involves an attacker pretending to be an enterprise AP advertising an enterprise SSID. A typical wireless client machine scans for the best AP and associates with it. That AP could be sitting in the parking lot with a 500mW amplifier attached to it. Once a client has associated with an -

-can be carried out, including stealing authentication credentials, worm and virus transmission, or emulation of enterprise services for the purpose of stealing passwords. Protecting against this type of "honeypot" attack includes monitoring usage of the enterprise SSID and disabling any unauthorized APs using it.

## 3.1.2 Denial of service (DoS)

Common type of DoS attack works within the 802.11 protocol framework. These types of attacks require only a laptop or PDA with a wireless NIC - equipment that is inexpensive and readily available. These attacks range from floods of 802.11 associate frames that attempt to consume all available client slots in the AP to 802.1x EAP (extensible authentication protocol) handshake floods that try to overwhelm an authentication server to the ubiquitous deauthenticate (i.e. deauth) flood that causes clients to drop their association with an AP. Deauth attacks are the most effective of 802.11

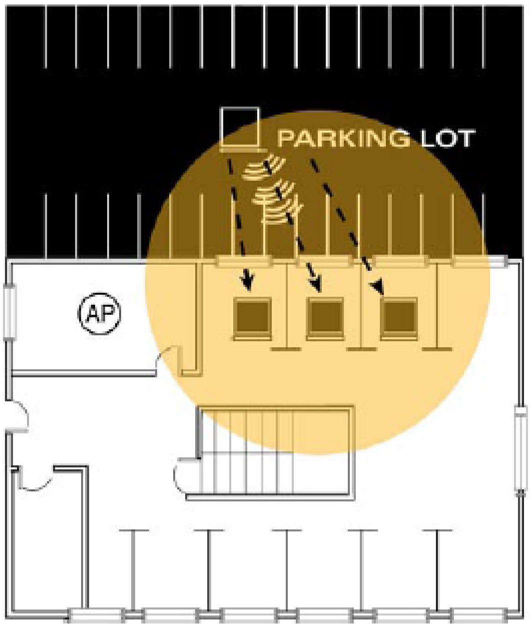


*Fig 3.2 Denial of service attack*

> *“Through year-end 2004, the employee’s ability to install unmanaged access points will result in more than 50% of enterprises exposing sensitive information through wireless networks.”*
>
> **Gartner**

DoS attacks. They exploit a weakness in the 802.11 protocol that forces stations and APs to use the source MAC address as the identifier of another 802.11 device. Frames are not authenticated – meaning that anyone can change the MAC address of their NIC card and send frames that appear to come from another device. Attackers exploit this weakness to send deauthenticate frames to stations that appear to come from the AP – stations respond according to the protocol requirements and drop their association to the AP. If this process is repeated enough times, stations will assume the wireless LAN scanning for a new AP.

## 3. Traditional Security Architecture In Context with Wi-Fi Security

The traditional security approach for network security is to divide the network into two zones: trusted and untrusted. The trusted zone is the area under your physical control, where access is limited by the security by the security guard at the front door. There is no need for network security protection within the trusted zone. By contrast, you regard untrusted zone as full of enemies. Internet access or even dedicated wired links through the public network are untrusted. Where the untrusted network, there is typically a firewall to prevent all the enemies getting in (Figure 4.1). The firewall is the electronic equivalent of the security guard.

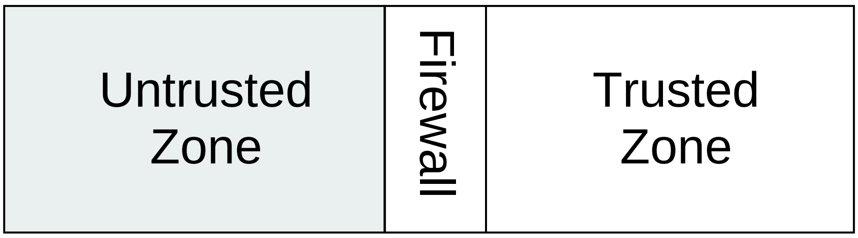


*Fig. 4.1 Conventional Security* Architecture

Difficulty arises when trusted people find themselves in the untrusted zone and want to access their home network. When you are traveling and staying in a hotel, you need a safe, you need a safe way to get back to the trusted zone in your company. VPN extends the trusted zone out into trusted zone out into the untrusted area through a secure tunnel, as shown in Figure 4.2.

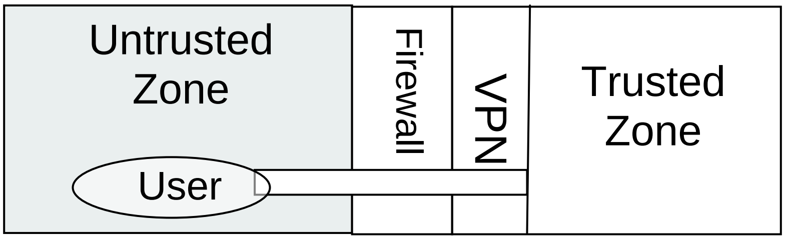


*Fig. 4.2 Remote User in "Trusted Bubble"*

VPN can suffer from deployment issues. VPN software must be installed on the computer of the remote user, and it must be compatible with that of the VPN server. Also, because the implementation on the remote computer is usually in software, VPN may limit the speed with which computer can communicate (although for remote access the speed of the internet is usually the limitation rather than the speed of the software).

## 4.1 How does Wi – Fi LAN fit into this conventional security architecture?

The important question is, “how does Wi – Fi LAN fits into this conventional security architecture?” Should it be deployed like the device in the trusted zone – connected straight to the network? Or should it be placed behind the firewall? In fact, a Wi – Fi LAN can be deployed either way. Placing wireless LANs outside the firewall means they must connect using VPN – a disadvantage. Connect directly to the network inside the firewall means you have to ensure that the wireless LAN is inherently secure.

## 4.2 Option 1: Put WLAN in the Untrusted Zone

In some situations the Wi – Fi LAN user is clearly in an untrusted zone. For example, some airports have installed Wi – Fi wireless LAN coverage in waiting areas or lounges. You can connect to the Wi – Fi LAN and have direct high – speed Internet access while waiting for your flight.

Now think about sitting at your desk in your desk in your office, although you are in a trusted zone, your wireless signal may be going down the corridor and out the window. Therefore, even though you are sitting at your desk, you are operating in an untrusted zone just like airport. In some sense you have created the situation shown in Figure 4.3, wherein the untrusted area extends inside your building.

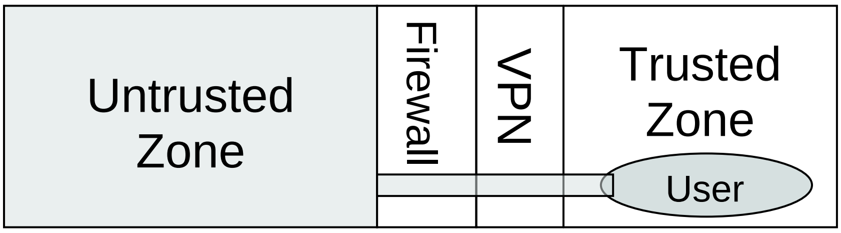


*Fig. 4.3 Wireless User in Untrusted Zone*

One response to this situation is to handle Wi – Fi LAN users in the same way that you handle remote users. Make them use VPN as if they were outside the building. Even though a person is sitting at her desk, all the communications from Wi – Fi LAN computer must be encrypted by VPN software before passing over the network. These communications then go to the wireless access point and on to an Ethernet connection that is outside the firewall. From there they go through the firewall and are decoded by the VPN server. Finally, the messages are placed on the trusted wired network and arrive at their destination, which might be the printer on the desk of a Wi – Fi LAN user. Figure 4.4 shows this design

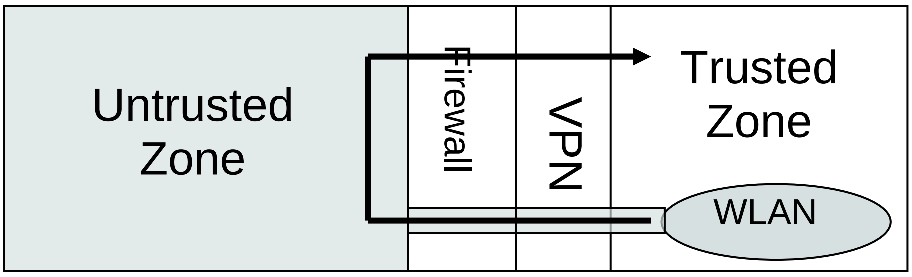


*Fig. 4.4 Treating a Wi-Fi LAN user like a Remote User*

There are several disadvantages with this approach:

- VPN software on the laptop can sometimes be intrusive, slowing down communication and limiting the types of operations that can be performed.
- The access point must be connected to the untrusted side of the firewall. They must have dedicated wiring and cannot share with the internal Ethernet system.

These disadvantages make this approach unattractive for many companies as well as impractical for small office and home users.

## 4.3 Option 2: Make Wi – Fi LAN Trusted

The alternative to treating Wi –Fi LAN as a pariah that can never be trusted is to make the Wi – Fi LAN itself fundamentally impenetrable by enemies. Your goal is to make the LAN so secure that it can be regarded in the same way as physical wiring and treated as part of the trusted zone. Thinking along these lines led to the original security system of IEEE 802.11 being called "WEP", which stands for Wired Equivalent Privacy.

# 4. WEP Security Architecture

In 1997, the IEEE produced the first wireless networking standard, 802.11. “WEP is an algorithm that’s used to protect wireless communications from eavesdropping and modification. A secondary function of WEP is to prevent unauthorized access to a wireless network. It relies on a secret key that is shared between a wireless station and an access point. The secret key is used to encrypt packets before they are transmitted and an integrity check is used to ensure the packets are not modified in transit. The 802.11 standard does not state how the shared key is established.

There are two parts to WEP security described in the standard. The first is the authentication phase and the second is the encryption phase. The idea goes roughly as follows: When a new mobile device wants to join to an access point, it must first prove its identity.

## *5.1 Authentication*

The IEEE 802.11 only ever specified the use of 40 – bit keys. In 1999 WEP specifiaction included 104 – bit key. The IEEE 802.11 (1999) defined two level of security: open and shared key.

### *5.1.1 Open System Authentication*

Open system authentication is the default authentication protocol for 802.11. Open system authentication authenticates anyone who requests authentication. Essentially, it provides a NULL authentication process. The authentication management frames used by this protocol are sent in clear text even when WEP is enabled.

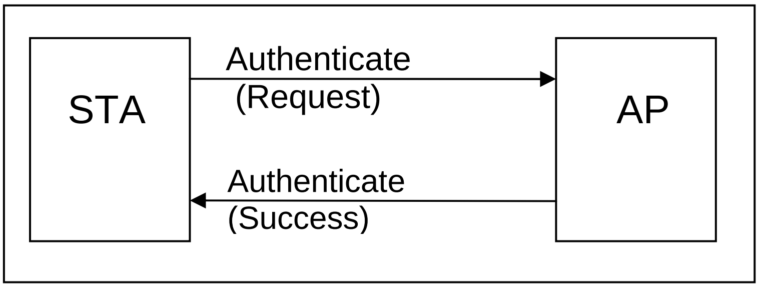


*Fig 5.1 Open Authentication*

### 5.1.2 Shared Key Authentication

WEP authentication is intended to prove to a legitimate access point that the mobile device known the secret key. When the mobile device requests authentication, the access point sends a random number called challenge text. This is an arbitrry 128 – bit number (preferably random). The mobile mobile device then encrypts this number with the secret key using WEP and sends it back to access point. Because the access point remembers the random number previously sent, it can check whether the result back was encrypted with the correct key; the mobile device must know the key.

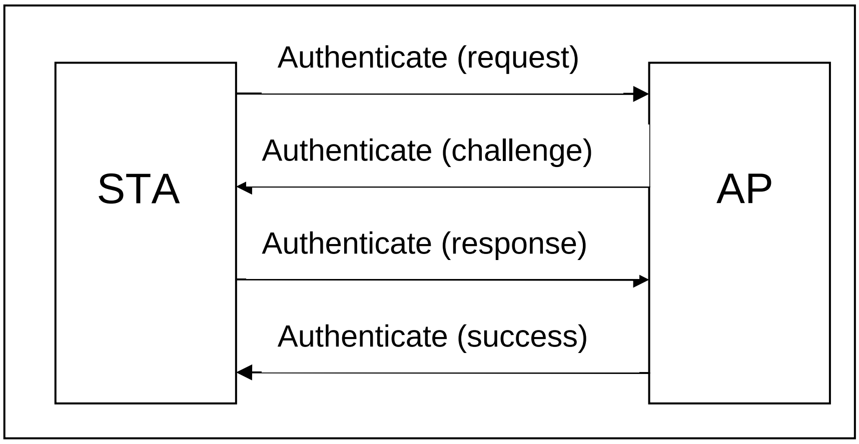


*Figure 5.2 WEP Authentication*

## 5.2 Privacy

It means that preventing stranger from intercepting and understanding your data. For Wi – Fi LAN security, privacy is a very desirable attribute, and was central to WEP's objectives.

WEP uses a stream cipher called RC4 to encrypt the data packets. At the highest level, RC4 is a black box that takes in one byte from a stream at a time and produces a corresponding but different byte for the output stream. One of the advantages of RC4 is that it is fairly easy to implement and does not use any complicated or time-consuming operation like multiplication.

### 5.2.1 Initialization Vector (IV)

As mentioned before, the original key length was 40 bits, which most manufacturers have increased to 104 – bit. Manufacturers often refer to their 104 – bit solution as “128 – bit” security. So what happens to the extra 24 bits? The answer lies in the initialization vector.

There is a problem in using a fixed key value. Although the key may be updated from time to time, it is fixed relative to the flood of data packets running through the system. Effectively all the data packets are encrypted using the same key values.

The solution to this problem is the Initialization vector (IV). This is a very simple concept. Instead of just using the fixed secret key to encrypt the packets, you combine the secret key with a 24 – bit number that changes for every packet sent. This extra number is called the IV and effectively converts the 104 – bit key into a 128 – bit key. But actually, calling this 128 – bit security is a minor contrick because the value of the IV is not secret at all but is transmitted openly with the encrypt frame. To prevent the use of a fixed key for encryption, the actual key used to initialize the RC4 algorithm is the combination of the secret key and the IV, as shown in Figure 5.2.

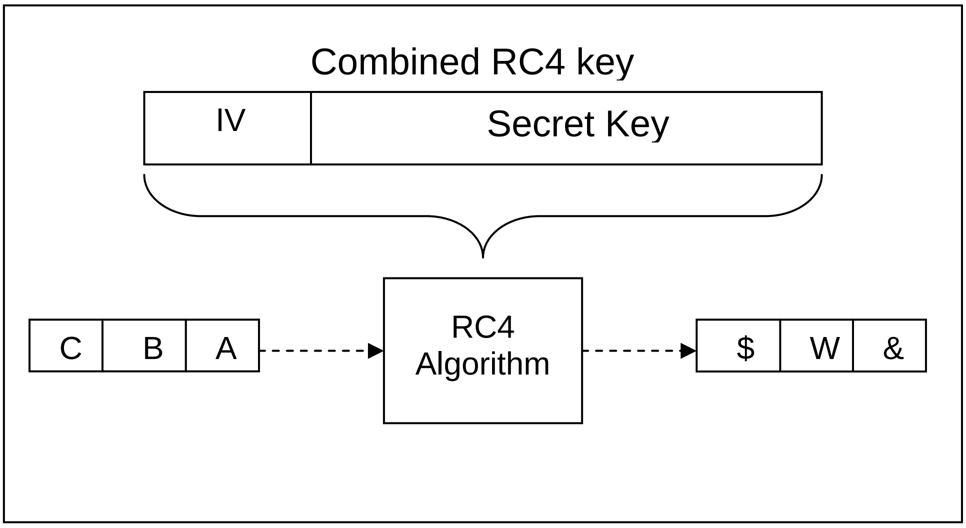


*Figure 5.2 WEP message Encryption using RC 4 streams ciphers*

## 5.2.2 Integrity Check Value (ICV)

The idea behind the ICV is to prevent anyone from tampering with the message in transit. In both encrypted and unencrypted messages, a check is made to detect whether any bits have been corrupted during transmission. All the bytes in the message are combined in a result called the CRC (cyclic redundancy check). This 4 – byte value is added on to the end of the frame immediately prior to processing for transmission. Even if a single bit in the message is corrupt, the receiving device will notice that CRC value does not match and reject the message.

ICV is similar to the CRC except that it is computed and added on before encryption. The conventional CRC is still added after encryption. The theory is that, because the ICV is encrypted, no attacker can recompute it when attempting to modify the message. Therefore, the message is rendered tamper – proof. So ICV is computed by combining all the data bytes to create a four – byte check word. This is then added on the end, as shown in Figure 5.3.

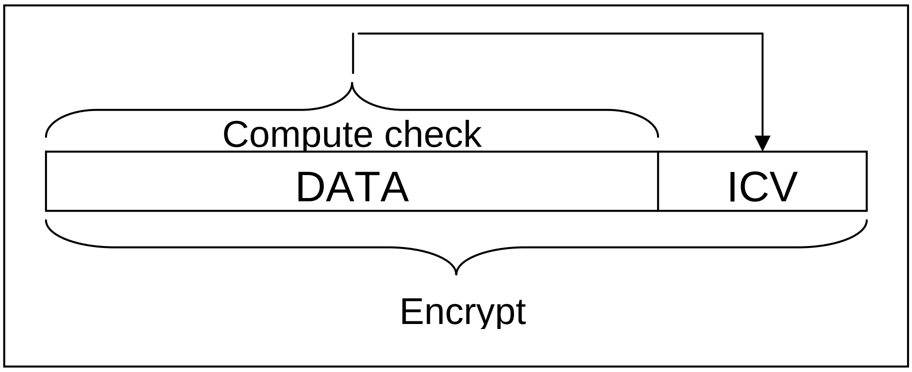


*Figure 5.3 Adding ICV*

### 5.2.3 Prepare Frame for transmission

After the ICV is appended, the frame is ready for encryption. First, the system must select an IV value and append it to the secret WEP key. Next, it initializes heRC4 encryption engine. Finally it passes each byte going in; an encrypted byte comes out until all the bytes are processed. This is a stream cipher.

For the receiver to know how to decrypt the message, the key number and IV value must be put on the front of the message. Four bytes are added for this purpose. The first three bytes contain the 24 – bit IV value and the last byte contains the KeyID number 0, 1, 2 or 3, as shown in Figure 5.4. KeyID is used to describe which key is in use from available set of key values (Details about KeyID and multiple keys can be found in standard draft)

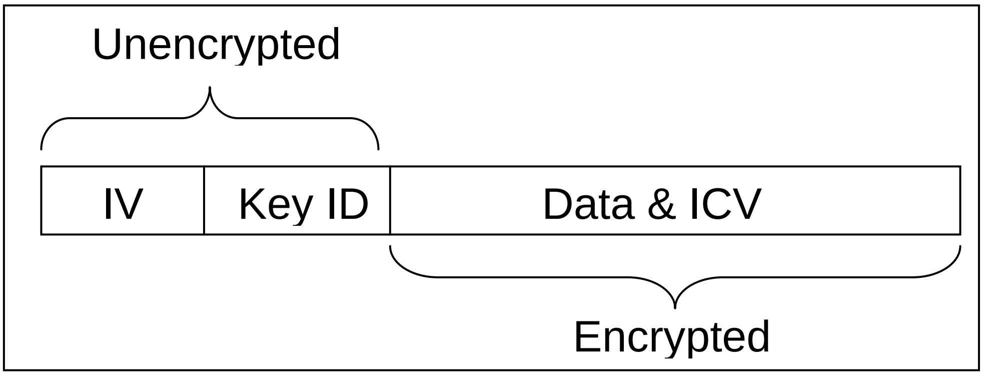


*Figure 5.4 Adding IV and Key ID*

# 5. Why WEP Is Not Secured

WEP was included in the original IEEE 802.11 standard in 1997, but it was not until 1999 that systems were widely deployed when IEEE 802.11b and Wi – Fi became established. The industry started to have concern about WEP security as designs were made and engineers started to point out some problems. In particular, the weakness of the authentication method was noted, and authentication phase was dropped altogether. However, the manufactures' concerns related to the strength of the security rather than the overall integrity. In other words, they were concerned that a serious and major attack might succeed. Nobody thought that it would be easy to break WEP.

In this section we will see in detail at the attacks on WEP that have been identified. WEP is general term that covers a number of security mechanisms. As a basis for evaluating WEP, we will consider following mechanism needed for security:

- Authentication
- Access Control
- Replay prevention
- Message modification detection
- Message privacy

Unfortunately, WEP fails to perform in all these areas. We'll look at each one separately.

## 6.1 Authentication

Authentication is about one party proving to other that he really is who he claims to be. Authentication is not a one time process – in other words, it is not enough to prove once that you are authentic. It is only useful if you can prove it every time you communicate. In the wireless world, you usually need mutual authentication. The network wants proof about the user but the user also wants proof that the network really is the expected one. This is important for Wi – Fi LANs because it is inexpensive to set up decoy access points.

Finally, security experts pointed out that it is essential to use different secret keys for authentication than you use for encryption. In summery, the basic requirements for authentication to wireless LANs are:

1. Robust method of proving identity that can not be spoofed
2. Method of preserving identity over subsequent transactions that cannot be transferred.
3. Mutual Authentication
4. Independent keys from encryption keys

Unfortunately, WEP fails all counts. As already mentioned, the key used for this process is the same WEP key used for encryption, thus breaking rule 4. the operation does not authenticate the access point to the mobile devices because a rogue access point can pretend it was able to check the encrypted string and send a success message without ever knowing the key. Hence rule 3 is broken.

Rule 2 is broken because there is no token provided to validate subsequent transaction, making the whole authentication process rather futile.

Rule 1 is rather irrelevant given the weaknesses already mentioned but it is necessary to understand how this rule fails.

802.11 define two forms of authentication: Open System (no authentication) and Shared Key authentication.

These are used to authenticate the client to the access point. The idea was that authentication would be better than no authentication because the user has to prove knowledge of the shared WEP key, in effect, authenticating him. In fact, the exact opposite is true: if you turn on authentication, you actually reduce the total security of your network and make it easier to guess your WEP key. Shared Key authentication involves demonstrating the knowledge of the shared WEP key by encrypting a challenge.

The problem is that a monitoring attacker can observe both the challenge and the encrypted response. From those, he can determine the RC4 stream used to encrypt the response, and use that stream to encrypt any challenge he receives in the future. So by monitoring a successful authentication, the attacker can later forge an authentication. An advantage of Shared Key authentication is that it reduces the ability of an attacker to create a denial-of-service attack by sending garbage packets into the network. Shared Key authentication also allows a wireless client to quickly determine if they know the correct WEP key, which is a nice “user friendly” configuration---and allows a malicious client to try a dictionary attack on the wireless network.

## 6.2 Access Control

Access control is, rather obviously, the process of allowing or denying a mobile device to communicate with the network. It is often confused with authentication. That entire authentication does is to establish who are; it does not follow that, because you are authenticated, you should be allowed access. In general, access is usually controlled by having a list of allowed devices. It may also be done by allowing access to anyone who can prove he has possession of a certificate or some other electronic pass.

IEEE 802.11 does not define how access control is implemented. However, identification of device is only done by MAC address, so there is an implication that a list of acceptable MAC addresses exists somewhere. However, given the ease with which Mac addresses can be forged, this cannot be considered as a serious security mechanism.

## 6.3 Replay Prevention

Let’s suppose you are an attacker with a wireless sniffer that is able to capture all the frames sent between an access point and a mobile device. You observe that a new user has turned on his/her laptop and connected to network. May be the first thing that happens is that the server sends him/her a login message and he/she enters her user name and password. Of course, you can’t see that actual messages because they are encrypted.

However, you might be able to guess what’s going on, based on the length of the message.

Later on, you notice the user has shut down and gone home. So now is your chance. Bring up your own client using her MAC address and connect to the network. Now, If you are lucky, you will receive a message to log in. Again, you won’t be able to decode it, but you can guess what it is from the size. So now you send a copy of the message the legitimate user sent at that point. You are replaying an old message without needing to know content. You as an attacker have successfully logged into the network and the server. The wireless security protocol should allow only one copy of a message to be accepted. *Ever*.

Replay protection is not broken in the WEP; it simply does not exist.

## 6.4 Message Modification Detection

WEP has a mechanism that is designed to prevent message modification. Message modification can be used in subtle ways. To prevent tampering, WEP includes a check field called the integrity check value (ICV). The WEP ICV is based on CRC-32, an algorithm for detecting noise and common errors in transmission. CRC-32 is an excellent checksum for detecting errors, but an awful choice for a cryptographic hash. Better-designed encryption systems use algorithms such as MD5 or SHA-1 for their ICVs. The CRC-32 ICV is a **linear** function of the message meaning that an attacker can modify an encrypted message and easily fix the ICV so the message appears authentic. Being able to modify encrypted packets provides for a nearly limitless number of very simple attacks. For example, an attacker can easily make the victim’s wireless AP decrypt packets for him. Simply capture an encrypted packet stream, modify the destination address of each packet to be the attacker’s wired IP address, fix up the CRC-32, and retransmit the packets over the air to the AP. The AP will happily decrypt the packets and forward them to the attacker.

## 6.5 Message Privacy

This is really big one: attacking encryption method of WEP. There are two main objectives in attacking the encryption: decode a message or get the keys. The ultimate success is to get the keys. Once attacker has the keys, He is free to explore and look for the valuables.

There are two weaknesses in the way RC4 is used in WEP and we will look at each case separately:

- IV reuse
- RC4 weak keys

### 6.5.1 IV Reuse

WEP's IV size of 24 bits provides for 16,777,216 different RC4 cipher streams for a given WEP key, for any key size. Remember that the RC4 cipher stream is XOR-ed with the original packet to give the encrypted packet which is transmitted, and the IV is sent in the clear with each packet. The problem is IV reuse. If the RC4 cipher stream for a given IV is found, an attacker can decrypt subsequent packets that were encrypted with the same IV, or, can forge packets. This means that you don't need to know the WEP key to decrypt packets if you know what the key stream was used to encrypt that packet. They sound like similar problems, but it's actually much easier to discover the key stream than it is to discover the WEP key.

Since there are only 16 million IV values, how the IV is chosen makes a big difference in the attacks based on IV. Unfortunately, WEP doesn't specify how the IV is chosen or how often the IV is changed. Some implementations start the IV at zero and increase it incrementally for each packet, rolling over back to zero after 16 million packets have been sent. Some implementations choose IVs randomly. That sounds like a good idea, but it really isn't. With a randomly chosen IV, there is a 50% chance of reuse after less than 5000 packets.

Additionally, there are many methods for discovering the cipher stream for a particular IV. For example, given two encrypted packets with the same IV, the XOR of the original packets can be found by XORing the encrypted packets. If the victim is on the Internet, the attacker can simply ping the victim or send an email message. If the attacker is able to send the victim packets and observe and analyze those encrypted packets, he can deduce the cipher stream.

### 6.5.2 RC4 Weak Keys

RC4 in its implementation in WEP has been found to have weak keys. Having a weak key means that there is more correlation between the key and the output than there should be for good security. Determining which packets were encrypted with weak keys is easy because the first three bytes of the key are taken from the IV that is sent unencrypted in each packet. This weakness can be exploited by a passive attack. All the attacker needs to do is be within a hundred feet or so of the AP.

Out of the 16 million IV values available, about 9000 are interesting to the most popular attack tool, meaning they indicate the presence of weak keys. The attacker captures “interesting packets”, filtering for IVs that suggest weak keys. After that attacker gathers enough interesting packets, he analyzes them and only has to try a small number of keys to gain access to the network. Because all of the original IP packets start with a known value, it’s easy to know when you have the right key. To determine a 104 bit WEP key, you have to capture between 2000 and 4000 interesting packets. On a fairly busy network that generates one million packets per day; a few hundred interesting packets might be captured. That would mean that a week or two of capturing would be required to determine the key.

# 6. WPA: An Intermediate Solution

The Wi-Fi Protected Access (WPA) is a standards-based interoperable security specification. The specification is designed so that only software or firmware upgrades are necessary for the existing or legacy hardware to meet the requirements. Its purpose is to increase the level of security for existing and future wireless LANs. WPA is based on a subset of soon-to-be-finished 802.11i standard, including the following key features to address WEP vulnerabilities:

- Implements 802.1X EAP based authentication to enforce mutual authentication
- Apply Temporal Key Integrity Protocol (TKIP) on existing RC4 WEP to impose strong data encryption
- Use Michael Message Integrity Check for message integrity

WPA is an interim security solution that targets on all known WEP vulnerabilities. It will be forward compatible with the upcoming 802.11i standard. The ultimate wireless security solution is still 802.11i. All products are supposed to comply with 802.11i standard once the standard is released.

## *7.1 802.1x EAP based Authentication*

WPA adopts 802.1X to address the issue of user authentication in WEP. 802.1X initially is designed for wired networks but is also applicable to wireless networks. The standard provides port-based access control and mutual authentication between clients and access points via an authentication server.

802.1X standard are comprised of three elements.

- A supplicant – A user or a client wants to be authenticated. It can be the client software on a laptop, PDA or other wireless device.
- An authentication server – An authentication system, such as a RADIUS server, handles actual authentications.
- An authenticator – A device acts as an intermediary between a supplicant and an authentication server. Usually, the device is an access point.

The mutual authentication in 802.1X involves several steps:

1. A supplicant initiates a connection with an authenticator. The authenticator detects the initiation and enables the port of the supplicant. However, all the traffic except 802.11X ones, including DHCP, HTTP, FTP, SMTP and POP3, are blocked.
2. The authenticator then requests the identity from the supplicant.
3. The supplicant then responds with the identity. The authenticator passes the identity to an authentication server.
4. The authenticator server authenticates the identity of the supplicant. Once authenticated, an ACCEPT message is sent to the authenticator. The authenticator then transitions the supplicant's port to an authorized state.
5. The supplicant then requests the identity from the authentication server. The authentication server passes its identity to the supplicant
6. Once suppliant authenticates the identity of authentication server, all the traffics are forwarded thereafter.

The exact method of supplying identity is defined in the Extensible Authentication Protocol (EAP). EAP is the protocol that 802.1X uses to manage mutual authentication. The protocol provides a generalized framework for a wireless network system to choose a specific authentication method to authenticate. The authentication method can be passwords, PKI certificates or other authentication tokens. With a standardized EAP, an authenticator does not need to understand the details about authentication methods. The authenticator simply acts as a middleman to package and repackage EAP packets to pass from a supplicant to an authentication server, in which here an actual authentication will take place.

There is a special case in 802.1X implementation. In the small user's environment such as home or small business, an authentication server may not be available for authentication. As such, a pre-shared key mechanism is used. The shared key is placed to a supplicant and an authenticator manually. A similar WEP-like authentication is operated.

## 7.2 TKIP

Temporal Key Integrity Protocol (TKIP) is another element derived from 802.11i9. It is aimed to address WEP's known vulnerabilities in the area of data encryption. Specifically, TKIP fixes the security flaw of key reuse in WEP.

TKIP packet is comprised of three parts: 1. A 128-bit temporal key that is shared by both clients and access points. 2. An MAC address of a client device. 3. A 48-bit initialization vector describes a packet sequence number. This combination guarantees various wireless clients use different keys. In order to be compatible with existing hardware, TKIP uses the same encryption algorithm (RC4) as WEP. As such, only software or firmware upgrade is required to implement TKIP. Compared with WEP, TKIP changes the temporal keys every 10000 packets. This dynamic distribution leaves potential hackers little room to crack TKIP key.

In next section we will see in detail, how exactly TKIP helps in solving problems with WEP.

## 7.3 Michael Message Integrity Check

Michael Message Integrity Check is used to enforce data integrity. A Message Integrity Code (MIC) is a 64-bit message calculated using “Michael” algorithm10. Its aim is to detect potential packet content altercation due to transmission error or deliberate manipulation. The MIC is inserted in a TKIP packet.

# 7. TKIP

TKIP is TGi’s response to the need to do something—anything—to improve security for already deployed 802.11 equipments. TGi has proposed TKIP as a mandatory-to-implement security enhancement for 802.11, and patches implementing it will likely be available for most equipment in late 2002. TKIP is a suite of algorithms wrapping WEP, to achieve the best security that can be obtained given the problem design constraints. The TKIP algorithms are designed explicitly for implementation on legacy hardware, hopefully without unduly disrupting performance. TKIP adds four new algorithms to WEP:

- A cryptographic *message integrity code*, or *MIC*, called Michael, to defeat forgeries
- A new *IV sequencing discipline*, to remove replay attacks from the attacker’s arsenal
- A per-packet *key mixing function*, to de-correlate the public IVs from weak keys
- A *rekeying* mechanism, to provide fresh encryption and integrity keys, undoing the threat of attacks stemming from key reuse.

The remainder of this section analyzes each of the TKIP components, and the next section indicates how they are intended to work together to rescue WEP. TKIP is an acronym for “Temporal Key Integrity Protocol.” The name is something of a misnomer. The TKIP rekeying mechanism updates what are called temporal keys, which are consumed by the WEP encryption engine and by the Michael integrity function.

## *8.1 Defeating forgeries: MICHAEL*

A MIC is cryptographic device to detect forgeries. The literature calls these *message authentication code*s, or *MAC*s. Since IEEE 802 had already appropriated the acronym MAC to mean “media access control,” TGi uses the MIC instead. Classical MICs include the CBC-MAC, constructed from a block cipher and used widely in banking applications, and HMAC, used by Ipsec (Internet Protocol Security).

Every MIC has three components: a secret authentication key *K* (shared only between the sender and receiver), a tagging function, and a verification predicate. The tagging function takes the key *K* and a message *M* as its inputs, and generates a tag *T*, also called the message integrity code, as its output. A protocol protects a message *M* from forgery by having the sender compute the tag *T* and send it with the message *M*. To check for a forgery, the receiver inputs *K*, *M* and *T* into the verification predicate. The predicate evaluates to TRUE if the reputed tag *T* is what should have been produced by the tagging algorithm, and FALSE otherwise. If verification indicates FALSE, the message is a failed forgery. If the verification function returns TRUE, the message is presumed authentic. A MIC is considered secure if it is infeasible for an attacker to select the correct tag for some new, never-seen-before message *M*, without knowing the key *K*.

Michael [2] is the name of the TKIP message integrity code. It is an entirely new MIC designed by Niels Ferguson.

The Michael key is 64-bits, represented as two 32-bit little-Endian words ($K_0$, $K_1$).

The Michael tagging function first pads a message with the hex value 0x5a and enough zero pad to bring the total message length to a multiple of 32-bits, then partitions the result into a sequence of 32-bit words $M_1\ M_2 \ldots M_n$, and finally computes the tag from the key and the message words using a simple iterative structure:

```
$(L, R) \leftarrow (K_0, K_1)$
**do** $i$ **from** 1 **to** $n$
    $L \leftarrow L \ \square\ M_{\mathrm{i}}$
    $(L, R) \leftarrow b\ (L, R)$
**return** $(L, R)$ as the tag
```

Where □ denotes exclusive or (XOR) and *b* is a simple function built up from rotates, little-Endian additions, and bit swaps.

The Michael verification predicate reruns the tagging function over the message and returns the result of a bit-wise compare of this locally computed tag and the tag received with the message.

Security is the other lead issue with Michael. The security level of a MIC is usually measured in bits. If the security level of a MIC is *s* bits, then, by definition, the time required for an attacker to construct a forgery is, on average, after about 2–*s*+1 packet.

On an 802.11b WLAN, an attacker could theoretically sustain a rate of 212 small packets per second. This means that an adversary could expect to create a forgery against Michael after about 27 seconds worth of messages (assuming a 20-bit security level), or somewhat over two minutes. The corresponding numbers for 802.11a are 213 packets and 26 seconds, respectively. This level of protection is much too weak to afford much benefit by itself, so TKIP complements Michael with counter-measures. The design goal of the counter-measures is to throttle the utility of forgery attempts, limiting the knowledge the attacker gains about the MIC key. If a TKIP implementation detects two failed forgeries in a second, the design assumes it is under active attack. In this case, the station deletes its keys, disassociates, waits for a minute, and then re - associates. While this disrupts communications, it is necessary to thwart active attack. The countermeasures algorithm thus limits the expected number of undetected forgeries such an active adversary might generate to about one per year per station.

## 8.2 Defeating replays: IV sequence enforcement

The standard way to address this problem is to associate a packet sequence number space with a MIC key, and to reinitialize the sequence space whenever the MIC key is replaced. This strategy requires that the transmitter refrain from sending data protected by the old MIC key once it exhausts the sequence number space. The choices available to the transmitter on exhaustion are (a) halt communications altogether, (b) try to rekey the MIC with a fresh key, or (c) send subsequent traffic without any cryptographic protections. Failing to adopt one of these strategies risks exposing the data already protected under the key.

TKIP closely follows this classical design. To defeat replays, TKIP reuses the WEP IV field as a packet sequence number. Both transmitter and receiver initialize the packet sequence space to zero whenever new TKIP keys are set, and the transmitter increments the sequence number with each packet it sends. TKIP requires the receiver to enforce proper IV sequencing of arriving packets. TKIP defines a packet as out-of-sequence if its IV is the same or smaller than a previous correctly received MPDU associated with the same encryption key. If an MPDU arrives out of order, then it is considered to be a replay, and the receiver discards it and increments a replay counter.

TKIP deviates somewhat from the usual design by associating the sequence number with the TKIP encryption key, not with its MIC key. This constraint derives from the choice of the WEP IV field as the sequence space. This was done in order to reuse the existing WEP hardware and hence packet formats. The 1999 IEEE 802.11 standard associates the IV field with packet fragments, or MPDUs, whereas legacy access points will of necessity calculate the TKIP MIC over the entire packet, or MSDU. We will explain why this works below, in the section called “Putting the pieces together.”

TKIP replay detection has at least one serious limitation. It will not work with the quality of service (QoS) enhancements being proposed by IEEE 802.11 Task Group e. It will be necessary to alter at least the enforcement step at the receiver to accommodate QoS, and perhaps the IV selection algorithm as well. The extension to properly detect replays in the presence of QoS is not yet defined.

## 8.3 Defeating weak key attacks: key mixing

Unlike a MIC, replay detection, or rekeying, all of which are mandatory for any protocol claiming to provide privacy, TKIP’s per-packet key construction is a feature uniquely necessary to correct WEP’s misuse of RC4. Recall that WEP constructs a per-packet RC4 key by concatenating a base key and the packet IV. The new per-packet key construction, called the TKIP key mixing function, substitutes a *temporal key* for the WEP base key and constructs the WEP per-packet key in a novel fashion. Temporal keys are so named because they have a fixed lifetime and are replaced frequently.

Doug Whiting and Ron Rivest invented the TKIP key mixing function. It transforms a temporal key and packet sequence counter into a per-packet key and IV. The mixing function operates in two phases, with each phase compensating for a particular WEP design flaw. Phase 1 eliminates the same key from use by all links, while Phase 2 de-correlates the public IV from known the per-packet key.

Phase 1 combines the 802 MAC addresses of the local wireless interface and the temporal key by iteratively XORing each of their bytes to index into an S-box[1], to produce an intermediate key. Stirring the local MAC address into the temporal key in this way causes different stations and access points to generate different intermediate keys, even if they begin from the same temporal key—a situation common in ad hoc deployments. This construction forces the stream of generated per-packet encryption keys to differ at every station, satisfying the first design goal. The Phase 1 intermediate key must be computed only when the temporal key is updated, so most implementations cache its value as a performance optimization.

Phase 2 uses a tiny cipher to “encrypt” the packet sequence number under the intermediate key, producing a 128-bit per-packet key. In actuality, the first 3 bytes of Phase 2 output corresponds exactly to the WEP IV, and the last 13 to the WEP base key, as existing WEP hardware expects to concatenate a base key to an IV to form the per-packet key. This design accomplishes the second mixing function design goal, by making it difficult for an adversary to correlate IVs and per-packet keys.

The tiny cipher defined for Phase 2 has a Feistel structure, which means its inner loop implements a transformation of the form (L, R) → (R, L □ f(R)). As with Michael, the tiny cipher’s inner loop can be implemented using only simple operations: XORs, shifts, rotates, and table look-ups - all cheap operations for processors commonly in 802.11 devices. Phase 2 represents the packet sequence number as a 16-bit little-Endian counter. Phase 2 assigns the 8 most significant bits of the counter to the first and second bytes of the WEP IV, and the least significant counter bits to the third IV byte. It then masks off the most significant bit of the second IV byte to prevent the WEP per-packet key concatenation from producing one of the known RC4 weak keys.

## 8.4 Defeating key collision attacks: rekeying

The definition of the last TKIP element, rekeying, is not yet complete. Rekeying delivers the fresh keys consumed by the various TKIP algorithms. However, the general outline of this scheme can be described.

The TGi rekey architecture will depend upon a hierarchy of at least three key types: temporal keys, key encryption keys, and master keys.
Occupying the lowest level of the hierarchy are the temporal keys consumed by the TKIP privacy and authentication algorithms proper. TKIP employs a pair of temporal key types: a 128-bit encryption key, and a second 64-bit key for data integrity. TKIP uses a separate pair of temporal keys in each direction of an association. Hence, each association has two pairs of keys, for a total of four temporal keys. TKIP identifies this set of keys by a two-bit identifier called a *WEP KeyID*.

Previously we observed that WEP IVs can never be reused with the same key without voiding the RC4 privacy guarantees, and that the TKIP key mixing function can construct at most $2^{16}$ IVs. This implies that TKIP requires a key-update mechanism operating at least every $2^{16}$ packets. This is accomplished using the mechanism described below. To affect rollover from one set of keys to another, each association allocates two WEP KeyIDs. When an association is first established, a first set of temporal keys is bound to one of these two WEP KeyIDs. As new keys are created; the association ping-pong between the two KeyIDs, with the new set of temporal keys being bound to the least-recently bound KeyIDs. After binding the new set of temporal keys, TKIP implementations still receive packets on the old key id and its temporal keys, but subsequently transmit only under the new KeyID and its keys. The requirements for new temporal keys are that they are fresh—that is, it is unlikely that any have been used, even with a prior session with the same or another peer, even across reboots—and there is no algorithmic relationships among the keys in the set.

The instantiation of new temporal keys requires careful coordination. TGi accomplishes these using special rekey key messages. The rekey key message distributes keying material from which both the station and AP derive the next set of temporal keys.

This exchange must itself be secure, or an attacker can compromise the temporal keys by compromising the rekey key message. The next level of the hierarchy, the key encryption keys, protects the temporal keys. There are two key encryption keys: one to encryption the distributed keying material, and a second to protect the rekey messages from forgery. The requirements for key encryption keys are similar to those for temporal keys: the station and access point must establish a fresh set on association or reassociation.

# 8. Transition To 802.11i: The Ultimate solution

In order to address WEP security issues, the 802.11 working group adopted the 802.1X standard for authentication, authorization and key management. At the same time, IEEE formed a Task Group "I" to develop 802.11i standard, with a purpose to produce a detailed specification to enhance the security features for wireless LANs dramatically. On 24 June 2004, the Institute of Electrical and Electronic Engineers (IEEE) approved the 802.11i security standard for wireless local-area networks (WLANs).

The 802.11i standard is basically a wrapper around 802.11. It has three components which are spread into two layers. The lowest layer consists of two improved encryption algorithms; temporal key integrity protocol (TKIP) and the counter mode with CBC-MAC protocol (CCMP). On top of TKIP and CCMP sits 802.1x. Not actually a part of 802.11i, but another IEEE standard which provides port based access control and encryption key distribution.

## 9.1 CCMP

Along with the TKIP algorithm, the 802.11i standard defines another encryption method. Current hardware will most likely not be able to use this technique because of the additional overhead it places on the processor in most APs and STAs. This additional overhead is caused by the use of the advanced encryption standard (AES) as its cipher.

AES can be run in a variety of modes. CCMP uses counter mode with CBCMAC (CCM). Encryption is provided by counter mode; authentication and packet integrity are provided for by CBC-MAC. AES is a symmetric block cipher which allows keys of various sizes. 802.11i calls for a key length of 128-bits. This particular mode of AES is designed for packet encryption only and has not been tested outside of that environment.

Other parts of the CCMP method include its own version of a MIC (not Michael) and a 48-bit IV called a packet number (PN). The CCMP algorithm is a required component of a valid 802.11i implementation.

AES is the latest encryption standard approved by the US Government for official use. It has no known weaknesses and withstood an extensive examination by well versed cryptographers. In addition to that, it beat out several other very fine algorithms in a head-to-head competition in order to become the official US encryption standard.

## 9.2 RSN

Robust Security Network (RSN) is the term applied to the strongest security model that 802.11i uses to authenticate, authorize, and protect the connection between the STA and AP. This is the combination of the most robust parts of the 802.11i standard: 802.1x for authentication and authorization, EAP for authentication transport, and support for stronger encryption algorithms such as AES. When there is a connection made between an AP and STA that association is referred to as a Robust Security Network Association (RSNA).

802.11i has all the advantages provided by WPA as mentioned above.
In additions, the 802.11i offers

- Stronger Encryption through the implementation of AES
- Roaming Support

The only issues of the 802.11i are:

- An extra requirement in hardware upgrade is required, in order to implement AES.

# 9. Conclusion

Wireless security has undergone major evolutions in last 7 years. WEP, the original security standard, is widely considered as broken. The IEEE 802.11 Group, the Wi-Fi Alliance and major network equipment vendors like Cisco are all working together to develop a new level of security standards.

WPA, an interim solution to the WEP vulnerability, is released in 2003. WPA, which uses a subset of 802.11i features, is generally believed as a major security improvement in wireless environment. WPA supports existing wireless infrastructure. Vendors can transit to the WPA standard through a software or firmware upgrade.

802.11i, the final solution to wireless security, is expected to provide the robust security required for wireless environment in the future.

# 10. References

## 11.1 Books

Real 802.11 Securities; WI – Fi Protected Access and 802.11i:

*By Jon Edney, William Arbaugh*

## 11.2 Internet sites

http://www.iss.net/wireless

http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html

http://www.cs.rice.edu/~astubble/wep/wep_attack.html

http://www.cs.umd.edu/~waa/1x.pdf

http://sourceforge.net/projects/wepcrack

http://wireless.newsfactor.com

http://security.itworld.com

http://standards.ieee.org/getieee802/download/802.1X-2001.pdf

http://www.sans.org/rr/papers/6/123.pdf

http://www.hackfaq.org/wireless-networks/802.11i.shtml

http://www.nwfusion.com/details/715.html

http://www.ieee802.org/1/files/public/docs2000/P8021XOverview.PDF

http://grouper.ieee.org/groups/802/11/

http://security.itworld.com/

## 11. Acknowledgement

I owe a special debt of gratitude to my guide **Prof. G. K. Kharate** for his inspiring supervision, critical assessment and valuable suggestions during the preparation of this seminar.

I feel obliged to **Prof. J. K. Bharadwaj** for his valuable help and support. I unhesitatingly acknowledge that this work would not have been completed without the help, inspiration and guidance of colleagues and teachers.

# Report Documentation & Accounting Page

| | |
|---|---|
| Report Code:<br>**IT – BE - Seminar 2004 -2005** | **Report Number: 17** |

**Address (Details):**
**Computer Department, K. K. Wagh Institute of Engineering Education & Research,**
**Hirabai Haridas Vidyanagari, Amrutdham, Nashik**
**Pin – 422 003, M.S. INDIA.**
**kkwcoe_nsk@sancharnet.in**

Report Title: "Wi – Fi (802.11) Security"

| | |
|---|---|
| **Author [with Address, phone, E-mail]:**<br>**Address:**<br>**Ph No:** 0-9890543141<br>**E-mail:** jigar_a_shah83@yahoo.com | **Author Details (Year, Branch, Roll):**<br>**Year:** 2004 – 2005.<br>**Branch:** Information Technology<br>**Roll:** 17 |

| | | | | |
|---|---|---|---|---|
| **Type of Report:**<br>FINAL | Time Covered (From – To)<br>**12-July-2004**<br>TO<br>**28-Sept-2004** | Date of Seminar<br>(DD-MM-YYYY)<br>**29-Sept–2004** | Date Of Report<br>(DD-MM-YYYY)<br>**-     -2004** | **Page Count**<br>37 |

***Key Words***: WPA, WEP, RSN, Wi – Fi

| | | | |
|---|---|---|---|
| **REPORT CHECKED BY:**<br>PROF. J. K. BHARADWAJ | Report Checked Date: | **Guides Complete Name:**<br>Prof. G. K. Kharate | Total Copies<br>2 |

**Abstract:**
The main objective of this seminar is to understand basic concepts and evolution of Wi – Fi Security. This report covers topics like problems with convention security architecture, evolution of new standards in Wi – Fi Security protocols and its pros and cons.