

LATEST TRENDS IN WIRELESS SECURITY

Written by: Shon Harris

Many security flaws within the wireless LAN (WLAN) 802.11's encryption protocol, Wired Equivalent Privacy (WEP), and the standard's weak authentication framework caused the IEEE to develop the 802.11i Task Group. Its main goal is defining a new standard for securing WLAN infrastructures and communications. The new standard outlines an authentication framework for WLANs, using 802.1X, Extensible Authentication Protocol (EAP), Message Integrity Code (MIC), and a way of dynamically creating encryption keys to be used on a per-packet basis with the use of the Temporal Key Integrity Protocol (TKIP).

WHAT IS WRONG WITH WHAT WE HAVE NOW?

The now well documented deficiencies within the 802.11 standard allow for attackers to easily sniff wireless traffic, modify packets without the receiving end being alerted, rogue access points can be erected to capture users' credentials and data, and the encryption process used to hide confidential wireless traffic can be broken very easily with free downloadable programs from the Internet. In most cases a WLAN implementation is similar to hanging a live network cable out your window and allowing anyone to connect and authenticate to your network.

Various vendors have come up with their own solutions to add more protection in different WLAN products, but they are only wrapping band-aids around the crux of the problem, which is a poorly developed and implemented standard.

The new wireless standard 802.11i outlines two improved cryptographic approaches that can be used, TKIP and Advanced Encryption Standard (AES). TKIP is backwards compatible with WEP so that the thousands of WLANs that are currently deployed worldwide can be secured with a firmware or software update instead of having to buy new equipment. TKIP feeds keying material into the current algorithm used by WEP, which is RC4, as illustrated in Figure 1. This allows for dynamic keys to be created and used for encryption and decryption purposes. The 802.11i can also use a stronger encryption algorithm (AES), instead of RC4, which shuts the door on backwards compatibility, but provides more protection and is a good choice for companies who have not yet deployed wireless devices in their environment.

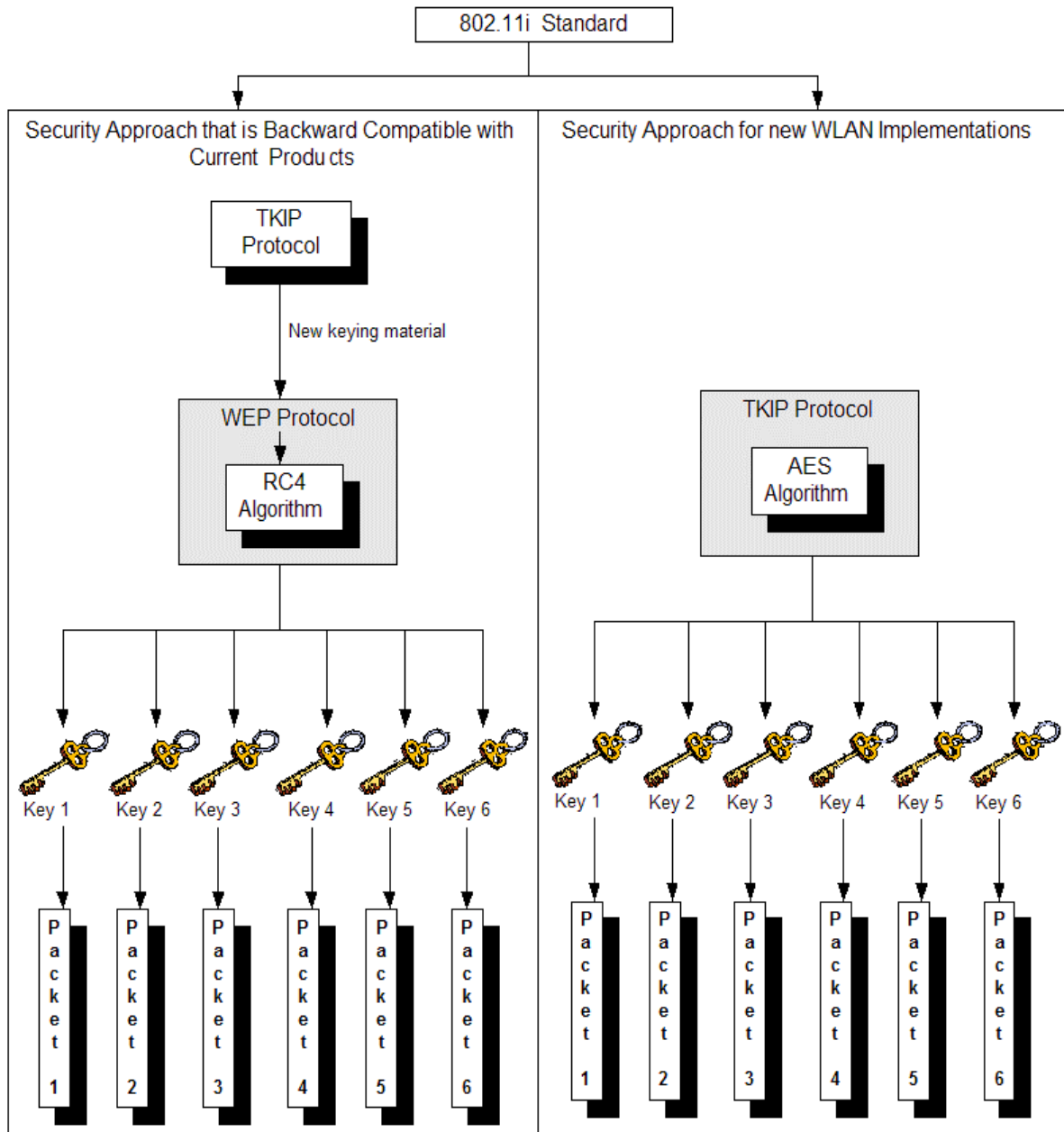


FIGURE 1

The protocols, algorithms, and techniques involved with improving the current 802.11 standard's security are complex and very interdependent. We will look at each of these components and see how they work together with the goal of raising the protection level and allowing us to implement and use wireless technology with more confidence.

802.1X

The 802.1X standard is a port based network access control that provides a framework for user authentication and dynamic encryption key distribution. There are three entities that are involved with this approach to authentication:

1. The supplicant
2. The authenticator
3. The authentication server

The supplicant software resides on the wireless device, the authenticator is the access point (AP), and the authentication server is most likely a RADIUS server. (The AP can be both authenticator and authentication server if an actual authentication server is not available.)

The first goal of 802.1X is to require a successful authentication of a user before a full network connection can be established. When the wireless device initiates a connection with the AP, the AP creates a logical port that works in an unauthorized state until the user has been properly authenticated. The unauthorized state means that no traffic (HTTP, SMTP, FTP, DHCP, etc.), other than authentication frames is allowed to pass. An analogy is having a chain on your front door, which will allow you to identify a person who knocks before you allow him to enter your house.

The AP acts basically as a middleman between the wireless device and the authentication server; it just passes information between the two entities. The AP will only allow the wireless device to communicate with the authentication server until all of the authentication steps are completed successfully. After this the wireless device can then participate with the full network and its resources.

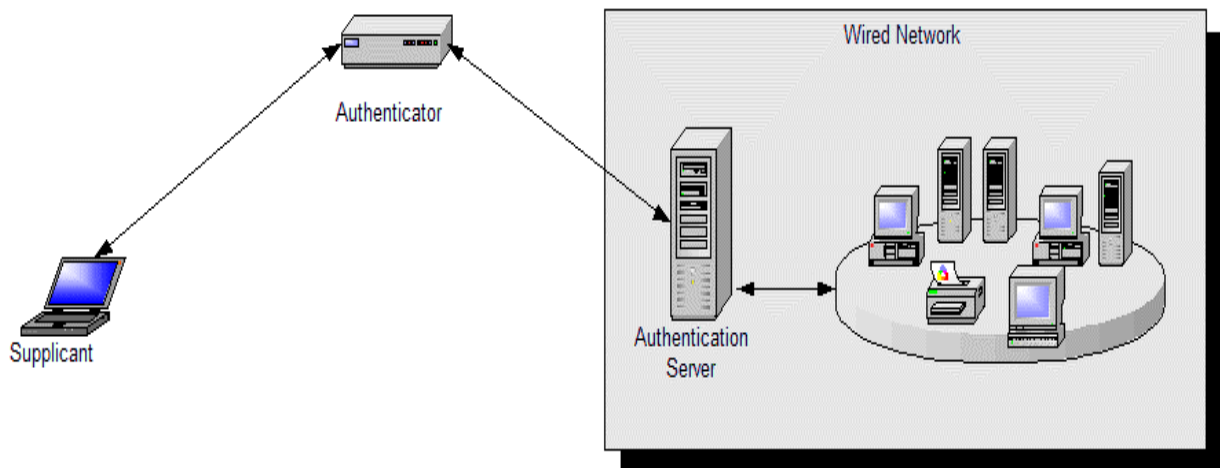


FIGURE 2

One weakness in WEP is that it only provides a way for the wireless device to authenticate to the AP and mutual authentication is not possible. This means a wireless device could send its credentials and data to a rogue AP set up by an attacker and the user would never know that she is communicating with a bogus and potentially dangerous AP since there is no way of requiring the AP to authenticate to the wireless device. The new wireless standard uses the Extensible Authentication Protocol (EAP), which expands the possible ways of authenticating a user to a WLAN and allows for mutual authentication between the user and an authentication server.

EAP allows for different types of authentication mechanisms to be used for verifying user and the server identities; passwords, one-time passwords, certificates, smart cards, token, etc. The wireless device and authentication server can have different authentication modules that plug-in to 802.1X and the actual module is agreed upon during the initial handshaking process. This allows for more flexibility when integrating a WLAN to an existing wired environment and it allows for a more robust authentication process since customers have more choices compared to the currently used username and password. Customers will have different combinations of these authentication methods available to them through different vendors, which they can choose to best fit their needs and environment. The user and authentication server can authenticate to each other with passwords (Cisco’s Lightweight Extensible Authentication Protocol, or LEAP), the two entities can exchange digital certificates (EAP-TLS), or the user can send a password and the server sends its certificate (Protected EAP). These different options are shown in Figure 3.

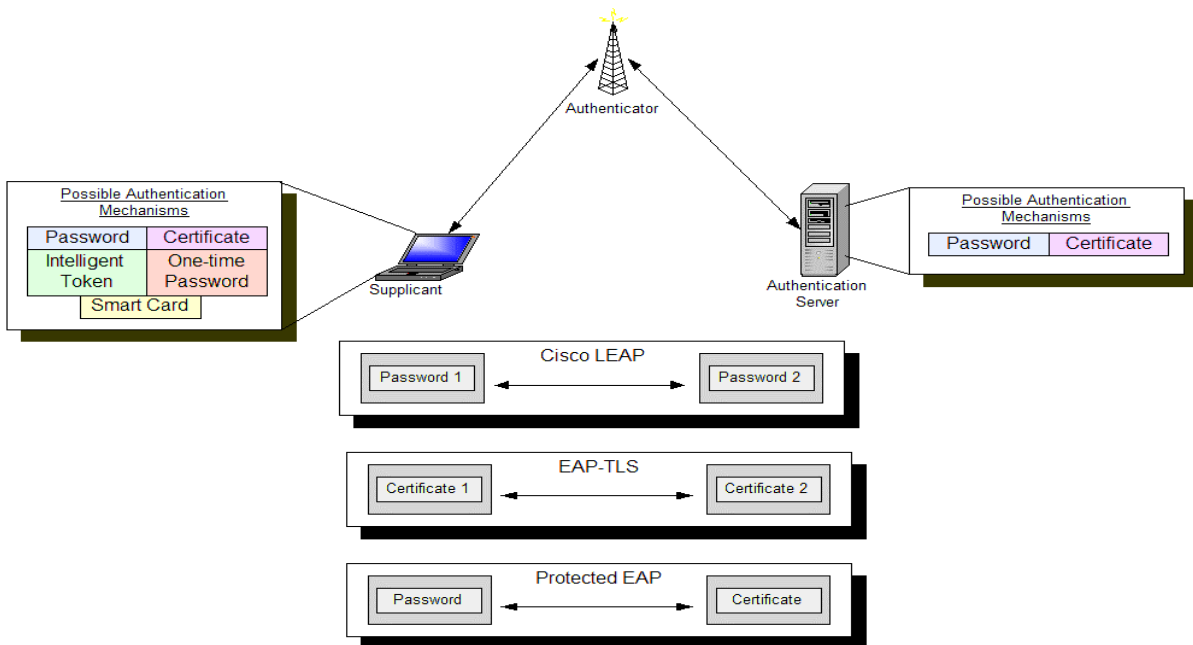


FIGURE 3

In EAP-TLS and Protected EAP (PEAP) the server authentication steps are basically the same as a web server's SSL authentication to a web browser, which is used today for securing Internet transactions. The server sends its digital certificate to the wireless device to be validated. Once the certificate is verified by the wireless device's software, the server's public key is extracted and used to encrypt a master key, which is sent to the authentication server. The wireless device and the server utilize this master key to generate new symmetric session keys, which are used to create a secure channel, just like in SSL connections. Both ends of the connection use these session keys for encryption and decryption purposes.

If PEAP is used, once the server has been properly authenticated and session keys are established at both ends of the connection, the user can encrypt her credential (password, one-time password, or smart card data) and send it to the authentication server protected. If EAP-TLS is used, the user will send a digital certificate to the server for authentication.

The reason a company would choose to use PEAP instead of EAP-TLS is because it does not require the company to install a digital certificate on each and every wireless device. In both methods a current or new Public Key Infrastructure (PKI) needs to be in place for proper authentication to be carried out. This can be an overwhelming task for companies that do not have this type of infrastructure in place and they may not want to go through this amount of work and funds just to secure wireless traffic.

Instead of requiring a PKI to secure wireless traffic, Cisco developed an alternate approach called LEAP, which uses a password-based algorithm. In this approach no PKI components or certificates are used, instead the server and user authenticate to each other by using pre-defined passwords.

Another strength in using EAP is that it allows the user to authenticate to the WLAN. In WEP the wireless device authenticated to the WLAN by proving that it had a static symmetric WEP key. In this scenario when a wireless device wants to authenticate to an AP it is sent a random value, which it encrypts with its symmetric key. The encrypted value is sent back to the AP and if it can properly decrypt and extract the original value, the device is authenticated, not the user. This means that if a wireless device is stolen it could be easily authenticated to the network and allows an attacker easy access to network resources. By using EAP the user has to input a credential set or present a certificate, which is mapped to the user's identity, not the wireless device. So if a device is stolen, the attacker still needs a valid credential set to access the network.

Three of the largest deficiencies in WEP are:

1. The use of static symmetric keys
2. Improper use of initialization vectors
3. The lack of integrity assurance.

WEP uses the RC4 symmetric stream cipher for encryption and decryption purposes. Symmetric means that the sender and receiver must use the same key for proper encryption and decryption functions. In the 802.11 standard the AP and the wireless device have to be configured with the same cryptographic key and because the standard did not include an automated way for updating these keys, most environments never change these keys out and many times each and every wireless device uses the exact same key. The more one key is used, the higher the probability of it being compromised, thus it should be changed out often. This is the same reason why passwords should be changed on a periodic basis to reduce the likelihood of them being uncovered and compromised.

N.B. (The rest of the article will explore the details of the TKIP mode of 802.11i being used, since backwards compatibility will be the concern for many individuals and companies.)

An initialization vector (IV) is a numeric seeding value used to add more randomness to the encryption process. This value is concatenated with the WEP symmetric key and inserted into the RC4 algorithm, which creates a key stream, as shown in Figure 4. The key stream values are XORed with the binary bits of the packet, which results in the encrypted format, or ciphertext. The 802.11 standard does not dictate that each packet should have a different IV value and does not outline how to properly increment or randomize these IV values. Many WLAN implementations use the same IV value over and over again and with the use of the same symmetric key continuously; this makes it simplistic for attackers to uncover the symmetric key. If the two ingredients (IV and key) are continuously the same, they generate the same output (key stream), and it is these patterns in the key stream that allow attackers to reverse-engineer the process and uncover the original WEP key. Once this original key is uncovered the attacker can decrypt all future packets that are encrypted with this instance of the key.

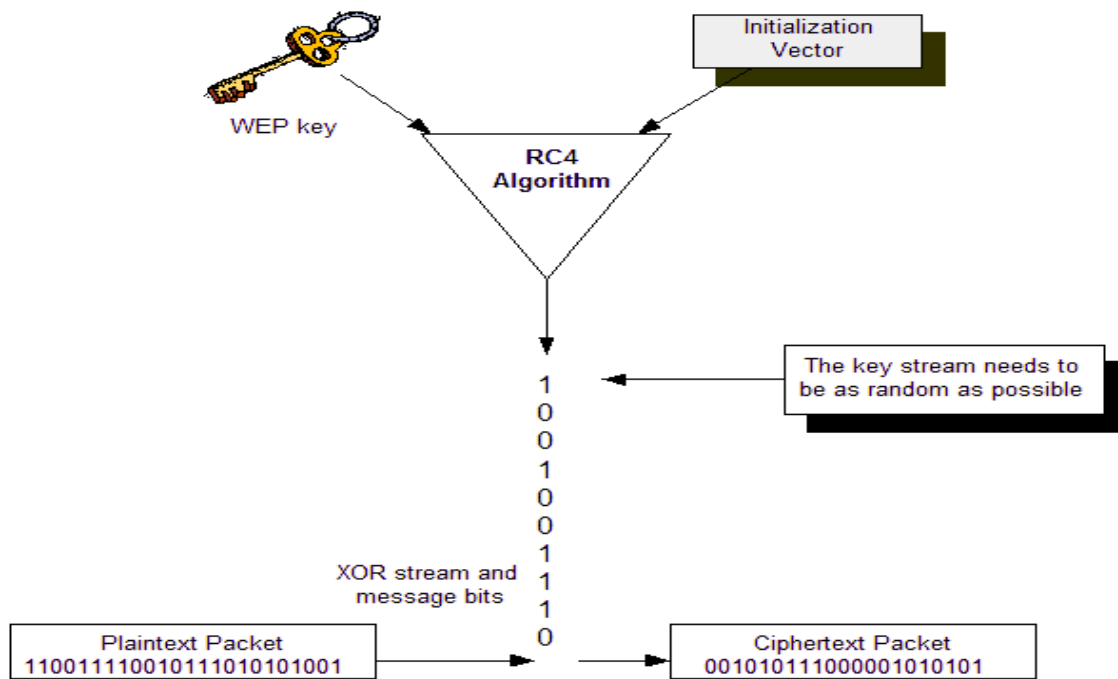


FIGURE 4

The third weakness is the lack of integrity assurance. Attackers can intercept 802.11 frames; change some data by flipping specific bits, and altering the Integrity Check Value (ICV) so that the receiver would never know that the modification took place. The ICV is a type of CRC function. The sender uses an algorithm to create an ICV value for the frame and inserts it into the frame's header. When the receiver accepts the frame, he calculates his own ICV value and compares it with the ICV value that was included with the frame. If the two values are the same, the receiver can be assured the frame was not modified, either intentionally or unintentionally, during transmission. If the values are different, the receiver is alerted that the frame was altered and it is discarded. WEP has a weakness where a frame can be modified and this type of check does not alert the receiver as it was supposed to.

So up to now our problems with the original 802.11 standard are; poor authentication, static WEP keys that can be easily uncovered, repetitive IV values that do not add enough randomness to the encryption process, and lack of proper integrity checking. The authentication issues are addressed with the use of 802.1X, which does not fully open the connection port until the user is properly authenticated, and the different types of EAP modules that can be used for mutual authentication. What about the other mentioned weaknesses?

HOW ARE THESE PROBLEMS FIXED IN THE NEW STANDARD?

The static WEP key, IV values, and integrity issues are addressed with what is referred to collectively as Temporal Key Integrity Protocol (TKIP).

The improper use of the key-scheduling algorithm within WEP allows tools like AirSnort and WEPCrack to be able to easily and quickly (minutes to hours) break WEP's encryption. This means that if a company is not using any third party wireless solution to add another layer of security, as in a VPN, their encrypted wireless traffic can be cracked whether they are using 40-bit or 128-bit keys, no questions asked. TKIP adds the ability to rotate encryption keys on a per-packet bases to help defeat these types of tools.

TKIP increases the IV value to ensure that each frame has a different IV value. Since the IV value is combined with the static WEP key, if the IV value changes for every packet that means that every packet is actually encrypted with a different key. (WEP key + IV = new key) This adds more randomness to the encryption process and makes the resulting key stream less predictable. In the end it means it makes it much harder for the attacker to uncover the original encryption key used.

TKIP also uses a Message Integrity Check (MIC) instead of an ICV function. MIC introduces a symmetric key into the hashing, or CRC, functions which will properly alert the receiver if a frame was modified during transmission. IV sequence numbers are also used, which helps protect against replay attacks. The receiving end will review each sequence number on each packet and if it receives a repetitive number it deduces that a replay attack may be underway and it discards the frame.

NOW, ARE WE HOME FREE?

The important question is does the use of 802.1X, EAP, TKIP, and AES mean that WLANs can now be securely implemented and we can now focus our security concerns on other technologies? Maybe, but then again, maybe not. Using TKIP is more of a quick fix than the necessary overhaul that is needed to make sure this standard provides the necessary level of protection. It is still based on the RC4 algorithm, which is not the best algorithm for the job. Secondly the use of all of these different protocols and technologies adds many more steps to the process and introduces more complexity. Security and complexity do not usually get along.

Security likes simplicity to ensure that all the entry points are clearly understood and properly protected. And thirdly, since there are different ways that authentication can take place (LEAP, EAP-TLS, and PEAP) companies will most likely run into interoperability issues between different vendor products. This means it maybe difficult to buy an AP from vendor A and wireless cards from vendor B and C and have them all work together seamlessly.

The easiest thing to do is criticize and point out the bad points in life, so it is important to emphasis that 802.11i does provide a much higher degree of security than the current 802.11 technology. It is just that 802.11i may have its own weaknesses and headaches that will come along. For example, in the use of EAP-TLS each wireless device would need its own digital certificate. How will the certificates be properly deployed to all the wireless devices? How will the certificates be maintained? Will the devices and authentication server verify that certificates have not been revoked by periodically checking a certificate revocation list (CRL)? And what if a rogue authentication server or AP was erected with a valid digital certificate? The wireless device would just verify this certificate and trust that this server is the entity it is supposed to be communicating with. And if the certificate authority is ever compromised the whole EAP-TLS infrastructure is compromised, as with any PKI environment.

But the 802.11i working group has been developing this standard for two years now and had some very knowledgeable and powerful companies helping them. The first wireless standard has been plagued with so much bad press, pertaining to security, it is highly likely that this group ensured that all of their 'T's were crossed and their 'I's dotted. It seems as though the technology chosen and put into place to protect future wireless communication is a better fit and will provide layers of the protection that is required. But time will tell. There are always three main items that fly in the face of good security when it comes to new or existing products: a standard that was not well thought out, vendors not interpreting the standard properly or not following it, and improper implementation of the product. Each of these components is troublesome, but the last problem is the most common. Individuals and companies need to understand the technology they are deploying, know how to properly test and secure it, and understand the product's deficiencies and vulnerabilities. Today's WLANs, which are based on a standard with flaws, could provide much more security and protection if the people who install and configure them had a better understanding of the technology itself. Security is based on education and knowledge. So the goal is to have a solid secure foundation in the products that we purchase and have the intelligence to properly implement them.

The WLAN products that will soon be developed and released should provide backward compatibility by using TKIP and have an option for customers to instead use the AES approach if the WLAN is a new extension to their environment. The Wi-Fi Alliance (WECA), who certifies Wi-Fi products, has already started to test the components of this new standard and assess the interoperability issues for proper compatibility. These new items are being integrated in their certification process so that customers will have a third party provide an assessment of the functionality and protection that is provided in the various WLAN products.