

Despite a sagging economy and the horrendous state of the telecommunications industry, demand for wireless products and bandwidth has remained strong. Certainly, the future of networking lies within the wireless realm. However, many obstacles remain in its path before wireless networking can become as common as wired computer networks.

Security is the single most frustrating issue presently facing wireless network developers and administrators. In a wire line network, a potential security violation must be internal to the network or over a tapped wire. However, a wireless network is vulnerable to anyone who knows how to intercept radio waves at the proper frequencies. Since the data is sent through the air, many traditional “wired” network security measures would be considerably less effective.

Security has been an issue for the wireless telephone market since its inception. It is a significant problem when people’s personal phone conversations are unknowingly monitored; but the damage when private and valuable data is easily pirated in the same way can be much greater. Consider a large company: vital, internal corporate data that involves millions of dollars and affects thousands of people could be stolen. Furthermore, almost all personal data for an individual is kept on a network, whether it is at home or in an office environment. If the future of networks is wireless, the issue of security clearly must be addressed.

802.11 standards

IEEE developed the 802.11 standard to specify wireless local area networks (WLANs). In 1999, the organization released 802.11a and 802.11b. Two different versions were released because each one standardizes equipment for a different frequency range. 802.11a deals with the 5 GHz range, which is costly but allows for superior performance. 802.11b deals with the 2.4 GHz range, the same as the original release. This range is cheaper but offers marginally reduced performance. 802.11b products are much more common today. However, many security issues went unresolved with both 802.11a and 802.11b.

802.11b made many improvements upon its predecessor, but its security

specifications were surprisingly light. 802.11b uses what is known as the Wired Equivalent Privacy (WEP) protocol

by WEP. This, theoretically, should keep unwanted users from transmitting packets over the wireless network if the encryption is functioning properly.

The final goal, data integrity, is taken care of by the integrity checksum portion of the protocol. The checksum is computed by both the receiver and the transmitter. If it does not match, then the packet is discarded. In this way, if the data has been manipulated in some fashion, it will be ignored.

The encryption algorithm used by WEP is based on the RC4 algorithm. The RC4 algorithm is both well known and well respected in the computer industry. It is considered a solid encryption algorithm by standard networking conventions. RC4 is based on the keystream method of encrypting. A long sequence of pseudo random bytes is generated, known as the initialization vector (IV). The IV is then processed through a function with a shared secret key, k . The result is the keystream. Note that the IV is attached to the message unencrypted, so that the receiver knows which one to use. (Of course, this means that the IV is known also to any attacker who is eavesdropping.)

The final step in the encryption process is to XOR the keystream and the plaintext together to obtain the ciphertext. The decryption process is completed by simply reversing the process with the same IV and k .

WEP also computes a checksum for every message that is sent across the network. This checksum is computed by summing the bits contained within the message. The value of the checksum is then concatenated on the end of the message to form the plaintext that will be encrypted with the RC4 algorithm. Once it has decrypted the message, the receiver recomputes the checksum. If the computed checksum matches the checksum concatenated to the message, then the integrity of the message has been verified. Figure 1 shows an overview of how the ciphertext is constructed and transmitted.

Problems with WEP

There are several reasons why this procedure has not proven to be entirely successful when implemented in real life WLANs. First, the key size used by the



to address security concerns. WEP itself is—more or less—an implementation of encryption with built-in message authentication and data integrity systems. The fact that the implementation is weak is only part of the problem. Providing a method for encryption is simply not sufficient to secure any type of system.

The WEP security protocol

The Wired Equivalent Privacy (WEP) protocol attempts to accomplish three security goals: confidentiality, access control and data integrity. These are typical main security functions for any network. WEP attempts to provide confidentiality by preventing any casual eavesdropping of packets being sent over the network. The encryption portion of the protocol is meant to address this goal. If an unauthorized user cannot decrypt the messages going across the network, then he or she cannot read the messages. Thus, they remain confidential.

The encryption algorithm used by WEP, in conjunction with shared key authentication, is also meant to attain the goal of access control. 802.11b compliant hardware has the option of discarding packets that are not properly encrypted

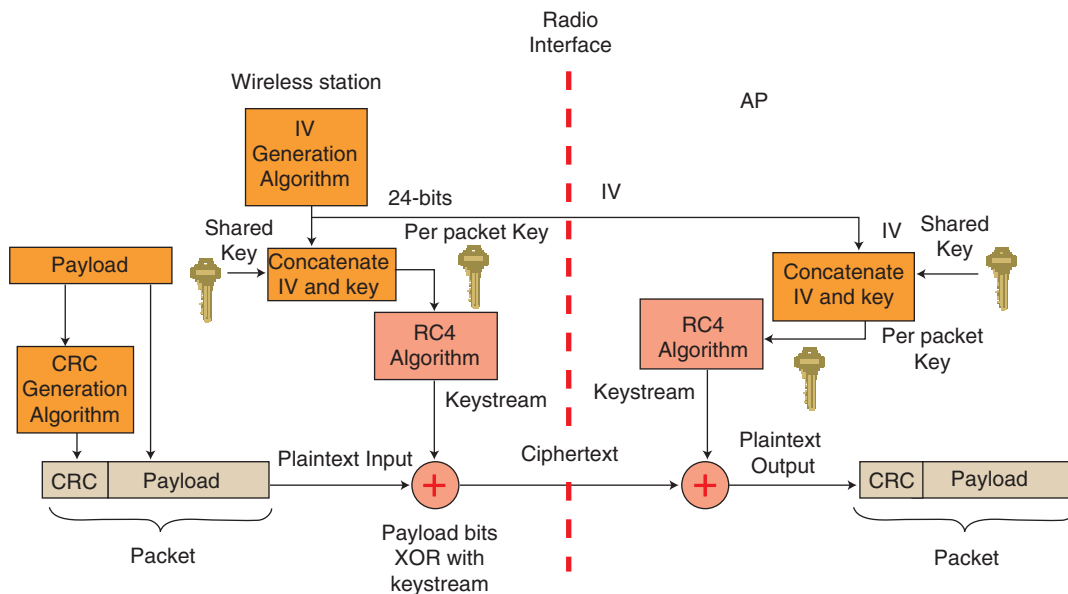


Fig. 1 An overview of how the ciphertext is constructed and transmitted.

original release of WEP is too small. The initial release of 802.11b called for the use of 40-bit keys. WEP keys 40-bits in length have been shown to be crackable in less than 50 hours by brute force methods. With a modest parallel attack, that number can be reduced to five hours. Vendors, as well as a later released update to WEP (known as WEP2), extended the protocol to use 104-bit keys, which are uncrackable by brute force methods on today's machines. However, other problems, external and internal to the protocol, make even the 104-bit keys crackable. Those problems include the heavy reuse of keys within WEP's encryption implementation, the ease of data access in a wireless network and the lack of any key management within the protocol.

Key reuse is a problem within WEP due to the nature of keystream encryption algorithms like RC4. Specifically, if we have the ciphertext of two messages that are encrypted with the same key, then applying the XOR function to the two ciphertexts will yield a result of the XOR of the two plaintexts.

Given certain properties of the English language (and most common languages spoken in the world), it is a computable problem using brute force methods to discern P1 and P2 given P1 XOR P2. Therefore, it is risky to reuse keystreams when using this method of encryption. However, that is exactly what WEP does.

WEP lacks any specifications on key management. This has caused some serious problems in some wireless networks and has led to many incompati-

ble, proprietary solutions. Key management is needed since the alternative is to use a single network wide secret key to distribute the message keys to users. Sharing a key with every user on the network greatly increases the chances that the key will not remain secret. For a system that relies so heavily upon the secrecy of its keys, the lack of any legitimate key management protocol within WEP is a serious vulnerability.

Remember that many of the weaknesses in WEP are exploited due to the nature of wireless networks. With wireless networks, intruders do not have to find a way through limited access points, firewalls and monitoring stations. Since all wireless networks operate by broadcasting their packets, anyone within range and with suitable equipment has access to the data being transmitted. Therefore, extra precautions need to be written into the protocol that WEP simply did not adequately address.

Another significant issue for WEP is its lack of any effective authentication protocol. Users of WEP have two choices when it comes to authentication: open or shared key. With open authentication, any user who requests authentication is authenticated. In other words, vendors who use this method need to use a proprietary protocol to authenticate users on the network. The other option provided by WEP is shared key authentication. Unfortunately, the shared key authentication implemented by the protocol is vulnerable because of the weakness of the WEP encryption explained earlier and the fact that the authentication is only done in one direction.

WEP's shared key authentication functions as a standard challenge-response system. The initiator sends an authentication request and the responder replies with a challenge. That challenge is encrypted by the initiator and returned to the responder. If the decrypted challenge matches the original sent by the responder, then the user is authenticated. Several vulnerabilities exist within this authentication protocol. The use of one-way authentication, in conjunction with the weak encryption provided by WEP, creates most of the

problems. These vulnerabilities have led to shared key attacks, as well as man-in-the-middle attacks, which allow unauthorized users access to the network.

Successful attack approaches

Since the packets being sent across a wireless network are so easy to intercept, the number of attacks on 802.11b networks has been astounding, especially in densely populated, corporate areas. Many popular technology publications have documented their successful ventures of tapping into corporate 802.11b networks.

First, wireless networks are quite vulnerable to sniffing. Sniffing occurs when an unauthorized user is able to gain access to the network and read all the packets that are transmitted across it. In a traditional LAN, one would need physical access to the network in order to sniff. However, with a wireless LAN, one simply needs a laptop, a wireless network card and a strong enough antenna. This ability would not be too much of a problem, if it were not for the weakness of WEP and the wide variety of freely available sniffing tools that can also crack WEP keys. (As a result, many WLAN administrators do not even bother to turn WEP on, which even further enhances this vulnerability.)

One of the most common sniffing tools is known as AirSnort. AirSnort was released in the summer of 2001 as free-ware shortly after the shortcomings of the original WEP release were documented publicly. AirSnort can automatically determine WEP keys on an 802.11b WLAN simply by sniffing

enough traffic. It takes advantage of the way WEP reuses keys and how the RC4 algorithm is implemented. Depending on how many packets AirSnort can sniff, it can crack the WEP master encryption key in as little as three hours. This is accomplished by exploiting the weakness in the key scheduling of the RC4 algorithm and the reuse of keys, which was discussed earlier. Once the master key is known, the user has access to the network as any legitimate node would.

Another common attack on 802.11b wireless LANs involves user authentication. WEP's shared key authentication protocol is vulnerable to man-in-the-middle attacks. A man-in-the-middle attack occurs when an intruder introduces himself as a new node by placing himself between a valid host and its access point. The intruder can then intercept the challenge from the responder and send it on to the initiator. The initiator sends its encrypted response, which is intercepted and passed on to the responder. Now the intruder has effectively been authenticated on the network. The intruder will be a valid user on the network at least until a reauthentication transmission takes place, depending on how the network is implemented. Figure 2 illustrates a man-in-the-middle attack.

The man-in-the-middle attack is only successful because the WEP encryption can be broken so easily. If the encryption functioned more effectively, the intruder would have a much more difficult time determining which messages were the challenge and the challenge response. Additionally, a man-in-the-middle attack would be much more difficult to successfully pull off if the access point were required to verify the identity of the node that sent it the authentication request and the node was required to verify the access node to which it is authenticating. This is known as two-way authentication.

802.11b networks have also been found to be vulnerable to denial-of-service (DoS) attacks. DoS attacks are a big problem for all types of networks as they are difficult to defend against most of the time. However, there are certain provisions within the WEP standard that make 802.11b networks more vulnerable to DoS attacks than they need to be.

One important feature of a wireless network is the ability for nodes to associate and disassociate with the network. This is how the network can expand

and contract in size with little hassle. However, with WEP associate and disassociate messages are not authenticated. Due to this fact, a rogue node can continually send streams of associate and disassociate packets. Since these are not authenticated in any way by WEP, the network must process all of these requests. At some point, this may disrupt the ability of legitimate nodes to associate, disassociate, or even pass normal traffic through the network.

All networks are vulnerable to certain flood attacks to some degree. However, the lack of authentication of the associate and disassociate packets on 802.11b networks makes them especially vulnerable. If these packets were authenticated, then the invalid messages could be ignored. This would significantly reduce the chance of a denial-of-service attack.

A simple plaintext attack that WEP is vulnerable to revolves around the reuse of keys and the weakness inherent in the way its Cyclic Redundancy Check (CRC) checksums are handled. Specifically, by using a form of induction, a key dictionary can be generated by simply knowing enough specific plaintext, which can be easily acquired.

The first step is to choose a pseudo random stream of size n , which is recovered by identifying the Dynamic Host Configuration Protocol (DHCP) discover messages from an external host. The second step is to create an administrative datagram, such as an Address Resolution Protocol (ARP) request or a Universal Datagram Protocol (UDP) open, of size $n-3$. The attacker then appends the first three bytes of the CRC to the end and XORs this message with the original pseudo random stream. The attacker now has his ciphertext minus the final byte of the CRC. This byte can be determined by iterating over its 255 possible values and then sending each version of the message to an access node. Whichever iteration gives the proper response from the node is the correct value.

With the value of the entire CRC known, we now have matching plaintext and ciphertext. With this information, we can then determine the value of the key that was used for this transmission. This, on average, would take about 40 minutes to complete. An entire key dictionary could be constructed with this attack in 46 hours. With eight attacking hosts working in parallel, this dictionary could be assembled in less than six hours. While the dictionary's

size would be somewhat impractical (~35 GB), it is still a serious threat to the network's security. This is especially true since the entire dictionary is not required to gain access to the network.

Improving WEP's authentication protocol, with the addition of a keyed message integrity check (MIC) would stop this attack. At the very least, the way the CRC is handled should be modified so that it cannot be easily manipulated. This would at least mitigate the attack and make it much more difficult to accomplish.

802.11i security specs

The sheer number and variety of vulnerabilities discovered within WEP shows what could arise when security is not designed from the ground up. There is hope on the horizon, however. The future of wireless LAN security is currently being entrusted to 802.11i. IEEE is developing this wireless LAN standard. It focuses strictly on security and improving upon the protocols offered by the previous 802.11 standards. It offers new intriguing security options and certainly appears to be more robust, at least on the surface.

There are three main areas that the IEEE 802.11i wants to improve on over 802.11b: 1) authentication, 2) key management and 3) data transfer. All of these areas were severely lacking in WEP. Some of the areas have received a significant facelift (data transfer, authentication) while others have been designed from scratch (key management). Each of them provides a new significant layer of security within 802.11 WLANs.

802.11i adds a significantly more robust authentication system to the standard. The shared key authentication of WEP was performed entirely between the access point and the host attempting to connect. Additionally, it was only a one-way authentication (from host to access point). To improve authentication robustness, 802.11i adds the need for an authentication server. It also implements a two-way authentication method to prevent the man-in-the-middle attacks that have been so prevalent on 802.11b networks.

Several new keys have been introduced in 802.11i to make two-way authentication possible. The first is the master key (MK). This key is a private, symmetric key that facilitates authentication of a host with the authentication server. Only those entities may possess the master key. The pairwise master key (PMK) is a private, symmetric key

that is used by the host and access point to control access to the network. Both of these new keys are only valid for the host's current session.

The authentication process can be divided into two distinct paths. The first is the host to access point communication and the second is the access point to authentication server communication. The host to access point communication is handled by a revised extensible authentication protocol (EAP). The original version of EAP was used in the 802.1X standard and proved to contain certain vulnerabilities. 802.11i hopes to have corrected those oversights by adding a two-way mechanism to the authentication.

The access point to authentication server communication is handled by the previously defined RADIUS protocol. RADIUS is outside the scope of the 802.11i specifications; however, it has been in practical use for long enough that it is assumed to be trusted. Note that this also means that the authentication server is assumed to be a trusted third party by 802.11i.

The second major area that 802.11i hopes to improve upon is key management. (However, since WEP lacks any real key management specifications, anything would be an improvement.) Besides the MK and PMK, there is the pairwise transient key (PTK), the key confirmation key (KCK), the key encryption key (KEK), the group transient key (GTK) and the temporal key (TK). Clearly, with all of these keys some form of reliable key management is required for 802.11i.

Key management functions in the following way in 802.11i. The first step is to use RADIUS to pass the PMK from the authentication server to the access point. This is a normal step in the authentication process described earlier. The second step is to use the PMK and a process known as the 4-way handshake to derive and verify the PTK. The final step is to use a procedure identified as the group key handshake to send the GTK from the access point to the host. Via this relatively simple process, 802.11i provides secure key management.

The last important area 802.11i sought to improve over the WEP is data transfer. Three different methods for

data transfer are specified. They are known as CCMP (Counter with Cipher Block Chaining Message Authentication Code Protocol), TKIP (Temporal Key Integrity Protocol) and WRAP (Wireless Robust Authenticated Protocol). They all serve different functions within the standard; however, the reason for three protocols is not technical. Clearly, one solid method would be preferable to allowing vendors to choose from a list of three. There are many issues surrounding these protocols that demanded they all be included, which are detailed below.

CCMP is seen as the long-term solution for data transfer over wireless LANs. It uses the newly approved AES encryption standard to encrypt data. This method encrypts both the payload and the MIC. CCMP was designed from the beginning to handle packet-based communications and it provides both

that is intended to patch the holes discovered in WEP. It is designed to encapsulate all WEP traffic in a secure manner that masks its known problems. This is similar to the strategy that many network administrators currently use. They often will employ a virtual private network (VPN) on top of any WEP traffic. The problems specifically addressed by these encapsulation are data forgery, replay attacks, encryption misuse and key reuse. The important aspect of TKIP here is that it can be implemented on existing 802.11b hardware. Therefore, only a software upgrade is necessary to convert from WEP to TKIP. While this does not provide as robust a solution as CCMP, it does improve the security of data transfer for pre-existing wireless LANs.

While 802.11i's focus has been on improving authentication, key management and data transfer for wireless

LANs, there are several other minor security related improvements in the standard. One is pre-authentication to support roaming hosts. This involves access points communicating to each other to confirm identity before the host has moved out of range of the original access point. That way a completely new authentication does not have to take

place between the host and the second access point. 802.11i also provides a method for further confirmation of the host's legitimacy on the network. In addition, there are provisions for password-to-key mappings and random number generation. Both of these aspects are completely ignored by WEP. While these minor features may not be as important, they certainly add to the robustness of the standard.

Improvements made

When examining the security provided by 802.11b and 802.11i, it is clear that 802.11i offers several significant improvements. The use of AES by 802.11i over the inferior RC4 algorithm by 802.11b is a significant step forward. AES has thus far proven to be an extremely solid encryption algorithm and likely will be quite usable for decades to come. The computing power

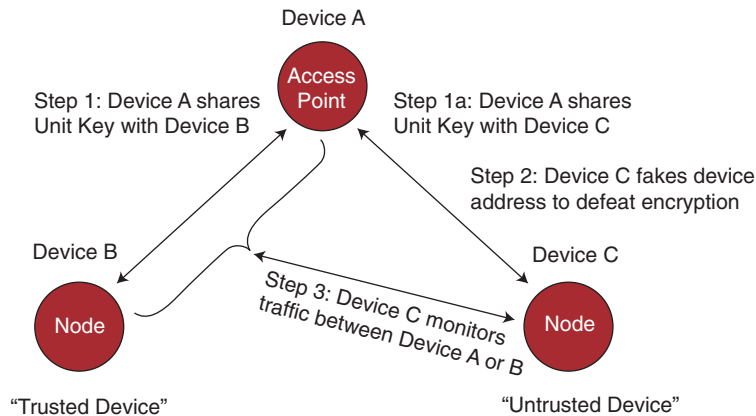


Fig. 2 Illustration visualizes a man-in-the-middle attack.

authenticity and privacy by encoding the plaintext before encrypting it. This method has been shown to be provably secure. However, the use of AES would require a hardware upgrade when converting 802.11b networks to 802.11i. This upgrade would prohibit the use of 802.11i because hardware upgrades are both expensive and time consuming. It is mainly for this reason that the other two protocols are included in the 802.11i standard.

WRAP is the original implementation of AES for wireless LANs. However, the methodology behind WRAP was found to be faulty and contain several exposable vulnerabilities, similar to the ones found in WEP. However, since several vendors have already implemented hardware based on this specification, it was included in the 802.11i standard for completeness.

TKIP provides a data transfer solution

needed to crack AES is nearly unfathomable at this point. The only weakness that AES may present is due to its relative newness. It has not been around quite long enough to ensure that it has no potential backdoors. However, this is a highly unlikely scenario given the thoughtfulness that went into the creation of the algorithm and the scrutiny it has already undergone.

The addition of two-way authentication also gives 802.11i an important advantage over 802.11b. One of the most exploited attacks on 802.11b networks is the man-in-the-middle attack, which takes advantage of the weaknesses inherent with one-way authentication. In addition, the fact that the MICs are now keyed is a significant step forward. The unkeyed MICs in 802.11b made message integrity a cantankerous problem for WLAN administrators.

802.11i also deals with 802.11b's other big weakness: the lack of any key management. The key management proposed in 802.11i is quite complex, but it needs to be considering all the administrative keys that are created by the protocol. The theory behind 802.11i's key management is strong, so it should provide the level of security needed.

802.11i strengthens the overall security architecture for wireless LANs. Security was not designed into these types of networks from the beginning. Nevertheless, 802.11i has done an admirable job in attempting to reorient the architecture as best as it could to improve the security of these networks.

An eye toward the future

802.11i does have some potential pitfalls. To implement CCMP (the strong AES encryption prescribed by the standard) a hardware upgrade of all 802.11b access nodes is necessary. This is a potentially prohibitive problem due to the cost that would be involved for pre-existing networks. Using WRAP or TKIP may prove to not be sufficient as 802.11i ages. However, this will not be a problem for new networks that are installed using the 802.11i standard.

In addition, 802.11i requires an authentication server for its two-way authentication. 802.11b does not have this requirement. This should not be too big a problem since almost all networks already have an authentication server of some sort. However, it is possible that some networks will need more hardware to implement this important feature.

802.11i also relies heavily on the secrecy of its session keys. The specifications appear to take this need for secrecy into account. However, if any unnoticed loophole were exploited that allowed these keys to be known to an attacker, much of 802.11i's security would be compromised.

The final potential weakness of 802.11i that concerns the author is the significant increase in complexity. Granted, this increase is necessary to adequately protect wireless LANs. However, this complexity could give rise to potential backdoors in the future that were not foreseen as the standard was being written.

The need and demand for new security standards in the WLAN arena has driven the formation of 802.11i. Clearly, its apparent strengths much outweigh any potential weaknesses it may contain at this point. Despite these facts, there has been no rush by vendors to implement any products that are 802.11i compliant. The hardware concerns as well as the general concerns over wireless network security have caused this hesitancy. The slumping economy has not helped the situation either. No one wants to get burned if 802.11i turns out to be nothing more than WEP3. A detailed analysis of the standard shows that this is unlikely to be the case, yet the vendors remain reticent.

There is another more subtle issue at hand here as well. The general principle guiding the 802.11i standard was fixing specific problems contained within 802.11b. Certainly, fixing existing problems is a necessary step; however, the process of simply patching 802.11b is not sufficient. 802.11i is much more than a patch of 802.11b, but does it go far enough?

Many of the problems that WEP ended up having were not envisioned during its inception; so, how can anyone know what problems 802.11i will face down the road? The most complete, long-term solution for WLAN security will be a complete renovation of the design methodology and an overhaul of the network architecture. Until security is designed in from day one on wireless local area networks, their level of trust is unlikely to ever be higher than adequate.

Read more about it

• Shapiro, Carl and Varian, Hal R., *Information Rules* Published by Harvard Business School Press, 1999 – pg. 236

• Foster, Matt, "Wireless Local Area Networking: An Introduction," <<http://www.tomshardware.com/network/01q3/010822/>>, February, 2003

• Borisov, Goldberg and Wagner, "Intercepting Mobile Communications: The Insecurity of 802.11," <<http://www.drizzle.com/~aboba/IEEE/wep-draft.zip>>, February 2001

• Arbaugh, William A. and Shankar, Narendar, "Your 802.11 Wireless Network has No Clothes," <<http://www.drizzle.com/~aboba/IEEE/wireless.pdf>>, March 2001

• Arbaugh, William A. and Mishra, Arunesh, "An Initial Security Analysis of the IEEE 802.1X Standard," <<http://www.cs.umd.edu/~waa/1x.pdf>>, February 2002

• Ellison, Carl, "Exploiting and Protecting Wireless Networks," <<http://www.extremetech.com/article/0,3396,apn%253D2%2526s%253D1024%2526a%253D13880%2526ap%253D1,00.asp>>, Sept, 2001

• Aboba, Bernard, "WEP2 Security Analysis," <<http://www.drizzle.com/~aboba/IEEE/11-01-253r0-I-WEP2SecurityAnalysis.ppt>>, May, 2001

• Arbaugh, William A., "An Inductive Chosen Plaintext Attack Against WEP/WEP2," <<http://www.cs.umd.edu/~waa/attack/frame.htm>>, March, 2002

• Cam-Winget, Moore, Stanley and Walker, "IEEE 802.11i Overview," <http://csrc.nist.gov/wireless/S10_802.11i%20Overview-jw1.pdf>, December, 2002

• Karygiannis, Tom and Owens, Les, "Wireless Network Security: 802.11, Bluetooth and Handheld Devices" NIST – Special Publication 800-48. November 2002

• Eaton, Dennis, "Diving into the 802.11i Spec: A Tutorial," <http://www.commsdesign.com/design_library/cd/hn/OEG20021126S0003>, September 2003

About the author

Brandon Brown is currently pursuing his MS-MIS and MBA degrees from North Central College in Naperville, Illinois. He received his bachelor's degree in Computer Engineering from the University of Michigan in Ann Arbor. Brandon worked at Tellabs Operations, Inc. in Lisle, Illinois for four years as a software engineer working on the Titan 5500 Digital Cross-Connect System. He is looking forward to continuing his career in the networking field upon the completion of his degree program at North Central College.