Security Analysis and Improvements for IEEE 802.11i

Changhua He John C Mitchell

Electrical Engineering and Computer Science Departments
Stanford University, Stanford CA 94305

Abstract

This paper analyzes the IEEE 802.11i wireless networking standard with respect to data confidentiality, integrity, mutual authentication, and availability. Under our threat model, 802.11i appears to provide effective data confidentiality and integrity when CCMP is used. Furthermore, 802.11i may provide satisfactory mutual authentication and key management, although there are some potential implementation oversights that may cause severe problems. Since the 802.11i design does not emphasize availability, several DoS attacks are possible. We review the known DoS attacks on unprotected management frames and EAP frames, and discuss ways of mitigating them in 802.11i. The practicality of a DoS attack against Michael MIC Failure countermeasure is discussed and improvements are proposed. Two new DoS attacks and possible repairs are identified: RSN IE Poisoning and 4-Way Handshake Blocking. Finally some tradeoffs in failure-recovery strategies are discussed and an improved variant of 802.11i is proposed to address all the discussed vulnerabilities.

1 Introduction

As Wireless Local Area Networks (WLANs) become more widely deployed, wireless security has become a serious concern for an increasing number of organizations [15, 38]. A summary of relevant literature on wireless security research appears in the Appendix, including review of standard definitions and acronyms. Generally, the security requirements for a WLAN include data confidentiality, integrity, mutual authentication, and availability.

IEEE 802.11i [21], an IEEE standard ratified June 24, 2004, is designed to provide enhanced security in the Medium Access Control (MAC) layer for 802.11 networks. The 802.11i specification defines two classes of security algorithms: Robust Security Network Association (RSNA), and Pre-RSNA. Pre-RSNA security consists of Wired Equivalent Privacy (WEP) and 802.11 entity authentication. RSNA provides two data confidentiality protocols, called the Temporal Key Integrity Protocol (TKIP) and the Counter-mode/CBC-MAC Protocol (CCMP), and the RSNA establishment procedure, including 802.1X authentication and key management protocols.

This paper analyzes security aspects of the 802.11i specification, considering data confidentiality, integrity, mutual authentication, and availability. Our analysis suggests that 802.11i is a well-designed standard for data confidentiality, integrity, and mutual authentication, promising to improve the security of wireless networks. At the same time, some vexing Denial-of-Service (DoS) attacks remain. We review the known DoS attacks and describe appropriate countermeasures. We also describe two new DoS attacks – RSN Information Element (RSN IE) Poisoning and 4-Way Handshake Blocking – and present countermeasures for these. We also analyze the failure-recovery strategy in 802.11i and discuss associated tradeoffs. Finally we outline an improved version of 802.11i that addresses all the vulnerabilities discussed in this paper.

In proceeding through the analysis of 802.11i, we describe several implementation considerations and suggest ways to avoid a range of security problems. Here is a concise list of our primary recommendations: First, CCMP should be used for data confidentiality whenever possible because WEP and TKIP have inherent weaknesses. Second, mutual authentication must be implemented carefully to achieve security objectives, as elaborated in the next paragraph. Third, several implementation details are important for addressing DoS vulnerabilities in the MAC layer. Finally, the efficiency of failure recovery may be improved by using a modified strategy, subject to certain tradeoffs.

Strong mutual authentication may be achieved by a combination of mechanisms. First, an appropriate Extensible Authentication Protocol (EAP) [9] method, such as EAP-TLS [2], should be used to prevent Man-in-the-Middle (MitM) attacks. Second, the RADIUS [34] secret and the passphrase for Pre-Shared Key (PSK) generation should be chosen carefully to defend against dictionary attacks. Third, if Pre-RSNA and RSNA

algorithms are allowed to run simultaneously in a WLAN, a user should have a chance to *manually* decide which to use prior to opening a connection, and Access Points (APs) should impose different privilege policies for each. Otherwise, an adversary may compromise the security of the entire system through a Security Level Rollback Attack. Finally, in order to avoid reflection attacks, no single device should serve as both the authenticator and the supplicant when executing the 4-Way Handshake under the same Pairwise Master Key (PMK). This restriction is straightforward for standard infrastructure networks, but may have some impact on possible uses of 802.11i in combination with ad hoc networks.

A number of seemingly minor modifications make 802.11i more robust against DoS attacks. First, 802.11i can eliminate known DoS attacks on the EAP packets by simply ignoring certain packets. Second, when TKIP is adopted, the Michael MIC (Message Integrity Code) Failure countermeasure could be implemented with improved TKIP Sequence Counter (TSC) updating and without re-keying in order to make the DoS attack more difficult. Third, in the RSN IE confirmation mechanism, the verification condition should be relaxed to mitigate a potential DoS attack. Finally, in the 4-Way Handshake, it is better for a supplicant to re-use the same nonce until one instance is completed successfully.

The paper is organized as follows. Section 2 describes threats in wireless networks. Section 3 analyzes the data confidentiality and integrity protocols of 802.11i. Section 4 analyzes the RSNA establishment procedure and indicates possible implementation mistakes. Section 5 discusses the practical DoS attacks and proposes an improved version of 802.11i. Section 6 concludes the paper.

2 Wireless Threats

In order to analyze the 802.11i protocol, it is important to characterize the likely capabilities of any adversary. From the Link Layer of a WLAN, there are three possible types of frames: Management Frames, Control Frames, and Data Frames. Any manipulation of these frames that directly or potentially jeopardizes data confidentiality, integrity, mutual authentication, and availability will be considered a threat. In this section, we outline some forms of attack that we consider and evaluate in our analysis.

Threat 1. Passive Eavesdropping/Traffic Analysis

Due to the characteristics of wireless communication, an adversary can easily sniff and store all the traffic in a WLAN. Even when messages are encrypted, it is important to consider whether an adversary may learn partial or complete information from certain messages. This possibility exists if common message fields are predictable or redundant; further, encrypted messages may be generated upon the requests from the adversary itself. In our analysis, we consider whether recorded packets and/or knowledge of the plaintext can be used to reveal the encryption key, decrypt complete packets, or gather other useful information through traffic analysis techniques.

Threat 2. Message Injection/Active Eavesdropping

An adversary is capable of inserting a message into the wireless network with moderate equipment, such as a station with a common wireless Network Interface Card (NIC) and some relevant software. Although the firmware of most wireless NICs may limit the interface for composing packets to the 802.11 standard, an adversary is still able to control any field of a packet using known techniques [8]. Hence, it is reasonable to assume that an adversary can generate any chosen packet, modify contents of a packet, and completely control the transmission of the packet. If a packet is required to be authenticated, the adversary may be able to break the data integrity algorithm to make a valid packet. The adversary can also insert a replayed packet, if there is no replay protection or the adversary is able to avoid it. Furthermore, by inserting some well-chosen packets, the adversary might be able to learn more information from the reaction of the system through active eavesdropping.

Threat 3. Message Deletion and Interception

We assume that an adversary is able to do message deletion, which means that an adversary is capable of removing a packet from the network before it reaches its destination. This could be done by interfering with the packet reception process on the receiver's antenna, for example by causing CRC (Cyclic Redundancy Checksum) errors so that the receiver drops the packet. This process is similar to ordinary packet errors due to noise, but may be instigated by an adversary.

Message interception means that an adversary is able to control a connection completely. In other words, the adversary can capture a packet before the receiver actually receives it, and decide whether to delete the packet or forward it to the receiver. This is more dangerous than the eavesdropping and message deletion. Furthermore, it differs from eavesdropping and replaying, because the receiver does not get the packet before the adversary forwards it. Message interception may seem difficult in wireless LANs because the legitimate receiver might detect a message as soon as the adversary does so. However, a determined adversary *does* have some potential ways to achieve message interception. For example, the adversary can use a directional antenna to delete a packet on the receiver side, while simultaneously using another antenna to receive the packet itself. Since message interception is relatively difficult to achieve, we

only consider this possibility when the damage caused is relatively severe. Note that it is not necessary for the adversary to perform a Man-in-the-Middle (MitM) attack in order to intercept packets.

Threat 4. Masquerading and Malicious AP

Because the plaintext MAC addresses are included in all packets transmitted through wireless links, an adversary can learn valid MAC addresses by eavesdropping. The adversary is also capable of modifying its MAC address to any value because most firmware provides the interface to do so. If a system uses MAC address as the only identification of the wireless devices, the adversary can masquerade as any wireless station by spoofing its MAC address; or can masquerade as an Access Point (AP) by spoofing its MAC address and functioning appropriately through appropriate freeware (e.g., HostAP). An adversary is also able to install his own AP, with a forged MAC address and a spoofed SSID. Alternatively, without masquerading as others, it is possible for a malicious AP to provide a strong signal and attempt to fool a wireless station into associating with it and leaking credentials or private data.

Threat 5. Session Hijacking

We consider that an adversary may be able to hijack a legitimate session after the wireless devices have finished authenticating themselves successfully. Here is one possible scenario for achieving this. First, the adversary disconnects a device from an existing session, and then masquerades as this device to obtain possible connections without the attention of the other device. In this attack, the adversary is able to receive all packets destined to the hijacked device and send out packets on behalf of the hijacked device. This attack could conceivably circumvent any authentication mechanism in the system. However, if data confidentiality and integrity protocols are used, the adversary must break them in order to read encrypted traffic and send out valid packets. Thus this attack against authentication can be prevented by sufficiently powerful data confidentiality and integrity mechanisms.

Threat 6. Man-in-the-Middle

This attack is different from message interception because the adversary must participate in communication continuously. If there is already a connection between a wireless station and the AP, the adversary must break this connection first. Then, the adversary masquerades as the legitimate station to associate with the AP. If the AP adopts any mechanisms to authenticate the station, the adversary must be able to spoof the authentication. And finally, the adversary must masquerade as the AP to fool the station to associate with it. Similarly, if the station adopts some mechanism to authenticate the AP, the adversary must spoof the AP's credentials. Another possible approach for the adversary to launch a MitM attack is ARP cache poisoning, as in a wired LAN [17].

Threat 7. Denial-of-Service

WLAN systems are quite vulnerable to DoS attacks. An adversary is capable of making the whole Basic Service Set (BSS) unavailable, or disrupting the connection between legitimate peers. Using characteristics of wireless networking, an adversary may launch DoS attacks in several ways. For example, forging the unprotected management frames (e.g., Deauthentication and Disassociation), exploiting some protocol weaknesses, or straightforward jamming of the frequency band will deny service to legitimate users. However, we only consider DoS attacks that require reasonable effort on the part of the adversary. For instance, deleting all packets, using the message deletion techniques described in *Threat 3*, consumes considerable resources and may not be considered a relevant DoS attack because it is just like a frequency jamming.

Threats 1, *2*, and *3* attack all three types of frames in the Link Layer, possibly breaking data confidentiality and integrity of a WLAN. *Threats 4*, *5*, and *6* defeat mutual authentication; generally they arise from compositions of *Threats 1*, *2*, and *3* on management frames. *Threat 7* interferes with availability, and could result from *Threats 1*, *2*, and *3* on any type of frames. The following sections will analyze the effectiveness of 802.11i for defending against these threats; appropriate suggestions or modifications are proposed if 802.11i cannot eliminate these threats.

3 Data Confidentiality and Integrity

IEEE 802.11i defines three data confidentiality protocols: Wired Equivalent Privacy (WEP), Temporal Key Integrity Protocol (TKIP), and Counter-mode/CBC-MAC Protocol (CCMP). The vulnerabilities of WEP and TKIP have been studied extensively [4, 5, 10, 11, 18, 28, 33, 39, 40] (See the Appendix). Therefore, this section will focus on analyzing CCMP. Note that a fresh Temporal Key (TK) is assumed to be shared between the peers before executing any data confidentiality protocols.

Unlike the RC4 stream cipher used in WEP and TKIP, CCMP uses the CCM (Counter with CBC-MAC) operation mode [41] of the AES encryption algorithm [31] with a 128-bit key and a 128-bit block size. CCMP combines the counter mode (CTR) for data confidentiality and the Cipher Block Chaining Message Authentication Code (CBC-MAC) for data integrity, using an 8-octet MIC (Message Integrity Code) and a 2-octet Length field. We assume that a 128-bit key is secure against brute-force

attacks on AES. With AES, it is possible to use a single 128-bit key to encrypt all packets, eliminating the problems of key scheduling algorithms associated with WEP and TKIP. CCMP also provides MIC protection over both the frame body and nearly the entire header in a MAC frame, which prevents an adversary from exploiting the MAC headers. In addition, CCMP uses a 48-bit Packet Number (PN) to prevent replay attacks and construct a fresh nonce for each packet. The sufficient space of PN eliminates any worry about PN re-usage during an association.

A possible vulnerability might arise from the fact that CCM uses the same key for both confidentiality and integrity. However, CCM appears unproblematic because it guarantees that the space of the CTR never overlaps with the space of the CBC-MAC initialization vector. If AES behaves like a pseudo-random permutation, which is a plausible assumption, and the cipher operates on two separate spaces, which is guaranteed by CCM, then the outputs of the cipher will be independent.

Another possibility is pre-computation attacks. While a 128-bit key is considered secure against brute-force attacks; CCMP uses an incremental PN to construct nonces, and the PN is initialized to one for every fresh TK. This allows a common pre-computation attack. An adversary might compute a table offline for one specific nonce and 2^{64} possible keys. Then the adversary starts to observe the online messages encrypted with this specific nonce and an unknown key. On average the adversary could find an overlap of keys after observing 2^{64} messages with the specific nonce and different keys, obtaining the TK for that session. This pre-computation attack reduces the key space from 2^{128} to 2^{64}, which is possible to be broken practically for a block cipher. However, since CCMP also includes the source MAC address in constructing the nonce, once the adversary chooses a specific value to build the offline table, the same nonce will only appear for the specific station. Furthermore, since a PN will never repeat for the same TK, the adversary may need to wait for refreshed TKs, which means different sessions, in order to observe more messages with the same nonce. Therefore, the combination of the PN and the MAC address requires that an adversary keep observing 2^{64} different sessions for a specific station in order to break one TK. Additionally, 2^{64} entries may consume significant resources to store and search efficiently. Hence, this is not considered a practical attack.

Although there has been some criticism [35], analysis suggests that CCM provides a level of confidentiality and authenticity comparable to other authenticated encryption modes, such as OCB mode [23]. Hence, it is reasonable to believe that, once CCMP is implemented, an adversary is not able to break the data confidentiality and integrity without the knowledge of the key. Furthermore, an adversary cannot obtain useful information about the key through analyzing the cipher text even if the corresponding plaintext is known.

For completeness, we discuss the threats listed in Section 2 in order. With regard to *Threat 1*, eavesdropping and traffic analysis, an adversary may eavesdrop on traffic, but it cannot decrypt the packets because it has no way to discover the TK. Furthermore, since the IP header of the messages is encrypted, the adversary can only obtain limited information through traffic analysis, compared to a higher layer encryption method such as IPsec and SSL. However, the adversary *does* have some ways to discover useful information, because the MAC header is not encrypted; the packet size and frequency are observable. Fortunately, in most scenarios such information leakage is not considered to be harmful.

Threat 2 on data frames is completely eliminated because a strong MIC prevents an adversary from inserting a forged data message. Data modifications are similarly prohibited by the MIC. Further, a replayed packet will be discarded silently because the corresponding PN is out of order. For *Threat 3*, the adversary is able to delete a packet in any case; however, this can be handled by the retransmission mechanism or higher layer protocols. On the other hand, the adversary is also able to intercept a packet and forward this packet to the receiver later. The forwarded packet could have been correctly encrypted with a valid MIC; however, the receiver is likely to recognize this as an out-of-order packet and discard it silently.

In summary, against *Threats 1*, *2*, and *3*, CCMP appears to provide satisfactory data confidentiality, integrity, and replay protection for data packets, as intended. However, since management frames and control frames are neither encrypted nor authenticated by the Link Layer encryption algorithm, they are still vulnerable to these threats. In addition, not surprisingly, CCMP requires hardware upgrades and might have some impacts on performance.

4 Authentication and Key Management

Prior work has shown that 802.11 entity authentication (Open System Authentication and Shared Key Authentication) are completely insecure [4, 10]. Therefore, 802.11i defines the Robust Security Network Association (RSNA) establishment procedure to provide strong mutual authentications and generate fresh TKs for the data confidentiality protocols. This section will analyze the RSNA handshakes in details.

4.1 RSNA Establishment Procedure

802.11i RSNA establishment procedure consists of 802.1X authentication and key management protocols. Three entities are involved, called the Supplicant (the wireless station), the Authenticator (the Access Point), and the Authentication Server (de facto a RADIUS server [34]). Generally, a successful authentication means that the supplicant and the authenticator verify each other's identity and generate some shared secret for subsequent key derivations. Based on this shared secret, the key management protocols compute and distribute usable keys for data communication sessions. The authentication server can be implemented either in a single device with the authenticator, or through a separate server, assuming the link between the authentication server and the authenticator is physically secure. The complete handshakes of establishing a RSNA are shown in Figure 1. For the purpose of analysis, these steps can be divided into 6 stages as follows.

Stage 1. Network and Security Capability Discovery

This stage consists of messages numbered (1) to (3). The AP (Access Point) either periodically broadcasts its security capabilities, indicated by RSN IE (Robust Security Network Information Element), in a specific channel through the Beacon frame; or responds to a station's Probe Request through a Probe Response frame. A wireless station may discover available access points and corresponding security capabilities by either passively monitoring the Beacon frames or actively probing every channel.

Stage 2. 802.11 Authentication and Association

This stage consists of messages numbered (4) to (7). The station chooses one AP from the list of available APs, and tries to authenticate and associate with that AP. Note that 802.11 Open System Authentication is included only for backward compatibility, and a station should indicate its security capabilities in the Association Request. After this stage, the station and the AP are in authenticated and associated state. However, the authentication achieved so far is weak, and will be supplemented by further steps. At the end of this stage, the 802.1X ports remain blocked and no data packets can be exchanged.

Stage 3. EAP/802.1X/RADIUS Authentication

This stage consists of messages numbered (8) to (14). The supplicant and the authentication server execute a mutual authentication protocol (de facto EAP-TLS [2]), with the authenticator acting as a relay. After this stage, the supplicant and the authentication server have authenticated each other and generated some common secret, called the Master Session Key (MSK). The supplicant uses the MSK to derive a Pairwise Master Key (PMK); The AAA key material on the server side is securely transferred to the authenticator, indicated by message (15). This allows the authenticator to derive the same PMK. This stage might be skipped if the supplicant and the authenticator are configured using a static Pre-Shared Key (PSK) as the PMK, or when a cached PMK is used during a Re-association.

Stage 4. 4-Way Handshake

This stage consists of messages numbered (16) to (19). Regardless of whether the PMK is derived from *Stage 3*, configured using a PSK, or reused from a cached PMK, the 4-Way Handshake must be executed for a successful RSNA establishment. The supplicant and authenticator use this handshake to confirm the existence of the PMK, verify the selection of the cipher suite, and derive a fresh Pairwise Transient Key (PTK) for the following data session. Simultaneously, the authenticator might also distribute a Group Transient Key (GTK) in message (18). After this stage, a fresh PTK (and maybe GTK) is shared between the authenticator and the supplicant; the 802.1X ports are unblocked for data packets.

Stage 5. Group Key Handshake

This stage consists of messages numbered (20) and (21). In case of multicast applications, the authenticator will generate a fresh GTK and distribute this GTK to the supplicants. These handshakes might not be present if the fresh GTK has been distributed in *Stage 4*; this stage may be repeated multiple times using the same PMK.

Stage 6. Secure Data Communication

This stage is indicated by (22). Using the PTK (or GTK) and the negotiated cipher suite from above stages, the supplicant and the authenticator can exchange protected data packets using data confidentiality protocols.

Through these handshakes, the supplicant and the authenticator mutually authenticate each other and establish a secure session for data transmissions.

4.2 RSNA Security Analysis

Based on the complete RSNA establishment procedure, we will analyze the security of 802.11i considering each possible threat separately. Since the management frames are not protected in a WLAN, an adversary is capable of interfering with *Stages 1* and *2* of the RSNA establishment. More specifically, *Stages 1* and *2* are vulnerable to *Threats 1*, *2*, *3*, and *4*. An adversary can send spoofed security capabilities and topological views of the network to a supplicant on behalf of an authenticator. Once this occurs, the supplicant will be forced to use inappropriate security parameters to communicate with the legitimate authenticator, or

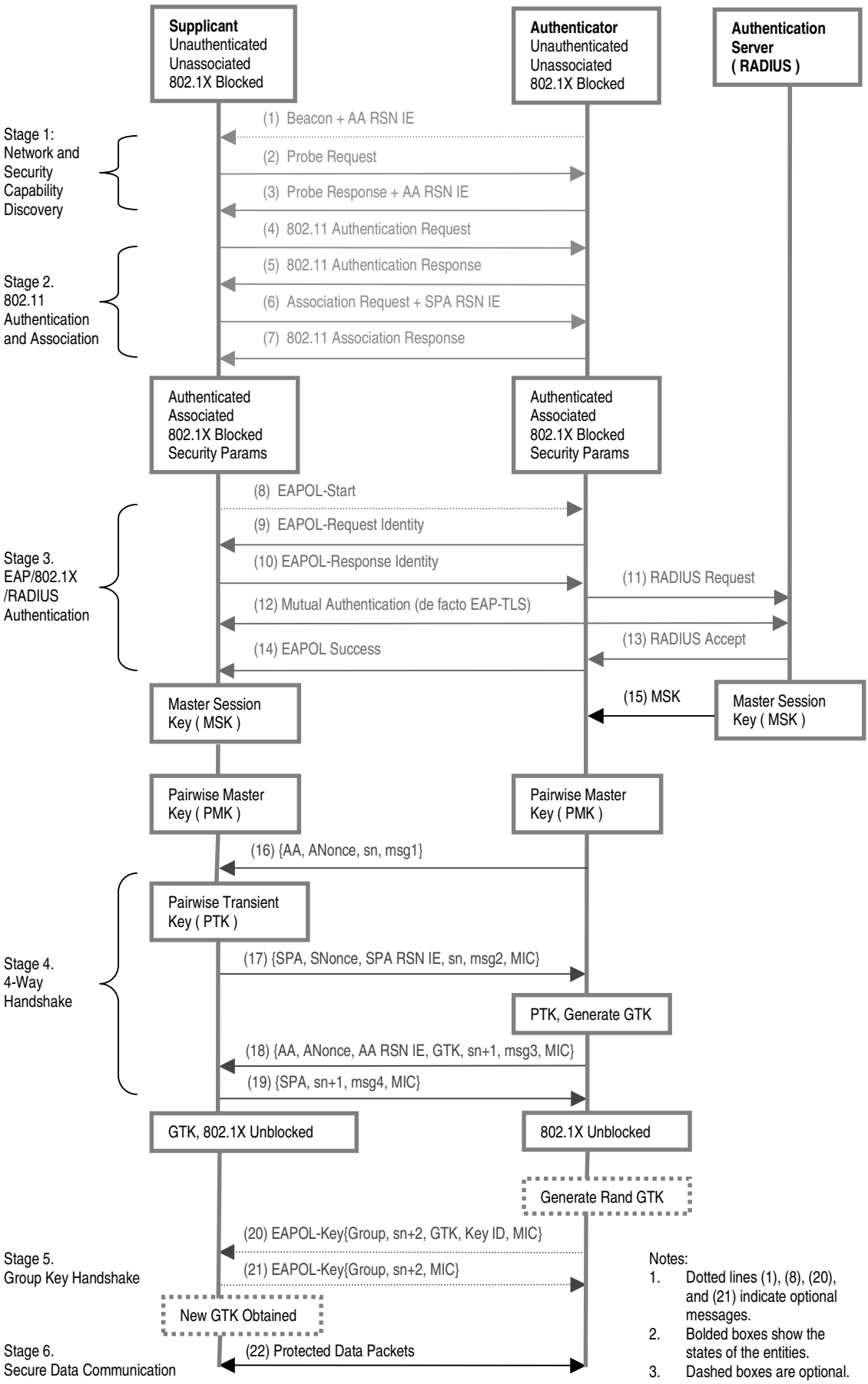

Figure 1. RSNA Establishment Procedures

associate with a malicious AP. Alternatively, an adversary can also forge Association Requests to the authenticator with possibly weak security capabilities, which might cause problems if no further protections are adopted. Fortunately, these threats are eliminated in *Stage 3* if a strong mutual authentication is implemented. The authentication mechanism (e.g., EAP-TLS) should prevent an adversary from forging, modifying, and replaying authentication packets, eliminating *Threats 1*, *2*, and *3*. In addition, since credentials other than MAC addresses must be provided for successful mutual authentication, *Threat 4* is not possible. After the peers have authenticated each other and exchanged some secret after *Stage 3*, the subsequent handshakes are also resistant to these threats. In the case that *Stage 3* is omitted because a PSK or a cached PMK is used, the peers can authenticate each other by verifying possession of the shared key in *Stage 4*, again preventing *Threats 1*, *2*, *3*, and *4*. In addition, *Stage 4* can also verify the security capability negotiations.

Threat 5 may exist even if a strong authentication mechanism is implemented. After a legitimate station has completed a successful authentication, the adversary could disconnect a station by forging Deauthentication or Disassociation messages, and resume the session with the AP on behalf of the legitimate station. There are two possibilities to consider. First, if the session allows the adversary only to accept packets, this appears to be just eavesdropping, which is prevented by data confidentiality mechanisms. Second, if the session requires the adversary for interaction, the adversary will need to obtain the authentication information, such as the PTK, in order to generate acceptable traffic. On the whole, *Threat 5* poses no more danger than eavesdropping and DoS attacks on the station.

Threat 6 is possible in a WLAN if no authentication mechanisms are implemented. An adversary could establish two separate connections to the supplicant and the authenticator to construct a MitM attack [25]. First, the adversary forges the Deauthentication frames to disconnect a station from a legitimate AP. Then, the victim station will automatically probe for a new AP after several failed retries, and eventually associate with the malicious AP, maybe on a different channel. At last, the adversary associates with the legitimate AP on behalf of the legitimate station. However, when 802.11i is implemented with a strong mutual authentication mechanism, like EAP-TLS, the adversary might not be able to authenticate itself to the station or the AP because it does not possess appropriate credentials. Of course the adversary can forward credentials between the AP and the station; but since the authentication packets cannot be modified or replayed, the adversary can only act as a relay, which causes no more damage than eavesdropping. However, if the mutual authentication mechanism is not appropriately implemented [6], the adversary will be able to launch a MitM attack and learn the PMK. Though this vulnerability is considered as a weakness of the specific mutual authentication protocol instead of 802.11i, any implementer of 802.11i should consider this problem carefully.

In summary, if the complete RSNA handshakes are performed, the authentication and key management process appear to be secure. However, since the adversary could interfere with *Stage 1* and *2*, it might be able to fool the authenticator and the supplicant, and prevent completion of the RSNA; this is described as a Security Level Rollback Attack in Section 4.3. In addition, some implementations might also allow a reflection attack in *Stage 4*; the details are described in Section 4.4. Furthermore, although we assume the link between the authenticator and the authentication server is secure, an adversary may still be able to discover the shared secret in RADIUS by offline dictionary attacks [1]. When a 256-bit PSK is used as a PMK, this PSK might be derived from a passphrase [30], which makes the PSK vulnerable to dictionary attacks. An implementation should carefully choose a good passphrase or directly use a 256-bit random value to eliminate this vulnerability.

4.3 Security Level Rollback Attack

When Pre-RSNA and RSNA algorithms are both used in a single WLAN, an adversary can launch a Security Level Rollback Attack, avoiding authentication and disclosing the default keys. Some might argue this is not a real vulnerability, because 802.11i explicitly disallows Pre-RSNA algorithms when RSNA is used. However, 802.11i *does* define a Transient Security Network (TSN) supporting both Pre-RSNA and RSNA algorithms, and this situation might naturally appear in a WLAN implementation. In general, new WLAN implementations may try to support Pre-RSNA algorithms in order to support migration to RSNA. In other words, a supplicant might enable accesses to both RSNA and Pre-RSNA capable networks to ensure Internet access under mobility; simultaneously, an authenticator might be configured in a similar way to provide services to various supplicants. This hybrid configuration will degrade the security of the entire system to the lowest level.

Figure 2 shows an attack to roll back the security level. In this figure, the solid lines represent legitimate message exchanges and the dashed lines indicate messages sent by the adversary. In this attack, the adversary impersonates the authenticator, forging the Beacon or Probe Response frames to the supplicant, and indicating that only Pre-RSNA (WEP) is supported. Alternatively, the adversary can impersonate the supplicant, forging the Association Request frame in a similar way. As a result, the supplicant and the authenticator will establish a Pre-RSNA

connection, even though both of them could support RSNA algorithms. Since there is no cipher suite verification in Pre-RSNA, the supplicant and the authenticator will not be able to detect the forgery and confirm the cipher suites. Even worse, the adversary is able to disclose the default keys by exploiting the weakness of WEP, which then completely undermines the security. This attack is practically feasible because the adversary could either perform as a MitM or forge the beginning management frames in a timely way, that is, Beacon or Probe Response frame to the supplicant and Association Request frame to the authenticator.

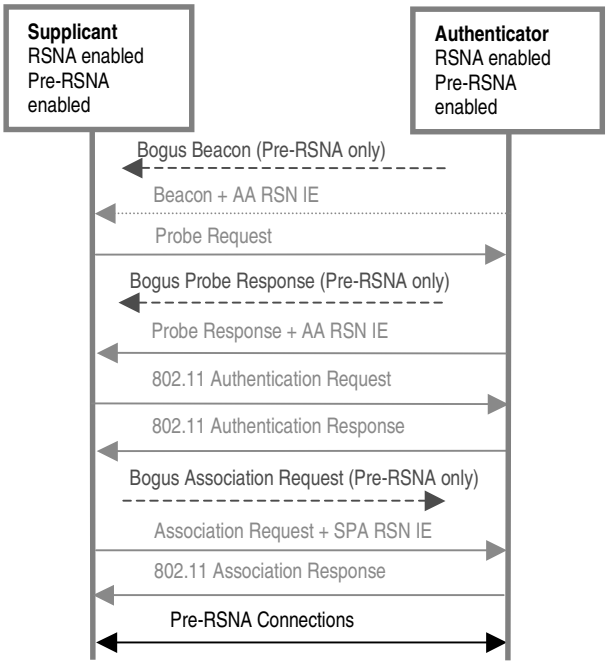

Figure 2. Security Level Rollback Attack

The solution is not complicated. In the simplest approach, both the authenticator and the supplicant could allow only RSNA connections. However, this is only acceptable when security is a strict requirement for the whole system. In most scenarios, TSN might be a better choice to provide services to more supplicants. Therefore, the supplicant and the authenticator could allow both Pre-RSNA and RSNA connections, but deploy appropriate policies on the choice of the security level. Specifically, the supplicant should decide whether it wants a more confidential connection (using RSNA), or it wants more availability of Internet access (using Pre-RSNA). In any event, the supplicant should have a chance to deny the Pre-RSNA algorithms, prior to initiating a connection, either manually or through some form of policy configuration. The authenticator could limit Pre-RSNA connections to only insensitive data. While this policy might cause some inconvenience, it may be worth the security it provides. It is absolutely unreliable to allow the devices to choose a security level transparently, because the authenticator and supplicant have no knowledge of the authenticity of *Stages 1* and *2*.

4.4 Reflection Attack

In *Stage 4*, the 4-Way Handshake uses symmetric cryptography to protect the integrity of the messages. Since the authenticator and the supplicant both know the shared PMK, they are the only two parties that are able to calculate correct MICs and compose valid messages. This fact supports authentication. However, if a device is implemented to play the role of both the authenticator and the supplicant under the same PMK, an adversary can launch a common reflection attack to this device, as shown in Figure 3. When the device initializes a 4-Way Handshake as an authenticator, the adversary will initialize another 4-Way Handshake, with the same parameters but with the victim device acting as the intended supplicant. Once the victim device is fooled to compute messages as a supplicant, the adversary could use these messages as valid responses to the 4-Way Handshake initialized by the victim.

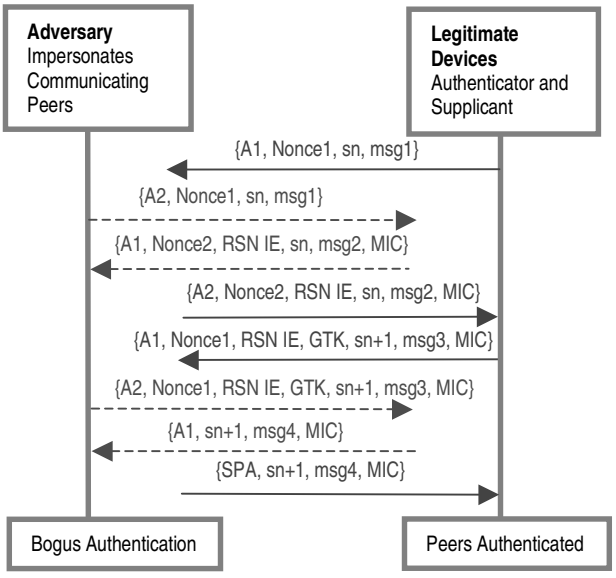

Figure 3. Reflection Attack on the 4-Way Handshake

Naturally this scenario will not appear in an infrastructure network, because a legitimate device will never implement the role of both the authenticator and the

supplicant. However, in ad hoc networks, a possible use of 802.11i could allow each device to serve both roles to distribute their own GTKs. This makes a reflection attack possible. Some might argue that this is not a real vulnerability because the adversary could not decrypt the following data packets without the appropriate key materials (like in Session Hijacking of *Threat 5*). However, it is still valuable to point out the problem because the attack violates mutual authentication; moreover, sometimes the adversary can store the encrypted data for further analysis. In order to eliminate this, the implementer should either limit a device to only one role, or require separate roles to have different PMKs.

5 Availability

IEEE 802.11i appears not to emphasize availability as a primary objective, leaving many DoS vulnerabilities even if the strongest data confidentiality and authentication protocols are used. Prior research has found numerous DoS attacks on a WLAN from the Physical Layer to the Application Layer [4, 7, 8, 12, 17, 24, 32, 42] (See Appendix). Compared to DoS attacks in the Physical Layer, DoS vulnerabilities in 802.11i appear to be more severe for several reasons. First, an adversary can launch an 802.11i attack much more easily than a physical layer attack, with only moderate equipment. Second, it is much more difficult for a network administrator to detect and locate these attacks. Furthermore, layer abstraction is a very important concept in networks, requiring each layer to provide independent functionality separately. Therefore, it is appropriate for 802.11i to resist DoS attacks. Moreover, a more robust 802.11i could help migration to other Physical Layer specifications in the future, which might be secure against DoS attacks. Therefore, it is valuable to strengthen 802.11i against DoS vulnerabilities.

Section 5.1 reviews known DoS attacks, analyzes their effects on 802.11i, and proposes corresponding defenses. Section 5.2 discusses the practicality of the DoS attack on the Michael algorithm countermeasure in TKIP. Section 5.3 describes the RSN IE Poisoning attack and proposes a feasible repair with minor modifications to the algorithm. Section 5.4 explains a DoS attack on *Message 1* of the 4-Way Handshake. Section 5.5 explores the efficiency of different failure-recovery strategies. Section 5.6 proposes an improved version of 802.11i to address all of these vulnerabilities.

5.1 Known DoS Attacks and Defenses

Since the management frames and control frames are unprotected in a WLAN, an adversary can easily forge these frames to launch a DoS attack. Among the management frame attacks, the most efficient attack is to forge and repeatedly send Deauthentication or Disassociation frames. In control frames, the most severe problem lies in the virtual carrier-sense mechanism like RTS frame [8, 12]. Unfortunately, these attacks persist even if 802.11i is used to protect the WLAN. It might be possible to adopt a Central Manager to handle these frames specifically and identify the forged frames by their abnormal behavior [14]. However, this requires extra functionality in the authentication server, and the server needs to keep the state of all supplicants. This increases the workload of the server and might be infeasible. Another approach is to respond to Deauthentication and Disassociation frames by restarting a 4-Way Handshake, with the result of the 4-Way Handshake indicating whether these frames are forged [29]. This method could restrict the impact of forged Disassociation and Deauthentication to the 802.11 MAC. However, this does not prevent the attack because periodically forcing a 4-Way Handshake could be an effective DoS attack. Based on these considerations, authenticating management frames appears to be a better approach. This is also described in Section 5.6 as an improvement to 802.11i. On the other hand, authenticating control frames might be inefficient and add too much overhead because the control frames could appear frequently; it might be better to handle forged control frames by checking the validity of the virtual carrier-sense according to the knowledge of the specific frames.

There are several DoS attacks that exploit the unprotected EAP messages in 802.1X authentication. Specifically, an adversary can forge EAPOL-Start messages repeatedly to prevent the 802.1X authentication from succeeding, forge EAPOL-Success message to maliciously bring up the 802.1X data port in the supplicant without authentication, and forge EAPOL-Failure message and EAPOL-Logoff message to disconnect the supplicant. Fortunately, these vulnerabilities can be eliminated in 802.11i by simply ignoring these messages. This does not affect the functionality and logic of the protocol. The outcome of the subsequent 4-Way Handshake could take the role of EAPOL-Success and EAPOL-Failure to indicate the authentication result; EAPOL-Logoff could be replaced by Deauthentication to disconnect a client; and EAPOL-Start is not necessary for the protocol.

An adversary can also launch a DoS attack on the AP by flooding forged Association Request frames. This will exhaust the EAP Identifier space, which is only 8 bits long (0-255). This vulnerability can be addressed by careful consideration during implementation. Since an EAP Identifier is only required to be unique within a single 802.11 association, it is not necessary for the AP to deny new connection requests when the EAP Identifier

space has been exhausted. Particularly, the AP can adopt a separate EAP Identifier counter for each association.

5.2 Michael Algorithm Countermeasure

In addition to these known DoS attacks, the countermeasure associated with the Michael algorithm (discussed in the 802.11i standard) is also vulnerable to DoS attacks. As a data confidentiality protocol, TKIP adopts the Michael algorithm to provide MIC protection for every MSDU (MAC Service Data Unit). The TKIP MPDU (MAC Protocol Data Unit) format is shown in Figure 4. The Michael algorithm is designed to provide only 20 bits (or possible slightly more) of security due to the limited computation power in legacy devices. This means it is possible for an adversary to construct a successful forgery after 2^{19} attempts. Therefore, TKIP implements the following countermeasures to limit the rate of the forgery attempts from an adversary. The first Michael MIC failure is logged as a security-relevant matter. Once two failures are detected within 60 seconds, the transmission and reception will cease for 60 seconds. Furthermore, the authenticator could re-key or deauthenticate the supplicant; the supplicant should send out a Michael MIC Failure Report frame and deauthenticate itself afterwards.

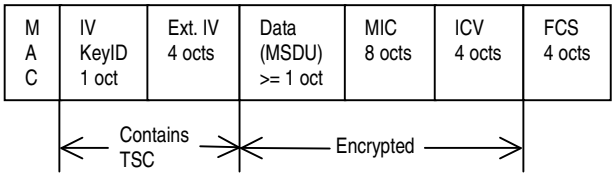

Figure 4. TKIP MPDU Format

As shown in the following calculation, the countermeasure ensures that a successful forgery could occur only every half year, which makes the forgery practically useless. In an 802.11b network, an adversary could send out approximately 2^{12} messages per second. Therefore, the adversary is able to make a successful forgery in about 2 minutes (approximately 2^7 seconds) without implementing the countermeasure. However, if countermeasures are deployed to limit the rate, for example, 2 forgery attempts per minute, the attacker is limited to make one successful forgery every 6 months (approximately 2^{18} minutes). Unfortunately, the countermeasure leaves an obvious DoS vulnerability: an adversary can send out unsuccessful forgery attempts to cause two Michael MIC failures and shutdown a connection. In order to prevent this DoS attack, the protocol checks the FCS (Frame Check Sequence), ICV (Integrity Check Value), TSC (TKIP Sequence Counter) and MIC sequentially. A MIC failure is only logged when the frame has been received with correct FCS, ICV, TSC but an invalid MIC. Checking FCS and ICV can detect packet errors caused by noise, while checking TSC can detect replayed packets. Moreover, if the adversary modifies the TSC, the per-packet key will be modified simultaneously, which causes packet decryption to fail before a log of MIC failure. Hence, checking FCS, ICV, TSC and MIC in a strict order could make DoS attacks more difficult. However, an adversary is still able to launch such an attack through interception. Furthermore, the inappropriate TSC update strategy might make this attack more convenient.

In *Threat 3*, an adversary is able to intercept a message before the receiver hears it. Through this approach, the adversary can obtain a packet with a valid TSC value. Keeping the TSC field unchanged, the adversary is capable of modifying some bits of the packet and updating the corresponding FCS and ICV fields to make them consistent, due to weakness in the ICV algorithm. Then the adversary has obtained the desired packet because the packet can pass the check for FCS, ICV, and TSC, but with an invalid MIC. By sending out this packet, the adversary could force a Michael MIC failure in the receiver side, and eventually launch a DoS attack. Even worse, since 802.11i suggests updating the TSC until an MSDU passes the Michael MIC check, if the receiver implementation resumes communication after 60 seconds without re-keying, the adversary can simply forward one modified message repeatedly because the TSC is still valid after the first failure. On the other hand, if the receiver re-keys the system or deauthenticates, the adversary will have enough time to construct the next modified packet and block the communication. Of course this is not that easy because *Threat 3* requires considerable work from the adversary. However, it is quite practical because the time required for re-keying and re-authentication is sufficient for an adversary to construct the packet.

This attack can be mitigated by careful consideration in the implementation. First, re-keying and deauthentication are not necessary when availability is an objective. The authenticator and the supplicant should just cease for 60 seconds, and then resume communication. Second, the TSC should be updated once a packet passes the check of FCS, ICV, and TSC, even if the Michael MIC failure occurs. Note that in this case the retransmitted packets must use a fresh TSC. These modifications will make the DoS attack more difficult; however, they do not eliminate the vulnerability. Fortunately, this attack disappears with TKIP when CCMP is implemented for data confidentiality.

5.3 RSN IE Poisoning

Another possible category of DoS attacks on 802.11i involves the RSN IE (RSN Information Element) verification mechanism. As shown in Figure 5, RSN IE contains authentication and pairwise key cipher suite selectors, a single group key cipher suite selector, an RSN Capabilities field, the PMKID (Pairwise Master Key Identifier) count, and the PMKID list. The authenticator should insert the supported RSN IEs in the Beacon and Probe Response and the supplicant should insert its chosen RSN IE in the (Re)Association Request. The authenticator and the supplicant use the negotiated security suites to perform the authentication and key management protocol, and use the negotiated cipher suites to encrypt data communications. In order to confirm the authenticity of the RSN IEs, the supplicant is required to include the same RSN IE in *Message 2* of the 4-Way Handshake as in (Re)Association Request. The authenticator is also required to include the same RSN IE in *Message 3* of the 4-Way Handshake as in the Beacon or Probe Response. After receiving a *Message 2*, the authenticator will bit-wise compare the RSN IE in the message with the one it receives in the (Re)Association Request from the supplicant, in order to confirm that they are exactly the same. The supplicant will bit-wise compare the RSN IE in *Message 3* with the one it receives in Beacon or Probe Response. If the RSN IEs are not exactly the same, the supplicant and the authenticator will deauthenticate each other and a security error should be logged. This confirmation process prevents an adversary from tricking the supplicant and the authenticator into using a weaker security scheme by forging the RSN IE negotiations. However, as a result, it is vulnerable to DoS attacks.

Element ID (1 octet)	Length (1 octet)
Version (2 octets)	
Group Key Cipher Suite (4 octets)	
Pairwise Key Cipher Suite Count (2 octets)	
Pairwise Key Cipher Suite List (4m octets)	
Authentication and Key Management Suite Count (2 octets)	
Authentication and Key Management Suite List (4n octets)	
RSN Capabilities (2 octets)	
PMKID Count (2 octets)	
PMKID List (16s octets)	

Figure 5. RSN Information Element Format

In *Message 2* of the 4-Way Handshake, the authenticator verifies the MIC before the RSN IE, which is the correct order; but in *Message 3*, the supplicant checks the RSN IE before the MIC verification, and aborts if the RSN IE is unmatched. Consequently, an adversary can easily modify the RSN IE in *Message 3* to cause the handshake to fail. This vulnerability could almost be considered a typo in the 802.11i documentation. However, even if the check order is correct, there is another fundamental attack to cause the RSN IE confirmation process to fail, which is shown in Figure 6.

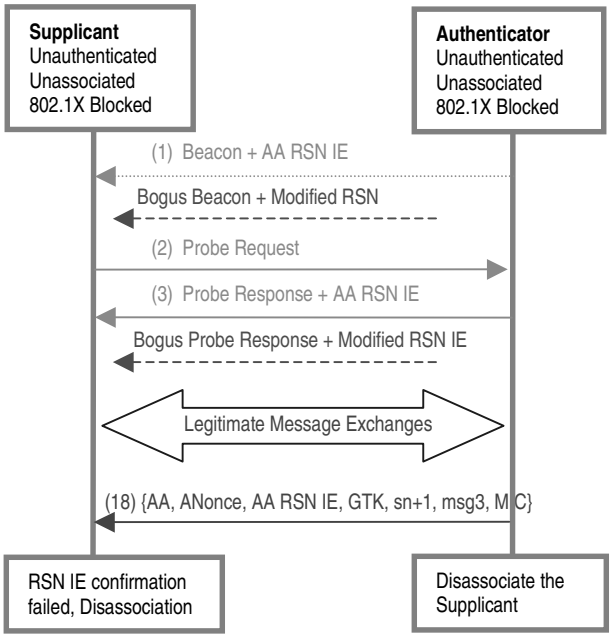

Figure 6. RSN IE Poisoning

An adversary can easily eavesdrop on the Beacon frames of a legitimate authenticator, modify several bits in the frame that are "insignificant", where "insignificant" means that, the modification of these bits will not effect the validity of the frame and the selection of the authentication cipher suites. For example, the Reserved bits and the Replay Counter bits in the RSN Capabilities field are "insignificant". The adversary then broadcasts this forged Beacon to poison the knowledge of RSN IEs in the supplicants. Because this forged Beacon only modifies "insignificant" bits, the supplicant and the authenticator are still able to continue the authentication and key management using the effective security suites. However, the 4-Way Handshake will never succeed because the RSN IE confirmation will fail. Accordingly, when the supplicant uses an active scan instead of a passive one, the adversary can forge a Probe Response

with the modified RSN IE, which requires the adversary to interfere with the handshake in a more timely way. The adversary can also forge a (Re)Association Request with modified RSN IE to poison the knowledge of the authenticator, but this approach is less efficient.

Based on analysis above, an adversary can always launch a DoS attack by RSN IE poisoning. This attack is different from the Security Level Rollback Attack, because the adversary does not aim to establish a successful connection; it simply blocks the protocol execution. This attack is considered to be harmful because it is quite easy for an adversary to implement and it will affect all the supplicants simultaneously when the forged Beacon is allowed. Furthermore, because the supplicant and the authenticator are unaware of the RSN IE poisoning, they might continue to exchange considerable number of messages, e.g., messages (2) to (18) in Figure 1, until the 4-Way Handshake fails. In other words, the legitimate entities do substantial work, while the adversary is able to successfully interfere with little work. This wastes the resources of the authenticator and the supplicant; moreover, the adversary will have more time to periodically repeat its attacks.

This weakness is exploitable for three reasons. First, the management frames like Beacon, Probe Response, and (Re)Association Request are not protected. Second, there are a number of message exchanges between the RSN IE negotiation and confirmation, which consume resources and leave more time for the adversary. Third, the bit-wise comparison in the 4-Way Handshake might be unnecessarily strict to confirm RSN IE. The vulnerability may be addressed accordingly. Authenticating the management frames is a good approach. However, in some scenarios it might not be considered acceptable to authenticate the Beacon and Probe Response frames because the authenticator and the supplicant share no secret at the beginning. Therefore, an acceptable approach is to do the RSN IE confirmation as soon as possible to avoid wasting message exchanges and make the attack less disruptive. Considering that the authentication server may know the security parameters that the authenticator supports, it can help to confirm the RSN IE much earlier, such as in the 802.1X authentication. However, this would require an EAP method supporting cipher suite negotiations (e.g., EAP-TLS), and considerable modifications to the existing standard might be necessary.

Alternatively, this attack can be mitigated by loosening the condition of the RSN IE confirmation. In other words, the authenticator and the supplicant can ignore the differences of the "insignificant" bits in the corresponding RSN IEs, while keeping the negotiation secure. Actually in a RSN IE, only the authentication and key management suite selector is essential for the subsequent handshakes, because the authenticator and the supplicant are always able to securely negotiate the encryption cipher suites after they finish the authentication and share some secret. If an adversary does not change the authentication and key management suite selector, the RSN IE could be accepted because the correct authentication has been executed. Afterwards, the authenticator and the supplicant can use the authenticated RSN IE in the 4-Way Handshake for the subsequent data encryptions. On the other hand, if the adversary modifies the authentication and key management suite selector, this can be detected at the beginning of the association. The association fails and the supplicant retries quickly without continuing message exchanges. In the worst case, this modification can be prevented in the 4-Way Handshake.

5.4 4-Way Handshake Blocking

The 4-Way Handshake is an essential component of the RSNA establishment. Its purpose is to confirm the possession of the shared PMK (Pairwise Master Key) in the authenticator and the supplicant, and derive a fresh PTK (Pairwise Transient Key) for subsequent data communication. In the handshake, the authenticator and the supplicant generate their own nonces and send them to each other. The PTK is derived from the shared PMK, the nonces, and the MAC address of the peers. *Message 1* and *3* carry the nonce generated by the authenticator; *Message 2* carries the nonce generated by the supplicant, and *Message 4* is an acknowledgment to indicate the handshake is successfully completed. While *Message 2*, *3*, and *4* are authenticated by the fresh PTK, *Message 1* is unprotected. In order to prevent an adversary from affecting the PTK through forging *Message 1*, 802.11i adopts a Temporary PTK (TPTK) to store the newly generated PTK until *Message 3* is verified. However, this approach does not prevent DoS attacks on *Message 1*.

The supplicant must accept all *Message 1*s it receives in order to ensure that the handshake can complete in case of packet loss and retransmission. This allows an adversary to cause PTK inconsistency between the supplicant and the authenticator by sending a forged *Message 1* with a different nonce value between the legitimate *Message 1* and *Message 3*. In order to accommodate the forged *Message 1*s, the supplicant has to store all the responding nonces and the derived PTKs. Only after a *Message 3* with a valid MIC is received, the supplicant can install the corresponding correct PTK for data communications and discard all others. Obviously, an adversary is able to launch a memory DoS attack by sending out numerous forged *Message 1*s, as shown in Figure 7. This attack is serious because it is simple for the adversary to perform, and a successful attack will cancel all efforts in the previous authentication process.

There are several approaches to address this vulnerability. First, the supplicant can implement a queue

with a random-drop policy. This method helps to mitigate the vulnerability, but does not eliminate it. Second, *Message 1* can be authenticated to defend against this attack, because the authenticator and the supplicant have already finished the authentication and shared some secret. However, this requires some modifications to the message format; moreover, the authenticator needs to include a monotonically increasing sequence number in each *Message 1* in order to prevent replays. Third, the supplicant can inherently eliminate this attack by re-using the same nonce for all received *Message 1*s until a 4-Way Handshake completes successfully. The supplicant only stores one nonce, calculates a PTK based on this stored nonce and the nonce in the received message, then verify the MIC. This approach only requires minor modifications on the algorithm in the supplicant side; the supplicant need store no more than one nonce, eliminating the possible memory exhaustion. However, the supplicant will consume more computation power because it needs to calculate the same PTK twice for the received *Message 1* and *3*, given that the received nonces and derived PTKs are not stored. The supplicant has to make a decision on the tradeoff between the memory and the CPU consumption. As a combined solution, the supplicant can re-use the same nonce for all *Message 1*s to eliminate the memory DoS vulnerability, and store one entry of the derived PTK to improve the performance. A more detailed discussion appears in our previous paper [19].

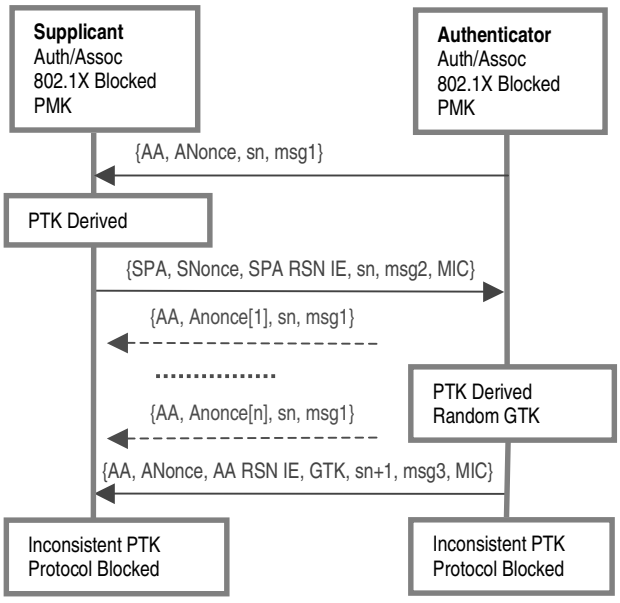

Figure 7. 4-Way Handshake Blocking

5.5 Failure Recovery

The 802.11i design decisions appear to emphasize other security objectives over availability. Once security-related events or timeouts occur, the specification always suggests Deauthentication or Disassociation. This mechanism reduces information leakage and prevents further attacks, but it also increases the possibility of potential DoS attacks. Therefore, different failure recovery schemes should be specified when DoS attacks are considered significant. A better failure recovery does not inherently prevent the DoS attacks; however, it will make the protocol more efficient, and cause more difficulties for an adversary trying to launch an attack. For example, in the Group Key Handshake, a timeout will cause the authenticator to disassociate or deauthenticate the supplicant. The supplicant then reassociates with the same AP or scans the channels for another AP, which is quite time consuming. Moreover, since the authenticator can only install the GTK after all the supplicants have done so, the reassociation delay of one supplicant will affect all others. Alternatively, if the supplicant and the authenticator just retry the Group Key Handshake or the 4-Way Handshake, they could resume the connection more quickly. On the other hand, if the Group Key Handshake timeout is due to the unavailability of the supplicant, e.g., the supplicant moving out of the range of the authenticator, retrying the Group Key Handshake or the 4-Way Handshake wastes more time compared to directly disassociating from the current AP and reassociating with another AP. This is a tradeoff the protocol implementer must make according to the networking environment.

This tradeoff could be generalized to any trust relationship discovery process in a malicious environment. Assuming a protocol instance is running between a pair of stations; at some point the instance fails due to some active attacks. If the protocol restarts from the beginning, this protocol will be vulnerable to a so-called “defensive DoS attack”. That means, if an adversary has the capability of causing the protocol to fail at some point, it can launch a DoS attack by periodically doing that. Since each time the legitimate peers need to waste some message exchanges to recover, the adversary will have more time to construct the attack. On the other hand, if the protocol recovers from the nearest point, the adversary might not have sufficient time to construct the “defensive DoS attack”; however, the protocol could be vulnerable to another type of DoS attack, so-called “captured DoS attack”. That means, if the adversary has the capability of propelling the protocol to some point with a legitimate user, it is possible for the adversary to capture the user for more time before the user could find a legitimate peer. Hence, the selection of the recovery point depends on the

assumptions of the networking scenario and the capability of the adversary.

Specifically, in 802.11i it might be reasonable to assume that it is difficult to forge an 802.1X authentication. Therefore, once the handshakes between the supplicant and the authenticator have proceeded beyond the 802.1X authentication, the recovery point could be chosen to the nearest point to improve the efficiency of the protocols, because both entities involving in the communications should be legitimate. On the other hand, if a supplicant and an authenticator do not finish an 802.1X authentication, it might be better to restart the protocol from the beginning. Of course, in a high mobility environment, a failure might occur more frequently because one entity is unavailable; thus, retrying the nearest point might waste more time. However, since the channel scanning time is significantly larger than the protocol execution time, retrying the nearest point will not increase the delay too much compared to the total recovery delay.

5.6 Improved 802.11i

Based on the above analysis, we propose an improved version of 802.11i to make more DoS resistant. Note that due to the physical vulnerability of wireless links, DoS attacks always exist through frequency jamming, network jamming, or other exploits. However, we improve the 802.11i protocol to achieve DoS resistance in the Link Layer. A flow chart of the improved 802.11i is shown in Figure 8.

(1) In order to eliminate the DoS attacks by Association Request flooding, it is better to perform authentication before association; this idea is originally developed in [16]. Specifically, in 802.11i, it is possible to replace the 802.11 entity authentication with 802.1X authentication; a secure association can occur after the 802.1X authentication. This improvement might require necessary modifications to the existing 802.1X implementations; however, the advantages appear worth it.

(2) The authentication and key management suite negotiation should be verified as soon as possible. Otherwise, the subsequent handshakes might waste time because the negotiated security suites could be forged by an adversary. Specifically, when an 802.1X authentication is adopted, the peers can verify their security parameters during the 802.1X authentication; when a PSK or a cached PMK is used, the peers can verify the information through a secure (Re)Association.

(3) The management frames should be authenticated to improve security, and some control frames can also be authenticated if necessary. Authenticating these frames should be performed as soon as possible. Once an authentication process is completed successfully, the derived common secret could be used to authenticate the subsequent management frames, especially the (Re)Association Request/Response frame. Through this approach the vulnerabilities of most of the management frames are eliminated, except the Beacon and Probe Request/Response frames, which cannot be authenticated because the common secret is unavailable. Of course, key asynchrony and state abnormality should be carefully considered to avoid blocking the protocols.

(4) An appropriate failure-recovery scheme is implemented to improve the efficiency of the overall protocol. In the infrastructure networks with low mobility, assuming the 802.1X provides strong authentication, the protocol will recover from the nearest point if 802.1X has been completed successfully, while the protocol will recover from the beginning if an 802.1X has not yet finished.

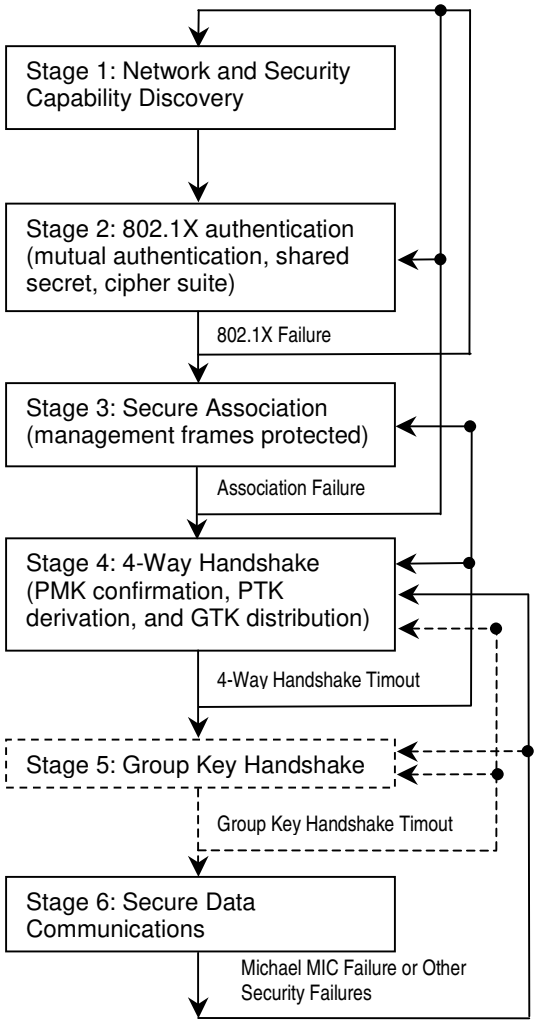

Figure 8. A Flow Chart of the Improved 802.11i
The original Stage 2 and 3 in Figure 1 are switched here; Possible recovery points for failures at different stages are indicated by arrows; Dashed lines indicate Stage 5 is optional.

Through these improvements, the 802.11i vulnerabilities discussed in this paper could be eliminated. Note that while the improvements result in DoS robustness, they may require more modifications to the existing implementations.

6 Conclusion

This paper analyzes the IEEE 802.11i protocols for data confidentiality, integrity, mutual authentication, and availability. Under the threats we consider, 802.11i appears to provide effective data confidentiality and integrity when CCMP is used. This requires a legacy WEP user to upgrade the hardware. Furthermore, 802.11i adopts a RSNA establishment procedure for mutual authentication and key management, which appears to be satisfactorily secure. However, several vulnerabilities might arise in a real implementation. If the mutual authentication mechanism is not implemented appropriately, there might be a Man-in-the-Middle attack that reveals the shared secret. If a passphrase is used to generate a 256-bit PSK, an adversary might be able to find the passphrase through dictionary attacks. An adversary is also able to discover the shared RADIUS secret through dictionary attacks. Furthermore, if Pre-RSNA and RSNA algorithms are implemented in a system simultaneously without careful considerations, an adversary is able to perform a Security Level Rollback Attack to force the communicating peers to use WEP, which is completely insecure. Moreover, if a wireless device is implemented to play the role of both the authenticator and the supplicant, an adversary can construct a reflection attack on the 4-Way Handshake. This scenario naturally appears in ad hoc networks.

Availability is another important security property in wireless networks. Since availability is not the primary design goal, 802.11i appears vulnerable to DoS attacks even if RSNA is implemented. We review the known DoS attacks and propose solutions appropriate to 802.11i. It appears that a better way to eliminate management frame vulnerabilities is to authenticate them. Furthermore, we find and analyze some new DoS attacks arising with 802.11i. First, we analyze the practicality of the DoS attack on the Michael algorithm countermeasures. Here, eliminating re-keying and updating the TSC carefully appear to provide significant improvements. Second, we describe a new DoS attack through RSN IE poisoning. Several repairs are discussed. Relaxing the condition for RSN IE verification seems the preferred approach because it only requires minor modifications to the algorithm. Third, a DoS attack on the unprotected *Message 1* of the 4-Way Handshake is described and the corresponding defenses are proposed. Fourth, tradeoffs in the failure-recovery strategy are discussed and an efficient failure recovery for 802.11i is proposed, based on the characteristics of wireless networks. Finally, we integrate all the improvements to construct a DoS resistant variant of the 802.11i protocols.

Acknowledgments

We thank Jesse Walker for his informative comments on the 802.11i protocols.

References

[1] B. Aboba, and A. Palekar. IEEE 802.1X and RADIUS security. Submissions to IEEE 802.11 TGi, November, 2001.

[2] B. Aboba, and D. Simon. PPP EAP TLS authentication protocol. *RFC 2716*, October, 1999.

[3] B. Aboba. Pros and Cons of upper layer network access. IEEE documents 802.11-00/382, November, 2000.

[4] W. A. Arbaugh, N. Shankar, and J. Wang. Your 802.11 Network has no Clothes. In *Proceedings of the First IEEE International Conference on Wireless LANs and Home Networks*, pages 131-144, December, 2001.

[5] W. A. Arbaugh. An inductive Chosen Plaintext Attack against WEP/WEP2. Presentations to IEEE 802.11 TGi, May, 2001.

[6] N. Asokan, V. Niemi, and K. Nyberg. Man-in-the-Middle in tunneled authentication protocols. Technical Report 2002/163, IACR ePrint archive, October, 2002. http://eprint.iacr.org/2002/163/.

[7] AusCERT AA-2004.02. Denial of Service vulnerability in IEEE 802.11 wireless devices. May 13, 2004. http://www.auscert.org.au/render.html?it=4091.

[8] J. Bellardo, and S. Savage. 802.11 Denial-of-Service attacks: real vulnerabilities and practical solutions. In *Proceedings of the USENIX Security Symposium*, pages 15-28, August, 2003.

[9] L. Blunk, J. Vollbrecht, B. Aboba, J. Carlson, and H. Levkowetz. Extensible Authentication Protocol (EAP). Internet Draft draft-ietf-eap-rfc2284bis-06.txt, September 29, 2003.

[10] N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications: the insecurity of 802.11. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, Rome, Italy, July, 2001.

[11] N. Cam-Winget, R. Housley, D. Wagner, and J. Walker. Security flaws in 802.11 data link protocols. SPECIAL ISSUE: Wireless networking security, Communications of the ACM, Volume 46, Issue 5, pages 35-39, May, 2003.

[12] D. Chen, J. Deng, and P. K. Varshney. Protecting wireless networks against a Denial of Service attack based on virtual jamming. In *Poster Session of MobiCom2003*, San Diego, CA, September, 2003.

[13] Cisco Systems. Cisco Aironet Response to University of Maryland's Paper, "An Initial Security Analysis of the IEEE 802.1X Standard". August 22, 2002. http://www.cisco.com/warp/public/cc/pd/witc/ao350ap/prodlit/1680_pp.htm.

[14] P. Ding, J. Holliday, and A. Celik. Improving the security of Wireless LANs by managing 802.1X Disassociation. In *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC'04)*, Las Vegas, NV, January, 2004.

[15] J. Duntemann. Wardriving FAQ. April 26, 2003. http://faq.wardrive.net/.

[16] D. B. Faria and D. R. Cheriton. DoS and authentication in wireless public access networks. In *Proceedings of the First ACM Workshop on Wireless Security (WiSe'02)*, Atlanta, Georgia, USA, September, 2002.

[17] B. Fleck and J. Dimov. Wireless access points and ARP poisoning: wireless vulnerabilities that expose the wired network. White paper by Cigital Inc. in 2001. http://www.cigitallabs.com/resources/papers/download/arppoison.pdf.

[18] S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. *Lecture Notes In Computer Science*, Revised Paper from *the 8th Annual International Workshop on Selected Areas in Cryptography*, pages 1-24, 2001.

[19] C. He and J. C. Mitchell. Analysis of the 802.11i 4-Way Handshake. In *Proceedings of the Third ACM International Workshop on Wireless Security (WiSe'04)*, Philadelphia, PA, October, 2004.

[20] IEEE Standard 802.11-1999. Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control and Physical Layer Specifications. 1999.

[21] IEEE P802.11i/D10.0. Medium Access Control (MAC) Security Enhancements, Amendment 6 to IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications. April, 2004.

[22] IEEE Standard 802.1X-2001. IEEE Standard for Local and metropolitan area networks – Port-Based Network Access Control. June, 2001.

[23] J. Jonsson. On the Security of CTR + CBC-MAC. *Lecture Notes In Computer Science*, Revised Paper from *the 9th Annual International Workshop on Selected Areas in Cryptography*, pages 76-93, 2002.

[24] P. Kyasanur and N. H. Vaidya. Detection and handling of MAC layer misbehavior in wireless networks. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN'03)*, San Francisco, CA, June, 2003.

[25] M. Lynn and R. Baird. Advanced 802.11 attack. *Black Hat Briefings*, Las Vegas, NV, July, 2002.

[26] T. Marshall. Antennas enhance WLAN security. October, 2001. http://www.trevormarshall.com/byte_articles/byte1.htm.

[27] A. Mishra and W. A. Arbaugh. An initial security analysis of the IEEE 802.1X standard. Technical Report CS-TR-4328, UMIACS-TR-2002-10, University of Maryland, February, 2002.

[28] V. Moen, H. Raddum, and K. J. Hole. Weakness in the Temporal Key Hash of WPA. *ACM SIGMOBILE Mobile Computing and Communications Review*, Volume 8, Issue 2, pages 76-83, April, 2004.

[29] T. Moore. Validating 802.11 Disassociation and Deauthentication messages. Submission to IEEE P802.11 TGi, September, 2002.

[30] R. Moskowitz. Weakness in Passphrase Choice in WPA Interface. November, 2003. http://wifinetnews.com/archives/002452.html.

[31] National Institute of Standards and Technology. FIPS Pub 197: *Advanced Encryption Standard (AES)*. November 26, 2001.

[32] D. Neoh. GSEC Version 1.4b Option 1, Corporate Wireless LAN: Know the risks and best practices to mitigate them. December, 2003. http://www.securitydocs.com/links/1153.

[33] J. S. Park and D. Dicoi. WLAN security: current and future. *IEEE Internet Computing, Volume 7, No. 5, pages 60-65*. September/October, 2003.

[34] C. Rigney, S. Willens, A. Rubens, and W. Simpson. Remote Authentication Dial In User Service (RADIUS). *RFC 2865*, June, 2000.

[35] P. Rogaway and D. Wagner. A Critique of CCM. Unpublished manuscript. February 2, 2003. http://www.cs.berkeley.edu/~daw/papers/ccm.html.

[36] A. Roy and H. Chang. Physical Layer security of WLANs. http://csiannual.com/classes/f2.pdf.

[37] B. Schneier. Applied Cryptography: Protocols, Algorithms, and Source Code in C. Second Edition, John Wiley and Sons, New York, NY, USA, 1996.

[38] P. Shipley. Open WLANs: the early result of Wardriving. 2001. http://www.dis.org/filez/openlans.pdf.

[39] A. Stubblefield, J. Ioannidis, and A. Rubin. Using the Fluhrer, Mantin, and Shamir attack to break WEP. In *Proceedings of the 2002 Network and Distributed Systems Symposium*, San Diego, CA, February, 2002.

[40] J. R. Walker. Unsafe at any key size; An analysis of the WEP encapsulation. IEEE Document 802.11-00/362, October, 2000.

[41] D. Whiting, R. Housley, and N. Ferguson. Counter with CBC-MAC (CCM). *RFC 3610*, September, 2003.

[42] C. Wullems, K. Tham, J. Smith, and M. Looi. Technical Summary of Denial of Service Attack against IEEE 802.11 DSSS based Wireless LAN's. http://www.isrc.qut.edu.au/wireless/.

Appendix. A Review of the Literature

1. Wireless Security Evolution

In order to provide data confidentiality equivalent to a wired network, the IEEE 802.11 Standard [20] originally defines Wired Equivalent Privacy (WEP). This mechanism adopts RC4, a common stream cipher [37], to encrypt messages with a shared key. This key is concatenated with a 24-bit Initialization Vector (IV) to construct a per-packet RC4 key. In order to provide data integrity, WEP calculates an Integrity Check Value (ICV) over the MSDU (MAC Service Data Unit), which is a common Cyclic Redundancy Checksum (CRC). The frame body, together with the corresponding ICV, is encrypted using the per-packet key. In addition, two authentication mechanisms are defined: the Open System Authentication, which is actually a null authentication, and the Shared Key Authentication, which is a Challenge-Response handshake based on the shared key.

However, numerous researches have shown that none of the data confidentiality, integrity, and authentication could be achieved through above mechanisms. First, the 40-bit shared key is too short for brute-force attacks [10, 33]. Though some vendors might support a longer key (104 bits), it is still possible for an adversary to recover the plaintext traffic because the small IV size and the static shared key result in a high possibility of key stream reuse [10, 40], which trivially defeats any stream cipher. Furthermore, the concatenation of the IV and the shared key has inherent weakness for generating the per-packet RC4 key [18]; an adversary can discover the key by eavesdropping several million packets [39]. Moreover, because ICV is a linear and unkeyed function of the message [10], data integrity cannot be guaranteed; even without any knowledge of the key stream, an adversary is able to arbitrarily modify a packet without detection, or forge a packet with a valid ICV. This weak integrity also enables much easier plaintext recovery, for example, IP redirection, reaction attacks [10], and inductive chosen plaintext attack [5]. Finally, an adversary can trivially spoof the Shared Key Authentication [4, 10] through observing an authentication process of a legitimate station. Additionally, WEP does not implement any mechanism to prevent replay attacks.

Although WEP fails to satisfy any security requirements, it is not practical to anticipate users to completely discard their devices with WEP already implemented. Hence, the Wi-Fi Alliance proposed an interim solution, called Wi-Fi Protected Access (WPA), to ameliorate the vulnerabilities by reusing the legacy hardware. WPA adopts a Temporal Key Integrity Protocol (TKIP) for data confidentiality, which still uses RC4 for data encryption, but includes a key mixing function and an extended IV space to construct unrelated and fresh per-packet keys. WPA also introduces Michael algorithm, a weak keyed Message Integrity Code (MIC), for improved data integrity under the limitation of the computation power available in the devices. Furthermore, in order to detect replayed packets, WPA implements a packet sequencing mechanism by binding a monotonically increasing sequence number to each packet.

In addition, WPA provides two improved authentication mechanisms. In one mechanism, the possession of a Pre-Shared Key (PSK) authenticates the peers; furthermore, a 128-bit encryption key and another distinct 64-bit MIC key can be derived from the PSK. Alternatively, IEEE 802.1X [22] and the Extensible Authentication Protocol (EAP) [9] can be adopted to provide a stronger authentication for each association, and generate a fresh common secret as part of the authentication process; all required keys can be derived from this shared secret afterwards.

TKIP is proposed to address all known vulnerabilities in WEP; it *does* enhance the security in all aspects. However, weakness is predestined since WPA appears due to the limitation of re-using the legacy hardware. Although TKIP key mixing function has stronger security than WEP key scheduling algorithm, it is not so strong as expected. It is possible to find the MIC key given one per-packet key; furthermore, the whole security is broken for the duration of a Temporal Key (TK) given two per-packet keys with the same IV32 [28]. This vulnerability does not mean that TKIP is insecure, but it discloses that parts of TKIP are weak on their own. Furthermore,

Michael algorithm is designed to provide only 20 bits (or possible slightly more) of security in order to minimize the impact on the performance, which means an adversary can construct one successful forgery every 2^{19} packets. Thus, countermeasures are necessarily adopted to limit the rate of the forgery attempts [21]. However, this countermeasure may allow DoS attacks. In addition, the 802.1X authentication may be vulnerable to Session Hijacking and Man-in-the-Middle (MitM) attacks [27]. Though these attacks disappear when mutual authentications and strong encryption are used [13], it discloses some deficiencies of grabbing 802.1X, which is originally designed for a switched LAN, to implement in a shared media WLAN.

As a long-term solution, IEEE 802.11i [21] is proposed to provide an enhanced MAC layer security. Under the assumption of upgrading the hardware, 802.11i defines a Counter-mode/CBC-MAC Protocol (CCMP) that provides strong confidentiality, integrity, and replay protection. In addition, an authentication process, combining the 802.1X authentication and key management procedures, is performed to mutually authenticate the devices and generate a fresh session key for data transmissions. Since 802.11i promises to be the right solution for wireless security, it should be able to prevent an adversary from advanced attacks even if the adversary could have the most powerful equipments and techniques for breaking into the system. In other words, an implementation of 802.11i protocols in a WLAN should be able to provide sufficient data confidentiality, integrity, and mutual authentication.

Note that some vendors have adopted a MAC-address-based Access Control List (ACL) for authorization. Specifically, only stations with their MAC addresses existing in the ACL list are able to access the network. This method is considered to be useless since an adversary can obtain valid MAC addresses easily by sniffing the traffic [4]. Another mechanism, called Closed System Authentication, is also proposed for access control, by disabling the SSID broadcast in the Beacon. The SSID is treated as a secret; only the user knowing the SSID can join the network. However, this approach is worthless because an adversary can obtain the SSID from the Probe Request/Response frame or (Re)Association Request frame of a legitimate user [25]. In this paper, the requirements of authorization and access control are not considered; instead, mutual authentication is required, from which flexible authorization and access control policies could be implemented.

Furthermore, note that there are also many other approaches to secure a WLAN from other layers than MAC. From the Physical Layer, proper antenna selecting and positioning can reduce the signal leakage, thus, enhance the security [26]. It is also possible to implement a RF Firewall architecture to protect the WLAN [36], but this requires significant modifications on the 802.11 PHY. In addition, some vendors have adopted different existing protocols to secure network connections through the IP or upper layer, such as IPsec and SSL. However, the mechanisms in the upper layers might not be a good solution for a WLAN, because the devices may need partial network access prior to authentication; the complexity and cost of wireless devices may increase; the performance may decrease; and typically such mechanism might be not extensible and ubiquitous [3]. Hence, 802.11i chooses to provide security in the Link Layer.

2. Availability and DoS Attacks

The past researches have extensively focused on the data confidentiality, integrity, and mutual authentication for wireless security. However, as another necessary requirement, availability has not been considered sufficiently. Some might think that the DoS attacks seems to be inevitable due to the physical characteristics of wireless links. However, since many DoS attacks can be mounted by an adversary with moderate equipments, and a successful DoS attack may facilitate other advanced attacks, such as Session Hijacking and Man-in-the-Middle (MitM), they should be considered to be real threats to a WLAN implementation. Many DoS attacks have been disclosed on the WLAN systems from the Physical Layer to the Application Layer. The key point to mitigate these attacks is to impose relatively higher cost for an adversary, e.g., more computation power, more message transmissions, or more memory consumption, which could make the DoS attacks impractical.

In the Physical Layer, a straightforward DoS attack is the frequency jamming; an adversary could interfere the whole frequency band with a strong noise signal, blocking the legitimate data transmissions. This appears to be inevitable. Fortunately, it is relatively expensive because the adversary needs special equipments and huge power consumption to jamming the whole spectrum. Currently Spread Spectrum technology has been widely adopted in wireless networks, which makes the frequency jamming much more difficult. Furthermore, an adversary performing this attack can be easily detected and located by a network administrator. Therefore, it is reasonable to assume that an adversary will not try to launch this attack for common purposes except military. There exists another easier approach to mount a frequency jamming by exploiting the Clear Channel Assessment (CCA) procedure in a WLAN implementing Direct Sequence Spread Spectrum (DSSS) [7, 42]. Particularly, most vendors do not remove the engineering function PLME-DSSSTESTMODE from their released products, which makes the attack more convenient through the off-the-shelf usage of a common wireless Network Interface Card

(NIC). There are no complete solutions for this DoS attack yet. Fortunately, the attack appears to only affect a WLAN system implementing DSSS (e.g., 802.11b/g), but not OFDM (Orthogonal Frequency Division Multiplexing, e.g., 802.11a/g).

In the MAC layer, an adversary can scramble the channel by MAC preemptive jamming because the WLAN is designed to be cooperative. For example, the adversary can send out a short jamming noise in every time interval of SIFS (Short Inter-Frame Space, 10 μs in 802.11b networks), which will surely collide with all the legitimate traffic, or cause the legitimate traffic to be deferred infinitely. However, this attack is not considered to be a real threat because the adversary needs to send out about 50000 packets per second in 802.11b networks [8]. As another possible attack, an adversary is able to transmit legitimate messages, but not completely comply to the standard. Specifically, the adversary could use a smaller "backoff" time, in order to obtain an unfair allocation of the channel bandwidth. If the adversary adopts no "backoff", he may ultimately cause a DoS attack for legitimate users [24]. More DoS vulnerabilities arise from the unprotected management frames and control frames. An adversary is able to easily launch a DoS attack on a specific station or the entire Basic Service Set (BSS) by forging the Deauthentication, Disassociation, Traffic Indication Map (TIM), or Poll messages [8]. Furthermore, DoS attacks could be mounted by exploiting the virtual carrier-sense scheme through forging any frame, especially RTS (Ready To Send) frame, with an extremely large value of NAV (Network Allocation Vector), which can fool the devices to consider the channel busy; thus, suppress the device from transmitting messages [8, 12]. Additionally, as in a wired LAN, an adversary can perform an ARP (Address Resolution Protocol) cache poisoning to mount a DoS attack if the adversary is able to access the network in someway [17].

Furthermore, if an IEEE 802.1X authentication is implemented for stronger authentication, the adversary has more choices to mount a DoS attack through forging EAP-Start, EAP-Logoff, and EAP-Failure messages. The adversary can also exhaust the space of the EAP packet identifier, which is only 8 bits long, by sending more than 255 authentication requests simultaneously [4].

In addition, due to the speed limitation of a WLAN comparing to a wired network, it is easy to perform a DoS attack from the IP or upper layer by simply network jamming, e.g., ICMP ping flooding from a high-speed wired network [32]. However, this attack could be prevented by appropriate filtering policies and traffic shaping in the Access Point. An adversary can also exploit the deficiencies of the upper layer protocols to mount a DoS attack; improvements on the upper layer protocol might be necessary for countermeasures.

As a summary, in the current WLAN system, DoS attacks are very easy to mount; furthermore, once an adversary successfully mounts a DoS attack, more advanced attacks, such as MitM [25], could be subsequently constructed. Therefore, it is necessary to deploy a security mechanism that can defend against DoS attacks. Since 802.11i does not emphasize such objective, it is definitely valuable if it can be improved to mitigate the DoS attacks. Note that this paper focuses on DoS vulnerabilities in the Link Layer; Other possible DoS attacks in PHY or upper layers are out of the scope of this paper.