Link-Layer Protection in 802.11i WLANs with Dummy Authentication

Zhimin Yang, Adam C. Champion, Boxuan Gu, Xiaole Bai, and Dong Xuan

Department of Computer Science and Engineering
The Ohio State University
2015 Neil Avenue
Columbus, OH, USA 43210-1277
{yang,champion,gub,baixia,xuan}@cse.ohio-state.edu

ABSTRACT

The current 802.11i standard can provide data confidentiality, integrity and mutual authentication in enterprise Wireless Local Area Networks (WLANs). However, secure communication can only be provided after successful authentication and a robust security network association is established. In general, the wireless link layer is not protected by the current standard in WLANs, which leads to many possible attacks, especially in public open-access wireless networks. We argue that regardless of the type of network under consideration, link-layer protection and data confidentiality are of great importance in wireless applications. In this paper, we first identify and analyze the security issues ignored by the current 802.11 security standard. Then we propose our solution to patch the current 802.11i standard and address all those issues with a new *dummy authentication* key-establishment algorithm. *Dummy* means *no real authentication for a user.* In dummy authentication, we apply public-key cryptography's key-establishment technique to the 802.11 MAC protocol. Our solution can provide link-layer data encryption in open-access wireless networks, separate session encryption keys for different users, and protection for important frames such as management and null data frames as well as Extensible Authentication Protocol (EAP) messages.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

Algorithms, Design, Security

Keywords

Dummy Authentication, Open Access, Security, WLAN

WiSec'09, March 16–18, 2009, Zurich, Switzerland.
Copyright 2009 ACM 978-1-60558-460-7/09/03 ...$5.00.

1. INTRODUCTION

Wireless networks offer organizations and users great benefits such as flexibility, portability and low installation cost [13]. Commercial Wireless Local Area Network (WLAN) products are widely available on the market and they facilitate easy setup and convenient access. Currently, most popular commercial wireless network interface cards and wireless routers support the following security options: *disabled*, *Wired Equivalent Privacy* (WEP), *Wi-Fi Protected Access-Enterprise* (WPA-Enterprise) and *WPA-Personal* (also known as *WPA-Pre-Shared Key* or WPA-PSK).

When security is disabled, the network is an *open-access* network that is open to everyone and requires no user authentication. Due to their simplicity and low management costs, open-access networks are widely used. However, they provide no protection whatsoever and are vulnerable to passive eavesdropping, traffic analysis and user fingerprinting. They are also vulnerable to active attacks such as wireless packet injection. It is very easy to inject malicious code in HTTP traffic or false DNS or DHCP responses for attack purposes such as phishing. We argue that *open-access wireless networks should be open* only *to access,* not *to attack, and they should also provide data confidentiality.*

WEP is used to provide secure data communications over wireless links. However, numerous researchers have shown that there are significant deficiencies in both the authentication and the encryption mechanisms [5–7, 9, 12, 14]. Recent research shows that an active attack on the WEP protocol can recover a 104-bit WEP key in less than 60 seconds [14].

WPA-Enterprise networks use the Extensible Authentication Protocol (EAP) to authenticate a user before he is allowed access to a network. Previous research in 802.11i (also known as WPA2) [10,11] has shown that it is a well-designed standard for data confidentiality, integrity and mutual authentication in enterprise networks. In WPA-PSK networks, the EAP authentication is reduced to a simple pre-shared common password instead of user-specific credentials. As long as a user provides the correct password for the network, he will be granted access to a WLAN. In all WPA networks, encrypted data communications can be provided only after a Robust Security Network Association (RSNA) is established, before which all wireless messages are sent in the clear. Because the four-way handshake is not protected, i.e., encrypted, the messages can be easily captured. This leads to vulnerabilities in PSK in practice. WPA-PSK with a weak key is subject to dictionary attacks. Although this is not a design flaw of the standard, in reality, most people use a simple password as a pre-shared passphrase. WPA-PSK is also vulnerable to insider attacks, passive eavesdropping, or active attacks. Although each user can have a different passphrase in theory, current products only allow *one* passphrase to be used by *all* users.

In general, wireless link frames are not protected. This includes *management* frames in all four types of networks, both *null* and *QoS null data* frames in these networks, and data frames that transfer 802.1x/EAP authentication messages in WPA-Enterprise networks. Without link-layer protection, many attacks can be launched by forging these frames, e.g., Denial-of-Service (DoS) attacks using forged disassociation frames or EAP over LAN (EAPOL) logoff messages and power-saving attacks using forged null data frames.

In order to provide secure WLANs and counter possible attacks, *link-layer protection against spoofing and eavesdropping is needed.* If link-layer protection is provided, none of the aforementioned attacks can occur. For instance, in open-access networks, an attacker can no longer obtain cleartext messages or inject malicious code.

However, providing wireless link-layer protection in WLANs is a challenging task. First, millions of wireless networking products have been released to market and are in use. Any new solution should be compatible with the existing 802.11 standards and require as few modifications as possible. Second, the computational, memory usage and wireless channel utilization overhead should be low, since both the wireless Access Points (APs) and wireless network interface cards are resource-constrained devices. Finally, the solution should resist existing attacks and *not* introduce any new vulnerabilities.

To address the issues in the standard, we propose protecting the link layer with a *dummy authentication* key-establishment algorithm in this paper. Here, *dummy* means *no real authentication for a user.* In dummy authentication, we apply public-key cryptography's key-establishment technique to the 802.11 MAC protocol. The only modifications to the 802.11i standard are a pairwise master key derivation function and the second stage of the RSNA establishment procedure, in which we replace the open-system authentication algorithm with our new dummy authentication key-establishment algorithm. We use the dummy authentication to set up a session key that is in use for a short time. Then we use this session key to derive a pairwise master key, which is also defined in the 802.11i standard and used to derive a set of encryption keys for link-layer data protection. Our method complements the standard; it is *not* a stand-alone protocol. It requires few modifications to the standard, yet it can greatly enhance the security of WLAN in a way that provides data confidentiality *without* user authentication. Our method is especially useful for public open-access WLAN environments and small-office and home networks.

In summary, we have two main contributions in this paper. First, we deeply analyze the security issues ignored by the current 802.11i standard as well as possible attacks resulting from each issue. Second, we propose a new dummy authentication key-establishment algorithm to setup a session key that is used for link-layer protection. Our solution has the advantages of simplicity, compatibility with the standard with few modifications, and low overhead in computation, memory usage and channel utilization. A device can support both open-system authentication and dummy authentication, which protects existing investments in 802.11 networks.

Numerous research has been done in WLAN security, public key-based key establishment protocols and opportunistic encryption. Due to page limitations, we do not discuss them in this paper. The rest of this paper is organized as follows. Section 2 presents some important issues neglected by the current standard. Section 3 illustrates our proposed method to address them. Section 4 presents our discussions. Finally, Section 5 gives our conclusions.

2. ISSUES IN 802.11I

In this section, we identify and analyze security issues ignored by the current 802.11i security standard.

2.1 No Protection of Open WLANs

Wireless LAN (WLAN) access points are not only deployed by organizations as a part of enterprise wireless networks, but they are also installed in public "hotspots" such as airports, hotels, conference rooms, shopping malls, and Internet cafés for public Internet access. Public WLANs —especially free Wi-Fi hotspots—are widely used worldwide and there is no sign of decreasing deployment.

Two business models are possible for a public WLAN at a hotspot: free access and paid access. No matter which business model is used, the problem is that, at present, most public Wi-Fi hotspots use open-access networks without data encryption.

It is well known that open-access wireless networks have many security issues; they are subject to numerous attacks. Anyone with a wireless device such as a PDA or laptop can sniff or monitor the air traffic. Session hijacking and traffic injection and modification can be easily performed. Several tools that do so are publicly available and a miscreant can launch an attack without any knowledge of network protocols. Examination of existing wireless network attacks shows that the lack of link-layer data encryption is the root of open-access system vulnerabilities.

One way to solve this problem is to utilize the existing 802.11i standard, deploy an authentication server (AS) behind APs, and set up a user account for each possible customer. However, much work is necessary for deployment and chaos can easily result if the network is used for a big conference or in an airport that serves thousands of customers every day.

Actually, Microsoft's Wireless Provisioning Services (WPS) are designed to simplify, automate, and standardize initial sign-up and subscription renewal so that users do not have to perform different sets of steps for each wireless provider to which they want to connect [2]. However, WPS requires several servers, including an Internet Information Services (IIS) provisioning server, a Remote Authentication Dial-In User Service (RADIUS) server configured with a certificate issued by a certification authority (CA), a Dynamic Host Configuration Protocol (DHCP) server to provide wireless customers with IP addresses, a database server to hold a promotion code database, and, finally, an Active Directory or a Lightweight Directory Access Protocol (LDAP) database to store customers' account information. Clearly, service providers' investment, management, and maintenance costs would be exorbitant.

Small businesses are justifiably reluctant to accept these burdens, not to mention setting up an AS that may cost a thousand dollars, or even the whole WPS infrastructure that requires *several* servers. In a paid-access business model, the service provider may have an incentive to set up an AS. However, for most small businesses, free access is an advertisement method to attract more customers. They just want to buy a wireless router, connect it to the Internet, turn it on, quickly set it up, and use it. Even for citywide WLANs, one requirement is the provision of free Internet access to every customer with the least possible management.

Another way to provide secure communication in public WLANs is through a Virtual Private Network (VPN), which creates an encrypted communication "tunnel" from the mobile station to a trusted VPN host. The problem is that not everyone has a VPN resource from which he can connect and open wireless networks provide a new platform for miscreants to attack the VPN itself [1].

2.2 Weak Protection of WPA-PSK

WLANs are not only deployed in enterprise networks and public hotspots: many individuals install them in their homes as a convenient extension to wired LANs. According to a recent report by Synergy Research Group, the consumer market represents over half the total WLAN market in recent years.

WPA requires an authentication server, such as RADIUS, to prevent unauthorized users from accessing the network. In situations where an AS is not feasible, such as home and small-business networks, a simpler option also exists: Pre-Shared Keys (PSKs). Later used as a Pairwise Master Key (PMK), a PSK is a 256-bit value generated by combining the WLAN's name, or *Service Set Identifier* (SSID), with an 8- to 63-byte passphrase via a predefined function. While the 802.11i standard allows unique keys to be created for each user, in practice, a single key is used for all users. Known as WPA-PSK, this approach authenticates everyone with the same secret passphrase who is connected to the access point (AP).

There is no design flaw in WPA-PSK or the 802.11i standard. But the way it is used and implemented make it vulnerable. People are accustomed to choosing short passphrases as their shared secrets. Wireless products provide no way for each user to choose a different passphrase.

2.2.1 Vulnerability of WPA-PSK to insider attacks

By *insider*, we mean anyone who obtains the key through legitimate channels. He may be a staff member in a small business or a tenant in a rented house. Although he is authorized to access the wireless network, he is *not* authorized to sniff others' communications across the network.

An insider can observe the four-way handshakes in the Pairwise Transient Key (PTK) establishment process. This four-way handshake occurs whenever a station (STA) connects to a WLAN using WPA or WPA2. It also occurs periodically thereafter whenever the AP refreshes transient keys.

Unfortunately, the way in which WPA/WPA2 encryption keys are generated and delivered makes it easy for an outside attacker to try to guess the PSK via a dictionary attack. Once an outsider has the PSK, he can steal service or decrypt data sent by legitimate users on the network. WPA-PSK has no means to prohibit a malicious insider from decrypting the data.

All the information needed to generate the PTK can be obtained from the first two messages. This includes the nonces exchanged between the AP and station, and two MAC addresses. An insider can calculate the PMK from the passphrase and therefore has all necessary information to calculate future PTKs using the following known functions.

$$\text{Function 1: } PMK = PSK = PBKDF2(\textit{passphrase}, \\ \textit{ssid}, \textit{sidLength}, 4096, 256)$$

$$\text{Function 2: } PTK = \textit{PRF-X}(PMK, \\ \text{"Pairwise key expansion"} \,||\, \min(AA, SA) || \max(AA, SA) \,|| \\ \min(ANonce, SNonce) \,||\, \max(ANonce, SNonce)),$$

where $PBKDF2$ (Password-Based Key Derivation Function) is a key derivation function defined in RSA Laboratories' Public-Key Cryptography Standards (PKCS) series. *PRF-X* is a Pseudo-Random Function based on a keyed-Hash Message Authentication Code using SHA-1 (HMAC-SHA1) that generates a PTK of size X bits, as defined in 802.11i section 8.5.1.1.

2.2.2 Vulnerability of WPA-PSK with a weak key

An outside attacker can try to guess the PSK by capturing and analyzing the four-way handshake messages. Recording handshake messages from an active WLAN is easily accomplished with a wireless capture program like Wireshark, Kismet or Airodump. Those who are impatient can use a frame injector like Airjack or Airoreplay to force legitimate users to reconnect.

From the captured data, a four-way handshake for the named WLAN can be extracted, including all of the values of interest from that handshake: the AP and station's MAC addresses, the *SNonce* and *ANonce* values, and the fourth message's payload and Message Integrity Code (MIC). Then the attacker examines the supplied dictionary file, trying words as possible passphrases to find the right PSK, which is the value that, when used as a PMK with all of these observed values, generates a matching MIC. This procedure can be automated by programs like aircrack-ng, KisMAC, and coWPAtty.

To make things worse, an attacker can use pre-computed password hashes like rainbow tables [3] to speed up cracking. For example, a 2006 ShmooCon demo showed coWPAtty testing 18,000 passphrases per second using a pre-hashed WPA PSK lookup table. One of the pre-hashed WPA PSK lookup tables posted online represents 170,000 words hashed against the top 1000 most common SSIDs.

The 802.11i standard suggests that in WPA-PSK, a pre-shared key should be very strong. To enforce that, a passphrase should contain at least 20 characters. The vulnerability of WPA-PSK with a weak key to dictionary attacks is *not* a design flaw of the standard. Still, in reality, most people use a simple password as a pre-shared secret.

2.3 No Protection of Wireless Frames

In general, wireless frames are not protected. Even after a Robust Security Network Association (RSNA) is established, certain data frames are still not encrypted.

2.3.1 No Protection of Management Frames

Management frames in WLANs are not protected by the current standard. Unlike data traffic, which is encrypted with protocols such as the Temporal Key Identity Protocol (TKIP) or the Advanced Encryption Standard (AES) with Counter Mode-Cipher Block Chaining (CBC)-Message Authentication Code (MAC) Protocol (AES-CCMP), 802.11 management frames are always sent in an unsecured manner. They are always unauthenticated and unencrypted, even when the very highest levels of WLAN security are used. By exploiting these unprotected frames, denial-of-service, impersonation and modification attacks can be launched. An attacker can either flood the access point (AP) with management frames in order to exhaust its resources or send faked management frames, e.g., de-authentication and disassociation frames, to break existing connections. The de-authentication- or disassociation-frame DoS attack can be launched against either the client or the AP by sending forged de-authentication or disassociation frames. These attacks can completely bring down an existing connection and force the station to reestablish a connection. Based on these two types of attacks, a series of attacks can be launched, including WEP attacks, WPA-PSK offline dictionary attacks and man-in-the-middle (MITM) attacks.

2.3.2 No Protection of Null Data Frames

Two special data frames are *null data* frames and *QoS null data* frames. Because there is no payload data in the frames, they cannot be protected by the encryption and message authentication schemes that the 802.11i standard provides. However, because they are short and lightweight, they have several uses. For example, in power saving mode, stations use null data frames to indicate whether they are "asleep" or "awake." During active scanning, stations use null data frames to indicate "asleep" before scanning other channels and "awake" after coming back to the original channel. An attacker can take advantage of these uses and send forged null data frames to steal buffered messages in the AP. Previous analysis and experiment results show that null data-based power-saving-mode attacks can achieve greater performance than faked power-saving-poll frame attacks [8].

Null data frames can also be used in *virtual jamming* attacks, in which an attacker fills a very large Net Allocation Vector (NAV) duration field in the frames to trick honest stations into thinking that the channel is busy.

2.3.3 No Protection of EAPOL Messages

Figure 1 gives a general overview of the message flow in the EAP/802.1x authentication process among a station (STA), AP, and authentication server (AS). The data frames sent during this process are *not* protected by the 802.11i standard since an RSNA has not been established at this stage. These frames are used to transmit EAPOL (EAP over LAN) messages, and by exploiting them, denial-of-service and user-fingerprinting attacks can be launched.

An attacker can either eavesdrop on the traffic to gather sensitive information such as usernames or accounts or launch DoS attacks. Possible EAPOL-related DoS attacks include but are not limited to the following:

- Faked EAPOL-logoff attacks, where an attacker sends forged EAPOL logoff frames to the AP to force a legitimate station out of service;
- Faked EAP-failure attacks, where an attacker spoofs EAP failure frames to disconnect an authenticating session;

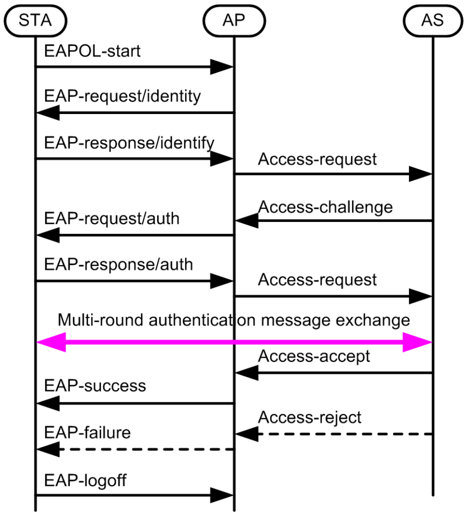

Figure 1: EAP messages exchange flow.

- Premature EAP-success attacks, where a rogue AP sends premature EAP success frames to a wireless station and forces it to drop an authenticating session.

In addition to the above issues, the first message of the four-way handshake does not use the MIC field, which means integrity and authenticity cannot be guaranteed. Thus by flooding forged initial messages, an attacker can force the AP and the station to produce inconsistent keys [9,10,13].

3. LINK-LAYER PROTECTION WITH DUMMY AUTHENTICATION

Our solution utilizes the existing symmetric key encryption algorithms, e.g., TKIP and AES, that are already used in the WPA and 802.11i products to protect wireless frames from spoofing and eavesdropping. In order to use the existing encryption algorithms, encryption keys are obviously needed. In this section, we first propose a new dummy authentication key-establishment algorithm. Then we use the established session key to protect wireless frames.

3.1 Dummy Authentication

In this subsection, we apply public-key cryptography's key-establishment technique to the 802.11 MAC protocol in dummy authentication. *Dummy* means no real authentication for a user. The only modifications to the 802.11i standard are a pairwise master key derivation function and the second stage of the RSNA establishment procedure, in which we replace the open-system authentication algorithm with our new dummy authentication key-establishment algorithm.

In order for a station to use dummy authentication, an AP should indicate that it supports dummy authentication in its beacon frames. Note that an AP can also support open-system authentication as well to provide backward compatibility. In this subsection, the dummy authentication algorithm is introduced first. Then we describe the resulting RSNA establishment procedure. Finally, we explain how to use the common session key obtained from dummy authentication to derive the pairwise master key.

3.1.1 Dummy Authentication Algorithm

We assume that each AP has a public-private key pair, denoted as pk and sk, e.g., RSA keys. The public key is contained in a CA-signed certificate or a self-signed certificate. We also assume that each AP has two secret symmetric keys k and k_{pre} that the AP updates regularly.

The dummy authentication key-establishment algorithm is as follows:

1). A station STA sends a request (ids) to AP, where ids is the station's MAC address. In the wireless frame, the authentication algorithm number is set to 2, which indicates the new dummy authentication. The Authentication Transaction Sequence Number (ATSN) is set to 1.

2). The AP creates a ticket $ticket := (ids, ts, l, hmac)$. Here ts is a timestamp from the AP's local clock, l defines a validity period for the ticket and $hmac := HMAC(ids, ts, l)$ is the keyed hash of (ids, ts, l) using AP's secret key k. The AP sends $(ida, code, ticket, cert)$ to the station. Here ida is the AP's MAC address, $cert$ is the AP's certificate and $code$ is a status code that indicates success or rejection. As before, an AP can reject a client due to MAC address access-control-list violation or for other reasons. The ATSN is set to 2.

3). Optionally, the station verifies the AP's certificate. If it is valid, then the station stores the ticket along with the AP's public key. The station generates a random number rnd and a pre-session key psk and encrypts them with the AP's public key pk. It sends $(ids, ticket, rnd, E_{pk}(rnd \,||\, psk))$ to the AP. The ATSN is set to 3. The station now can derive the common session key csk using a pseudo-random function PRF defined in section 8.5.1.1 of the 802.11i standard as follows:

$$\text{Function 3: } csk := PRF(psk, \text{"dummy authentication"}, ts \,||\, ida \,||\, ids \,||\, hash(pk)).$$

4). The AP verifies the ticket by checking that: (a) the identifier ids matches the one in the message; (b) the current local time is within the validity period l starting from timestamp ts; and (c) the ticket digest code $hamc$ is correct. If all checks pass, the AP considers it as a valid key-establishment request and recovers the pre-session key psk by decrypting its encrypted form using AP's private key sk. Now the AP derives the common session key csk using Function 3, and encrypts the pre-session key psk with it using an existing encryption algorithm such as AES. The AP sends $(ida, code, E_{csk}(psk))$ to the station and considers the station's dummy authentication as successful. The ATSN is set to 4.

5). If the code is *success* and the station can recover the pre-session key from the last message, then a common session key is established between the station and the AP. If something goes wrong during the dummy authentication process, then the station restarts it from the beginning. As in the standard, an authenticated station must be de-authenticated before it can authenticate itself again.

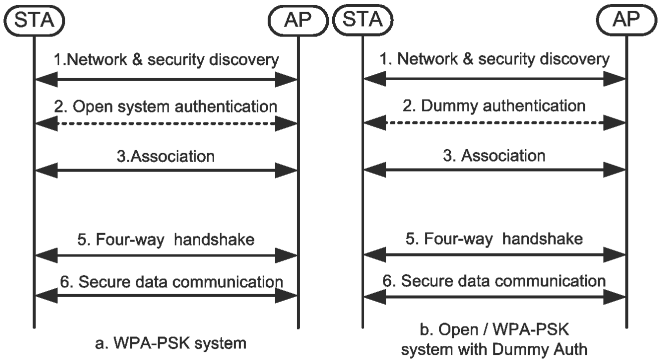
STA
AP
1.Network & security discovery
2. Open system authentication
3.Association
5. Four-way handshake
6. Secure data communication
a. WPA-PSK system
STA
AP
1. Network & security discovery
2. Dummy authentication
3. Association
5. Four-way handshake
6. Secure data communication
b. Open / WPA-PSK system with Dummy Auth

Figure 2: RSNA Establishment Procedure.

After a predefined period timeout, the AP updates its secret symmetric keys by setting $k_{pre} := k$ and generates a new k. When the AP checks a ticket, it determines which key the timestamp should use. The key-update period is longer than the largest ticket-validity period. A ticket has several functions:

1). The validity period of a ticket allows the reuse of the ticket in that period. Then steps 1 and 2 in the dummy authentication and key-establishment algorithm can be omitted, which can enhance efficiency.

2). It allows the AP to *not* allocate storage resources for a station during dummy authentication, i.e., the dummy authentication is stateless on the AP's side. For example, if we use a scheme similar to TLS, the AP needs to generate a 32-byte random number for each request and remember it in order to verify the request later. If the maximum number of resources is predefined, it will suffer from DoS attacks which are similar to association-number depletion attacks and TCP SYN flooding attacks.

3). It changes an asymmetric-cryptographic operation to a symmetric-cryptographic operation, which reduces computation time. If there is no ticket in the third step, then an attacker can waste valuable AP computation time by flooding the channel with false request messages to be decrypted with its private key.

4). It binds to a specific MAC address and introduces a timestamp as a measure to prevent replay attacks.

3.1.2 Resulting RSNA Establishment Procedure

By replacing open-system authentication with dummy authentication, the resulting RSNA establishment procedure between an AP and a station STA is shown in Figure 2(b).

Before using dummy authentication, an open-access network only has stages 1, 2 and 3. The data is not encrypted in stage 6. As shown in Figure 2(a), a WPA-PSK network does not have stage 4 which is EAP (802.1x) authentication stage in enterprise networks. Figure 2(b) shows the RSNA establishment procedure when dummy authentication is used in both an open-access and a WPA-PSK network. In a WPA-PSK network with dummy authentication, the second stage is replaced with the dummy authentication key-establishment algorithm. The other stages remain the same. In an open-access network with dummy authentication, not only is the authentication algorithm in the second stage changed, but an extra four-way handshake stage is introduced. The communication in stage 6 is encrypted by the keys derived in the four-way handshake in stage 5.

3.1.3 Pairwise Master Key Derivation

The dummy authentication key-establishment algorithm can be used in WPA-PSK WLANs as well as in open-access WLANs. The session key obtained in dummy authentication is used to derive PMKs in new open-access public WLANs, to derive PMKs in new WPA-PSK networks, and to protect management and null data frames from spoofing. While it seems redundant to use dummy authentication in enterprise networks, there is still the benefit of protecting EAPOL data messages against eavesdropping and spoofing. The lifetime of a session key spans from its establishment to the end of the four-way handshakes during which a set of keys are derived as the standard specifies. After that, the communication uses new keys.

In WPA-PSK WLANs with dummy authentication key establishment, the PMK can be calculated as follows:

$$PMK := csk \oplus PBKDF2(passphrase, SSID, SSIDlength, 4096, 256),$$

where $\oplus$ denotes the exclusive OR operation and the second part is defined in the 802.11i standard. In open-access WLANs with dummy authentication, the same algorithm is used with the *passphrase* set to the string “open system”. Having obtained the PMK, the four-way handshake can be performed to confirm the possession of the shared PMK in the AP and a station and to derive a fresh pairwise transient key that protects subsequent data communications between them.

Our method only modifies the open-system authentication procedure and key-derivation algorithm. Certainly the beacon should indicate that the new authentication algorithm is supported. Everything else remains the same as in the standard. Up to now, it minimally modifies the 802.11i specification, which protects existing investments therein. We can further enhance WLAN security by using this session key to perform per-frame authentication, as in the next subsection. However, this must modify the frames that we want to protect.

3.2 Link-Layer Protection

After the successful completion of dummy authentication, the AP and the client already share a session key. This session key can be used to authenticate and validate unicast management and null data frames and encrypt the data frames that are transmitted during the EAP authentication procedure. It also helps to prevent unintended disclosure of sensitive system parameters.

A general frame protection scheme is to encrypt the frame body and attach a message integrity code (MIC) with it that is computed with a cryptographic hash function. The resulting frame has the format:

$$frame := (\text{MAC Header}, eBody, pArgs, MIC, FCS).$$

Here, the encrypted frame body $eBody := E_{tk}(\text{plaintext frame body}, pArgs)$. The message integrity code $MIC := H_{tk}(\text{plaintext frame body}, pArgs)$, where E and H denote some encryption and keyed-hash function, respectively. tk denotes a temporal key. $pArgs$ are private arguments or parameters of a given encryption function, such as an initialization vector and TKIP sequence counter (TSC) in TKIP or a positive number (PN) in CCMP. The PN is incremented by one for each new frame. $pArgs$ can be plaintext, ciphertext or a combination thereof. The MIC is added to the end of the frame before the Frame Check Sequence (FCS). The station or AP checks the MIC before accepting a frame. If a frame fails the decryption or authentication check, it is dropped. Any attempt to copy, alter, or replay the frame invalidates either the $pArgs$ (sequence number) or the MIC. The frame body can be unencrypted if only message authentication is needed.

A goal of this paper is to modify the standard as little as possible. Therefore, we apply the existing algorithms or protocols defined in the standard as much as possible.

3.2.1 Management frame authentication

Since the upcoming 802.11w amendment [4] will provide additional security measures for data origin authenticity, replay detection, and robust management frame protection, we do not duplicate that functionality here. We only propose a simple method that uses the existing message protection schemes defined in the standard. Although the 802.11i standard does not define any mechanism to protect management frames, it does define two data confidentiality and integrity protocols: TKIP and CCMP. We adopt the same encryption protocols used for the data frames to protect unicast management frames. The difference is that the *management frame body* is used in TKIP or CCMP instead of the data frame body. Since we don’t want to expose the common session key *csk*, we define a temporal key used for those protocols as

$$tk := PRFX(csk, \text{“temporal key”}, null).$$

We use $PRF256$ and $PRF128$ for TKIP and CCMP, respectively. This temporal key is only used until the end of the four-way handshake, when a set of new keys will be derived. Afterwards, these new keys will be used to protect frames. Because of the key’s short term, we do not define the *tk* renewal procedure.

We also point out that a session key in our method is derived earlier than the keys used in the standard, in which the keys are established after EAP authentication, four-way and group handshakes. We can have additional protection of the frames before and during EAP authentication and handshakes, which the new amendment cannot provide.

3.2.2 Null data frame authentication

The existing standard does not protect null and QoS null data frames, since there are no payloads in the frames. However, by applying the general protection scheme, we can extend the current standard by modifying TKIP and CCMP to calculate the MIC only. The timestamp in the previous beacon is treated as filed plaintext data, even though it is not in the resulting frame, i.e.,

$$frame := (\text{MAC Header}, null, pArgs, H_{tk}(\text{“last timestamp”}, pArgs), FCS).$$

In this case, even if there is no data to be encrypted, the MIC is different for each frame because of the changing timestamp and increased sequence number TCS or PN. This makes forging and replaying the null data frames useless.

3.2.3 EAP and handshake messages protection

There are data protection schemes (e.g., TKIP and AES-CCMP) in the 802.11i standard, but these protections can only be provided after successful EAP authentication. We can further extend them to data frames sent before and during the EAP authentication process using the temporal key *tk* derived in 3.2.1. EAP messages and (four-way) handshake messages can be protected during encryption and decryption using TKIP or CCMP with the temporal key *tk*. The sender encrypts the payload, adds the message integrity check code, and sends the resulting frame. The receiver then checks the originality and integrity of the frame and processes it after decryption.

4. DISCUSSIONS

Existing protocols, such as the Transport Layer Security (TLS) protocol, are widely used in wired networks as well as in some EAP methods. However, they cannot be used in 802.11 MAC protocols directly, since (1) they are not MAC protocols and require reliable transport-layer services that can only exist after the establishment of a RSNA; (2) the complex handshake procedure with many messages exchanged is inefficient for noisy wireless communication; One major difference between TLS and our dummy authentication is that while TLS is designed to provide mutual authentication, dummy authentication provides no *real* authentication—it is just used to establish a short-term session key instead of proving the identity of the client as in the traditional meaning of authentication.

In our method, all algorithms except public key encryption/decryption, such as TKIP, MIC, and AES, are already implemented in current products. We do not implement negotiation and multiple encryption/hash-function algorithms, so there are fewer messages exchanged and lower complexity than SSL/TLS protocols. Hence there is less program code and memory usage compared to methods with an authentication server, such as a free RADIUS implemented in an AP. On the AP side, most wireless routers on the market provide firmware upgrade functions. It is easy to upgrade them to support dummy authentication. On the station side, the software driver is more easily updated than the firmware.

4.1 Rogue-AP and MITM Attack

The proposed method partially solves the rogue AP threat in open public WLANs. If each AP in a public WLAN has a legitimate digital certificate signed by an well-known Certification Authority, a client can verify the AP's identity, hence protecting it from rogue APs. In case an AP only has a self-signed certificate, the rogue AP threat is a pre-existing vulnerability that is neither alleviated nor enhanced by the presence of dummy authentication.

When a self-signed certificate is used, man-in-the-middle attacks (MITMs) cannot be prevented in an open public WLAN. However, it will not be a problem in enterprise WLANs since mutual authentication is adopted in enterprise WLANs. In WPA-PSK, outsider MITM attacks can be prevented. Insider MITM attacks can also be prevented if the attacker does not have the AP's private key.

4.2 Denial-of-Service Attack

An attacker may launch a DoS attack by flooding messages. Since the response frame with a public key or certificate is large, an attacker can saturate a wireless channel by flooding the first message. This can achieve the same effect as probe-request frame flooding attacks. We can alleviate this attack effect by modifying the protocol as described later in this section. An attacker may try to flood the third message hoping to exhaust the AP's computation and memory resources. In order to do this, he must send frames with a different MAC address and forge tickets. We believe our method can resist this attack, since the AP can verify the ticket using its secret symmetric key instead of its public key. The AP only continues to decrypt the pre-session key after the ticket is validated. The AP will not respond to a *second* such authentication message because once the AP processes it, the station enters the "authenticated" state. According to the 802.11 MAC protocol, the expected messages at this state are associate request and de-authentication. We can also include digital signatures of the AP's messages in steps 2 and 4 to prevent them being forged.

Table 1: Performance Comparisons

Authentication algorithm	Message number	AP processing	Connection setup time
Open	2	least	shortest
PSK	$2 + 4$	2nd least	2nd shortest
Dummy	4+4	most	middle
EAP-PEAP	$> 2 + 4 + 8$	middle	longest

4.3 Performance Evaluation

We compare the performance of our proposed solution with other wireless network settings. The results are shown in Table 1. It easily follows that open networks have the fewest number of wireless messages, the least AP processing burden, and the shortest connection setup time. Although public key processing is performed in the authentication server in WPA-EAP-PEAP authentication, the connection setup time is the longest because this setting has the most wireless messages to exchange, which takes the greatest amount of time. Dummy authentication requires the AP to perform extra processing. The connection setup time is still less than that of EAP-PEAP authentication.

The dummy authentication process also works without the last message. In this case, it reduces to a three-way handshake. However, if something goes wrong, it will not be discovered until the next message.

To reduce the number of messages in dummy authentication, the AP's public key can be sent in probe-response or beacon frames. In that case, only two messages are needed. This process is as follows: (1) The client generates a session key encrypted with the AP's public key and sends it to AP in a dummy authentication request; (2) The AP returns a response. However, since probe-response frames are sent more often than authentication request/response frames, this method has greater air interface overhead than the proposed method, because a certificate (the payload) is needed in every probe-response frame.

A variation of the method is to separate the key transfer procedure and the dummy authentication procedure as shown in Figure 3(a) and 3(b).

In this improved method, we add two more messages—key request and key response—for the public-key transfer procedure. Dummy authentication and key transfer are separated into two procedures to reduce the frequency of sending large

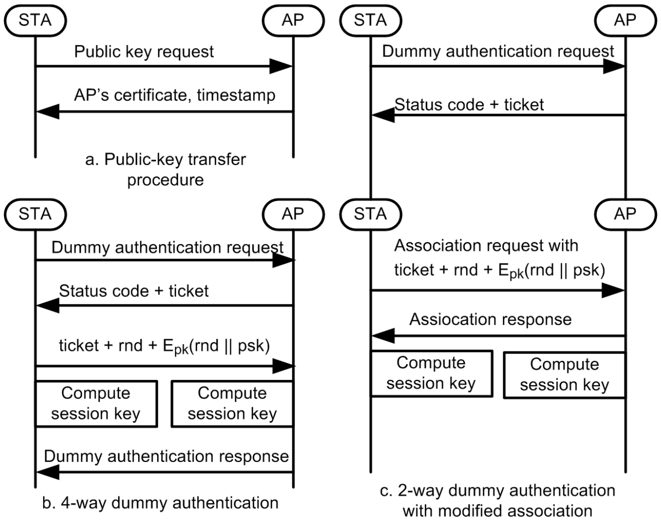
STA
AP
Public key request
AP's certificate, timestamp
a. Public-key transfer procedure
STA
AP
Dummy authentication request
Status code + ticket
ticket + rnd + E_{pk}(rnd || psk)
Compute session key
Compute session key
Dummy authentication response
b. 4-way dummy authentication
STA
AP
Dummy authentication request
Status code + ticket
STA
AP
Association request with ticket + rnd + E_{pk}(rnd || psk)
Assiocation response
Compute session key
Compute session key
c. 2-way dummy authentication with modified association

Figure 3: Variations of Dummy Authentication.

certificates, thus improving efficiency and alleviating possible DoS attacks. In the key-response frame, there is a timestamp that indicates when the certificate was last updated. This information and certificate signature can also be included in the beacon frames so that a station can check consistency or obtain the latest information. This variation has the advantage that a station only needs to request the AP's certificate once and the AP can process the key request in batch mode. This is to delay the key response and broadcast it once for awhile. The disadvantage is that two more messages are needed.

We can reduce the number of messages in dummy authentication to further improve the efficiency by transmitting some information in (re)association request/response frames as in Figure 3(c). Thus only two messages remain for the dummy authentication procedure. This improvement also has some disadvantages, such as more modification (association, re-association request and response frames) to the standard, which cannot prevent forged de-association frames during the dummy authentication process. The resulting improved method includes the public-key transfer shown in Figure 3(a) and the dummy authentication and modified association procedure shown in Figure 3(c).

We can authenticate unicast frames that communicate from one peer to another. It is expensive to protect broadcast frames or frames that have information used by all parties, although we can use the AP's public key to sign the frames. Management frame authentication can reduce the risk of DoS attacks. Currently, these attacks, such as physical jamming, cannot be prevented in principle.

5. CONCLUSION

In this paper, we identify and analyze the security issues ignored by the current 802.11 security standard. Then we propose a new dummy authentication algorithm based on public-key cryptography to address all these issues. We also discuss some variations to reduce the number of messages and improve the efficiency of the new method. Although based on a simple idea, the new method can provide link-layer data encryption in open wireless networks, separate session keys for different users and protection for important frames such as management frames, null data frames, and EAP authentication messages.

6. ACKNOWLEDGEMENT

We would like to thank the anonymous reviewers for their comments and suggestions. This work was supported in part by the US National Science Foundation (NSF) CAREER Award CCF-0546 668, the Army Research Office (ARO) under grant No. AMSRD-ACC-R 50521-CI. Any opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

7. REFERENCES

[1] http://searchnetworking.techtarget.com/generic/0,295582,sid7 gci1173698 tax303099,00.html.

[2] http://technet.microsoft.com/en-us/library/bb878131.aspx.

[3] http://www.antsight.com/zsl/rainbowcrack/.

[4] D. Akin. 802.11w - Management Frame Protection. *http://www.cwnp.com/community/index2.php?option=com_content&do_pdf=1&id=54.*

[5] J. Bellardo and S. Savage. 802.11 Denial of Service Attacks: Real Vulnerabilities and Practical Solutions. In *12th USENIX Security Symposium*, Aug. 2003.

[6] J.-C. Chen, M.-C. Jiang, and Y. wen Liu. Wireless LAN Security and IEEE 802.11i. *IEEE Wireless Communications*, 12(1):27–36, Feb. 2005.

[7] D. B. Faria and D. R. Cheriton. DoS and Authentication in Wireless Public Access Networks. In *ACM Workshop on Wireless Security (WiSe '02)*, Sept. 2002.

[8] W. Gu, Z. Yang, C. Que, D. Xuan, and W. Jia. On Security Vulnerabilities of Null Data Frames in IEEE 802.11-based WLANs. In *IEEE International. Conference on Distributed Computing Systems (ICDCS)*, June 2008.

[9] C. He and J. C. Mitchell. Analysis of the 802.11 4-way Handshake. In *ACM Workshop on Wireless Security (WiSe '04)*, pp. 43–50, October 2004.

[10] C. He and J. C. Mitchell. Security Analysis and Improvements for IEEE 802.11i. In *12th Annual Network and Distributed System Security Symposium (NDSS '05)*, pp. 90–110, Feb. 2005 .

[11] C. He, M. Sundararajan, A. Datta, A. Derek, and J. C. Mitchell. A Modular Correctness Proof of IEEE 802.11i and TLS. In *12th ACM Conference on Computer and Communications Security (CCS '05)*, Nov. 2005.

[12] J. S. Park and D. Dicoi. WLAN Security: Current and Future. *IEEE Internet Computing*, 7(5):60–65, Sept.–Oct. 2003.

[13] F. D. Rango, D. C. Lentini, and S. Marano. Static and Dynamic 4-way Handshake Solutions to Avoid Denial of Service Attack in Wi-Fi Protected Access and IEEE 802.11i. *EURASIP Journal of Wireless Communication Networks*, 2:73, Apr. 2006.

[14] E. Tews, R.-P. Weinmann, and A. Pyshki. Breaking 104-bit WEP in Less Than 60 Seconds. *Cryptology ePrint Archive: Report 2007/120, available at http://eprint.iacr.org/2007/120*, 2007.