Alternative Pair-wise Key Exchange Protocols for Robust Security Networks (IEEE 802.11i) in Wireless LANs

Hayriye Altunbasak and Henry Owen
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332--0250
Email: [hayriye, owen]@ece.gatech.edu

Abstract

The security flaws of the current the IEEE 802.11 standard are well known and have widely been publicized. To provide improvements in security and enhance the current 802.11 MAC, the IEEE 802.11i task group has proposed a new security architecture called Robust Security Network (RSN). The proposed improvements in the new architecture focus on two areas; the IEEE 802.1X standard for access control and encryption using Advanced Encryption Standard (AES). RSN uses a pair-wise key exchange protocol utilizing 802.1X for mutual authentication. In this paper, we present two alternative pair-wise key exchange protocols to provide mutual authentication in Robust Security Networks. We utilize the same message structures used in RSN and reduce the number of handshake messages. These alternative protocols improve the timing requirements for mobility in WLANs, reduce channel contention, and decrease computational load on all users in WLANs. We also discuss the advantages and disadvantages of our proposed protocols in terms of complexity and vulnerabilities.

1. Introduction

Wireless Local Area Networks (WLANs) have quickly become part of everyday life. The need for security solutions for a wide variety of users has become inevitable with the rapid growth of the wireless systems. The IEEE 802.11i task group has proposed a new security architecture called Robust Security Network (RSN) to improve the security of the current 802.11 MAC. This new architecture utilizes the IEEE 802.1X standard for access control and Advanced Encryption Standard (AES) for encryption. RSN uses a pair-wise key exchange (four-way handshake) protocol utilizing 802.1X for mutual authentication and key management process.

In RSN, the four-way handshake protocol is designed to be generic. Using only one type of handshake protocol for key management does not allow using different key management processes at different users. It is possible to define a number of key management schemes for applications depending on their performance constraints. Specifically, it is desirable to minimize the number of messages so as to reduce the delay during the handoff process as well as reducing the channel contention. In RSN, since the four-way handshake protocol is used every time a user associates/re-associates with an access point and during a handoff process to setup new keys it contributes to the delay and the channel contention in the system. Delay during a handoff process is composed of probing/beacon delay, authentication delay, and association/re-association delay. Recent studies show that in an open network, probe/beacon delay is responsible for the majority of the measured delay [1], [2], [3]. The four-way handshake protocol contributes as an association/re-association delay during a handoff process.

In this paper, we propose two alternative pair-wise key exchange protocols to reduce computational load on the users, reduce the time required to establish a pair-wise key, and reduce the channel contention to improve performance. We first provide an overview of the IEEE 802.11 and Robust Security Networks with the IEEE 802.1X. This is followed by the description of the pair-wise key exchange and key management in RSN as it proposed in the IEEE 802.11i drafts. Next, we present the proposed pair-wise key exchange protocols with a discussion on advantages and disadvantages of these protocols. Finally, we present some concluding remarks and discuss a number of open issues.

2. IEEE 802.11 and Robust Security Networks (RSN)

The first medium access control (MAC) and physical layer (PHY) specifications for wireless networks (IEEE 802.11) was released by the IEEE 802 LAN/MAN Standards Committee in 1997 [4]. Since then new updates on the standard have been released improving the speed and RF design. The first standard included an infrared (IR)

0-7803-8367-2/04/$20.00 ©2004 IEEE

layer and two spread-spectrum radio layers: frequency hopping (FH) and direct sequence (DS). The speed was limited to 1-2 Mbps in this standard. In 1999, the IEEE 802.11a was released. The IEEE 802.11a allowed the speed of 54 Mbps in 5 GHz frequency band. The IEEE 802.11a introduced a third radio technique; orthogonal frequency division multiplexing (OFDM), as well. However, the implementation of the IEEE 802.11a was not practical at that time. In 1999, the IEEE 802.11b was also released increasing the speed up to 11 Mbps in 2.4 GHz frequency band. That speeded up the growth of the WLAN market. In addition, the IEEE 802.11g was released in 2003. The IEEE 802.11g operates at 2.4 GHz bandwidth and supports the speed of 54 Mbps.

The IEEE 802.11 standard specifies two operation modes: infrastructure (Basic Service Set (BSS), sometimes technical acronym ESS is used instead), and ad-hoc (Independent Basic Service Set (IBSS)). In the infrastructure mode, each client or station (STA) communicates to an access point (AP), whereas in the ad-hoc mode each client communicates with other clients.

The security in WLANs has two parts: authentication and encryption. In the IEEE 802.11, authentication is used to identify a client to an access point. To ensure data privacy, the Wired Equivalent Privacy (WEP) encryption method is defined by the IEEE 802.11. In order to establish network connectivity, a STA first authenticates itself to an AP and then associates with the AP. There are two methods for authentication and access control in the IEEE 802.11 standard: open-system, and shared-key authentication. In addition to these, MAC-address based access control lists are implemented by vendors. Open-system authentication is a NULL authentication. It does not provide any keys for authentication. On the other hand, shared-key authentication assumes that a pre-shared key has been distributed to both the client and the access point. Shared-key authentication is a challenge/response mechanism with a secret key using WEP, which is based on a stream cipher RC4 encryption algorithm, to authenticate a client to an access point. After authentication, a STA sends an association request message. If the AP replies with an association response message indicating success, the STA becomes eligible to send and receive data from the network.

Recent studies have shown that the mechanisms used in the IEEE 802.11 are insecure [5], [6], [7], [8], [9], [10], [11], [12], [13]. To improve security, the IEEE 802.11i task group was formed. At the time of writing, 802.11i has not been ratified and is still in draft form. The IEEE 802.11i proposes a new security architecture called Robust Security Network (RSN). The proposed improvements to the 802.11 architecture focus on two areas: the IEEE 802.1X standard and Advanced Encryption Standard (AES), for access control and encryption, respectively. The new standard also includes enhancements to increase the security of the existing hardware (pre-RSN) with software upgrades and defines a Transient Security Network (TSN) allowing both RSN and WEP systems operate in parallel. In the 802.11i draft, an RSN defines two data privacy protocols: Temporal Key Integrity Protocol (TKIP) for pre-RSN WLAN hardware and AES-based Counter-Mode/CBC-MAC protocol (CCMP). TKIP, the next generation of WEP, has been adopted by Wi-Fi Alliance ahead of the 802.11i standard. This subset of RSN is also called Wi-Fi Protected Access (WPA) [13], [14].

In this paper, we discuss the security in RSN in the infrastructure mode, which is used more commonly. However, most of the discussion applies to the ad-hoc mode as well. In RSN, in the beginning, the process of establishing network connectivity is very similar to 802.11. A STA first authenticates itself to an AP using open-system (NULL) authentication, and then associates with the AP. During the association phase, a STA sends an association request message. This message is also used to identify the capabilities of the STA to the AP. If these capabilities are acceptable, the AP sends an association response message indicating success. However, unlike 802.11, when the AP replies with the message indicating success, the STA does not become eligible to send and receive data from the network. The STA still should follow the 802.1X authentication or pre-shared key method, and the four-way handshake protocol before it can start sending or receiving data in RSN. Figure 1 illustrates an overview of this process in the infrastructure mode.

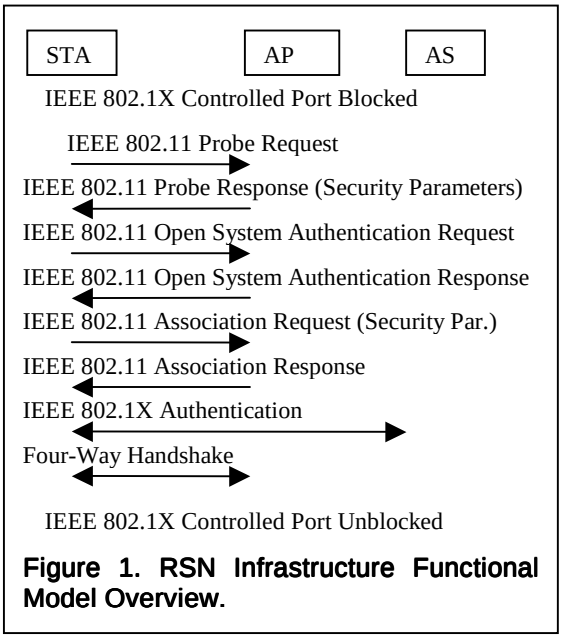

Figure 1. RSN Infrastructure Functional Model Overview.

RSN is capable of supporting two different models of operation: the IEEE 802.1X authentication and a pre-

shared key for key establishment. In RSN, the IEEE 802.1X authentication method uses an Authentication Server (AS). This model requires an upper-layer authentication process to generate matching keys (Pair-wise Master Key (PMK)) securely at a STA and an AS. The AS provides a copy of the key to the AP in a secure fashion. On the other hand, the pre-shared key method does not utilize an upper-layer authentication process. As the name suggests, it uses Pre-Shared Keys (PSKs) installed in advance in the STA and AP. In this method, the authenticity of the parties is verified by proving possession of the key. RSN uses an Information Element (IE) to negotiate the type of security in the WLAN. The IE, broadcasted in the AP's beacon, identifies if the AP uses the pre-shared key or authentication server model, the group security model the AP using, and a list of the pair-wise key mechanisms it is supporting.

In RSN, keys used for data privacy have a limited lifetime. RSN uses a pair-wise key for unicast traffic and a group key for multicast traffic. There are many keys used in RSN forming a key hierarchy. The key hierarchy starts with a Pair-wise Master Key (PMK). If the pre-shared key method is used, the PSK is the PMK. The PMK is not directly used to provide data privacy. It is used for mutual authentication of a STA and an AP, as well as for deriving the other keys in the four-way handshake protocol. A Pair-wise Transient Key (PTK) is derived from the PMK and used for data privacy and integrity. The PTK represents a set of keys derived in the four-way handshake: Temporal Keys (TK), EAPOL-Key Encryption Key (KEK), and EAPOL-Key Confirmation Key (KCK). These keys are never-before-used per-link keys. Every time a STA tries to associate to the AP they are re-computed. When the STA and AP establish a fresh PTK, the AP uses it to deliver a Group Transient Key (GTK) securely to the STA. The AP uses the group key handhsake protocol to deliver the subsequent GTKs to the STAs. After the four-way handshake, both the AP and the STA allow general the IEEE 802.11 data packets to flow.

The IEEE 802.11i provides a security solution for WLANs combining the IEEE 802.11, 802.1X, and Extensible Authentication Protocol (EAP). RSN utilizes the 802.1X standard to provide authentication, access control and key management. The purpose of the IEEE 802.1X Standard for Port-Based Network Access Control [15] is to implement access control at the point at which a user joins the network. The standard defines the Extensible Authentication Protocol over LANs (EAPOL), which uses an authentication server to authenticate each user on the network. Extensible Authentication Protocol (EAP) is an extension to Point-to-Point Protocol (PPP). EAP, defined by RFC 2284 [16], is designed to support multiple authentication methods.

The IEEE 802.1X is composed of three entities: Supplicant, Authenticator, and Authentication Server (Authorizer). An authenticator represents the network port at which a supplicant connects to the network. A supplicant authenticates via an authenticator to an Authentication Server (AS). If the AS confirms the supplicant's credentials, it directs the authenticator to provide services.

The authenticator must allow EAP traffic before authentication process completes. This is achieved by a dual-port model. In the IEEE 802.1X, an authenticator has two ports of access to the network: uncontrolled port, and controlled port. The uncontrolled port only accepts the IEEE 802.1X packets regardless of the authorization state, whereas the controlled port accepts packets from authenticated devices.

In the IEEE 802.11i context, a supplicant represents a STA, and an authenticator represents an AP. The AS can be a separate sever (e.g. RADIUS server, etc.) or built into the AP. In RSN, at the end of communication between the supplicant and AS, if an EAP Success message is delivered to the supplicant (STA), the authenticator (AP) starts a four-way handshake protocol. At the end of the IEEE 802.1X EAP authentication process between the STA and AS, both the STA and AS possess a fresh Pair-wise Master Key (PMK). The AS provides the PMK to the AP separately. To ensure delivery of the PMK to the legitimate AP, a secure connection between the AP and AS is required. If, instead of the upper-layer authentication, the pre-shared key method is used, PSK becomes the PMK. The AS makes a decision whether to admit or block the STA and informs both the AP and STA. In RSN, in addition to the IEEE 802.1X access control mechanism, the STA and AP must mutually authenticate each other before the STA establishes the network connectivity. Until the mutual authentication is completed, the controlled port at the AP does not accept any data packets. After the four-way handshake protocol, the mutual authentication completes and the AP starts providing services to the STA via a secure channel.

3. Pair-wise Key Exchange and Key Management in RSN

There are many keys used in RSN forming a key hierarchy. The key hierarchy starts with a Pair-wise Master Key (PMK). The PMK is used for mutual authentication of a STA and an AP. In addition, it is used to derive the other keys during the four-way handshake. The Pair-wise Transient Key (PTK) represents a set of keys: Temporal Key (TK), EAPOL-Key Encryption Key (KEK), and EAPOL-Key Confirmation Key (KCK). These keys are never-before-used per-link keys. Each time a STA tries to

associate to an AP they are re-computed. The liveliness of the TKs is achieved by nonces. After the STA and AP establish a fresh PTK, the AP uses it to deliver a Group Transient Key (GTK) to the STA. The group handshake enables group key updates as needed, as well.

In RSN, the four-way handshake protocol uses EAOPL-Key messages. The IEEE 802.1X defines a key message, EAPOL-Key, to allow a supplicant and authenticator to exchange secret key information. The IEEE 802.11i standard modifies this key message and uses it in the four-way handshake protocol.

The four-way handshake protocol starts with an EAPOL-Key message (M1) from the AP to the STA. The AP generates a nonce (a random or pseudo-random number) value (*256* bits) before sending the first message. This nonce is called ANonce. ANonce is included in the first message in the clear. The M1 is not encrypted or protected in any way. Meanwhile, the STA generates a nonce value called SNonce. When the STA receives the M1, it computes the PTK. The STA and AP need five inputs to compute the PTK: PMK, SNonce, ANonce, the MAC addresses of the APand the STA. The IEEE 802.11i describes the algorithm to compute the PTK. The AP is not able to compute the PTK yet since it does not know the value of the SNonce. After computing the PTK, the STA sends an EAPOL-Key message (M2) to the AP. The M2 contains the SNonce value (*256* bits) unencrypted. The M2 includes a message integrity code (MIC) to detect any modifications in the message. The AP first extracts the SNonce from the message and then computes the PTK. To ensure that both the AP and STA have the same key and that the M2 is not modified in any way by a third party, the AP verifies the MIC over the whole message. The MIC value is calculated using EAPOL-Key Confirmation Key (KCK). At this point, the AP knows that it has the correct PTK, but the STA does not know if the AP has the correct key. The AP sends an EAPOL-Key message (M3) to the STA including the ANonce, a starting sequence number, and a MIC check. The M3 informs the STA that the AP is ready to use the TK for encryption. The STA verifies that the AP knows the temporal keys and that the M3 is not altered in any way by computing the new MIC over the whole message. The last message (M4) in the four-way handshake protocol is sent by the STA to the AP. The purpose of this message is to acknowledge the completion of the four-way handshake. The STA installs its keys after sending the unencrypted M4. When the AP receives the M4, it installs its keys too. This completes the four-way handshake. Figure 2 illustrates the four-way handshake protocol and shows some of the important fields in the handshake messages.

During the four-way handshake, the AP delivers a GTK to the STA, as well. After the four-way handshake protocol, the AP may update the GTK as needed. The GTK is derived by the AP using a Group Master Key (GMK), MAC address of the AP, and GNonce (a random or pseudo-random value). The GTK is encrypted with the KEK. When the AP receives the M2 from the STA, it includes the GTK in the M3. In Figure 2, GTK[N] represents the GTK encapsulated with its KeyID. The GTK[N] must be encrypted with the KEK in the M3. The group key updates done after the four-way handshake protocol require two handshake messages. The first message of the group handshake protocol sent by the AP delivers a new GTK to the STA. This EAPOL-Key message contains the encrypted GTK, last transmit sequence number for the GTK, and the MIC computed over the body of the EAPOL-Key frame. The STA sends an EAPOL-Key message in response. This message acknowledges the new group key and includes a MIC code.

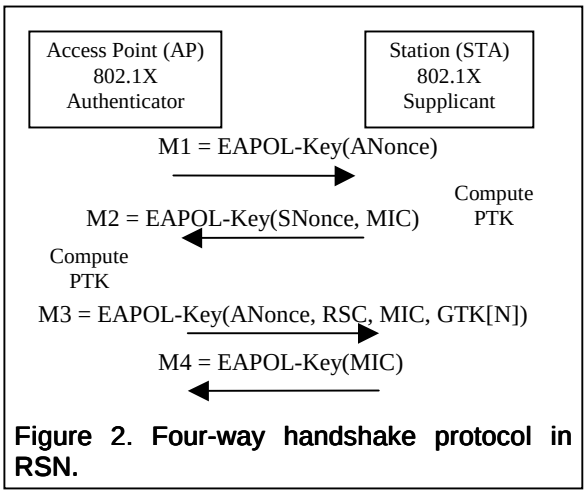

Figure 2. Four-way handshake protocol in RSN.

4. Proposed Pair-wise Key Exchange Protocols

4.1. Three-Way Handshake Protocol

In this paper, first we propose a three-way handshake protocol to establish the PTK. The proposed protocol is a modified version of the four-way handshake protocol in RSN. In this protocol, we omit the last message (M4) in the four-way handshake protocol. The AP waits for a timeout period after sending the M3 to the STA. If the AP does not receive a repeat of the M2 from the STA during that period, it installs the PTK. The STA installs its keys after it receives the M3 from the AP. The proposed protocol concludes the PTK establishment process in three handshake messages instead of four.

The last message in the four-way handshake protocol does not have any significance. Removing the M4 from the four-way handshake protocol does not prevent the

PTK establishment at the AP. The AP installs the keys even if the M4 is lost. In the four-way handshake protocol, the AP uses a timeout counter after it sends the M3. If the AP does not receive the M4, it timeouts and installs the TK. The STA installs the TK after it sends the M4 to the AP. In the end, even though the M4 is lost, the AP and STA communicate using the TK or GTK.

In the proposed protocol, we change the key installation process. The AP installs the TK after the M3. However, it waits for a timeout period to install the TK after sending the M3. If the AP receives a repeat of the M2 in the timeout period, it may re-send the M3 for a certain number of retries. The STA starts the timeout counter after receiving the M3 from the AP, and installs the TK when the timeout counter reaches a limit. The disadvantage of this protocol is that the AP and STA are not allowed to send any encrypted data before the timeout period ends. In contrast, in the four-way handshake protocol, the AP installs the TK as soon as it receives the M4. Similarly, the STA installs the keys after sending the M4. Including the M4 reduces the time required to install the keys at the AP and STA.

In the IEEE 802.11i, each EAPOL-Key message is composed of *95+n* octets, where *n* is the number of data octets. In the four-way handshake protocol, the M4 does not include any data octets. It is an empty message (*95* octets) with a MIC code. The proposed protocol reduces the channel contention in the system, since it does not use the M4. Moreover, it is desirable to minimize the number of messages for faster roaming. In addition, including the M4 in the pair-wise key establishment process increases the processing time at the AP and STA. The STA computes the MIC for the M4, and the AP verifies it when it receives the M4. Finally, the M4 provides an extra message for an attacker to try computing the KCK for that session. In conclusion, the M4 in the four-way handshake protocol makes it easier to synchronize an AP and STA, whereas removing the M4 reduces the channel contention and computational load on an AP and STA.

4.2. Two-Way Handshake Protocol

We also propose a two-way handshake protocol to reduce the number of messages required to establish the PTK. In this protocol, we use a nonce and two counters. Figure 3 illustrates the proposed two-way handshake protocol. In the proposed two-way handshake protocol, the AP computes an ANonce. Then, it computes an *f(ANonce)* value, where *f( )* is a publicly known function. The AP uses the *f(ANonce)* instead of a SNonce in the PTK computation. This enables the AP to compute the PTK before sending the first message to the STA. In the M1, the AP sends the ANonce in clear and calculates the MIC over the whole message using the KCK. In this protocol, the M1 is the same as the M3 in the four-way handshake protocol with the exception of two fields: Boot Counter (BC), and Time Counter (TC). These counters (each *64* bits) are included in the message to prevent replay attacks. When the STA receives the M1, first it extracts the nonce and computes the SNonce, which is equal to the *f(ANonce)*, and the PTK. Then, it verifies the MIC and checks the counter values against a replay attack. The STA sends the M2 to the AP including the *f(ANonce)* in clear. The *f(ANonce)* may be as simple as *ANonce+1*. To prevent attacks, the M2 includes the local boot and time counter values at the STA and the MIC calculated over the whole message. In this protocol, the M1 and M2 provide a mutual authentication. However, it is possible to replay the M1 with the same ANonce value to impersonate the AP. For that reason the STA and AP check the counter fields against attacks. The implementation of the counters with a MIC code determines the security of the proposed protocol. If the STA loses the information regarding the counters or the boot counter value is wrapped at the AP, a new PMK must be delivered to the STA and AP. Each time the STA receives a new never used before PMK, it may re-initialize the local counters to zero. After the STA sends the M2 and the AP receives the M2, the AP and STA wait for a timeout period before installing the keys.

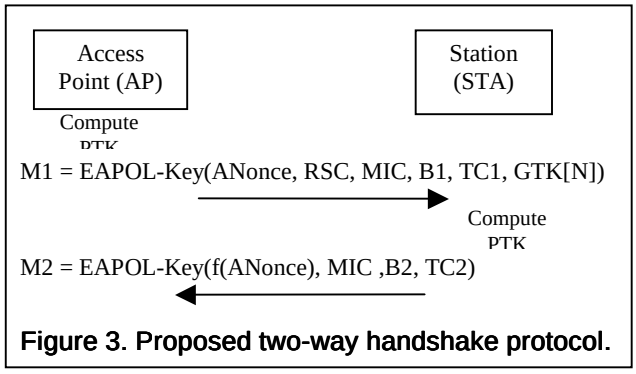

Figure 3. Proposed two-way handshake protocol.

In this protocol, a third party in the system can compute the SNonce value by listening the channel and extracting the ANonce value from the M1. This is not much different from the four-way handshake protocol. In both protocols, an observer can extract the ANonce and SNonce values from the messages since they are sent in clear. However, in the proposed two-way handshake protocol, the SNonce value is determined by the ANonce. As a result, the ANonce value defines the PTK. Even though an observer knows the SNonce value in advance, it is not able to compute the PTK as long as it does not know the PMK. Observing the M1, an attacker is not going to be able to compute the PTK to generate the correct MIC field to impersonate the STA. However, in another session, it is possible to replay the M1 with the same ANonce value to

impersonate the AP. To prevent that, the boot and time counters are included in the messages. We use the same counter concept from the Simple Network Management Protocol (SNMP) v3 [17]. We assume that the boot counter at the AP is incremented each time the AP is started or the time counter wraps around, whereas the time counter is incremented each time the AP sends a message in a two-way handshake protocol. Initially the counters are set to zero at the AP and STA. The STA caches the counter values for the APs. When the STA the first time associates with an AP, the local counters at the STA are set to zero for that AP. The STA and AP check the counter fields against attacks. Every message the STA receives in the handshake protocol should have the time counter value greater than the local time counter value and the boot counter value equal or greater than the local boot counter value cached for that AP. If the STA verifies the MIC field and the counter values satisfy the rule, first the STA sends the M2, and then, it updates the local counter values for that AP. A simple check to validate the M1 is done at the STA as follows:

```
if M1_MIC OK
        if local_boot_counter = BC1
                if local_time_counter < TC1
                        send M2(BC2 =
local_boot_counter, TC2 = local_time_counter)
                        local_time_counter = TC1
                else
                        disassociate
        else if local_boot_counter < BC1
                send M2(BC2 = local_boot_counter, TC2 =
local_time_counter)
                local_time_counter = TC1
                local_boot_counter = BC1
        else if local_boot_counter > BC1
                disassociate
else
        disassociate
```

If the STA loses the information regarding the counters, a new PMK is required to avoid replay attacks. Similarly, if the boot counter value wraps around at the AP, the STA and AP are required to setup a new PMK. Whenever a new PMK is setup between the AP and STA, the STA may re-initialize the local counter values for that AP. In addition, the STA may check whether the same ANonce value is used with the same PMK and AP pair to detect replay attacks. Nevertheless, if the pre-shared key mode is used, keeping a list of the nonces for the same PMK may not be feasible. In that case, the STA may prefer to use the counter fields only against replay attacks. Moreover, as an additional security measure, one may combine the ANonce, BC1, and TC1 to calculate the SNonce value. Note that including only the boot and time counters fields, without a MIC calculated over the whole message, do not prevent replay attacks. The mutual authentication is done assuming that the MIC code generated over the whole message changes whenever a bit value changes in the message.

One of the advantages of the proposed protocol is that it reduces the number of handshake messages for the PTK computation, therefore reducing the channel contention. Moreover, it enables the AP to pre-compute the PTK. In addition, since the number of messages used is less than the four-way handshake protocol, the proposed protocol reduces the processing time at the AP and STA. The proposed protocol uses counters and a MIC field against replay attacks. The disadvantage of using a counter is that it is not trivial to synchronize the STA and AP. Furthermore, the STA requires additional memory to cache the counters for each AP. Specifically, this protocol becomes complex in an IBSS. In an IBSS, a STA communicates with the other STAs in its neighborhood. The pair-wise key exchange protocol is used between each pair of the STAs that want to communicate with each other. In RSN, two STAs establish a secure connection when they complete the four-way handshake protocol. The unicast data frames between the STAs are protected with a pair-wise key derived during the four-way handshake. To allow broadcast/multicast frames, the broadcast/multicast (group) key of a STA must be sent to all other STAs in the IBSS. This key is sent in an EAPOL-Key message encrypted with the KEK of the PTK during the four-way handshake. Since each STA uses a different broadcast/multicast key, two STAs are required to complete two four-way handshakes. In general, $Nx(N-1)$ handshakes are required for N STAs [18]. In the proposed two-way handshake protocol, each STA should keep track of the counter values in addition to the keys for the other STAs in the IBSS and update the PMKs as needed. It is not feasible to use the proposed two-way handshake protocol in an IBSS. One solution is to define two key management schemes for a STA. The STA may choose the four-way handshake protocol or the proposed three-way handshake protocol in an IBSS, whereas it uses the proposed two-way handshake protocol in a BSS. Note that the preference between the protocols becomes a trade-off between flexible usage and performance.

5. Conclusion

In this paper, we present two alternative pair-wise key exchange protocols for Robust Security Networks (IEEE 802.11i) in wireless local area networks (WLAN). The IEEE 802.11i uses the four-way handshake protocol to provide mutual authentication and compute a never used before Pair-wise Transient Key (PTK). We propose three-way and two-way handshake protocols as alternatives to the four-way handshake protocol. The proposed three-way

handshake protocol removes the last message from the four-way handshake protocol. The three-way handshake protocol reduces the computational load and channel contention by reducing the number of handshake messages to establish the PTK between an AP and STA. It also maintains the same mutual authentication process as the four-way handshake protocol. On the other hand, the proposed two-way handshake protocol introduces a new concept to the IEEE 802.11i standard. The two-way handshake protocol uses counters in addition to a nonce in the mutual authentication process. It allows pre-computation of the PTK at the AP. However, it requires further storage space for the counters at the STA. Because of the dependence on the counters, this protocol is not feasible to implement in an IBSS. We solve this problem by using a different key management scheme in an IBSS. The STA may choose the four-way handshake protocol or the proposed three-way handshake protocol in an IBSS, whereas it uses the proposed two-way handshake protocol in a BSS. In the two-way handshake protocol, it is essential that the MIC code generated over the whole message changes whenever a bit value changes in the message. Nevertheless, the two-way handshake protocol, compared to the four-way handshake protocol, is more efficient in terms of computational load and channel contention in a BSS.

Future work should focus on the performance of these alternative protocols in WLANs. In addition, MIC attacks in the two-way handshake protocol should be investigated. Finally, the two-way handshake protocol can be modified to allow PTK pre-computations at both an AP and STA for fast handoffs in WLANs.

10. References

[1] A. Mishra, M. Shin, W. Arbaugh, I. Lee, K. Jang, "Improving the Latency of the Probe Phase during IEEE 802.11 Handoff". Available at http://www.drizzle.com/~aboba/IEEE/.

[2] A. Mishra, M. Shin, W. Arbaugh, "An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process". Available at http://www.cs.umd.edu/~waa/pubs/handoff-lat-acm.pdf.

[3] M. R. Jeong, F. Watanabe, T. Kawahara, "Fast Active Scan for Measurement and Handoff". Available at http://www.drizzle.com/~aboba/IEEE/.

[4] *IEEE 802.11 Standards*. Available at http://grouper.ieee.org/groups/802/11/.

[5] S. Fluhrer, I Mantin, A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4", *8th Annual Workshop on Selected Areas of Cryptography*, Toronto, Aug. 2001.

[6] A. Stubblefield, J. Ioannidis, A.D. Rubin, *Using the Fluhrer Mantin and Shamir attack to break WEP*, AT&T Labs Technical Report TD-4ZCPZZ, AT&T Labs 2001.

[7] B. Aboba, "WEP2 Security Analysis", IEEE doc.: 802.11-00/253, May 2001.

[8] W.A. Arbaugh, "An Inductive Chosen Plaintext Attack Against WEP/WEP2", IEEE doc.:802.11-01/230, May 2001.

[9] S. Convery, D. Miller, *SAFE: WLAN Security in Depth*, Cisco white paper. July 2002. Available at http://www.cisco.com/warp/public/cc/so/cuso/epso/sqfr/safwl_wp.pdf.

[10] W. A. Arbaugh, N. Shankar, and Y. J. Wan, "Your 802.11 wireless network has no clothes". Available at http://www.cs.umd.edu/~waa/wireless.pdf, Mar. 2001.

[11] D. Simon, B. Aboba, and T. Moore, "IEEE 802.11 security and 802.1X", IEEE Document 802.11-00/034r1, Mar. 2000.

[12] M. S. Gast, *802.11 Wireless Networks: Definitive Guide*, O'Reilly, Sebastopol CA, USA, 2002.

[13] J. Edney and W. A. Arbaugh, *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*, Addison-Wesley, Boston MA, USA, 2003.

[14] *WiFi (Wireless Fidelity) Protected Access*. Available at http://www.wi-fi.org/OpenSection/protectedaccess.asp?.

[15] *IEEE 802.1X-2001* IEEE Standards for Local and Metropolitan Area Networks: Port-Based Network Access Control (EAPOL).

[16] *PPP Extensible Authentication Protocol (EAP) RFC 2284*. Available at http://www.ietf.org/rfc/rfc2284.txt.

[17] W. Stallings, *Network Security Essentials: Applications and Standards*, Prentice Hall, New Jersey, USA, 2000.

[18] *IEEE P802.11i/D7.0* Unapproved Draft Supplement to Standard for Telecommunications and Information Exchange Between Systems-LAN/MAN Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specification for Enhanced Security.