An Initial Security Analysis of the IEEE 802.1X Standard

Arunesh Mishra
William A. Arbaugh*
{*arunesh,waa*}*@cs.umd.edu*
Department of Computer Science
University of Maryland
College Park, Maryland 20742

CS-TR-4328
UMIACS-TR-2002-10

6 Feb 2002

Abstract

The current IEEE 802.11 standard is known to lack any viable security mechanism. However, the IEEE has proposed a long term security architecture for 802.11 which they call the Robust Security Network (RSN). RSN utilizes the recent IEEE 802.1X standard as a basis for access control, authentication, and key management. In this paper, we present two security problems (session hijacking, and the establishment of a man-in-the-middle) we have identified and tested operationally. The existence of these flaws highlight several basic design flaws within 802.1X and its combination with 802.11. As a result, we conclude that the current combination of the IEEE 802.1X and 802.11 standards does not provide a sufficient level of security, nor will it ever without significant changes.

1 Introduction

Wireless local area networks (WLANs) are quickly becoming ubiquitous in our every day life. Users are adopting the technology to save the time, cost, and mess of running wires in providing high speed network access. *Hot spot* areas such as airports and coffee houses are embracing the technology to provide additional value to their customers with the hopes of increasing their revenue. To control access and provide authentication (a fundamental aspect of the business model for many of these enterprises), the IEEE 802.1X [7] standard has quickly become the mechanism of choice. While monitoring access, and uniquely identifying the users of the network is fundamental to many business models in the wireless space, providing confidentiality is not. As a result, many organizations plan to use IEEE 802.1X without encryption enabled.

One of the main reasons organizations are quickly adopting 802.1X based security is that the current security problems with wireless local area networking based on the IEEE 802.11 standard are well known [2, 4, 5], and the IEEE 802.11 standards Task Group on security (TGi) have been working on solving the problem for some time. A cornerstone of the new Robust Security Network (RSN) is the recently approved IEEE 802.1X Standard for Port based Network Access Control. The 802.1X standard is intended to provide strong authentication,

*This work was funded by a Critical Infrastructure Grant from the National Institute of Standards.

access control, and key management. Unfortunately, our initial analysis of the protocol when used in conjunction with the WLAN 802.11 standard shows that the protocol fails to provide strong access control and authentication. Using the software and tools being developed as part of the *Open1x* effort[1], we were able to mount successful *man-in-the-middle* and *session hijacking* attacks against a commercially available client/supplicant with little trouble or development effort.

Our attacks succeed because of several design flaws within IEEE 802.1X, EAP, and IEEE 802.11. Interesting, the flaws are similar in each protocol– lack of message authenticity, and lack of state machine synchronization– and the resulting composition of these protocols creates the vulnerabilities described in this paper.

In this paper, we present two attacks against the IEEE 802.1X authentication and access control mechanisms as used in an IEEE 802.11 based ISP network. We first begin by describing the basic state machine of the IEEE 802.11 protocol. This is followed by describing the elements of the Robust Security Network as it currently stands proposed. Next, we describe our attacks, and follow the attack descriptions with a discussion on how the attacks can be prevented by protocol changes. Finally, we conclude and provide an appendix of several potential denial of service attacks.

2 The IEEE 802.11 Network: Basic security Mechanisms

The IEEE 802.11 standard specifies the Medium Access Control (MAC) and physical (PHY) characteristics for devices capable of operation in the unlicensed band (2.4 Ghz and 5Ghz). It specifies operation in one of two modes : *ad-hoc* (Independent Basic Service Set) or *infrastructure* (Basic Service Set) mode. In ad-hoc mode, each client communicates directly with other clients (in RF range). On the other hand, in the infrastructure mode, there is a central entity: the *access point* (AP). Each client or station (STA) sends packets to the AP which transmits to the destination client. In this paper, we are only concerned with the security issues with infrastructure mode. In order to obtain network connectivity, a wireless client must establish a relation with an access point, called an *association*. Complete association with an access point involves transition among three states:

1. *Unauthenticated and unassociated,*
2. *Authenticated and unassociated,* and
3. *Authenticated and associated.*

Figure 1 shows the classic 802.11 state machine. An 802.11 frame can be of two basic types: *a management* frame or a *data* frame. A client transitions between the states, using specific management frames. To transition between state 1 and 2, the STA and AP exchange *Authentication Management* frames. The primary methods for authentication and access control are *open-system*, *shared-key* authentication and MAC-address based access-control lists. The Wired Equivalent Privacy Protocol (WEP) was designed to provide confidentiality for the network traffic. However, recent work [2, 4, 11, 5] has shown that all of the above mechanism are completely insecure. In order to evict these security problems, the IEEE standards group has designed a new security architecture for wireless local area networks - the *Robust Security Network* (RSN). The communication framework of RSN revolves around the IEEE 802.1X standard.

[1]The Open1x project is building open source implementations of the IEEE 802.1X standard.

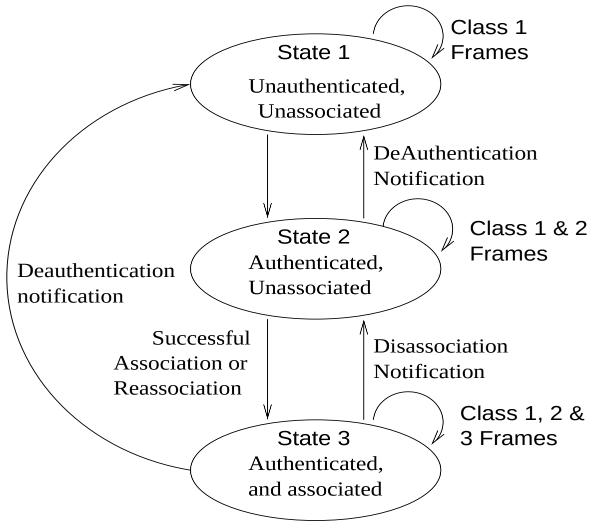

Figure 1: *The Classic 802.11 state machine.*

3 IEEE 802.1X standard and The Robust Security Network

This section describes the Robust Security Network(RSN) and elucidates the role played by the IEEE 802.1X standard. In a wireless environment, where network access cannot be restricted by physical perimeters, a security framework must provide *network access authentication*. RSN provides mechanisms to restrict network connectivity (at the MAC layer) to authorized entities only via 802.1X. Network connectivity is provided through the concept of a port which depends on the particular context in which this mechanism is used. In IEEE 802.11, a network port is an *association* between a station and an access point.

The IEEE 802.1X standard provides an *architectural framework* on top of which one can use various authentication methods such as certificate-based authentication, smartcards, one-time passwords, etc. It provides *port-based* network access control for hybrid networking technologies, such as Token Ring, FDDI(802.5), IEEE 802.11 and 802.3 local area networks. RSN leverages the 802.1X mechanism for wireless 802.11 networks.

RSN provides a security framework by abstracting three entities as specified in the IEEE 802.1X standard [7]: the *supplicant*, the *authenticator* or network port, and the *authentication* server. Figure 2 shows the communication setup. A *supplicant* is an entity that desires to use a service (MAC connectivity) offered via a port on the *authenticator*(switch, access point). Thus for a single network there would be many ports available (access points) through which the supplicant can authenticate the service. The supplicant authenticates via the authenticator to a central *authentication server* which directs the authenticator to provide the service after successful authentication. Here it is assumed that all the authenticators communicate with the same backend server. In practice this might be distributed over many servers for load-balancing or other concerns, but for all practical purposes, we can regard them as a single logical authentication server without loss of generality.

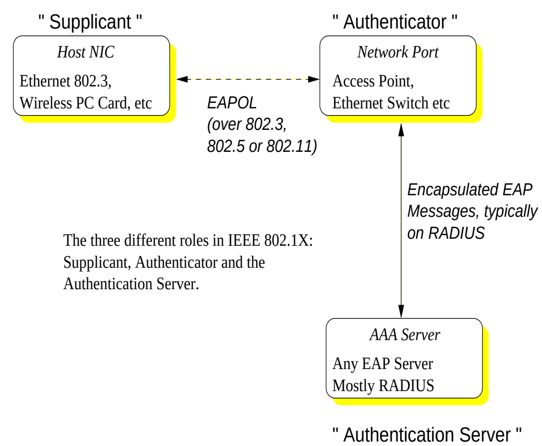

Figure 2: *The IEEE 802.1X Setup*

The IEEE 802.1X standard employs the *Extensible Authentication Protocol* [3] to permit a wide variety of authentication mechanisms. Figure 3 shows the *EAP stack*. EAP is built around the *challenge-response* communication paradigm. There are four types of messages: EAP *Request*, EAP *Response*, EAP *Success* and EAP *Failure*. Figure 7 shows a typical authentication session using EAP. The EAP Request

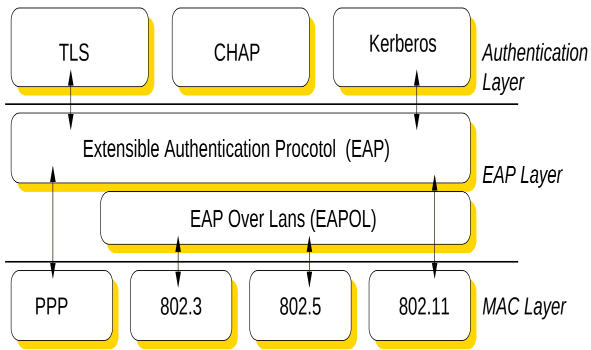

Figure 3: *The EAP stack*

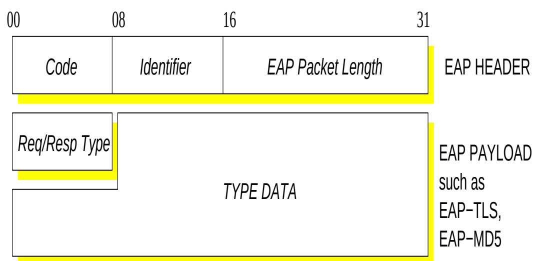

Figure 4: *The EAP Packet.*

message is sent to the supplicant indicating a challenge, and the supplicant replies using the EAP Response message. The other two messages notify the supplicant of the outcome. Figure 4 shows the EAP packet format. The protocol is 'extensible', i.e any authentication mechanism can be encapsulated within the EAP *request/response* messages. EAP gains flexibility by operating at a network layer rather than the link layer. Thus, EAP can route messages to a centralized server (an EAP server such as RADIUS) rather than have each network port (access point) make the authentication decisions.

The access point must permit the EAP traffic before the authentication succeeds. In order to accommodate this, a *dual-port* model is used. Figure 5 shows the dual-port concept employed in IEEE 802.1X. The authenticator system has two ports of access to the network: the *Uncontrolled port* and the *Controlled* port. The *Uncontrolled* port filters all network traffic and allows only EAP packets to pass. This model

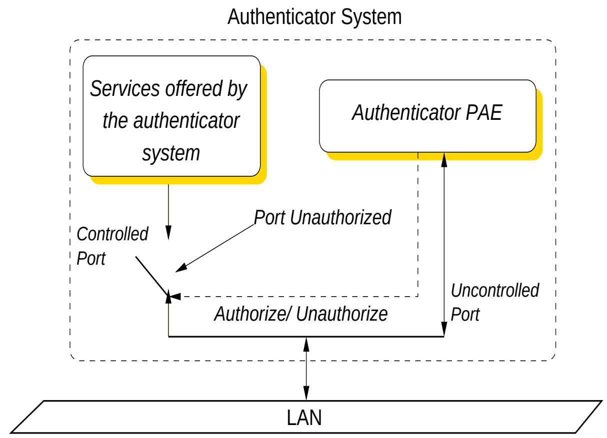

Figure 5: *The Uncontrolled and Controlled ports in the authenticator*

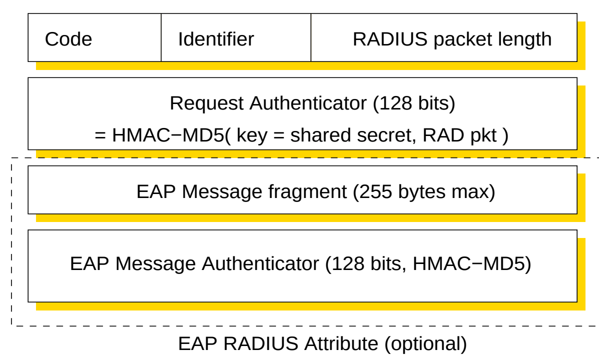

Figure 6: *Format of a typical RADIUS packet used in 802.1X authentication.*

also enables *backward compatibility* with clients incapable of supporting RSN: an administrative decision could allow their traffic through the *Uncontrolled* port.

The EAP messages are themselves encapsulated. The *EAP Over Lan*(EAPOL) protocol carries the EAP packets between the authenticator and the supplicant. It primarily [7] provides EAP-encapsulation, and also has session *start*, session *logoff* notifications. An EAPOL *key* message provides a way of communicating a higher-layer (Eg: TLS) negotiated session key. The EAP and the EAPOL protocols do not contain any measures for integrity or privacy protection.

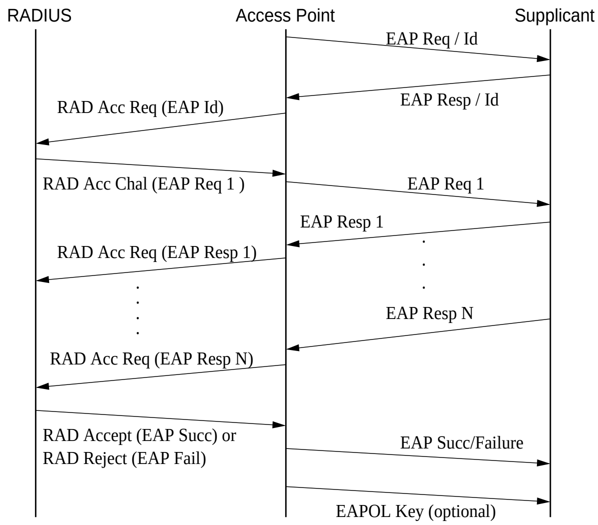

Figure 7: *A complete 802.1X authentication session showing the EAP and RADIUS messages.*

The authentication server and the authenticator communicate using the *Remote Authentication Dial-In User Service* (RADIUS) protocol [8]. The EAP message is carried as an attribute in the RADIUS protocol. Figure 6 shows a typical RADIUS packet for this scenario. The RADIUS protocol contains mechanisms for per-packet authenticity and integrity verification between the AP and the RADIUS server. Figure 7 shows a complete 802.1X authentication session.

The IEEE 802.1X standard requires that the operation of the three entities conform to the execution of specific state machines. For example, the supplicant specification has a core supplicant state machine, a *port timers* and a *key receive* state machine. A condensed form of the core supplicant state machine is shown in figure 8. The execution of the state machine determines the sequences of packets sent, the success or failure of the authentication process, the retry timeouts etc. Thus, the state machines are central to the security of the entire setup.

4 The Design Flaws in IEEE 802.1X

This section describes the design flaws we have identified when implementing the IEEE 802.1X standard. We start by listing the general goals and construct a trust model of the wireless network.

A wireless network is broadcast by nature. The media is *reachably-broadcast* i.e. only clients within a sender's RF-signal range get the transmission. This is a key distinction between wired networks. Another important difference is the centralized nature of traffic i.e. all traffic is sent to/from a central entity - the access point. Influenced by these factors, listed below are the design goals of a security framework for IEEE 802.11 LANs.

Goals of a security framework for 802.11:

1. *Access control and mutual authentication*: Because of the inherent broadcast nature, it is difficult to limit the RF signal availability to within a particular perimeter. To protect from *parking lot attacks* [2] strong access control, ideally on a per packet basis, must be a feature. Mutual authentication should also be performed as access points are untrusted entities from the supplicant's point of view.

2. *Flexibility*: Wireless networks have various environments of usage ranging from an Enterprise network (restricted use, strong confidentiality requirements) to a public wireless ISP (subscribers only, no encryption) at airports and hotels. Tailoring to the constraints of such diverse environments, the architecture should be able to flexibly include confidentiality and access control.

3. *Ubiquitous Security*: An inherent property of a wireless network is mobility. Thus the framework needs to provide authentication irrespective of the user being in the home

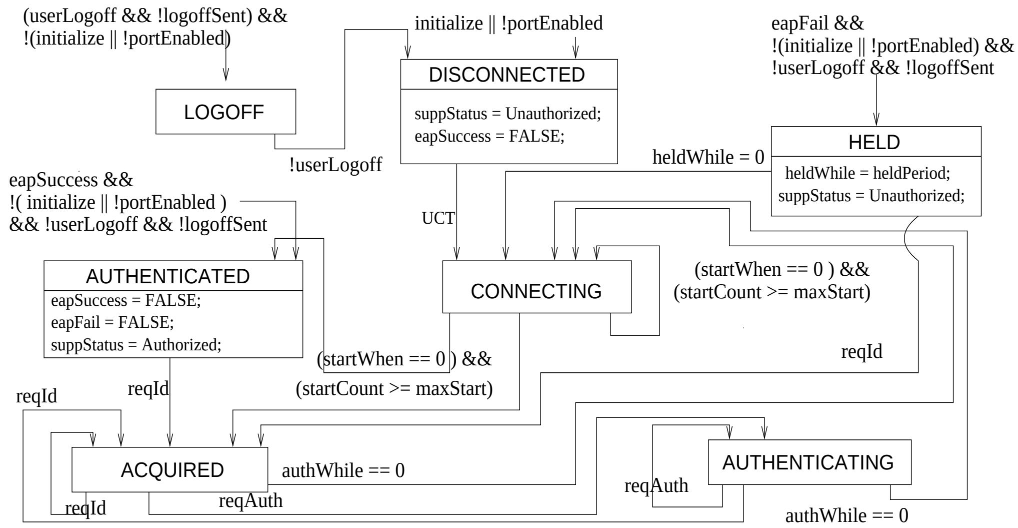

Figure 8: *This diagram shows the supplicant state machine, relevant portions only. For more details refer [7], page 66 section 8.5.10.*

or foreign network. By having a logically centralized authentication server separate from the entity providing the service (access point), a framework can provide such *ubiquitous security*.

4. *Strong Confidentiality*: Wireless is inherently broadcast, thus it is trivial for an adversary with a good receiver to eavesdrop on a station's traffic. Hence the framework needs to provide strong confidentiality guarantees (if the network policy desires). This was the primary weakness in static WEP [4]. Dynamic rekeying needs to be an inherent part of the design.

5. *Scalability*: The scheme has to be scalable in terms of the number of users and also in terms of varying mobility of a particular user (moving from one AP to another). It should have fast and secure reauthentication mechanisms.

Tailoring to the above goals, the design of RSN has abstracted the role of the three entities mentioned earlier: the supplicant, the authenticator and the authentication server. We describe the trust relationships that are inherently present in such a setup.

The trust model:

The primary role of the authentication server is to provide strong authentication and session keys to supplicants. Thus both the authenticator and the supplicant trust the integrity of the backend server which performs the authentication and issues any keys. Apart from this there is no inherent-trust between any other entities. Thus the backend server needs to ascertain the identity of the authenticator and the supplicant to provide them with a session key. Also the authentication process itself must protect against integrity and Man-In-Middle attacks because of the inherent broadcast nature of wireless.

What RSN provides:

1. *Per-packet authenticity and integrity between the RADIUS server and AP:* As mentioned earlier, the backend server and the AP (authenticator) communicate using the RADIUS protocol [8]. Each authenticator has a unique shared secret with the RADIUS (backend) server. All the RADIUS messages contain a *Request Authenticator* field which is an HMAC-MD5 of the

entire packet using the shared secret as the key. This field is set by the RADIUS server and verified by the AP. The reverse is done by the *EAP Authenticator* attribute which is present with the *EAP Message* attribute [9]. The *EAP Authenticator* is a similar hash done by the AP. These two attributes provide the per packet mutual authentication and also preserve the integrity of the communication between the RADIUS server and the AP.

2. *Scalability and Flexibility:* By separating the authenticator from the authentication process itself, RSN provides good scalability in terms of the number of access points. It provides the flexibility of including confidentiality using the optional EAPOL *key* message.

3. *Access control:* Using strong higher-layer authentication, RSN can provide good access control. Unfortunately, because of race-conditions in the loose consistency between the 802.1X and 802.11 state machines, a session-hijack attack can be performed (section 4.2).

4. *One-way Authentication:* The 802.1X state machines provide for only one-way authentication. The supplicant is authenticated to the access point. The lack of mutual authentication can be exploited to mount Man-In-Middle attacks elucidated in section 4.1

The following sections detail the primary design flaws and the exploits.

4.1 Absence of Mutual Authentication

The primary flaw in the design is the asymmetrical treatment of supplicants and access points (authenticator) in the state machines. According to the standard, the authenticator (figure 5) port is in the *Controlled* state only when the session is authenticated. This is untrue for the supplicant, whose port is essentially always in the authenticated state. The one-way authentication of the supplicant to the access point, can expose the supplicant to potential Man-In-Middle attacks with an adversary acting as an access point to the supplicant and as a client to the network access point. Figure 9 shows the details.

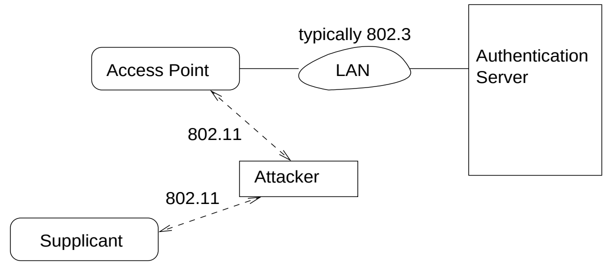

Figure 9: *The Man-In-Middle setup for the attack in section 4.1.*

The 802.1X authenticator state machine (refer [7] section 8.5.4 page 51) accepts only EAP *response* messages from the supplicant and sends only EAP *request* messages to the supplicant. Similarly, the supplicant state machine (8) does not send any EAP *request* messages. Observably, the state machines perform only a one-way authentication. The trust assumption that is reflected from this design is that the access points are trusted entities which is a misjudgment. The entire framework is rendered insecure if the higher-layer protocol also performs a one-way authentication (like EAP-MD5 [3, 10]).

EAP-TLS [1] does provide strong mutual authentication but is NOT mandatory and can be overridden. Even if it is used, the above design error can bypass the entire EAP-TLS authentication. As an artifact, a simple Man-In-Middle attack is detailed below which does this.

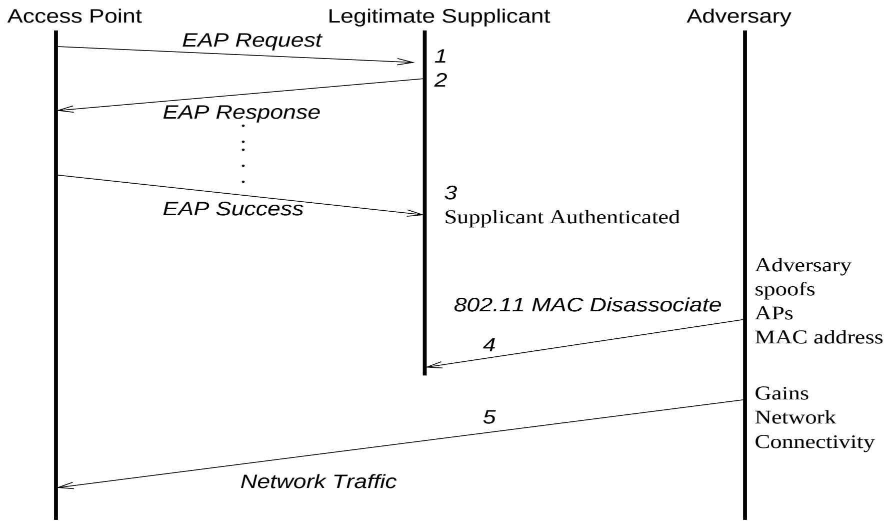

Figure 10: *The Session Hijack by spoofing a 802.11 MAC disassociate message.*

4.1.1 EAP Success Message MIM Attack

An EAP Success message is sent from the authenticator to the supplicant, on receipt of a RADIUS Access Accept message from the authentication server (RADIUS). This indicates to the state machines that the authentication has been successful. Irrespective of the higher-layer authentication method used (EAP-TLS, EAP-MD5), this message contains no integrity preserving information. Also in the supplicant state machine [7] as shown in figure 8, there is an *unconditional transfer* to the *Authenticated* state irrespective of the current state. The EAP Success message sets the *eapSuccess* flag, which makes a direct transition to the *Authenticated* state irrespective of the current state.Typically this would cause the interface to come up and provide network connectivity.

Thus, an attacker could forge this packet on behalf of the authenticator and potentially start a simple Man-In-Middle(MIM) attack. The adversary can thus get all network traffic from the supplicant to pass through it. This completely bypassed any higher-layer authentication and renders the authentication mechanism ineffective.

4.2 Session Hijacking

Figure 11 shows the RSN state machine. The primary change is the addition of a fourth state *RSN Associated*. With IEEE 802.1X, higher-layer authentication takes place after RSN association/reassociation. Thus there are two state machines: the RSN and the 802.1X state machine. Their combined action dictates the state of authentication. Because of a lack of clear communication between these state machines and message authenticity, it is possible to perform a simple session hijacking taking advantage of the loose coupling. Figure 10 shows how an adversary could defeat the access-control mechanisms and gain network connectivity. The attack proceeds as follows:

1. *Messages 1, 2 and 3:* A Legitimate supplicant authenticates itself. The EAP authentication phase has more than three messages, they omitted for brevity.

2. *Message 4:* An adversary sends a 802.11 MAC *disassociate* management frame using the APs MAC address. This causes

the supplicant to get disassociated. This message transitions the RSN state machine to the *Unassociated* state while the 802.1X state machine of the authenticator still remains in the *authenticated* state.

3. *Message 5:* The adversary gains network access using the MAC address of the authenticated supplicant because the 802.1X state machine in the authenticator is still in the *authenticated* state.

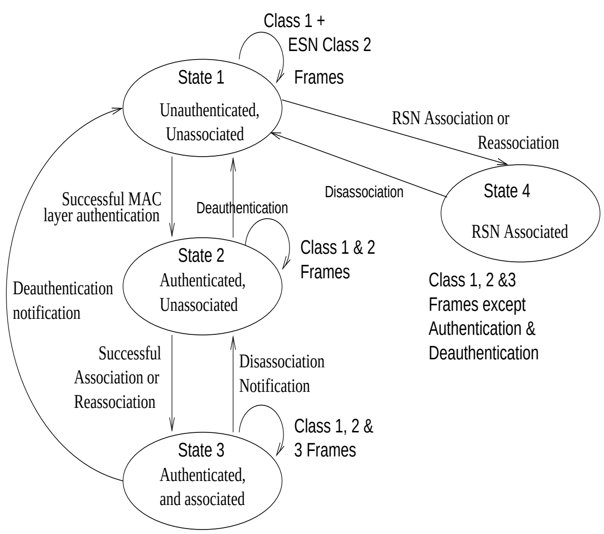

Figure 11: *The 802.11i Robust Security Network state machine.*

5 Proposed Solutions

In this section we present the changes that need to be made to the IEEE 802.11 and 802.1X standards to prevent the previously discussed attacks.

5.1 Per-packet authenticity and integrity

Lack of per-packet authenticity and integrity in IEEE 802.11 frames (data and management) has been a key contributor in many of the protocol's security problems. The session hijack attack presented in section 4.2 primarily exploited the lack of authenticity in management frames. Authenticity and integrity of data frames must also be assured to prevent simple packet forgery attacks. While the integrity of data frames is being added when confidentiality is used, there are currently no plans by the IEEE to add integrity protection to management frames.

5.2 Authenticity and Integrity of EAPOL messages

The lack of authenticity of 802.1X messages themselves was one of the primary exploits in the MiM attacks detailed in section 4.1. This could be accomplished by using an attribute such as the *EAP-Authenticator* (refer figure 6) present in RADIUS messages. The EAP-Authenticator needs to be added only to the decision message i.e. *EAP-Success*. The key for this attribute can come from the higher-layer authentication protocol such as the EAP-TLS session key. Another approach could be to eliminate an explicit *EAP-Success* message and use the *EAPOL-key* as an indication of success at the EAP layer. Figure 12 shows the EAPOL packet along with the added attribute.

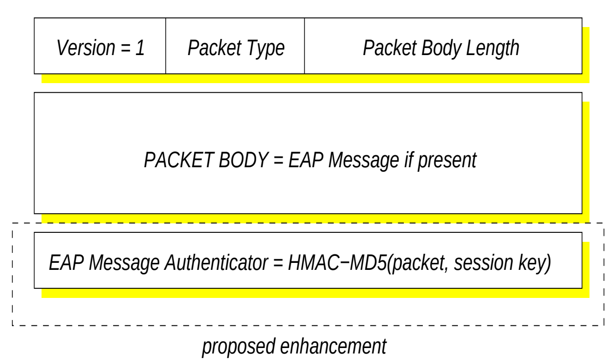

Figure 12: *The changes to EAPOL: addition of an EAP authenticator attribute.*

5.3 A peer-to-peer based authentication model

This section lists two essential properties that need to be built into the RSN framework. As a result of these, the model becomes more of a peer-to-peer authentication using a central trusted entity. An advantage of building such a framework could be added applicability in the ad-hoc wireless scenario.

Symmetric authentication: Both supplicants and access points should be considered untrusted entities. Hence a more symmetric (mutual) authentication model would be built into IEEE 802.1X. The supplicant state machine should be similar to the authenticator, including the *dual-port* model. The RADIUS server needs to treat APs and STAs in a similar manner as far as authentication is concerned. The only difference is that the STA communicates to the RADIUS server via the AP.

Scalable authentication: In order to support high mobility, the RADIUS server needs to handle the APs in a scalable manner. The current use of per-AP based shared secret is clearly not an easily manageable solution. A scalable scheme needs to be built to authenticate the APs.

6 Conclusions

The importance of security in a wireless environment can not be under stated. Because the transport medium is shared– potentially beyond the physical security controls of the organization– permits attackers easy and unconstrained access. As a result, strong access control and authentication become essential in protecting the organization's information resources. Unfortunately, our attacks demonstrate that the current RSN architecture does not provide strong access control and authentication due to a series of flaws in the composition of protocols that make up RSN.

Fortunately, however, our attacks can easily be prevented through the addition of message authenticity to EAP, and IEEE 802.11 management messages and through additional steps ensuring the synchronization of the various state machines.

References

[1] B. Aboba and D. Simon. Ppp eap tls authentication protocol. *RFC 2716*, October 1999.

[2] 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*, December 2001.

[3] L. Blunk and J. Vollbrecht. Ppp extensible authentication protocol (eap). *RFC 2284*, March 1998.

[4] N. Borisov, I. Goldberg, and D. Wagner. Intercepting Mobile Communications: The Insecurity of 802.11. In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networkging*, pages 180–188, 2001.

[5] S. Fluhrer, I. Martin, and A. Shamir. Weaknesses in the key scheduling algorithm of rc4. *Eighth Annual Workshop on Selected Areas in Cryptography*, August 2001.

[6] IEEE. Lan man standards of the ieee computer society. wireless lan medium access control (mac) and physical layer(phy) specification. *IEEE Standard 802.11*, 1997.

[7] IEEE. Standards for local and metropolitan area networks: Standard for port based network access control. *IEEE Draft P802.1X/D11*, March 2001.

[8] C. Rigney and et. al. Remote authentication dial in user service(radius). *RFC 2138*, April 1997.

[9] C. Rigney, W. Willats, and P. Calhoun. Radius extensions. *RFC 2869*, June 2000.

[10] W. Simpson. Ppp challenge handshake authentication protocol (chap). *RFC 1994*, August 1996.

[11] A. Stubblefield, J. Ioannidis, and A. D. Rubin. Using the fluhrer, mantin, and shamir attack to break wep. *ATT Labs Technical Report*, TD-4ZCPZZ, August 2001.

Appendix: Denial of Service Attacks

This section lists the attacks which could potentially cause a denial-of-service affecting the end-host or the network availability itself.

EAPOL Logoff , EAPOL Start Message spoofing

The EAPOL *Logoff* message is sent from the supplicant to the authenticator indicating that it desires to leave the authenticated use of the service offered[7, 3]. As shown in figure, all fields of this packet can be easily altered by a simple Man-In-Middle(MIM) setup. A simple spoofed message can thus cause an authenticated client to get logged off. To accomplish this, the adversary has to send an EAPOL Logoff to the access point on behalf of the supplicant. This attack could also be done at the MAC layer by sending a MAC *disassociate* message [6].

The EAPOL *Start* message is sent from the supplicant to the authenticator to start the authentication process with the authentication server. Figure 13 shows the EAPOL packet format. Like the EAPOL Logoff message this message can also be easily spoofed.

EAP Failure Message spoofing

The EAP Failure message is sent from the access point to the supplicant when the authentication process between the authentication server (RADIUS) and the supplicant fails. This message can also be spoofed and sent with the access point's (AP) MAC address to an authenticated supplicant. According to the specification for the supplicant state machine (figure 8) [7] , on receipt of the EAP Failure message, it transitions to the HELD state irrespective of its current state. Once into the HELD state, because of the *heldWhile* timer, it remains there for 60 seconds (default value). Thus in order to prevent a supplicant from even trying to reauthenticate, an adversary just has to spoof the EAP Failure message once every 60 seconds.

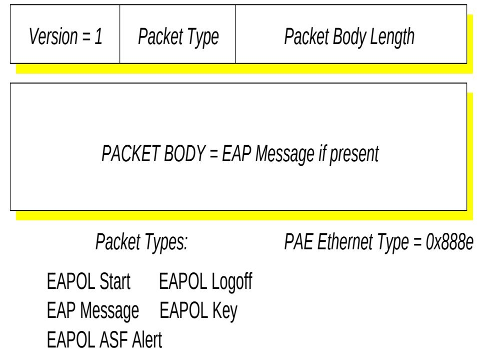

Figure 13: The EAPOL packet format.

Spoofing of 802.11 management frames

Since the IEEE 802.11 management frames contain no authentication element, they can be spoofed causing a supplicant to get logged off from an authenticated session. This *disassociate* denial-of-service attack can be performed even with dynamic WEP.

Large number of associate requests

The 802.1X authentication takes place after the association phase at the 802.11 layer is complete with the access point. An access point maintains considerable state information after association and before 802.1X completes. Since at this point, the station is not authenticated, a large number of such associations can be made by a single station using random MAC addresses. The *identifier* field in the EAP packet is 8 bits in length. Thus even if an access point has limited the number of parallel associations to 255, a single station can take part in 255 parallel authentication requests and prevent any other station from joining the access point.