EAP Methods for 802.11 Wireless LAN Security

Overview

Because they broadcast data on the open airways, wireless networks present unique challenges for authentication mechanisms not encountered on wired networks. This tutorial explores how wireless networks are different from wired networks with regard to authentication and presents the requirements that an authentication method must meet in order to be appropriate for wireless networks. It then considers several families of authentication methods that have been designed specifically around the needs of wireless networks – the public key certificate-based methods, the password methods, and the strong password methods. One particular strong password method, known as SPEKE for Simple Password-authenticated Exponential Key Exchange, is examined in some detail. The tutorial concludes with a table comparing the properties of these authentication methods to each other and to earlier legacy methods.

Topics

1. Introduction

Authentication is the process of verifying a claimed identity. In perhaps the earliest form of authentication, the person being authenticated – called the user in this tutorial – would present a password to the authority requiring authentication – called the authenticator. If the user were able to present the correct password, he or she would be authorized to gain access to something or to receive services. For some purposes, simple password authentication can provide relatively strong security, but in order to do so, certain assumptions must hold true:

- The user must have some assurance that the authenticator is in fact the authority in question.
- The communication channel between the user and the authenticator must itself be secure (user and authenticator can be sure that no one is listening).
- It must be highly unlikely that an attacker would be able to guess the password. Usually this is accomplished by limiting the number of wrong guesses.
- If the user is a human being (as opposed, say, to a software process running on a computer), the password must be easy to remember – but not so easy that it can be easily guessed!

Today's wireless networks are not your father's timesharing system. Consider a user with a laptop computer accessing an 802.11 wireless network. The first problem is that the user has no way of knowing whether the access point is, in fact, operated by the administrator of that network. It might be a rogue access point operated by another user (an imposter) who may have a connection to the target network. If so, the user we're concerned with may not even know that the data is being routed through an imposter's computer.

The second problem is that the communication channel in this case is a radio network that can be monitored by anyone with a radio receiver. It is easy for an attacker to monitor legitimate users' access attempts and collect their passwords without being detected. This problem can be mitigated somewhat through using a challenge/response authentication system in which the password is not itself transmitted over the air, but the user is presented with a challenge that is joined with the password and hashes with a secure hash function.

But now we have a new problem. The attacker can make password guesses on a separate computer by observing a single challenge and response and then attempting to join the challenge to his guesses, computing the resulting response, and comparing it to the observed response. Guesses can then be made at a very fast rate with neither the user nor the network administrator knowing about it. This form of attack is known as a dictionary attack because the attacker selects his guesses from a cracker's "dictionary" of possible passwords.

Offline dictionary attacks can be mitigated by using a large random number in place of an easily remembered password. This makes it unlikely that the password would be in the attacker's dictionary. But this violates the fourth assumption, that the password be easy to remember. To get around this problem, the password can be stored on the user's computer, but now the user has to prevent the attacker from gaining access to it by walking up to the computer without the user's knowledge or stealing the computer or, more alarmingly, by gaining unauthorized access to the user's computer over the very network the user is trying to use.

As you can see, the requirements for wireless network authentication are much more stringent than those placed by a dialup timesharing system.

In this tutorial, we will first compile a list of requirements that an authentication method must meet in order to be appropriate for use over a wireless network. This list includes additional features that an authentication method should have and a list of features that some wireless authentication methods do have that may be helpful in some environments.

Next we consider the two main families of authentication methods that meet the wireless requirements. The first family consists of those methods that incorporate the use of public key certificates. The second family contains the password authentication methods. We consider a specific strong password method, SPEKE, which has particularly good characteristics for wireless use. Finally, in the conclusion we summarize the characteristics of the authentication methods in a table that also contrasts them with older legacy methods.

2. Requirements for Wireless Authentication

What then are the requirements for an authentication method that will be used to gain access to a wireless network? The following sections list requirements that an authentication method must meet (must haves), additional characteristics that are highly desirable (should haves), and features that may be quite useful in certain environments (may haves).

2.1. REQUIREMENTS (MUST HAVES)

Mutual – It must provide mutual authentication, that is, the authenticator must authenticate the user, but the user must be able to authenticate the authenticator as well. Mutual authentication is particularly important over wireless networks because of the ease with which an attacker can set up a rogue access point. There are two possible attacks here. In one, the rogue is not connected to the target network and merely wishes to trick the user into divulging authentication credentials. In the other, the rogue is connected to the target network. The attacker may then ignore the credentials presented by the user and "authorize" network access. The user's session may then be recorded or even altered because the attacker has been inserted in the data path.

Self-protecting – It must protect itself from eavesdropping since the physical medium is not secure. The authentication must proceed in such a way that eavesdroppers cannot learn anything useful that would allow them to impersonate the user later.

Immune to Dictionary Attacks – It must not be susceptible to online or offline dictionary attacks. An online attack is one where the imposter must make repeated tries against the authenticator "on line". These can be thwarted by limiting the number of failed authentication attempts a user can have. An offline attack is one where attackers can make repeated tries on their own computers, very rapidly, and without the knowledge of the authenticator. Simple challenge/response methods are susceptible to offline attacks because if attackers capture a single challenge/response pair, they can try all the passwords in the dictionary to see if one produces the desired response.

Produces Session Keys – It must produce session keys that can be used to provide message authentication, confidentiality, and integrity protection for the session the user is seeking to establish. These keys will be passed to the user's device drivers to be used as WEP or TKIP keys during the ensuing session.

2.2. ADDITIONAL CHARACTERISTICS (SHOULD HAVES)

Authenticates User – It should authenticate the user rather than the user device. In that way it will be hardened against attacks against the user device. One useful way to meet this requirement would be for the method to depend on a simple secret that can easily be remembered by the user. Another way is to encase the secret in a smart card that is carried by the user and is separate from the device.

Forward Secrecy – It should provide forward secrecy. Forward secrecy means that the user's secret, whether password or secret key, cannot be compromised at some point in the future. An attacker who recorded a user's session encrypted by a key produced during authentication cannot, given knowledge of the user's secret, decrypt the recorded session. Once secure, the session data stays secure forever.

Access Points – It should work with all access points that support 802.1x with EAP authentication.

Quick and Efficient – The authentication should complete in a minimal number of protocol round trips, and computations necessary to complete the authentication should require a minimal amount of computing resources.

Low Maintenance Cost – It should be easy to administer. A method that requires the installation of a certificate on each user device, for example, is not easy to administer. Maintenance of certificate revocation lists can be a costly administrative burden.

Convenient for Users – It should be convenient enough to use that users will not balk. For example, using a certificate stored on a device, though, burdensome to administrators, is convenient for users. Smart cards, though inconvenient for users, are easier for administrators. Users don't mind typing a small, easy to remember password, but most would object to typing a long string of hex digits.

2.3. OTHER USEFUL FEATURES (MAY HAVES)

Augments Legacy Methods – It may protect a less secure, legacy method in such a way that the combination of the wireless authentication method and legacy method meet the above requirements. This feature is useful in environments with legacy authentication systems that cannot quickly be replaced.

Fast Reauthentication – It may provide a reauthentication mechanism that is less time and/or compute intensive than the legacy authentication. Of particular concern is enabling fast handoffs for mobile users. Since the time constraints on a handoff may be very tight, a reauthentication mechanism that takes few round trips or can be accomplished by a server in the service provider's domain rather than the user's home domain would be helpful. However, care should be taken that such reauthentication mechanisms provide strong security.

3. Certificate based Authentication methods

Today's 802.11 networks authenticate users according to the IEEE 802.1x standard. 802.1x specifies how to run the Extensible Authentication Protocol (EAP) directly over a link layer protocol. EAP is essentially a transport protocol that can be used by a variety of different authentication types known as EAP methods. EAP was standardized by the IETF in March 1998 for use over point-to-point network connections.

Among the EAP methods developed specifically for wireless networks are a family of methods based on public key certificates and the Transport Layer Security (TLS) protocol. These are EAP-TLS, EAP-TTLS, and PEAP. We will consider each of these in this section, and then consider another family of EAP methods, the strong password methods (sometimes known as Zero Knowledge Password Proof – ZKPP).

3.1. EAP-TLS

EAP-TLS uses the TLS public key certificate authentication mechanism within EAP to provide mutual authentication of client to server and server to client. With EAP-TLS, both the client and the server must be assigned a digital certificate signed by a Certificate Authority (CA) that they both trust.

Features of EAP-TLS include:

- Mutual authentication (server to client as well as client to server)
- Key exchange (to establish dynamic WEP or TKIP keys)
- Fragmentation and reassembly (of very long EAP messages necessitated by the size of the certificates, if needed)
- Fast reconnect (via TLS session resumption)

3.2. EAP-TTLS

The Tunneled TLS EAP method (EAP-TTLS) provides a sequence of attributes that are included in the message. By including a RADIUS EAP-Message attribute in the payload, EAP-TTLS can be made to provide the same functionality as PEAP (discussed below). If, however, a RADIUS Password or CHAP-Password attribute is encapsulated, TTLS can protect the legacy authentication mechanisms of RADIUS. When the TTLS server forwards RADIUS messages to the home server, it decapsulates the attributes protected by EAP-TTLS and inserts them directly into the forwarded message. Because this method is so similar to PEAP, it is being used less frequently.

Figure 1

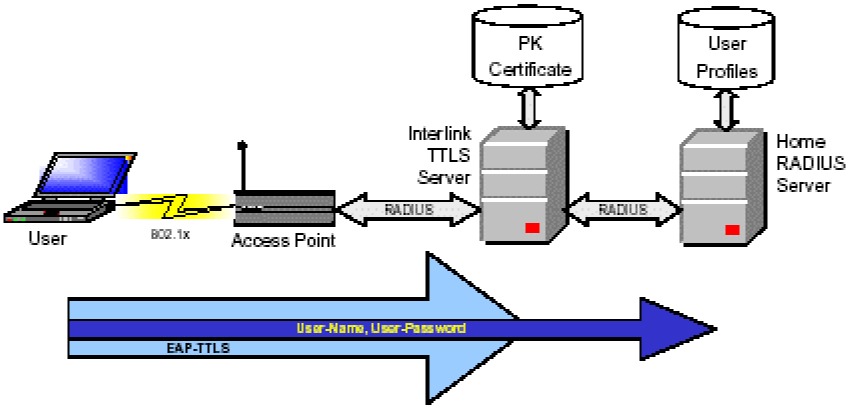

Fig. 1 — How a TTLS server interacts with a legacy RADIUS server

3.3. PEAP

Like the competing standard TTLS, PEAP makes it possible to authenticate wireless LAN clients without requiring them to have certificates, simplifying the architecture of secure wireless LANs. Protected EAP (PEAP) adds a TLS layer on top of EAP in the same way as EAP-TTLS, but it then uses the resulting TLS session as a carrier to protect other legacy EAP methods. PEAP uses TLS to authenticate the server to the client but not the client to the server. This way, only the server is required to have a public key certificate; the client need not have one. The client and server exchange a sequence of EAP messages encapsulated within TLS messages, and the TLS messages are authenticated and encrypted using TLS session keys negotiated by the client and the server.

PEAP provides the following services to the EAP methods it protects:

- Message authentication (Imposters may neither falsify nor insert EAP messages.)
- Message encryption (Imposters may neither read nor decipher the protected EAP messages.)

- Authentication of server to client (so that the protected method only needs to authenticate client to server)

- Key exchange (to establish dynamic WEP or TKIP keys)

- Fragmentation and reassembly (of very long EAP messages, if needed)

- Fast reconnect (via TLS session resumption)

PEAP is especially useful as a mechanism to augment the security of legacy EAP methods that lack one or more of the above features.

Microsoft PEAP

Microsoft PEAP supports client authentication by onlyMS-CHAP Version 2, which limits user databases to those that support MS-CHAP Version 2, such as Windows NT Domains and Active Directory.

To use Microsoft's PEAP, users must purchase individual certificates from a third-party certification authority (CA) to install on their IAS, and a certificate must be installed in the user's local computer certificate store. For wireless clients to validate the IAS certificate chain properly, the root CA certificate must be installed on each wireless client.

Windows XP, however, includes the root certificates of many third-party CAs. If the IAS server certificates correspond to an included root CA certificate, no additional wireless client configuration is required. If users purchase IAS server certificates for which Windows XP does not include a corresponding root CA certificate, they must install the root CA certificate on each wireless client.

Cisco PEAP

Cisco PEAP supports client authentication by One-Time Password support (OTP) and logon passwords. This allows support for OTP databases from vendors such as RSA Security and Secure Computing Corporation, and also supports logon password databases like LDAP, Novell NDS, and Microsoft databases. In addition, the Cisco PEAP client can protect user name identities until the TLS encrypted tunnel is established. This provides additional assurance that user names are not being broadcast during the authentication phase.

3.4. PROBLEMS WITH CERTIFICATE BASED METHODS

Despite the many advantages of certificate-based EAP types, there are some disadvantages as well.

3.4.1. Cost of Administration

The biggest down side to certificates is the cost of administration. All of the methods in this family require the authenticator to have a public key certificate signed by an authority that is recognized by the clients (the users' devices). This requires network administrators either to purchase server certificates from a commercial certificate authority (CA) or to acquire the software and expertise to create their own. Next, each device that will access the network must be configured to recognize the certificates of the authenticator and the CA. The EAP-TLS method requires all the user devices to have certificates as well. This significantly increases the cost of administration. Not only do certificates have to be created or purchased for each user device, but distribution can be a problem as well – there must be a method of securely installing the certificates on the user devices. Also, it can be difficult to maintain a Certificate Revocation List (CRL) so that the authenticator will know which certificates are good and which are not.

3.4.2. Lengthy Protocol Exchange

A second disadvantage of using a certificate-based EAP method is the number of sequential protocol exchanges (round trips) that are required between the user client and the authenticator in order to complete the authentication. For example, to authenticate a single user via EAP-MD5 protected by PEAP requires six round trips between the user station and the authenticator. Requiring a large number of protocol exchanges both lengthens the authentication delay for the user and uses more computing resources on the authenticator. Because the authentication delay is a particular problem for mobile users who must be reauthenticated when moving from one access point to another and who require a seamless handoff so as not to disrupt ongoing sessions, these methods all permit use of the TLS session resumption feature. This mitigates the handoff problem, but does not help the initial authentication.

3.4.3. Authenticates the Device Instead of the User or Requires a Smart Card

A third disadvantage is that the certificate must either be stored on the user device or on a smart card that the user carries. When certificates are stored on the user's device, it is the device that is authenticated rather than the individual user. In environments where the device cannot be sufficiently secured or where many individuals use the device, it is important to authenticate each individual user. A smart card is a way users can carry their certificates with them, but they are a source of inconvenience and require all the devices to have a card interface.

4. Password Authentication Methods

Although password authentication methods are more convenient than certificate-based methods, they still have vulnerabilities. They are specifically vulnerable to offline dictionary attacks, where an attacker can select guesses from a cracker's "dictionary" of possible passwords.

4.1.1. LEAP and Cisco CCX

LEAP is Cisco's Lightweight Extensible Authentication Protocol, and is based on mutual authentication, which means that both the user and the access point must be authenticated before access onto the corporate network is allowed. Mutual authentication protects against unauthorized (or "rogue") access points attempting to gain entry into the network. Cisco LEAP is based on a username/password scheme and is proprietary to Cisco access points. Cisco CCX (Cisco Compatible Extensions Program) provides assurance of compatibility between Cisco Aironet wireless infrastructure products and wireless client devices from third-party companies. This helps to maintain compatibility with Cisco features and protocols, including LEAP.

4.1.2. LEAP

With Cisco's LEAP, security keys change dynamically with every communications session, preventing an attacker from collecting the packets required to decode data. The new keys generated through LEAP use a shared secret key method between the user and the access point. Because LEAP is proprietary to Cisco, it can be used only with a Cisco access point. LEAP also adds another level of security to the network by authenticating all connections to the network before allowing traffic to pass to a wireless device. Using constantly changing secret keys coupled with user authentication provides additional security for wireless data.

4.1.3. Strong Password Authentication Methods

In response to the cost and inconvenience of using certificate-based authentication methods, security researchers have developed a whole new family of authentication methods based on the use of passwords, but addressing all the deficiencies of traditional password methods. We will use the term strong password to refer to this family.

The main benefit of the strong password methods is that two parties can prove to each other that they both know a secret without revealing that secret to a third party who may be listening in on the conversation. In fact, they neither reveal the secret nor make it easier for the attacker to discover the secret. Strong password methods achieve strong authentication by using a small, easily remembered password.

At the core of these methods is a Diffie-Hellman exchange. A Diffie-Hellman exchange permits two parties to create encryption keys in such a way that an observer watching the entire session will not be able to learn the keys. Diffie-Hellman exchanges take place between web browsers and online merchants, for example, in order to encrypt personal information such as credit card numbers. If the customer and merchant have never done business before, how are they to agree on an encryption key without third parties who may be eavesdropping on the session finding out what it is? Diffie-Hellman supplies the solution.

4.1.4. The Power of SPEKE

The SPEKE method uses a series of random-looking messages exchanged between devices. SPEKE modules perform computations with these messages, then determine whether the password used at the other device was correct. When the passwords match, SPEKE puts out a shared key for each device.

To a third-party observer, SPEKE messages look like random numbers and cannot be used to verify any guesses as to what the password might be. SPEKE's additional power comes from the public key computations that are central to this method. There is no need for any long-lived public keys, private keys, or any sensitive data other than the password. SPEKE uses the Zero Knowledge Password Proof (ZKPP) authentication method to securely transmit passwords, which prevents revealing information to any participant unless they use the exact password in the protocol.

Because of this, SPEKE makes password-based authentication stronger and safer. With SPEKE, even a small or poorly chosen password receives greater protection from attack. Other security characteristics of SPEKE include:

- Strong, unlimited length of key can be negotiated
- Protection from off-line attacks that crack hash-based challenge/response methods
- Client and server are authenticated simultaneously
- No other security infrastructure requirements
- No client or server certificates are required
- Complete benefits of modern cryptography using an ordinary small password

Ease of Use
To implement SPEKE, users perform a one-time setup when installing the device driver or contacting an access point for the first time. There is no need for additional infrastructure (unlike TLS and other 802.1x authentication alternatives) to get the same level of authentication, and can be built into simple wireless access point devices.

SPEKE vs. LEAP
Cisco LEAP (Lightweight Extensible Authentication Protocol) is a proprietary protocol that may be used with Cisco access points only. It is a derivative of EAP, providing mutual authentication between client and server, but is proprietary at the access point level of the network. SPEKE is access point independent and will work with any 802.1x compliant access point. This provides maximum flexibility for mixed networks or networks that do not exclusively use Cisco WLAN infrastructure.

SPEKE vs. PEAP
Protected EAP (PEAP) provides support for one-time token authentication, password change and expire support, and database extensibility to support LDAP/NDS directories. PEAP encrypts the conversation between the EAP client and the server, and security is maintained by using a TLS channel. Mutual authentication is required between the EAP client and the server. SPEKE, however, does not require using tokens or certificates, and provides simultaneous authentication. Passwords are exchanged securely, without revealing information to third parties, and there is no need for a TLS channel.

5. Conclusion

Securing your wireless network provides tremendous cost savings, productivity benefits, and a competitive market advantage. It’s not a question of whether enterprises will require wireless network security, but when. Choosing the highest level of security available is a good investment, because security breaches can be a significant expense. Most attacks go unnoticed, and enterprises can be vulnerable to damages. Security breaches such as stolen information, corrupt data, and network downtime can be expensive. They can also result in consequential damages, such as those resulting from increasing a competitor’s position or market share at the expense of your future revenues and profitability. The cost can be both significant and recurring.

In table 1, we compare several families of EAP methods we have considered in this tutorial: legacy, certificate, password, and strong password. For an explanation of the requirements and features found in the left hand column, see

Section 2. As shown in the table, older EAP methods such as EAP-MD5 are not suitable for wireless authentication because they do not meet all the requirements.

Both the certificate-based methods and the strong password methods meet the basic requirements and may be used on wireless networks. Certificate-based methods possess some special properties that may be valuable in some environments, such as the ability to protect and augment legacy methods that may already be in use. However, the password method is much easier to set up and administer.

The SPEKE method fits especially well into environments where certificates are not practical; such as for SOHO users and public hot spots. SOHO users will find SPEKE is easy to implement and low cost. Carriers and service providers will find SPEKE very flexible, since it is not proprietary to specific infrastructures. SPEKE can be implemented easily into SOHO and hot spot environments where client distribution can be controlled and managed, because clients can be downloaded from a website or provided on an installation CD with the access points.

Note: Readers who are interested in the technical aspects of EAP-SPEKE should read APPENDIX A.

	Legacy EAP Methods (EAP-MD5)	**Certificate (TLS, TTLS, PEAP)**	**Password (LEAP)**	**Strong Password (SPEKE, etc.)**
Must Haves				
Mutual	No	Yes	Yes	Yes
Self Protecting	Yes	Yes	Yes	Yes
Immune to dictionary attacks	Only with long, randomly generated passwords	Yes	No	Yes
Produces session keys	No	Yes	Yes	Yes
Credential Security	None	Strong	Weak	Strong
Should Haves				
Authenticates User	Not with long, randomly generated passwords	Not if cert is stored on disk	Yes	Yes
Foreward Secrecy	N/A	Not with commonly used cipher suites	Yes	Yes
Quick and efficient	Yes	No	Yes	Yes
Low maintenance cost	Yes	No	Yes	Yes
Convenient for users	Yes	Only if cert is stored on disk	Yes	Yes
Broad AP Support	Yes	Yes	No	Yes
May Haves				
Augments legacy	N/A	Yes	No	No
Fast Reauthentication	No, must go to home domain	Yes	No	No, must go to home domain

Table 1 - Comparison of EAP Methods

6. Appendix A

The SPEKE method relies on exponentiation involving large random numbers modulo a large prime number. The exponentiation operator can be considered a one-way function due to the difficulty of calculating discreet logarithms (the inverse of exponentiation). Each party – the user station and the authenticator – will work from its knowledge of the user's password p to calculate a common master session key K. To do this, each party generates a large random number. The user station generates the value a, and the authenticator generates the value b. Each party only knows one value; no one ever knows both. The password p, which is known to both the user and the authenticator, is small and easy to remember.

In the discussion below, we consider two parties, the user (or the user's device) and the authenticator. In wireless networks, the authenticator is the access point (AP). In practice, however, there is often a third party – a backend authentication server which the AP consults. In that case it is the authentication server that actually plays the role of authenticator in the discussion below with the AP acting in a pass-through role. The AP kicks off the EAP conversation by sending an EAP-Identity Request to the user station. The user station replies with the user's identity as shown in fig. 2. If a backend authentication server is involved, the AP forwards the user's EAP-Identity Response on to the authentication server in its initial access request. From that point on, the authentication server takes over and conducts the EAP-SPEKE conversation which which consists of two more exchanges.

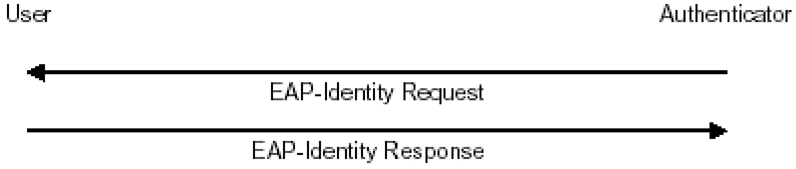

Fig. 2 – EAP Identity Request/Response

When the authenticator (be it the AP or a backend authentication server) receives the user's EAP-Identity Response, it looks up the user in its access repository and retrieves the user's password p. Next, the authenticator creates a large random number b and calculates

$$\mathbf{B = p^{2b} \bmod m}$$

where B is an intermediate value and m is a large prime number used as the modulus. The authenticator sends m and B to the user station in an EAP-SPEKE

Request message. The user station creates another large random number “a” and calculates

$$\mathbf{A = p^{2a} \bmod m}$$

Next the user station calculates

$$\mathbf{K = B^{a} \bmod m}$$

where K is the user station’s calculation of the master session key and B is the value received from the authenticator. Finally the user station calculates

$$\mathbf{Proof_{AK} = h\ (\text{“A”}\ |\ A\ |\ K)}$$

where **Proof$_{AK}$** is the proof that A knows K, h is a secure, one-way hash function, “A” is the ASCII string containing a capital A and | is the concatenation operator. The user station now sends an EAP response to the Authenticator containing A and **Proof$_{AK}$**. This EAP Request/Response pair is shown in fig. 3.

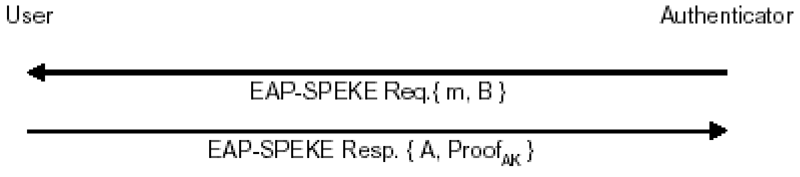

Fig. 3 – First EAP-SPEKE Request/Response

When the authenticator receives the response to its first EAP-SPEKE Request, it calculates

$$\mathbf{K = A^{b} \bmod m}$$

where K is the authenticator’s calculation of the master session key and A is the value received from the user. Next the authenticator calculates

$$\mathbf{Test_{AK} = h\ (\text{“A”}\ |\ A\ |\ K)}$$

and

$$\mathbf{Proof_{BK} = h\ (\text{“B”}\ |\ B\ |\ K)}$$

Now the authenticator compares **Test$_{AK}$** to the value **Proof$_{AK}$** received from the user station. If they are not equal, the authenticator signals failure. If they are equal, the authenticator sends a second EAP-SPEKE Request to the user station as shown in fig. 4.

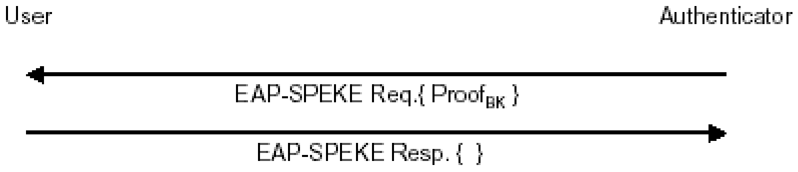

Fig. 4 – Second EAP-SPEKE Request/Response

When the user station receives the second EAP-SPEKE Request, it calculates

$$\mathbf{Test_{BK} = h\ (``B" \mid B \mid K)}$$

from values received or calculated earlier. The user station compares **Test$_{BK}$** to the value **Proof$_{BK}$** received from the authenticator. If they are not equal, the user station aborts the attempted session. If they are equal, the user station returns an empty EAP-SPEKE Response to the authenticator to signal that it is satisfied with the authentication.

When the authenticator receives the empty response, it returns an EAP Success message to the user station as a final signal that the authentication succeeded.

Note that the session key K is independently calculated by each party. This works due to the associative property of exponentiation. It is computationally infeasible for an attacker to work backward from the values A or B to calculate p due to the difficulty of calculating discrete logarithms, the inverse function to exponentiation. Note also that if p is compromised, an attacker listening in on an authentication between the user and the authenticator is still unable to calculate K. To calculate K he would need to know the values of both a and b which are very large random numbers. Neither can the attacker work backward from **Proof$_{AK}$** or **Proof$_{BK}$** to calculate K because h is a one-way hash function. This inability to calculate K even if p is known is what gives SPEKE the property of forward secrecy defined.

The master session key K itself is not used as a WEP or TKIP key to encrypt the wireless data session being established. Those keys are derived from K using a key derivation function.

Self-Test

1. Today's wireless network's present new security problems not seen on legacy dial-up networks because _______________.

    a. Without a physical connection the user has no way of knowing whether he is connecting to the intended network or a rogue access point.

    b. The radio network can be monitored by anyone with a radio receiver

    c. Wireless connections are not stable

    d. a and b only

    e. All of the above

2. Wireless networks can only be secured using public key certificates.

    a. True

    b. False

3. A wireless authentication protocol must include mutual authentication where the authenticator is authenticated to the user in order to guard against rogue access points.

    a. True

    b. False

4. Limiting the number of online authentication attempts is sufficient to thwart dictionary attacks.

    a. True

    b. False

5. Session keys provide ____________.

    a. Message authentication

    b. Message confidentiality

    c. Message integrity

    d. All of the above

6. It is preferable to authenticate the user rather than the user device.

a. True

b. False

7. EAP is the authentication protocol specified in IEEE 802.1x and supports a variety of authentication methods.

    a. True

    b. False

8. _______________ is an EAP method that uses public key certificates to authenticate both the client and server.

    a. EAP-TLS

    b. EAP-TTLS

    c. PEAP

    d. LEAP

9. ______________ is an EAP method that uses a TLS tunnel to protect legacy authentication protocols after using a public key certificates to authenticate the server.

    a. EAP-TLS

    b. EAP-TTLS

    c. PEAP

    d. LEAP

10. _____________ is an EAP method that uses a TLS tunnel to protect other EAP methods and thereby add features such as key exchange and fast reconnect.

    a. EAP-TLS

    b. EAP-TTLS

    c. PEAP

    d. LEAP

11. Some PEAP implementations are limited by _____________.

a. The ability of access points to support them.

b. Which backend databases can be used to store user credentials.

c. The uniqueness of the keys generated.

d. None of the above.

12. A disadvantage of certificate-based authentication is ______________.

a. Cost of administration

b. Lengthy protocol exchanges

c. Authentication of the user device instead of the user

d. All of the above

13. ________________ is a proprietary EAP method only supported by a single vendor's access points.

a. EAP-TTLS

b. PEAP

c. LEAP

d. None of the above

14. Strong password methods make it possible to achieve strong authentication using small easily remembered passwords.

  a. True

  b. False

15. SPEKE is a strong password authentication method whose advantage is _______________.

  a. Ease of use since no certificates are required

  b. Access point independent

  c. Protection against off-line dictionary attacks

  d. All of the above

Correct Answers

1. Today's wireless network's present new security problems not seen on legacy dial-up networks because _______________.

  a. Without a physical connection the user has no way of knowing whether he is connecting to the intended network or a rogue access point.

  b. The radio network can be monitored by anyone with a radio receiver

  c. Wireless connections are not stable

  d. a and b only

  e. All of the above

2. Wireless networks can only be secured using public key certificates.

  a. True

  b. False

3. A wireless authentication protocol must include mutual authentication where the authenticator is authenticated to the user in order to guard against rogue access points.

    a. True

    b. False

4. Limiting the number of online authentication attempts is sufficient to thwart dictionary attacks.

    a. True

    b. False

5. Session keys provide ____________.

    a. Message authentication

    b. Message confidentiality

    c. Message integrity

    d. All of the above

6. It is preferable to authenticate the user rather than the user device.

    a. True

    b. False

7. EAP is the authentication protocol specified in IEEE 802.1x and supports a variety of authentication methods.

    a. True

    b. False

8. ______________ is an EAP method that uses public key certificates to authenticate both the client and server.

    a. EAP-TLS

    b. EAP-TTLS

    c. PEAP

    d. LEAP

9. _____________ is an EAP method that uses a TLS tunnel to protect legacy authentication protocols after using a public key certificates to authenticate the server.

    a. EAP-TLS

    b. EAP-TTLS

    c. PEAP

    d. LEAP

10. ____________ is an EAP method that uses a TLS tunnel to protect other EAP methods and thereby add features such as key exchange and fast reconnect.

    a. EAP-TLS

    b. EAP-TTLS

    c. PEAP

    d. LEAP

11. Some PEAP implementations are limited by ____________.

    a. The ability of access points to support them.

    b. Which backend databases can be used to store user credentials.

    c. The uniqueness of the keys generated.

    d. None of the above

12. A disadvantage of certificate-based authentication is _____________.

a. Cost of administration

b. Lengthy protocol exchanges

c. Authentication of the user device instead of the user

d. All of the above

13. _______________ is a proprietary EAP method only supported by a single vendor's access points.

a. EAP-TTLS

b. PEAP

c. LEAP

d. None of the above

14. Strong password methods make it possible to achieve strong authentication using small easily remembered passwords.

a. True

b. False

15. SPEKE is a strong password authentication method whose advantage is ______________.

a. Ease of use since no certificates are required

b. Access point independent

c. Protection against off-line dictionary attacks

d. All of the above

Glossary

Acronyms Guide

AP
access point - the network access device for an 802.11 wireless network. It contains a radio receiver/transmitter. It may be an 802.1x authenticator.

CA
certification authority - an entity that issues digital certificates (especially X.509 certificates) and vouches for the binding between the data items in a certificate.

CRL
certificate revocation list - a data structure that enumerates digital certificates that have been invalidated by their issuer prior to when they were scheduled to expire.

EAP
extensible authentication protocol - a protocol used between a user station and an authenticator or authentication server. It acts as a transport for authentication methods or types. It, in turn may be encapsulated in other protocols, such as 802.1x and RADIUS.

EAP-LEAP
(Lightweight Extensible Authentication Protocol) - is a Cisco proprietary EAP-Type. It is designed to overcome some basic wireless authentication concerns through Mutual Authentication and the use of dynamic WEP keys.

EAP-PEAP
(Protected Extensible Authentication Protocol) - is a two-phase authentication like EAP-TLS. In the first phase the Authentication Server is authenticated to the Supplicant using an X.509 certificate. Using TLS, a secure channel is established through which any other EAP-Type can be used to authenticate the Supplicant to the Authentication Server during the second phase. A certificate is only required at the Authentication Server. EAP-PEAP also supports identity hiding where the Authenticator is only aware of the anonymous username used to establish the TLS channel during the first phase but not the individual user authenticated during the second phase.

EAP-TLS
(Transport Layer Security) - is an EAP-Type for authentication based upon X.509 certificates. Because it requires both the Supplicant and the Authentication Server to have certificates, it provides explicit Mutual Authentication and is resilient to man-in-the-middle attacks. After successful authentication a secure TLS link is established to securely communicate a unique session key from the Authentication Server to the Authenticator. Because X.509 certificates are required on the Supplicant, EAP-TLS presents significant management complexities.

EAP-TTLS
(Tunneled TLS) - is an EAP-Type for authentication that employs a two-phase authentication process. In the first phase the Authentication Server is authenticated to the Supplicant using an X.509 certificate. Using TLS, a secure channel is established through which the Supplicant can be authenticated to the Authentication Server using legacy PPP authentication protocols such as PAP, CHAP, and MS-CHAP. EAP-TTLS has the advantage over EAP-TLS that it only requires a certificate at the Authentication Server. It also makes possible forwarding of Supplicant requests to a legacy RADIUS server. EAP-TTLS also supports identity hiding where the Authenticator is only aware of the anonymous username used to establish the TLS channel during the first phase but not the individual user authenticated during the second phase.

SPEKE
Simple Password-authenticated Exponential Key Exchange - an authentication method, based on a Diffie-Hellman key exchange, that provides strong authentication using small passwords. SPEKE does not require a certificate for either client or server. SPEKE protects passwords and user information during the authentication dialog, allowing customers to take advantage of existing password models. It may be implemented as an EAP method, and does not require any PKI support or certificate infrastructure.

TKIP
Temporal Key Integrity Protocol a protocol being considered for standardization in the draft IEEE 802.11i standard as a replacement for WEP. It has been endorsed by the Wi-Fi Alliance for use in Wi-Fi Protected Access (WPA).

WEP
Wired Equivalent Privacy - a protocol utilized by the IEEE 802.11 standard for protecting the session between a user station and an Access Point. Since the publication of IEEE 802.11-1999, WEP has been demonstrated to be easily crackable.

ZKPP
Zero Knowledge Password Proof - the process by which strong password authentication methods may enable two parties to prove to each other that they know a password without revealing anything about the password to an eavesdropper listening in on the exchange.

Definitions

802.1X
The IEEE 802.1X standard, Port Based Network Access Control, defines a mechanism for port-based network access control that makes use of the physical access characteristics of IEEE 802 LAN infrastructure. It provides a means of authenticating and authorizing devices attached to a LAN port that has point-to-point connection characteristics. The 802.1X specification includes a number of features aimed specifically at supporting the use of Port Access Control in IEEE 802.11 Wireless LANs (WLANs). These include the ability for a WLAN Access Point to distribute or obtain global key information to/from attached stations, following successful authentication.

authentication
the process of verifying a claimed identity.

authentication server
in 802.1x, an entity that provides an authentication service to an authenticator. This service determines, from the credentials provided by the supplicant, whether the supplicant is authorized to access the services provided by the authenticator.

authenticator
in 802.1x, an entity at one end of a point-to-point LAN segment that facilitates authentication of the entity attached to the other end of that link.

authorization
the process of granting permission to access and utilize a network service.

Diffie-Hellman key exchange
The Diffie-Hellman key agreement protocol (also called exponential key agreement) was developed by Diffie and Hellman in 1976 and published in the groundbreaking paper "New Directions in Cryptography." The protocol allows two users to exchange a secret key over an insecure medium without any prior secrets. Interlink Networks' implementation of the SPEKE authentication method uses a hash of the password as the Diffie-Hellman generator. This prevents man-in-the-middle attacks.

rogue access point
any access point that is operated by some party other than the service provider who operates a local network and that impersonates an access point operated by the service provider.

strong password authentication methods
any of a family of authentication methods that provide strong authentication using small passwords.

supplicant
in 802.1x, an entity at one end of a point-to-point LAN segment that is being authenticated by an authenticator attached to the other end of that link.

user
a person or software process that accesses network services and uses network resources.

user station
the system or device by which a user accesses a network service.