



System Configuration: Authentication and Certificates

This chapter addresses authentication and certification features in the System Configuration section of Cisco Secure Access Control Server Release 4.0 for Windows, hereafter referred to as ACS.

This chapter contains the following topics:

- [About Certification and EAP Protocols, page 10-1](#)
- [Global Authentication Setup, page 10-19](#)
- [ACS Certificate Setup, page 10-26](#)

About Certification and EAP Protocols

ACS uses Extensible Authentication Protocol-Transport Layer Security (EAP-TLS), Extensible Authentication Protocol-Flexible Authentication via Secure Tunneling (EAP-FAST), and Protected Extensible Authentication Protocol (PEAP) authentication protocols in combination with digital certification to ensure the protection and validity of authentication information. Digital certification, EAP-TLS, PEAP, EAP-FAST, and machine authentication are described in the topics that follow.

This section contains the following topics:

- [Digital Certificates, page 10-1](#)
- [EAP-TLS Authentication, page 10-2](#)
- [PEAP Authentication, page 10-5](#)
- [EAP-FAST Authentication, page 10-8](#)

Digital Certificates

You use the ACS Certificate Setup pages to install digital certificates to support EAP-TLS, EAP-FAST, and PEAP authentication, as well as to support Secure HyperText Transfer Protocol (HTTPS) protocol for secure access to the ACS web interface. ACS uses the X.509 v3 digital certificate standard. Certificate files must be in Base64-encoded X.509 format or Distinguished Encoding Rules (DER)-encoded binary X.509 format. Also, ACS supports manual certificate enrollment and provides the means for managing a certificate trust list (CTL) and certificate revocation lists (CRL).

Digital certificates do not require the sharing of secrets or stored database credentials. They can be scaled and trusted over large deployments. If managed properly, they can serve as a method of authentication that is stronger and more secure than shared secret systems. Mutual trust requires that ACS have an installed certificate that can be verified by end-user clients. This server certificate may be issued from a certification authority (CA) or, if you choose, may be a self-signed certificate. For more information, see [Installing an ACS Server Certificate, page 10-26](#), and [Using Self-Signed Certificates, page 10-33](#).

**Note**

Depending on the end-user client involved, the CA certificate for the CA that issued the ACS server certificate is likely to be required in local storage for trusted root CAs on the end-user client computer.

EAP-TLS Authentication

This section contains the following topics:

- [About the EAP-TLS Protocol, page 10-2](#)
- [EAP-TLS and ACS, page 10-3](#)
- [EAP-TLS Limitations, page 10-4](#)
- [Enabling EAP-TLS Authentication, page 10-4](#)

About the EAP-TLS Protocol

EAP and TLS are Internet Engineering Task Force (IETF) RFC standards. The EAP protocol carries initial authentication information, specifically the encapsulation of EAP over LANs (EAPOL) as established by IEEE 802.1X. TLS uses certificates for user authentication and dynamic ephemeral session key generation. The EAP-TLS authentication protocol uses the certificates of ACS and of the end-user client, enforcing mutual authentication of the client and ACS. For more detailed information on EAP, TLS, and EAP-TLS, refer to the following IETF RFCs: [PPP Extensible Authentication Protocol \(EAP\) RFC 2284](#), [The TLS Protocol RFC 2246](#), and [PPP EAP TLS Authentication Protocol RFC 2716](#).

EAP-TLS authentication involves two elements of trust:

- The EAP-TLS negotiation establishes end-user trust by validating, through RSA signature verifications, that the user possesses a keypair that a certificate signs. This process verifies that the end user is the legitimate keyholder for a given digital certificate and the corresponding user identification in the certificate. However, trusting that a user possesses a certificate only provides a username-keypair binding.
- Using a third-party signature, usually from a CA, that verifies the information in a certificate. This third-party binding is similar to the real-world equivalent of the stamp on a passport. You trust the passport because you trust the preparation and identity-checking that the particular country's passport office made when creating that passport. You trust digital certificates by installing the root certificate CA signature.

Some situations do not require this second element of trust that is provided by installing the root certificate CA signature. When such external validation of certificate legitimacy is not required, you can use the ACS self-signed certificate capability. Depending on the end-user client involved, the CA certificate for the CA that issued the ACS server certificate is likely to be required in local storage for trusted root CAs on the end-user client computer. For more information, see [About Self-Signed Certificates, page 10-34](#).

EAP-TLS requires support from the end client and the Authentication, Authorization, and Accounting (AAA) client. An example of an EAP-TLS client includes the Microsoft Windows XP operating system.

EAP-TLS-compliant AAA clients include:

- Cisco 802.1x-enabled switch platforms (such as the Catalyst 6500 product line)
- Cisco Aironet Wireless solutions

To accomplish secure Cisco Aironet connectivity, EAP-TLS generates a dynamic, per-user, per-connection, unique session key.

EAP-TLS and ACS

ACS supports EAP-TLS with any end-user client that supports EAP-TLS, such as Windows XP. To learn which user databases support EAP-TLS, see [Authentication Protocol-Database Compatibility, page 1-7](#). For more information about deploying EAP-TLS authentication, see *Extensible Authentication Protocol Transport Layer Security Deployment Guide for Wireless LAN Networks* at http://www.cisco.com/en/US/products/hw/wireless/ps430/products_white_paper09186a008009256b.shtml

ACS can use EAP-TLS to support machine authentication to Microsoft Windows Active Directory. The end-user client may limit the protocol for user authentication to the same protocol that is used for machine authentication; that is, use of EAP-TLS for machine authentication may require the use of EAP-TLS for user authentication. For more information about machine authentication, see [Machine Authentication, page 13-11](#).

ACS supports domain stripping for EAP-TLS authentication by using Windows Active Directory. For more information, see [EAP-TLS Domain Stripping, page 13-10](#).

ACS also supports three methods of certificate comparison and a session resume feature. This topic discusses these features.

To permit user access to the network or computer authenticating with EAP-TLS, ACS must verify that the claimed identity (presented in the EAP Identity response) corresponds to the certificate that the user presents. ACS can accomplish this verification in three ways:

- **Certificate SAN Comparison**—Based on the name in the Subject Alternative Name field in the user certificate.
- **Certificate CN Comparison**—Based on the name in the Subject Common Name field in the user certificate.
- **Certificate Binary Comparison**—Based on a binary comparison between the user certificate in the user object in the LDAP server or Active Directory and the certificate that the user presents during EAP-TLS authentication. This comparison method cannot be used to authenticate users who are in an ODBC external user database.



Note

If you use certificate binary comparison, the user certificate must be stored in a binary format. Also, for generic LDAP and Active Directory, the attribute that stores the certificate must be the standard LDAP attribute named **usercertificate**.

When you set up EAP-TLS, you can select the criterion (one, two, or all) that ACS uses. For more information, see [Configuring Authentication Options, page 10-19](#).

ACS supports a session resume feature for EAP-TLS-authenticated user sessions, which is a particularly useful feature for WLANs, wherein a user may move the client computer to set a different wireless access point. When this feature is enabled, ACS caches the TLS session that is created during EAP-TLS authentication, provided that the user successfully authenticates. If a user needs to reconnect and the

original EAP-TLS session has not timed out, ACS uses the cached TLS session, resulting in faster EAP-TLS performance and lessened AAA server load. When ACS resumes an EAP-TLS session, the user reauthenticates by a secure sockets layer (SSL) handshake only, without a certificate comparison.

In effect, enabling an EAP-TLS session resume allows ACS to trust a user based on the cached TLS session from the original EAP-TLS authentication. Because ACS only caches a TLS session when a new EAP-TLS authentication succeeds, the existence of a cached TLS session is proof that the user has successfully authenticated in the number of minutes that the EAP-TLS session timeout option specified.

**Note**

Session timeout is based on the time of the initial, full authentication of the session. It does not depend on an accounting start message.

The Session resume feature does not enforce changes to the group assignment in an external user database; because group mapping does not occur when a user session is resumed. Instead, the user is mapped to the same ACS group to which the user was mapped at the beginning of the session. At the start of a new session, group mapping enforces the new group assignment.

To force an EAP-TLS session to end before the session timeout is reached, you can restart the CSAuth service or delete the user from the ACS user database. Disabling or deleting the user in an external user database has no effect because the session resume feature does not involve the use of external user databases.

You can enable the EAP-TLS session resume feature and configure the timeout interval on the Global Authentication Setup page. For more information about enabling this feature, see [Global Authentication Setup, page 10-19](#).

EAP-TLS Limitations

The limitations in the ACS implementation of EAP-TLS are:

- **Server and CA certificate file format**—If you install the ACS server and CA certificates from files, rather than from certificate storage, server and CA certificate files must be in Base64-encoded X.509 format or DER-encoded binary X.509 format.
- **LDAP attribute for binary comparison**—If you configure ACS to perform binary comparison of user certificates, the user certificate must be stored in the Active Directory or an LDAP server by using a binary format. Also, the attribute storing the certificate must be named **usercertificate**.
- **Windows server type**—If you want to use Active Directory to authenticate users with EAP-TLS when ACS runs on a member server, additional configuration is required. For more information, including steps for the additional configuration, see the *Installation Guide for Cisco Secure ACS for Windows*.

Enabling EAP-TLS Authentication

This explains the procedures that are required to configure ACS to support EAP-TLS authentication.

**Note**

You must configure end-user client computers to support EAP-TLS. This procedure is specific to the configuration of ACS only. For more information about deploying EAP-TLS authentication, see *Extensible Authentication Protocol Transport Layer Security Deployment Guide for Wireless LAN Networks* at http://www.cisco.com/warp/public/cc/pd/sqsw/sq/tech/acstl_wp.htm.

Before You Begin

For EAP-TLS machine authentication, if you have configured a Microsoft certification authority server on the domain controller, you can configure a policy in Active Directory to produce a client certificate automatically when a computer is added to the domain. For more information, see the [Microsoft Knowledge Base](#).

To enable EAP-TLS authentication, follow these steps:

-
- Step 1** Install a server certificate in ACS. EAP-TLS requires a server certificate. For detailed steps, see [Installing an ACS Server Certificate, page 10-26](#).



Note If you have previously installed a certificate to support EAP-TLS, or PEAP user authentication, or to support HTTPS protection of remote ACS administration, you do not need to perform this step. A single server certificate is sufficient to support all certificate-based ACS services and remote administration; however, EAP-TLS, EAP-FAST and PEAP require that the certificate be suitable for server authentication purposes.

- Step 2** Edit the certification trust list so that the CA issuing end-user client certificates is trusted. If you do not perform this step, ACS only trusts user certificates that were issued by the same CA that issued the certificate that is installed in ACS. For detailed steps, see [Editing the Certificate Trust List, page 10-28](#).
- Step 3** Establish a certificate revocation list (CRL) for each CA and certificate type in the certificate trust list (CTL). As part of EAP-TLS authentication, ACS validates the status of the certificate presented by the user against the cached CRL to ensure that it has not been revoked. For detailed steps, see [Editing the Certificate Trust List, page 10-28](#).
- Step 4** Enable EAP-TLS on the Global Authentication Setup page. In ACS, you complete this step only after you have successfully completed Step 1. For detailed steps, see [Configuring Authentication Options, page 10-19](#).
- Step 5** Configure a user database. To determine which user databases support EAP-TLS authentication, see [Authentication Protocol-Database Compatibility, page 1-7](#).

ACS is ready to perform EAP-TLS authentication.

PEAP Authentication

This section contains the following topics:

- [About the PEAP Protocol, page 10-6](#)
- [PEAP and ACS, page 10-6](#)
- [PEAP and the Unknown User Policy, page 10-7](#)
- [Enabling PEAP Authentication, page 10-8](#)

About the PEAP Protocol

The PEAP protocol is a client-server security architecture that you use to encrypt EAP transactions; thereby protecting the contents of EAP authentications. IRSA, Cisco, and Microsoft have posted a PEAP IETF Internet Draft that is available at:

<http://tools.ietf.org/html/draft-josefsson-pppext-eap-tls-eap-05>.

PEAP authentications always involve two phases:

- In the first phase, the end-user client authenticates ACS. This action requires a server certificate and authenticates ACS to the end-user client, ensuring that the user or machine credentials sent in phase two are sent to a AAA server that has a certificate issued by a trusted CA. The first phase uses a TLS handshake to establish an SSL tunnel.



Note Depending on the end-user client involved, the CA certificate for the CA that issued the ACS server certificate is likely to be required in local storage for trusted root CAs on the end-user client computer.

- In the second phase, ACS authenticates the user or machine credentials by using an EAP authentication protocol. The SSL tunnel that was created in phase one protects the EAP authentication. The authentication type that is negotiated during the second conversation may be any valid EAP type, such as EAP-GTC (for Generic Token Card). Because PEAP can support any EAP authentication protocol, individual combinations of PEAP and EAP protocols are denoted with the EAP protocol in parentheses, such as PEAP (EAP-GTC). For the authentication protocols that ACS supports in phase two of PEAP, see [Authentication Protocol-Database Compatibility, page 1-7](#).

One improvement in security that PEAP offers is identity protection. This improvement is the potential of protecting the username in all PEAP transactions. After phase one of PEAP, all data is encrypted, including username information that is usually sent in clear text. The Cisco Aironet PEAP client sends user identity through the SSL tunnel only. The initial identity, used in phase one and which is sent in the clear, is the MAC address of the end-user client with **PEAP_** as a prefix. The Microsoft PEAP client does not provide identity protection; the Microsoft PEAP client sends the username in clear text in phase one of PEAP authentication.

PEAP and ACS

ACS supports PEAP authentication by using the Cisco Aironet PEAP client or the Microsoft PEAP client that is included with Microsoft Windows XP Service Pack 1. ACS can support the Cisco Aironet PEAP client with PEAP(EAP-GTC) only. For the Microsoft PEAP client in the Windows XP Service Pack 1, ACS supports only PEAP(EAP-MS-CHAPv2). For information about which user databases support PEAP protocols, see [Authentication Protocol-Database Compatibility, page 1-7](#).

When the end-user client is the Cisco Aironet PEAP client, and PEAP(EAP-GTC) and PEAP(EAP-MS-CHAPv2) are enabled on the Global Authentication Setup page, ACS first attempts PEAP(EAP-GTC) authentication with the end-user client. If the client rejects this protocol (by sending an EAP NAK message), ACS attempts authentication with PEAP(EAP-MS-CHAPv2). For more information about enabling EAP protocols that PEAP supports, see [Global Authentication Setup, page 10-19](#).

ACS can use PEAP(EAP-MS-CHAPv2) to support machine authentication to Microsoft Windows Active Directory. The end-user client may limit the protocol that is used for user authentication to the same protocol that is used for machine authentication; that is, use of PEAP for machine authentication requires the use of PEAP for user authentication. For more information about machine authentication, see [Machine Authentication, page 13-11](#).

ACS supports a session resume feature for PEAP-authenticated user sessions. When this feature is enabled, ACS caches the TLS session that is created during phase one of PEAP authentication, provided that the user successfully authenticates in phase two of PEAP. If a user needs to reconnect and the original PEAP session has not timed out, ACS uses the cached TLS session, resulting in faster PEAP performance and lessened AAA server load.

**Note**

Session timeout is based on the time that authentication succeeds. It does not depend on accounting.

You can enable the PEAP session resume feature and configure the timeout interval on the Global Authentication Setup page. For more information about enabling this feature, see [Global Authentication Setup, page 10-19](#).

ACS also supports a fast reconnect feature. When the session resume feature is enabled, the fast reconnect feature causes ACS to allow a PEAP session to resume without checking user credentials. In effect, ACS can trust a user based on the cached TLS session from the original PEAP authentication when this feature is enabled. Because ACS only caches a TLS session when phase two of PEAP authentication succeeds, the existence of a cached TLS session is proof that the user has successfully authenticated in the number of minutes that the PEAP session timeout option specifies.

The session resume feature does not enforce changes to group assignment in an external user database; group mapping does not occur when the session resume feature extends a user session. Instead, the user is mapped to the same ACS group that the user was mapped to at the beginning of the session. At the start of a new session, group mapping enforces the new group assignment.

The fast reconnect feature is particularly useful for wireless LANs, wherein a user may move the client computer so that a different wireless access point is in use. When ACS resumes a PEAP session, the user reauthenticates without entering a password, provided that the session has not timed out. If the end-user client is restarted, the user must enter a password; even if the session timeout interval has not ended.

You can enable the PEAP fast reconnect feature on the Global Authentication Setup page. For more information about enabling this feature, see [Global Authentication Setup, page 10-19](#).

PEAP and the Unknown User Policy

During PEAP authentication, ACS might not know the real username to be authenticated until phase two of authentication. While the Microsoft PEAP client does reveal the actual username during phase one, the Cisco PEAP client does not; therefore, ACS does not attempt to look up the username that is presented during phase one and the use of the Unknown User Policy is irrelevant during phase one, regardless of the PEAP client used.

When phase two of PEAP authentication occurs and the username that the PEAP client presents is unknown to ACS, ACS processes the username in the same way that it processes usernames that are presented in other authentication protocols. If the username is unknown and the Unknown User Policy is disabled, authentication fails. If the username is unknown and the Unknown User Policy is enabled, ACS attempts to authenticate the PEAP user with unknown user processing.

For more information about unknown user processing, see [About Unknown User Authentication, page 16-3](#).

Enabling PEAP Authentication

This procedure provides an overview of the detailed procedures that are required to configure ACS to support PEAP authentication.


Note

You must configure end-user client computers to support PEAP. This procedure is specific to configuration of ACS only.

To enable PEAP authentication:

- Step 1** Install a server certificate in ACS. PEAP requires a server certificate. For detailed steps, see [Installing an ACS Server Certificate, page 10-26](#).


Note

If you have previously installed a certificate to support EAP-TLS or PEAP user authentication, or to support HTTPS protection of remote ACS administration, you do not need to perform this step. A single server certificate is sufficient to support all certificate-based ACS services and remote administration; however, EAP-TLS and PEAP require that the certificate be suitable for server authentication purposes.

- Step 2** Enable PEAP on the Global Authentication Setup page. You use ACS to complete this step only after you have successfully completed Step 1. For detailed steps, see [Configuring Authentication Options, page 10-19](#).
- Step 3** Configure a user database. To determine which user databases support PEAP authentication, see [Authentication Protocol-Database Compatibility, page 1-7](#).
ACS is ready to perform PEAP authentication for most users. For more information, see [PEAP and the Unknown User Policy, page 10-7](#).
- Step 4** Consider enabling the Unknown User Policy to simplify PEAP authentication. For more information, see [PEAP and the Unknown User Policy, page 10-7](#). For detailed steps, see [Configuring the Unknown User Policy, page 16-8](#).

EAP-FAST Authentication

This section contains the following topics:

- [About EAP-FAST, page 10-9](#)
- [About Master Keys, page 10-10](#)
- [About PACs, page 10-11](#)
 - [Automatic PAC Provisioning, page 10-13](#)
 - [Manual PAC Provisioning, page 10-14](#)
- [Master Key and PAC TTLs, page 10-15](#)
- [Replication and EAP-FAST, page 10-15](#)
- [Enabling EAP-FAST, page 10-17](#)

About EAP-FAST

The EAP Flexible Authentication via Secured Tunnel (EAP-FAST) protocol is a client-server security architecture that encrypts EAP transactions with a TLS tunnel. While similar to PEAP in this respect, it differs significantly in that EAP-FAST tunnel establishment is based on strong secrets that are unique to users. These secrets are called Protected Access Credentials (PACs), which ACS generates by using a master key known only to ACS. Because handshakes based on shared secrets are intrinsically faster than handshakes based on PKI, EAP-FAST is the significantly faster of the two solutions that provide encrypted EAP transactions. No certificate management is required to implement EAP-FAST.

EAP-FAST occurs in three phases:

- **Phase zero**—Unique to EAP-FAST, phase zero is a tunnel-secured means of providing an EAP-FAST end-user client with a PAC for the user requesting network access. (See [Automatic PAC Provisioning, page 10-13](#).) Providing a PAC to the end-user client is the sole purpose of phase zero. The tunnel is established based on an anonymous Diffie-Hellman key exchange. If EAP-MS-CHAPv2 authentication succeeds, ACS provides the user with a PAC. To determine which databases support EAP-FAST phase zero, see [Authentication Protocol-Database Compatibility, page 1-7](#).



Note Phase zero is optional and PACs can be manually provided to end-user clients. (See [Manual PAC Provisioning, page 10-14](#).) You control whether ACS supports phase zero by checking the Allow automatic PAC provisioning check box in the Global Authentication Configuration page.

The Allow anonymous in-band PAC provisioning option provisions an end-user client with a PAC by using EAP-FAST phase zero. If this check box is selected, ACS establishes a secured connection with the end-user client for the purpose of providing the client with a new PAC. This option allows an anonymous TLS handshake between the end-user client and ACS. (EAP-MS-CHAP will be used as inner method only.)

The Allow authenticated in-band PAC provisioning option provisions an end-user client with a PAC by using EAP-FAST phase zero with TLS server-side authentication. This option requires that you install a server certificate and a trusted root CA on ACS.

By default, ACS supports TLS server-side authentication; however, if the client sends the user certificate to ACS, mutual TLS authentication is performed and inner methods are bypassed.

Phase zero of EAP-FAST does not enable a network service; therefore, even a successful EAP-FAST phase zero transaction is recorded in the ACS Failed Attempts log.

If the Accept client on authenticated provisioning option is selected, ACS always sends an Access-Reject at the end of the provisioning phase (phase zero), forcing the client to reauthenticate by using the tunnel PAC. This option sends an Access-Accept to the client and can be enabled only when you check the Allow authenticated in-band PAC provisioning check box.

- **Phase one**—In phase one, ACS and the end-user client establish a TLS tunnel based on the PAC that the end-user client presents. This phase requires that the end-user client has been provided a PAC for the user who is attempting to gain network access and that the PAC is based on a master key that has not expired. The means by which PAC provisioning has occurred is irrelevant; you can use automatic or manual provisioning.

No network service is enabled by phase one of EAP-FAST.

- **Phase two**—In phase two, ACS authenticates the user credentials with EAP-GTC, which is protected by the TLS tunnel that was created in phase one. TLS and MS-CHAP are supported as inner methods. No other EAP types are supported for EAP-FAST. To determine which databases support EAP-FAST phase two, see [Authentication Protocol-Database Compatibility, page 1-7](#).

ACS authorizes network service with a successful user authentication in phase two of EAP-FAST and logs the authentication in the Passed Authentications log, if it is enabled. Also, if necessary, ACS may refresh the end-user client PAC, which creates a second entry in the Passed Authentication log for the same phase two transaction.

**Note**

This version of ACS supports EAP-FAST phase two for NAC phase two and is for wired clients only.

EAP-FAST can protect the username in all EAP-FAST transactions. ACS does not perform user authentication based on a username that is presented in phase one; however, whether the username is protected during phase one depends on the end-user client. If the end-user client does not send the real username in phase one, the username is protected. The Cisco Aironet EAP-FAST client protects the username in phase one by sending `FAST_MAC address` in place of the username. After phase one of EAP-FAST, all data is encrypted, including username information that is usually sent in clear text.

ACS supports password aging with EAP-FAST for users who are authenticated by Windows user databases. Password aging can work with phase zero or phase two of EAP-FAST. If password aging requires a user to change passwords during phase zero, the new password would be effective in phase two. For more information about password aging for Windows user databases, see [Enabling Password Aging for Users in Windows Databases, page 6-19](#).

About Master Keys

EAP-FAST master keys are strong secrets that ACS automatically generates and of which only ACS is aware. Master keys are never sent to an end-user client. EAP-FAST requires master keys for two purposes:

- **PAC generation**—ACS generates PACs by using the active master key. For details about PACs, see [About PACs, page 10-11](#).
- **EAP-FAST phase one**—ACS determines whether the PAC that the end-user client presents was generated by one of the master keys it is aware of: the active master key or a retired master key.

To increase the security of EAP-FAST, ACS changes the master key that it uses to generate PACs. ACS uses time-to-live (TTL) values that you define to determine when it generates a new master key and the age of all master keys. Based on TTL values, ACS assigns master keys one of the these states:

- **Active**—An active master key is the master key used by ACS to generate PACs. The master key TTL setting determines the duration that a master key remains active. At any time, only one master key is active. When you define TTLs for master keys and PACs, ACS permits only a PAC TTL that is shorter than the active master key TTL. This limitation ensures that a PAC is refreshed at least once before the expiration of the master key used to generate the PAC, provided that EAP-FAST users log in to the network at least once before the master key expires. For more information about how TTL values determine whether PAC refreshing or provisioning is required, see [Master Key and PAC TTLs, page 10-15](#).

When you configure ACS to receive replicated EAP-FAST policies and master keys, a backup master key is among the master keys received. The backup master key is used only if the active master key retires before the next successful master key replication. If the backup master key also retires before the next successful master key replication, EAP-FAST authentication fails for all users requesting network access with EAP-FAST.



Tip

If EAP-FAST authentication fails because the active and backup master keys have retired and ACS has not received new master keys in replication, you can force ACS to generate its own master keys by checking the **EAP-FAST Master Server** check box and clicking **Submit**.

ACS records the generation of master keys in the logs for the CSAuth service.

- **Retired**—When a master key becomes older than the master key TTL settings, it is considered retired for the duration that the Retired master key TTL settings specify. ACS can store up to 255 retired master keys. While a retired master key is not used to generate new PACs, ACS needs it to authenticate PACs that were generated by using it. When you define TTLs for master keys and retired master keys, ACS permits only TTL settings that require storing 255 or fewer retired master keys. For example, if the master key TTL is one hour and the retired master key TTL is four weeks, this would require storing up to 671 retired master keys; therefore, an error message appears and ACS does not allow these settings.

When a user gains network access by using a PAC that is generated with a retired master key, ACS provides the end-user client with a new PAC that the active master key generated. For more information about ACS master keys and PACs, see [Master Key and PAC TTLs, page 10-15](#).

- **Expired**—When a master key becomes older than the sum of the master key TTL and retired master TTL settings, it is considered expired and ACS deletes it from its records of master keys. For example, if the master key TTL is one hour and the retired master key TTL is one week, a master key expires when it becomes greater than one week and one hour old.

PACs that an expired master key cannot be used to access your network. An end-user client presenting a PAC that generated an expired master key requires a new PAC by using automatic or manual provisioning before phase one of EAP-FAST can succeed.

About PACs

PACs are strong shared secrets that enable ACS and an EAP-FAST end-user client to authenticate each other and establish a TLS tunnel for use in EAP-FAST phase two. ACS generates PACs by using the active master key and a username.

PAC comprises:

- **PAC-Key**—Shared secret bound to a client (and client device) and server identity.
- **PAC Opaque**—Opaque field that the client caches and passes to the server. The server recovers the PAC-Key and the client identity to mutually authenticate with the client.
- **PAC-Info**—At a minimum includes the server's identify to enable the client to cache different PACs. Optionally, it includes other information such as the PACs expiration time.

An EAP-FAST end-user client stores PACs for each user accessing the network with the client. Additionally, a AAA server that supports EAP-FAST has a unique Authority ID. An end-user client associates a user's PACs with the Authority ID of the AAA server that generated them. PACs remove the need for PKI (digital certificates).

During EAP-FAST phase one, the end-user client presents the PAC that it has for the current user and Authority ID that ACS sends at the beginning of the EAP-FAST transaction. ACS determines whether the PAC was generated using one of the master keys it is aware of: active or retired. (A PAC that is generated by using a master key that has since expired can never be used to gain network access.) When an end-user client has a PAC that is generated with an expired master key, the end-user client must

receive a new PAC before EAP-FAST phase one can succeed. The means of providing PACs to end-user clients, known as PAC provisioning, are discussed in [Automatic PAC Provisioning, page 10-13](#) and [Manual PAC Provisioning, page 10-14](#).

After end-user clients are provided PACs, ACS refreshes them as that master key and PAC TTL values specify. ACS generates and sends a new PAC as needed at the end of phase two of EAP-FAST; however, if you shorten the master key TTL, you might require that PAC provisioning occur. For more information about how master key and PAC states determine whether ACS sends a new PAC to the end-user client at the end of phase two, see [Master Key and PAC TTLs, page 10-15](#).

Regardless of the master key TTL values that you define, a user will require PAC provisioning when the user does not use EAP-FAST to access the network before the master key that generated the user's PAC has expired. For example, if the master key TTL is one week old and the retired master key TTL is one week old, each EAP-FAST end-user client used by someone who goes on vacation for two weeks will require PAC provisioning.

Provisioning Modes

ACS supports out-of-band and in-band provisioning modes. The in-band provisioning mode operates inside an Authenticated Diffie-HellmanKey Agreement Protocol (ADHP) tunnel before the peer authenticates the ACS server.

Since an unauthenticated server is provisioned, it is not possible to use a plain text password; so only MS-CHAP credentials can be used inside the tunnel. MS-CHAPv2 is used to prove the peer's identity and receives a PAC for further authentication sessions. This method minimizes the risk of exposing the user's credentials.

EAP-FAST has been enhanced to support an authenticated tunnel (using the server certificate) inside which PAC provisioning occurs. The new cipher suites that are enhancements to EAP-FAST and specifically the server certificate are used.

Since the server is authenticated as part of setting up the tunnel, weaker EAP methods, such as EAP-GTC can be used inside the tunnel to provide supplicant authentication.

At the end of a provisioning session that uses an authenticated tunnel, network access can be granted; since the server and user have authenticated each other.

ACS supports the following EAP types inside the tunnel for provisioning:

- EAP-GTC
- EAP-MS-CHAPv2
- EAP-TLS

Types of PACs

ACS provisions supplicants with a PAC that contains a shared secret that is used in building a TLS tunnel between the supplicant and ACS. ACS provisions supplicants with PAC that have a wider contextual use.

The following types of PACs are provisioned to ACS, as per server policies:

- **Tunnel (Shared Secret) PAC, user or machine**—Distributed shared secret between the peer and ACS that is used to establish a secure tunnel and convey the policy of what must and can occur in the tunnel. The policy can include EAP methods, TLV exchanges, and identities that are allowed in the tunnel. It is up to the server policy to include what's necessary in PAC to enforce the policy in subsequent authentications that use the PAC. For example, in EAP-FAST Protocol Version 1, user identity I-ID is included as the part of the server policy. It limits the inner EAP methods to be carried only on the user identity that is provisioned. Other types of information can also be included, such

as which EAP method and which cipher suite is allowed, for example. If the server policy is not included in the PAC, then no validation or limitation on the inner EAP methods or user identities inside the tunnel established by use of this PAC. The PAC user of machine contains a type field. The format for the machine will be **host/name of machine**.

- **User Authorization PAC**—Distributed user authentication and authorization result based on a previous authentication. You can use it a with combination of the Tunnel PAC to bypass subsequent user authentication. This result is intended to be short-lived and also controlled by the peer. If any state of the user has changed that will affect the user authentication result (for example, user has logged on or off), the peer should discard it and not use it again. You can use the User Authorization PACs in combination of Tunnel PAC to allow a stateless server session resume as described in [Stateless Session Server Resume, page 10-18](#).
- **Posture PAC**—Distributed posture checking and authorization result based on a previous posture validation. You can use a posture PAC to optimize posture validation in the case of frequent revalidations. This result is specific to the posture validation application and may be used outside the contents of EAP-FAST.

The various means by which an end-user client can receive PACs are:

- **PAC provisioning**—Required when an end-user client has no PAC or has a PAC that is based on an expired master key. For more information about how master key and PAC states determine whether PAC provisioning is required, see [Master Key and PAC TTLs, page 10-15](#).

The two supported means of PAC provisioning are:

- **Automatic provision**—Sends a PAC by using a secure network connection. For more information, see [Automatic PAC Provisioning, page 10-13](#).
- **Manual provision**—Requires that you use ACS to generate a PAC file for the user, copy the PAC file to the computer that is running the end-user client, and import the PAC file into the end-user client. For more information, see [Manual PAC Provisioning, page 10-14](#).
- **PAC refresh**—Occurs automatically when EAP-FAST phase two authentication has succeeded, and master key and PAC TTLs dictate that the PAC must be refreshed. For more information about how master key and PAC states determine whether a PAC is refreshed, see [Master Key and PAC TTLs, page 10-15](#).

PACs have the following two states, which the PAC TTL setting determines:

- **Active**—A PAC younger than the PAC TTL is considered active and can be used to complete EAP-FAST phase one, provided that the master key that was used to generate it has not expired. Regardless of whether a PAC is active, if it is based on an expired master key, PAC provisioning must occur before EAP-FAST phase one can succeed.
- **Expired**—A PAC that is older than the PAC TTL is considered expired. Provided that the master key used to generate the PAC has not expired, an expired PAC can be used to complete EAP-FAST phase one and, at the end of EAP-FAST phase two, ACS will generate a new PAC for the user and provide it to the end-user client.

Automatic PAC Provisioning

Automatic PAC provisioning sends a new PAC to an end-user client over a secured network connection. Automatic PAC provisioning requires no intervention of the network user or a ACS administrator, provided that you configure ACS and the end-user client to support automatic provisioning.

EAP-FAST phase zero requires EAP-MS-CHAPv2 authentication of the user. At successful user authentication, ACS establishes a Diffie-Hellman tunnel with the end-user client. ACS generates a PAC for the user and sends it to the end-user client in this tunnel, along with the Authority ID and Authority ID information about this ACS.

**Note**

Because EAP-FAST phase zero and phase two use different authentication methods (EAP-MS-CHAPv2 in phase zero versus EAP-GTC in phase two), some databases that support phase two cannot support phase zero. Given that ACS associates each user with a single user database, the use of automatic PAC provisioning requires that EAP-FAST users are authenticated with a database that is compatible with EAP-FAST phase zero. For the databases with which ACS can support EAP-FAST phase zero and phase two, see [Authentication Protocol-Database Compatibility, page 1-7](#).

No network service is enabled by phase zero of EAP-FAST; therefore, ACS logs a EAP-FAST phase zero transaction in the Failed Attempts log, including an entry that PAC provisioning occurred. After the end-user client has received a PAC through a successful phase zero, it sends a new EAP-FAST request to begin phase one.

**Note**

Because transmission of PACs in phase zero is secured by MS-CHAPv2 authentication and MS-CHAPv2 is vulnerable to dictionary attacks, we recommend that you limit use of automatic provisioning to initial deployment of EAP-FAST. After a large EAP-FAST deployment, PAC provisioning should be performed manually to ensure the highest security for PACs. For more information about manual PAC provisioning, see [Manual PAC Provisioning, page 10-14](#).

To control whether ACS performs automatic PAC provisioning, you use the options on the Global Authentication Setup page in the System Configuration section. For more information, see [Global Authentication Setup Page, page 10-21](#).

Manual PAC Provisioning

Manual PAC provisioning requires an ACS administrator to generate PAC files, which must then be distributed to the applicable network users. Users must configure end-user clients with their PAC files. For example, if your EAP-FAST end-user client is the Cisco Aironet Client Utility (ACU), configuring the ACU to support EAP-FAST requires that you import a PAC file. For more information about configuring a Cisco ACU, see the applicable configuration guide for your ACU.

You can use manual PAC provisioning to control who can use EAP-FAST to access your network. If you disable automatic PAC provisioning, any EAP-FAST user denied a PAC cannot access the network. If your ACS deployment includes network segmentation, wherein access to each network segment is controlled by a separate ACS, manual PAC provisioning enables you to grant EAP-FAST access on a per-segment basis. For example, if your company uses EAP-FAST for wireless access in its Chicago and Boston offices and the Cisco Aironet Access Points at each of these two offices are configured to use different ACSs, you can determine, on a per-employee basis, whether Boston employees visiting the Chicago office can have wireless access.

**Note**

Replicating EAP-FAST master keys and policies affects the ability to require different PACs per ACS. For more information, see [Table 10-2](#).

While the administrative overhead of manual PAC provisioning is much greater than automatic PAC provisioning, it does not include the risk of sending the PAC over the network. When you first deploy EAP-FAST, using manual PAC provisioning would require a lot of manual configuration of end-user clients; however, this type of provisioning is the most secure means for distributing PACs. We recommend that, after a large EAP-FAST deployment, you should manually perform PAC provisioning to ensure the highest security for PACs.

You can generate PAC files for specific usernames, groups of users, lists of usernames, or all users. When you generate PAC files for groups of users or all users, the users must be known or discovered users and cannot be unknown users. ACS for Windows Server supports the generation of PAC files with `CSUtil.exe`. For more information about generating PACs with `CSUtil.exe`, see [PAC File Generation, page D-25](#).

Master Key and PAC TTLs

The TTL values for master keys and PACs determine their states, as described in [About Master Keys, page 10-10](#) and [About PACs, page 10-11](#). master key and PAC states determine whether someone requesting network access with EAP-FAST requires PAC provisioning or PAC refreshing.

[Table 10-1](#) summarizes ACS behavior with respect to PAC and master key states.

Table 10-1 Master Key versus PAC States

Master key state	PAC active	PAC expired
Master key active	Phase one succeeds. PAC is <i>not</i> refreshed at end of phase two.	Phase one succeeds. PAC is refreshed at end of phase two.
Master key retired	Phase one succeeds. PAC is refreshed at end of phase two.	Phase one succeeds. PAC is refreshed at end of phase two.
Master key expired	PAC provisioning is required. If automatic provisioning is <i>enabled</i> , phase zero occurs and a new PAC is sent. The end-user client initiates a new EAP-FAST authentication request using the new PAC. If automatic provisioning is <i>disabled</i> , phase zero does not occur and phase one fails. You must use manual provisioning to give the user a new PAC.	PAC provisioning is required. If automatic provisioning is <i>enabled</i> , phase zero occurs and a new PAC is sent. The end-user client initiates a new EAP-FAST authentication request using the new PAC. If automatic provisioning is <i>disabled</i> , phase zero does not occur and phase one fails. You must use manual provisioning to give the user a new PAC.

Replication and EAP-FAST

The Database Replication feature supports the replication of EAP-FAST settings, Authority ID, and master keys. Replication of EAP-FAST data occurs only if on the:

- Database Replication Setup page of the primary ACS, under Send, you have checked the EAP-FAST master keys and policies check box.
- Global Authentication Setup page of the primary ACS, you have enabled EAP-FAST and checked the EAP-FAST master server check box.
- Database Replication Setup page of the secondary ACS, under Receive, you have checked the EAP-FAST master keys and policies check box.
- Global Authentication Setup page of the secondary ACS, you have enabled EAP-FAST and unchecked the EAP-FAST master server check box.

EAP-FAST-related replication occurs for three events:

- **Generation of master keys**—A primary ACS sends newly generated active and backup master keys to secondary ACSs. This event occurs immediately after master key generation, provided that you configure the replication properly and it is not affected by replication scheduling on the Database Replication Setup page.

- **Manual replication**—All EAP-FAST components that can be replicated are replicated if you click **Replicate Now** on the Database Replication Setup page of the primary ACS. Some of the replicated components are configurable in the web interface. [Table 10-2](#) shows whether an EAP-FAST component is replicated or configurable.



Note EAP-FAST replication is not included in scheduled replication events.

- **Changes to EAP-FAST settings**—If, on a primary ACS, you change any EAP-FAST configurable components that are replicated, ACS begins EAP-FAST replication. Whether an EAP-FAST component is replicated or configurable is detailed in [Table 10-2](#).

The Database Replication log on the primary ACS records replication of master keys. Entries related to master key replication contain the text `MKEYReplicate`.

Table 10-2 EAP-FAST Components and Replication

EAP-FAST Component	Replicated?	Configurable?
EAP-FAST Enable	No	Yes, on the Global Authentication Setup page.
Master key TTL	Yes	Yes, on the Global Authentication Setup page.
Retired master key TTL	Yes	Yes, on the Global Authentication Setup page.
PAC TTL	Yes	Yes, on the Global Authentication Setup page.
Authority ID	Yes	No, generated by ACS.
Authority ID info	Yes	Yes, on the Global Authentication Setup page.
Client initial message	Yes	Yes, on the Global Authentication Setup page.
Master keys	Yes	No, generated by ACS when TTL settings dictate.
EAP-FAST master server	No	Yes, on the Global Authentication Setup page.
Actual EAP-FAST server status	No	No, determined by ACS.

The EAP-FAST master server setting has a significant effect on EAP-FAST authentication and replication:

- **Enabled**—When you check the EAP-FAST master server check box, the `Actual EAP-FAST server status` is `Master` and ACS ignores the EAP-FAST settings, Authority ID, and master keys it receives from a primary ACS during replication, preferring instead to use master keys that it generates, its unique Authority ID, and the EAP-FAST settings that are configured in its web interface.

Enabling the EAP-FAST master server setting requires providing a PAC from the primary ACS that is different than the PAC from the secondary ACS for the end-user client. Because the primary and secondary ACSs send different Authority IDs at the beginning of the EAP-FAST transaction, the end-user client must have a PAC for each Authority ID. A PAC that the primary ACS generates is not accepted by the secondary ACS in a replication scheme where the EAP-FAST master server setting is enabled on the secondary ACS.



Tip

In a replicated ACS environment, use the EAP-FAST master server feature in conjunction with disallowing automatic PAC provisioning to control EAP-FAST access to different segments of your network. Without automatic PAC provisioning, users must request PACs for each network segment.

- **Disabled**—When you do not check the EAP-FAST master server check box, ACS continues to operate as an EAP-FAST master server until the first time it receives replicated EAP-FAST components from the primary ACS. When *Actual EAP-FAST server status* displays the text *Slave*, ACS uses the EAP-FAST settings, Authority ID, and master keys that it receives from a primary ACS during replication; rather than using the master keys that it generates and its unique Authority ID.



Note When you uncheck the EAP-FAST master server check box, the *Actual EAP-FAST server status* remains *Master* until ACS receives replicated EAP-FAST components and then the *Actual EAP-FAST server status* changes to *Slave*. Until *Actual EAP-FAST server status* changes to *Slave*, ACS acts as a master EAP-FAST server by using master keys that it generates, its unique Authority ID, and the EAP-FAST settings that are configured in its web interface.

Disabling the EAP-FAST master server setting eliminates the need for providing a different PAC from the primary and secondary ACSs. This elimination occurs because the primary and secondary ACSs send the end-user client the same Authority ID at the beginning of the EAP-FAST transaction; therefore, the end-user client uses the same PAC in its response to either ACS. Also, a PAC that one ACS generated for a user in a replication scheme where the EAP-FAST master server setting is disabled is accepted by all other ACSs in the same replication scheme.

For more information about replication, see [ACS Internal Database Replication, page 9-1](#).

Enabling EAP-FAST

This section explains the procedures to configure ACS to support EAP-FAST authentication.



Note You must configure the end-user clients to support EAP-FAST. This procedure is specific to configuring ACS only.

Before You Begin

The steps in this procedure are a suggested order only. Enabling EAP-FAST at your site may require recursion of these steps or performing these steps in a different order. For example, in this procedure, determining how you want to support PAC provisioning comes after configuring a user database to support EAP-FAST; however, choosing automatic PAC provisioning places different limits on user database support.


To enable ACS to perform EAP-FAST authentication:

- Step 1** Configure a user database that supports EAP-FAST authentication. To determine which user databases support EAP-FAST authentication, see [Authentication Protocol-Database Compatibility, page 1-7](#). For user database configuration, see [Chapter 13, “User Databases.”](#)



Note User database support differs for EAP-FAST phase zero and phase two.

ACS supports use of the Unknown User Policy and group mapping with EAP-FAST, as well as password aging with Windows external user databases.

- Step 2** Determine master key and PAC TTL values. While changing keys and PACs more frequently could be considered more secure, it also increases the likelihood that PAC provisioning will be needed for machines left offline so long that the PACs on them are based on expired master keys.
- Also, if you reduce the TTL values with which you initially deploy EAP-FAST, you may force PAC provisioning to occur because users would be more likely to have PACs based on expired master keys.
- For information about how master key and PAC TTL values determine whether PAC provisioning or PAC refreshing is required, see [Master Key and PAC TTLs, page 10-15](#).
- Step 3** Determine whether you want to use automatic or manual PAC provisioning. For more information about the two means of PAC provisioning, see [Automatic PAC Provisioning, page 10-13](#), and [Manual PAC Provisioning, page 10-14](#).
-  **Note** We recommend that you limit the use of automatic PAC provisioning to initial deployments of EAP-FAST, followed by using manual PAC provisioning for adding small numbers of new end-user clients to your network and replacing PACs based on expired master keys.
- Step 4** Using the decisions during [Step 2](#) and [Step 3](#), enable EAP-FAST on the Global Authentication Setup page. For detailed steps, see [Configuring Authentication Options, page 10-19](#).
- ACS is ready to perform EAP-FAST authentication.

Stateless Session Server Resume

To provide better support for server performance, load balancing and peer roaming to different servers, EAP-FAST supports the stateless-server session resume by using the short-lived Authorization PACs. Once a peer establishes a TLS session and is authenticated, the EAP server can provision it with a Tunnel PAC. The tunnel PAC can be used to establish a TLS session much more quickly than a normal TLS handshake. With the normal TLS session resume, the EAP server must maintain the TLS session cache, as well as the peer's authentication and authorization result. This storage requirement often hinders the server's performance, as well as introduces difficulties with server load balancing and peer roaming to different servers. The use of Tunnel PAC eliminates the server's need to maintain a TLS session cache. The TLS session can be quickly established in a fast and secure way; however, the server still has to cache the peer's previous authentication and authorization state for a quick session resume.

You can further optimize by using the User Authorization PAC in combination with the Tunnel PAC. The server generated key protects User Authorization PACs which store previous authentication and authorization states on the peer. If the peer has the authorization PACs corresponding to the EAP server connected (by matching A-ID), and detects no state change affecting the peer, the peer can piggyback the opaque part of these PACs in the PAC-TLV with Client TLS Finished as TLS application data, which the TLS cipher suite that is negotiated protects. This method prevents attackers from snooping the authorization PACs without introducing an extra round trip. Once the EAP server receives and decrypts the authorization PAC, the EAP server can recreate its previous state information based on the peer's authentication and authorization result. If the state information in these PACs is still valid, based on a server side policy, it might bypass one or all of the inner EAP method authentications. In case inner methods are bypassed, the EAP Server sends the Result TLV only without the Crypto-binding TLV, and the peer responds with Result TLV with Success. The EAP-Server may start a full sequence of EAP authentication or a partial sequence if one or all of the PACs are not present or accepted.

ACS supports the following inner methods and TLV exchange support combinations:

- EAP-MS-CHAP Authentication + Posture Validation TLV exchange
- EAP-GTC Authentication + Posture Validation TLV exchange
- EAP-TLS Authentication + Posture Validation TLV exchange
- Posture Validation TLV exchange without authentication

Global Authentication Setup

You use the Global Authentication Setup page to enable or disable some of the authentication protocols that ACS supports. You can also configure other options for some of the protocols on the Global Authentication Setup page.

This section contains the following topics:

- [Configuring Authentication Options, page 10-19](#)
- [Global Authentication Setup Page, page 10-21](#)



Caution

Network Access Profile settings override the global authentication settings.

Configuring Authentication Options

Use this procedure to select and configure how ACS handles options for authentication. In particular, use this procedure to specify and configure the varieties of EAP that you allow, and to specify whether you allow MS-CHAP Version 1, MS-CHAP Version 2, or both.

For more information on the EAP-TLS Protocol, see [EAP-TLS Authentication, page 10-2](#). For more information on the PEAP protocol, see [PEAP Authentication, page 10-5](#). For more information on the PEAP protocol, see [EAP-FAST Authentication, page 10-8](#). For details about how various databases support various password protocols, see [Authentication Protocol-Database Compatibility, page 1-7](#).

You use the [Global Authentication Setup Page](#) to set up authentication configuration options.



Note

If users access your network by using a AAA client that is defined in the Network Configuration section as a RADIUS (Cisco Aironet) device, you must enable one or more of the LEAP, EAP-TLS, or EAP-FAST protocols on the Global Authentication Setup page; otherwise, Cisco Aironet users cannot authenticate.

Before You Begin

For information about the options see the [Global Authentication Setup Page, page 10-21](#).

To configure authentication options:

-
- Step 1** In the navigation bar, click **System Configuration**.
- Step 2** Click **Global Authentication Setup**.
- The Global Authentications page appears.

- Step 3** Configure options, as applicable. For more information about the significance of the options, see [Global Authentication Setup Page, page 10-21](#).
- Step 4** If you want to immediately implement the settings that you have made, click **Submit + Restart**. ACS restarts its services and implements the authentication configuration options that you selected.
- Step 5** If you want to save the settings that you have made but implement them later, click **Submit**.



Tip You can restart ACS services at any time by using the Service Control page in the System Configuration section.

ACS saves the authentication configuration options that you selected.

Global Authentication Setup Page

This page contains:

Field	Description
PEAP	<p>You can configure the following options for PEAP:</p> <ul style="list-style-type: none"> • Allow EAP-MSCHAPv2—Whether ACS attempts EAP-MS-CHAPv2 authentication with PEAP clients. • Allow EAP-GTC—Whether ACS attempts EAP-GTC authentication with PEAP clients. <p>Note If you check the Allow EAP-MS-CHAPv2 and the Allow EAP-GTC check boxes, ACS negotiates the EAP type with the end-user PEAP client.</p> <ul style="list-style-type: none"> • Allow Posture Validation—To enable use of PEAP for posture validation of Network Admission Control (NAC) clients, check this check box. • Cisco client initial message—The message that you want to appear during PEAP authentication. The message that the PEAP client initially displays is the first challenge that a user of a Cisco Aironet PEAP client sees when attempting authentication. It should direct the user what to do next; for example, <i>Enter your message</i>. The message is limited to 40 characters. • PEAP session timeout (minutes)—The maximum PEAP session length to allow users, in minutes. A session timeout value that is greater than zero (0) enables the PEAP session resume feature, which caches the TLS session that was created in phase one of PEAP authentication. When a PEAP client reconnects, ACS uses the cached TLS session to restore the session, which improves PEAP performance. ACS deletes cached TLS sessions when they time out. The default timeout value is 120 minutes. To disable the session resume feature, set the timeout value to zero (0). • Enable Fast Reconnect—This option is related to MS CHAP only, and does not apply to EAP-GTC. If you want ACS to resume sessions for MS PEAP clients without performing phase two of MS PEAP authentication, check this check box. Unchecking this check box causes ACS to perform phase two of MS PEAP authentication, even when the PEAP session has not timed out. <p>Fast recondition can occur only when ACS allows the session to resume because the session has not timed out. If you disable the PEAP session resume feature by entering zero (0) in the PEAP session timeout (minutes) box, checking the Enable Fast Reconnect check box has no effect on PEAP authentication and phase two of PEAP authentication always occurs.</p>
EAP-FAST	<p>EAP-FAST Configuration—Select to open the EAP-FAST Configuration Page.</p> <p>Note If you are using ACS to implement NAC, enable each option and then click Submit. When the page reappears, select EAP-FAST Configuration to open the EAP-FAST Settings page.</p>

Field	Description
EAP-TLS	<p>Check this box to use the EAP-TLS Authentication protocol and configure EAP-TLS settings. You can specify how ACS verifies user identity as presented in the EAP Identity response from the end-user client. User identity is verified against information in the certificate presented by the end-user client. This comparison occurs after an EAP-TLS tunnel is established between ACS and the end-user client. Select one of:</p> <p>Note EAP-TLS is a certificate-based authentication protocol. EAP-TLS authentication can occur only after you have completed the required steps on the ACS Certificate Setup page. See Installing an ACS Server Certificate, page 10-26 for more information.</p> <p>Select one or more EAP-TLS comparison methods. If you select more than one comparison type, ACS performs the comparisons in the order listed. If the one comparison type fails, ACS attempts the next enabled comparison type. Comparison stops after the first successful comparison.</p> <ul style="list-style-type: none"> • Certificate SAN comparison—If you want ACS to verify user identity by comparing the name in the Subject Alternative Name field of the end-user certificate to the username in the applicable user database, check this check box. • Certificate CN comparison—If you want ACS to verify user identity by comparing the name in the Common Name field of the end-user certificate to the username in the applicable user database, check this check box. • Certificate Binary comparison—If you want ACS to verify user identity by doing a binary comparison of the end-user certificate to the user certificate stored in Active Directory, check this check box. <p>EAP-TLS session timeout (minutes)—Enter a value in minutes for that defines the maximum time for the EAP-TLS session.</p> <p>ACS supports an EAP-TLS session resume feature that caches the TLS session created during a new EAP-TLS authentication. When an EAP-TLS client reconnects, the cached TLS session is used to restore the session without performing a certificate comparison, which improves EAP-TLS performance. ACS deletes cached TLS sessions when they time out. If ACS or the end-user client is restarted, certificate comparison is required even if the session timeout interval has not ended. To disable the session resume feature, set the timeout value to zero (0).</p>
LEAP	<p>The Allow LEAP (For Aironet only) check box controls whether ACS performs LEAP authentication. LEAP is currently used only for Cisco Aironet wireless networking. If you disable this option, Cisco Aironet end-user clients who are configured to perform LEAP authentication cannot access the network. If all Cisco Aironet end-user clients use a different authentication protocol, such as EAP-TLS, we recommend that you disable this option.</p> <p>Note If users who access your network by using a AAA client that is defined in the Network Configuration section as a RADIUS (Cisco Aironet) device, then you must enable LEAP, EAP-TLS, or both on the Global Authentication Setup page; otherwise, Cisco Aironet users cannot authenticate.</p>
EAP-MD5	To enable EAP-based Message Digest 5 hashed authentication, check this check box.

Field	Description
Allow EAP request timeout (seconds)	<p>You use this option to instruct Cisco Aironet Access Points (APs) to use the specified timeout value during EAP conversations. The value that is specified must be the number of seconds after which Cisco Aironet APs should assume that an EAP transaction with ACS has been lost and should be restarted. A value of zero (0) disables this feature.</p> <p>During EAP conversations, ACS sends the value that is defined in the AP EAP request timeout box by using the IETF RADIUS Session-Timeout (27) attribute.</p>
MS-CHAP Configuration	<p>For RADIUS authentication, ACS supports MS-CHAP versions 1 and 2. You can configure whether ACS authenticates users with MS-CHAP when the AAA protocol is RADIUS and, if so, which versions it uses.</p> <p>To enable MS-CHAP in RADIUS-based authentication, check the check box corresponding to the MS-CHAP version that you want to use. To allow MS-CHAP to use either version, check both check boxes.</p> <p>To disable MS-CHAP in RADIUS-based authentication, clear both check boxes.</p> <p>Note For TACACS+, ACS supports only MS-CHAP version 1. TACACS+ support for MS-CHAP version 1 is always enabled and is not configurable.</p>

EAP-FAST Configuration Page

This page contains:

Field	Description
Allow EAP-FAST	Whether ACS permits EAP-FAST authentication.
Active master key TTL	<p>The duration that a master key is used to generate new PACs. Enter a value for the amount of time that a master key is used to generate new Protected Access Credentials (PACs). When the time to live (TTL) that is defined for the Master Key expires, the master key is considered retired and a new master key is generated. The default master key TTL is one month. Decreasing the master key TTL can cause retired master keys to expire because a master key expires when it is older than the sum of the master key TTL and the retired master key TTL; therefore, decreasing the master key TTL requires PAC provisioning for end-user clients with PACs that are based on the newly expired master keys. For more information about master keys, see About Master Keys, page 10-10.</p>
Retired master key TTL	<p>Enter a value for the amount of time that PACs that are generated by using a retired master key are acceptable for EAP-FAST authentication. When an end-user client gains network access by using a PAC that is based on a retired master key, ACS sends a new PAC to the end-user client. The default retired master key TTL is three months.</p> <p>Note Decreasing the retired master key TTL can cause retired master keys to expire; therefore, decreasing the retired master key TTL requires PAC provisioning for end-user clients with PACs based on the newly expired master keys.</p>
Tunnel PAC TTL	<p>The duration that a PAC is used before it expires and must be replaced. Enter a value for the amount of time that a PAC is used before it expires and must be replaced. If the master key that is used to generate the Tunnel PAC has not expired, new PAC creation and assignment is automatic. If the master key used to generate the Tunnel PAC that expired, you must use automatic or manual provisioning to provide the end-user client with a new PAC.</p> <p>For more information about PACs, see About PACs, page 10-11.</p>

Field	Description
Client initial display message	Specify a message to be sent to users who authenticate with an EAP-FAST client. Maximum length is 40 characters. A user will see the initial message only if the end-user client supports its display.
Authority ID Info	The textual identity of this ACS server, which an end user can use to determine which ACS server to be authenticated against. Filling in this field is mandatory.
Allow anonymous in-band PAC provisioning	ACS provisions an end-user client with a PAC by using EAP-FAST phase zero. If you check this check box, ACS establishes a secured connection with the end-user client for the purpose of providing the client with a new PAC. This option allows an anonymous TLS handshake between the end-user client and ACS. EAP-MS-CHAP will be used as the only inner method in phase zero.
Allow authenticated in-band PAC provisioning	ACS provisions an end-user client with a PAC by using EAP-FAST phase zero with SSL server-side authentication. This option requires that a server certificate and a trusted root CA are installed on ACS. One of the allowed inner methods will then be used to authenticate the user. In addition, the client may send its certificate to the server, causing the mutual TLS authentication. In this case, ACS skips the inner methods and provisions the PAC.
Accept client on authenticated provisioning	This option is only available when the allow authenticated in-band PAC provisioning option is selected. The server always sends an Access-Reject at the end of the provisioning phase, forcing the client to reauthenticate using the tunnel PAC. This option enables ACS to send an Access-Accept to the client at the end of the provisioning phase.
Require client certificate for provisioning	Allows provisioning PACs based on certificates only. Other inner EAP methods for PAC provisioning are not allowed. If the client does not present its certificate during the first TLS handshake, the server initiates a TLS renegotiation. The renegotiation requests the client to start a new TLS handshake; the cipher that was negotiated in the first handshake protects it. During the second TLS handshake, the server requests the client's certificate. If the certificate is not sent, the handshake fails and the user is denied access.
Allow Machine Authentication	ACS provisions an end-user client with a machine PAC and performs machine authentication (for end-user clients who do not have the machine credentials). The machine PAC can be provisioned to the client by request (in-band) or by administrator (out-of-band). When ACS receives a valid machine PAC from the end-user client, the machine identity details are extracted from the PAC and verified in the ACS database or external databases. After these details are correctly verified, no further authentication is performed. Note After performing machine authentication and when the Required or Posture Only check boxes are checked, ACS also requests the posture credentials.
Machine PAC TTL	Enter a value for the amount of time that a machine PAC is acceptable for use. When ACS receives an expired machine PAC, it automatically reprovisions the end-user client with a new machine PAC (without waiting for a new machine PAC request from the end-user client).
Allow Stateless session resume	Uncheck this option: <ul style="list-style-type: none"> • If you do not want ACS to provision authorization PACs for EAP-FAST clients. • To always perform phase two of EAP-FAST.
Authorization PAC TTL	This option determines the expiration time of the user authorization PAC. When ACS receives an expired authorization PAC, Allow Stateless session resume fails and, therefore, phase two EAP-FAST authentication is performed.

Field	Description
Allowed inner methods	<p>This option determines which inner EAP methods can run inside the EAP-FAST TLS tunnel. For anonymous in-band provisioning, you must enable EAP-GTC and EAP-MS-CHAP for backward compatibility. If you selected Allow anonymous in-band PAC provisioning, you must select EAP-MS-CHAP (phase zero) and EAP-GTC (phase two). If you selected Allow authenticated in-band PAC provisioning, the inner method in the authentication phase is negotiable. (EAP-GTC is used by default in phase zero.) Select one or more of the following inner methods:</p> <ul style="list-style-type: none"> • EAP-GTC—To enable EAP-GTC in EAP FAST authentication, check this check box. • EAP-MS-CHAPv2—To enable EAP-MS-CHAPv2 in EAP FAST authentication, check this check box. • EAP-TLS—To enable EAP-TLS in EAP FAST authentication, check this check box. <p>Note ACS always runs the first enabled EAP method. For example, if you select EAP-GTC and EAP-MS-CHAPv2, then the first enabled EAP method is EAP-GTC.</p>
Select one or more of the following EAP-TLS comparison methods:	<p>Select one or more EAP-TLS comparison methods. If you select more than one comparison type, ACS performs the comparisons in the order listed. If the one comparison type fails, ACS attempts the next enabled comparison type. Comparison stops after the first successful comparison.</p> <ul style="list-style-type: none"> • Certificate SAN comparison—Verifies user identity by comparing the name in the Subject Alternative Name field of the end-user certificate to the username in the applicable user database, check this check box. • Certificate CN comparison—Verifies user identity by comparing the name in the Common Name field of the end-user certificate to the username in the applicable user database, check this check box. • Certificate Binary comparison—Verifies user identity by doing a binary comparison of the end-user certificate to the user certificate stored in Active Directory, check this check box.
EAP-TLS session timeout (minutes)	<p>EAP-TLS session timeout (minutes)—Enter a value in minutes that defines the maximum time for the EAP-TLS session.</p> <p>ACS supports an EAP-TLS session resume feature that caches the TLS session created during a new EAP-TLS authentication. When an EAP-TLS client reconnects, the cached TLS session is used to restore the session without performing a certificate comparison, which improves EAP-TLS performance. ACS deletes cached TLS sessions when they time out. If ACS or the end-user client is restarted, certificate comparison is required; even if the session timeout interval has not ended. To disable the session resume feature, set the timeout value to zero (0).</p>
EAP-FAST Master Server	<p>Select this check box to determine whether ACS creates its own master keys, and uses its own EAP-FAST settings and Authority ID; or, if it uses the EAP-FAST settings, master keys, and Authority ID received from another (slave or replicated) ACS that has been replicated. If you change this setting, click Submit + Restart.</p>

Field	Description
Actual EAP-FAST server status	<p>This option displays the status of the ACS. If you uncheck the EAP-FAST master server check box, the server status does not change to <code>slave</code> until after ACS receives replicated EAP-FAST settings.</p> <p>Note If you uncheck the EAP-FAST Master Server check box, EAP-FAST server status remains <code>Master</code> until ACS receives replicated EAP-FAST components.</p>

ACS Certificate Setup

This section contains the following topics:

- [Installing an ACS Server Certificate, page 10-26](#)
- [Adding a Certificate Authority Certificate, page 10-28](#)
- [Editing the Certificate Trust List, page 10-28](#)
- [Managing Certificate Revocation Lists, page 10-29](#)
- [Generating a Certificate Signing Request, page 10-32](#)
- [Using Self-Signed Certificates, page 10-33](#)
- [Updating or Replacing an ACS Certificate, page 10-36](#)

Installing an ACS Server Certificate

Perform this procedure to install (that is, enroll) a server certificate for your ACS. You can perform certificate enrollment to support EAP-TLS and PEAP authentication, as well as to support HTTPS protocol for GUI access to ACS.

The three options for obtaining your server certificate are:

- Obtain a certificate from a CA.
- Use an existing certificate from local machine storage.
- Generate a self-signed certificate.

Before You Begin

You must have a server certificate for your ACS before you can install it. With ACS, certificate files must be in Base64-encoded X.509. If you do not already have a server certificate in storage, you can use the procedure in [Generating a Certificate Signing Request, page 10-32](#), or any other means, to obtain a certificate for installation.

If you are installing a server certificate that replaces an existing server certificate, the installation could affect the configuration of the CTL and CRL settings on your ACS. After you have installed a replacement certificate, you should determine whether you need to reconfigure any CTL or CRL settings.

If you want to use a server certificate from local machine storage, we recommend that you read *Extensible Authentication Protocol Transport Layer Security Deployment Guide for Wireless LAN Networks*, available on the ACS CD and at <http://www.cisco.com/warp/public/cc/pd/sqsw/sq/tech/index.shtml>. This white paper provides information about how to add a certificate to machine storage and how to configure a Microsoft certification authority server for use with ACS.

To install an existing certificate for use on ACS:

Step 1 In the navigation bar, click **System Configuration**.

Step 2 Click **ACS Certificate Setup**.

Step 3 Click **Install ACS Certificate**.

ACS displays the Install ACS Certificate page.

Step 4 You must specify whether ACS reads the certificate from a specified file or uses a certificate already on the local machine. To specify that ACS:

- Reads the certificate from a specified file, select the **Read certificate from file** option, and then type the full directory path and filename of the certificate file in the Certificate file box.
- Uses a particular existing certificate from local machine certificate storage, select the **Use certificate from storage** option, and then type the certificate CN (common name or subject name) in the Certificate CN box.



Tip Type the certificate CN only; omit the **cn=** prefix.

Step 5 If you generated the request by using ACS, in the Private key file box, type the full directory path and name of the file that contains the private key.



Note If the certificate was installed in storage with the private key, you do not have the private key file and do not need to type it.



Tip This is the private key that is associated with the server certificate.

Step 6 In the Private key password box, type the private key password.



Tip If you used ACS to generate the certificate signing request, this is the same value that you entered in Private key password on the Generate Certificate Signing Request page. If the private key file is unencrypted, leave this box empty.

Step 7 Click **Submit**.

To show that the certificate setup is complete, ACS displays the Installed Certificate Information table, which contains:

- Issued to: *certificate subject*
 - Issued by: *CA common name*
 - Valid from:
 - Valid to:
 - Validity:
-

Adding a Certificate Authority Certificate

Use this procedure to add new CA certificates to ACS local certificate storage.



Note

If the clients and ACS are getting their certificates from the same CA, you do not need to perform this procedure because ACS automatically trusts the CA that issued its certificate.

When a user certificate is from an unknown CA (that is, one that is different from the CA that certifies the ACS), you must specifically configure ACS to trust that CA or authentication fails. Until you perform this procedure to explicitly extend trust by adding another CA, ACS only recognizes certificates from the CA that issued its own certificate.

Configuring ACS to trust a specific CA is a two-step process that comprises this procedure of adding a CA's certificate and the procedure in [Editing the Certificate Trust List, page 10-28](#), in which you specify that the particular CA is to be trusted. (ACS comes configured with a list of popular CAs, none of which is enabled until you explicitly specify trustworthiness.)

To add a certificate authority certificate to your local storage:

Step 1 In the navigation bar, click **System Configuration**.

Step 2 Click **ACS Certificate Setup**.

Step 3 Click **ACS Certification Authority Setup**.

ACS displays the CA Operations table on the Certification Authorities Setup page.

Step 4 In the CA certificate file box, type the full path and filename for the certificate to use.

Step 5 Click **Submit**.

The new CA certificate is added to local certificate storage. And, if it is not already there, the name of the CA that issued the certificate is placed on the CTL.



Tip

To use this new CA certificate to authenticate users, you must edit the certificate trust list to specify that this CA is trusted. For more information, see [Editing the Certificate Trust List, page 10-28](#).

Editing the Certificate Trust List

ACS uses the CTL to verify the client certificates. For ACS to trust a CA, its certificate must be installed and the ACS administrator must explicitly configure the CA as trusted by editing the CTL. If the ACS server certificate is replaced, the CTL is erased; you must then configure the CTL explicitly each time you install or replace a ACS server certificate.



Note

The single exception to the requirement that you must explicitly specify a CA as trustworthy occurs when the clients and ACS are getting their certificates from the same CA. You do not need to add this CA to the CTL because ACS automatically trusts the CA that issued its certificate.

How you edit your CTL determines the type of trust model that you have. Many use a restricted trust model wherein very few privately controlled CAs are trusted. This model provides the highest level of security; but restricts adaptability and scalability. The alternative, an open trust model, allows for more CAs or public CAs. This open trust model trades increased security for greater adaptability and scalability.

We recommend that you fully understand the implications of your trust model before editing the CTL in ACS.

Use this procedure to configure CAs on your CTL as trusted or not trusted. Before you can configure a CA as trusted on the CTL, you must have added the CA to the local certificate storage; for more information, see [Adding a Certificate Authority Certificate, page 10-28](#). If a user's certificate is from a CA that you have not specifically configured ACS to trust, authentication fails.

To edit the CTL:

Step 1 In the navigation bar, click **System Configuration**.

Step 2 Click **ACS Certificate Setup**.

Step 3 Click **Edit Certificate Trust List**.

The Edit the Certificate Trust List (CTL) table appears.



Warning

Adding a public CA, which you do not control, to your CTL may reduce your system security.

Step 4 To configure a CA on your CTL as trusted, check the corresponding check box.



Tip

You can check, or uncheck, as many CAs as you want. Unchecking a CA check box configures the CA as not trusted.

Step 5 Click **Submit**.

ACS configures the specified CA (or CAs) as trusted or not trusted in accordance with checking or unchecking check boxes. The selected Certificate Trust Lists automatically appear on the CRL Issuers page.

Managing Certificate Revocation Lists

Certificate revocation lists (CRLs) are the means by which ACS determines that the certificates employed by users who seek authentication are still valid, according to the CA that issued them.

This section contains the following topics:

- [About Certificate Revocation Lists, page 10-30](#)
- [Certificate Revocation List Configuration Options, page 10-30](#)
- [Editing a Certificate Revocation List Issuer, page 10-32](#)

About Certificate Revocation Lists

When a digital certificate is issued, you generally expect it to remain valid throughout its predetermined period of validity. However, various circumstances may call for invalidating the certificate earlier than expected. Such circumstances might include compromise or suspected compromise of the corresponding private key, or a change in the CAs issuance program. Under such circumstances, a CRL provides the mechanism by which the CA revokes the legitimacy of a certificate and calls for its managed replacement.

ACS performs certificate revocation by using the X.509 CRL profile. A CRL is a signed and time-stamped with a data structure that a CA (or CRL issuer) issues and which is freely available in a public repository (for example, in an LDAP server). Details on the operation of the X.509 CRL profile are contained in RFC3280.

CRL functionality in ACS includes:

- **Trusted publishers and repositories configuration**—In the ACS web interface, you can view and configure CRL issuers, and their CRL Distribution Points (CDPs) and periods.
- **Retrieval of CRLs from a CDP**—Using a transport protocol (LDAP or HTTP), ACS is configured to periodically retrieve CRLs for each CA that you configure. These CRLs are stored for use during EAP-TLS authentication. Note that there is no timestamp mechanism; instead ACS waits for a specified period of time and then automatically downloads the CRL. If the new CRL differs from the existing CRL, the new version is saved and added to the local cache. CRL retrievals appear in the log for the CSAuth service only when you have configured the level of detail in service logs to `full`. The status, date, and time of the last retrieval appears on the Certificate Revocation List Issuer edit page of the ACS web interface.



Note Automatic CRL retrieval scheduling only functions if EAP-TLS is enabled.

- **Verification of certificate status**—During EAP-TLS authentication, ACS checks the certificate that the user against the corresponding CRL that the CA of the user's certificate issues. If, according to the CRL that ACS currently stores, the certificate has been revoked and authentication fails.

CRL issuers can only be added in association with trusted CAs (that is, CAs on the CTL). If you install a new server certificate for ACS, your CTL is cleared of all trust relationships. While you must reestablish CAs on the CTL, the associated CRLs that you previously configured remain in place and do not have to be reconfigured.

Certificate Revocation List Configuration Options

The Certificate Revocation List Issuers edit page contains the following configuration options:

- **Name**—The name given by the CA Issuer.
- **Description**—A description that you give this CRL issuer.
- **CRL Distribution URL**—The URL that ACS should use to retrieve the CRL. If a CA certificate contains a `CRL distribution points` parameter, this field will be populated automatically. Otherwise, ensure that you specify a URL for the CRL corresponding to the CA that you selected from the Issuer's Certificate list. You can specify a URL that uses HTTP, LDAP, or FTP. Alternatively, you can specify the URL for the file itself; however, this is only necessary when the repository URL lists multiple files.

An example of an HTTP URL is:

<http://crl.verisign.com/pca1.1.1.crl>

An example of an LDAP URL is:

```
ldap://10.36.193.5:388/CN=development-CA,CN=acs-westcoast2,CN=CDP,CN=Public Key
Services,CN=Services,CN=Configuration,DC=cisco,DC=com
```

**Note**

In LDAP, the default placement for the CRL is under objectclass=crIDistributionPoint. ACS adds the object class information to the URL. If the CRL is located elsewhere, you must add the object class to the URL. For example, if the CRL is situated under objectclass=CertificateRevocationList the URL should be: `ldap://10.36.193.5:388/CN=development-CA,CN=acs-westcoast2,CN=CDP,CN=Public Key Services,CN=Services,CN=Configuration,DC=cisco,DC=com?(objectclass=CertificateRevocationList)`.

**Tip**

The URL must specify the CRL itself when the repository contains multiple files.

- **Retrieve CRL**—Initially ACS attempts to download a CRL from the CA. The CRL folder and file are created in the installation directory after a CRL is successfully downloaded. The CRL issuer is not modifiable. The Next Update field in the CRL file contains a value for the Next Update. Select the method that ACS should use for retrieving a CRL:
 - **Automatically**—Uses the value in the Next Update field in the CRL file to retrieve a new CRL from the CA. If unsuccessful, ACS tried to retrieve the CRL every 10 minutes after the first failure until it succeeds.
 - **Every**—Determines the frequency between retrieval attempts. Enter the amount in units of time.

**Note**

For the automatic CRL retrieval function to operate, ensure that you have enabled EAP-TLS.

**Note**

In both modes, if retrieval fails, a reattempt occurs every 10 minutes.

- **Last Retrieve Date**—This entry lists the status, and the date and time of the last CRL retrieval or retrieval attempt.
- **Options**—You check the **Ignore Expiration Date** check box to check a certificate against an outdated CRL.

When the **Ignore Expiration Date** is unchecked, ACS examines the expiration date of the CRL in the Next Update field in the CRL file and continues to use this CRL; even though it has expired. If the expiry date passed, the CRL is not valid and all EAP-TLS authentications will be rejected.

When the **Ignore Expiration Date** is checked, ACS continues to use the expired CRL and permits or rejects EAP-TLS authentications according to the contents of the CRL.
- **CRL is in Use**—When checked, the CRL is active and is used in the EAP-TLS authentication process.
- **Submit**—Click **Submit** to download and verify the CRL with the public key of the issuer. Inconsistencies generate CRL Issuer Configuration errors.

When submission succeeds, you must restart ACS to apply the new configuration.

Editing a Certificate Revocation List Issuer

To edit a certificate revocation list issuer:

Step 1 In the navigation bar, click **System Configuration**.

Step 2 Click **ACS Certificate Setup**.

Step 3 Click **Certificate Revocation Lists**.

The CRL Issuers page appears.

Step 4 Click the name of the CRL issuer that you want to edit.

The system displays the CRL Issuer Edit page for the CRL that you chose.

Step 5 Edit the information and settings that you want to change.

Step 6 Click **Submit**.

The corresponding CRL is changed in ACS to that of the edited issuer (or is scheduled to be changed at the time that you specify in the Retrieve CRL field).



Tip You can refer to the **Last Retrieve date** box to see the status, date, and time of the last CRL retrieval attempt.

Generating a Certificate Signing Request

You can use ACS to generate a certificate signing request (CSR). After you generate a CSR, you can submit it to a CA to obtain your certificate. You perform this procedure to generate the CSR for future use with a certificate enrollment tool.



Note If you already have a server certificate, you do not need to use this portion of the ACS Certificate Setup page.

To generate a certificate signing request:

Step 1 In the navigation bar, click **System Configuration**.

Step 2 Select **ACS Certificate Setup**, then **Generate Certificate Signing Request**.

ACS displays the Generate Certificate Signing Request page.

Step 3 In the Certificate subject box, type values for the certificate fields that the CA to which to submit the CSR. Filling in the CN field is mandatory. The format is:

field=value, field=value, . . .

where *field* is the field name, such as *CN*, and *value* is the applicable value for the field, such as *acs01primary*. You can type a maximum of 256 characters in the Certificate subject box. Separate multiple values with commas (,); for example:

CN=acs01primary, O=WestCoast, C=US, S=California

Table 10-3 defines the valid fields that you can include in the Certificate subject box.

Table 10-3 Certificate Subject Fields

Field	Field Name	Min. Length	Max. Length	Required?
CN	commonName	1	64	Yes
OU	organizationalUnitName	—	—	No
O	organizationName	—	—	No
S	stateOrProvinceName	—	—	No
C	countryName	2	2	No
E	emailAddress	0	40	No
L	localityName	—	—	No

- Step 4** In the **Private key file** box, type the full directory path and name of the file in which the private key is saved; for example, `c:\privateKeyFile.pem`.
- Step 5** In the **Private key password** box, type the private key password (that you have invented).
- Step 6** In the **Retype private key password** box, retype the private key password.
- Step 7** From the **Key length** list, select the length of the key to use.



Tip The choices for Key length are 512 or 1024 bits. The default and more secure choice is 1024 bits.

- Step 8** From the **Digest to sign with** list, select the digest (or hashing algorithm). The choices are MD2, MD5, SHA, and SHA1. The default is SHA1.
- Step 9** Click **Submit**.
ACS displays a CSR on the right side of the browser.
- Step 10** Submit the CSR to the CA of your choice.
After you receive the certificate from the CA, you can perform the steps in [Installing an ACS Server Certificate](#), page 10-26.

Using Self-Signed Certificates

You can use ACS to generate a self-signed digital certificate to use for the PEAP authentication protocol or HTTPS support of ACS administration. This capability supports TLS/SSL protocols and technologies without the requirement of interacting with a CA.

This section contains the following topics:

- [About Self-Signed Certificates](#), page 10-34
- [Self-Signed Certificate Configuration Options](#), page 10-34
- [Generating a Self-Signed Certificate](#), page 10-35

About Self-Signed Certificates

ACS supports TLS/SSL-related protocols, including PEAP, EAP-FAST, and HTTPS, that require the use of digital certificates. Employing self-signed certificates is a way for administrators to meet this requirement without having to interact with a CA to obtain and install the certificate for the ACS. The administrator uses the self-signed certificate feature in ACS to generate the self-signed digital certificate, and use it for the PEAP and EAP-FAST authentication protocols or for HTTPS support in web administration service.

Other than the lack of interaction with a CA to obtain the certificate, installing a self-signed certificate requires exactly the same user actions as any other digital certificate. Although ACS does not support the replication of self-signed certificates, you can export a certificate for use on more than one ACS. To do this, you copy the certificate file (.cer format) and the corresponding private key file (.pvk format) to another ACS where you then install the certificate in the standard manner. For information on installing certificates, see [Installing an ACS Server Certificate, page 10-26](#).

To ensure that a self-signed certificate interoperates with the client, refer to your client documentation. You may find that you must import the self-signed server certificate as a CA certificate on your particular client.

Self-Signed Certificate Configuration Options

The Generate Self-Signed Certificate edit page contains the following mandatory configuration fields:

- **Certificate subject**—The subject for the certificate, prefixed with **cn=**. We recommend using the ACS name. For example, **cn=ACS11**. The Certificate subject field here can contain a number of content entries as comma-separated items; these include:
 - **CN**—common name (the mandatory entry)
 - **OU**—organizational unit name
 - **O**—organization name
 - **S**—state or province
 - **E**—email address
 - **L**—locality name

For example, the Certificate subject field might appear as:

```
cn=ACS 11, O=Acme Enterprises, E=admin@acme.com
```



- **Certificate file**—The full path and filename for the certificate file that you want to generate. For example, **c:\acs_server_cert\acs_server_cert.cer**. When you submit this page, ACS creates the certificate file by using the location and filename that you specify.
- **Private key file**—The full path and filename for the private key file you want to generate. For example, **c:\acs_server_cert\acs_server_cert.pvk**. When you submit this page, ACS creates the private key file using the location and filename you specify.
- **Private key password**—A private key password for the certificate. Minimum length for the private key password is 4 characters, and the maximum length is 64 characters.
- **Retype private key password**—The private key password typed again, to ensure accuracy.
- **Key length**—Select the key length from the list. The choices include 512 bits, 1024 bits, and 2048 bits.
- **Digest to sign with**—Select the hash digest to use to encrypt the key from the list. The choices include SHA1, SHA, MD2, and MD5.

- **Install generated certificate**—Select this check box if you want ACS to install the self-signed certificate that it generates when you click **Submit**. If you employ this option, you must restart ACS services after you submit the page for the new settings to take effect. If you do not select this option, the certificate file and private key file are generated and saved; but are not installed into local machine storage.

Generating a Self-Signed Certificate

All fields on the Generate Self-Signed Certificate page are mandatory. For information on the fields' contents, see [Self-Signed Certificate Configuration Options, page 10-34](#).

To generate a self-signed certificate:

-
- Step 1** In the navigation bar, click **System Configuration**.
- Step 2** Click **ACS Certificate Setup**.
- Step 3** Click **Generate Self-Signed Certificate**.
- The Generate Self-Signed Certificate edit page appears.
- Step 4** In the **Certificate subject** box, type the certificate subject in the form **cn=XXXX**. You can enter additional information here, for more information see [Self-Signed Certificate Configuration Options, page 10-34](#).
- Step 5** In the **Certificate file** box, type the full path and file name for the certificate file.
- Step 6** In the **Private key file** box, type the full path and file name for the private key file.
- Step 7** In the **Private key password** box, type the private key password.
- Step 8** In the **Retype private key password** box, retype the private key password.
- Step 9** In the **Key length** box, select the key length.
- Step 10** In the **Digest to sign with** box, select the hash digest to be used to encrypt the key.
- Step 11** To install the self-signed certificate when you submit the page, select the **Install generated certificate** option.
-
-  **Note** If you select the Install generated certificate option, you must restart ACS services after submitting this form for the new settings to take effect.
-
-  **Tip** If you do not select the Install generated certificate option, the certificate file and private key file are generated and saved when you click Submit in the next step; but are not installed in local machine storage.
-
- Step 12** Click **Submit**.
- The specified certificate and private key files are generated and stored. If you selected the Install generated certificate option, the certificate becomes operational, only after you restart ACS services.
-

Updating or Replacing an ACS Certificate

Use this procedure to update or replace an existing ACS certificate that is out of date or out of order.

**Caution**

This procedure eliminates your existing ACS certificate and erases your CTL configuration.

To install a new ACS certificate:

Step 1 In the navigation bar, click **System Configuration**.

Step 2 Click **ACS Certificate Setup**.

ACS displays the Installed Certificate Information table on the ACS Certificate Setup page.

**Note**

If your ACS has not already been enrolled with a certificate, you do not see the Installed Certificate Information table. Rather, you see the Install new certificate table. If this is the case, proceed to Step 5.

Step 3 Click **Enroll New Certificate**.

A confirmation dialog box appears.

Step 4 To confirm that you intend to enroll a new certificate, click **OK**.

The existing ACS certificate is removed and your CTL configuration is erased.

Step 5 You can now install the replacement certificate in the same manner as an original certificate. For detailed steps, see [Installing an ACS Server Certificate, page 10-26](#).
