RSA Keon Validation Server 2.0
Administrator's Guide

Contact Information

See our Web sites for regional Customer Support telephone and fax numbers.

RSA Security Inc.	RSA Security Ireland Limited
`www.rsasecurity.com`	`www.rsasecurity.ie`

Trademarks

License agreement

Third party licenses

This product may include software developed by parties other than RSA Security. The text of the license agreements applicable to third party software in this product may be viewable in the **thirdpartylicense.pdf** file.

This product includes software developed by Apache Software Foundation (http://www.apache.org/).

Note on encryption technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import or export of encryption technologies, and current use, import and export regulations should be followed when exporting this product.

Distribution

Limit distribution of this document to trusted personnel.

Portions of this product use technologies patented under U.S. patent numbers 5,922,074 and 6,249,873.

Contents

Preface

About RSA Security

With thousands of customers around the globe, RSA Security (NASDAQ: RSAS) provides interoperable solutions for establishing online identities, access rights, and privileges for people, applications, and devices. Built to work seamlessly and transparently in the complex environments of thousands of customers, the Company's comprehensive portfolio of identity and access management solutions—including authentication, Web access management, and developer solutions—is designed to allow customers to confidently exploit new technologies for competitive advantage. RSA Security's strong reputation is built on its history of ingenuity and leadership, proven technologies, and long-standing relationships with more than 1,000 technology partners.

About RSA Keon Validation Server

RSA Keon Validation Server (Keon VS) provides certificate status information to public key infrastructure (PKI) applications. Using Online Certificate Status Protocol (OCSP), Keon VS acquires status information for certificates issued by one or more certificate authorities (CAs). A Keon VS signer signs status responses, which ensures that OCSP clients can validate and trust the status responses they receive. A trusted CA certifies the Keon VS signer.

About the RSA Keon Validation Server Administrator's Guide

This guide describes the administration of RSA Keon Validation Server. For the latest information about the RSA Keon Validation Server product, see the *RSA Keon Validation Server README* on the RSA Keon Validation Server CD-ROM or the SecurCare Online Web site. In the event of a discrepancy, the readme files take precedence over this document and the online documentation. In the event of a discrepancy between this document and the online documentation, the online documentation should take precedence.

Conventions

These alerts are used in the *RSA Keon Validation Server Administrator's Guide*:

Caution	This alert warns you of instances where an instruction or procedure not followed exactly could result significant or irrevocable damage to your installation (hardware or software).
Important	This alert highlights information that you need to know to keep the software operating correctly.
Note	This alert points to tips that may make the software run more smoothly or provides additional information about a concept or procedure.

These typographic conventions are used in the *RSA Keon Validation Server Administrator's Guide*:

Bold	Interface items such as menus, menu commands, and buttons.
Bold Italics	Hyperlinks in the user interface.
`Fixed-width font`	Code fragments and command line arguments, parameters, options, URLs, and directories.

These writing conventions are used in the *RSA Keon Validation Server Administrator's Guide*:

<installed-dir>	The directory where the product is installed.
>	Indicates selecting an item from a menu in an application. For example, the instruction "Click **File > New**" indicates that a user should click the **File** menu and select **New**.

RSA Keon Product Suite

RSA Keon Certificate Authority

RSA Keon Certificate Authority (Keon CA) is the central component of the RSA Keon product suite. One copy of Keon CA allows you to manage multiple certificate authorities and issue certificates under each of them. The rest of the RSA Keon product suite works with Keon CA to provide a full PKI solution.

RSA Keon Registration Authority

RSA Keon Registration Authority (Keon RA) works with Keon CA to easily enroll large numbers of customers or users for certificate usage. You can use Keon RA to create and manage RAs (registration authorities) to perform registration services on

behalf of a CA. The most common service is vetting end-users who enroll for a certificate. The RAs created by Keon RA also serve to provide greater control in limiting access to Keon CA.

RSA Keon OneStep

RSA Keon OneStep (Keon OneStep) is a feature of Keon CA that enables automatic issuance of certificates. Keon OneStep includes an API that enables your enterprise to build customized plug-ins for autovetting certificate requests through the optionally installed Keon OneStep CGI program. The Keon OneStep CGI program handles the process of authenticating, approving, issuing, and installing certificates in one automatic operation, without human intervention. A subset of the Keon OneStep API enables your enterprise to build customized plug-ins for the CMP server.

RSA Keon WebSentry

RSA Keon WebSentry enables Web servers to offer high-assurance user authentication through digital certificates by seamlessly PKI-enabling Web servers and providing real-time validity checking of a user's certificate. Web administrators can also use certificates to limit access to private or sensitive files stored on their Web sites.

RSA Keon Key Recovery Module

RSA Keon Key Recovery Module (Keon KRM) is an add-on module to Keon CA. It can be used to store and recover users' private keys used for data encryption. Keon KRM works with a standard Keon CA installation. Its administrative functions are integrated into the Keon CA interface. Keon KRM generates RSA public key encryption keypairs in a secure hardware module on a central server.

RSA Keon Certificate Authority API

RSA Keon Certificate Authority API (Keon CA API) allows Windows NT, Windows 2000, and Unix Solaris developers to manage Keon CA, including the ability to issue certificates, create new CAs, revoke certificates, and check certificate status.

RSA Keon Web PassPort

RSA Keon Web PassPort is a standards-based product that makes PKI easy to use. It is designed for environments where a desktop footprint is not appropriate and interoperability with applications such as browsers, mail clients, and web authorization systems is required. The small, downloaded plug-in seamlessly integrates with browsers, mail clients, and other applications to enable digital signing, user-authenticated SSL, secure e-mail (S/MIME), and VPN via PKCS #11 and Microsoft Certificate API interfaces.

RSA Keon Validation Server

RSA Keon Validation Server (Keon VS) provides certificate status information to public key infrastructure (PKI) applications. Using OCSP, Keon VS acquires status information for certificates issued by one or more certificate authorities (CAs). A Keon VS signer signs status responses, which ensures that OCSP clients can validate and trust the status responses they receive. A trusted CA certifies the Keon VS signer.

RSA Keon Validation Client

RSA Keon Validation Client (Keon VC) is an OCSP client that can query OCSP servers and obtain realtime certificate status. In addition, Keon VC can import and store revocation lists (that is, complete CRLs, ARLs, and delta CRLs) from Lightweight Directory Access Protocol (LDAP) servers and determine the status of a certificate using the appropriate revocation list. Keon VC can obtain status of certificates signed by different certificate authorities (CAs) using the method configured for the given CA.

Using Other RSA Products with Keon VS

RSA BSAFE

RSA BSAFE software is embedded in today's most successful Internet applications, including Web browsers, wireless devices, commerce servers, e-mail systems, and virtual private network products. Built to provide implementations of standards such as SSL, S/MIME, WTLS, IPSec and PKCS, RSA BSAFE products can save developers time and risk in their development schedules, and provide the security that only comes from a decade of proven, robust performance.

RSA ClearTrust

RSA ClearTrust Web access management software enables secure access to Web-based resources. It is designed to work within intranets, extranets, portals, and exchange infrastructures—all while providing users with transparent, single sign-on both within and across multiple sites and domains. The RSA ClearTrust solution centrally controls and manages user access privileges to Web-based resources based on definable user attributes, business rules, and security policies that directly reflect the objectives of an e-business strategy.

RSA e-Sign

RSA e-Sign software is a digital signature solution for trusted end-to-end e-business processes. Using RSA e-Sign software, organizations can complete their business processes online with the assurance of transactional security and non-repudiation.

RSA Mobile

RSA Mobile technology protects access to Web-based resources through the use of existing mobile phones. With RSA Mobile software, companies can positively identify users and cost-effectively protect Web resources with two-factor authentication. RSA Mobile technology takes advantage of a device that the end-user already has-a mobile phone. This reduces costs by eliminating the need to deploy any hardware or software at all to the end user. End users can readily accept the use of RSA Mobile technology, as it does not require them to carry or learn to use any additional device.

RSA SecurID

RSA SecurID systems are a leading solution for two-factor user authentication. RSA SecurID software is designed to protect valuable network resources by helping to ensure that only authorized users are granted access to e-mail, Web servers, intranets, extranets, network operating systems, and other resources. The RSA SecurID family offers a wide range of easy-to-use authenticators—from time-synchronous tokens to smart cards—that help to create a strong barrier against unauthorized access, helping to safeguard network resources from potentially devastating accidental or malicious intrusion.

Worldwide Service and Support

RSA Security offers a full complement of world-class service and support offerings to ensure the success of each customer's project or deployment through a range of ongoing customer support and professional services. These include assessments, project consulting, implementation, education and training, and developer support. RSA Security's technical support organization is known for resolving requests in the shortest possible time, gaining customers' confidence, and exceeding expectations.

Getting Support and Service

SecurCare Online	`https://knowledge.rsasecurity.com`
Customer Support Information	`www.rsasecurity.com/support`

Customer Support SecurCare Online

Visit the SecurCare Online area of the RSA Security Customer Support Web site to find information about new releases, recent patches to supported versions of the software, important technical news, and enhancements to service offerings. The SecurCare Online knowledge database contains answers to common questions and solutions to known problems.

Professional Services

If you or your organization require assistance at any point in your project's life cycle, the RSA Security Professional Services consulting team can help. This group draws on a breadth of experience to provide PKI customers with a comprehensive set of services. Working within the context of customers' unique requirements, services are tailored to suit project-scale and application-specific needs.

RSA Professional Services focus areas include:

- **Planning**—to ensure that application security strategy is aligned with business and IT objectives
- **Designing, building, and integrating**—to preserve technology investments
- **Implementation and training**—to reduce business risk and augment in-house expertise

For further information, please contact your account manager.

chapter 1. Introduction

RSA Keon Validation Server (Keon VS) is part of the world of network security and digital certificates. In preparation for using Keon VS, you should be familar with the following concepts:

- Network security—why is it needed
- Digital certificates—what are they and who issues them

This chapter provides an introduction to Keon VS and its components, introduces the Online Certificate Status Protocol (OCSP), and explains product-specific terminology used in this guide.

Network Security and Digital Certificates

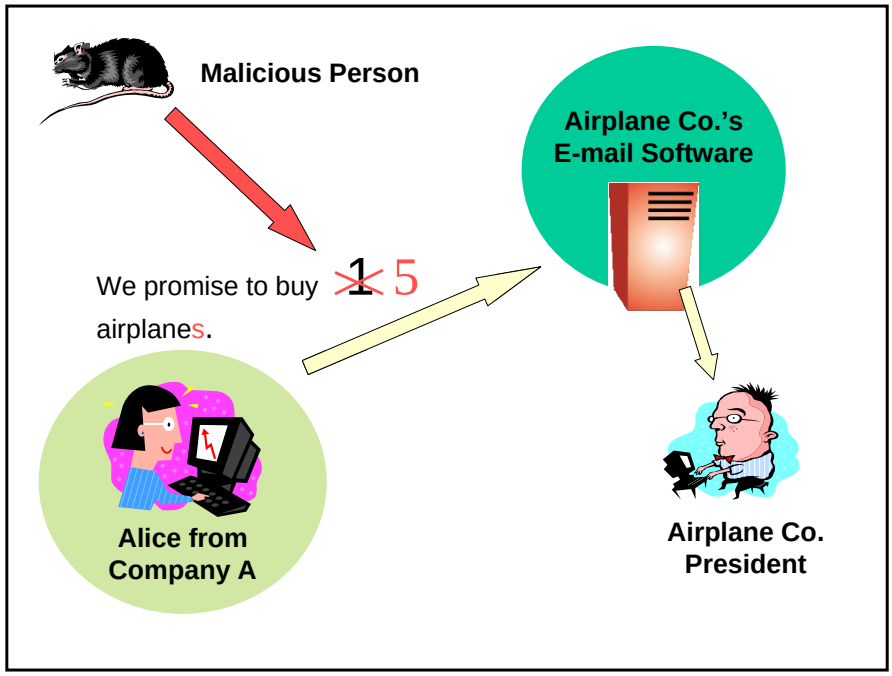
If you are an Internet user, you know that you can connect to many sites on the World Wide Web. You can access documents, do your banking, and buy and sell over the Web. At work, network users can access documents, send and receive documents, and access all kinds of software applications.

Some activities over the Web and in the workplace involve sensitive information, which is where network security comes in: how can two people that are communicating over a network be certain of the identity of the other? For example, to gain access to your building, you might use an identity card. To gain access to documents over a network, you might present a digital identifier, called a digital certificate. The software which receives the certificate, such as the computer program guarding documents to which you are seeking access, will process the digital certificate. A digital certificate is used to prove the identity of an individual electronically.

Another problem is trusting the contents of an e-mail message. How do people know that the message has not been altered by a malicious hacker? Digital certificates can also be used to check for message tampering.

Figure 1 illustrates a security problem in sending and receiving e-mails.

Figure 1: The need for network security



What Are Digital Certificates?

A digital certificate provides a way for software applications to verify your identity in electronic transactions. This is similar to the way that company badges or national passports prove identity in face-to-face interactions.

Not all software applications know how to process digital certificates; therefore, they can only be used by softwar applications that know how to process them.

How is identity verified?

Every certificate is associated with two cryptographic keys known as a key pair. Cryptographic keys are large numbers used to encrypt and decrypt data with complex cryptographic algorithms. The keys are generated when you request a certificate. Each key pair consists of a public key and a corresponding private key. End users must keep their private keys secret. Private keys are stored in a user's software applications (for example, Web browsers, e-mail) or on a special card (for example, a smart card). However, their public keys appear in their certificates. One key encrypts the data, while the other decrypts the data.

Using encryption and decryption techniques, the software can verify that a message has been sent or signed by the person holding the private key matching the public key in a certificate.

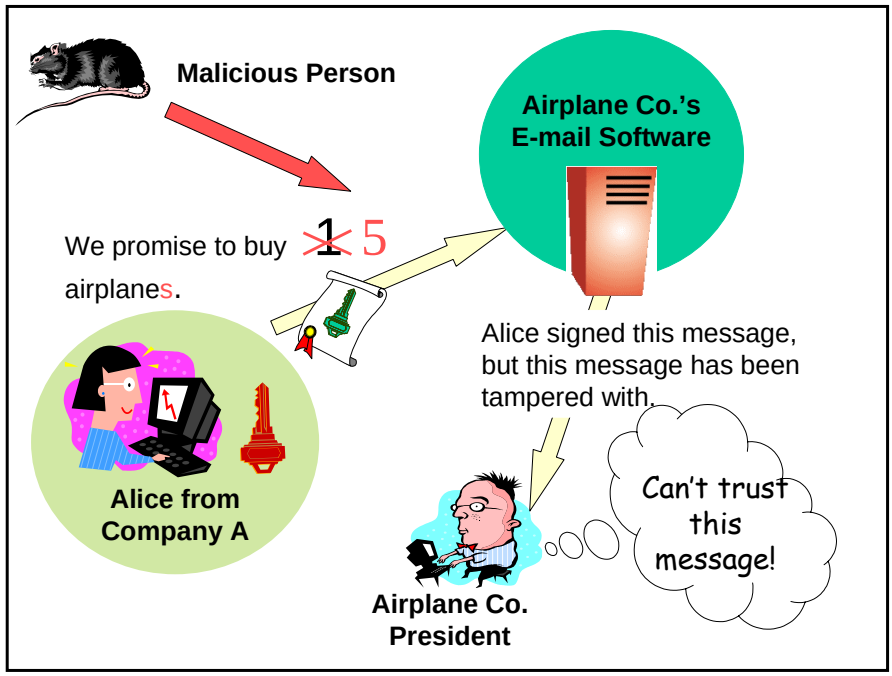
When do you use a digital certificate?

For example, if you try to access a document over the internet by presenting your certificate, your software will encrypt some data with your private key and send it along with the certificate. The software guarding the document will decrypt the data using the public key in the certificate. If decryption is successful, the program knows that the request came from you because you are the only one with the private key.

Another example is signature verification using certificates. If you ask your e-mail application to sign an e-mail, the application automatically accesses the private key and signs the e-mail, and sends it to the addressee. At the other end, the e-mail software that receives the signed e-mail uses your certificate to verify that you did indeed sign the e-mail with your private key. It also uses this process to verify that the message has not been tampered with.

Figure 2 illustrates the process of verifying a message.

Figure 2: Verifying message sender and integrity



What is in a Digital Certificate?

In addition to the public key described in the previous section, certificates contain information about the holder of the certificate; for example, it can contain the name, address, organization, department, or telephone number of the certificate holder. It also contains an expiry date and information about who issued the certificate.

Who Issues Digital Certificates?

Certificates are issued by something called a certificate authority (CA). A CA is part of a software program that can create, sign, and manage digital certificates. An organization uses the software to operate one or more CAs. For example, a company might have one CA that issues certificates to its employees in North America and one that issues certificates to employees in Europe.

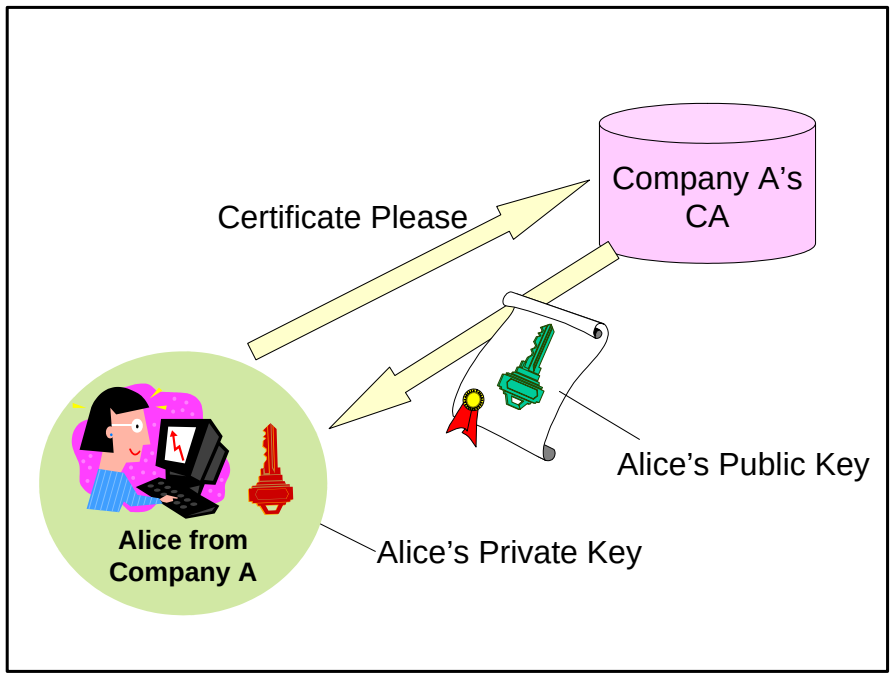
When you request a certificate from an organization, the organization must make sure you are who you say you are before using the CA to issue you a certificate. The CA signs the certificate containing the public key, stores the certificate, and returns a copy to you.

Because the CA signs the certificate it issues, you can see that the CA also uses keys. The CA uses its private key to sign certificates. The CA has a key pair and certificate that proves its identify. The CA can issue a certificate to itself or obtain it from another CA.

Sometimes organizations want to check the signature on certificates, to make sure the certificates were actually signed by a CA that it trusts. Software that receives a certificate signed by the CA can process the signature to verify the identity of the CA. For this, the software requires the CA certificate.

Figure 3 illustrates an end user obtaining a certificate from a CA.

Figure 3: Certificates and key pairs



Who Can Have a Certificate?

So far we have talked about individuals identifying themselves over the network using certificates. However, servers or hardware devices must identify themselves to other servers and devices. Therefore certificates can be issued by CAs to people, servers, and hardware devices (examples of devices are routers or Virtual Private Networks or VPNs).

CAs also require certificates issued to themselves or from other CAs.

When is a Certificate Suspended or Revoked?

Occasionally the privileges of a certificate must be withdrawn. Some examples of this situation are if a person leaves a company, loses the card containing the private key, or doesn't pay for services acquired using the certificate.

When a certificate is no longer trustworthy, the CA can temporarily suspend or permanently revoke the certificate. The certificate is placed on a revocation list published by the CA. These revocation lists are placed in a secondary database accessible to those who want to check status; however, revocation lists can be very large and could take a long time to download and process.

Certificate status can be acquired by obtaining the revocation list or by requesting status from software designed to provide it, such as Keon VS.

What is OCSP?

OCSP stands for Online Certificate Status Protocol. A protocol specifies how messages are exchanged. The OCSP standard provides rules on how software programs using OCSP should exchange messages in order to request and return certificate status. An OCSP client issues a status request to an OCSP responder and suspends acceptance of the certificate in question until the responder provides a response.

An OCSP client issues a status request to an OCSP responder and suspends acceptance of the certificate in question until the responder provides a response. For more information on OCSP, see IETF RFC 2560, which is available at:

```
http://www.ietf.org/rfc/rfc2560.txt?number=2560
```

The certificate status values returned in the response may be one of the following:

If the certificate status is	then the certificate status value is
revoked	revoked
suspended	revoked
active	good
unknown	unknown

The following responses are also possible if the responder experiences an error:

If the OCSP responder error is	the description of the problem is
malformed request	The request is not a correctly formed OCSP request.
internal error	An internal error is affecting the OCSP responder.
try later	The OCSP responder exists but is unable to return an OCSP response at this time.
signature required	The OCSP request must be signed by the client before it can be accepted by the OCSP responder.
unauthorized	The client is not authorized to submit an OCSP request to this server.

Overview of Keon VS

This section provides an overview of Keon VS including what it is, why you would use it, and how it works.

What is Keon VS?

Keon VS is a software application that processes status requests for the status of certificates issued by CAs, acquires the status, and returns the status to the requestor, known as the client.

When Keon VS receives a request for the status of a certificate issued by a CA, Keon VS will search its database to see if it is set up to respond to requests for that CA. If the status of the certificate can be found, Keon VS will return the status.

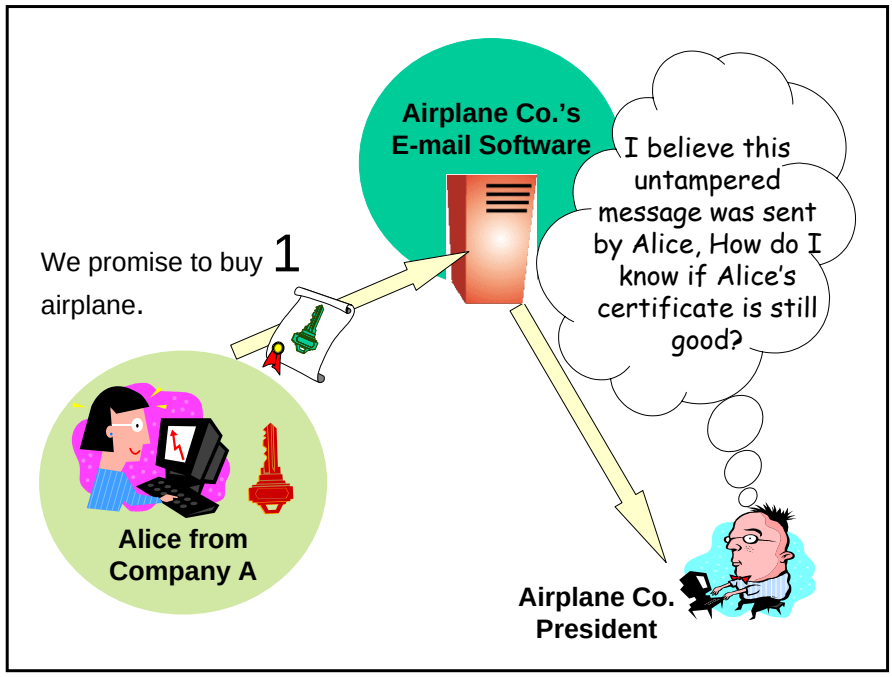
Keon VS communicates with clients using the OCSP protocol. Keon VS is known as an OCSP Responder and accepts status requests from a variety of OCSP clients.

Why Use Keon VS?

What if the private key is stolen or a person leaves the organization? Then the certificate is not to be trusted. How can the software receiving the certificate check to see if the certificate is trustworthy?

Figure 4 illustrates a person who doesn't know whether or not to trust a signed message that he has received, because he doesn't know whether or not to trust Alice.

Figure 4: Why verify the status of the certificate?



This is where Keon VS comes in. Keon VS checks with the CA to see if the CA has temporarily suspended or permanently revoked the user's certificate. Keon VS provides the status of certificates to software applications (and therefore the person using that software) that request the status of a certificate. Keon VS must, of course, be set up to recognize the CA and must know where to obtain this status information.

Figure 5 and Figure 6 illustrate the role of Keon VS in obtaining certificate status.

Figure 5: Obtaining certificate status from Keon VS

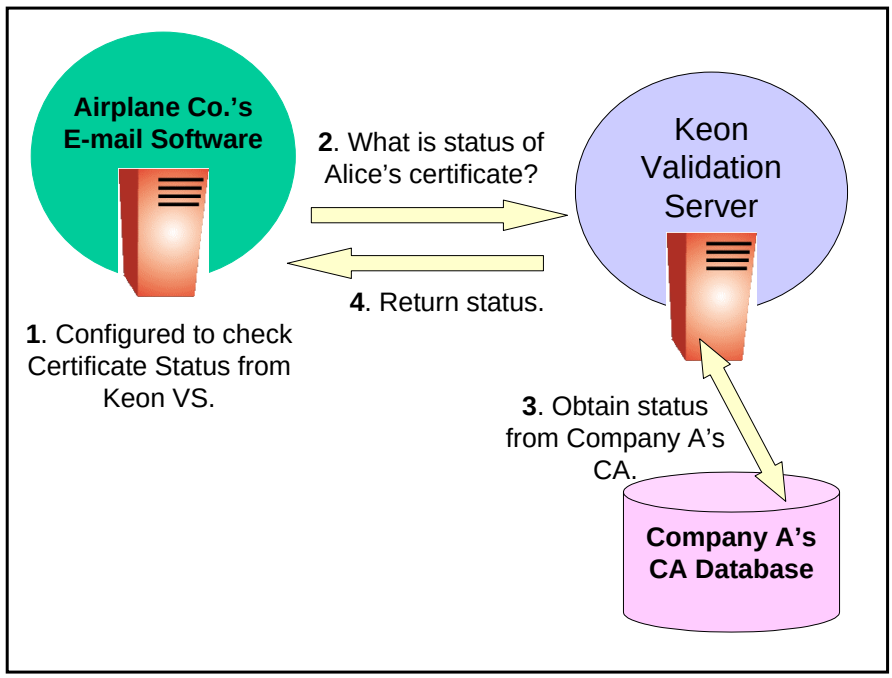
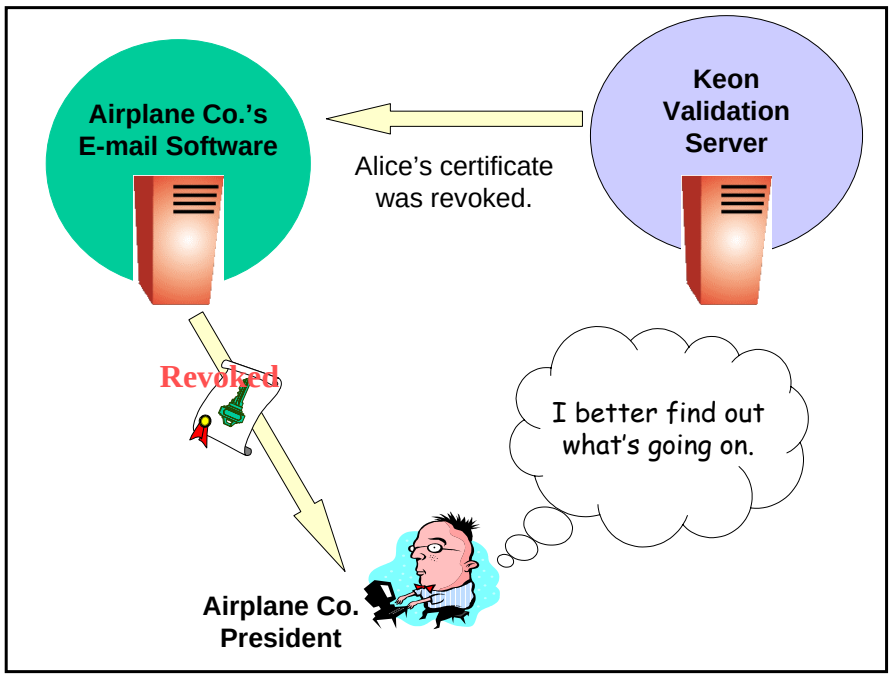


Figure 6: Obtaining certificate status from Keon VS

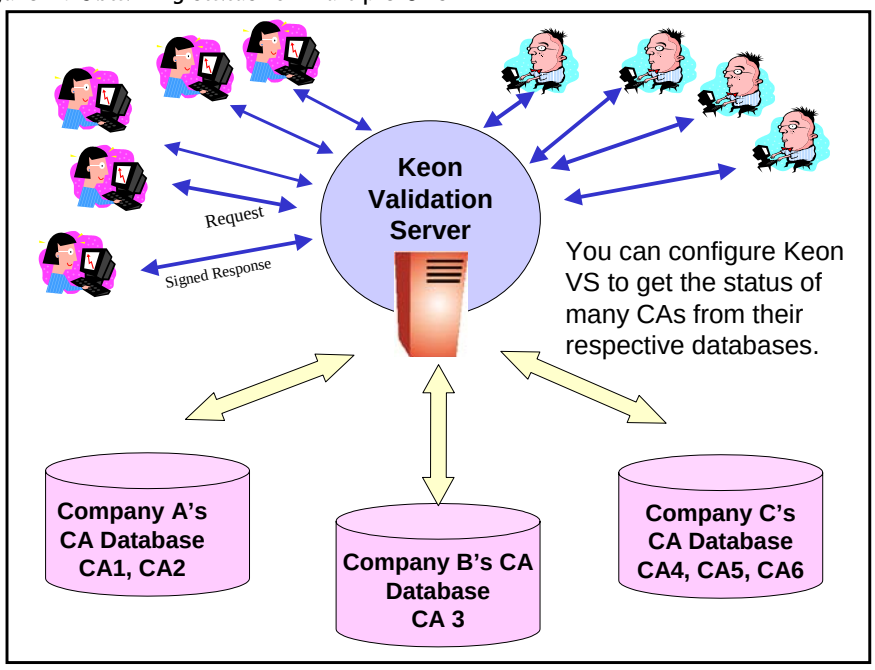


Keon VS does the work of looking up the status for other software applications. The CA does not have to interact with a large number of applications, just Keon VS.

Keon VS can provide status information for more than one CA, and for CAs from different locations. The software applications can be configured to obtain status of certificates issued from various CAs from one location, Keon VS, making the Administrator's jobs easier.

Figure 7 illustrates the efficiency of having Keon VS serve as a central location for obtaining certificate status for many CAs.

Figure 7: Obtaining status for multiple CAs



How Does Keon VS Work?

When Keon VS receives a certificate status query, it obtains status information from the following:

- Revocation lists that CAs publish
- Another OCSP responder

Figure 8 illustrates how Keon VS obtains status information from revocation lists stored in an LDAP directory.

Figure 9 illustrates how Keon VS obtains status information from another OCSP responder.

Figure 8: Keon VS using revocation lists to find certificate status

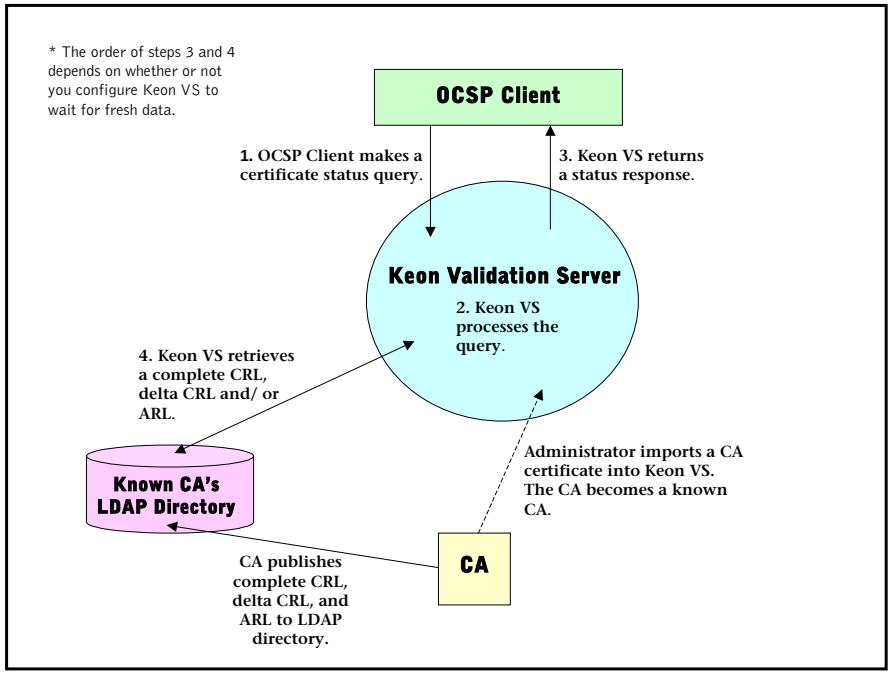
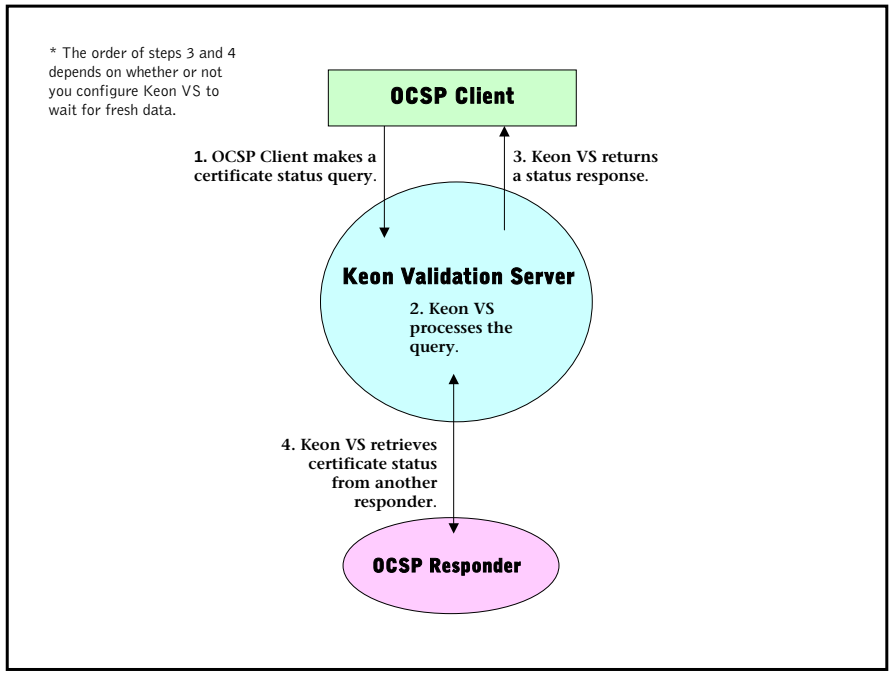


Figure 9: Keon VS using an OCSP responder to find certificate status



Keon VS checks the status of the certificate against its internal database and may return a signed OCSP response immediately to the OCSP client based on this data or it may wait for fresh status data before it returns a response. This response tells your OCSP client that the status of the certificate in question is either `good`, `revoked`, or `unknown`.

Why Are OCSP Responses Signed?

In Keon VS, OCSP signers sign OCSP responses. The signed response ensures the OCSP client that the expected OCSP responder sent the response.

chapter 2. Getting Started with Keon VS

This chapter describes how to start and stop RSA Keon Validation Server (Keon VS), and how to access and set up Keon VS after you install it.

This chapter contains information on the following topics:

- "Starting Keon VS"
- "Accessing the Keon VS User Interface"
- "Using Keon VS the First Time"

Starting Keon VS

Keon VS has two servers: the User Interface (UI) Server and OCSP Server. Starting Keon VS starts both these servers. Keon VS logs each startup event to the system log (`syslog`) and audit log file.

To start Keon VS:

1. Change the current directory to `<installed-dir>/Util`
2. Type `startupVS` at the command line.

   If Keon VS is already running, it is restarted.
3. If prompted, enter the passphrases protecting the system, smart card, or signer private keys.

Passphrases

Passphrases protect the private keys that Keon VS uses. When you start Keon VS, Keon VS may prompt you for a number of passphrases:

- System passphrase—the passphrase entered during installation.

  Note: In the Solaris operating environment, the passphrase characters you type do not display. You submit the entry by pressing the Enter/return key.

- nCipher smart card passphrase—if your system uses a smart card, Keon VS will prompt you for the PIN of the inserted smart card.

  For more information on nCipher hardware security modules (HSMs), see "appendix B. Cryptographic Hardware Interoperability".

- Signer key passphrase—if the signer was created with a passphrase, Keon VS will prompt you for this passphrase. The first time you access Keon VS, there will be no existing signers.

The system will not prompt you for the same unique passphrase more than once with the exception of the Keon VS UI Server. For example, if the nCipher smart card, system passphrase, and signer passphrase are all `1234abcd`, you will only need to provide a passphrase once.

Stopping Keon VS

To stop Keon VS:

1. Change the current directory to `<installed-dir>/Util`
2. At the command line, type `shutdownVS` and press ENTER.

Attended and Unattended Startup

An attended startup is starting or restarting Keon VS by entering passphrases that protect the private keys of the system, smart cards (if used), and signers. Attended startup is the most secure configuration of the system. In this mode, the passphrases are not stored anywhere on the system. If the keys are not protected with a passphrase, no prompts appear.

Note: If Keon VS was installed with an nCipher HSM and there is a smart card in the slot, Keon VS will prompt you for the smart card PIN during startup. If you wish to generate a new nCipher-based OCSP responder key pair or sign a response using an existing nCipher-based key pair during the operation of the OCSP Server, you must enter the PIN in order to log into the smart card. If you do not want to use the nCipher HSM, enter `SKIP` instead of a PIN and startup will proceed without logging into the smart card.

An unattended startup is starting or restarting Keon VS without having to enter passphrases. If you configure Keon VS for unattended startup and the server shuts down (for example, a power failure), you can configure Keon VS to automatically restart when power is restored and the server restarted.

To support an unattended startup of the Keon VS, you must create a `startup.conf` file as follows:

1. Open a text editor.
2. Enter at least one line with the following format:

   ```
   passphrase <passphrase>
   ```

   or

   ```
   pin <pin>
   ```

   The `startup.conf` file may include multiple lines for multiple passphrases. For example, if the nCipher smart card passphrase is `1234abcd`, the system passphrase is `5678efgh`, and the signer passphrase is `2468bdfh`, the `startup.conf` file will contain the following lines in any order:

   ```
   pin 1234abcd # nCipher smart card passphrase
   passphrase 5678efgh # system passphrase
   passphrase 2468bdfh # signer passphrase
   ```

Note: If you need to quote passphrase values, follow these rules:

- Begin with the literal passphrase with no quoting or escaping.
- Replace double quote characters (") with the three literal characters %22. This is URL-escaping. You can represent any character with %xx where xx is the hexadecimal representation of its ASCII value (this includes new lines and other non-printable characters).
- Surround the entire passphrase value with double quotes (").

For example, if your passphrase is 1"2'3\4, use:

```
passphrase  "1%222'3\4"
```

3. Save the file to the top level directory of Keon VS.

If the machine where you installed Keon VS has an attached nCipher HSM that you do not intend to use for Keon VS, you can modify the `startup.conf` to include the ignore directive. For example, type:

```
ignore nCipher
```

Note: While creating a startup.conf file is a convenient means to safeguard against the accidental shutdown of Keon VS, care should be taken to safeguard the `.conf` file. Placing a passphrase in a text file or `.conf` file on a hard disk is not a secure practice as passphrases are stored in clear text.

Keon VS reads this file when starting. Keon VS will not use the `startup.conf` file unless its permissions allow read access to the user or group that starts Keon VS.

If the directive is not present, the behavior will be as if the file does not exist (the attended mode). If the passphrase is incorrect, Keon VS will prompt you for the passphrase as in the attended mode.

Accessing the Keon VS User Interface

You use the Keon VS Graphical User Interface (GUI) to manage Keon VS. To access the UI, open your browser and type the URL:

```
https://<host.subdomain.com>:<ui-port>
```

where

- `https://<host.subdomain.com>` is the FQDN of the server
- `<ui-port>` is the UI Server port number

For more information on these values (entered during installation), see the *RSA Keon Validation Server Installation Guide*.

If you installed Keon VS, you must log in using the userID and password that you entered during installation. With a successful login, the RSA Keon Validation Server home page appears. If you access Keon VS later, you may be presented with the home page without logging in.

If you are accessing a new installation of Keon VS for the first time, see the following section “Using Keon VS the First Time”.

For any additional users of Keon VS, you must obtain a userID and password (and possibly a certificate) from your system administrator before accessing the Keon VS GUI (see “chapter 3. Managing Users”).

Using Keon VS the First Time

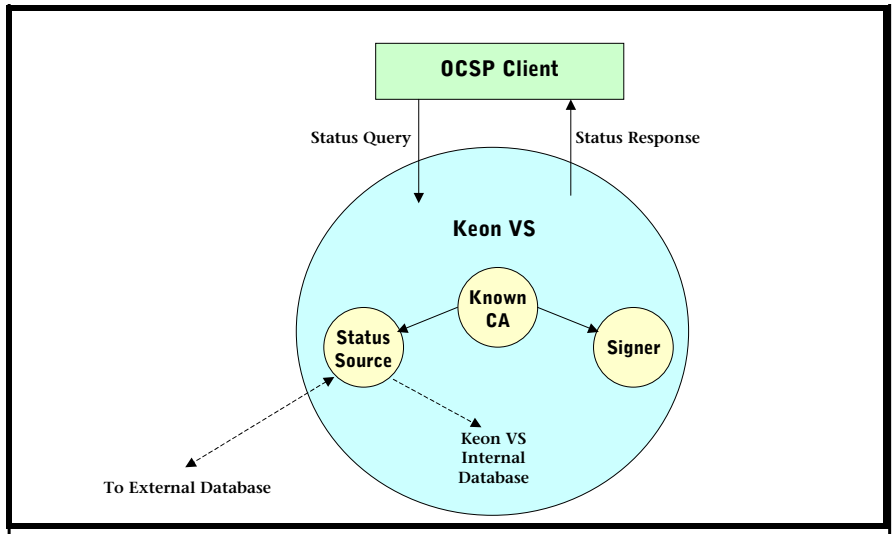
After installing and starting up Keon VS, you must complete the following tasks before Keon VS can respond to certificate status requests:

- Create an OCSP signer to sign OCSP responses for a CA
- Create a status source for a CA
- Import a CA certificate

You may want to configure how Keon VS handles OCSP requests and responses.

Figure 1 illustrates the internal Keon VS entities used to process OCSP client certificate status queries.

Figure 1: Keon VS and its entities



When Keon VS receives a request for the status of a certificate, Keon VS checks to see if it knows the CA that signed the certificate. You must import a CA certificate into Keon VS so that Keon VS can recognize the CA.

Creating an OCSP Signer

To create a signer:

1. From the top-level menu, select **Signers > Add New**.

   The Add Signer page appears.

2. In the **OCSP Signer Nickname** text box, enter a signer nickname.

   You can enter any combination of alphanumeric characters to uniquely identify the signer. “Signer 1” is the default value.

3. In the **Cryptographic Provider** drop-down list, select a cryptographic provider to generate the signer’s key pair.

   The possible selections depend on whether or not a Hardware Security Module (HSM) is connected to the machine on which you installed Keon VS. **Software** is the default and, if there is no HSM, it is the only selection.

   The option you select should depend on the network security policies of your organization.

4. In the **Key Size** drop-down list, select a key size for the key pair (in bits).

   The possible selections depend on the cryptographic provider you selected. Usually these values are “1024”, “2048”, and “4096”.

   For more information on supported key sizes, see the *RSA Keon Validation Server Installation Guide*. For more information on key sizes available with supported HSMs, see “appendix B. Cryptographic Hardware Interoperability”.

   The maximum key size supported by Microsoft Internet Explorer is normally 1024, but this can vary based on the cryptographic provider you select. The maximum key size supported by Netscape Navigator is 2048.

5. If you selected **Software** from the **Cryptographic Provider** drop-down list:

   a. In the **Enter Passphrase** text box, enter a passphrase to protect the signer’s private (signing) key.

      The passphrase must be at least eight characters long and contain at least one numeric and one alphabetic character.

   b. In the **Confirm Passphrase** text box, re-enter the passphrase.

6. Click **Save** to save this signer in the Keon VS database.

   The Signer Search/List page appears.

   The confirmation message “Your changes to <signer nickname> have been saved.” appears in the message area near the top of the page.

   Keon VS issues a self-signed certificate for the signer and selects it as the default signer’s default certificate.

7. If desired, restart Keon VS services (see “Starting Keon VS” on page 21).

Note: It is recommended that you restart Keon VS services after you create a signer. This is due to the fact that the OCSP Server is based on Apache 2.0, which starts a parent and a child process, and signer’s key is known only to the child process on creation. Restarting Keon VS services makes the signer’s key known to the parent process.

Creating a Status Source

To create a status source:

1. From the top-level menu, select **Status Sources > Add New**.

   The Add Status Source page appears.

2. In the **Status Source Nickname** text box, enter a status source nickname.

   You can enter any combination of alphanumeric characters to uniquely identify the status source. “Source 1” is the default value.

3. In the **Status Source Type** drop-down list, select a type.

4. In the **Retrieval Method** drop-down list, select a retrieval method.

   The possible selections depend on the status source type you selected. For OCSP, **HTTP** is the only option. For Revocation List, you may select **LDAP** or **Manual**.

5. If you selected a value other than **Manual** from the **Retrieval Method** drop-down list:

   a. In the **Hostname** text box, enter the hostname of the status source.

      This is the location of the remote server or database (directory server) providing status information.

   b. In the **Port Number** text box, enter the port number of the status source.

      This identifies the location within the remote server or database where Keon VS will find the status information.

6. Click **Save** to save this status source in the Keon VS database.

   The Status Source Search/List page appears.

   The confirmation message “Your changes to <status source nickname> have been saved.” appears in the message area near the top of the page.

Importing a CA Certificate

To import a CA certificate:

1. From the top-level menu, select **CAs > Add New**.

   The Add CA page appears.

2. In the **CA Nickname**, enter a CA nickname.

   You can enter any combination of alphanumeric characters that uniquely identifies the CA. “CA 1” is the default value.

3. Enter the CA certificate. Do one of the following:

   - Paste the Base64 format of the CA certificate text into the text box.
   - Browse the local file system for the filename containing the CA certificate or enter the filename in the text box.

4. Click **Save** to save this CA in the Keon VS database.

   The CA Search/List page appears.

   The confirmation message "Your changes to <CA nickname> have been saved." appears in the message area near the top of the page.

   If you want to import more than one CA, click **Save & Add Another**. The confirmation message "<CA nickname> has been saved." is displayed near the top of the page. All text fields are reset to their default values.

By default, the signer and status source for all new CAs are the default signer and default status source.

Configuring OCSP Requests and Responses

Before receiving requests (or queries) from or sending responses back to OCSP clients, you may want to modify how Keon VS handles OCSP requests and responses. You can make these modifications on a system-wide or per-CA basis.

Keon VS supports the following modes of OCSP request validation:

- No validation is performed (default)
- Accepts unsigned requests, but performs validation on signed requests
- Rejects unsigned requests and performs validation on signed requests

When you configure Keon VS to validate signed requests, Keon VS will support the following levels of validation:

- A signed request with a signature that Keon VS can verify (default)
- In addition to the previous level, a requestor's certificate that was issued by a known CA (or chains to a known CA)

  Important: If the OCSP validation level is set to require known requestors, the CA purpose for the requestor's CA must include verify OCSP clients. If it does not, Keon VS will return the unauthorized OCSP response.

- In addition to the previous two levels, a requestor's certificate status that is not suspended, revoked, or unknown

The selections made per-CA override the system-wide settings.

To view the default system-wide OCSP configuration or the OCSP configuration for a particular CA (and make possible modifications), see Help .

chapter 3. Managing Users

This chapter describes the roles and responsibilities of the users of RSA Keon Validation Server (Keon VS), as well as how to set up additional users to use Keon VS.

All users must follow the security practices established by their organization.

This chapter contains the following topics:

Roles and Responsibilities

The users of Keon VS are also called Administrators.

Administrators

Administrators install, configure, and manage Keon VS. An administrator can add other administrators.

Installing Keon VS

The Administrator has primary responsibility for installing and upgrading Keon VS. Before installing or upgrading Keon VS, the Administrator should consult the pre-installation and post-installation checklists provided in the *RSA Keon Validation Server Installation Guide*.

Adding Administrators

During installation, one Administrator with access privileges to Keon VS is created. After installation, the Administrator can add an unlimited number of Administrators to Keon VS. Multiple Administrators can share workloads and serve as backup in case other Administrators are absent. Administrator tasks may include planning the implementation, installing or upgrading Keon VS, configuring Keon VS, and managing daily administration tasks.

Performing Initial Configuration

After installing Keon VS, Administrators must perform certain tasks before Keon VS can send certificate status responses. See “Using Keon VS the First Time” on page 24 for more information.

Managing Daily Tasks

After installing Keon VS and performing initial configuration tasks, Administrators can choose to add or delete Administrators, manage certificates issued by the System CA, manage CAs, signers, and status sources, and configure Keon VS.

Administrators manage the imported CA certificates and revocation lists.

Finally, all Administrators can also configure or reconfigure:

- Event logging
- User authentication
- Online Certificate Status Protocol (OCSP) request and response configuration

Configuring User Authentication

Keon VS supports three types of user authentication:

- UserID / Password Authentication (default)
- Certificate Authentication
- UserID / Password and Certificate authentication

Your organization must decide which type of authentication best follows your security practices and your user’s needs.

How and where you obtain your certificate is outside the scope of Keon VS and will not be discussed here.

Note: It is recommended that you back up the `web.xml` file before you make any modifications.

Caution! Incorrect changes to the `web.xml`, `ssl.conf`, and `system.cert` files can make Keon VS unusable. For detailed instructions on how to configure user authentication, see the following sections of this guide or the SecurCare Online knowledge database. Discuss any additional changes you want to make with RSA Customer Support.

Supporting UserID / Password Authentication

To support userID and password authentication:

1. Shut down Keon VS services (see “Stopping Keon VS” on page 22).
2. Open `/<installed-dir>/GUIServer/webapps/vsadmin/WEB-INF/web.xml` in a text editor.

3. Comment out the `filter` and `filter-mapping` elements (enclose the lines in `web.xml` by `<!--` and `-->`). For example, you want the following result:

```
<!--
    <filter>
         <filter-name>SSL Password Authentication Filter
</filter-name>

         <filter-class>com.rsa.vs.ui.gui.filters.SSLPassword
AuthenticationFilter</filter-class></filter>
    -->
<!--
    <filter-mapping>
        <filter-name>SSL Password Authentication Filter
</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
    -->
```

4. Change the `auth-method` element (within the `login-config` element) to the value "`FORM`". For example, you want the following result:

```
<login-config>
   <auth-method>FORM</auth-method>
...
</login-config>
```

5. Ensure that the `form-login-config` element specifies a `form-login-page` and `form-error-page` and is not commented out (not enclosed by `<!--` and `-->`). For example, you want the following result:

```
<form-login-config>
   <form-login-page>/jsp/login.jsp</form-login-page>
   <form-error-page>/jsp/login.jsp?ErrorMessage=Login Failed
</form-error-page>
</form-login-config>
```

6. Save and close the file.
7. Restart Keon VS services (see "Starting Keon VS" on page 21).

All users must enter a userID and password to access the Keon VS GUI.

Supporting Certificate Authentication

To configure Keon VS for certificate authentication, the following sections show you how to:

- Modify `web.xml` so that Keon VS knows the type of authentication to use.
- Modify `ssl.conf` so that Keon VS will verify that a trusted CA issued the Administrator's certificate.
- Add the certificate of the CA that issued the Administrator's certificate to the list of trusted CAs.

To support certificate authentication:

1. Shut down Keon VS services (see "Stopping Keon VS" on page 22).
2. Configure Keon VS to authenticate the user's certificate (see "Authenticate user certificate" on page 32).

3. Add the CA certificate to the list of trusted CAs (see "Add CA certificate" on page 33).
4. Open `/<installed-dir>/GUIServer/webapps/vsadmin/WEB-INF/web.xml` in a text editor.
5. Comment out the `filter` and `filter-mapping` elements (enclose the lines in `web.xml` by `<!--` and `-->`).

   For an example on how to comment out element, go to step 2 under "Supporting UserID / Password Authentication" on page 30.
6. Change the `auth-method` element (within the `login-config` element) to `CLIENT-CERT`. For example, you want the following result:

```
<login-config>
   <auth-method>CLIENT-CERT</auth-method>
...
</login-config>
```

7. Comment out the `form-login-config` element (enclose the lines by `<!--` and `-->`). For example, you want the following result:

```
<!--
<form-login-config>
   <form-login-page>/jsp/login.jsp</form-login-page>
   <form-error-page>/jsp/login.jsp?ErrorMessage=Login Failed
</form-error-page>
</form-login-config>
-->
```

8. Save and close the file.
9. Restart Keon VS services (see "Starting Keon VS" on page 21).

All users must have a certificate installed in their browser to access the Keon VS GUI.

Authenticate user certificate

You must configure Keon VS to require verification of the Administrator's certificate.

To configure Keon VS to authenticate a user's certificate:

1. Open `/<installed-dir>/ValidationServer/conf/ssl.conf` in a text editor.
2. Search for the Keon VS UI Server `VirtualHost` definition (it loads `jk2_module`).
3. In this definition, change the `SSLVerifyClient` directive to `required`. For example, you want the following result:

```
SSLVerifyClient required
```

4. Note the filename and location of the trusted CA list in the `SSLCACertificateFile` directive. For example:

```
SSLCACertificateFile "<installed-dir>/ValidationServer/
../SystemCA/system.cert"
```

5. Save and close the file.

Add CA certificate

You must supply Keon VS with information about the CA that issued their user certificate. Keon VS uses this CA certificate to verify that the Administrator certificate was issued by this trusted CA. Each user certificate may be issued by a different CA, so you must add each CA certificate separately.

For each CA certificate you add:

1. Obtain the Base64-encoded CA certificate and open it in a text editor.
2. Open the file noted in step 4 of the preceding section in a text editor. For example:

   `<installed-dir>/SystemCA/system.cert`
3. Append the CA certificate (in Base64 format) to the end of the `system.cert` file.

   Important: You must include the header and footer in each CA certificate that you add.
4. Save and close the `system.cert` file.

Supporting UserID / Password and Certificate Authentication

To support authentication that involves a userID, password and certificate, the following sections show you how to:

- Modify `web.xml` so that Keon VS knows the type of authentication to use.
- Modify `ssl.conf` so that Keon VS will verify that a trusted CA issued the Administrator's certificate.
- Add the certificate of the CA that issued the Administrator's certificate to the list of trusted CAs.

To support userID, password, and certificate authentication:

1. Shut down Keon VS services (see "Stopping Keon VS" on page 22).
2. Configure Keon VS to authenticate the user's certificate (see "Authenticate user certificate" on page 32).
3. Add the CA certificate to the list of trusted CAs (see "Add CA certificate" on page 33).
4. Open `/<installed-dir>/GUIServer/webapps/vsadmin/WEB-INF/web.xml` in a text editor.
5. Ensure that the `filter` and `filter-mapping` elements are not commented out (not enclosed by `<!--` and `-->`). For example:

```
<filter>
    <filter-name>SSL Password Authentication Filter
</filter-name>

    <filter-class>com.rsa.vs.ui.gui.filters.SSLPassword
AuthenticationFilter</filter-class>
    </filter>
```

```
<filter-mapping>
    <filter-name>SSL Password Authentication Filter
</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

6. Change the `auth-method` element (within the `login-config` element) to `FORM`. For example:

```
<login-config>
   <auth-method>FORM</auth-method>
...
</login-config>
```

7. Ensure that the `form-login-config` element specifies a `form-login-page` and `form-error-page` and is not commented out.

   For an example on how to comment out element, go to step 4 in "Supporting UserID / Password Authentication" on page 30.

8. Save and close the file.
9. Restart Keon VS services (see "Starting Keon VS" on page 21).

All users must enter a userID and password and have a certificate installed in their browser to access the Keon VS GUI.

Managing Users

You can store users in either the Keon VS database or your LDAP directory. One user (Administrator) was created during installation. By default, there is only one user (the Administrator created during installation) and that user exists in the Keon VS database.

Note: It is recommended that you back up the `server.xml` and `tomcat-users.xml` files before you make modifications.

Storing Users in the Keon VS database

You must supply each user that you add with a password. Each password must be eight characters long and contain at least one alphabetic and one numeric characater.

To select the Keon VS database to store users:

1. Open `/<installed-dir>/GUIServer/conf/server.xml` in a text editor.
2. Ensure that `MemoryRealm` is not commented out. For example:

```
<Realm className="org.apache.catalina.realm.MemoryRealm"
digest="SHA" debug="0"/>
```

3. Save and close the file.

To add a new user:

1. Change the current directory to `<installed-dir>/GUIServer/bin/ .`

2. Encrypt the user's password by typing the following at the command line:

   a. For Solaris platforms:

      ```
      ./digest.sh -a SHA <cleartext password>
      ```

      For example:

      ```
      ./digest.sh -a SHA "abcd1234"
      ```

   b. For Windows platforms:

      ```
      ./digest.bat -a SHA <cleartext password>
      ```

      For example:

      ```
      ./digest.bat -a SHA "abcd1234"
      ```

   The output is the encrypted password. For example:

   ```
   abcd1234:7ce0359f12857f2a90c7de465f40a95f01cb5da9
   ```

3. Copy the encrypted password (the part after the ":").

4. Open `/<installed-dir>/GUIServer/conf/tomcat-users.xml` in a text editor.

5. If you configured Keon VS to use only userID and password authentication (see "Supporting UserID / Password Authentication" on page 30):

   - Add the user's name and paste the encrypted password (from step 2) within the `tomcat-users` element. For example:

     ```
     <tomcat-users>
         <user name>="Administrator"
     password="e7d537e128158790157ea057bb883e0292a84930"
     roles="administrator" />
         <user name>="lsmith"
     password="7ce0359f12857f2a90c7de465f40a95f01cb5da9"
     roles="administrator" />
     </tomcat-users>
     ```

6. If you configured Keon VS to use certificate authentication (see "Supporting Certificate Authentication" on page 31 or "Supporting UserID / Password and Certificate Authentication" on page 33):

   - Add the subject DN of the user's certificate as the user's name and encrypted password (from step 2) within the tomcat-users element. For example, if the user's certificate has the subject DN "EMAILADDRESS=lsmith@rsasecurity.com, CN=Linda Smith, OU=Engineering, O=RSA Security, C=CA":

     ```
     <tomcat-users>
         <user name>="Administrator"
     password="e7d537e128158790157ea057bb883e0292a84930"
     roles="administrator" />
         <user name>="EMAILADDRESS=lsmith@rsasecurity.com,
     CN=Linda Smith, OU=Engineering, O=RSA Security, C=CA"
     password="7ce0359f12857f2a90c7de465f40a95f01cb5da9"
     roles="administrator" />
     </tomcat-users>
     ```

7. Save and close the file.

8. Restart Keon VS services (see "Starting Keon VS" on page 21).

To remove a user, delete the appropriate element information from the `tomcat-users.xml` file.

Storing Users in an LDAP directory

To select your database to store users:

1. Open `/<installed-dir>/GUIServer/conf/server.xml` in a text editor.
2. Ensure that `JNDIRealm` is not commented out. For example:

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
debug="0" connectionURL="ldap://hostname:ldapport"
userPassword="userPassword" userSearch="(uid={0})"
userBase="ou=people,dc=mycompany,dc=com"
roleBase="ou=groups,dc=mycompany,dc=com" roleName="cn"
roleSearch="(uniqueMember={0})"/>
```

Important: Passwords are not encrypted by default. You must add the digest attribute to the end of the `JNDIRealm` element and turn on digested password support in the LDAP directory.

3. Configure the LDAP attributes in the `JNDIRealm` element to reflect your database (LDAP directory).
4. Save and close the file.
5. Restart Keon VS services (see “Starting Keon VS” on page 21).

When you add a new user, the authentication method determines the username format. The following examples show the user entries in LDIF format:

- For only userID and password authentication:

```
# Define a user entry for Linda Smith
dn: uid=lsmith,ou=people,dc=rsasecurity,dc=com
objectClass: inetOrgPerson
uid: lsmith
sn: smith
cn: linda smith
mail: lsmith@rsasecurity.com
userPassword: lsmithpass
```

- For certificate authentication:

```
# Define a user entry for TLS certificate with subject DN
"EMAILADDRESS=lsmith@rsasecurity.com, CN=Linda Smith,
OU=Engineering, O=RSA Security, C=CA"
dn: uid=EMAILADDRESS=lsmith@rsasecurity.com\, CN=Linda Smith\,
OU=Engineering\, O=RSA Security\,
C=CA,ou=people,dc=rsasecurity,dc=com
objectClass: inetOrgPerson
uid: CN=Linda Smith, OU=Engineering, O=RSA Security, C=CA
sn: smith
cn: linda smith
mail: lsmith@rsasecurity.com
userPassword: lsmithpass
```

For more information on configuring the LDAP attributes and adding new users, go to:

```
http://jakarta.apache.org/tomcat/tomcat-4.1-doc/
realm-howto.html#JNDIRealm
```

chapter 4. Managing CAs

This chapter introduces Certificate Authorities (CAs), their role in RSA Keon Validation Server (Keon VS), and their relationship to revocation lists and the certificate status request/response process.

This chapter contains information on the following topics:

- "Managing CAs"
- "Managing Revocation Lists"
- "Selecting an OCSP Signer"
- "Selecting a Status Source"
- "Revoking Certificates Locally"

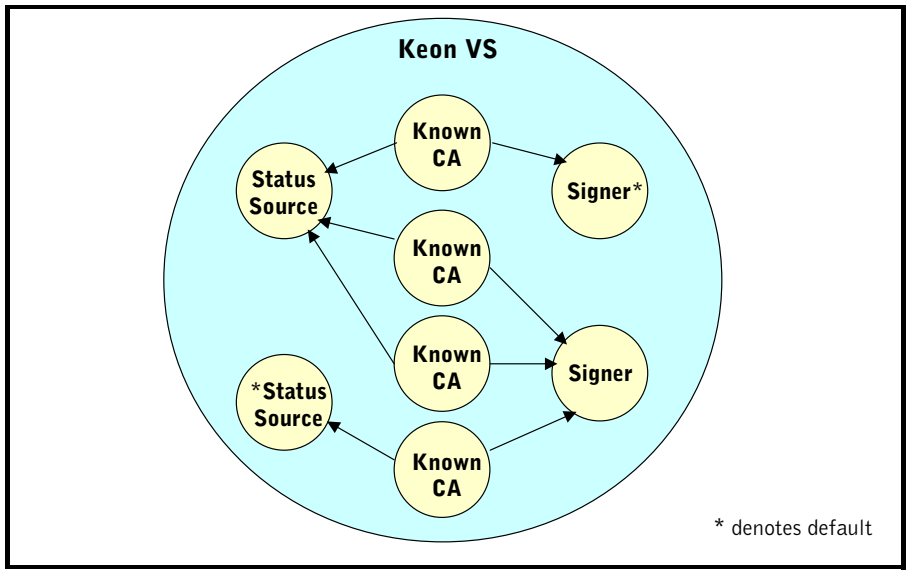
Managing CAs

Before Keon VS can return the status of a certificate issued by a CA, Keon VS must have a way of recognizing the CA. Importing the CA certificate makes the CA known to Keon VS.

Keon VS can support multiple CAs. You associate each *known CA* with one signer and one status source; however, you can associate a signer or status source with more than one CA.

Figure 1 illustrates possible relationships between CAs, signers and status sources.

Figure 1: CAs, signers, and status sources



Managing CAs in Keon VS involves the following tasks:

- Adding a CA by importing the CA certificate
- Modifying the CA nickname
- Modifying the list of CA purposes
- Locally revoking or reinstating the CA
- Selecting a signer and signer certificate
- Selecting a status source
- Deleting a CA

Note: When you delete a CA in Keon VS, Keon VS also deletes the CA certificate and any revocation lists for that CA.

You can also manage the following CA-related items in Keon VS:

- Revocation lists—complete certificate revocation lists (CRLs), delta CRLs, and authority revocation lists (ARLs)
- Certificates issued by known CAs

For instructions on how to perform these tasks, see Help.

Adding a CA

You make a CA known to Keon VS by importing a CA certificate. For the certificate to be accepted by Keon VS, it must exhibit the following characteristics:

- It cannot contain the same public key as a currently known CA.
- It must contain either a Subject DN or a Subject Alternative Name extension.
- It must contain either an Issuer DN or a Issuer Alternative Name extension.
- If the CA certificate contains a Basic Constraints extension, the cA field must be set to true.

You can import the certificate in one of two ways:

- Pasting the Base64 (PEM-encoded) format of the certificate into a text box
- Entering the filename where the Base64 or binary (DER-encoded) format of the certificate is stored or by browsing for the filename

Specifying Purposes of a CA

An Administrator can specify a known CA's purposes. A known CA can have the following purposes:

- Provide certificate status—process status requests for CA and end-entity certificates issued by this CA
- Verify OCSP clients—accept signed requests only from clients with a certificate issued by this CA

Important: If the OCSP validation level is set to require known requestors, the CA purpose for the requestor's CA must include verify OCSP clients. If it does not, Keon VS wil return the unauthorized OCSP response.

By default, both purposes are assigned to a new CA.

CA States

The local state of a CA certificate lets Keon VS know whether or not it can trust the CA certificate. There are two ways that Keon VS can change the local state of a CA certificate: by obtaining the new status from the revocation list published by a CA, or by the Administrator locally revoking the CA.

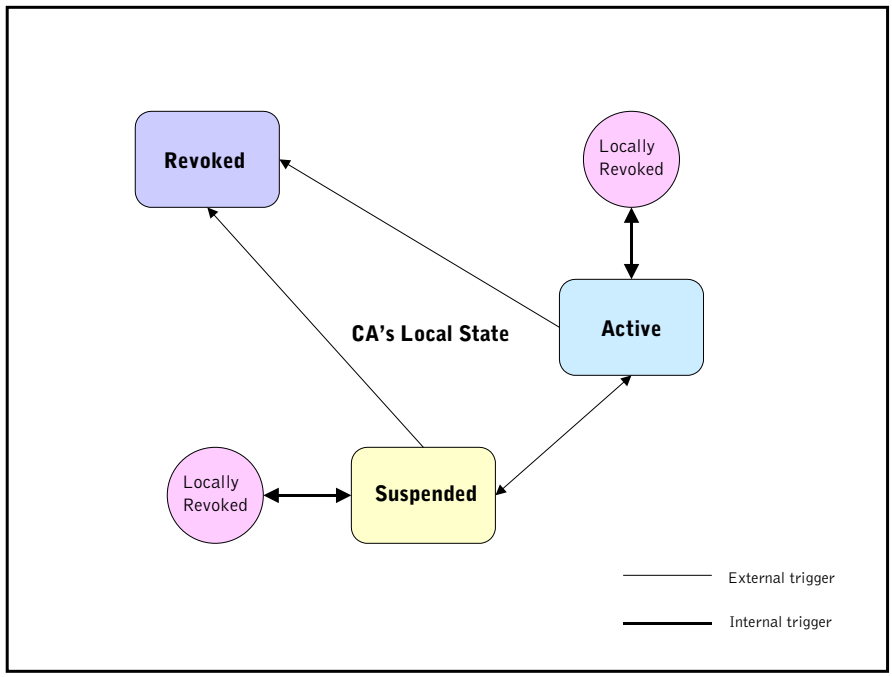
When a CA certificate is first imported, the local state of the CA is the same as the status of the CA certificate (usually *active*). The local state becomes *suspended* or *revoked* if a published revocation list marks the CA certificate status as suspended or revoked. A suspended certificate becomes active again if the certificate is later *reinstated*. Once a certificate is revoked it stays revoked.

An Administrator can *locally revoke* a CA certificate that is active or suspended. You would locally revoke a known CA when the CA is no longer trusted, but has not yet been formally revoked through entry in a revocation list. If a CA certificate is locally revoked, Keon VS will return a status of `revoked` for status requests on any certificates issued by that CA; the certificate's status external to Keon VS is not altered.

If the locally revoked CA can be trusted again, you can reinstate it.

Figure 2 illustrates the local states of known CAs.

Figure 2: When can a CA be locally revoked?



Managing Revocation Lists

You can configure Keon VS to import the following revocation lists to obtain certificate status:

- Complete CRL—a list of certificates that have been suspended or revoked by a particular CA
- Delta CRL—a list of certificates that have been suspended, reinstated, or revoked by a particular CA since the last complete CRL
- ARL—a list of CA certificates that have been suspended or revoked by a particular CA

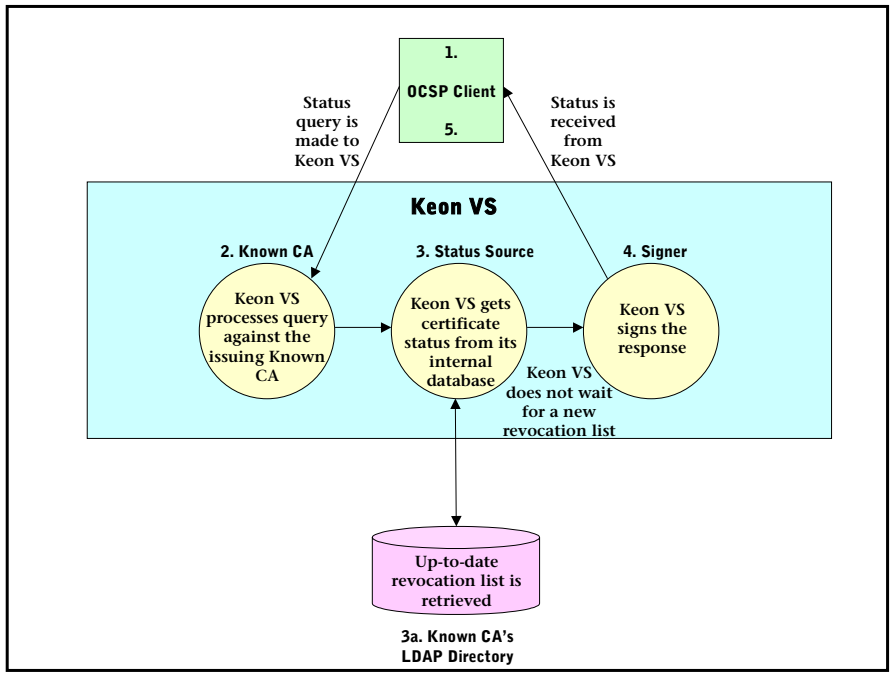
A revocation list is valid until the refresh time that is contained in it.

You can set up Keon VS to import revocation lists automatically or manually. Keon VS automatically retrieves a revocation list based on several factors: whether or not the revocation list has been retrieved before, whether or not the refresh time in the revocation list has been reached, and whether or not the Keon VS refresh time has been reached. For more information on retrieving revocation lists, see "Using a Revocation List-Based Status Source" on page 52.

After obtaining the revocation list(s), Keon VS processes the certificate status information published in the revocation list(s) to Keon VS's internal database and responds to certificate status queries with this information. The certificate status can be one of: `good`, `revoked`, or `unknown`. Keon VS processes all types of revocation lists in the same manner.

Figure 3 illustrates the flow of information through Keon VS when Keon VS automatically retrieves revocation lists from an LDAP directory.

Figure 3: Keon VS certificate status with revocation list retrieval



If Keon VS receives a certificate status query for a certificate, Keon VS responds with a status of `unknown` if it has no current information about this certificate (it is not included in any current revocation lists). If the status of a certificate is listed in an expired revocation list as revoked, Keon VS responds with a `revoked`, not `unknown`, status. At the same time that Keon VS is responding to the certificate status query, Keon VS retrieves the latest revocation list for this certificate. If configured to wait for fresh data, Keon VS waits to respond until retrieves the latest revocation list.

To avoid responding to certificate status queries with `unknown`, you can configure Keon VS with a grace period. For revocation list-based status sources, you can also configure Keon VS with a refresh time. For more information on refresh time and grace period, see "Using a Revocation List-Based Status Source" on page 52.

When a known CA certificate is listed as revoked in a revocation list, Keon VS considers the CA certificate and any certificate directly issued by the CA as revoked.

If a revocation list is rejected for any reason (for example, the CRL number is less than the previously imported revocation list or the delta CRL does not contain a delta CRL indicator extension), the fact that it was rejected is logged to the current audit log file, and Keon VS saves a copy of the rejected revocation list in the local file system.

Selecting an OCSP Signer

When you first add a CA to Keon VS, the default signer for Keon VS becomes the signer for that CA. This signer signs all OCSP responses for all certificates issued by the CA. By default, Keon VS will automatically select the default signer's certificate for that CA. Keon VS includes this signer certificate in all OCSP responses for all certificates issued by the CA.

You can replace the initial signer selection with another signer at any time. You can also replace the initial signer certificate selection with another signer certificate for that signer. Keon VS includes the signer certificate in all OCSP responses for all certificates issued by the CA.

If you delete the current signer for a CA, Keon VS reassigns the default signer as the signer for that CA.

For more information on OCSP signers, see "chapter 5. Managing OCSP Signers".

Selecting a Status Source

When you first add a CA to Keon VS, the default status source for Keon VS becomes the status source for that CA. This status source provides the location to retrieve certificate status for all certificates issued by the CA.

You can replace the initial status source selection with another status source at any time.

If you delete the current status source for a CA, Keon VS reassigns the default status source as the status source for that CA.

For more information on status sources, see "chapter 6. Managing Status Sources".

Revoking Certificates Locally

When a certificate issued by a CA appears on a revocation list, the local state of the certificate can be one of *suspended* or *revoked*.

Any certificate issued by a known CA can be *locally revoked* if it is not listed in a revocation lists or appears in a revocation list as suspended. To locally revoke a certificate, you must import it into Keon VS. You would locally revoke a certificate when it is no longer trusted, but has not yet been formally revoked through entry in a revocation list. Locally revoking a certificate causes Keon VS to report its status as `revoked`. Locally revoking a certificate does not alter the certificate's status external to Keon VS.

You cannot locally revoke or reinstate a certificate when it has appeared on a revocation list as revoked.

chapter 5. Managing OCSP Signers

This chapter introduces Online Certificate Status Protocol (OCSP) signers, their role in RSA Keon Validation Server (Keon VS), and their relationship to CAs and the certificate status request/response process.

This chapter contains information on the following topics:

- "Managing Signers"
- "Managing Requests for Signer Certificates"
- "Managing Signer Certificates"

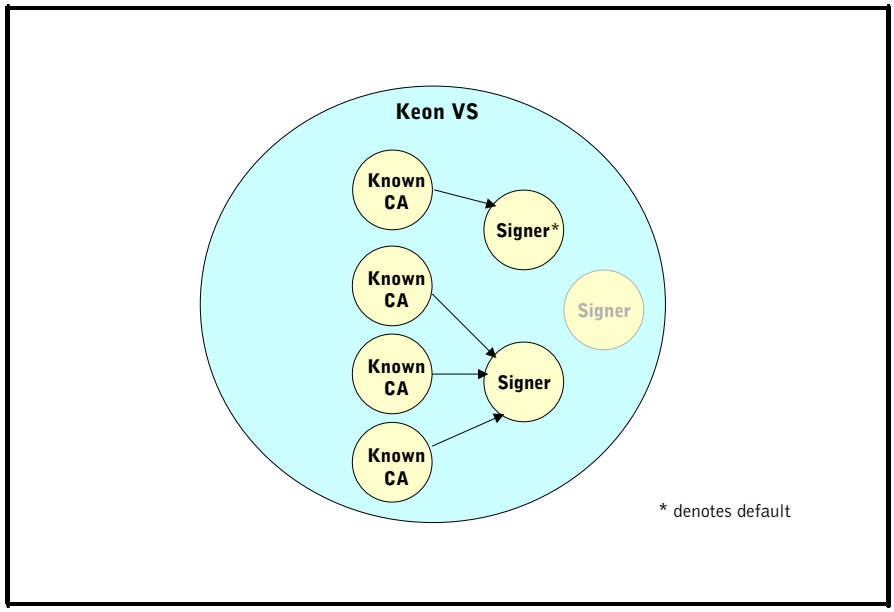
Managing Signers

In Keon VS, the entities that sign OCSP responses for a particular CA are called OCSP signers (or just *signers*). Signers have a self-signed certificate and can have certificates issued by a CAs.

Keon VS can support multiple signers and multiple *signer certificates*. You associate each known CA with only one signer; however, you can associate a signer with zero, one, or more than one CA. A signer that is not associated with any CAs is called *inactive*. For more information on CAs, see "chapter 4. Managing CAs".

Figure 4 illustrates the possible relationship between CAs and signers.

Figure 4: CAs and signers



Keon VS can respond to an OCSP client about any certificate. If the certificate was issued by a CA that is unknown to Keon VS, the response will have the value unknown. If the certificate was issued by a known CA, Keon VS signs the response with a signer certificate (that may have been issued by that known CA).

Keon VS can manage multiple signers and automatically keeps track of the various signer certificates and certificate requests.

All software-based OCSP signers require a passphrase to protect the private key. You can modify this passphrase at any time after you create the signer.

The first signer added to Keon VS is automatically selected as the default signer. You can set another signer as the default at any time.

Managing signers in Keon VS involves the following tasks:

- Adding signers

  Note: It is recommended that you restart Keon VS services after you create a signer (see "Starting Keon VS" on page 21).

- Modifying signer nicknames
- Setting a default signer
- Requesting signer certificates from a CA
- Adding signer certificates
- Setting a default signer certificate
- Viewing a list of CAs associated with a signer
- Deleting signers

  Note: You cannot delete the current default signer. You must select another signer as the default, and then delete the signer. Any CAs associated with a deleted signer are automatically associated to the new default signer.

  When deleting a signer in Keon VS, the signer's key pair, signer certificates, and signer cerificate requests are also deleted.

- Deleting signer certificates
- Deleting requests for signer certificates

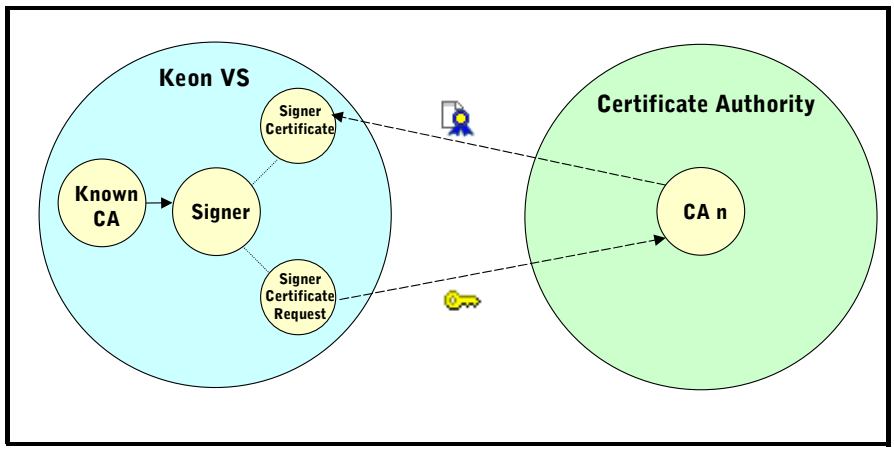
For instructions on how to perform these tasks, see Help.

Managing Requests for Signer Certificates

To obtain a signer certificate, you must first create a certificate request. Once you determine which CA you want to issue the certificate, you must send the request (via e-mail, for example) to the installation where the CA resides. Keon VS allows you to download the request to a file. The CA issues a certificate and returns it to you. Now you can add the signer certificate into the Keon VS database to use when signing OCSP responses.

Figure 5 illustrates the certificate request process.

Figure 5: Certificate request process



Keon VS can manage multiple requests. You can reuse any request (send it to another CA for certificate issuance). You can delete requests singly or in groups.

When creating a signer certificate request, you must specify at least one of the following distinguished name (DN) attributes:

- Common name
- Organizational unit
- Organization
- Locality
- Province/state
- Country

By default, Keon VS creates signer certificate requests with the following extensions:

- Extended key usage
- OCSP nocheck
- Key usage

For more information on these extensions, see Help.

Managing Signer Certificates

When you add a signer to Keon VS, a self-signed certificate is created and is selected as the signer's default certificate. You can set another signer certificate as the default at any time.

The self-signed certificate is valid for 12 months from the time of creation. It is automatically renewed when at least 11 months have elapsed since it was issued and it is about to be used in an OCSP response.

You must add the CA-issued signer certificates into Keon VS. If you import a second certificate (issued by the same CA) for a particular signer, then the new certificate replaces the existing one.

To renew a signer certificate, you can reuse a previous request or create another request for it. You must send the request to the installation where the CA resides for certificate reissuance.

You can delete signer certificates singly or in groups.

When Keon VS receives an OCSP request for a certificate issued by a known CA, Keon VS will include one of the following signer certificates in the OCSP response:

- Signer certificate issued by the same CA
- Default signer certificate
- Another of the signer's signer certificates

When Keon VS receives an OCSP request for a certificate issued by an unknown CA, Keon VS includes the default signer certificate in the OCSP response.

If you imported the signer certificate as part of the chain of certificates, the entire chain is included in the OCSP response.

chapter 6. Managing Status Sources

This chapter introduces status sources, their role in RSA Keon Validation Server (Keon VS), and their relationship to CAs and the certificate status request/response process.

This chapter contains information on the following topics:

- "Managing Status Sources"
- "Using an OCSP-Based Status Source"
- "Using a Revocation List-Based Status Source"

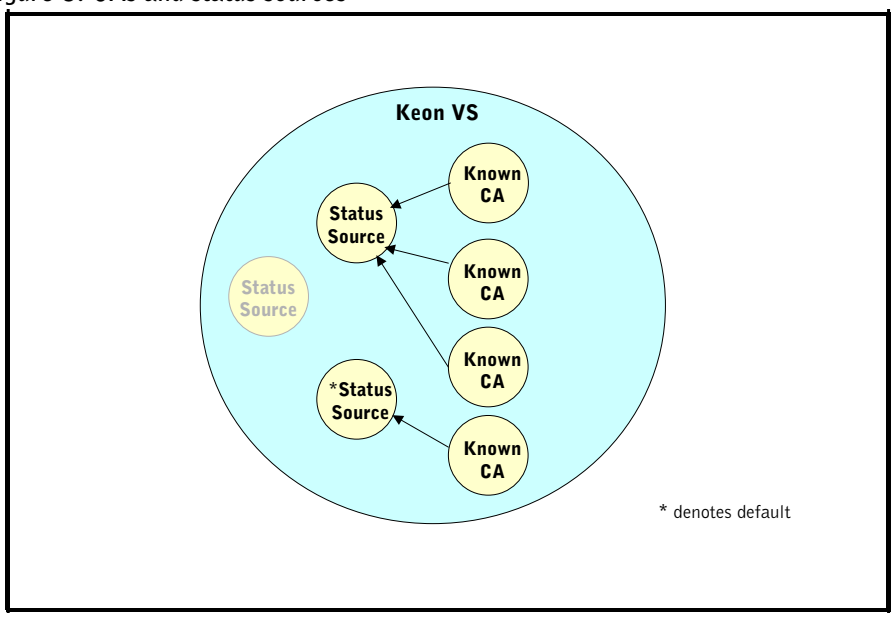
Managing Status Sources

In Keon VS, the location and method for obtaining status about certificates are defined in an entity called a *status source*.

Keon VS can support multiple status sources. You associate each known CA with only one status source; however, you can associate a status source with zero, one, or more than one CA. You can configure a status source without associating it with any known CAs. For more information on CAs, see "chapter 4. Managing CAs".

Figure 6 illustrates the possible relationship between CAs and status sources.

Figure 6: CAs and status sources



Keon VS supports the following two types of status sources:

- *OCSP-based*—Keon VS obtains certificate status from other OCSP servers
- *revocation list-based*—Keon VS obtains certificate status from revocation lists published by known CAs

The type of status source determines how Keon VS retrieves status (known as the *retreival method*) and how Keon VS contacts the status source.

Keon VS can manage multiple status sources and automatically keeps track of the various locations and methods and times of retrieval.

The first status source you add to Keon VS is automatically becomes the default status source. You can set another status source as the default at any time.

Managing status sources in Keon VS involves the following tasks:

- Adding status sources
- Modifying the status source nicknames
- Setting a default status source
- Modifying the retrieval method for status sources
- Configuring advanced settings for status sources
- Deciding whether or not to wait for fresh data
- Deleting status sources

Note: You cannot delete the current default status source. You must select another status source as the default, and then delete the status source. Any CAs associated to a deleted status source are automatically associated to the new default status source.

For instructions on how to perform these tasks, see Help.

Using an OCSP-Based Status Source

Keon VS supports the Hypertext Transfer Protocol (HTTP) for retrieving OCSP status.

If you configure Keon VS to use an OCSP-based status source for a known CA, Keon VS obtains certificate status by querying another OCSP server in two ways:

- *Proxying*—Keon VS passes the unchanged OCSP client requests to a remote OCSP server, and returns the unchanged remote server's response to the client
- *Forwarding*—an OCSP client request triggers Keon VS to send a second OCSP request to a remote OCSP server and use the remote server's response to construct its own response (default)

If you configure Keon VS for OCSP proxying, Keon VS will:

- Ignore signatures on remote responses
- Ignore any extensions on the remote response (even if critical)

If you configure Keon VS for OCSP forwarding, Keon VS will:

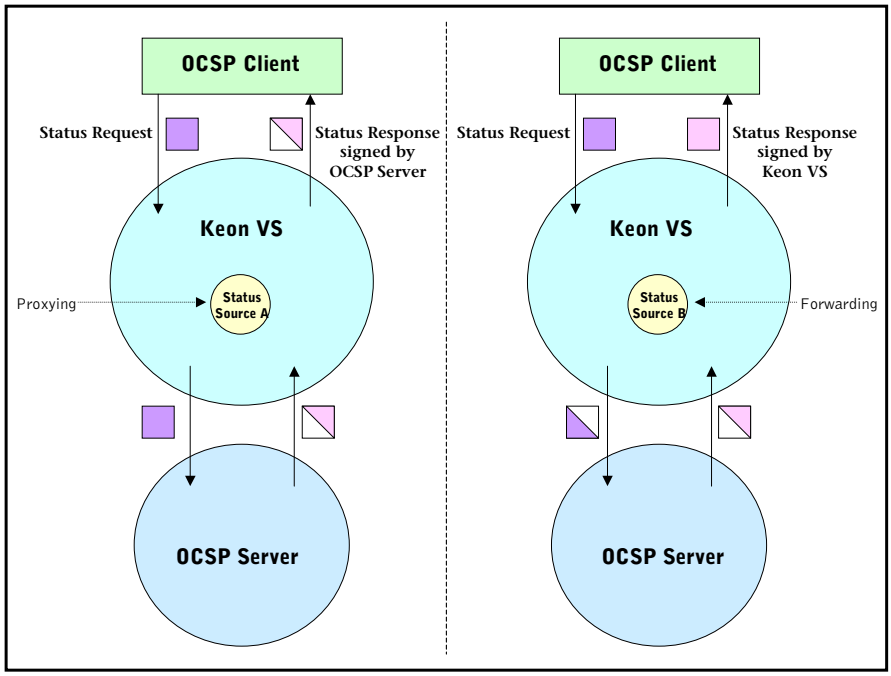
- Verify signatures on remote responses (rejecting the response if signature fails to verify and using `unknown` as status value)
- Process remote responses containing certain extensions (nonce, CRL references, archive cutoff, and CRL entry extensions) and reject any response that contains an unknown critical extension

- Sign the request sent to the remote server with the default signer, if you have configured Keon VS to sign requests

Keon VS logs any rejected responses to the current audit log file.

Figure 7 illustrates the differences between proxying and forwarding.

Figure 7: OCSP proxying and forwarding



You can, optionally, verify the remote server's OCSP responses using a known CA certificate or the remote OCSP server's certificate. You may want to check the status of the OCSP server's response-signing certificate.

For each OCSP-based status source, you can specify when Keon VS will retrieve updated certificate status information from the remote OCSP server. You can configure the *refresh time* to be one of the following times:

- The next update time stored in the last response received from the remote OCSP server (default)
- A number of seconds, minutes, hours, days, or weeks after the last response was received
- A number of seconds, minutes, hours, days, or weeks before the next update time stored in the last response

After Keon VS receives a remote OCSP response for a given certificate, any attempt to retrieve a new response for that same certificate will not occur until after the refresh time has passed. During this time, Keon VS will reuse data from the previous remote response.

For each OCSP-based status source, you can also specify a *grace period*, a period of time after the refresh time has elapsed during which data from a previous remote response can be reused. The status value used in the response depends on the value of the grace period:

- When a remote response is in its grace period, if Keon VS does not receive new status information for the same certificate, Keon VS uses the status value from the previous response (when configured for forwarding), or the entire previous response (when configured for proxying) unless the request contains a *nonce*. In this case, Keon VS returns a `tryLater` OCSP error.
- When a remote response has exceeded its grace period, if Keon VS does not receive new status information for the same certificate, Keon VS uses the `unknown` status value in its own response (when configured for forwarding), or `tryLater` is used (when configured for proxying).

Using a nonce in the request to a remote server allows Keon VS to verify that the response (that includes the same nonce) was in fact sent by the remote server in response to Keon VS's request, and is not being replayed by an adversary.

Using a Revocation List-Based Status Source

Keon VS supports the following protocols for retrieving revocation lists:

- Manual—manual import through the Keon VS GUI
- Lightweight Directory Access Protocol (LDAP) v2 or v3—using the DN of the LDAP object (the associated CA `SubjectDN`, by default, or a different LDAP object DN that you specify)

By default, Keon VS will retrieve complete Certificate Revocation Lists (CRLs) only. For more information on the types of revocation lists that Keon VS supports, see "Managing Revocation Lists" on page 42.

For each revocation list-based status source, you can specify when Keon VS will retrieve a fresh revocation list. You can configure the refresh time to be:

- The next update time stored in the revocation list (default)
- A number of seconds, minutes, hours, days, or weeks after obtaining the last revocation list
- A number of seconds, minutes, hours, days, or weeks before the next update time stored in therevocation list

For each revocation list-based status source, you can also specify a grace period, a period of time after the refresh time has elapsed during which data from a previous revocation list can be considered valid. The revocation list is considered *stale* once its status source's refresh time elapses, and the revocation list is considered *expired* once the referesh time plus the grace period elapses.

When Keon VS receives an OCSP request for a certificate issued by a known CA that uses a revocation lists-based status source, has not obtained a revocation list for a known CA as yet, or considers a revocation list stale, it attempts to obtain all revocation lists that it is configured to retrieve.

Keon VS reports any certificate that is not listed on a current or stale revocation list as having a `good` status, and any certificate that is not listed on an expired revocation list as having an `unknown` status. When the grace period has elapsed, Keon VS will report any certificate listed on the revocation list according to its status in the Keon VS database.

chapter 7. Managing Audit Logs

This chapter describes event logging operations in RSA Keon Validation Server (Keon VS). The OCSP Server and the UI Server log all operational events, such as certificate and revocation list import, to an audit log file (also called an audit log).

This chapter contains information on the following topics:

Overview of Audit Log Files

Keon VS records log entries for operational and system events and distributes the audit log in a textual format.

The filename and location of the audit log is in the `httpd.conf` configuration file. The default value is `<installed-dir>/Audit/audit.log` .

The format of each entry is:

```
[<date and time>][<user ID>][<event ID>][<success|failure>]
message
```

where:

- `<date and time>`—the date and time the event was logged
- `<user ID>`—the user ID of the Administrator that initiated the event, either through the GUI or command line (the string "no userID" is logged if the event was not initiated by an Administrator)
- `<event ID>`—the unique identifier for the event (a list of these events can be found in the Help)
- `<success|failure>`—whether the event was successful or failed
- `message`—event specific text

Events to be Logged

Keon VS enables logging by default. Startup and shutdown of the Keon VS installation are logged to the audit log. Any change in the Keon VS configuration (whether made through the Keon VS GUI or the command line) is logged as is the creation of CAs, OCSP signers, and status sources. You can configure Keon VS to log each events on successful completion or failure.

To view or modify the current event logging configuration, select **Configure System > Event Logging** from the top-level menu. Click **Save** to save any modification or **Cancel** to quit viewing this page.

By default, Keon VS logs all events on failure. Keon VS also logs the following events by default on success:

- Keon VS OCSP Server was started up
- Keon VS OCSP was shut down
- Initialize trace logging
- Import a revocation list
- Login by a user
- Audit log configuration at startup

chapter 8. Configuring Keon VS

This chapter describes the administrative tasks that you can perform to configure RSA Keon Validation Server (Keon VS) on an ongoing basis.

This chapter contains information on the following topics:

- "Changing System Passphrases"
- "Managing System Certificates"
- "Configuring OCSP over HTTPS"

Changing System Passphrases

All software-based system keys are passphrase protected as are all PKCS #12 files. You can change the passphrases entered during installation using the `<installed-dir>/Util/changepp` utility.

To change a passphrase:

1. Change the current directory to `<installed-dir>/Util/`
2. At the command line, type:

   ```
   changepp [-c <current passphrase>] [-n <new passphrase>] [-o
   <output file>] <key file|P12 file>
   ```

   If the `-o` option is omitted, the `key file` (or `P12 file`) is overwritten by the new file that has the passphrase changed. If the `-o` option is specified, the original file in the installation is replaced with the new file.

   For example:

   ```
   changepp -c 1234abcd -n abcd1234 systemca.key
   ```

Managing System Certificates

During installation, Keon VS creates the necessary Transport Layer Security (TLS) certificates and keys for the following servers, utilities, and users:

- Keon VS OCSP Server
- Keon VS UI Server
- Keon VS command line utility (vsadmin)
- Administrator using vsadmin to administer Keon VS
- Administrator using the GUI to administer Keon VS

You can replace these certificates and keys using the <installed-dir>/SystemCA/SystemCA application. At the command line, parameters are case-sensitive. Any error messages that are output by the SystemCA application are listed in "appendix A. Troubleshooting Keon VS".

Note: You must restart Keon VS after you modify any system certificates (see "Starting Keon VS" on page 21).

Caution! Incorrect changes to system certificate and key files can cripple Keon VS or make it unusable.For detailed instructions on how to manage system certificates, see the following sections of this guide or the SecurCare Online knowledge database. Discuss any additional changes you want to make with RSA Customer Support.

Generating New System Certificates and Keys

You can replace the system certificates and keys after installation. You may be required to do this because the certificates are about to expire or the certificates must be replaced or re-signed by an external CA as a result of your organization's security practices.

If you replace the System CA key file (the new key file is created in the <installed-dir>/SystemCA/ directory), you must re-sign and replace all TLS certificates (see "Generating system certificates and keys" on page 57). If you re-sign and replace the certificates and keys for the Administrator (using the GUI or vsadmin), you must generate new PKCS #12 files (see "Generating PKCS #12 files" on page 58).

Generating a New System CA

To generate a new System CA:

1. Change the current directory to <installed-dir>/SystemCA/
2. At the command line, type:

   ```
   SystemCA selfcert <provider> <alg> <key size> <validity>
   <template file> <cert file> <key file> <request file>
   ```

   where:

   - <provider> is either software-based or nCipher
   - <alg> is the signing algorithm (by default, SHA-1)
   - <key size> is either 1024 or 2048
   - <validity> is the validity period (in days) of the certificate
   - <template file> contains the certificate details such as Subject DN
   - <cert file> contains a Base64-encoded X.509 certificate
   - <key file> contains a Base64-encoded encrypted private key
   - <request file> contains a certificate request (in PKCS #10 format)

   For example:

   ```
   SystemCA selfcert "XCSP Default Provider" RSA-SHA1 2048 365
   "systemca.template" "systemca.cert" "systemca.key"
   "systemca.p10"
   ```

3. Type a passphrase when prompted.

   This passphrase is used to encrypt the private key.

A new certificate file, key file, and request file are created.

Generating system certificates and keys

To generate the system certificates and keys, complete these two steps:

- Generate a certificate request
- Issue a certificate

You can find the template files for the system certificates and keys in `<installed-dir>/SystemCA/`:

- `ocsp.template`—for the Keon VS OCSP Server
- `ui.template`—for the Keon VS UI Server
- `vsadmin.template`—for the Keon VS command line utility (vsadmin)
- `admin.template`—for the Administrator using vsadmin to administer Keon VS
- `guiadmin.template`—for the Administrator using the GUI to administer Keon VS

If you want to replace the certificates and keys, you must specify the certificate, key, and request files as follows:

- certificate files:
  - `<installed-dir>/ValidationServer/tls/certs/OCSPServer.cert`
  - `<installed-dir>/ValidationServer/tls/certs/UIServer.cert`
  - `<installed-dir>/ValidationServer/tls/certs/VSAdmin.cert`
  - `<installed-dir>/Util/admin.cert`
  - `<installed-dir>/GUIServer/webapps/vsadmin/config/guiadmin.cert`
- key files:
  - `<installed-dir>/ValidationServer/tls/keys/OCSPServer.key`
  - `<installed-dir>/ValidationServer/tls/keys/UIServer.key`
  - `<installed-dir>/ValidationServer/tls/keys/VSAdmin.key`
  - `<installed-dir>/Util/admin.key`
  - `<installed-dir>/GUIServer/webapps/vsadmin/config/guiadmin.key`
- request files:
  - `<installed-dir>/ValidationServer/tls/certs/OCSPServer.p10`
  - `<installed-dir>/ValidationServer/tls/certs/UIServer.p10`
  - `<installed-dir>/ValidationServer/tls/certs/VSAdmin.p10`
  - `<installed-dir>/Util/admin.p10`
  - `<installed-dir>/GUIServer/webapps/vsadmin/config/guiadmin.p10`

To generate a certificate request and issue the certificate:

1. Change the current directory to `<installed-dir>/SystemCA/`

2. At the command line, type:

   ```
   SystemCA genp10 <provider> <alg> <key size> <template file>
   <request file> <key file>
   ```

   where:

   - <provider> is either software-based or nCipher
   - <alg> is the signing algorithm (by default, SHA-1)
   - <key size> is either 1024 or 2048
   - <template file> contains the certificate details such as Subject DN
   - <request file> contains a certificate request (in PKCS #10 format)
   - <key file> contains a Base64-encoded encrypted private key

   For example, the following command generates a new key for the Administrator using vsadmin to administer Keon VS:

   ```
   SystemCA genp10 “XCSP Default Provider” RSA-SHA1 1024
   “vsadmin.template” “vsadmin.p10” “vsadmin.key”
   ```

3. Type a passphrase when prompted.

   This passphrase is used to encrypt the private key.

4. At the command line, type:

   ```
   SystemCA issuecert <provider> <issuer key file> <issuer cert
   file> <validity> <request file> <cert file>
   ```

   where:

   - <provider> is either software-based or nCipher
   - <issuer key file> contains a Base64-encoded encrypted private key of the issuer
   - <issuer cert file> contains a Base64-encoded X.509 certificate of the issuer
   - <validity> is the validity period (in days) of the certificate
   - <request file> contains a certificate request (in PKCS #10 format)
   - <cert file> contains a Base64-encoded X.509 certificate

   For example, the following command generates a new certificate for the Administrator using vsadmin to administer Keon VS:

   ```
   SystemCA issuecert “XCSP Default Provider” “systemca.key”
   “systemca.cert” 365 “vsadmin.p10” “vsadmin.cert”
   ```

Generating PKCS #12 files

Administering Keon VS requires a Java application and as such works with PKCS #12 files. These files are created from the certificate and keys files (created in the previous section). There are two such files in a Keon VS installation:

- <installed-dir>/Util/admin.p12
- <installed-dir>/GUIServer/webapps/vsadmin/config/guiadmin.p12

To generate a PKCS #12 file:

1. Change the current directory to <installed-dir>/SystemCA/

2. At the command line, type:

   ```
   SystemCA createp12 <friendly name> <key file> <cert file>
   <issuer cert file> <p12 file>
   ```

   where:

   - `<friendly name>` specifies an association between a name and a private key/certificate pair
   - `<key file>` contains a Base64-encoded encrypted private key
   - `<cert file>` contains a Base64-encoded X.509 certificate
   - `<issuer cert file>` contains a Base64-encoded X.509 certificate of the issuer
   - `<p12 file>` contains a PKCS# 12 (associates certificate and private key)

   For example, the following command generates a new PKCS #12 file certificate for the Administrator using the GUI to administer Keon VS:

   ```
   SystemCA createp12 Administrator “admin.key” “admin.cert”
   “systemca.cert” “../GUIServer/webapps/vsadmin/config/
   guiadmin.p12”
   ```

Replacing System Certificates

Replacing System CA certificate

During installation, Keon VS generates a certificate request for the System CA as well as the system certificate and key files. You can submit this request to an external CA (outside the Keon VS installation) to request a certificate.

The new System CA certificate file must contain a complete certificate chain (PKCS #7 format) including the new System CA certificate, the external CA certificate, and any certificates in the external CA certificate chain up to and including the root CA certificate.

Replacing system TLS certificates

If you replace the system certificates and keys as described in “Generating system certificates and keys” on page 57, you must replace certificates in the PKCS #12 files with the newly re-signed certificates.

To replace the certificate in the PKCS #12 file:

1. Change the current directory to `<installed-dir>/SystemCA/`
2. At the command line, type:

   ```
   SystemCA replacecert <cert file> <p12 file>
   ```

   where:

   - `<cert file>` contains a Base64-encoded X.509 certificate
   - `<p12 file>` contains a PKCS# 12 file (associates certificate and private key)

   For example, the following command replaces the `admin.cert` in the PKCS #12 file certificate for the Administrator using the vsadmin to administer Keon VS:

   ```
   SystemCA replacecert “admin.cert” “../Util/admin.p12”
   ```

Replacing the System CA

If you want to replace the System CA with an external CA (outside the Keon VS installation), you must:

- Replace the `<installed-dir>/SystemCA/system.cert` file with the external CA certificate chain (any certificates in the external CA certificate chain up to and including the root CA certificate)
- Replace the system TLS certificates using the external CA as the issuer (see "Generating system certificates and keys" on page 57)

Configuring OCSP over HTTPS

You can configure Keon VS to serve OCSP over HTTPS (secure HTTP). Only the Keon VS side of the connection is authenticated (using TLS). No authentication of OCSP clients is performed.

To configure Keon VS for OCSP over HTTPS:

1. Open the `<installed-dir>/ValidationServer/conf/httpd.conf` file in a text editor.
2. Uncomment the directives (shown in bold below) to enable server authenticated OCSP over HTTPS. (This script is found at the end of the `httpd.conf` file.) For example:

```
<VirtualHost _default_:8080>
    DocumentRoot /tmp
    ServerName myocsp.example.com
    LoadModule ocsp_module modules/mod_ocsp.so
    # Configuration for mod_ocsp
    ProtocolOCSP On

    #
    # Uncomment the following directives to enable server
    authenticated TLS
    # (i.e., OCSP over HTTPS) for this virtual host.
    #
    #SSLEngine on
    #SSLCipherSuite
    ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:
    +eNULL
    #SSLCertificateFile /installdir/ValidationServer/tls/
    certs/OCSPServer.cert
    #SSLCertificateKeyFile /installdir/ValidationServer/tls/
    keys/OCSPServer.key
    #SLVerifyClient none
</VirtualHost>
```

Important: You must restart Keon VS after you modify the `httpd.conf` file (see "Starting Keon VS" on page 21).

appendix A. **Troubleshooting Keon VS**

This appendix contains solutions to problems that you may encounter during the RSA Keon Validation Server (Keon VS) installation process.

If a problem occurs during installation, check the following:

- For last-minute information or bugs that exist in the installation process, see the R*SA Keon Validation Server README*, available from the `/doc/` directory on the CD-ROM
- Check the list below for a possible solution

You can configure Keon VS to generate trace log files, which RSA Customer Support can use to assist you in determining the source of any problems in a timely manner. Any system logs created by Keon VS will also be useful.

Trace Logging

Keon VS includes a tracing mechanism to reduce the time spent troubleshooting customers' problems and to provide the most detailed information possible to narrow down a solution quickly.

Note: Trace logs are intended as a source of diagnostics information for debugging purposes. However, in some cases, errors may be reported that do not indicate a failure or unexpected conditions. For example, a not found error may be a legitimate result when searching for a particular data value.

By default, Keon VS writes trace log data to the `vs_trace_log` file in the `<installed-dir>/logs` directory. You can find the name and location of the trace log file in the `TraceLogFile` directive in the `<installed-dir>/ValidationServer/conf/httpd.conf` file.

Each entry in the trace log file has the format:

```
[<date and time>] message
```

The `message` has no defined format, but where error conditions are logged, the filename, line number, and error code are included in the entry.

At startup or restart, the OCSP Server logs the state of its trace logging to the audit log and the trace log files.

If the trace log file cannot be created or opened, the OCSP Server logs a warning to the system log, but continues to operate.

System Logging

The purpose of the system log is to provide an external record of the system state, such as whether or not the system has started or stopped, and the state of the other logging subsystems (audit logging and trace logging). For the Solaris operating environment, the `syslog` facility is used (configuration in `\etc\syslog.conf`).

System log messages are logged to the system's default facility. The destination filename depends on the system's syslog configuration.

Each entry in the syslog file has the format:

```
<native header> [<event ID>] [<success|failure>] message
```

where:

- Date and time of the message is include in the `<native header>`.
- `<event ID>` is one of:
  - `OCSPServerStartup`—startup of the Keon VS OCSP Server
  - `OCSPServerShutdown`—shutdown of the Keon VS OCSP Server
  - `AuditLogConfig`—audit log configuration at startup
  - `TraceLogInit`—initialize trace logging
- Messages that exceed the maximum message size are logged as:

  ```
  <native header> [<part>] [<event ID>] [<success|failure>]
  message
  ```

  where `<part>` indicates the part number and the total number of parts.

Troubleshooting Keon VS

The OCSP response is `unknown` when it should be good

The solution depends on whether or not you configure Keon VS to use fresh status data (on the **System Settings** screen).

If you configure Keon VS not to use fresh status data, and the status source used to check certificate status uses revocation lists, then Keon VS will return `unknown` for all certificates if it has not imported any revocation lists, and it will return `unknown` for good certificates if the imported revocation lists have expired (current time is past the refresh time plus the grace period). A revocation list retrieval is still initiated, but Keon VS will not wait for the import to complete before returning responses. Once Keon VS successfully imports the new revocation list(s), then `good` status will be returned again.

Otherwise, the user can configure Keon VS to wait for fresh status data and Keon VS will wait for the fresh status before returing a response (but the client may conceivably have to wait a while for this).

Revocation List import logged late

When importing revocation lists, the importation is not logged until the operation is complete, so for large revocation lists there may be some delay before Keon VS logs the event.

Only successful importing of revocation lists is logged

When the CA is using multiple revocation list types, if one (or two) of the imports fails and the other(s) are successful, then Keon VS only logs the successful imports. The absence of the successful importation(s) from the audit log will indicate that a failure occurred. Examining the trace log shows the failure.

appendix B. Cryptographic Hardware Interoperability

This appendix provides information on how RSA Keon Validation Server (Keon VS) and cryptographic hardware products work together.

Note: RSA Security Interoperability Lab tests products on an ongoing basis. For additional interoperability information, visit the Support Web site.

nCipher nForce and nShield

Manufacturer: nCipher Corporation Ltd.

Web site: `http://www.ncipher.com`

Introduction

nCipher's nForce and nShield are devices that generate and protect private keys, and provide secure hardware key management for the RSA Keon product suite. nForce and nShield are based on nCipher's powerful encryption acceleration technology.

When using nCipher hardware (nForce and nShield), keys are never revealed to the outside world or even to the main memory of the Keon VS installation in an unencrypted format, vastly increasing the security of key data. Keon VS uses nCipher hardware to generate keys using true hardware-based random number generation, encrypt keys for secure storage, and guarantee key security in highly sensitive applications where federal standards level security is critical. Depending on how they are used, nCipher hardware devices are Federal Information Processing Standard 140 (FIPS 140-1) Level 3 compliant.

Keon VS supports the use of nCipher hardware modules nShield and nForce, and the nCipher software package 6.14 on the Solaris 8 platform.

Functionality

Keypair Generation

Keon VS can create keys using nCipher smart cards. These smart cards support key generation for key pairs based on RSA algorithms.

Key Storage

Keon VS supports the storage of cryptographic keys using nCipher smart cards.

Signing with Keys

Keon VS supports the usage of keys stored in nCipher's key storage for signing operations.

Important: Users should ensure that the required nCipher smart card is inserted into the nCipher card reader before initiating any signing operations through Keon VS. The smart card should not be removed from the card reader until signing operations are complete. The smart card is not needed for signing/verification if its PIN was entered at startup.

Configuration

To use the KeySafe application, you must install Java 1.3 or higher before you install the nCipher hardware and software.

Installing nCipher Hardware

Full installation instructions are available in nCipher's *Getting Started Guide* (start.pdf) on the nCipher installation CD-ROM. Instructions for upgrading the module firmware are available in nCipher's *nForce User Guide* (nforce.pdf) or *nShield User Guide* (nshield.pdf).

Installing the nCipher Server Software

Installation instructions are available in the nCipher user guide on the nCipher installation CD-ROM.

Note: On some older versions of the nCipher hardware, the server software cannot be upgraded to v6.14 as they are no longer supported by nCipher.

If the module type code is '2' or '4' (obtained by running the "enquiry" command), then the nCipher software cannot be upgraded beyond v5.x .

nCipher Security World

Before you can use the nCipher Hardware Security Module (HSM), you must create a Security World. A Security World consists of one or more hardware modules, a set of smart cards, and some encrypted data stored on computer.

In order to create a Security World you must set the hardware module in pre-initialization mode, create a Security World using the KeySafe application or the new-world command, and finally set the hardware module in operational state. Detailed information on the Security World can be found in the nCipher user guide.

Card Sets in the Security World

A Security World is designed to ensure all keys remain secure throughout their life cycle. Within a given security world, there are two types of card sets: an Administrator Card Set, and Operator Card Sets. The Administrator Card Set is used to control access to recovery functions, and is created during Security World initialization. There is only one per Security World.

Operator Card Sets are used to control access to application keys. Each user can access the keys protected by the Security World and protected by their Operator Card Set only. When you initialize a smart card, you must provide new Operator Card Set passphrases. The Operator Card Set smart cards are used by Keon VS to protect public and private keys. Further information on smart cards can be found in the nCipher user guide.

Note: The smart cards used as Operator Cards must be erased before reinitializing the nCipher module. Otherwise, these cards must be discarded because they cannot be used, erased, or reformatted without the old Security World key.

nCipher Smart Card Labels

Smart card labels are defined when the smart card is initialized. They are simply a name that you attach to the smart card to help you organize your smart cards and keep track of what each one is used for. Labels can only be changed by reinitializing the smart card.

Initializing nCipher Smart Cards for Use with Keon VS

nCipher smart cards must be initialized before they can be used with Keon VS. Use one of the following methods to initialize a new smart card:

- Using nCipher's KeySafe utility

  nCipher smart cards can be initialized using nCipher's KeySafe utility. For instructions, see "Creating Operator Card Sets" in the nCipher user guide.

- Using "createoc-simple"

  <path>/createoc-simple [--force] <module> <slot> <label> <persist> <timeout>
  where:

--force	Allows for the overwriting of non-blank cards
module	Module number of the HSM, usually 1
slot	Usually 0
label	Name of token
persist	Should be 'yes' or 'no'
timeout	Must be 0

  For example: createoc-simple 1 0 token1 no 0

  Refer to the nCipher user guide on the nCipher installation CD-ROM. Instructions for using "createoc-simple" are available in "Creating Operator Card Sets."

- Using “createocs” or “ckinittoken”

  Refer to the nCipher user guide on the nCipher installation CD-ROM. Instructions for using “createocs” are available in “Creating Operator Card Sets”.

Keon VS Installation

If nCipher hardware and server software are installed before Keon VS, the smart card must be initialized prior to beginning the Keon VS installation.

Caution! The nCipher smart card must be initialized with a passphrase.

Keon VS must be installed by the “root” user or by a user belonging to the nfast user group to allow operation with nCipher hardware.

Adding nForce or nShield Support to an Existing Keon VS Installation

To add smart card support to an existing Keon VS installation that does not use smart cards, you do not need to reinstall Keon VS.

To add nCipher smart card support to an existing installation:

1. Install the nCipher hardware and server software as described in previous sections. Make sure the nFast server is running.
2. Initialize a smart card and insert it into the reader.
3. Stop and restart the Keon VS services.

Recovery Features

All recovery options described below require that the recovery option must have been enabled when creating the Security World.

Loss of Smart Card

To recover from the loss of an nCipher smart card, nCipher’s replaceocs or sw-racs utility can be used to replace any lost smart card. For detailed instructions, see “Replacing Operator Card Sets” or “Replacing the Administrator Card Set”, in the nCipher user guide. For the replacement card to work with the existing servers that used the original smart cards:

- the replacement card set must have the same name as the original
- the old card set must be cleared and removed from the Security World

Loss of Hardware

The actual hardware module (nForce or nShield) needs to be replaced. After replacing the nCipher hardware, you must recreate the Security World using the new-world command, as documented in “Adding a Module to the Security World” in the nCipher user guide. If the replacement module had been used with another Security World, then you must initialize that module using initunit command before using new-world.

This procedure can be used to move the Security World from one machine to another, including across platforms.

Loss of Hard Drive Data

If files containing the Security World data are lost (for example, `/kmdata/` directory), you can simply restore the files from backup. Any data created since the last backup will be lost. If the complete nCipher software installation is damaged or lost, follow the procedure under "Loss of Hardware" on page 68, provided a backup of the Security World data exists.

Useful nCipher Commands

- **enquiry**—to confirm versions of server software and module firmware
- **ckcheckinst**—to confirm the nCipher PKCS #11 library version and to determine the label of any cards inserted in the reader
- **nfkminfo**—to get information about the Security World, i.e. whether recovery option is enabled

Glossary

Abstract Syntax Notation One (ASN.1)

Abstract Syntax Notation One (ASN.1) is an International Standards Organization (ISO) standard notation for defining the syntax of information data. It defines a number of simple data types and specifies a notation for referencing these types and for specifying values of these types.

Administrator

A person with an end-entity certificate who has access to certain administration features through the Administration Server, which may include the CA Operations, Administrator Operations, and System Configuration Workbenches. Administrator tasks may include planning the PKI implementation, installing Keon VS, performing initial configuration, and managing the daily administration tasks.

Anonymity

The ability of participating entities to use public keys while revealing only that information that is pertinent to the situation.

ARL

See **Authority Revocation List**.

Attribute Certificate

A certificate that emphasizes the holder's access rights and constraints. This is in contrast to identity certificates, which bind a distinguished name (DN) and a public key. Commonly, attribute certificates are issued with short validity periods and do not contain a public key value.

Audit Log

A tamper resistant log Keon VS uses to record operational and configuration changing events.

Authentication

Certificates are used to identify the author of a message or entity, such as a Web server or client. People or applications who receive a certificate can verify the identity of the certificate's owner and the validity of the certificate. This process is called authentication.

Authority Revocation List (ARL)

A list of CA certificates that have been revoked or suspended by a particular CA. ARLs can be used to check the status of CA certificates offline.

Backend

A collection of functions within the OCSP Server that are invoked when a request is made to a particular part of the LDAP directory information tree.

Base64

See **Privacy Enhanced Mail (PEM) Format**.

CA

See **Certificate Authority**.

CA Certificate

A certificate that identifies a CA. When a CA issues a certificate to a client, a server, or other entity, the certificate is signed by the CA's private key. The signature can be verified using the public key in the CA's certificate. See also **root CA**.

CA Purposes

CA purposes define the use of the CA within Keon VS. For example, if a CA's purpose is to provide certificate status, then Keon VS will process status requests for certificates issued by that CA. By default, Keon VS assigns the following purposes to a CA: provide certificate status and verify OCSP clients.

Certificate

Certificates are used to verify the identity of an individual, organization, Web server, or hardware device. They are also used to ensure non-repudiation in business transactions, as well as to enable confidentiality through the use of public-key encryption. Three main kinds of certificates are used in a PKI: CA certificates, server certificates (also referred to as SSL certificates), and end-entity certificates.

Certificate Authority (CA)

An entity that issues and manages certificates within a PKI. CAs are usually created and managed using a CA software application, such as Keon CA.

Certificate Extension

See **X.509 v3 Certificate Extension**.

Certificate Fingerprint

The MD5 of a PEM-encoded representation of the certificate (without the header) certificate used to identify a certificate when setting LDAP and Web server ACL rules.

Certificate ID

A certificate identifier used by Keon CA to search for or label a certificate in the Secure Directory.

Certificate Policy (CP)

A CP explains the conditions and limitations of use for a digital certificate.

Certificate Revocation List (CRL)

A list of certificates (CA or end-entity) for a particular CA. CRLs can be used to check the status of certificates offline.

See **Complete CRL** and **Delta CRL**.

Certification Practice Statement (CPS)

A CPS defines an organization's security policies for the issuance and management of certificates.

Client Certificate

See **End-Entity Certificate**.

Complete CRL

A complete CRL contains the serial numbers of certificates that a CA has revoked or suspended.

CRL

See **Certificate Revocation List**.

Cryptographic Provider

The library Keon VS uses for private-key cryptographic operations (such as key pair generation and digital signatures). The method is either software-based or hardware-based (using nCipher).

Delta CRL

A delta CRL contains the serial numbers of certificates that a CA has suspended, reinstated, or revoked since the last complete CRL.

Digital Signature Algorithm (DSA)

A digital signature algorithm used in the Digital Signature Standard (DSS) created by the U.S. government. For more information, see the standard designation `FIPS 186-2+ChangeNotice` at:

`http://csrc.nist.gov`

Directory

LDAP-based directories are databases that can be used to search for and retrieve attribute-value pairs. Directories can be configured to use (or support) authentication and access control protection. The schema of a directory describes the objects in the directory. The Keon Secure Directory schema defines objects that support PKI.

Distinguished Encoding Rules (DER)

The Distinguished Encoding Rules (DER) are an ASN.1 encoding standard.

DER encoding is used for signature calculation for end-entity certificates and revocation lists, that is CRLs, delta CRLs and ARLs. It is also known as **Binary**.

Distinguished Name (DN)

The DN of a certificate is formed from the combination of the following possible attributes in the certificate:

- Common Name
- User ID
- E-mail Address
- Organizational Unit
- Organization
- Locality
- State or Province
- Country
- Domain Component.

To avoid potential problems, all CAs in the PKI, including trusted CAs, should have a unique DN.

End-Entity Certificate

A certificate issued to an entity that cannot itself issue certificates (that is, the entity is not a CA). Because the entity that requests such a certificate is sometimes referred to as the client, end-entity certificates are sometimes called *client certificates*.

End User (or End-Entity)

An end user is an individual, group, or organization that either requests or holds an end-entity certificate. An end user can also be an individual who requests an end-entity certificate for a hardware device (such as a router), a server, a software application, or a piece of code. An end user that requests a certificate is sometimes called a requestor. An end user that is issued a certificate

is sometimes called a certificate owner, certificate subject, or end-entity. An end user that relies upon someone else's certificate to verify that other person's identity is sometimes called an end user, certificate user, or relying party.

Enterprise

An organization that uses computers and applications. In general use, this term is applied to businesses or organizations that operate on a large scale. Applications that are designed for these organizations are often referred to as *enterprise applications*.

Entity

A person, organization, or device (such as a router). In a PKI, an entity may be thought of as anything you can issue a certificate to.

Expired Status Data

Refers to the freshness of a revocation list or status value. A list or status value is considered expired once the refresh time plus the grace period elapse.

Extension

See **X.509 v3 Certificate Extension**.

FIPS 140-1 Level 2 & 3
FIPS 140-2 Level 2 & 3

A standard developed by the National Institute of Standards and Technology (NIST) for implementation of cryptographic modules. Level 3 provides greater security than Level 2.

Firewall

A firewall is a system designed to prevent unauthorized access to or from a private network.

Fresh Status Data

Refers to the freshness of a revocation list or status value. A list or status value is considered fresh if its status source's refresh time has not elapsed. For example, if the refresh time for Keon VS to retrieve a new list or status value has not arrived, the list or status value within the Keon VS database is considered fresh.

Fully Qualified Domain Name (FQDN)

The full name of a system, consisting of its local host name and its domain name. For example, "venera" is a host name and "venera.isi.edu" is the FQDN.

Grace Period

A period of time during which Keon VS can reuse a stale status value, but must also attempt to obtain a newer status value. For example, when a remote OCSP response is in its grace period, and Keon VS cannot fetch a new response for the same certificate, Keon VS will use the status value from the previous OCSP response. The grace period specifies how long after the refresh time that the previous status value is valid. A status value expires once its refresh time and grace period elapse.

Hardware Security Module (HSM)

Hardware used to perform cryptographic functions and store cryptographic keys in a secure fashion.

HTTP

Hypertext Transfer Protocol (HTTP) is a set of rules for exchanging files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Web browsers are HTTP clients that send requests to server machines. Users enter page requests by either typing in a URL or clicking a hypertext link. The browser builds an HTTP request for the user and sends it to the Internet Protocol (IP) address indicated in the URL. The HTTP daemon in the destination server receives the request and, after any necessary processing, returns the requested page.

HTTPS

HTTP over an SSL connection.

Identity Certificate

A certificate that links a public key value to a real-world entity such as a person, a computer, or a Web server. Server certificates, CA certificates, and most end-entity certificates are all examples of identity certificates.

Keon VS Installation

An instance of Keon Validation Server. This may comprise a single machine hosting single instances of the various Keon VS servers, or it may be a farm of servers residing behind a Network Address Translator machine such as a load balancer. The servers within a Keon VS installation are generally under a single administrative domain.

Key Pair

A public key and a private key associated with an entity that needs to authenticate its identity electronically or to sign or encrypt data. The public key is published, and the corresponding private key is kept secret. Data encrypted with the public key can be decrypted only with the private key.

Key Size

The size (in bits) of the key pair used to sign status responses. A larger key size provides greater security. Keon VS supports 1024, 2048, and 4096 bit keys.

Known CA

A CA that is know to the system. That is, the CA's certificate has been imported into the system.

Lightweight Directory Access Protocol (LDAP)

LDAP is the standard Internet protocol for accessing directory servers over a network. LDAP is a "lightweight" (smaller amount of overhead) version of DAP (Directory Access Protocol), which is part of X.500, a standard for directory services in a network. The Keon VS Secure Directory is an LDAP directory.

Locally Revoked Certificate

A certificate that is revoked within Keon VS, but not revoked by a CA. A locally revoked certificate is not listed on a revocation list. Keon VS will return a status of revoked for all enquires on this certificate's status.

Logging Server

The Logging Server records PKI events for one or more client applications such as the Secure Directory Server. Event data is stored in a signed log file that can be verified and exported in XML or comma-separated value format.

Message Digest 5 (MD5)

An algorithm for creating a cryptographic hash (or "fingerprint") of a message or data. MD5 can also refer to the value resulting from the application of the MD5 algorithm. A certificate's MD5 value is unique to that certificate, and is often used to identify a certificate when setting LDAP and Web server ACL rules.

Nickname

A user-friendly character string used to uniquely identify a CA, signer, status source, or certificate recipient.

Nonces

Random numbers used in security protocols to prove that a message is part of a current message exchange.

Non-repudiation

The author of a message cannot deny having created that message at a later date (that is, repudiation cannot occur). Digital signatures help ensure the non-repudiation of transactions.

OCSP

See **Online Certificate Status Protocol**.

OCSP Client

The entity that issues a certificate status request to an OCSP Responder. The OCSP client suspends acceptance of the certificate until the responder returns the certificate status.

OCSP Forwarding

One of two ways Keon VS queries a remote OCSP server. When queries are forwarded, a client request triggers Keon VS to send a second OCSP request to a remote OCSP server, and then use the remote server's response to construct a second response to send to the client. Keon VS can also use OCSP proxying to query a remote OCSP server.

OCSP Performance

The number of OCSP responses per second an OCSP server is capable of processing.

OCSP Proxying

One of two ways Keon VS queries a remote OCSP server. When queries are proxied, Keon VS passes client requests unchanged to the remote OCSP server and returns the remote server's response unchanged to the client. Keon VS can also use OCSP forwarding to query a remote OCSP server.

OCSP Request

A client issues an OCSP request to obtain the status of a certificate. The client suspends acceptance of the certificate until an OCSP response is received.

OCSP Responder

The OCSP Responder is a virtual Web server host that is a component of the Keon CA Web Server. The OCSP Responder accepts certificate status requests from OCSP-enabled clients, looks up a certificate's status, and responds with the certificate's current status.

OCSP Response

Keon VS obtains the status of a certificate and returns an OCSP response to the client who issued the certificate status request.

OCSP Signer

A logical entity that signs OCSP responses.

Online Certificate Status Protocol (OCSP)

OCSP is a protocol, defined in RFC 2560, that enables applications to check the status of a certificate every time the certificate is used. If your PKI is configured to use OCSP, CRLs are unnecessary.

Online Validation

Online validation occurs when a CA can be queried directly about a certificate's validity every time the certificate is used.

Operator Card Set (OCS)

This term is specific to the nCipher security world. It describes a card set within the security world that is used to generate, protect, and access the private keys created within it.

PKCS #7

PKCS #7 is the Cryptographic Message Syntax Standard. For more information, see the standard at:

`http://www.rsasecurity.com/rsalabs/pkcs/pkcs-7/`

PKCS #10

PKCS #10 is the Certification Request Syntax Standard. For more information, see the standard at:

`http://www.rsasecurity.com/rsalabs/pkcs/pkcs-10/`

PKI Performance

The number of CRLs per hour that an OCPS server is capable of importing.

PKIX

Public Key Infrastructure X.509 (PKIX) is the evolving Internet Engineering Task Force (IETF) standard for PKI using X.509 certificates. For more information, see the standard at:

`http://www.ietf.org/html.charters/pkix-charter.html`

Privacy Enhanced Mail (PEM) format

PEM was originally created to provide secure e-mail services on the Internet. It turned out to be too unwieldy for widespread use, and now "PEM format" usually refers to the base64 encoding algorithm that was part of the PEM proposal.

Basically, PEM encoding is useful whenever binary data needs to be presented in a text-readable form; for example, to allow it to be copied and pasted between applications. See **Base64**.

Private Key

The private part of a public-key key pair. With Keon VS, private keys are generated on the client whenever a certificate request is made. Private keys must be securely stored to prevent unauthorized access and accidental deletion.

A digital signature involves encrypting a message digest with a private key and allows anyone with the corresponding public key to decrypt the message digest to be certain of who sent the message and that it has not been tampered with.

Information encrypted with a public key can only be decrypted with the corresponding private key.

Public Key

The public and widely distributed part of a public-key key pair. For example, a certificate contains information about the certificate subject, the certificate's signer, and a public key value. In general, information encrypted with a public key can only be decrypted with the corresponding private key.

Public-Key Cryptography Standards (PKCS)

PKCS is a set of standard protocols developed by RSA Security for making secure information exchange possible. The standards include RSA encryption, password-based encryption, and cryptographic message syntax. For more information, see the standards at:

`http://www.rsasecurity.com/rsalabs/pkcs/`

Public Key Infrastructure (PKI)

A PKI is a system for publishing, distributing, and managing the public key values used in public key cryptography. All PKIs involve issuing public key certificates to individuals, organizations, and other entities and verifying that these certificates are valid.

Refresh Time

The time after which Keon VS will attempt to retrieve a fresh status value. A status value is considered stale after its status source's refresh time elapses.

Response Cache Lifetime

A period of time during which Keon VS can reuse a previously generated and signed response. You can configure Keon VS to reuse responses for a period of time after the response is generated, a period of time prior to the response's Next Update time, or until the Next Update time within the response.

Response Caching

Reusing a previously generated and signed response.

Revocation

Revoking a certificate makes the certificate invalid, effectively removing all of the certificate's privileges in the PKI. Revocation is necessary if the CA administrator wants to invalidate the certificate before it expires. Certificates are revoked by marking them as invalid in the Secure Directory. Users of the PKI are notified of a certificate's revoked status during online validation or with CRLs.

Revoking a CA makes the CA's certificate invalid, effectively removing all the CA's PKI privileges. The action of revoking a CA should be taken based on organizational-based security concerns and only as a last resort.

Rivest-Shamir-Adleman (RSA)

A highly secure cryptography method created by the three founders of RSA Security: Professors Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman.

RSA uses a two-part key. The private key is kept by the owner; the public key is published. Data that is encrypted using the recipient's public key can only be decrypted by the recipient's private key, and vice-versa.

The RSA algorithm is computation intensive. Therefore, it is often used to create a digital envelope, which holds an RSA-encrypted symmetric key (often 3-DES or AES) and symmetric key-encrypted data. This method encrypts the secret symmetric key so that it can be transmitted over the network, but encrypts and decrypts the actual message using the much faster symmetric key algorithm.

The RSA algorithm is also used for authentication using digital signatures. In this case, the sender's private key is used for signing, and the sender's public key is used for verification. The RSA algorithm is also implemented in hardware. As RSA chips get faster, RSA encoding and decoding will add less overhead to the operation.

root CA

A CA whose certificate is self-signed; that is, the issuer and the subject are the same. A root CA is at the top of a hierarchy.

Secure Hash Algorithm (SHA-1)

An algorithm developed by the U.S. National Institute of Standards & Technology (NIST). SHA-1 is used to create a cryptographic hash (or "fingerprint") of a message or data. SHA-1 is considered to be somewhat stronger than MD5. SHA-1 is defined in FIPS Publication 180-2, the Secure Hash Standard (SHS).

Secure Sockets Layer (SSL)

SSL is a protocol layer created by Netscape to manage the security of message transmissions in a network. Security is achieved via encryption. The "sockets" part of the term refers to the sockets method of passing data back and forth between client and server programs in a network or between program layers in the same computer.

Security World

A Security World is made up of at least one hardware module, a set of smart cards, and encrypted data stored on a computer.

Server Certificate

An end-entity certificate issued to a server. Servers present their certificates to Web browsers so browsers can verify (authenticate) the identity of the server. Server certificates are sometimes called SSL certificates.

Signer

See **OCSP Signer**.

Signer Certificate

A certificate, signed by a known CA, that the signer uses to sign status responses. To create a signer certificate, the CA signs a signer certificate request that you send to the CA. The same request can be reused to obtain signer certificates from different CAs.

S/MIME

Secure Multi-Purpose Internet Mail Extensions (S/MIME) is a secure method of sending e-mail. S/MIME is included in the latest versions of e-mail clients from Microsoft and Netscape and has been endorsed by other vendors that make messaging products.

MIME itself, described in the IETF standard RFC 1521, defines how an electronic message is organized. S/MIME allows encryption information and a digital certificate to be included as part of the message body. S/MIME has extended the syntax provided in PKCS #7. For more information, see the standard at:

`http://www.ietf.org/html.charters/smime-charter.html`

SSL Client Authentication

The process whereby a server authenticates a client by verifying the end-entity certificate presented by the client during SSL operations.

SSL-LDAP

LDAP over an SSL connection.

SSL Server Authentication

The process whereby a client application authenticates a server by verifying the certificate chain presented by the server during SSL operations, starting with a CA trusted by the client.

Stale Status Data

Refers to the freshness of a revocation list or status value. A list or status value is considered stale once its status source's refresh time elapses. For example, if the refresh time for Keon VS to retrieve a new list or status value has passed, the list or status value within the Keon VS database is considered stale.

Status

The validity of a certificate: reinstated, revoked, or suspended.

Status Data Caching

Reusing previously obtained status data.

Status Source

A location and method for obtaining the status of certificates.

Suspension

Suspending a certificate marks it as temporarily invalid. The end-user presenting the suspended certificate is denied access where the certificate previously allowed access. Reinstating a certificate returns all removed PKI privileges.

Suspending a CA certificate marks it as temporarily invalid, effectively removing all of the CA's PKI privileges. Reinstating a CA certificate returns all removed PKI privileges.

System CA

The CA created during installation of Keon VS to issue the server certificates.

Synchronization Performance

The number of status values per second that can be updated between two servers.

System Log

An operating system specific file that Keon VS uses to record systemic events not related to regular operations or configuration changes.

Trace Log

A file containing information suitable for debugging purposes.

Transition Layer Security (TLS)

Internet protocol that provides privacy between server and client.

UTF-8 Encoding

An ASCII compatible multibyte Unicode and UCS encoding, used by current browsers, Java and Plan 9.

Validation

The process of verifying that a certificate is valid. Validation can occur online or through the use of CRLs.

Validation Server

A Validation Server is a server that accepts requests from clients to check the validity of certificates. Clients and server can communicate with a number of standard protocol. Keon VS supports the Online Certificate Status Protocol (OCSP).

Validity

Whether a certificate is valid or invalid. A certificate is valid if it has not expired and the information in the certificate is true.

Web Server

An Apache-based server that is the primary interface to Keon VS.

X.509

An International Standards Organization (ISO) standard that describes a basic electronic format for digital certificates.

X.509 v3 Certificate Extension

The RSA Keon product suite supports X.509 v3 certificate extensions, including extensions for PKIX, SET, and SSL. These extensions are certificate attributes conforming to version 3 of the X.509 standard and specify additional constraints or capabilities on the certificate subject.

Acronyms

API	application programming interface
ARL	authority revocation list
ASN.1	Abstract Syntax Notation One
CA	certificate authority
CP	certificate policy
CPS	certification practice statement
CRL	certificate revocation list
DER	Distinguished Encoding Rules
DSA	Digital Signature Algorithm
FQDN	fully qualified domain name
GUI	graphical user interface
HSM	hardware security module
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol (over an SSL connection)
I18N	Internationalization
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IPSec	IP Security Protocol
ISO	International Standards Organization
ITU/CCITT	International Telecommunication Union
KCA	Keon Certificate Authority
KKRM	Keon Key Recovery Module
KRA	Keon Registration Authority
KVS	Keon Validation Server

LDAP	Lightweight Directory Access Protocol
MD5	Message Digest 5
MSIE	Microsoft Internet Explorer
OCSP	Online Certificate Status Protocol
PEM	Privacy Enhanced Mail format
PIN	personal identification number
PKCS	Public-Key Cryptography Standards
PKI	public key infrastructure
PKIX	Public Key Infrastructure (X.509)
RA	registration authority
RAM	random access memory
RSA	Rivest-Shamir-Adleman
S/MIME	Secure Multi-Purpose Internet Mail Extensions
SHA-1	Secure Hash Algorithm
SSL	Secure Sockets Layer
SSL-LDAP	Lightweight Directory Access Protocol over a Secure Sockets Layer connection
TLS	Transport Layer Security
UCS	Universal Character Set (the superset of all other character sets)
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTF-8	UCS Transformation Format
VPN	virtual private network

Index