



# **FORUM SENTRY™ VERSION 9**

## **SECURITY POLICIES AND PKI GUIDE**



### **Legal Marks**

No portion of this document may be reproduced or copied in any form, or by any means – graphic, electronic, or mechanical, including photocopying, taping, recording, or information retrieval system – without expressed permission from Forum Systems, Inc.

FORUMOS™ Firmware, Forum Systems XMLSec™ WebAdmin, Forum Systems XML Security Appliance™, Forum Sentry™, Forum Presidio™, Forum XWall™, Forum Sentry™ Web Services Gateway, Forum Presidio™ OpenPGP Gateway, Forum FIA Gateway™, Forum XWall Type-PCI™, Forum XWall® Web Services Firewall and Forum XRay™ are trademarks and registered trademarks of Forum Systems, Inc.

All other products are trademarks or registered trademarks of their respective companies.

Copyright © 2002-2024 Forum Systems, Inc. – All Rights Reserved.

Forum Sentry™ Version 9 Security Policies and PKI Guide, published July 2024.

D-ASF-SE-030346

## Table of Contents

INTRODUCTION TO THE SECURITY POLICIES AND PKI GUIDE .....	1
Audience for the Security Policies and PKI Guide .....	1
PKI ENGINE AND KEY MANAGEMENT .....	2
KEYS.....	2
Key Pairs and Public Certificates.....	3
Full View and Compact View with Keys .....	3
Sort by Name and Sort by Expiration .....	3
PKCS Key File Format Definitions .....	3
Key Pair Concepts .....	5
Public Key Certificate Concepts .....	6
PKCS Key and X.509 Certificate Formats Supported .....	7
How PKCS 12 Key Pairs are Stored .....	8
Keys Examples .....	11
Generate a PKCS Key Pair .....	12
Generating a Key Pair and Root Certificate .....	12
Using a Generated Self-signed Certificate Valid for n Days Option.....	14
Using Enroll with Registering Authority Option.....	15
Import a PKCS 12 Key Pair and Associate CRL to Signer Group .....	16
Associating a CRL to a Signer Group .....	17
Delete a Key or Signer Group Policy that is Referenced Elsewhere on the System .....	17
Import an X.509 or PKCS 7 Public Certificate as a File Upload.....	18
Import an X.509 or PKCS 7 Public Certificate as an LDAP Request.....	19
Import an X.509 or PKCS 7 Public Certificate through Paste from Clipboard .....	21
SIGNER GROUPS (X509 Path Validation).....	22
Authenticating X509.....	23
The DEFAULT Signer Group.....	24
Signer Groups and Establishing Trust .....	25
CERTIFICATE REVOCATION LISTS.....	26
CRL Fetching .....	27
Default CDP .....	28
CRL Policy Types .....	29
CRL Policies Screen Terms.....	30
Example X.509 CRL File .....	31
Example CDP File .....	32
Supported CRL Formats.....	33
CRL Policy via LDAP .....	34
CRL Policy via File Upload .....	35
CRL Policy via Copy and Paste .....	36
CRL Policy via CDP.....	38
View CRL Details from CRL Policies Screen .....	39
Clear CRL Cache of CRL Policies of Type LDAP or CDP .....	40
Delete a CRL Policy Currently In Use .....	41
SSL POLICIES.....	42
SSL Initiation Policy Screen Terms .....	43
SSL Termination Policy Screen Terms.....	44
Relationships Between SSL Policies and Signer Groups .....	46
How the Appliance Manages the SSL Connection in Termination and Initiation Policies.....	48
SSL Policy Examples.....	49
Add SSL Initiation Policy .....	49
Add SSL Termination Policy.....	50
ENCRYPTION POLICIES .....	51
Authenticate X509 .....	51
Encryption policy Terms .....	52
DECRYPTION POLICIES.....	53

Complimentary Algorithms for Decryption and Encryption Policies .....	54
Decryption Policy Terms .....	55
SIGNATURE POLICIES .....	56
Signature Policy Terms .....	57
VERIFICATION POLICIES .....	58
Verification Policies with a Trusted Pre-stored Peer Certificate .....	59
Verification Policies with a Doc-embedded Certificate Trusted by Signers .....	60
Require KeyUsage nonRepudiation .....	61
Require KeyUsage digitalSignature .....	62
Verification Policy Terms .....	63
APPENDIX .....	64
Appendix A - Constraints in Security Policies and PKI Guide .....	65
Appendix B - Specifications in Security Policies and PKI Guide .....	66
Index .....	67

### **List of Figures**

Figure 1: Signer Group Example How the Appliance Breaks Apart and Stores a PKCS 12 Key Pair Package .....	9
Figure 2: Key Generation .....	12
Figure 3: Certificate Signing Request .....	12
Figure 4: Key Details .....	13
Figure 5: Certificate Enrollment .....	15
Figure 6: Key Import .....	16
Figure 7: PKCS#12 Key Import .....	16
Figure 8: Delete Error Message .....	17
Figure 9: X.509 Or PKCS#7 Key Import .....	19
Figure 10: Public Certificates Verification .....	25
Figure 11: CRL Policy Deletion Error Message .....	41
Figure 12: The SSL Connection in Termination and Initiation Policies .....	48
Figure 13: Encryption policy .....	51

# INTRODUCTION TO THE SECURITY POLICIES AND PKI GUIDE

## Audience for the **Security Policies and PKI Guide**

The *Forum Systems Sentry™ Version 9 Security Policies and PKI Guide* is for System Administrators who will manage:

- Generating or importing PKCS private keys and Certificates.
- Creating Signer Groups for X509 Certificate Authentication
- Creating CRL policies for revocation checking
- Creating SSL Initiation and SSL Termination policies.
- Configuring Encryption or WS-Security Encryption
- Configuring Decryption or WS-Security Decryption
- Configuring Signatures or WS-Security Signatures
- Configuring XML DSIG, or WS-Security DSIG Verification

## Conventions Used

A red asterisk ( \* ) aligned with a field term means that this field is required. In this and other documentation, the Web Administration Interface is referred to as the WebAdmin and the Forum API Security Gateway™ is referred to as the 'device', 'product' or 'system'.

In this document, all data or commands that must be entered or selected are displayed in boldface. Example:

User name:     **johnsmith**  
Password:     **\*\*\*\*\***

UI screens, which display a STATUS column, represent the following states:

- Green status light = enabled policy.
- Yellow status light = a required functional element of this policy is disabled.
- Red status light = disabled policy.

Throughout this and other documents in the Documentation Set, repetitive actions such as:

- View / edit a policy.
- Enable / disable a policy.
- Delete a policy.
- Rename a policy.
- Limit display of policies with Search or Max Results fields.

are not shown. For more information, refer to the Common Operations section of the *Forum Systems Sentry™ Version 9 Web-based Administration Guide*.

Any specifications and constraints referenced in this volume appear in the Appendices of this volume.

## **PKI ENGINE AND KEY MANAGEMENT**

Forum Sentry provides a comprehensive key management policy infrastructure that takes the complexity out of managing keys and certificates. Keys created or imported on Sentry can be used to create security policies ranging from SSL, to Encryption, Decryption, Signatures, and Signature Verification. Security policies can then be migrated or promoted to other Sentry instances and the Key policies will be seamlessly and securely transferred as well. Providing one central location where key policies and the security policies associated with them can be managed

Forum Sentry PKI Engine and Key Management meets the requirements set forth in the following PKI industry standards:

- Department of Defense (DoD) Public Key Infrastructure (PKI)
- X.509 Certificate Policy For The U.S. Federal PKI Common Policy Framework v1.17
- NIST Recommendation for X.509 Path Validation

## **KEYS**

The Keys screen provides a workspace for generating or importing PKCS, OpenPGP and SSH key pairs as well as importing public certificates. The following certificate types are supported: DSA, RSA, and ECC.

## Key Pairs and Public Certificates

The Keys screen serves as a management interface for the repository of keys and certificates that will be used by your back-end server. From the Keys screen, Administrators may:

- Supports creation and import of DSA, RSA, and ECC Keys
- Import X.509 public Certificates
- Import PKCS#1 key pairs
- Import PKCS#7 public Certificates
- Import PKCS#8 key pairs
- Import OpenPGP public keys
- Import OpenPGP private keys
- Import OpenPGP Batch key files
- Import SSH public key
- Import SSH private key
- Generate OpenPGP key pairs
- Generate SSH key pairs
- Import or generate PKCS#12 key pairs (including MS Certificate Server .pfx-formatted keys).
- Import Java Key Store (JKS) files
- View PKCS key details.
- Limit display of Key policies view with the Search or Max Results field.
- Delete a key pair or public certificate unless referenced elsewhere on the system.

## Full View and Compact View with Keys

The KEYS screen may be toggled between Full View and Compact View. Both views are populated with a key name, key type, size and status. The Full view also includes Created / Imported date, Last used date, Expiration date and Email address of key owner.

## Sort by Name and Sort by Expiration

The KEYS screen may also be toggled between Sort by Name or Sort by Expiration views, as the next sections and graphics display. To sort by expiration, click the **Sort by Name** link. To sort by expiration, click the **Sort by Name** link.

## PKCS Key File Format Definitions

The PKCS key file formats supported in the system are defined as:

- **PKCS #1** is an older format for representing a private key that may be encrypted with a passphrase. PKCS#1 may be PEM or DER encoded.
- **PKCS #7** is a format for representing a collection of one or more X.509 Certificates. The public Certificates may include the root CA Certificate. PKCS#7 may be PEM or DER encoded.
- **PKCS #8** is a format for representing a private key that may be encrypted with a passphrase. PKCS#8 may be PEM or DER encoded
- **PKCS #12** is a format for representing a collection of public/private keys pairs with optional signer CA Certificates that may include the root CA Certificate. PKCS#12 keys may be PEM, DER or BER encoded.

Administrators may import, view and delete key pairs or public key Certificates. When entering key pairs, it is expected that the keys you enter will be complementary. When working in the Keys screen, consider:

- PKCS key names must be unique, from 1 to 32 alphanumeric characters, are case sensitive, and may include underscores, dashes and spaces.
- Key types
  - An X.509 public key Certificate. These may be for an individual, an individual organization, an SSL-enabled site, or an individual machine.
  - An X.509 key pair (i.e., a private key and complementary X.509 public key certificate).
- For both HSM-enabled and non-HSM enabled systems, key sizes supported are 512 - 4096.



## Key Pair Concepts

The following concepts are important to understand when working with key pairs:

- Key pairs can be either RSA or DSA.
- When a key pair is referenced by one or more policies, the key pair may not be deleted.
- Once created, key pair aliases cannot be edited. To modify a key pair name, delete the key pair and re-create it with a new name.
- To create a server policy that uses SSL, create the key pair, create an SSL policy that refers to the key pair, then create a Server policy that refers to the SSL policy because:
  - To create a Server policy that uses SSL, an SSL policy must first exist to reference it.
  - To create an SSL policy, a key pair must first exist to reference it.

## Public Key Certificate Concepts

The following concepts are important to understand when working with public Certificates:

- Public key Certificates are used for SSL authentication.
- DSA public keys cannot be used for encryption.
- To modify a Certificate name, delete the Certificate and re-create it.
- A certificate cannot be deleted if there are existing policies referencing it.

## PKCS Key and X.509 Certificate Formats Supported

The system supports the following PKCS key and X.509 Certificate formats:

KEY FORMAT	PEM	DER	BER
PKCS#1	X	X	
PKCS#7	X	X	
PKCS#8	X	X	
PKCS#12 *	X	X	X
X.509 Certificate	X	X	

\* PKCS#12 keys pairs are managed differently than other PKCS keys on the system. For example, the system supports MS Certificate Server .pfx formatted keys. For step-by-step details, review the section entitled Generate a PKCS#12 Key Pair.

## How PKCS 12 Key Pairs are Stored

For a PKCS#12 key pair that contains more than one public Certificate, the system stores one key pair, and additionally stores each included public Certificate. Breaking apart and storing PKCS#12 key pairs in this manner allows a user to reference any individual public Certificate.

This single file is decrypted and MAC-verified by the File Integrity Password that the user specified during the import operation.

PKCS#12 key pairs adhere to the Personal Information Exchange Syntax Standard (PIESS), which specifies a portable format for storing or transporting this particular key pair.

The PKCS#12 key pair, held in one file, may contain many keys and/or many Certificates for various users. The trusted root authority for an imported PKCS#12 public Certificate must be present in the file.

The integrity of the PKCS#12 file is verified with the File Integrity Password. The private key is decrypted with the private key passphrase. PKCS#12 supports privacy and integrity either through passwords or through public-keys. The system supports privacy and integrity only through passwords. The system also supports MS Certificate Server .pfx formatted keys.

When the imported PKCS#12 file populates the KEYS screen, there are several new keys added, as well as a new signer group added to the listing of Signer Groups. In the following example, the imported PKCS#12 file generated:

- A. The PKCS#12 key pair.
- B. The PKCS#12's user's certificate.
- C. The PKCS#12's ROOT certificate.
- D. The PKCS#12's Signer group.

**Note:** You may have more key entries created depending on the number of certificates provided in the PKCS#12 key chain. The following example has a Root CA and an end-entity certificate.

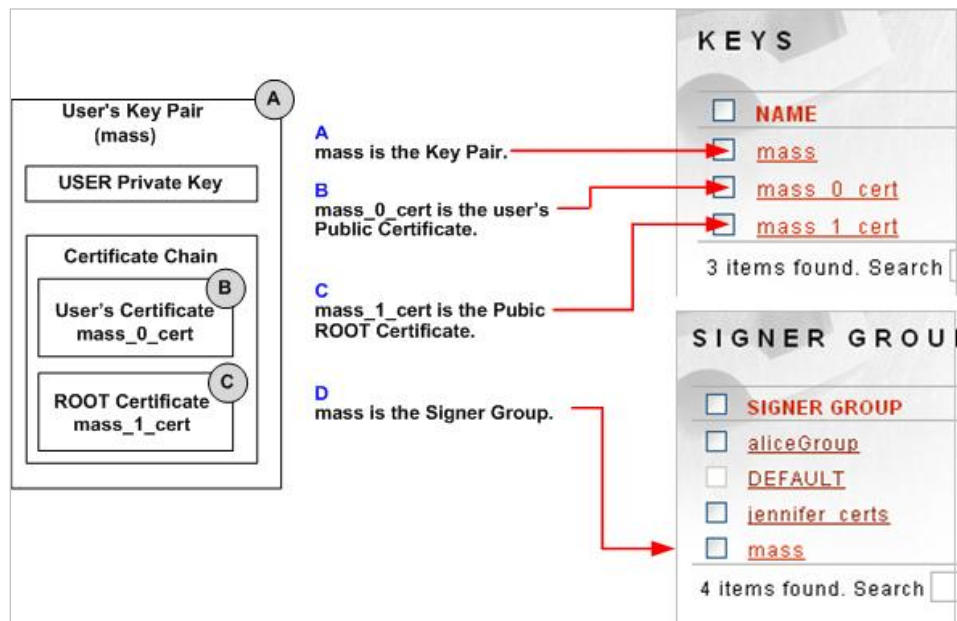


Figure 1: Signer Group Example How the Appliance Breaks Apart and Stores a PKCS 12 Key Pair Package.

## Key Screen Terms

The following table defines various screen terms and definitions for the Keys screen:

TERM	DEFINITION
Name	The name of a key.
Labels	An identifier to group policies of the same type
Type	The type of a key: certificate, key pair, SSH key pair and OpenPGP Key Pair
Size	The size of a key ranging from 1024 to 4096
Status	<div>The status of the key:<ul style="list-style-type: none"><li>• <b>Active:</b> PKCS key that exists on the system and is available for use.</li><li>• <b>Awaiting Certificate Response:</b> Key pair that has been generated on the system, but cannot be used until the corresponding signed CSR request is associated with the private key.</li></ul></div>

## Keys Examples

Examples for keys include:

- Generate a PKCS#12 Key Pair.
- Import a PKCS#12 Key Pair.
- Delete a PKCS#12 Key Pair.
- Delete a Key that is Referenced Elsewhere on the System.
- Import an X.509 / PKCS#7 Public Certificate as a File Upload.
- Import an X.509 / PKCS#7 Public Certificate as an LDAP Request.
- Import an X.509 / PKCS#7 Public Certificate as a Paste from Clipboard.

## Generate a PKCS Key Pair

This example displays generating a Root Certificate PKCS key pair. This operation includes two separate steps:

- Generate a PKCS key pair, which includes a self-signed Root Certificate, required to create a Signer group for this key pair.
- Select the Generated Self-signed Certificate option or the Enroll with **Registering Authority**
- Add the Signer group for this key pair.

## Generating a Key Pair and Root Certificate

The screenshot shows a web interface for 'KEYS > KEY GENERATION'. The main heading is 'GENERATE NEW KEY PAIR'. Below this, there are several input fields and radio button options:

- Name\*:** A text input field with a small icon to its right.
- Labels:** A text input field.
- Algorithm:** Three radio button options: ☒ RSA, ☐ DSA, and ☐ EC.
- Key Size (in bits):** Four radio button options: ☐ 1024, ☒ 2048, ☐ 4096, and ☐ Custom (with a text input field next to it).
- Seed Entry:** A text input field.

A red 'Next' button is located at the bottom right of the form.

Figure 2: Key Generation

The screenshot shows a web interface for 'KEYS > KEY: APRIL\_ROOT > CERTIFICATE SIGNING REQUEST'. The form is divided into several sections:

- IDENTIFYING INFORMATION:** Fields for Common Name\* (april\_root), Organizational Unit(s) (HR), Organization(s) (ABCCCompany), City/Locality (Boston), State/Province (MA), and Country (US: United States).
- SUBJECT ALTERNATIVE NAMES:** Email Address (subject alternative name) field and an 'Include in Subject DN' checkbox.
- KEY USAGE:** Four checkboxes: Client Authentication (checked), Server Authentication (checked), Data Signing (including nonRepudiation) (checked), and Data Encryption (checked).
- REQUEST CERTIFICATE:** Signature Hash Algorithm (SHA-384), and two radio button options: ☐ Enroll with Registering Authority (generate PKCS#10 / CSR) and ☒ Generate certificate valid for 365 days.
- CERTIFICATE GENERATION:** New certificate policy name (april\_root\_cert), Labels field, and three radio button options: ☐ Sign certificate with a local CA key pair, ☒ Self sign certificate, and ☐ Generate certificate signing (CA) certificate.

A red 'Next' button is located at the bottom right of the form.

Figure 3: Certificate Signing Request



KEYS > KEY DETAILS

KEY DETAILS

Name:	april_root
Algorithm:	RSA
Key Size:	2048
Status:	Active
Public Key Details:	<a href="#">Click here</a>
Labels:	

CERTIFICATE

Type:	X.509 version 3
Subject:	C=US, ST=MA, L=Boston, OU=HR, O=ABCCompany, CN=april_root
Issuer:	C=US, ST=MA, L=Boston, OU=HR, O=ABCCompany, CN=april_root
Serial Number (Hex):	7F:89:BC:5D:90:6B:82:E5:2B:D8:DC:28:03:A8:9B:A1:A5:8A:4E:72
Validity:	Nov 3, 2017 11:45:29 AM EDT to Nov 3, 2018 11:45:29 AM EDT
Basic Constraints (OID: 2.5.29.19):	CA: false
Key Usage (OID: 2.5.29.15):	digitalSignature nonRepudiation keyEncipherment dataEncipherment
Extended Key Usages (OID: 2.5.29.37):	2.5.29.37.0 (Key Purpose Anyusage)
Subject Key Identifier (OID: 2.5.29.14) (Hex):	1A3C90E2 E7C685CD 38E1FCDB 997797E8 5BF87F68
Critical Extensions:	2.5.29.15 (Key Usage) 2.5.29.19 (Basic Constraints) 2.5.29.14 (Subject Key Identifier) 2.5.29.37 (Extended Key Usage)
Noncritical Extensions:	2.5.29.14 (Subject Key Identifier) 2.5.29.37 (Extended Key Usage)
SHA1 Hash (Hex):	2FFEA59F 07E8A47A FBD7AF3A CE62DC7D 1C9FBBE8
MD5 Fingerprint (Hex):	4B33 AFDD E25B FEFD 1FBB 8BBF 5639 5A8A
SHA1 Fingerprint (Hex):	F8FC 4D17 3FA2 6155 029B 2EE3 16A3 9F90 2156 9E44
Signature Algorithm:	SHA384withRSA (OID:1.2.840.113549.1.1.12)
View PEM:	<a href="#">Click here</a>
Download:	<a href="#">PEM</a> <a href="#">DER</a>

Re-request Certificate Import Save

Figure 4: Key Details

- Navigate to the **Keys** screen,
- On the KEYS screen, click **New**.
- On the NEW KEY screen, click the **PKCS Key Pair** radio button, and then click **Next**.
- On the KEY GENERATION screen, in the Name field, enter a **name** for this key.
- In the Algorithm section, click the **RSA, DSA or EC** radio button.
- In the Key Size section, click a **radio button** that corresponds to the key size desired, or click **Custom** to add a key size between 512 and 4096.
- In the Seed Entry field, enter some random data or leave blank for the system to generate the random data. Click **Next**.
- On the CERTIFICATE SIGNING REQUEST screen, from the IDENTIFYING INFORMATION section, in the Common Name field, enter a **common name** for this key pair.
- In the Email Address field, enter an **email address**.
- In the Organizational Unit(s) field, enter an **organizational**.
- In the Organizational Name field, enter an **organizational name**.

**Note:** Comma delimited values may be entered in the "Organizational Unit(s)" and "Organizational Name(s)" fields. If comma delimited values are used, the components should be in the order least specific to most specific components (i.e. an entry in the "Organizational Unit(s)" field may be "Product Development, Engineering, Quality Assurance, QA Team 1").

- In the City field, enter a **city**.
- In the State/Province field, enter a **State or Province**.
- In the Country Code field, select a **country**.

At this point, continue with either one of these options:

- Use the **Enroll with Registering Authority** option, or
- Use the **Generated Self-signed Certificate valid for <n> Days** option.

### Using a Generated Self-signed Certificate Valid for n Days Option

Continuing from the previous topic and instructions, follow these steps to complete generating a Key Pair and Root Certificate:

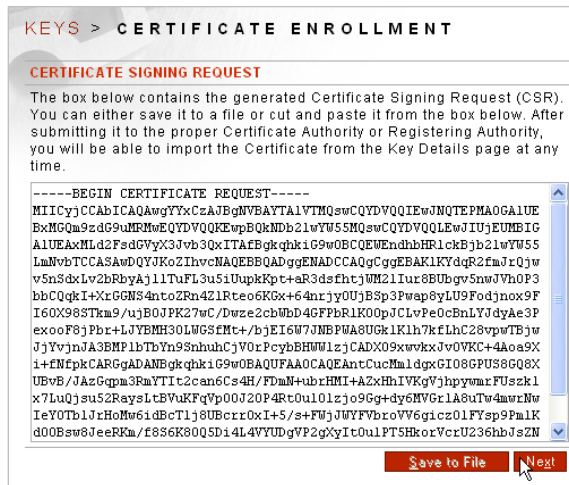
- Select the **Generate Self-signed Certificate valid for <n> days** radio button.
- Enter the **number** of days for this key pair to be valid in the validity field. Click **Next**.
- To view further details for this key pair, on the KEY DETAILS screen click the **Click here** link. The Certificate Contents screen appears.

**Note:** If a certificate is not in the validity period, the part of the validity constraint that is being violated is displayed in red.

## Using Enroll with Registering Authority Option

Continuing from the previous topic and instructions, follow these steps to complete generating a Key Pair and Root Certificate:

- Select the **Enroll with Registering Authority** radio button.
- On the Certificate Signing Request (CSR) screen, either copy the CSR from the text field provided or click **Save to File** to save the CSR to a file.



The screenshot shows a web interface for 'KEYS > CERTIFICATE ENROLLMENT'. Under the heading 'CERTIFICATE SIGNING REQUEST', there is a text box containing a generated CSR. The text starts with '-----BEGIN CERTIFICATE REQUEST-----' and ends with '-----END CERTIFICATE REQUEST-----'. Below the text box are two buttons: 'Save to File' and 'Next!'. A mouse cursor is pointing at the 'Next!' button.

```
-----BEGIN CERTIFICATE REQUEST-----
MIICyCCAbICAQAwgYtYxOzAABgNVBAYTA1VTHQswCQYDVQQIEwJUNQTEPMA0GA1UE
BxMGM9zdg9uMRHwEQYDVQQKEwBQKNDb21wTW55SHQswCQYDVQQLEwJlUjEUMBIG
A1UEAxMLd2FsZGVyX3Jvb3QxLTAtBgkqhkiG9w0BCQEWEndhbHR1ckBjb21wTW55
LmNvbTCCASAwDQYIKoZIhvcNAQEBBQADggENADCCAQgCggEBAK1KYdgR2fmJrQjw
vSnSdxLv2bRbyAj11TuFL3u51UupkRpt+aR3dsfhtjUM21Iur8BUbgv5nwJvH0P3
bbCQqkI+XrGGNS4nto2Rn4Z1Rteo6K0x+64nrjy0UjB8p3Pwap8yLU9Fodjnox9F
I6OX98STkm9/u3B0JPK27wC/Dwze2cbWbD4GFPbR1K00pJCLvPe0cBnLYJdyAe3P
exooF8jPbr+LJYEMH30LWGSfMt+/b3EI6W7JNBFWA8UGk1K1h7kfLhC28vpwTBjw
JjYvjnJA3BMP1bTbYn9SnhuhCjV0rPcybBHWU1zjCADX09xwvxxJvOVKC+4Aoa9X
i+ENfPkCARGgADANBgkqhkiG9w0BAQUFAAQEAntCucMm1dgxGI08GPFUS8GQ8X
UBvB/JAzGqm3RmYTIc2cam6Cs4H/FDmM+ubzHMI+AZxHhIVKgvjhpymarFUSzk1
x7LuQjsu52RaysLtbVuKfQp00J20P4Rt0u101zjo9Gg+dy6MVGrlA8uTw4mwrNw
IeYOTb1JrHoMw61dBcTl1j8UBcrr0xI+5/s+FMjJWYFVbroVVG6icZ01FYsp9FmLK
d00Bsw8JeeRkm/f8S6K80Q5D14L4VYUDgVP2gXyIt0u1PT5HkorVcrU236hbJsZN
-----END CERTIFICATE REQUEST-----
```

Figure 5: Certificate Enrollment

- Click **Next** to save the Key Policy in “awaiting certificate response” status pending receipt of the Certificate from the CA.

## Import a PKCS 12 Key Pair and Associate CRL to Signer Group

This example displays importing a PKCS#12 key pair, which is completed in the following steps:

- Import the PKCS 12 Key Pair.
- Associate the CRL to a Signer Group.

### Importing a PKCS 12 Key Pair



Figure 6: Key Import

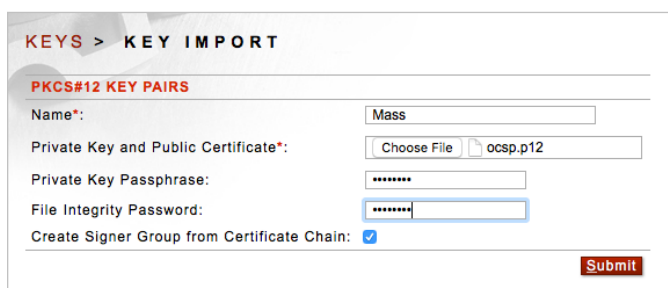


Figure 7: PKCS#12 Key Import

- Navigate to the **Keys** screen.
- On the KEYS screen, click **Import**.
- On the KEY IMPORT screen, click the **PKCS#12 Key Pairs** radio button, and then click **Next**.
- On the PKCS#12 KEY PAIRS screen, in the Name field, enter a **name** for this key.
- Click **Browse** aligned with the Private Key & Public Key Certificate field. The Choose file screen appears. Navigate your file system, locate and click a **PKCS#12 Key Pair**, and then click **Open**.
- Enter your **Private Key Passphrase** in the Private Key Passphrase field.
- Enter your **File Integrity Password** in the File Integrity Password field.
- Check the **Create Signer Group from Certificate Chain** checkbox.
- Click **Submit**.

Continue with the next topic.

## Associating a CRL to a Signer Group

Associating a CRL to a Signer Group will enforce CRL revocation checking of the end-entity, and for all CA certificates in the chain to ensure no parent has revoked its child certificate.

## Delete a Key or Signer Group Policy that is Referenced Elsewhere on the System

When deleting any key pair or public certificate that is referenced elsewhere on the system, the following message appears:



**KEYS**

'CorpSSL' cannot be removed because it is being used by the following: SSL Policy: SSL\_Init\_100, SSL Policy: SSL\_Term100, SSL Policy: SSL\_Policy\_556551636

[Show Full View](#)

<input type="checkbox"/>	NAME	TYPE	SIZE	STATUS
<input checked="" type="checkbox"/>	<a href="#">CorpSSL</a>	Key Pair	1024	Active
<input type="checkbox"/>	<a href="#">CorpSSL_cert</a>	Certificate	1024	Active
<input type="checkbox"/>	<a href="#">larrysmith</a>	Key Pair	1024	Active

17 items found. Search , max results  [Show](#) [Settings](#) [Delete](#) [Import](#) [New](#)

**Figure 8: Delete Error Message**

If you still want to delete the underlying Key, you must first disassociate the current policy from and dependent policies (as listed in the interface), and then the key pair or public certificate will be available for deletion.

## Import an X.509 or PKCS 7 Public Certificate as a File Upload

Importing an X.509 / PKCS#7 public certificate as a file upload requires that you have the public key certificate in your file system.

- Navigate to the **Keys** screen.
- On the KEYS screen, click **Import**.
- On the KEY IMPORT screen, click **X.509 or PKCS#7 Public Certificates** radio button. Click **Next**.
- On the X.509 OR PKCS#7 PUBLIC CERTIFICATES screen, select the **File Upload** radio button aligned with Import Source, and then click **Next**.
- A second X.509 OR PKCS#7 PUBLIC CERTIFICATES screen appears. In the Name field, enter a **name** for this certificate
- Click **Browse** aligned with the Public Key Certificate field. The Choose file screen appears. Navigate your file system and click an **X.509 or PKCS#7 public certificate**, and then click **Open**.
- Skip the Create Signer Group from Certificate Chain checkbox.
- Click **Submit**.

## Import an X.509 or PKCS 7 Public Certificate as an LDAP Request

Importing an X.509 / PKCS#7 public certificate as an LDAP request requires that you have completed the LDAP screen. If a PKCS#7 public certificate that you are importing contains multiple keys, the system will break them apart into separate keys. This will be demonstrated later in the Import a PKCS#12 Key Pair section that contains multiple keys.

**KEYS > KEY IMPORT**

---

**X.509 OR PKCS#7 PUBLIC CERTIFICATES**

Name\*:

Create Signer Group from Certificate Chain: ☒

---

**LDAP IMPORT**

LDAP Server\*:

Port\*:

User:

Password:

---

**SSL**

Use SSL: ☐

Authenticate Server: ☐

---

**LDAP QUERY**

LDAP Query\*:

Attribute\*:

**Submit**

Figure 9: X.509 Or PKCS#7 Key Import

- Navigate to the **Keys** screen.
- On the KEYS screen, click **Import**.
- On the KEY IMPORT screen, click **X.509 or PKCS#7 Public Certificates** radio button. Click **Next**.
- ON the X.509 OR PKCS#7 PUBLIC CERTIFICATES screen, select the **LDAP Request** radio button aligned with Import Source, and then click **Next**.  
A second X.509 OR PKCS#7 PUBLIC CERTIFICATES screen appears.
- In the Name field, enter a **name** for this certificate.
- Accept the default check in the Create Signer Group from Certificate Chain checkbox.
- In the LDAP Server field, enter the **LDAP Server IP**. This IP must match with the LDAP Server IP on the LDAP screen.
- Accept the default LDAP port number (389) in the Port field.
- In the User field, enter an **LDAP user**. This LDAP user must be one, which was brought on to the system after the LDAP screen was configured.
- In the Password field, enter the **password** for this LDAP user.
- To transfer this public certificate via SSL, check the Use SSL **checkbox**. This step is optional.
- To authenticate the LDAP Server, check the Authenticate Server **checkbox**. This step is optional.
- In the LDAP Query field, enter an appropriate **LDAP query**. This step is optional.
- In the Attribute field, accept the pre-populated object class name(s) provided.
- Click **Submit**.



## Import an X.509 or PKCS 7 Public Certificate through Paste from Clipboard

Follow these steps to import an X.509 or PKCS#7 public certificate by pasting from the Clipboard:

- Navigate to the **Keys** screen.
- On the KEYS screen, click **Import**.
- On the KEY IMPORT screen, click the **X.509 or PKCS#7 Public Certificates** radio button. Click **Next**. On the X.509 OR PKCS#7 PUBLIC CERTIFICATES screen, select the **Paste from Clipboard** radio button aligned with Import Source, and then click **Next**. A second X.509 OR PKCS#7 PUBLIC CERTIFICATES screen appears.
- In the Name field, enter a **name** for this certificate.
- Accept the default Create Signer Group from Certificate Chain checkbox.
- Navigate your file system to click and highlight an **X.509** or **PKCS#7** public certificate.
- From the **Edit** menu, select **Select All**.
- From the Edit menu, select **Copy**. Close the file and return to the WebAdmin.
- With your cursor positioned inside the Paste Public Certificate (PEM format) from clipboard text box, press **<Ctrl V>** to paste.
- Click **Submit**.

## **SIGNER GROUPS (X509 Path Validation)**

The Forum Sentry X509 Path validation has been certified by the US Department of Defense.

## Authenticating X509

A Signer group defines one or more trusted chains to achieve the successful authentication of an end-entity X.509 certificate. The signer group defines the trust chain by the certificate DN's for **subject** and **issuer** up to the Root Certificate where **subject=issuer**. Once the X.509 is obtained, Sentry builds an X509 trust chain (issuer->parent subject->...->trusted root) using the Signer Group associated with the transaction.

When importing certificates, depending on the format, Sentry may be able to generate the Signer Group automatically. Regardless of whether it is manually configured, or generated automatically, the selection of CA certificates in the Signer Group will represent 1 or more authentication chains that will be allowed for security policies based on this signer group.

For example, your organization may have 2 sets of CA certificates. You could import both PKCS#7 certificate chains for each CA set and then select those sets of CA certificates in the Signer Group. This has the effect of enabling authentication for any X509 certificates issued by either of these CAs.

To properly validate a certificate, the entire certificate chain has to be built including the Root Certificate and any intermediate certificates.

- Navigate to the **Signer Groups** screen. On the SIGNER GROUPS screen, click **New**.
- In the Signer Group Name field, enter a **Signer Group name** as the name for this Signer Group.
- Under the Include column, check the **checkbox** aligned with any Signer Group root certificate other than the Signer Group root certificate of the Signer Group you are creating.
- In the CERTIFICATE REVOCATION section, in the CRL Policies drop down list, click **DEFAULT**, and then click **Create**.

Continue with the Add an SSL Termination Policy section to apply this key pair when creating an SSL Termination Policy.

## **The DEFAULT Signer Group**

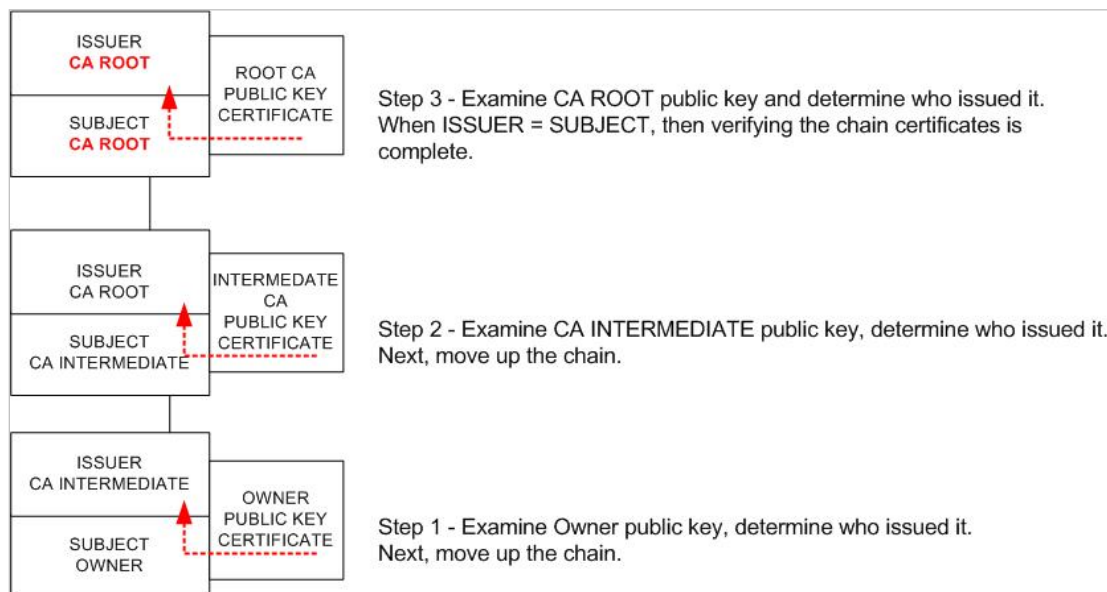
The DEFAULT Signer group is pre-loaded on the system, cannot be deleted, and provides Administrators with self-signed certificates from recognized industry leaders to use when verifying CA certificates. The DEFAULT Signer group cannot be edited or deleted, but only viewed.

## Signer Groups and Establishing Trust

The following concepts are important to understand when working with Signer groups:

- A Signer group must include at least one ROOT Certificate.
- Once created, Signer group aliases cannot be edited. To modify a Signer group name, delete the Signer group and re-create it.
- Signer group names may be from 1 to 32 alphanumeric characters.

The following graphic displays the sequence of steps required when a signed public certificate is verified up a Trusted Certificate Chain:



**Figure 10: Public Certificates Verification**

## CERTIFICATE REVOCATION LISTS

A Certificate Revocation List (CRL) identifies revoked Certificates, it is signed by a Certificate Authority (CA) and it is time-stamped. The CA's signature in a CRL allows CRLs to be distributed by distrusted channels in public directories, just as Certificates are. Each CA issues CRLs on a regular basis.

CRLs are used to help establish the validity of a signed public key Certificate. If a certificate has been revoked because of a compromised key or other reason, an entry is made into the signing authority's CRL.

Importing and updating CRLs in the system can be performed via file import for the Lightweight Directory Access Protocol (LDAP) protocol or CRL Distribution Point (CDP). The CRL Distribution Point (CDP) is an X.509 extension that contains URI references to CRL files on HTTP or LDAP servers. The product currently supports only HTTP CDPs.

Each CRL includes:

- CRL version #
- Issuer - the CA who has issued the CRL.
- Effective date - the date that the CRL becomes effective.
- Next update - the data that the next version of this CRL will be issued.
- Signature algorithm - the algorithm used for the CA's signature.
- Serial numbers - for revoked Certificates.
- Revocation dates - which correspond with each serial number.
- Revocation entry - attributes or values which further detail a Certificate revocation.

Administrators may perform a variety of tasks with CRL policies, including:

- Add a CRL policy of type LDAP.
- Add a CRL policy of type Local File through upload retrieval, HTTP retrieval or through copy and paste retrieval.
- Edit a CRL policy of type CDP.
- View CRL policy details of type Local File from CRL Policies screen or from the actual CRL file.
- Edit a CRL policy.
- Clear the CRL cache of CRL policies of type LDAP or CDP.
- Delete a CRL policy.

## **CRL Fetching**

CRLs can be used on the system by loading the CRL file (that was previously downloaded or retrieved to the Administrator's local system) directly on to the system by retrieving the CRL automatically from an LDAP server, or by dereferencing a distribution point URI from within an X.509.

LDAP and CDP CRL policies will cache the CRL for a specified time period. The first time the CRL is referenced and used, it is cached, and subsequent uses of that CRL will use the cached version until its expiration, at which time a new query will be made and the cache time reset. A Certificate will be considered invalid if its serial number is matched against the serial number in a CRL in the applicable CRL Policy.

## **Default CDP**

The device includes the Default CDP CRL policy, which cannot be deleted. You may not add a CRL policy of type CDP. However, you may edit the cache timeout value in the Default CDP Policy.



## CRL Policy Types

CRL policies include the following types:

- A CRL of type CRL File is displayed by a filename, usually resident in your file system.  
Example: ***verisignpub1.crt***.
- A CRL of type LDAP is the LDAP Server name displayed by its IP address.  
Example: ***11.11.11.10***.
- A CRL of type CDP is an X.509 extension that contains URIs that point to CRL files.

## CRL Policies Screen Terms

FIELD NAME	DEFINITION
<b>CRL Policy</b>	
Policy name	The identifier for this CRL policy.
Type	LDAP is for accessing directory services from an LDAP Server. Local File is for importing an individual CRL file. CDP is for importing an individual CRL file referenced by URIs within an X.509 CDP extension.
<b>CACHE TIMEOUT</b>	
Timeout (in seconds)	In seconds, the maximum length of time a CRL is cached before a new LDAP or CDP query is triggered. May be from 1-6 numeric characters, and must be an integer between 0 and 1,000,000. For example, entering 86400 seconds (1 day) would cache the CRL for 1 day. Entering 604800 seconds (1 week) would cache the CRL 1 week. After the cache expiration, the next time a Certificate CRL check is to be performed, the LDAP or CDP Query will occur. If set to 0, the CRLs will not be cached at all.
<b>LDAP</b>	
LDAP Server	Must be a valid IPv4 address or an RFC 1035-complaint, fully-qualified host name.
LDAP Server Port	The port of the LDAP server. The LDAP Port automatically defaults to port 389. Must be an integer between 1 and 65535.
<b>Local File</b>	
Retrieval Type	<p>The methods of retrieval for a CRL file include:</p> <ul style="list-style-type: none"> <li>• Select <b>Upload</b> to upload the file from a local file system.</li> <li>• Select <b>HTTP</b> to retrieve the file via HTTP.</li> <li>• Select <b>Text</b> to retrieve a text version of the CRL file which may be copied and pasted into the clipboard.</li> </ul>
Upload	Click Browse to search your local file system with Upload or Text retrieval options.
HTTP URL	Enter a URL for the CRL file when selecting the HTTP retrieval option.
Paste PEM from Clipboard	Paste contents of the CRL text file into the Paste PEM from clipboard text block.

## Example X.509 CRL File

The CRL file may be encoded in PEM or DER. The following example displays a Base-64 PEM-encoded X.509 CRL:

```
-----BEGIN X509 CRL-----
LCBJbmMuMSIwIAYDVQQLFBlHbG9iYWwgU2VydmIjZXMgJiBTdXBwb3J0MR0wGwYDVQQDExRSZWQgS
GF0IFRlc3QgUm9vdCBDQTEsMCoGCSqGSib3DQEJARYdc3Ryb25naG9sZC1zdXBwb3J0QHJlZGhhdC
GCSqGSib3DQEBBAUAA4GBAIgeX5VaOkNOKn8MrbxFiqpOrH/M9Vocu9oDeQ6EMTeA5xIWBGN53BZ/
HUJ1Njs32VDG
-----END X509 CRL-----
```

## Example CDP File

An example of a CDP extension taken from an X.509 certificate:

```
ObjectId: 2.5.29.31 Criticality=false
CRLDistributionPoints [
  DistributionPoint:
    [URIName: http://digitalcertificate.citigroup.com/CombinedCRL/crl.crl]
]
```

## Supported CRL Formats

Forum Sentry provides comprehensive support across all CRL distribution and retrieval mechanisms, including FILE, HTTP, LDAP, OCSP, CDP, and XKMS.

## CRL Policy via LDAP

Follow these steps to add a CRL policy of the type LDAP.

- Navigate to the **CRL Policies** screen.
- On the CERTIFICATE REVOCATION LISTS screen, click **New**.
- On the CRL DETAILS screen, in the Policy Name field, enter a unique **CRL Policy name**.
- In the Type field, select the **LDAP** radio button.
- In the Timeout (in seconds) field, enter **86400**, the number of seconds for this LDAP query to cache Certificates.
- In the LDAP Server field, enter the **IP address** for the LDAP Server.
- In the LDAP Port field, enter the default port, **389**.
- Skip the remainder of fields. Click **Create**.

## CRL Policy via File Upload

Follow these steps to add a CRL policy of the type Local File through upload:

- Navigate to the **CRL Policies** screen.
- On the CERTIFICATE REVOCATION LISTS screen, click **New**.
- On the CRL DETAILS screen, in the Policy Name field, enter a unique **CRL Policy name**.
- In the Type field, select the **Local File** radio button.
- Skip the fields in the CCHE TIMEOUT and LDAP sections.
- Aligned with Retrieval Type, select the **Upload** radio button.
- Aligned with the Upload field, click **Browse**. The Choose file screen appears.
- Navigate your file system to locate and click a **CRL file name**.
- Click **Open**.
- Click **Create**.

## CRL Policy via Copy and Paste

Follow these steps to add a CRL policy of the type Local File through copy and paste:

- Navigate to the **CRL Policies screen**.
- On the CERTIFICATE REVOCATION LISTS screen, click **New**.
- On the CRL DETAILS screen, in the Policy Name field, enter a unique **CRL Policy name**.
- In the Type field, select the **Local File** radio button.
- Skip the CACHE TIMEOUT and LDAP sections.
- Aligned with Retrieval Type, select the **Text** radio button.
- Navigate your file system to locate and highlight a **CRL file**.
- Right-click with your mouse to select **Open with**, and then **Notepad**. The CRL file appears in Notepad. From the **Edit** menu, select **Select All**.
- From the Edit menu, select **Copy**. Close the file and return to the WebAdmin.
- With your cursor positioned inside the Paste PEM from clipboard text box, press **<ctrl V>** to paste. Click **Create**.

## ▪ CRL Policy via HTTP

Follow these steps to add a CRL policy of the type Local File through HTTP retrieval:

- Navigate to the **CRL Policies screen**.
- On the CERTIFICATE REVOCATION LISTS screen, click **New**.
- On the CRL DETAILS screen, in the Policy Name field, enter a unique **CRL Policy name**.
- In the Type field, select the **Local File** radio button.
- Skip the CACHE TIMEOUT and LDAP sections.
- Aligned with Retrieval Type, select the **HTTP** radio button.
- In the HTTP URL field, enter or paste an **HTTP URL** for locating this CRL file.
- Click **Create**.



## CRL via URL option

- Navigate to the **CRL Policies screen**.
- On the CERTIFICATE REVOCATION LISTS screen, click **New**.
- On the CRL DETAILS screen, in the Policy Name field, enter a unique **CRL Policy name**.
- In the Type field, select the URL radio button.
- Skip the CACHE TIMEOUT and LDAP sections.
- In the URL field, enter or paste a Distribution **URL** pointing to a CRL
- Click **Create**.

## CRL via XKMS option

XML Key Management Specification (XKMS) is an XML-based protocol that enables you to establish the trustworthiness of a certificate over the Internet. The API Gateway can query an XKMS responder to determine whether a given certificate can be trusted. Services can access an XKMS compliant server in order to receive updated key information for encryption and authentication.

- Ensure you have a remote policy configured.
- Navigate to the **CRL Policies screen**.
- On the CERTIFICATE REVOCATION LISTS screen, click **New**.
- On the CRL DETAILS screen, in the Policy Name field, enter a unique **CRL Policy name**.
- In the Type field, select the **XKMS** radio button.
- Skip the CACHE TIMEOUT and LDAP sections.
- In the Remote Policy field under XKMS, select the remote policy that provides the CRL through XKMS
- **Click Create.**

## CRL Policy via CDP

Follow these steps to edit a CRL policy of the type Certification Revocation List Distribution Point (CDP):

- Navigate to the **CRL Policies screen**.
- Select **Default\_CDP**.
- On the CRL DETAILS screen in the CACHE TIMEOUT section, edit the Timeout (in seconds) **value**, and then click **Create**.

## View CRL Details from CRL Policies Screen

You may view details of a CRL of type CRL in two ways; first from the CRL Policies screen, then actually opening the CRL file for inspection. Follow these steps to view details of a CRL from the CRL Policies screen:

- Navigate to the **CRL Policies** screen.
- Select a **CRL Policy name** link.
- On the CRL DETAILS screen, click **View**.
- On the CRL FILE CONTENTS screen, click **Click Here** to view any further details on the screen.
- In this example, a PEM formatted CRL is visible. Click **X** to close the screen.

**Note:** When editing a CRL of type **CRL**, the CRL file name is not revealed in the CRL file field.

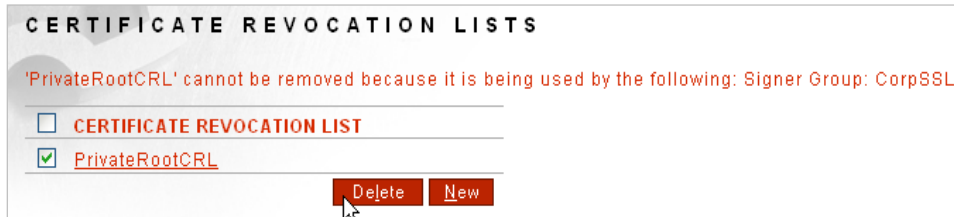
## Clear CRL Cache of CRL Policies of Type LDAP or CDP

Users may clear the cache for individual CRL policies of the type LDAP and CDP.

- Navigate to the **CRLs screen**.
- Select a **CRL Policy name** link of a CRL policy of the type LDAP or CDP.
- On the CRL DETAILS screen, select **Clear Cache**. The screen refreshes with the "CRL cache has been cleared" message visible at the top of the screen.

## Delete a CRL Policy Currently In Use

If trying to delete a CRL policy that is currently in use by the system, the following message appears at the top of the CRL Policies screen:



The screenshot shows a web interface titled "CERTIFICATE REVOCATION LISTS". At the top, a red error message states: "'PrivateRootCRL' cannot be removed because it is being used by the following: Signer Group: CorpSSL". Below this is a table with two columns: a checkbox and a text field. The first row has an unchecked checkbox and the text "CERTIFICATE REVOCATION LIST". The second row has a checked checkbox and the text "PrivateRootCRL". At the bottom right of the table are two buttons: "Delete" and "New". A mouse cursor is pointing at the "Delete" button.

<input type="checkbox"/>	CERTIFICATE REVOCATION LIST
<input checked="" type="checkbox"/>	PrivateRootCRL

Delete New

**Figure 11: CRL Policy Deletion Error Message**

Edit the listed policy (policies), disassociate the CRL from the policy (policies), and then the CRL will be available for deletion.

## SSL POLICIES

The SSL Policies screen manages SSL policies for initiating and terminating SSL/TSL. Forum Sentry hardware devices provide patented cryptographic acceleration of SSL traffic to provide enterprise-scalable SSL offloading and optimized transaction performance.

SSL policies can be enabled for 1-way or 2-way SSL. For 2 way SSL, clients can perform X.509 SSL authentication to the system when Sentry is terminating the SSL connection, and when Sentry is initiating the SSL connection, can present a certificate as the client to authenticate against the server.

Any Network Policy on Sentry can be secured using an SSL policy and you can define any number of different SSL policies, or share the same SSL policy across different Network Policies.

## SSL Initiation Policy Screen Terms

When working with SSL initiation policies, consider the following:

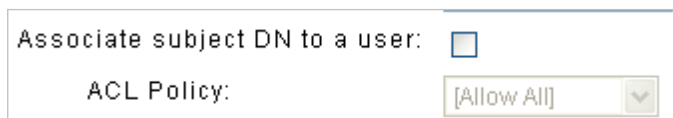
TERM	DEFINITION
Name	The identifier for this SSL Initiation policy.
Authenticate Appliance to Remote Server using Key Pair	The key pair name.
Authenticate the Remote Server using Signer Group	The Signer group used when authenticating the Certificate. When selecting <b>Do not Authenticate</b> , then no Remote Server authentication is performed.
Ignore Server Hostname Verification	Option to ignore the name on the public Certificate being presented by the server.
<b>PROTOCOL</b>	
TLSv1.2	When checked, allow use of the selected protocol.
TLSv1.1	
TLSv1	
SSLv3	
<b>CIPHER SUITE</b>	
	List of supported cipher suites.

## SSL Termination Policy Screen Terms

When working with SSL Termination policies, consider the following:

TERM	DEFINITION
Name	The identifier for this SSL Termination policy.
Key Pair	The key pair name.
Authenticate the Client	<ul style="list-style-type: none"><li>When checked, use the client's Signer Group to authenticate the client.</li><li>When unchecked, do not use the client's Signer Group to authenticate the client.</li></ul>
Signer Group	Identifier for the Signer Group.
Associate subject DN to a user	<ul style="list-style-type: none"><li>When checked, associate the subject DN to a known user.</li><li>When unchecked, do not associate the subject to a known user.</li></ul>
ACL Policy	Identifier of an ACL Policy. The options available are:

**No access control is active.** The user is identified from the protocol but is not matched to any known user in Forum or in any third party user store. For example, if the user provided an X.509 certificate, Forum may verify that the certificate is valid and identify the subject of the certificate as the user, but Forum does not in any way restrict the set of allowed users.



Associate subject DN to a user: ☐

ACL Policy: [Allow All] ▼

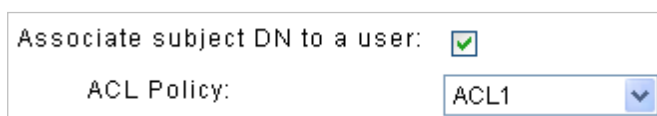
**No Forum ACL is active.** The user is identified from the protocol and is matched to a known user in Forum or in a third party user store. Any Forum ACL does not restrict the user. When selecting **Allow All**, then no user authorization is performed.



Associate subject DN to a user: ☒

ACL Policy: [Allow All] ▼

**Forum ACL is active.** The user is identified, matched to a known user, and restricted by the selected Forum ACL. The ACL is also used to restrict the Groups used for authentication. When using an external Identity Server, only Groups with access to the ACL are checked for authentication.



Associate subject DN to a user: ☒

ACL Policy: ACL1 ▼



TERM	DEFINITION
<b>PROTOCOL</b>	
TLSv1.2	List of supported protocols
TLSv1.1	
TLSv1	
SSLv3	
SSLv2Hello	
<b>CIPHER SUITE</b>	
	List of supported cipher suites.

## **Relationships Between SSL Policies and Signer Groups**

Because Signer groups are referenced in SSL policies, Signer groups must be created before the SSL policies that reference them.

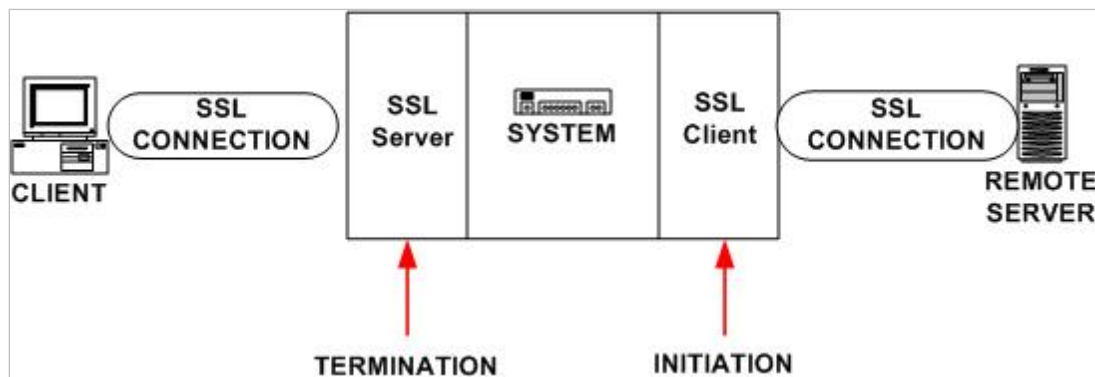
## SSL Caveats

The following points should be considered when dealing with SSL:

- Servers always manage SSL sessions. ***Clients never manage SSL sessions.***
- With SSL Termination policies, the system acts as the server and the SSL session is terminated at the system.
- With SSL Initiation policies, the system acts as the client for a remote server and the SSL session is initiated at the system.
- You will probably need one SSL Initiation policy per SSL back end or remote server.
- You will need one SSL Termination policy per client you want to authenticate the policy for Signer Group if you are doing client authentication.
- You may create one generic SSL Termination policy for all clients whom you will not authenticate, but for whom you will provide a secure SSL-enabled session.

## How the Appliance Manages the SSL Connection in Termination and Initiation Policies

The following graphic displays the boundaries of SSL Termination and SSL Initiation policies:



**Figure 12: The SSL Connection in Termination and Initiation Policies.**

### Terminate an SSL Connection with the Appliance

SSL Termination policies use a key pair to authenticate themselves to the client. They use the Certificate passed to them by the client to authenticate the client. The client certificate is verified using the Signer Group Policy.

## SSL Policy Examples

Examples for SSL policies include:

- Add an SSL Initiation Policy.
- Add an SSL Termination Policy.

### Add SSL Initiation Policy

With an SSL Initiation policy, you define the settings for enforcing security between the system (acting as the client) and a remote server (managing the SSL session). You will now create an SSL Initiation policy.

USE CASE AND COMMENTS	INITIATION POLICY				
	Authenticate system for remote server	Key pair name	Authenticate remote server for system	Reference Signer group	Ignore server hostname verification
The system authenticates itself to the remote server, presents its public key, and verifies the identity of the remote server by using the remote server's Signer group, and verifies the remote server hostname.	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	

- Navigate to the **SSL Policies** screen.
- On the SSL POLICIES screen, click **New**.
- On the NEW SSL POLICY screen, select the **Initiation** radio button, and then click **Next**.
- On the SSL INITIATION POLICY screen, overwrite this name and enter a **unique SSL Policy Name** in the SSL Policy Name field.
- To allow a Remote Server to authenticate the system, select a **Key Pair** from the Authenticate the Appliance to Remote Server using Key Pair drop down list.
- To authenticate the system with a Signer Group, select a **Signer Group** from the Authenticate the Remote Server using Signer Group drop down list (the Signer Group used to validate the server's public Certificate).
- Skip the Ignore Server Hostname Verification checkbox.
- From the PROTOCOL section, check the **desired protocols**
- From the CIPHER SUITE section, check all the **checkboxes** prefacing the cipher suites that may be applied during the SSL handshake. Click **Create**.

## Add SSL Termination Policy

With an SSL Termination policy, you define the settings for enforcing security between the system (acting as the server) and the client. You will now create an SSL Termination policy.

USE CASE AND COMMENTS	TERMINATION POLICY		
	Key pair name	Authenticate client	Reference Signer group
The client is not authenticated; therefore, no Signer group for the client is referenced.	<b>X</b>		<b>X</b>

- Navigate to the **SSL Policies** screen.
- On the SSL POLICIES screen, click **New**.
- On the NEW SSL POLICY screen, select the **Termination** radio button, and then click **Next**.
- On the SSL TERMINATION POLICY screen, overwrite this name and enter a **unique SSL Policy Name** in the SSL Policy Name field.
- From the SSL TERMINATION section of the screen, select a **Key Pair** from the Key Pair drop down list.
- Skip the Authenticate the Client checkbox.
- Skip the Signer Group drop down list.
- Skip the Associate subject DN to a user checkbox.
- Skip the ACL Policy drop down list.
- From the PROTOCOL section, check the **TLSv1 checkbox**, **SSLv3 checkbox**, or **both**.
- From the CIPHER SUITE section, check all the **checkboxes** prefacing the cipher suites that may be applied during the SSL handshake. Click **Create**.

## ENCRYPTION POLICIES

The Encryption policies are provided to abstract the underlying keying material and algorithms associated with the encryption from the actual task policies that will perform the encryption. This allows any number of XML or WS-Security Encryption tasks to be derived from the base Encryption policy without having to manage keys with each one. Further, by abstracting this into an Encryption policy, if any key changes need to occur, this can simply be performed on the Encryption policy and all dependent policies will automatically get updated.

The following elements make up an Encryption policy:

- An Encryption policy name.
- An algorithm to be used for encryption, 3DES with RSA for key protection, AES-128, AES-192 or AES-256.
- A certificate name used to reference the certificate.
- A Signer Group to validate against.

During the creation process, when selecting the Certificate, you are selecting an algorithm to the Encryption policy that you are creating.

Filter any Encryption policies with the [Search](#) or [max results](#) fields in the same manner that you filtered keys. When clicking New from the Encryption screen, a randomly numbered Encryption policy name is displayed; but you may overwrite this name to any unique policy name of your choice. An Encryption policy created in the product allows Administrators to:

- Encrypt all the content of a document.
- Encrypt a single element of a document as well as all the content of a document.
- Use XML, WS-Security 1.1, and WSS-2004 Encryption Specifications
- Use Symmetric

### Authenticate X509

The **Validate against Signer Group** option in the ENCRYPTION POLICY screen allows for an X.509 path validation against the defined Signer Group, which can validate the certificate for encryption or revocation prior to using it for encryption.

ENCRYPT XML > XML ENCRYPTION POLICY

**XML ENCRYPTION POLICY**

Policy Name: ENC\_Jack

Algorithm: AES-256

Encryption Mode:

- ☐ Use the same certificate used for client signature verification
- ☒ Use this pre-stored peer certificate

Jack\_cert

Validate against Signer Group: jack\_group

Save

Figure 13: Encryption policy

## Encryption policy Terms

The following table describes each term and definition for the categories displayed above.

TERM	DEFINITION
Policy Name	The name identifier for this policy
Algorithm	The algorithm to be used for all task policies that derive from this Encryption policy. Options include 3DES, AES-128, AES-192, AES-256, AES-128-GCM, AES-192-GCM, and AES-256-GCM.
Key Wrap Algorithm	The algorithm used for the key wrap. Options include 3DES, RSA, AES-128, AES-192, AES-256 and RSA-OAEP.
Encryption Mode	<p>The location from where to obtain the X509 certificate to use for the derived encryption tasks. Options include:</p> <ul style="list-style-type: none"><li>• <b>Use the certificate from an identified user</b> This option will retrieve the X509 dynamically from the inbound processing (via SSL, SOAP Header, LDAP, etc.)</li><li>• <b>Use the same certificate used for client signature verification</b> This option will use the certificate that was selected in a prior task for this transaction for signature verification.</li><li>• <b>Use this pre-stored peer certificate</b> This option will retrieve the X509 from an existing Key policy defined on Sentry.</li><li>• <b>Use a certificate stored in the specified attribute</b> This option will retrieve the X509 from the specified session attribute</li><li>• <b>Use a symmetric key</b> Use the configured symmetric key</li></ul>
Validate Against Signer Group	Determines whether to authenticate the X509 certificate being used for encryption against a Signer Group.
Symmetric Key:	This option is only used when performing Symmetric encryption using a shared key, rather than using the recommended more secure PKI asymmetric cryptography



## DECRYPTION POLICIES

The Decryption policies are provided to abstract the underlying keying material and algorithms associated with the decryption from the actual task policies that will perform the encryption. This allows any number of XML or WS-Security Decryption tasks to be derived from the base Decryption policy without having to manage keys directly with each one. Further, by abstracting this into an Decryption policy, if any key changes need to occur, this can simply be performed on the Decryption policy and all dependent policies will automatically get updated.

When working with Decryption policies, first create a key pair. Next, create the Decryption policy that refers to this key pair.

The following elements make up an Decryption policy:

- An Decryption policy name.
- An algorithm to be used for decryption: 3DES with RSA for key protection, AES-128, AES-192, AES-256, AES-128-GCM, AES-192-GCM or AES-256-GCM. Administrators may also allow any algorithm specified by the incoming document by selecting the Any option.
- A Key wrap algorithm: AnyRSA, 3DES, AES-128, AES-192, AES-256, RSA or RSA-OAEP
- A key pair name used to reference the key pair.
- A symmetric key

## Complimentary Algorithms for Decryption and Encryption Policies

Although the system allows you to select a preferred encryption algorithm when creating an Encryption policy, you must match the same algorithm when decrypting or choose Any. The following table displays the algorithms that are complimentary for Encryption and Decryption tasks later on:

ALGORITHM IN ENCRYPTION POLICY	ALGORITHM IN DECRYPTION POLICY
3DES	Any or 3DES
AES-128	Any or AES-128
AES-192	Any or AES-192
AES-256	Any or AES-256
AES-128-GCM	Any or AES-128-GCM
AES-192-GCM	Any or AES-192-GCM
AES-256-GCM	Any or AES-256-GCM
RSA	RSA
RSA-OAEP	RSA-OAEP

During the creation process, when selecting the key pair, you are selecting an algorithm to the Decryption policy that you are creating.

When clicking **New** from the Decryption screen, the product displays a randomly numbered Decryption policy name for you, but you may overwrite this name to any unique Decryption policy name of your choice. An Decryption policy allows Administrators to decrypt any content encrypted with the public key, which corresponds to this Decryption policy.

## Decryption Policy Terms

The following table describes each term and definition for the categories displayed above.

TERM	DEFINITION
Policy Name	The name identifier for this policy
Algorithm	<p>The algorithm to be used for the task policies that derive from this Decryption policy. Options include [ANY], 3DES, AES-128, AES-192, AES-256, AES-128-GCM, AES-192-GCM or AES-256-GCM.</p> <p>[ANY] is the recommended setting for maximum interoperability where Forum Sentry will use the algorithm specified in the encrypted message automatically to decrypt.</p>
Key Wrap Algorithm	<p>The algorithm used for the key wrap. Options include [ANY-RSA], 3DES, AES-128, AES-192, AES-256, AES-128-GCM, AES-192-GCM, AES-256-GCM, RSA and RSA-OAEP.</p> <p>[ANY-RSA] is the recommended setting for maximum interoperability where Forum Sentry will use the key wrap algorithm specified in the encrypted message automatically during decryption.</p>
Key Pair	The Key Policy on Sentry to use for performing the asymmetric decryption.
Symmetric Key:	This option is only used when performing Symmetric decryption using a shared key, rather than using the recommended more secure PKI asymmetric decryption.

## SIGNATURE POLICIES

The Signature policies are provided to abstract the underlying keying material and algorithms associated with the digital signature from the actual task policies that will perform the signature in the message format (XML, SOAP, etc.). This allows any number of XML or WS-Security Signature tasks to be derived from the base Signature policy without having to manage keys with each one. Further, by abstracting this into an Signature policy, if any key changes need to occur, this can simply be performed on the Signature policy and all dependent policies will automatically get updated.

The Signature screen manages XML signing policies. From this screen, Administrators may create, edit and delete Signature policies. Signatures that are created in the product allow Administrators to apply a digital signature to a document.

When working with Signature policies, first create a key pair. Next, create the Signature policy that refers to this key pair.

The following elements make up an Signature policy:

- Signature policy name.
- Key pair alias (for signing), or Certificate (for verifying a signature).

Algorithms supported for Signature policies are:

- DSA
- RSA
- ECDSA
- ECDSA SHA256
- ECDSA SHA384
- ECDSA SHA512
- HMAC-SHA1
- HMAC-SHA256
- HMAC-SHA384
- HMAC-SHA512
- RSA-SHA256
- RSA-SHA384
- RSA-SHA512

Digest algorithms supported for Signature policies are:

- RIPEMD160
- SHA-1
- SHA-224
- SHA-256
- SHA-384
- SHA-512

## Signature Policy Terms

The following table describes each term and definition for the categories displayed above.

TERM	DEFINITION
Policy Name	The name identifier for this policy
Signature Algorithm	The signature algorithm
Digest Algorithm	The algorithm used to compute the target data digest
Key Pair	The Key Policy on Sentry to use for performing the digital signature
Symmetric Key	This option is only used when performing Symmetric signature using a shared key, rather than using the recommended more secure PKI asymmetric cryptography

## VERIFICATION POLICIES

The Verification policies are provided to abstract the underlying keying material and algorithms associated with the verification engine from the actual task policies that will perform the verification on the message format (XML, SOAP, etc.). This allows any number of XML or WS-Security Signature Verification tasks to be derived from the base Verification policy without having to manage keys with each one. Further, by abstracting this into an Verification policy, if any key changes need to occur, this can simply be performed on the Verification policy and all dependent policies will automatically get updated.

When working with Verification policies, first create a Signer Group or a trusted peer Certificate. Next, create the Verification policy that refers to the signer Group or the Certificate.

The following elements make up an Verification policy:

- An Verification policy name.
- A verification algorithm: DSA, ECDSA, EDDSA-SHA256, EDDSA-SHA384, EDDSA-SHA512, HMAC-SHA1, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, RSA, RSA-SHA256, RSA-SHA384, or RSA-SHA512. Administrators may also allow any algorithm specified by the incoming document by selecting the [Any] option.
- A digest algorithm to be used for decryption: RIPEMD160, SHA-1, SHA-224, SHA-256, SHA-384 or SHA-512. Administrators may also allow any algorithm specified by the incoming document by selecting the [Any] option.
- A verification mode.

The distinct verification modes include:

- Trusted pre-stored peer Certificate
- Doc-embedded certificate trusted by signers
  - Require KeyUsage nonRepudiation.
  - Require KeyUsage digitalSignature

## **Verification Policies with a Trusted Pre-stored Peer Certificate**

If a trusted Certificate is already in the Keys screen, when selecting the Use a trusted pre-stored peer certificate option, the signature processed by this policy will be verified with this public key certificate.

## **Verification Policies with a Doc-embedded Certificate Trusted by Signers**

When selecting the Use a doc-embedded certificate trusted by signers in the following Signer group option, Administrators are able to establish TRUST by using the Signer group for the doc-embedded Certificate.



## **Require KeyUsage nonRepudiation**

Require KeyUsage nonRepudiation means that the X.509 certificate in the document must contain a KeyUsage extension with the nonRepudiation bit set. The nonRepudiation bit is appropriate when the signature will persist for later retrieval. The nonRepudiation bit is set in the X.509 certificate by the certificate authority that issued the certificate in order to authorize the certificate to verify digital signatures that may be stored for later non-repudiation.

For example, if a signed document will be verified and archived for later non-repudiation, it might be necessary to require that the certificate used for signature verification be legally and contractually authorized for the purpose of non-repudiation as indicated by the nonRepudiation bit.

## **Require KeyUsage digitalSignature**

Require KeyUsage digitalSignature” means that the X.509 certificate in the document must contain a KeyUsage extension with the digitalSignature bit set. The digitalSignature bit is appropriate when the signature is used for one-time authentication of the document signer. The digitalSignature bit is set in the X.509 certificate by the certificate authority that issued the certificate in order to authorize the certificate to verify digital signatures that may be used for user authentication during a specific transaction.

For example, if a signed document will be verified to establish the identity of the user who authored the document, it might be necessary to require that the certificate used for signature verification be legally and contractually authorized for the purpose of user authentication as indicated by the digitalSignature bit.

## Verification Policy Terms

The following table describes each term and definition for the categories displayed above.

TERM	DEFINITION
Policy Name	The name identifier for this policy
Symmetric Algorithm	The list of allowed symmetric algorithms
Asymmetric Algorithms	The list of allowed asymmetric algorithms.
Digest Algorithm	<p>The algorithm used to compute the target data digest</p> <p>[ANY] is the recommended setting for maximum interoperability where Forum Sentry will use the algorithm specified in the signed message automatically to verify.</p>
Verification Mode	<p>Determines where to obtain the X509 certificate to verify the target signature.</p> <ul style="list-style-type: none"><li>▪ <b>Use a doc-embedded certificate trusted by signers in the Signer Group</b> This option will retrieve the X509 dynamically from the inbound message and authenticate this X509 against the selected Signer Group prior to using the certificate to verify the signature.</li></ul> <p><b>Require KeyUsage nonRepudiation</b> Require KeyUsage nonRepudiation means that the X.509 certificate in the document must contain a KeyUsage extension with the nonRepudiation bit set.</p> <p><b>Require KeyUsage digitalSignature option</b> Require KeyUsage digitalSignature" means that the X.509 certificate in the document must contain a KeyUsage extension with the digitalSignature bit set</p> <ul style="list-style-type: none"><li>▪ <b>Use this pre-stored peer certificate</b> This option will retrieve the X509 from an existing Key Policy defined on Sentry.</li><li>▪ <b>Use a symmetric key</b></li></ul>

## APPENDIX

## Appendix A - Constraints in Security Policies and PKI Guide

ELEMENT	CONSTRAINTS	CHAR COUNT
PKCS Key Name	Unique, case sensitive, and accepts the '@' character, underscores, dashes and spaces.	1-32
PKCS Private Key Passphrase	Unique and accepts underscores, dashes and spaces.	4-8191
PKCS#12 File Integrity Password	Unique and accepts underscores, dashes and spaces	4-8191
Seed Entry	Unique and accepts underscores, dashes and spaces.	0-255
Signer Group Name	Unique and accepts underscores and dashes.	1-32
CRL Policy Name	Unique and accepts underscores and dashes but not spaces.	1-32
LDAP Server	Must be a valid IPv4 address or an RFC 1035-complaint, fully-qualified host name.	1-255
LDAP Port	Must be an integer between 1 and 65535.	1-5
Timeout (in Seconds)	May be from 1-6 numeric characters, and must be an integer between 0 and 1,000,000. For reference, 86400 sec = 1 day; and 604800 = 1 week.  If set to 0, the CRLs will not be cached at all.	1-6
SSL Policy Name	Unique, and accepts underscores and dashes.	1-32
Encryption policy Decryption policy Signature policy Verification policy	Unique. May include the '@' character, underscores and dashes.	1-32

**Note:** Encryption, Decryption, XML Signing and Verification policy names may be from 1 to 32 alphanumeric characters, may include the '@' character, underscores and dashes.

However, **Forum Systems recommends** that policy names should be between 5 and 32 characters. These names should not include a space, and not include any of the following characters:

~ ! # \$ % ^ & \* ( ) + { } [ ] " < > ? / \

Even though the system actually allows shorter and longer names, it is not a good practice to use names that are too short or too long.

## Appendix B - Specifications in Security Policies and PKI Guide

ELEMENT SUPPORTED	SPECIFICATIONS
PKCS Keys	The maximum number of keys supported by the system is 1000.
PKCS Key Size	<p>For both HSM-enabled and non-HSM enabled systems, key sizes supported are 1024 – 4096bits.</p> <p>The custom field holds any key size (multiple of 8) between 1024 and 4096bits on HSM-enabled and non-HSM enabled systems.</p>
CRL Policies	Unlimited *
SSL Policies	Unlimited *
Encryption policy Decryption policy Signature policy Verification policy	Unlimited *

\* Limited only by disk space.

## Index

3	
3DES	
algorithm for Decryption .....	54
algorithm for Encryption .....	52
4	
4096 bit keys.....	5
A	
add a CRL policy of type LDAP .....	35
add a CRL policy of type Local File through copy and paste .....	37
add a CRL policy of type Local File through upload .....	36
add a CRL policy of type Local File with HTTP.....	37
add an SSL Initiation Policy .....	50
add an SSL Termination Policy .....	51
algorithm	
3DES with Decryption policy.....	55
3DES with Encryption policy .....	52
AES-192 with Decryption policy.....	55
Alias .....	5
C	
clear CRL cache for policies of type LDAP and CDP .....	41
compact view	
with PKCS keys .....	4
conventions used .....	2
CRL policy	
adding of type Local File with HTTP .....	37
CDP .....	31
components .....	27
CRL fetching .....	28
CRL local file .....	31
CRL Policy Name .....	31
Default CDP.....	29
default for LDAP Server Port.....	31
editing type CDP .....	39
example CDP file .....	33
example X.509 CRL file.....	32
HTTP retrieval type .....	31
LDAP Cache Timeout Seconds .....	31
LDAP Server .....	31
paste PEM from clipboard option.....	31
retrieval type .....	31
text retrieval type .....	31
upload retrieval type .....	31
D	
Decryption	
complimentary algorithm for Encryption .....	55
DEFAULT Signer group .....	25
delete a CRL policy currently in use.....	42
delete PKCS key pair or cert referenced elsewhere .....	18
digest algorithm	
RIPEMD160 with Signature policy .....	57
Doc-embedded Certificate with Verification Policy.....	61
E	
edit a CRL policy of type CDP .....	39
F	
full view	
with PKCS keys .....	4
G	
generate Root Certificate PKCS key pair.....	13
H	
HTTP CDPs	
supports.....	27
HTTP policy screen terms .....	53, 56, 58, 64
I	
import an X.509 public certificate as a File Upload .....	19
import an X.509 public certificate as LDAP request .....	20
import an X.509 public certificate through Paste from Clipboard.....	22
import PKCS#12 key pair .....	17
initiating an SSL handshake .....	50
K	
Key Type	
Certificate or Key Pair .....	5
Keys screen .....	3
Keys screen terms and definitions.....	11
KeyUsage digitalSignature	
in Verification Policy .....	63
KeyUsage nonRepudiation	
in Verification Policy .....	62
L	
LDAP support for CRLs .....	27
M	
MS Certificate Server .pfx formatted keys supported .....	9
multiple keys in PKCS#7 public certificate .....	20

## *N*

name	
of key .....	11

## *P*

Personal Information Exchange Syntax Standard.....	9
PIESS .....	9
PKCS key definitions .....	4
PKCS key formats supported .....	8
PKCS keys	
examples.....	12
key size supported on HSM-enabled systems	
(1504/1504G.....	5
PKCS#12 Key Pairs	
importing .....	8
PKCS#12 Keys	
storing.....	9

## *R*

RIPEMD160	
algorithm for Verification .....	59

## *S*

Signer Groups with SSL .....	47
size	
of key .....	11
sort by name	
with PKI keys.....	4
SSL	
screen terms .....	44, 45

SSL policies	
examples.....	50
SSL Policy	
sequence for creating with Signer Groups .....	47
SSL Screen .....	43
SSLv3 protocol for SSL .....	44, 46
status	
of key .....	11
status of HTTP/S Network policy .....	53, 56, 58, 64

## *T*

terminating an SSL handshake.....	51
TLSv1 protocol for SSL .....	44, 46
Trusted Certificate Chain .....	26
Trusted Pre-stored Certificate	
with Verification Policy .....	60
type	
of key .....	11

## *V*

view CRL details from CRL Policies screen .....	40
---	----

## *X*

Decryption policy	
sequence for creating .....	54
Decryption screen.....	54
Signature policy	
sequence for creating .....	57
Signature screen .....	57