Chapter 9

Configuring Digital Certificates

This chapter describes how to configure digital certificates; it contains the following sections:

Overview

You can use digital certificates in place of preshared keys for IKE negotiations. For more information about IKE, see *IKE Overview* in *Chapter 6, Configuring IPSec*.

Digital Certificate Terms and Acronyms

Table 15 describes terms and abbreviations that are used in this discussion of digital certificates.

Table 15: Digital Certificate Terms and Acronyms

Term or Abbreviation	Description
3DES	Triple DES encryption/decryption algorithm
Base64	Method used to encode certificate requests and certificates before they are sent to or from the CA
CA	Certificate authority; an organization that creates digital certificates
Certificate	Binds a person or entity to a public key using a digital signature

Table 15: Digital Certificate Terms and Acronyms (continued)

Term or Abbreviation	Description
CRL	Certificate revocation list; a list of certificates that a CA has revoked
ESP	Encapsulating Security Payload; provides data integrity, data confidentiality and, optionally, sender's authentication
IKE	Internet Key Exchange
PKCS	Public-Key Cryptography Standards; a series of standards established by RSA Laboratories
PKCS10	PKCS #10; describes a syntax for certification requests
Root CA	CA that signs the certificates of subordinate CAs
Root certificate	Self-signed public key certificate for a root CA; root certificates are used to verify other certificates
RSA	Rivest-Shamir-Adleman encryption algorithm
SA	Security association; the set of security parameters that dictate how IPSec processes a packet, including encapsulation protocol and session keys. A single secure tunnel uses multiple SAs.
SCEP	Simple certificate enrollment protocol; used to submit requests and to download certificates and CRLs

Platform Considerations

Digital certificates are supported on all ERX routers that support configuration of IPSec.

For information about modules that support IPSec on ERX-14xx models, ERX-7xx models, and the ERX-310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support IPSec.

NOTE: The E120 router and the E320 router do not support configuration of IPSec and digital certificates.

References

For information about digital certificates, see the following references:

- RFC 2409—The Internet Key Exchange (IKE) (November 1998)
- RFC 2459—Internet X.509 Public Key Infrastructure Certificate and CRL Profile (January 1999)
- RFC 2986—PKCS #10: Certification Request Syntax Specification Version 1.7 (November 2000)
- RFC 3280—Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (April 2002)
- RFC 3447—Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1 (February 2003)

For more information about IPSec and IKE, see *Chapter 6, Configuring IPSec*.

IKE Authentication with Digital Certificates

As part of the IKE protocol, one security gateway needs to authenticate another security gateway to make sure that IKE SAs are established with the intended party. The router supports two authentication methods:

- Digital certificates (using RSA algorithms)

  For digital certificate authentication, an initiator signs message interchange data using his private key, and a responder uses the initiator's public key to verify the signature. Typically, the public key is exchanged via messages containing an X.509v3 certificate. This certificate provides a level of assurance that a peer's identity—as represented in the certificate—is associated with a particular public key. E-series routers provide both an offline (manual) and an online (automatic) process when using digital certificates.

- Preshared keys

  With preshared key authentication, the same secret must be configured on both security gateways before the gateways can authenticate each other.

The following sections provide information about digital certificates. For information about using preshared keys, see *IKE Overview* on page 156.

You can also use public keys for RSA authentication without having to obtain a digital certificate. For details, see *IKE Authentication Using Public Keys Without Digital Certificates* on page 238.

Signature Authentication

The following are key steps for using public key cryptography to authenticate a peer. These steps are described in more detail in the following sections.

1. Generating a private/public key pair

   Before the router can place a digital signature on messages, it requires a private key to sign, and requires a public key so that message receivers can verify the signature.

2. Obtaining a root CA certificate

   The router requires at least one root CA certificate to send to IKE peers and also to verify that a peer's certificate is genuine.

3. Obtaining a public key certificate

   The router requires at least one public key certificate, which binds the router identity to its public key. The CA verifies the identity represented on the certificate and then signs the certificate. The router sends the certificate to IKE peers during negotiations to advertise the router public key.

4. Authenticating the peer

   As part of IKE negotiations, the router receives its peer's digital signature in a message exchange. The router must verify the digital signature by using the peer's public key. The public key is contained in the peer's certificate, which often is received during the IKE negotiation. To ensure that the peer certificate is valid, the router verifies its digital signature by using the CA public key contained in the root CA certificate. The router and its IKE peer require at least one common trusted root CA for authentication to work.

Generally, only Step 4 is required each time a phase 1 negotiation happens. The first three steps are required only if keys are compromised or router certificates require renewal.

Generating Public/Private Key Pairs

The ERX router needs at least one valid pair of public/private keys whenever it uses any of the public key methods for authenticating an IKE peer. The ERX router can generate its own public/private key pairs. The public/private key pair supports the RSA standard (1024 or 2048 bits).

The private key is used only by the ERX router. It is never exchanged with any other nodes. It is used to place a digital signature on IKE authentication messages. When generated, it is securely stored internally to the ERX router in nonvolatile storage (NVS). Access to the private key is never allowed, not even to a system administrator or a network management system. Private key storage includes protection mechanisms to prevent improper private key usage, including encryption with 3DES using a unique internally generated key. The key is also tied to SRP-specific data to prevent swapping flash disks between routers.

The public key is used in the generation of the router certificate request, which is sent to a CA. Based on the certificate request, the CA generates a public key certificate for the E-series router.

The router public/private key pair is a global system attribute. It does not matter how many IPSec Service modules (ISMs) exist in the router; only one set of keys is available at any given moment. The private/public key pair applies across all virtual routers and is persistent across reloads and booting to factory defaults.

Obtaining a Root CA Certificate

The ERX router enables the use of either a manual or automatic method to download the root CA's self-signed certificate. The standards supported for obtaining root CAs are X.509v3, base64, and basic-encoding-rules (BER)–encoded certificates.

In the manual method, an operator obtains the root CA certificate, typically through a Web browser, and copies the certificate file to the E-series router so that the router can use it as part of IKE negotiations.

In the automatic method, the router uses SCEP and HTTP to authenticate with the CA and retrieve the certificate. The requested root CA certificate is automatically downloaded to the router.

Obtaining a Public Key Certificate

After the public key is generated, the router must obtain a public key certificate from a CA, a process called certificate enrollment. The procedure to obtain public keys depends on whether the offline or online digital certificate process is being used.

The standards supported for certificate enrollment are PKCS #10 certificate requests, PKCS #7 responses, and X.509v3 certificates. For manual enrollment, certificates are encoded in base64 (MIME) so that the files are easily transferred through cut-and-paste operations and e-mail.

Offline Certificate Enrollment

Offline certificate enrollment works as follows:

1. An operator generates a certificate request by supplying identity information.
2. The ERX router creates a certificate request file and makes it available to the operator.
3. The operator supplies the certificate request file to a CA for approval, typically by copying and pasting the file to a Web page.
4. The CA approves the request and generates a certificate.
5. The operator copies the certificate file onto the ERX router so that it can be used for IKE negotiations.

Online Certificate Enrollment

Online certificate enrollment works as follows:

NOTE: The ERX router must have a root CA certificate for the specified CA before online certificate enrollment.

- The router uses SCEP and HTTP to enroll with the specified CA and retrieve the certificate that the router uses in IKE negotiations.

Authenticating the Peer

The ERX router validates X.509v3 certificates from the peer by confirming that the ID payload passed in IKE matches the identifiers in the peer certificate. The router also verifies that the signature is correct, based on the root CA public key.

The ERX router also validates the certificate based on its time window, so correct UTC time on the router is essential. In addition to the certificate checks, the router confirms that message data received from the peer has the correct signature based on the peer's public key as found in its certificate. After the IKE authentication is done, quick-mode negotiation of SAs can proceed.

Verifying CRLs

You can control how the router handles CRLs during negotiation of IKE phase 1 signature authentication. Both the offline and online digital certificate processes enable you to verify CRLs.

To verify CRLs in the offline certificate process, you must copy CRL files that are published by CAs to the ERX router. Using the **ipsec crl** command, you can control how the router handles CRLs during negotiation of IKE phase 1 signature authentication.

In the online certificate method you use the **crl** command to control CRL verification. The router uses HTTP to support CRL verification when the CRL distribution point that appears in the certificate has an http://*name* Uniform Resource Indicator (URI) format.

The **ipsec crl** and **crl** commands have three possible settings:

- Ignored—Allows negotiations to succeed even if a CRL is invalid or the peer's certificate appears in the CRL; this is the most lenient setting.
- Optional—If the router finds a valid CRL, the router uses it.
- Required—Requires a valid CRL, and the certificates belonging to the E-series router or the peer must not appear in the CRL; this is the strictest setting.

Based on the CRL setting, you can expect the phase 1 IKE negotiations to succeed or fail depending on the following conditions:

- CRL OK—The certificate revocation list is present for the CA and valid (not expired).
- CRL expired—The CRL is present on the ERX router but is expired.
- Missing CRL—There is no CRL on the router for the CA.
- Peer Cert revoked—The CRL contains the peer certificate.
- ERX Cert revoked—The CRL contains the E-series router's certificate.

Table 16 presents how the CRL setting affects the outcome of IKE phase 1 negotiations. It lists common problem conditions such as ERX Cert revoked.

Table 16: Outcome of IKE Phase 1 Negotiations

	CRL Setting		
Condition	**Ignored**	**Optional**	**Required**
CRL OK	Succeed	Succeed	Succeed
CRL expired	Succeed	Succeed	Fail
Missing CRL	Succeed	Succeed	Fail
Peer Cert revoked	Succeed	Fail	Fail
ERX Cert revoked	Succeed	Fail	Fail

File Extensions

Table 17 describes the file extensions that the ERX routers use for digital certificates that are created by the offline process.

During the online digital certificate process, the certificate files are kept in NVS in hidden areas and are not visible to users (the files do not appear when you enter a **dir** shell command). Use the **show** commands to display information for the online certificate files. The router's private keys are similarly hidden from users.

Table 17: File Extensions (Offline Configuration)

File Extension	Description
.crq	Used for certificate request files that are generated on the ERX router and taken to CAs for obtaining a certificate.
.cer	Used for public certificate files. The public certificates for root CAs and the router public certificates are copied to the ERX router. They are automatically recognized as belonging to the ERX router or CA by certificate subject name and issuer name (in a CA they are the same). The ERX router supports multiple CAs.
.crl	Used for certificate revocation lists that are obtained offline from CAs and copied to the ERX router. CRLs indicate which certificates from a particular CA are revoked.

Certificate Chains

In a basic CA model, there is a single CA from which the ERX router obtains the root CA certificates and the router's public key certificates. The E-series router also supports CA hierarchies, which consist of a top-level root CA and one or more sub-CAs (also called issuing CAs).

In a CA hierarchy, the router obtains its public key certificates and the CA certificate from a sub-CA. The sub-CA's certificate is signed by the root CA.

This process creates a certificate chain of trust in which the E-series router must verify all certificates in the chain until the router reaches a trusted CA, such as the root CA. For example, if the router receives traffic from a peer with a certificate signed by a sub-CA, the router first verifies the sub-CA's signature on the peer's certificate, then verifies the sub-CA's certificate, which is signed by the trusted root CA.

The ERX router supports CA hierarchies consisting of the root CA and one level of sub-CAs. When using a CA hierarchy, the router authenticates and enrolls for its public certificate with the sub-CA. When you use the **show ipsec ike-certificates** command, the root CA and sub-CA certificates are listed as CA certificates, and the router's public certificates are signed by the sub-CA.

IKE Authentication Using Public Keys Without Digital Certificates

During IKE negotiations, peers exchange public keys to authenticate each other's identity and to ensure that IKE SAs are established with the intended party. Typically, public keys are exchanged in messages containing an X.509v3 digital certificate.

As an alternative to setting up digital certificates, you can configure and exchange public keys for IKE peers and use these keys for RSA signature authentication *without* having to obtain a digital certificate. This method offers the simplicity and convenience of using preshared key authentication without its inherent security risks.

With this method, you no longer need a digital certificate to do the following:

- Associate the router with its own public key
- Enable a remote peer to display the router's public key
- Learn the remote peer's public key

Configuration Tasks

To set up public keys and peer public keys without obtaining a digital certificate, you use router commands to perform the following tasks:

- Display the router's public key by using the **show ipsec key mypubkey rsa** command. You can use the output from this command to provide information to the remote peer about the public key configured on the router. The remote peer can then enter the router's public key on its own system.
- Manually enter the public key for the remote peer with which you want to establish IKE SAs by using the **ipsec key pubkey-chain rsa** and **key-string** commands.
- Display the remote peer's public key by using the **show ipsec key pubkey-chain rsa** command.

For instructions on setting up peer public keys without a digital certificate, see *Configuring Peer Public Keys Without Digital Certificates* on page 250.

Public Key Format

RSA encryption and authentication require the use of a public key on both the ERX router and on the remote peer with which the router seeks to establish IKE SAs.

The length of the public key can be 1024 bits or 2048 bits, and the format conforms to the RSA standard defined in RFC 3447—Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1 (February 2003).

The public key consists of three components:

- Abstract Syntax Notation 1 (ASN.1) header information
- RSA public key modulus
- RSA public key exponent

In the following example of a 1024-bit public key, the first portion of the key (shown in **bold** typeface) represents the ASN.1 header information. The second portion of the key (shown in regular typeface) represents the RSA public key modulus. The third portion of the key (shown in **bold** typeface) represents the RSA public key exponent.

```
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00A7E43C
3E2D399F 34EF6E16 F84464A9 8A145997 CC7F34C8 3DFF8216 57780FE9 D5CE2717
86239050 7A331044 EBA90120 EC13A78D C1B24285 333A9193 D94A59C8 492D8CB9
A46403A4 37461E00 768CF45C 580211AC 72793764 51E3AB3C F9A6665E 562E3681
F120405E 30235690 6FC093AA EB0FE956 51C38EE1 54D81E40 7687C387 07020301
0001
```

For more information about the format of an RSA public key and about ASN.1 syntax, see RFC 3447—Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1 (February 2003).

Configuring Digital Certificates Using the Offline Method

To use the offline method to set up digital certificates on the router:

1. Generate RSA key pairs.

```
host1(config)#ipsec key generate rsa 2048
Please wait.....................................................
.....................
IPsec Generate Keys complete
```

2. In your IKE policy, set the authentication method to RSA signatures.

```
host1(config)#ipsec ike-policy-rule 1
host1(config-ike-policy)#authentication rsa-sig
host1(config-ike-policy)#exit
host1(config)#
```

NOTE: For more information about setting up IKE policies, see *Defining an IKE Policy* in *Chapter 6, Configuring IPSec*.

3. Enter IPSec Identity Configuration mode.

```
host1(config)#ipsec identity
host1(config-ipsec-identity)#
```

4. Specify the information that the router uses to generate a certificate request.

   a. Specify a country name.

```
host1(config-ipsec-identity)#country CA
```

   b. Specify a common name.

```
host1(config-ipsec-identity)#common-name Jim
```

   c. Specify a domain name.

```
host1(config-ipsec-identity)#domain-name myerx.kanata.junipernetworks.com
```

   d. Specify an organization.

```
host1(config-ipsec-identity)#organization juniperNetworks
host1(config-ipsec-identity)#exit
host1(config)#
```

5. Generate a certificate request using certificate parameters from the IPSec identity configuration.

```
host1(config)#ipsec certificate-request generate rsa myrequest.crq
```

6. After the certificate request is generated, you need to copy the file from the router and send it to the CA. Typically, you copy the file and paste it to a CA's Web page.

7. When you receive the certificate from the CA, copy the certificate to the router, and then inform the router that the new certificate exists.

   host1(config)#**ipsec certificate-database refresh**

8. (Optional) Set the sensitivity of how the router handles CRLs.

   host1(config)#**ipsec crl ignored**

9. (Optional) To delete RSA key pairs, use the **ipsec key zeroize** command.

   host1(config)#**ipsec key zeroize rsa**

authentication

- Use to specify the authentication method that the router uses. For digital certificates, the method is set to RSA signature.
- Example

  host1(config-ike-policy)#**authentication rsa-sig**

- Use the **no** version to restore the default, preshared keys.

common-name

- Use to specify a common name used to generate certificate requests.
- Example

  host1(config-ipsec-identity)#**common-name Jim**

- Use the **no** version to remove the common name.

country

- Use to specify a country name used to generate certificate requests.
- Example

  host1(config-ipsec-identity)#**country CA**

- Use the **no** version to remove the country name.

domain-name

- Use to specify the domain name that the router uses in IKE authentication messages and to generate certificate requests.
- The domain name is used in the SubjectAlternative DNS certificate extensions and as an FQDN (fully qualified domain name) ID payload for IKE negotiations.
- Example

  host1(config-ipsec-identity)#**domain-name myerx.kanata.junipernetworks.com**

- Use the **no** version to remove the domain name.

ike crl

- Use to control how the router handles CRLs during negotiation of IKE phase 1 signature authentication. Specify one of the following keywords:
  - **ignored**—Allows negotiations to succeed even if a CRL is invalid or the peer's certificate appears in the CRL; this is the most lenient setting
  - **optional**—If the router finds a valid CRL, it uses it; this is the default setting
  - **required**—Requires a valid CRL; either the certificates that belong to the E-series router or the peer must not appear in the CRL; this is the strictest setting
- Example

  host1(config)#**ike crl ignored**

- Use the **no** version to return the CRL setting to the default, optional.

NOTE: This command has been replaced by the **ipsec crl** command and may be removed completely in a future release.

ipsec certificate-database refresh

- Use to inform the ERX router that a public key certificate has been copied to the router. The router then verifies public certificates found on its disk against its private key and prepares the certificates for use.

NOTE: On reload, the router scans all certificate files and determines which files are router public certificates and which are root CA certificates.

- Example

  host1(config)#**ipsec certificate-database refresh**

- There is no **no** version.

ipsec certificate-request generate

- Use to cause the router to generate a certificate request using certificate parameters from the IPSec identity configuration.
- Include a name for the certificate request file. The file name must have a .crq extension.
- After the router generates the certificate, use offline methods to send the certificate request file to the CA.
- Example

  host1(config)#**ipsec certificate-request generate rsa myrequest.crq**

- There is no **no** version.

ipsec crl

- Use to control how the router handles CRLs during negotiation of IKE phase 1 signature authentication. Specify one of the following keywords:
  - **ignored**—Allows negotiations to succeed even if a CRL is invalid or the peer's certificate appears in the CRL; this is the most lenient setting
  - **optional**—If the router finds a valid CRL, it uses it; this is the default setting
  - **required**—Requires a valid CRL; either the certificates that belong to the E-series router or the peer must not appear in the CRL; this is the strictest setting
- Example

  host1(config)#**ipsec crl ignored**

- Use the **no** version to return the CRL setting to the default, optional.

NOTE: This command replaces the **ike crl** command, which may be removed completely in a future release.

ipsec identity

- Use to enter IPSec Identity Configuration mode in which you can specify information that the router uses in certificate requests and during negotiations with its peers.
- Example

  host1(config)#**ipsec identity**
  host1(config-ipsec-identity)#

- Use the **no** version to remove the identity configuration.

ipsec ike-policy-rule

- Use to define an ISAKMP/IKE policy.
- When you enter the command, you include a number that identifies the policy and assigns a priority to the policy. You can number policies in the range 1–10000, with 1 having the highest priority.
- Example

  host1(config)#**ipsec ike-policy-rule 3**
  host1(config-ike-policy)#

- Use the **no** version to remove policies. If you do not include a priority number with the **no** version, all policies are removed.

NOTE: This command replaces the **ipsec isakmp-policy-rule** command, which may be removed completely in a future release.

ipsec isakmp-policy-rule

- Use to define an ISAKMP/IKE policy.
- When you enter the command, you include a number that identifies the policy and assigns a priority to the policy. You can number policies in the range 1–10000, with 1 having the highest priority.
- Example

  host1(config)#**ipsec isakmp-policy-rule 3**
  host1(config-ike-policy)#

- Use the **no** version to remove policies. If you do not include a priority number with the **no** version, all policies are removed.

NOTE: This command has been replaced by the **ipsec ike-policy-rule** command and may be removed completely in a future release.

ipsec key generate

- Use to generate RSA key pairs. Include a length of either 1024 or 2048 bits. The generated keys can be used only after the CA issues a certificate for them.
- Example

  host1(config)#**ipsec key generate rsa 2048**
  Please wait.....
  .....
  IPsec Generate Keys complete

- There is no **no** version. To remove a key pair, use the **ipsec key zeroize** command.

ipsec key zeroize

- Use to delete RSA key pairs. Include one of the following keywords:
  - **rsa**—Removes the RSA key pair from the router
  - **pre-share**—Removes all preshared keys from the router
  - **all**—Removes all keys within the VR context from the router
- Example

  host1(config)#**ipsec key zeroize rsa**

- There is no **no** version.

organization

- Use to specify the organization used in the Subject Name field of certificates.
- Example

  host1(config-ipsec-identity)#**organization juniperNetworks**

- Use the **no** version to remove the organization name.

Configuring Digital Certificates Using the Online Method

To use the online configuration method to set up digital certificates on the router:

1. Generate the RSA key pair.

   ```
   host1(config)#ipsec key generate rsa 2048
   Please wait.....................................................
   ..........................
   IPsec Generate Keys complete
   ```

2. In your IKE policy, set the authentication method to RSA signatures.

   ```
   host1(config)#ipsec ike-policy-rule 1
   host1(config-ike-policy)#authentication rsa-sig
   host1(config-ike-policy)#exit
   ```

NOTE: For more information about setting up IKE policies, see *Defining an IKE Policy* in *Chapter 6, Configuring IPSec*.

3. Enter IPSec CA Identity Configuration mode, and specify the name of the certificate authority.

   ```
   host1(config)#ipsec ca identity trustedca1
   host1(config-ca-identity)#
   ```

4. Specify the name of the CA issuer.

   ```
   host1(config-ca-identity)#issuer-identifier BetaSecurityCorp
   ```

5. Specify the URL of the SCEP server from which the CA certificates and the router's public certificates is retrieved.

   ```
   host1(config-ca-identity)#enrollment url http://192.168.99.105/scepurl
   ```

6. (Optional) Set the sensitivity of how the router handles CRLs.

   ```
   host1(config-ca-identity)#crl ignored
   ```

7. (Optional) Specify the wait period between certificate request retries.

   ```
   host1(config-ca-identity)#enrollment retry-period 5
   ```

8. (Optional) Specify the absolute time limit on enrollment.

   ```
   host1(config-ca-identity)#enrollment retry-limit 60
   ```

9. (Optional) Specify the URL of your network's HTTP proxy server.

   ```
   host1(config-ca-identity)#root proxy url http://192.168.5.45
   host1(config-ca-identity)#exit
   ```

10. Retrieve the CA certificate.

    host1(config)#**ipsec ca authenticate trustedca1**

11. Enroll with the CA and retrieve the router's certificate from the CA.

    host1(config)#**ipsec ca enroll trustedca1 My498pWd**

12. (Optional) To delete RSA key pairs, use the **ipsec key zeroize** command.

authentication

- Use to specify the authentication method that the router uses. For digital certificates, the method is set to RSA signature.
- Example

  host1(config-ike-policy)#**authentication rsa-sig**

- Use the **no** version to restore the default, preshared keys.

crl

- Use to control how the router handles certificate revocation lists (CRLs) during negotiation of online IKE phase 1 signature authentication. Specify one of the following keywords:
  - **ignored**—Allows negotiations to succeed even if a CRL is invalid or the peer's certificate appears in the CRL; this is the most lenient setting
  - **optional**—If the router finds a valid CRL, it uses it; this is the default setting
  - **required**—Requires a valid CRL; either the certificates that belong to the E-series router or the peer must not appear in the CRL; this is the strictest setting
- Example

  host1(config-ca-identity)#**crl ignored**

- Use the **no** version to return the CRL setting to the default, optional.

enrollment retry-limit

- Use to set the time period during which the router continues to send a certificate request to the CA. You can specify a time period in the range 0–480 minutes, with 0 specifying an infinite time period.
- Example

  host1(config-ca-identity)#**enrollment retry-limit 200**

- Use the **no** version to restore the default of 60 minutes.

enrollment retry-period

- Use to set the number of minutes that the router waits after receiving no response before resending a certificate request to the CA. You can specify a wait period in the range 0–60 minutes.
- Example

  host1(config-ca-identity)#**enrollment retry-period 40**

- Use the **no** version to restore the default, 1 minute.

enrollment url

- Use to specify the URL of the SCEP server, in the format http://*server_ipaddress*. You can then use the **ipsec ca authentication** command to retrieve CA certificates from the SCEP server, and the **ipsec ca enroll** command to retrieve the router's public key certificates from the server.
- Example

  host1(config-ca-identity)#**enrollment url http://192.168.99.105/scepurl**

- Use the **no** version to delete the enrollment URL specification.

ipsec ca authenticate

- Use to retrieve the specified CA's certificate. If authentication is successful, the fingerprint is sent, and an ikeEnrollment message is logged at severity info.
- The CA must be previously declared by the **ipsec ca identity** command.
- Example

  host1(config)#**ipsec ca authenticate trustedca1**
  host1(config)#INFO 10/18/2003 03:45:16 ikeEnrollment (): Received CA certificate for ca:trustedca1
  INFO 10/18/2003 03:45:16 ikeEnrollment (): Received CA certificate for ca:trustedca1 fingerprint:28:19:ba:76:d8:e0:bb:22:60:cd:b9:2d:dc:b8:58:01
  host1(config)#

- Use the **no ipsec ca identity** command for the specified CA, or boot the router using the factory defaults to remove the CA certificate that was generated during the online configuration.
- There is no **no** version.

ipsec ca enroll

- Use to enroll with the specified CA and to retrieve the router's public key certificate during online digital certificate configuration. If enrollment is successful, the CA sends the certificate to the router and logs an ikeEnrollment message is logged at severity info.
- Use the password option, if required by the CA, to access the CA and enable enrollment.
- The CA must be previously declared by the **ipsec ca identity** command.

- Example

  host1(config)#**ipsec ca enroll trustedca1 My498pWd**
  host1(config)#INFO 10/18/2003 03:49:33 ikeEnrollment (): Received erx certificate for ca:trustedca1
  host1(config)#

- Use the **no ipsec ca identity** command for the specified CA or boot the router using the factory defaults to remove the router's public certificate that was generated during the online configuration.
- There is no **no** version.

ipsec ca identity

- Use to specify the CA that the ERX router uses for online certificate requests and to enter IPSec Identity Configuration mode.
- In IPSec Identity Configuration mode you specify information that the router uses in certificate requests and during negotiations with its peers.
- Example

  host1(config)#**ipsec ca identity trustedca1**
  host1(config-ipsec-identity)#

- Use the **no** version to remove the identity configuration.

ipsec ike-policy-rule

- Use to define an ISAKMP/IKE policy.
- When you enter the command, you include a number that identifies the policy and assigns a priority to the policy. You can number policies in the range 1–10000, with 1 having the highest priority.
- Example

  host1(config)#**ipsec ike-policy-rule 3**
  host1(config-ike-policy)#

- Use the **no** version to remove policies. If you do not include a priority number with the **no** version, all policies are removed.

NOTE: This command replaces the **ipsec isakmp-policy-rule** command, which may be removed completely in a future release.

ipsec isakmp-policy-rule

- Use to define an ISAKMP/IKE policy.
- When you enter the command, you include a number that identifies the policy and assigns a priority to the policy. You can number policies in the range 1–10000, with 1 having the highest priority.
- Example

```
host1(config)#ipsec isakmp-policy-rule 3
host1(config-ike-policy)#
```

- Use the **no** version to remove policies. If you do not include a priority number with the **no** version, all policies are removed.

NOTE: This command has been replaced by the **ipsec ike-policy-rule** command and may be removed completely in a future release.

ipsec key generate

- Use to generate RSA key pairs. Include a length of either 1024 or 2048 bits. The generated keys can be used only after the CA issues a certificate for them.
- Example

```
host1(config)#ipsec key generate rsa 2048
Please wait.....................................................
.....................
IPsec Generate Keys complete
```

- There is no **no** version. To remove a key pair, use the **ipsec key zeroize** command.

ipsec key zeroize

- Use to delete RSA key pairs. Include one of the following keywords:
  - **rsa**—Removes the RSA key pair from the router
  - **pre-share**—Removes all preshared keys from the router
  - **all**—Removes all keys within the VR context from the router
- Example

```
host1(config)#ipsec key zeroize rsa
```

- There is no **no** version.

issuer-identifier

- Use to specify the name of the CA issuer for online digital certificate configuration. The identifier and the enrollment URL specified by the **enrollment url** command are used together to create the CA authentication requests.
- Example

```
host1(config-ca-identity)#issuer-identifier BetaSecurityCorp
```

- Use the **no** version to remove the name from the configuration.

root proxy url

- Use to specify an HTTP proxy server that can submit HTTP requests on the E-series router's behalf to retrieve the root CA certificate. Use this command if your network has an HTTP proxy server installed between the E-series router and the Internet. Use the format http://*server_ipaddress* to specify the URL of the proxy server.
- Example

  host1(config-ca-identity)#**root proxy url http://192.168.5.45**

- Use the **no** version to remove the root proxy URL from the configuration.

Configuring Peer Public Keys Without Digital Certificates

During IKE negotiations, peers exchange public keys to authenticate each other's identity and to ensure that IKE SAs are established with the intended party. Typically, public keys are exchanged in messages containing an X.509v3 digital certificate. As an alternative, however, you can configure and exchange peer public keys and use them for RSA authentication *without* having to obtain a digital certificate.

To configure and exchange peer public keys without obtaining a digital certificate:

1. Generate the RSA key pair on the router.

   host1(config)#**ipsec key generate rsa 1024**
   Please wait...
   IPsec Generate Keys complete

2. In your IKE policy, set the authentication method to RSA signature.

   host1(config)#**ipsec ike-policy-rule 1**
   host1(config-ike-policy)#**authentication rsa-sig**
   host1(config-ike-policy)#**exit**
   host1(config)#**exit**

NOTE: For more information about setting up IKE policies, see *Defining an IKE Policy* in *Chapter 6, Configuring IPSec*.

3. Display the router's public key.

   ```
   host1#show ipsec key mypubkey rsa
    30819f30 0d06092a 864886f7 0d010101 05000381 8d003081 89028181 00daaa65
    8082ac0a ec42e552 10e3489b 37463ed8 9bfa2541 f46a7b30 0e908749 5b652ae5
    ae604e9a 81bc3268 270e7f68 69ffd2a8 be268afa 92849fd0 4e8c96be 3eddf1c2
    12d9fe7a 68e8507c 99b59ff3 bb0c3942 b0a90c76 3ae3acbb 4a777037 31527ea0
    23693bdc e5393c6f 2ef3e7e7 bb1a308e d42ce0ad a095273e d718384c dd020301
    0001
   ```

   For information about the format of an RSA public key, see *Public Key Format* on page 239.

4. Use the output from the **show ipsec key mypubkey rsa** command to provide information to the remote peer about the public key configured on the E-series router. Providing this information enables the remote peer to enter the router's public key on its own system.

   The **show ipsec key mypubkey rsa** command enables you to display the contents of the router's public key without having to obtain a digital certificate.

5. Obtain the public key from the remote peer.

   For example, you might receive an e-mail message from the remote peer containing the public key information.

6. Configure the public key for the remote IKE peer.

   a. Access IPSec Peer Public Key Configuration mode.

      You must identify the remote peer associated with the public key by specifying the remote peer's IP address, fully qualified domain name (FQDN), or FQDN preceded by an optional *user@* specification. For example, the following command enables you to enter the peer public key for the remote peer identified by IP address 192.168.15.5.

      ```
      host1(config)#ipsec key pubkey-chain rsa address 192.168.15.5
      host1(config-peer-public-key)#
      ```

   b. Enter the peer public key that you obtained in Step 5.

```
host1(config-peer-public-key)#key-string "
Enter remainder of text message.  End with the character '"'.
 30820122 300d0609 2a864886 f70d0101 01050003 82010f00 3082010a 02820101
 00effc6f d91cbf23 5de66454 420db27a 0bacfc92 63a54e60 587c3e1c 951be4e8
 09e7d130 da924040 0ceb797c ddc0df10 dabeb3fc a17145ff 6e7ff977 68ac0698
 748d30f4 478252ed 29bf3e4e a6657cc8 cfaf1de4 e7dc2473 33231286 0ecfb15b
 4aac505b 255f17ca faf884ca f0402022 5ad6f446 e0f3fb1e d48bbc00 5d4fe9b6
 35f88b53 1bf4f07c b168e47b b7143181 5bad4586 0abb7b03 6dba9668 b45e3714
 0b64ca82 3a53f69b 357a7d41 f512da37 71901b14 08212648 277f6d38 6bc34164
 8c3ac8d4 d9c8baac dc006dac 8c09ce37 44a5d124 b69fec24 df0fc3a8 98e6efc8
 5a1d65eb e4b832ba adc26c63 1996fe37 e797ecff 6e2acdd6 0981ef2c 3dd2f506
 01020301 0001"
```

   c. (Optional) Verify the peer public key configuration.

```
host1#show ipsec key pubkey-chain rsa address 192.168.15.5

 30820122 300d0609 2a864886 f70d0101 01050003 82010f00 3082010a 02820101
 00effc6f d91cbf23 5de66454 420db27a 0bacfc92 63a54e60 587c3e1c 951be4e8
 09e7d130 da924040 0ceb797c ddc0df10 dabeb3fc a17145ff 6e7ff977 68ac0698
 748d30f4 478252ed 29bf3e4e a6657cc8 cfaf1de4 e7dc2473 33231286 0ecfb15b
 4aac505b 255f17ca faf884ca f0402022 5ad6f446 e0f3fb1e d48bbc00 5d4fe9b6
 35f88b53 1bf4f07c b168e47b b7143181 5bad4586 0abb7b03 6dba9668 b45e3714
 0b64ca82 3a53f69b 357a7d41 f512da37 71901b14 08212648 277f6d38 6bc34164
 8c3ac8d4 d9c8baac dc006dac 8c09ce37 44a5d124 b69fec24 df0fc3a8 98e6efc8
 5a1d65eb e4b832ba adc26c63 1996fe37 e797ecff 6e2acdd6 0981ef2c 3dd2f506
 01020301 0001
```

authentication

- Use to specify in the ISAKMP/IKE policy that the router uses the RSA signature authentication method for IKE negotiations.
- Example

  host1(config-ike-policy)#**authentication rsa-sig**

- Use the **no** version to restore the default authentication method, preshared keys.

ipsec ike-policy-rule

- Use to access IPSec IKE Policy Configuration mode to define an ISAKMP/IKE policy.
- For information about how to use this command, see **ipsec ike-policy-rule** on page 243.
- Example

  host1(config)#**ipsec ike-policy-rule 2**
  host1(config-ike-policy)#

- Use the **no** version to remove policies. If you do not include a priority number with the **no** version, all policies are removed.

ipsec key generate

- Use to generate a 1024-bit or 2048-bit RSA key pair.
- Example

  host1(config)#**ipsec key generate rsa 2048**
  Please wait.....
  .....
  IPsec Generate Keys complete

- There is no **no** version. To remove a key pair, use the **ipsec key zeroize** command.

ipsec key pubkey-chain rsa

- Use to access IPSec Peer Public Key Configuration mode to configure the public key for a remote peer with which you want to establish IKE SAs.
- The **ipsec key pubkey-chain rsa** command enables you to manually enter the public key data for the remote peer without having to obtain a digital certificate.
- To specify the IP address of the remote peer associated with the public key, use the **address** keyword followed by the IP address, in 32-bit dotted decimal format.
- To specify the identity of the remote peer associated with the public key, use the **name** keyword followed by either:
  - The fully qualified domain name (FQDN)
  - The FQDN preceded by an optional *user@* specification; this is also referred to as user FQDN format

- The FQDN and user FQDN identifiers are case-sensitive.
- To ensure that the public key is associated with the correct remote peer, the router requires an exact match for the identifier string. For example, a public key for user FQDN mjones@sales.company_abc.com does not match a public key for FQDN sales.company_abc.com.
- From IPSec Peer Public Key Configuration mode, use the **key-string** command to enter the peer public key data. For information about how to use this command, see **key-string** on page 253.
- Example 1—Enables you to configure the public key for a remote peer with IP address 192.168.50.10

  host1(config)#**ipsec key pubkey-chain rsa address 192.168.50.10**
  host1(config-peer-public-key)#

- Example 2—Enables you to configure the public key for a remote peer with the FQDN sales.company_xyz.com

  host1(config)#**ipsec key pubkey-chain rsa name sales.company_xyz.com**
  host1(config-peer-public-key)#

- Example 3—Enables you to configure the public key for a remote peer with the FQDN tsmith@sales.company_xyz.com

  host1(config)#**ipsec key pubkey-chain rsa name tsmith@sales.company_xyz.com**
  host1(config-peer-public-key)#

- Use the **no** version to remove the peer public key from the router.

key-string

- Use to manually enter a 1024-bit or 2048-bit public key for a remote peer with which you want to establish IKE SAs.
- The key string represents the public key hexadecimal data that includes the ASN.1 object identifier and sequence tags for RSA encryption.
- Enter an alphanumeric key string with a maximum of 1999 characters.
- You must use the same character (for example, " or x) at the beginning and end of the string to delimit the key string. The delimiter character is case-sensitive and must not occur anywhere else in the key string.
- For information about the format of an RSA public key, see *Public Key Format* on page 239.
- Example 1—Configures the public key for a remote peer with IP address 192.168.50.10, using " (double quotation marks) as the key string delimiter character

  host1(config)#**ipsec key pubkey-chain rsa address 192.168.50.10**
  host1(config-peer-public-key)#**key-string "**
  Enter remainder of text message. End with the character '"'.
  30819f30 0d06092a 864886f7 0d010101 05000381 8d003081 89028181 00d3a447
  0b997844 213de4ae 13a2c09b f74051cd d404a187 c5e86867 d525cb6e 571a44f2
  92bac7e8 bb282857 fb20357c d94ec241 b651596c 350dd770 6853526b c95e60c1
  52ec06ce 094882a7 4a7275a6 af1b738f 29d1124d 21e49b2a 3b0b7f2f fe31f0cc
  178ddbfe a587a7a9 83aa0601 e86e7de4 3ca78f60 89a758bf 4c1247ba cb020301
  0001"

- Example 2—Configures the public key for a remote peer with the FQDN sales.company_xyz.com, using ‘ (single quotation mark) as the key string delimiter character

```
host1(config)#ipsec key pubkey-chain rsa name sales.company_xyz.com
host1(config-peer-public-key)#key-string '
Enter remainder of text message.  End with the character '''.
30820122 300d0609 2a864886 f70d0101 01050003 82010f00 3082010a 02820101
00c03cc6 0bad55ea b4f8a01f 5cf69de5 f03185e2 1338b5cb fa8418c3 6cbe1a77
bfefba5b 7a8f0ac2 6e2b223b 11e3c316 a30f7fb0 7bd2ab8a a614bb3d 2fce97bf
d6376467 0d5d1a16 d630c173 3ed93434 e690f355 00128ffb c36e72fa 46eae49a
5704eabe 0e34776c 7d243b8b fcb03c75 965c12f4 d68c6e63 33e0207c a985ffff
2422fb53 23d49dbb f7fd3140 a7f245ee bf629690 9356a29c b149451a 691a2531
9787ce37 2601bdf9 1434b174 4fd21cf2 48e10f58 9ac89df1 56e360b1 66fb0b3f
27ad6396 7a491d74 3b8379ea be502979 8f0270b2 6063a474 fadc5f18 f0ca6f7a
ddea66c7 cf637598 9cdb5087 0480af29 b9c174ab 1b1d033f 67641a8c 5918ddce
1f020301 0001'
```

- Example 3—Configures the public key for a remote peer with the user FQDN tsmith@sales.company_xyz.com, using lowercase x as the key string delimiter character

```
host1(config)#ipsec key pubkey-chain rsa name tsmith@sales.company_xyz.com
host1(config-peer-public-key)#key-string x
Enter remainder of text message.  End with the character 'x'.
30819f30 0d06092a 864886f7 0d010101 05000381 8d003081 89028181 00bcc106
8694a505 0b92433e 4c27441e 3ad8955d 5628e2ea 5ee34b0c 6f82c4fd 8d5b7b51
f1a3c94f c4373f9b 70395011 79b4c2fb 639a075b 3d66185f 9cc6cdd1 6df51f74
cb69c8bb dbb44433 a1faac45 10f52be8 d7f2c8cd ad5172a6 e7f14b1c bba4037b
29b475c6 ad7305ed 7c460779 351560c6 344ccd1a 35935ea3 da5de228 bd020301
0001x
```

- There is no **no** version. Use the **no** version of the **ipsec key pubkey-chain rsa** command to remove the peer public key from the router.

Monitoring Digital Certificates and Public Keys

Use the following **show** commands to display information about IKE certificates, IKE configurations, CRLs, public keys, and peer public keys.

show ipsec ca identity

- Use to display information about IKE CA identities used by the router for online digital certificate configuration. You can display information for a specific CA or for all CAs configured on the router.
- Field descriptions
  - CA—Certificate authority that the router uses to generate certificate requests
  - enrollment url—URL of the SCEP server where the router sends certificate requests
  - issuer id—Name of the CA issuer providing the digital certificates
  - retry period—Number of minutes that the router waits after receiving no response from the CA before resending a certificate request

- retry limit—Number of minutes during which the router continues to send a certificate request to the CA
- crl setting—Setting that controls how the router checks the certificate revocation lists
- proxy url—HTTP proxy server used to retrieve the root CA certificate, if any

- Example

```
host1#show ipsec ca identity mysecureca1

CA: mysecureca1 parameters:
enrollment url:http://192.168.10.124/scepurl
issuer id     :BetaSecurityCorp
retry period  :1
retry limit   :60
crl setting   :optional
proxy url     :
```

show ipsec certificates
show ike certificates

NOTE: The **show ike certificates** command has been replaced by the **show ipsec certificates** command and may be removed completely in a future release.

- Use to display the IKE certificates and CRLs on the router. Specify the type of certificate you want to display:
  - **all**—All certificates configured on the router
  - **crl**—Certificate revocation lists
  - **peer**—Peer certificates
  - **public-certs**—Public certificates
  - **root-cas**—Root CA certificates
- Use the **hex-format** keyword to display certificate data, such as serial numbers, in hexadecimal format. Doing so allows easier comparison with CAs, such as Microsoft, that display certificates in hexadecimal format.
- Field descriptions
  - Ca identity—Certificate authority that the router uses to generate certificate requests
  - SubjectName—Distinguished name for the certificate
  - IssuerName—Organization that signed and issued the certificate
  - SerialNumber—Unique serial number assigned to the certificate by the CA
  - SignatureAlgorithm—Algorithm used for the digital signature
  - Validity—Beginning and ending period during which the certificate is valid

- PublicKeyInfo—Information about the public key
- Extensions—Fields that provide additional information for the certificate
- Fingerprints—Unique hash of the certificate, which can be used to verify that the certificate is valid

- Example 1

```
host1#show ipsec certificates public-certs

---------- Public Certificates: ----------

Ca Identity:[trustedca1]Certificate =
  SubjectName = <C=us, O=junipernetworks, CN=jim>
  IssuerName = <C=CA, ST=ON, L=Kanata, O=BetaSecurityCorp, OU=VT Group,
CN=VT Root CA>
  SerialNumber= 84483276204047383658902
  SignatureAlgorithm = rsa-pkcs1-sha1
  Validity =
    NotBefore = 2003 Oct 21st, 16:14:42 GMT
    NotAfter  = 2004 Oct 21st, 16:24:42 GMT
  PublicKeyInfo =
    PublicKey =
      Algorithm name (SSH) : if-modn{sign{rsa-pkcs1-md5}}
      Modulus n  (1024 bits) :

134091279653070615030540500538006424883565376680781606052426226613116 25

198766078066868468220703596586495463741285408762134168585142880305841 24

058965208235335250989603354939442080197472615242413893452088725512650 97

585427735881258246124244228778707000289561722844010730391924576190024 85
        536605332111770428470 2619
      Exponent e (  17 bits) :
        65537
  Extensions =
    Available = authority key identifier, subject key identifier, key usage,
      subject alternative name, authority information access, CRL
distribution
      points
    SubjectAlternativeNames =
      Following names detected =
        DNS (domain name server name)
      Viewing specific name types =
        DNS = host1.kanata.junipernetworks.com
    KeyUsage = DigitalSignature
    CRLDistributionPoints =
      % Entry 1
      FullName =
        Following names detected =
          URI (uniform resource indicator)
        Viewing specific name types =
          URI = http://vtsca1/CertEnroll/VTS%20Root%20CA.crl
      % Entry 2
      FullName =
        Following names detected =
          URI (uniform resource indicator)
        Viewing specific name types =
        No names of type IP, DNS, URI, EMAIL, RID, UPN or DN detected.
    AuthorityKeyID =
      KeyID =
        15:0a:17:4d:36:b6:49:96:fa:d5:be:df:51:3e:e4:90:51:a2:c0:95
```

```
          AuthorityCertificateIssuer =
            Following names detected =
              DN (directory name)
            Viewing specific name types =
            No names of type IP, DNS, URI, EMAIL, RID, UPN or DN detected.
          AuthorityCertificateSerialNumber =
    795928825084374259598581129948925061 78
        SubjectKeyID =
          KeyId =
            78:e0:3e:f7:24:65:2d:4b:01:d4:91:f9:66:c7:67:26:06:74:6c:5c
        AuthorityInfoAccess =
          AccessMethod = 1.3.6.1.5.5.7.48.2
          AccessLocation =
            Following names detected =
              URI (uniform resource indicator)
            Viewing specific name types =
            No names of type IP, DNS, URI, EMAIL, RID, UPN or DN detected.
          AccessMethod = 1.3.6.1.5.5.7.48.2
          AccessLocation =
            Following names detected =
              URI (uniform resource indicator)
            Viewing specific name types =
            No names of type IP, DNS, URI, EMAIL, RID, UPN or DN detected.
      Fingerprints =
        MD5 = c4:c9:22:b6:19:07:4e:4f:ee:81:7a:9f:cb:f9:1f:7e
        SHA-1 = 58:ba:fb:0d:68:61:42:2a:52:7e:19:82:77:a4:55:4c:25:8c:c5:60
```

- Example 2

```
host1#show ipsec certificates root-cas

---------- Root CAs: ----------

Ca Identity:[trustedca1]Certificate =
  SubjectName = <C=CA, ST=ON, L=Kanata, O=Juniper Networks, OU=VTS Group, CN=VTS
Root CA>
  IssuerName = <C=CA, ST=ON, L=Kanata, O=BetaSecurityCorp, OU=VT Group, CN=VT
Root CA>
  SerialNumber= 795928825084374259598581129948925061 78
  SignatureAlgorithm = rsa-pkcs1-sha1
  Certificate seems to be self-signed.
      * Signature verification success.
  Validity =
    NotBefore = 2003 Mar 26th, 15:50:53 GMT
    NotAfter  = 2006 Mar 26th, 15:59:59 GMT
  PublicKeyInfo =
    PublicKey =
      Algorithm name (SSH) : if-modn{sign{rsa-pkcs1-md5}}
      Modulus n  (1024 bits) :
        144248074987660012010604335256719344018162132468668237226501170070305 00
        124141524728006297377738455493108338046539752882464863817590030102246 72
        533705755418539582720728754129158582608340560690539663699122443362882 29
        094433819000056156526315600443048638564217398483268658776617873141444 47
        827650232323210894115707 7
      Exponent e (  17 bits) :
        65537
  Extensions =
    Available = subject key identifier, key usage, basic constraints(critical),

CRL distribution points, unknown
KeyUsage = DigitalSignature NonRepudiation KeyCertSign CRLSign
BasicConstraints =
cA         = TRUE
```

```
[critical]
CRLDistributionPoints =
% Entry 1
FullName =
Following names detected =
URI (uniform resource indicator)
Viewing specific name types =
URI = http://vtsca1/CertEnroll/VTS%20Root%20CA.crl
% Entry 2
FullName =
Following names detected =
URI (uniform resource indicator)
Viewing specific name types =
No names of type IP, DNS, URI, EMAIL, RID, UPN or DN detected.
SubjectKeyID =
KeyId =
15:0a:17:4d:36:b6:49:96:fa:d5:be:df:51:3e:e4:90:51:a2:c0:95
Unknown 1.3.6.1.4.1.311.21.1 =
02:01:00                                              ...
Fingerprints =
MD5 = 8c:56:fb:a6:bd:ab:13:67:e6:13:09:c1:d0:de:1f:24
SHA-1 = 22:3d:84:6d:d4:5f:18:87:ae:2c:15:7d:2a:94:20:ff:c6:12:fb:6f
```

show ipsec identity
show ike identity

NOTE: The **show ike identity** command has been replaced by the **show ipsec identity** command and may be removed completely in a future release.

- Use to display the IKE identity configuration.
- Field descriptions
  - Domain Name—Domain name the router uses in IKE authentication messages and to generate certificate requests
  - Common Name—Common name used to generate certificates
  - Organization—Name of the organization used in the Subject Name field of certificates
  - Country—Country used to generate certificates
- Example

```
host1#show ipsec identity

Ike identity:
        Domain Name :myerx.kanata.junipernetworks.com
        Common Name :jim
        Organization:junipernetworks
        Country     :ca
```

show ipsec ike-configuration
show ike configuration

NOTE: The **show ike configuration** command has been replaced by the **show ipsec ike-configuration** command and may be removed completely in a future release.

- Use to display a summary of the IKE configuration.
- Field descriptions
  - Ike identity—Information from your IKE identify configuration that the router uses to generate certificate requests
  - CRL Check—Setting of the CRL check: optional, required, ignored
- Example

```
host1#show ipsec ike-configuration

Ike configuration:
Ike identity:
        Domain Name :treverxsys2.juniper.net
        Common Name :Sys2 ERX
        Organization:Juniper Networks
        Country     :CA
CRL Check:optional
```

show ipsec key mypubkey rsa

- Use to display the 1024-bit or 2048-bit RSA public key configured on the router.
- The public key is generated as part of a public/private key pair used to perform RSA authentication during ISAKMP/IKE SA negotiations.
- For information about the format of an RSA public key, see *Public Key Format* on page 239.
- Example

```
host1#show ipsec key mypubkey rsa
 30819f30 0d06092a 864886f7 0d010101 05000381 8d003081 89028181 009cfbde
 a16cf72c 49fbd3c1 10d5d9d4 8ba15ec0 9adcb19e 18d488f8 e0370c51 2d10e751
 ddd81be4 dfc78aad 9deb797f b2c51172 18967cfb e18f6efa 69285fef 10337527
 78ca6bbc 907abb9e 44b12713 ab70cb0e a86d9c6c 80c99bd1 e2bf6b70 91222295
 616a88bb cc479e15 be04f3a5 a6160645 844598c3 314b66af 3a8b7602 ed020301
 0001
```

show ipsec key pubkey-chain rsa

- Use to display a 1024-bit or 2048-bit ISAKMP/IKE public key that a remote peer uses for RSA authentication.
- To display a brief summary of the remote peers for which public keys are configured on the router, use the **summary** keyword.
- To display the public key for a remote peer with a specific IP address, use the **address** keyword followed by the IP address, in 32-bit dotted decimal format.
- To display the public key for a remote peer with a specific identity, use the **name** keyword followed by either:
  - The fully qualified domain name (FQDN)
  - The FQDN preceded by an optional *user@* specification; this is also referred to as user FQDN format
- The FQDN and user FQDN identifiers are case-sensitive and must exactly match the identifier specified in the **ipsec key pubkey-chain rsa** command. For example, a public key for user FQDN mjones@sales.company_abc.com does not match a public key for FQDN sales.company_abc.com.
- For information about the format of an RSA public key, see *Public Key Format* on page 239.
- Field descriptions
  - Remote Peer—IP address, FQDN, or user FQDN identifier of the remote peer for which the peer public key can be used
  - Key Type—Type of remote peer identifier: ip address (if IP address is specified) or identity (if FQDN or user FQDN is specified)
- Example 1—Displays a summary of the remote peers for which peer public keys are configured

```
host1#show ipsec key pubkey-chain rsa summary
         Remote Peer               Key Type
-----------------------------     ----------
192.168.32.3                      ip address
grp003.cust535.isp.net            identity
tsmith@grp003.cust535.isp.net     identity
```

- Example 2—Displays the peer public key for a remote peer with the specified IP address

```
host1#show ipsec key pubkey-chain rsa address 192.168.32.3

 30819f30 0d06092a 864886f7 0d010101 05000381 8d003081 89028181 0082065f
 841aa03a fadfda9f bf8be05c d2fe3596 abc3e265 0b86b99a df9b4907 29c7a737
 8bf08491 5c96e72d 28471a12 f0735ff4 04d76ad1 3a80f10c 23dcadda b68ce8ec
 5fdfbe58 a52008db 9a11f867 d38d0483 e4abd53c 89a4dc3c 985ea450 f17748c4
 3f04def0 a3cf5d89 b62dfeae 5990641b 370bb113 73105ba7 585a41fc 3b020301
 0001
```

- Example 3—Displays the peer public key for a remote peer with the specified FQDN identifier

```
host1#show ipsec key pubkey-chain rsa name grp003.cust535.isp.net

30820122 300d0609 2a864886 f70d0101 01050003 82010f00 3082010a 02820101
 00c03cc6 0bad55ea b4f8a01f 5cf69de5 f03185e2 1338b5cb fa8418c3 6cbe1a77
 bfefba5b 7a8f0ac2 6e2b223b 11e3c316 a30f7fb0 7bd2ab8a a614bb3d 2fce97bf
 d6376467 0d5d1a16 d630c173 3ed93434 e690f355 00128ffb c36e72fa 46eae49a
 5704eabe 0e34776c 7d243b8b fcb03c75 965c12f4 d68c6e63 33e0207c a985ffff
 2422fb53 23d49dbb f7fd3140 a7f245ee bf629690 9356a29c b149451a 691a2531
 9787ce37 2601bdf9 1434b174 4fd21cf2 48e10f58 9ac89df1 56e360b1 66fb0b3f
 27ad6396 7a491d74 3b8379ea be502979 8f0270b2 6063a474 fadc5f18 f0ca6f7a
 ddea66c7 cf637598 9cdb5087 0480af29 b9c174ab 1b1d033f 67641a8c 5918ddce
 1f020301 0001
```

- Example 4—Displays the peer public key for a remote peer with the specified user FQDN identifier

```
host1#show ipsec key pubkey-chain rsa name tsmith@grp003.cust535.isp.net

30819f30 0d06092a 864886f7 0d010101 05000381 8d003081 89028181 00bcc106
 8694a505 0b92433e 4c27441e 3ad8955d 5628e2ea 5ee34b0c 6f82c4fd 8d5b7b51
 f1a3c94f c4373f9b 70395011 79b4c2fb 639a075b 3d66185f 9cc6cdd1 6df51f74
 cb69c8bb dbb44433 a1faac45 10f52be8 d7f2c8cd ad5172a6 e7f14b1c bba4037b
 29b475c6 ad7305ed 7c460779 351560c6 344ccd1a 35935ea3 da5de228 bd020301
 0001
```